

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Dissertação de Mestrado

Projeto de Infra-estrutura de TI pela Perspectiva de
Negócio

Filipe Teixeira Marques

Campina Grande, Paraíba, Brasil

Julho - 2006

Projeto de Infra-estrutura de TI pela Perspectiva de Negócio

Filipe Teixeira Marques

Dissertação submetida à Coordenação de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Informática.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Redes de Computadores e Sistemas Distribuídos
(Gestão da Tecnologia de Informação)

Jacques Philippe Sauvé

(Orientador)

Campina Grande, Paraíba, Brasil

©Filipe Teixeira Marques, Julho 2006

UFCG - BIBLIOTECA - CAMPUS I	
4306	27-10-06

M357p.

Marques, Filipe Teixeira

*Projeto de Infra-estrutura de TI pela perspectiva de Negócio /
Filipe Teixeira Marques. – Campina Grande: UFCG / Centro de
Engenharia Elétrica e Informática, 2006.*

xi, 154 f. : il. ; 31 cm.

Orientador: Jacques Philippe Sauvé

*Dissertação (mestrado) – UFCG / CEEI / Programa de
Pós-Graduação em Ciência da Computação, 2006.*

Referências bibliográficas: f. 34-42

1. Redes de Computadores e Sistemas Distribuídos. 2. Gestão da Tecnologia da Informação. 3. Análise e Modelagem de Desempenho. 4. Teoria das Filas. 5. Teoria da Confiabilidade. 6. Ciência da Computação - Tese. I. Sauvé, Jacques Philippe. III. Universidade Federal de Campina Grande, Programa de Pós-Graduação em Ciência da Computação. IV. Projeto de TI pela Perspectiva de Negócio.

UFCG/BC

CDU: 004.7(043)

**“PROJETO DE INFRA-ESTRUTURA DE TI PELA
PERSPECTIVA DE NEGÓCIO”**

FILIFE TEIXEIRA MARQUES

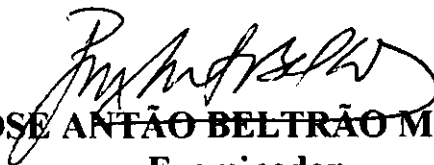
DISSERTAÇÃO APROVADA COM DISTINÇÃO EM 07.07.2006



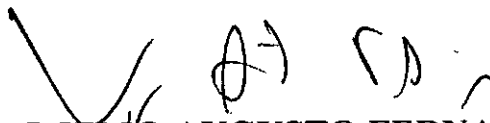
PROF. JACQUES PHILIPPE SAUVÉ, Ph.D
Orientador



PROF. MARCUS COSTA SAMPAIO, Dr.
Examinador



PROF. JOSÉ ANTÃO BELTRÃO MOURA, Ph.D
Examinador



PROF. VIRGÍLIO AUGUSTO FERNANDES ALMEIDA, Ph.D
Examinador

CAMPINA GRANDE – PB

Resumo

Considerando o papel crucial que a Tecnologia da Informação (TI) desempenha ao apoiar e melhorar as operações de negócio, as empresas estão mais dependentes de seus serviços e infra-estrutura. Este fato implica em uma maior atenção para com os serviços de TI, os quais devem ser cuidadosamente projetados e mantidos para suportar as necessidades do negócio. As abordagens convencionais tratam o projeto de infra-estruturas escolhendo a configuração de menor custo que suporte eficientemente os serviços de TI. No entanto, as necessidades do negócio vão muito além do custo da infra-estrutura e precisam ser capturadas e relacionadas às questões técnicas de TI. Esta dissertação apresenta uma investigação formal de técnicas de projeto de infra-estruturas de TI e acordos de níveis de serviços onde as métricas de negócio são levadas em consideração - esta é a principal diferença em relação às abordagens existentes. A abordagem proposta é baseada no conceito de *Business-Driven IT Management* (BDIM), que estabelece uma relação formal entre a infra-estrutura de TI, os serviços de TI e o negócio propriamente dito.

Abstract

Since Information Technology (IT) plays a crucial role in leveraging the business operations, corporations are getting more dependent on their IT services and infrastructure. This high level of criticality has significantly raised the visibility of IT services and these must be designed and provisioned ever more carefully to support business needs. The conventional capacity planning approaches tries to solve this problem by choosing the most cost-effective IT infrastructure configuration alternative. However, the needs of the business must somehow be captured and linked to the IT world so that adequate service be designed and deployed. This dissertation presents a formal investigation of techniques for designing IT infrastructure and Service Level Agreements, where business metrics are considered – that is the main difference from existing approaches. The proposed approach is based on Business-Driven IT Management (BDIM). BDIM establishes a formal link between IT infrastructure, IT services and the business itself.

Agradecimentos

Em primeiro lugar agradeço a Deus.

A minha "verdadeira" família pelo imenso amor, apoio e incentivo recebido durante todo esse período; sem isso não teria conseguido. Aos meus irmãos Rodrigo (vovô), Rafa (hipopotamo) e Flávio pelo companheirismo.

Agradeço imensamente ao professor Jacques, um verdadeiro mestre, pela sua orientação. Também, sou grato aos professores Antão Moura e Marcus Sampaio pelas inúmeras discussões e ensinamentos. Em especial, agradeço também ao professor Dalton pelas palavras certas em um momento crucial.

Aos poucos verdadeiros amigos que, mesmo diante das minhas ausências semanais e no momento de maior escuridão da minha vida, mostraram a essência da verdadeira amizade.

Ao resto do pessoal do grupo Bottom-Line (Rodrigo Rebouças e Rodrigo "Biscoito") pelos incontáveis debates, momentos de descontração e consultorias diversas. Também sou grato aos amigos(as) do mestrado pelo companheirismo, pelos momentos de descontrações e apoio.

A Aninha e Lili, por terem sido tão prestativas e carinhosas.

Agradeço aos amigos Flávio, Fábio (Tio Binho), Gustavo e Lauro, os quais foram meus companheiros de casa em Campina Grande, por terem, literalmente, me "suportado" durante todo esse tempo.

Agradeço a todos aqueles que apesar de já não fazerem parte da minha vida, tiveram fundamentais importância e influência na minha caminhada; em especial, meu eterno tio Hendrik e meus avôs Toinho e Teixeira.

Agradeço em especial a Elise Marianni. Apesar de tudo, deixar de reconhecer sua contribuição seria extremamente injusto diante de seu apoio e incentivo.

Conteúdo

1	Introdução	1
1.0.1	Gerência de Serviços de TI	2
1.0.2	BDIM: uma extensão à Gerência de Serviços de TI	4
1.1	Objetivos	6
1.2	Organização da Dissertação	7
2	Projetando Infra-estruturas de TI e Acordos de Níveis de Serviços	8
2.1	Abstração da Infra-estrutura de TI	11
2.2	Sobre os problemas de projeto de infra-estruturas e SLAs	12
2.3	Sobre o modelo de custo da infra-estrutura de TI	13
2.4	Sobre o modelo particular de impacto negativo	14
2.5	Sobre o modelo de disponibilidade do serviço de TI	15
2.6	Sobre a probabilidade de desistência	15
3	Resultados da Pesquisa	17
3.1	Projeto Ótimo de Infra-Estruturas de TI pela Perspectiva do Negócio	17
3.2	Estudo do Efeito de Picos de Demanda no Projeto Ótimo de Infra-Estruturas de TI pela Perspectiva do Negócio	19
3.3	Negociação Ótima de SLAs Internos pela Perspectiva do Negócio	20
3.4	Negociação Ótima de SLAs Envolvendo Serviços Terceirizados pela Perspectiva do Negócio	21
4	Discussão sobre a Validação dos Modelos	23

5	Conclusões e Trabalhos Futuros	27
5.1	Conclusões	29
5.2	Trabalhos Futuros	31
A	Optimal Design of e-Commerce Site Infrastructure from a Business Perspective	43
A.1	Introduction	44
A.2	Informal Problem Description	45
A.3	Problem Formalization	46
A.3.1	The Design Optimization Problem	46
A.3.2	Characterizing the Infrastructure	47
A.3.3	The Response Time Performance Model	49
A.3.4	The Business Impact Model	54
A.4	An Example E-Commerce Site Design	55
A.5	Related Work	61
A.6	Conclusions	63
B	Business-Oriented Capacity Planning of IT Infrastructure to handle Load Surges	65
B.1	Introduction	66
B.2	Capacity Planning Using a Cost Perspective	66
B.2.1	IT Infrastructure Abstraction	66
B.2.2	Cost-Oriented Capacity Planning Formalization	68
B.3	Adding a Business-Oriented View	69
B.3.1	Handling Load Surges: The Challenge	69
B.3.2	Loss Model Components	70
B.4	Evaluating the Model: an Example	70
B.5	Conclusions	74
C	SLA Design from a Business Perspective	76
C.1	Introduction	77
C.2	Gaining a Business Perspective on IT Operations	77

C.2.1	Addressing IT Problems through Business Impact Management . . .	78
C.2.2	SLA Design: An Optimization Problem	78
C.3	Problem Formalization	79
C.3.1	The Entities and their Relationships	79
C.3.2	The Cost Model	80
C.3.3	Loss Considerations	80
C.3.4	The SLA Design Problem	82
C.3.5	A Specific Loss Model	83
C.3.6	The Availability Model	84
C.3.7	The Response Time Performance Model	84
C.4	A Numerical Example of SLA Design	86
C.5	Related Work	89
C.6	Conclusions	90
D	Business-Driven Design of Infrastructures for IT Services	92
D.1	Introduction: The Problem	92
D.2	A Method to Capture the Business Perspective in IT Infrastructure Design .	95
D.2.1	A Layered Model	95
D.2.2	Optimization problems dealing with infrastructure design	98
D.3	Performance Models	99
D.3.1	Calculating infrastructure cost	100
D.3.2	Calculating business loss from a general business impact model . . .	101
D.3.3	Calculating service availability	103
D.3.4	Calculating customer defection probability	105
D.4	Application of the Method and Models	114
D.4.1	Behavior of the loss metric	114
D.4.2	Comparing business-oriented design with ad hoc design	117
D.4.3	The influence of business process importance	120
D.4.4	Provisioning two services	121
D.4.5	The effect of varying load on optimal design	122
D.4.6	Designing infrastructure for load surges	124

D.5	Related Work	127
D.6	Conclusions	131
E	Business-Driven Service Level Agreement Negotiation and Service Provisioning	134
E.1	Introduction	135
E.2	Related Work	136
E.3	Conventional SLA Negotiation Approach	138
E.3.1	IT Infrastructure Basics	138
E.3.2	Defining and evaluating IT metrics	139
E.3.3	Defining and evaluating business metrics	141
E.3.4	Conventional SLA Negotiation Formalization	143
E.4	Business-Driven SLA Negotiation Approach	144
E.4.1	Improving the Service Customer-Client Linkage	145
E.4.2	Business-Driven SLA Negotiation	147
E.5	Evaluating the Approach: an Example	148
E.6	Conclusions	153

Lista de Figuras

2.1	Esboço da Solução Proposta	9
2.2	Principais Entidades do Modelo	12
A.1	Model entities	45
A.2	CBMG for the e-commerce site	51
A.3	E-commerce Business Loss	55
A.4	Sensitivity of total cost plus loss due to load	59
A.5	Sensitivity of cost and loss due to load	61
A.6	Response time for various designs	62
B.1	Load States	67
B.2	Cost and BDIM Dimensions Comparison	72
B.3	Influence of Surge Duration on Optimal Design	73
B.4	Influence of Surge Height on Optimal Design	74
B.5	Comparison of Three Design Alternatives	75
C.1	Entities and their relationships	81
C.2	Effect of Load on Loss	88
C.3	Sensitivity of Loss due to Redundancy	88
D.1	Model Layers	95
D.2	Cost and Availability Model Entities	100
D.3	Business Impact Model – Crossing the IT-Business Gap	102
D.4	CBMG for the e-commerce site	107
D.5	Customer behavior model	108
D.6	Flow of a request between tiers and cache model	109

D.7	Sensitivity of Loss due to Redundancy	115
D.8	Effect of Load on Loss	117
D.9	Influence of BP importance on optimal design	119
D.10	Influence of load on optimal design	120
D.11	Variation of Optimal Design due to Different BPs Importance	121
D.12	Sensitivity of Total Cost Plus Loss due to Load	122
D.13	Sensitivity of cost and loss due to load	123
D.14	Response time for various designs	124
D.15	Cost Dimension Comparison	125
D.16	Cost and BIM Dimensions Comparison	126
D.17	Influence of Surge Duration on Optimal Design	127
D.18	Influence of Surge Height on Optimal Design	128
E.1	Conventional SLA Negotiation Approach	139
E.2	Business-Driven SLA Negotiation Approach	145
E.3	Service Provider Cost Comparison	152
E.4	Service Provider Cost and Profit Comparison	152
E.5	Client Loss, Cost and Profit Comparison (Business-driven Approach)	153

Lista de Tabelas

2.1	Definição formal do problema de projetar infra-estruturas de TI.	13
2.2	Definição formal do problema de projetar SLAs.	13
A.1	Notational summary for problem definition	47
A.2	Formal definition of design problem	47
A.3	Notational summary for problem definition	48
A.4	Notational summary for problem definition	50
A.5	Transition probabilities in NRG CBMG	53
A.6	Average number of visits to each state	56
A.7	Parameters for example site	57
A.8	Service demand in milliseconds in all tiers	57
A.9	Comparing infrastructure designs	59
B.1	Cost-Oriented Capacity Planning Problem Formalization	69
B.2	Business-Oriented Capacity Planning Problem Formalization	70
B.3	Example scenario parameters	71
C.1	Parameters for example	86
C.2	Comparing designs	89
D.1	Notational summary for main optimization problem	99
D.2	Notational summary for infrastructure cost	100
D.3	Notational summary for business impact model	102
D.4	Notational summary for availability model	104
D.5	Notational summary for response time analysis	106
D.6	Transition probabilities in NRG CBMG	114

D.7	Average number of visits to each state	115
D.8	Parameters for examples	116
D.9	Service demand in milliseconds in all tiers	116
D.10	Comparing infrastructure designs	119
E.1	Design Problem Associated with Conventional SLA Negotiation	144
E.2	Design Problem Associated with Business-Driven SLA Negotiation	148
E.3	Average Number of visits to each CBMG state during a customer session	149
E.4	Parameters for example scenario	150
E.5	Resource demand in milliseconds	150

Capítulo 1

Introdução

Uma vez que a Tecnologia da Informação (TI) tem fundamental importância no apoio às operações de negócio, as empresas estão cada vez mais dependentes de sua infra-estrutura e seus serviços de TI. Concomitantemente, com o passar do tempo, a TI teve que evoluir de forma a tornar-se capaz de atender às novas regras do mercado. Neste sentido, as metodologias de gerência de TI também tiveram que evoluir de forma a acompanhar a TI. Ao longo desse processo, a gerência de TI passou por diversos níveis de maturidade, iniciando com a gerência de infra-estrutura, *Information Technology Infrastructure Management* [Lew99a] (ITIM) em inglês, que é constituída de vários sub-níveis de escopo: gerência de dispositivos, de redes de computadores, de sistemas, de aplicações e, finalmente, na gerência integrada abrangendo todos estes níveis. Em seguida a gerência de TI evoluiu para a gerência de serviços de TI, *Information Technology Service Management* [BKP02] (ITSM) em inglês, à medida que a TI propriamente dita alcançava o nível de proporcionar vantagem competitiva. Note que, enquanto ITIM objetiva a gerência dos componentes de TI, ITSM busca a gerência dos *serviços* de TI. Neste sentido houve diversas mudanças. Como exemplos, podemos citar a mudança do foco dos componentes de TI para os clientes dos serviços – constituindo sua principal mudança –, a existência de um catálogo descrevendo os serviços de TI oferecidos, a promessa de níveis de qualidade em torno de tais serviços, entre outros. Um dos principais objetivos da gerência de serviços de TI é prover e suportar serviços que atendam às necessidades dos clientes destes. Para tal, ITSM é composto de diversos processos, tais como: gerência de mudanças, gerência de segurança, gerência de níveis de serviço, gerência de incidentes, gerência de configuração, entre outros. Cada um desses processos engloba uma

ou mais atividades do departamento de TI ou do provedor de serviços de TI, quando a TI é terceirizada. Neste momento, é interessante definir bem os termos cliente e usuário acima referidos e que serão amplamente usados ao longo desta dissertação. Cliente é a entidade que recebe, do departamento de TI, ou contrata, perante um provedor, serviços de TI. Já usuário é a pessoa que usará os serviços de TI disponibilizados.

Dado que a infra-estrutura de TI é um componente indispensável no contexto corporativo atual, mantê-la funcionando e atendendo às necessidades do negócio propriamente dito é crucial. Para tal, faz-se necessário entender tais necessidades e relacioná-las com a TI, de forma que o negócio seja suportado por serviços de TI. Tais serviços estão, por sua vez, apoiados em uma infra-estrutura que fornece qualidade apropriada. Contudo, tal tarefa não é trivialmente resolvida. Primeiramente, é bastante comum centrar-se na eficiência da TI propriamente dita [TT05b], o que pode resultar em uma infra-estrutura de TI pouco eficaz em relação ao negócio. Para melhor entendimento desse fato, faz-se necessário conhecer um pouco mais sobre a gerência de serviços de TI, o que é apresentada na subseção seguinte. Em segundo lugar, a própria atividade de projeto de infra-estruturas de TI (*Capacity Planning*, em Inglês) é uma atividade bastante complexa. Projetar uma infra-estrutura de TI para atender a determinados requisitos de performance envolve a configuração de hardware e software em diversas camadas (por exemplo, camada Web, camada de aplicação e camada de dados), definição do número de servidores em modo ativo e em modo *standby* – para prover valores adequados de tempo de resposta e disponibilidade, respectivamente –, *caching*, considerações sobre escala e carga submetida durante períodos normais de demanda, bem como durante períodos com pico de demanda.

1.0.1 Gerência de Serviços de TI

Como relatado previamente, uma das principais contribuições trazidas pela gerência de serviços de TI foi a mudança de foco gerencial dos recursos de TI para o cliente propriamente dito, pois esta não está mais focada apenas nos componentes de TI em si. Todavia, as métricas adotadas continuam sendo técnicas, a exemplo de: disponibilidade, tempo de resposta e vazão. Um problema da adoção de métricas técnicas está relacionado à dificuldade de correlacioná-las com objetivos de negócio, o que é fundamental para que se projete uma infra-estrutura com foco em sua eficácia em relação aos objetivos de negócio; qual a

disponibilidade, ou tempo de resposta, que uma determinada infra-estrutura de TI deve oferecer para um dado serviço? Qual a relação existente entre 99.99% de disponibilidade, ou 1,5 segundos de tempo de resposta, com um determinado objetivo de negócio? Além de não exprimirem claramente os objetivos de negócio, métricas técnicas dificultam a comunicação entre o departamento de TI e o resto da empresa, a qual não está familiarizada com um jargão técnico, mas sim com um jargão de negócios.

A maneira padrão atual de relacionar as necessidades do negócio e da TI é através de contratos conhecidos como Acordos de Nível de Serviço ou *Service Level Agreements* (SLAs) [BTdZ99; LR99; SM00; Lew99a], em inglês. A elaboração de SLAs fornece, entre outras coisas, um consenso sobre os valores de níveis de serviços requeridos pelo cliente e os níveis de serviços oferecidos pelo departamento de TI ou pelo provedor de serviços de TI, quando a TI é terceirizada. Um SLA é aplicável tanto para especificação de serviços intra-empresa quanto para especificação inter-empresas, isto é, envolvendo serviços terceirizados. Tais valores de níveis de serviços, quando expressos em um SLA, tomam a forma de *Service Level Objectives* (SLOs), em inglês, e são limiares aplicados a métricas de desempenho, conhecidas como *Service Level Indicators* (SLIs) em inglês. "Disponibilidade" e "tempo médio de resposta do serviço" são exemplos de SLIs utilizados num SLA; enquanto que, os valores 99.99% de disponibilidade ou 1,5 segundos de tempo de resposta são exemplos de SLOs. Ainda, diversos outros parâmetros podem ser definidos em um SLA, tais como penalidades – a serem pagas pelo provedor do serviço quando este falhar em cumprir o nível de serviço prometido –; recompensas – a serem pagas pelo cliente do serviço quando o provedor do serviço superar as expectativas do nível de serviço acordado –; período de vigência do contrato e período de avaliação dos SLOs. Neste contexto, SLAs são componentes de fundamental importância para a gerência de níveis de serviços, *Service Level Management* (SLM), conforme pode ser constatado nas seguintes metodologias: *IT Infrastructure Library* (ITIL) [Ele03; BKP02], *IT Service Management* (ITSM) da Hewlett Packard (HP) [HP], *Microsoft Operations Framework* (MOF) [Mic] da Microsoft e *Control Objectives for Information and related Technology* (COBIT) [Ins01]. Estas quatro abordagens podem ser vistas como coleções de boas práticas e recomendações que abordam diversos aspectos da gerência de TI, em particular da gerência de serviços. Além da especificação de níveis de serviço, SLM também aborda a negociação, implementação,

monitoração e otimização dos serviços de TI a serem oferecidos [BTdZ99; LR99; SM00; Lew99a]. Neste sentido, SLM é uma parte crucial da gerência de serviços de TI [BKP02]. Com relação à adoção de SLAs, vale a pena salientar que apesar de serem comumente adotados para expressar as necessidades do negócio com relação aos serviços de TI, a escolha de seus parâmetros (técnicos), em especial SLOs, também não é uma atividade trivial. Além disso, tal escolha tem sido feita de maneira *ad-hoc*, sem uma metodologia apropriada, resultando em acordos difíceis de serem cumpridos e que podem não atender às expectativas do negócio, e, por conseqüência gerar possíveis grandes perdas no negócio, como se constata na referência [TT05b]. Esta referência consiste basicamente de um estudo com grandes empresas americanas e européias que demonstrou uma enorme deficiência na escolha dos parâmetros de SLAs. Dessa forma, tal estudo veio a reforçar, baseado em dados de contratos reais, a necessidade de métodos e modelos mais maduros que resultem em SLAs adequadamente negociados.

1.0.2 BDIM: uma extensão à Gerência de Serviços de TI

Recentemente, uma nova área da gerência de TI, chamada de *Business-Driven Impact Management* (BDIM) [Dub02; SMS⁺04; Pat02; Mas02; DCR04] estende a gerência de serviços através da adoção de métricas de negócio em adição às métricas técnicas convencionais. Como apresentado na referência [SMS⁺05], uma métrica de negócio pode ser vista como uma quantificação de um atributo de um processo de negócio (*business process* (BP), em inglês) ou de uma unidade de negócio. Lucro, faturamento, número de funcionários ou de BPs afetados por um determinado evento ocorrendo na infra-estrutura e perda de vazão de BP são exemplos de métricas de negócios. Um BP representa uma seqüência de tarefas que deve ser executada para a realização de algum evento de negócio que resulte em agregação de valor para a empresa ou para o cliente. Por exemplo, um processo de venda – que constitui um dos objetivos do negócio – envolve diversas atividades, geralmente apoiadas por TI ou seus serviços, tais como verificar crédito do comprador, compra de matéria-prima, disparar processo de fabricação e realizar a entrega dos produtos. Nenhuma destas atividades (tarefas) de forma isolada permite que realizemos o processo de venda – precisamos de uma visão "integrada" da TI para tal. Neste sentido, um processo de negócio provavelmente, mas não necessariamente, está apoiado por TI para prover maior eficiência.

Contudo, como será visto a seguir, BDIM vai além da proposta de utilização de uma nova modalidade de métricas para gerir a TI. O principal objetivo de BDIM é melhorar e ajudar o negócio propriamente dito [SMS⁺06]. Tal objetivo é alcançado na medida em que BDIM visa propôr um conjunto de ferramentas, modelos e técnicas que permitem mapear eventos ocorridos na TI com o impacto quantitativo destes nos resultados do negócio. A quantificação de tal impacto pode servir como fonte de informação de forma a propiciar a tomada de decisões que melhorem os resultados do negócio em si e também da TI, esta última como elemento de suporte às operações de negócio – isto é a essência de BDIM. Neste sentido BDIM busca capturar o impacto negativo nos negócios, ou simplesmente perda, causado pelas falhas inerentes à infra-estrutura de TI subjacente, assim como por degradações de desempenho dos serviços de TI. Os Apêndices A, B, C, D e E discutem BDIM mais profundamente. Tais apêndices representam os resultados obtidos com o trabalho desenvolvido ao longo do período do presente mestrado, resultando nesta dissertação. Tais resultados, por sua vez, foram publicados como artigos científicos em conferências internacionais.

Neste momento é interessante entender quão sério é o problema de *não* considerar a perspectiva de negócio no projeto de infra-estruturas de TI ou de SLAs. A seguir, encontra-se ilustrado tal fato com alguns números obtidos dos Apêndices A, B, C, D e E. Com isso, objetiva-se antecipar alguns resultados gerais de forma a motivar o presente trabalho e explicitar a importância do tema em questão. Como mostrado no apêndice C, o custo de se escolher erroneamente uma configuração de infra-estrutura de TI, ou valores inapropriados de SLOs, pode ser bastante alto. Por exemplo, no cenário descrito no referido apêndice, gasta-se desnecessariamente entre cerca de, 5830% (US\$ 1.700.000,00, em valor absoluto) e 105% (US\$30.500,00 em termos absolutos), dependendo da infra-estrutura escolhida. Esse mesmo cenário exemplo mostra que projetar infra-estruturas TI realizando *overdesign* também produz possíveis grandes perdas; aproximadamente 31% (US\$ 9.000 percentualmente) no referido cenário. Também, projetar infra-estruturas capazes de lidar com picos de demanda (vide Apêndice B), sem o devido entendimento da necessidade do negócio, pode ter um custo bastante elevado; em média, o custo total (custo da infra-estrutura de TI mais perda financeira) é 67% maior que a abordagem orientada aos negócios. Tal diferença pode chegar a ser superior a 100% em alguns picos de demanda. Percebe-se, baseado nos exemplos acima, que a perda potencial ao se desconsiderar a perspectiva dos negócios pode ser alta, e,

portanto, nos motiva a elaborar soluções que levem em conta tal perspectiva.

A nova perspectiva aberta por BDIM permite reconsiderar as atividades de projeto de infra-estruturas de TI e de SLAs através da consideração de métricas de negócio. No tocante ao projeto de infra-estruturas de TI, tem-se o seguinte argumento informal: projetar uma infra-estrutura de melhor qualidade, portanto mais complexa, implica em um alto custo associado e em uma pequena possível perda financeira por impacto de falhas de TI ou degradações de performance. Por outro lado, projetar uma infra-estrutura de baixa qualidade resulta em um baixo custo associado e em um provável maior impacto negativo nos negócios, originado a partir de falhas de TI ou degradações de desempenho. Tem-se claramente um *tradeoff*. Achar o ponto ótimo deste *tradeoff* significa achar a infra-estrutura ótima segundo a perspectiva dos negócios; pode-se, por exemplo, obter a infra-estrutura de TI com o menor valor para perda ou para a soma da perda financeira com o custo da infra-estrutura.

1.1 Objetivos

O objetivo geral deste trabalho é apresentar uma investigação formal de técnicas para (i) projetar infra-estruturas de TI e (ii) escolher parâmetros de SLAs segundo uma perspectiva de negócio, de forma a otimizar métricas de negócios. Como exemplo de tais métricas, pode-se citar: custo de provisionamento de infra-estrutura ou perda, oriunda da degradação de performance nos serviços especificados num SLA ou devido às falhas na infra-estrutura de TI subjacente. A supracitada perspectiva de negócio – capturada por meios de uma visão BDIM – permitirá ponderar os custos relativos à obtenção de serviços face às perdas oriundas do impacto negativo das falhas da infra-estrutura nos negócios – *tradeoff* acima apresentado.

Neste trabalho, considera-se apenas a infra-estrutura relacionada ao *server farm*, não sendo considerados elementos como a rede ou os computadores pessoais dos usuários que acessam os serviços suportados pela infra-estrutura de TI – a consideração de tais elementos constitui um possível trabalho futuro. Com relação ao projeto de infra-estruturas de TI, apesar de estarem sendo consideradas infra-estruturas estáticas, também são apresentados estudos sobre o efeito de picos de demanda no projeto da infra-estrutura ótima e uma breve discussão sobre o uso de BDIM com provisionamento dinâmico. No tocante à escolha de parâmetros de SLA, foca-se na escolha de SLOs e aborda-se tanto serviços que envolvem

consumidor e provedor no mesmo domínio administrativo, quanto serviços terceirizados.

De forma mais específica objetiva-se adicionalmente apresentar um modelo analítico usado para representar uma configuração típica da infra-estrutura de TI necessária para suportar determinada qualidade de serviço.

1.2 Organização da Dissertação

Considerando a existência de vários artigos científicos relatando a evolução do trabalho em questão, e que os temas neles discutidos constituem-se alvo de maior comparação com trabalhos da comunidade internacional do que da comunidade nacional, esta dissertação apresenta uma estrutura bastante enxuta em português e faz freqüentes referências aos apêndices que representam as contribuições trazidas à gerência de TI, especificamente às atividades de projeto de infra-estruturas de TI e de Acordos de Níveis de Serviço de TI, e que resultaram nesta dissertação. Vale salientar que tais apêndices estão escritos em inglês, pois representam artigos científicos, que figuram entre os trabalhos pioneiros de BDIM, publicados ou submetidos a algum veículo de circulação internacional.

O restante da dissertação está organizado como descrito a seguir. No Capítulo 2 descrever-se-á, em linhas gerais, a solução proposta pelo presente trabalho aos problemas acima apresentados. Um guia objetivando facilitar a leitura dos apêndices da dissertação será apresentado no Capítulo 3. Tal guia apresentará um resumo e os principais resultados de cada artigo científico. O Capítulo 4 discorrerá sobre aspectos relacionados à validação do trabalho desenvolvido, principalmente no que diz respeito à validade e acurácia dos modelos analíticos propostos. Por último, o Capítulo 5 sumarizará a abordagem proposta, relatará as conclusões obtidas e discutirá idéias de possíveis próximos passos para o trabalho em questão.

Capítulo 2

Projetando Infra-estruturas de TI e Acordos de Níveis de Serviços

Este capítulo descreve rapidamente a solução apresentada por esta dissertação para os problemas relacionados com as atividades de projeto de infra-estruturas de TI e de SLAs que foram apresentados no capítulo anterior. Para detalhes específicos sobre cada solução, vide os Apêndices A, B, C, D e E.

Desconsiderar a perspectiva dos negócios quando projetando infra-estruturas de TI para atender às necessidades do negócio pode gerar grande perda financeira. Primeiramente, é preciso entender tais necessidades e relacioná-las com a TI, para que seja possível projetar infra-estruturas de TI, ou especificar SLAs, que ofereçam uma qualidade de serviço apropriada. Neste sentido, é fundamental a adição da visão de negócio nestas atividades. Tal visão de negócio significa entender qual a relação existente entre a TI e o negócio, ou seja, capturar qual o impacto no negócio caso determinado evento na infra-estrutura de TI ocorra.

Este trabalho está baseado no argumento informal de que projetar uma infra-estrutura de melhor qualidade, portanto mais complexa, implica em um maior custo associado e em uma possivelmente menor perda financeira por impacto de falhas de TI ou degradações de performance. Por outro lado, projetar uma infra-estrutura de menor qualidade resulta em um menor custo associado e em um possível maior impacto negativo nos negócios, originado a partir de falhas de TI ou degradações de desempenho. Observa-se claramente um *tradeoff*. A abordagem proposta por este trabalho, e adotada nos Apêndices A, B, C, D e E, consiste em achar o ponto ótimo desse *tradeoff*, através de uma otimização. Este ponto ótimo repre-

senta a infra-estrutura ótima de TI segundo a perspectiva de negócio, e, portanto de acordo com os objetivos gerais de negócio. Note que a abordagem aqui apresentada não captura os requisitos específicos dos objetivos de negócio, mas apenas requisitos gerais no sentido de minimizar o impacto financeiro nos negócios. Após obter a solução ótima, caso se deseje especificar um SLA, pode-se calcular os valores dos SLOs a partir da infra-estrutura obtida. Neste sentido, esta abordagem para especificação de SLAs é substancialmente diferente das abordagens atuais, onde tais valores são escolhidos sem um perfeito entendimento das necessidades do negócio. Além disso, nas abordagens atuais os SLOs são escolhidos *a priori*, isto é, primeiro se define os valores dos SLOs e depois projeta-se uma infra-estrutura de TI que os suporte. Tem-se o processo inverso na abordagem proposta: primeiro se define uma infra-estrutura de TI adequada às necessidades do negócio para então calcular os SLOs resultantes. O que doravante chamar-se-á de uma escolha *a posteriori* dos SLOs. A Figura 2.1 ilustra a abordagem proposta.

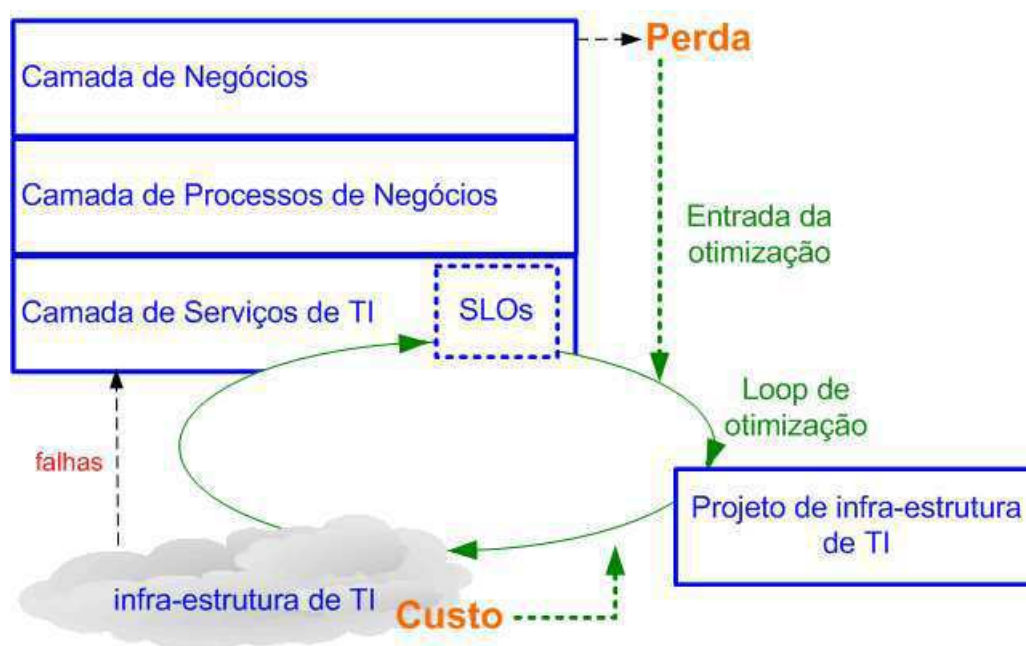


Figura 2.1: Esboço da Solução Proposta

Para capturar o impacto negativo nos negócios oriundo de falhas associadas à infra-estrutura de TI, assim como de degradações de desempenho dos serviços de TI, é preciso criar uma relação de causa-efeito entre a TI e o negócio propriamente dito. No contexto BDIM, tal relação causa-efeito é obtida através de um modelo, chamado de *modelo de im-*

pacto. Logo, a tarefa desempenhada por tal modelo não é trivial, sendo equivalente a relacionar métricas técnicas em métricas de negócio de forma a obter uma perspectiva de negócio. Neste sentido, com o objetivo de estabelecer a referida relação causa-efeito, adota-se um modelo em três camadas (vide Figura 2.1), composto das seguintes camadas: de TI, de processos de negócios e de negócios. Neste modelo a camada de processos de negócio (BPs) tem um papel fundamental no cruzamento de métricas entre a camada de TI e a camada de negócios. A camada de BPs é responsável pelo mapeamento entre métricas técnicas – tais como disponibilidade e tempo médio de resposta – em métricas de BP, tal como vazão de transações de negócio. Estas métricas técnicas são obtidas a partir da camada de TI. Na seqüência, a camada de negócios, através de um submodelo chamado de *modelo de receita*, relaciona métricas de BP a métricas de negócio, tal como faturamento perdido.

O modelo de impacto proposto e utilizado neste trabalho considera que a perda financeira, $L(\Delta T)$, durante o intervalo de tempo ΔT , tem duas fontes: indisponibilidade dos serviços de TI e desistência de compra por parte dos consumidores finais, isto é, aqueles que acessam o BP suportado pelos serviços de TI, a fim de realizarem suas compras (comércio eletrônico ou *e-commerce*, em inglês). No modelo utilizado aqui, um consumidor final desiste de comprar sempre que experimentar um tempo de resposta maior que um determinado limiar, T^{DEF} ; o valor de 8 segundos é comumente referido na literatura como um valor apropriado para T^{DEF} [MAD04]. Portanto, para capturar o impacto negativo, é preciso calcular a disponibilidade dos serviços de TI, o que é feito utilizando teoria da confiabilidade convencional [Tri82], e obter a probabilidade, $B(T^{DEF})$, de que o tempo de resposta seja maior que o dado limiar. O cálculo destes dois valores acima mencionados permite a derivação do valor da perda financeira, $L(\Delta T)$, durante o intervalo de tempo ΔT . Note que, além de obter $L(\Delta T)$, é preciso também obter $C(\Delta T)$, o custo da infra-estrutura de TI durante o período ΔT , para que a abordagem proposta, minimizar o valor da soma da perda financeira com o custo da infra-estrutura, seja efetivamente aplicada. Observe que, o modelo específico de impacto adotado neste trabalho depende essencialmente de BPs para estabelecer a relação de causa-efeito entre a TI e o negócio propriamente dito. Este fato implica que tal modelo em particular é aplicável apenas em um contexto no qual há uma forte, ou total, dependência na TI. Porém, este fato não constitui uma restrição do presente trabalho, pois a abordagem proposta é perfeitamente geral, de forma a permitir a utilização de outros mod-

elos de impacto, aplicáveis em cenários diferentes do considerado. Como exemplo típico de cenário que atende claramente ao requisito de forte dependência na TI pode-se citar o comércio eletrônico (*e-commerce*). Tal cenário será usado na exposição das idéias trazidas pela abordagem proposta pela presente dissertação.

A seguir é apresentado o modelo formal básico de infra-estrutura que permite a obtenção das duas métricas de interesse no processo de otimização: custo da infra-estrutura de TI e impacto negativo nos negócios (perda). Para derivar o impacto negativo é necessário obter a disponibilidade dos serviços de TI e a desistência de compra por parte dos consumidores finais. Comentários gerais sobre a obtenção de todas essas métricas são feitos a partir do próximo parágrafo. Vale salientar que diversas extensões foram acrescentadas a esse modelo básico. Tais extensões, trazidas nos artigos científicos constantes nos apêndices A, B, C, D e E, serão explicitamente apresentadas no próximo capítulo desta dissertação. No capítulo seguinte apresentar-se-á de maneira formal e detalhada cada um destes artigos científicos.

2.1 Abstração da Infra-estrutura de TI

Conforme mostrado na Figura 2.2, considera-se que cada BP está apoiado em apenas um serviço de TI (*IT service*). Ainda, considera-se o caso de apenas um serviço de TI. Apesar da generalização destas duas simplificações ser uma tarefa relativamente fácil, complicaria de forma desnecessária a apresentação do modelo. Um dado serviço S baseia-se em um conjunto RC de classes de recursos de TI (*IT resource classes*); tome um conjunto de servidores de banco de dados ou de servidores de aplicação como exemplos. Uma determinada classe de recursos, RC_j , é composta de um *cluster* de n_j recursos de TI (*IT resources*) idênticos. Como exemplos de recursos de TI, pode-se citar um servidor de banco de dados ou um servidor *web*. Do total de servidores de RC_j , m_j recursos de TI estão configurados em modo ativo, ou de balanceamento de carga, para lidar com a carga recebida oferecendo tempo de resposta aceitável, e $n_j - m_j$ recursos de TI estão configurados em modo *standby* (*fail-over*), oferecendo uma melhor disponibilidade ao serviço S .

Além disso, um dado recurso de TI R_j , pertencente à classe de recursos RC_j , é formado de um conjunto $P_j = \{P_{j,1}, \dots, P_{j,k}, \dots\}$ de componentes de TI (*IT components*) – tome hardware do servidor e sistema operacional como exemplos de componentes de TI que po-

dem fazer parte de R_j . Caso um ou mais componente de TI falhem, o respectivo recurso de TI também falha. O formalismo matemático apresentado corresponde ao modelo formal básico para representar uma infra-estrutura de TI. Apesar de haver diversas extensões a este modelo de infra-estrutura de TI, estes não são apresentadas aqui (vide apêndices A, B, C, D e E). O objetivo é fornecer apenas o necessário para um entendimento um pouco mais formal da abordagem proposta.

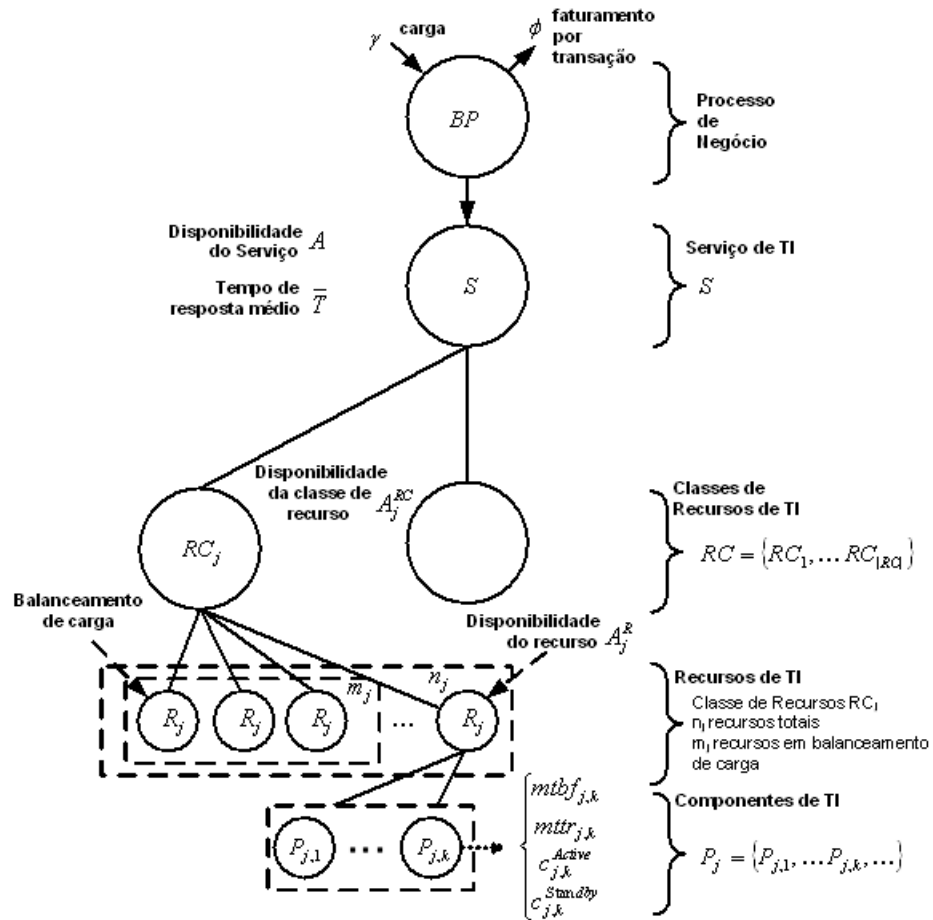


Figura 2.2: Principais Entidades do Modelo

2.2 Sobre os problemas de projeto de infra-estruturas e SLAs

Como já foi visto, projetar infra-estruturas de TI não é uma atividade trivial. O problema de projetar infra-estruturas de TI pode ser definido como mostrado na Tabela 2.1.

Tabela 2.1: Definição formal do problema de projetar infra-estruturas de TI.

Achar:	O número total de recursos, n_j , e o número de recursos em balanceamento de carga, m_j , para cada classe de recursos RC_j .
Minimizando:	$C(\Delta T) + L(\Delta(T)$, o impacto financeiro nos negócios durante o período ΔT .
Sujeito a:	$n_j \geq m_j$ e $m_j \geq 1$.

Considerando que na presente abordagem primeiro se define uma infra-estrutura de TI adequada às necessidades do negócio para então calcular os SLOs resultantes, pode-se especificar o problema de projetar SLAs conforme mostrado na Tabela 2.2.

Tabela 2.2: Definição formal do problema de projetar SLAs.

Achar:	Os parâmetros do SLA em especificação: A^{MIN} , \bar{T} e B^{MAX} .
Minimizando:	$C(\Delta T) + L(\Delta(T)$, o impacto financeiro nos negócios durante o período ΔT .
Variando:	O número total de recursos, n_j , e o número de recursos em balanceamento de carga, m_j , para cada classe de recursos RC_j .
Sujeito a:	$n_j \geq m_j$ e $m_j \geq 1$.
Onde:	A^{MIN} é o valor mínimo aceitável para a disponibilidade do serviço S ; \bar{T} é o valor máximo aceitável para o tempo médio de resposta do serviço S ; B^{MAX} é o valor máximo aceitável para a probabilidade de desistência.

2.3 Sobre o modelo de custo da infra-estrutura de TI

O custo da infra-estrutura, $C(\Delta T)$, durante o período ΔT , é simplesmente a soma dos custos de todos os recursos de TI. Uma vez que os recursos de TI podem estar configurados em modo ativo (balanceamento de carga) ou em modo *standby* (*fail-over*), eles devem ter custos diferentes quando configurados em modos diferentes [JST03]. Recursos configurados no modo ativo têm maiores custos devido a gastos com energia elétrica, licenças de *software*, etc. Considerando que recursos de TI são compostos de componentes de TI, é possível

expressar suas taxas de custo em termos das taxas de custo dos componentes de TI que o compõem; $c_{j,k}^{active}$ e $c_{j,k}^{standby}$ representam as taxas de custo do componente $P_{j,k}$ quando configurado nos modos ativo e em espera respectivamente. Tais valores de custo são expressos em taxa de custo e representam custo por unidade de tempo, o que possibilita o cálculo do custo da infra-estrutura de TI para um determinado período. Para obter detalhes de como calcular $C(\Delta T)$, o custo da infra-estrutura consulte os Apêndices A, B, C, D e E.

2.4 Sobre o modelo particular de impacto negativo

Como definido anteriormente, para que seja possível aplicar o método proposto, além do valor de o custo da infra-estrutura de TI é preciso também estimar o valor do impacto negativo nos negócios, $L(\Delta T)$, durante o período ΔT . Dado que γ transações por segundo chegam ao BP BP gerador de faturamento e que está suportado por S , caso a infra-estrutura de TI subjacente fosse perfeita, todas as transações de entrada seriam transformadas em faturamento. Um BP é dito gerador de faturamento caso produza renda a cada transação finalizada com sucesso. O faturamento médio por transação do BP é ϕ . No entanto, as imperfeições da TI – leia-se indisponibilidade e desistência dos consumidores finais – reduzem a vazão de BP para X transações por segundo existindo, portanto uma perda na vazão de BP de ΔX transações por segundo. A vazão de um BP representa a sua taxa de transações finalizadas com sucesso. Logo, o impacto negativo nos negócios é $L(\Delta T) = \Delta T \times \Delta X \times \phi$, mas $\Delta X = \Delta X^T + \Delta X^A$, onde $\Delta X^T = \gamma \times B(T^{DEF}) \times A$ é o impacto negativo atribuído às desistências por parte dos consumidores finais e $\Delta X^A = \gamma \times (1 - A)$ é o impacto negativo atribuído à indisponibilidade do serviço S . Dessa forma, $L(\Delta T) = \Delta T \times (\Delta X^T + \Delta X^A) \times \phi = \Delta T \times \gamma \times \phi \times (B(T^{DEF}) \times A + 1 - A)$. A seguir comentários sobre os modelos de disponibilidade e de performance que propiciam, respectivamente, o cálculo da disponibilidade do serviço de TI e da probabilidade de desistência de compra por parte dos consumidores finais.

2.5 Sobre o modelo de disponibilidade do serviço de TI

Para calcular a disponibilidade, A , do serviço S é preciso ter em mente que, devido à imperfeição inerente da infra-estrutura subjacente, cujos componentes falham, o serviço S torna-se indisponível. A notação A é uma referência à palavra *availability* que significa disponibilidade em inglês. Lançando-se mão da teoria da confiabilidade, a disponibilidade individual dos componentes é achada através de seus valores de Tempo Médio entre Falhas (*Mean Time Between Failures* – MTBF, em inglês) e de Tempo Médio para Reparo (*Mean Time to Repair* – MTTR, em inglês). Os valores de MTBF podem ser obtidos a partir de especificações de componentes ou por meio de registros (*logs*). Por outro lado, Os valores de MTTR dependem do tipo de contrato de serviço de manutenção desses componentes. Na seqüência, partindo-se da disponibilidade individual dos componentes de TI e adotando uma abordagem *bottom-up* obtém-se a disponibilidade A do serviço de TI. Para tal, sempre se valendo da teoria da confiabilidade e do modelo de infra-estrutura adotado, calcula-se a disponibilidade de um recurso de TI, digamos A_j^R . A partir dos valores de disponibilidade dos recursos de TI deriva-se a disponibilidade das classes de recursos. Por fim, tendo as disponibilidades das classes de recursos, determina-se A . Os detalhes específicos de como obter cada uma dessas disponibilidades são apresentados nos apêndices A, B, C, D e E.

2.6 Sobre a probabilidade de desistência

Por fim, resta mostrar como derivar $B(T^{DEF})$, a probabilidade de que o tempo de resposta seja maior que o dado limiar. Para calcular $B(T^{DEF})$ é preciso achar *ResponseTimeDistribution*(x), a distribuição cumulativa de probabilidade do tempo de resposta. Para tal fim, é preciso modelar a carga aplicada aos recursos de TI, os processos de chegada e de atendimento de transações, etc. Tal modelo é baseado em um modelo aberto de filas [Kle76a]. Modelos abertos de filas são adequados quando há um número grande de potenciais clientes, situação comum no comércio eletrônico. No trabalho aqui desenvolvido, assume-se que o processo de chegada é caracterizado por uma distribuição de Poisson, o que é de fato uma boa aproximação para curtos espaços de tempo [MAR⁺03c] – e que o tempo de serviço é caracterizado por uma distribuição exponencial. Uma vez que

uma dada transação vai usar recursos de possivelmente todas as classes de recursos, o seu tempo de resposta é a soma de $|RC|$ variáveis aleatórias – cada classe de recursos acessada tem sua própria distribuição –, o que pode ser resolvido através do produto de suas transformadas de Laplace. Vale a pena salientar que, caso os efeitos de *cache* estejam sendo considerados, o número de classes de recursos acessadas é uma variável aleatória. Portanto, o tempo de resposta de uma transação não é obtido através da simples soma $|RC|$ variáveis aleatórias. Este caso é um pouco mais delicado e é tratado no apêndice D. Obtida a distribuição $ResponseTimeDistribution(x)$, a probabilidade de desistência do consumidor final do serviço é $B(T^{DEF}) = 1 - ResponseTimeDistribution(T^{DEF})$. O cálculo de $B(T^{DEF})$ é apresentado detalhadamente nos apêndices A, B, C, D e E.

Capítulo 3

Resultados da Pesquisa

Este capítulo apresenta um guia para facilitar a leitura dos apêndices da dissertação. Cada apêndice constitui um artigo científico publicado ou submetido a algum veículo de publicação internacional. Sendo assim, estes artigos estão escritos em Inglês e destacam-se entre os artigos pioneiros de BDIM. Em alguns destes artigos o autor desta dissertação atuou como primeiro autor, enquanto que, nos demais artigos, apesar de não ser o primeiro autor, teve significativa participação na elaboração destes. A seguir apresenta-se um resumo, destacando as contribuições de cada um dos estudos realizados nos artigos científicos. Além disso, a participação do autor desta dissertação na elaboração de cada um dos artigos é explicitada. Tais artigos representam as contribuições trazidas à gerência de TI, especificamente às atividades de projeto de infra-estruturas de TI e de Acordos de Níveis de Serviço, que resultaram nesta dissertação. Note que, com o intuito de facilitar o entendimento do trabalho, os apêndices não são apresentados na mesma ordem com que foram publicados ou submetidos.

3.1 Projeto Ótimo de Infra-Estruturas de TI pela Perspectiva do Negócio

Dois artigos abordam o projeto ótimo de infra-estruturas de TI pela perspectiva do negócio. O primeiro deles (apêndice A) corresponde ao artigo científico publicado no veículo internacional *39^o Hawaii International Conference on System Science* (HICSS 2006). O segundo (apêndice D) diz respeito ao artigo científico submetido ao periódico *Performance Evalua-*

tion. A seguir discorre-se sobre cada um deles. O primeiro artigo, intitulado de "*Optimal Design of e-Commerce Site Infrastructure from a Business Perspective*", discute um método formal para projetar infra-estruturas de TI segundo a perspectiva do negócio. Enquanto a maioria dos estudos atuais trata de metodologias que se restringem a três métricas – tempo de resposta, disponibilidade e custo –, o modelo formal proposto repensa tal atividade através da adoção de uma quarta métrica, o impacto negativo nos negócios. O modelo apresentado neste artigo traz um diferencial em relação ao modelo básico descrito no capítulo anterior: a carga aplicada aos recursos de TI é modelada através de um modelo conhecido por *Customer Behavior Model Graph* (CBMG) [MAD04]. CBMG permite representar a carga que uma dada sessão iniciada por um consumidor final dos serviços impõe à infra-estrutura de TI.

Um CBMG compreende um conjunto de estados com probabilidades de transição associadas (grafo direcionado e ponderado). Cada estado do CBMG pode ser visto como uma página do *site* de comércio eletrônico sendo acessada. Há basicamente dois tipos de sessões: as sessões geradoras de faturamento – *revenue generating* (RG) *sessions* – e as sessões não-geradoras de faturamento — *non-revenue generating* (NRG) *sessions*. Sessões geradoras de faturamento são aquelas que visitam algum estado gerador de faturamento do CBMG. Como exemplo de estado gerador de faturamento tome a página do *site* de comércio eletrônico onde o pagamento dos itens previamente adicionados ao carrinho de compras é realizado. Sessões não-geradoras de faturamento não visitam nenhum estado gerador de faturamento do CBMG, no entanto impõem carga aos recursos de TI. Consumidores finais que só realizam consultas de preços sem efetivamente concretizar a compra, por exemplo, geram sessões não-geradoras de faturamento. Note que, a consideração de sessões não-geradoras de faturamento acrescenta mais realismo ao modelo básico apresentado no capítulo anterior. Por fim, considerando que a modelagem da carga imposta é feita por meio de CBMG, os serviços de TI são agora modelados por meio de um modelo multi-classe aberto de teoria das filas [Kle76a]. Neste modelo, cada classe CBMG representa uma classe no modelo de teoria das filas.

Neste trabalho, além do projeto ótimo de infra-estruturas de TI pela perspectiva do negócio, o uso da metodologia proposta em infra-estruturas adaptativas [KC03] também foi discutido.

Já o segundo artigo, cujo título é "*Business-Driven Design of Infrastructures for IT Services*" tem uma abordagem semelhante àquela do primeiro artigo discutido acima. A diferença básica diz respeito ao modelo utilizado. Neste último artigo diversas extensões foram feitas ao modelo utilizado no primeiro artigo descrito acima. Enumera-se a consideração dos efeitos de *cache* entre as classes de recursos e a possibilidade de diversas visitas a um dado estado do CBMG. Diante de tais extensões este modelo é significativamente mais completo. Neste artigo, todos os estudos dos outros artigos (inclusive os que serão mostrados nas próximas subseções) são refeitos baseados neste modelo. Portanto, tal artigo pode ser visto como uma versão final do trabalho desenvolvido durante o mestrado, condensando os resultados de todos os demais trabalhos.

Apesar de também ter sido escrito por outras pessoas, o autor desta dissertação participou ativamente do desenvolvimento matemático das extensões ao modelo analítico e da obtenção dos resultados, sendo dessa forma o segundo autor dos artigos em questão.

3.2 Estudo do Efeito de Picos de Demanda no Projeto Ótimo de Infra-Estruturas de TI pela Perspectiva do Negócio

Um outro estudo realizado nesta dissertação corresponde a uma generalização do modelo básico apresentado no capítulo anterior, cujo intuito é o de permitir que um dado serviço de TI, S , seja submetido a diferentes estados de demanda. Por exemplo, serviço S pode ser submetido a uma demanda de 10 transações por segundo durante um determinado período de tempo e, em seguida, ser submetido a uma nova demanda de 20 transações por segundo por um outro período de tempo. Tal generalização permitiu formalizar e apresentar uma nova abordagem para projeto de infra-estruturas estáticas de TI que considera picos de demanda, a qual foi apresentada no artigo aceito no veículo internacional *10^o IEEE/IFIP Network Operations and Management Symposium (NOMS 2006)*, constante no apêndice B. Picos de demanda são bastante comuns no contexto de comércio eletrônico, especialmente nos períodos que precedem datas comemorativas tradicionais como Natal ou dia das mães. Não considerar *a priori* a existência de tais variações de demanda pode resultar no não atendi-

mento, durante os tais picos de demanda, dos requisitos de desempenho (leia-se tempo de resposta). Como implicação direta deste último fato tem-se uma possível grande perda financeira devido à desistência dos consumidores finais, o que é justificada pelos altos tempos de resposta. Alternativamente, poder-se-ia superdimensionar a infra-estrutura focando o projeto da infra-estrutura no pico de demanda. No entanto, esta alternativa tende a resultar em uma infra-estrutura de alto custo e que provavelmente será subutilizada durante os períodos normais de demanda.

A idéia básica deste artigo científico é estudar o efeito que picos esperados de demanda têm no projeto ótimo de infra-estruturas estáticas de TI pela perspectiva do negócio. Picos inesperados de demanda estão fora do escopo deste artigo e, portanto, não foram investigados. Picos esperados de demanda são aqueles que ocorrem durante os períodos que precedem datas comemorativas tradicionais ou promoções planejadas de venda, cujo acontecimento é de conhecimento prévio. Tal trabalho provê respostas às seguintes perguntas: quão diferente é a infra-estrutura de TI resultante quando picos de demanda são considerados durante o processo de projeto de infra-estrutura? Quão caro é não considerar picos de demanda? Como a infra-estrutura ótima é influenciada por variações nas características de um pico de demanda?

Neste artigo, o autor da presente dissertação atua como primeiro autor. Sua participação se deu na extensão ao modelo analítico apresentado, geração dos resultados e escrita do mesmo.

3.3 Negociação Ótima de SLAs Internos pela Perspectiva do Negócio

O apêndice C contém o artigo científico publicado no veículo internacional intitulado *16^o IFIP/IEEE Distributed Systems: Operations and Management (DSOM 2005)*. A essência por trás deste trabalho está relacionada à especificação de valores ótimos de SLOs segundo a visão dos negócios. O referido estudo está restrito ao caso onde ambos, provedor e consumidor do serviço, estão no mesmo domínio administrativo. Nas abordagens tradicionais de projeto para projeto de SLAs, os valores dos SLOs são escolhidos *a priori* para depois projetar uma infra-estrutura de TI que os suporte. A abordagem proposta neste artigo se

contrapõe às abordagens tradicionais na medida que os valores dos SLOs são calculados *a posteriori*, ou seja, a partir da infra-estrutura ótima obtida com a otimização do impacto financeiro (soma do custo da infra-estrutura de TI com a perda financeira), como relatado na abordagem descrita no capítulo anterior. Desta maneira, evita-se que os SLOs sejam escolhidos de forma *ad hoc* e, portanto, sem nenhuma correlação com as necessidades dos negócios.

O autor desta dissertação teve participação na elaboração do modelo analítico e participou ativamente da seção de resultados constantes neste artigo. Desta forma, o mesmo é o segundo autor do artigo em questão.

3.4 Negociação Ótima de SLAs Envolvendo Serviços Terceirizados pela Perspectiva do Negócio

No estudo apresentado na subseção anterior, os valores ótimos de SLOs são obtidos sob o ponto de vista do cliente do serviço. Tal abordagem é perfeitamente aceitável quando tanto o provedor quanto o cliente do serviço confundem-se na mesma empresa. Neste caso, as métricas custo da infra-estrutura e perda financeira incorrem sobre a mesma empresa. Além disso, o conceito de penalidade ou recompensa geralmente não se aplica neste tipo de SLA.

Todavia, é importante lembrar que a terceirização dos serviços de TI para um provedor externo de serviços é um cenário bastante comum atualmente. Logo, se faz necessário considerar também o ponto de vista do provedor e a existência de penalidades e recompensas e, portanto a abordagem descrita no estudo apresentado na subseção anterior não é adequada. Há uma separação lógica das duas métricas utilizadas anteriormente: o custo da infra-estrutura está no lado provedor do serviço, enquanto a perda financeira está no lado no cliente. Convencionalmente, o provedor do serviço irá projetar uma infra-estrutura e considerar as penalidades sob o seu ponto de vista, isto é, de forma a maximizar o seu lucro. Desta forma, não necessariamente há uma expressão clara da correlação existente com os objetivos do negócio do cliente do serviço. A idéia básica deste estudo é apresentar uma abordagem orientada aos negócios, capaz de projetar SLAs que envolvam serviços terceirizados. Neste sentido, propõe-se que a perda financeira do cliente do serviço também seja considerada durante o projeto do SLA, melhorando assim a relação cliente-provedor e resultando em

melhores valores de lucro para ambos. Tal consideração se dá através da *maximização* da margem de lucro do provedor e da margem de lucro do consumidor. Logo se tem uma otimização multi-objetivo, o que difere da abordagem convencional que otimiza apenas o lucro do provedor. Para que essa finalidade seja alcançada, o referido trabalho define uma série de métricas de negócio que permitem obter as margens de lucro do cliente e do provedor do serviço de TI. Faturamento do cliente e do provedor do serviço TI, assim com o preço cobrado pelo provedor por este serviço são exemplos de métricas que foram definidas neste trabalho.

Este artigo consiste do apêndice E. Nele, o autor da presente dissertação atua como primeiro autor onde sua participação se deu na extensão ao modelo analítico apresentado, geração dos resultados e escrita do mesmo.

Capítulo 4

Discussão sobre a Validação dos Modelos

Este capítulo discorrerá acerca dos aspectos relacionados à validação do trabalho desenvolvido, principalmente no que diz respeito à validade e acurácia dos modelos analíticos. Note que apesar de que nem todos os passos aqui descritos terem sido realizados, o principal objetivo aqui é mostrar possíveis caminhos de validação dos modelos analíticos desenvolvidos, caracterizando assim um guia para validação. O motivo pelo qual não se realizou um exercício completo de validação dos modelos apresentados neste trabalho está diretamente relacionado à quantidade de artigos publicadas ou submetidas a conferências periódicos internacionais.

A abordagem apresentada neste trabalho está fortemente baseada em um modelo de infraestrutura de TI capaz de suportar determinados serviços de TI. Tal modelo é composto de diversos sub-modelos analíticos, apresentados brevemente no capítulo 2 e detalhados nos apêndices A, B, C, D e E. Os referidos sub-modelos permitem a obtenção de métricas tais como o custo da infra-estrutura, a disponibilidade do serviço de TI suportado e a probabilidade de desistência dos consumidores finais. Os valores das referidas métricas são usados como entrada no processo de otimização da abordagem proposta, de forma a se obter a infra-estrutura de TI ótima segundo a perspectiva dos negócios. Por modelo analítico entende-se uma representação simplificada da realidade, a qual está baseada em um formalismo matemático implicando, portanto, em um entendimento não ambíguo do mesmo. O objetivo de se modelar apenas parte da realidade é o de permitir estudar isoladamente um ou mais comportamentos do sistema real.

Uma das grandes dificuldades associadas à modelagem de um sistema real diz respeito

à sua complexidade. O comportamento estudado é, em geral, influenciado por um grande número de fatores. Estes fatores, por sua vez, podem dizer respeito tanto ao próprio sistema quanto a questões sociais, culturais, políticas, econômicas, etc. Logo, saber isolar os fatores que realmente influenciam o problema estudado evita que seja preciso criar um modelo cujo nível de detalhes seja desnecessariamente grande, concentrando esforços no que é realmente importante. A identificação destes fatores implica, acima de tudo, na compreensão da natureza do problema estudado. Isto permite a criação de um modelo que corresponda a uma abstração válida do sistema real, isto é, capaz de reproduzir os tais fenômenos de interesse. Por exemplo, ao se modelar o componente de TI *hardware*, a sua cor não desempenha influência nas métricas almejadas na abordagem em questão, e, portanto não precisa ser incluído no modelo de infra-estrutura de TI. Uma outra dificuldade relacionada à arte de modelagem é a obtenção de valores válidos e representativos para os parâmetros de entrada do modelo. Diante das considerações acima apresentadas sobre a concepção de um modelo, fica clara a necessidade de uma atividade de validação de forma a garantir que o modelo é uma abstração representativa do sistema real, e, portanto, seus resultados são críveis. Por fim, objetivando facilitar o processo de análise e compreensão dos fenômenos de interesse é interessante que se implemente o modelo.

A implementação do modelo foi realizada no *MATLAB* [Wor]. Além de ser uma linguagem de programação, *MATLAB* também é um ambiente para desenvolvimento de algoritmos que apresenta diversas facilidades, tais como: facilidade de geração e manipulação de gráficos e disponibilidade de bibliotecas de funções estatísticas e matemáticas, o que justificaram a sua adoção. Após a conclusão da implementação do modelo, ele teoricamente estaria pronto para permitir análise e compreensão dos fenômenos alvo de estudo do sistema real. Porém há de se considerar também a possibilidade de erros introduzidos durante a sua implementação. Portanto, é preciso realizar também uma *verificação* da implementação. A verificação tem por objetivo mostrar que a implementação do modelo no *MATLAB* funciona como esperado e planejado, sendo, portanto uma correta implementação lógica do modelo. A verificação da implementação do modelo foi feita através de ferramenta *test_tools* [Pie]. *Test_tools* facilita a criação de testes de unidade para funções escritas em *MATLAB*, onde os resultados esperados para diversos cenários foram calculados caso a caso à mão e confrontados com os resultados obtidos a partir da execução software. Tal procedimento forneceu uma

maior confiabilidade aos resultados oriundos da implementação lógica do modelo.

Considerando que a abordagem proposta é composta de diversos sub-modelos – sub-modelo de tempo de resposta, sub-modelo de custo, sub-modelo de disponibilidade e sub-modelo de impacto – deve-se também discutir sobre a validade destes para que seja possível obter alguma conclusão sobre a abordagem propriamente dita. A seguir é apresentado um resumo de como a validação poderia ser realmente realizada.

A validação do sub-modelo de tempo de resposta pode ser realizada através da sua comparação com aplicações multi-camadas reais de comércio eletrônico rodando em uma determinada configuração em *cluster* de servidores. Tais aplicações teriam que ser instrumentadas para possibilitar a obtenção, a partir de seus *logs*, das seguintes métricas: tempo de resposta médio, distribuição do tempo de resposta, probabilidade de desistência. Uma vez obtidas tais métricas, seus valores seriam confrontados com os valores oriundos do sub-modelo utilizando-se uma configuração de intra-estrutura semelhante. Uma outra alternativa interessante é a comparação com ferramentas de modelagem e avaliação de performance tais como *SPECweb2005* [SPEb], *TPC* [TPCb], *WebStone* [Web] e *jAppServer2004* [SPEa].

A validação do sub-modelo de disponibilidade pode ser realizada através da comparação dos valores de disponibilidade obtidos com o sub-modelo proposto com valores oriundos de três fontes: *logs* históricos de aplicações reais de comércio eletrônico, ferramentas de modelagem e avaliação de disponibilidade como *Möbius* [CDD⁺04] e *SHARPE* [ST87] e, por fim, outros modelos de disponibilidade de infra-estrutura de TI [JST03; UPS⁺05]. Vale salientar que os valores de MTTR e MTBF utilizados como parâmetros no referido sub-modelo são típicos da tecnologia atual [MAD04; JST03; MV00].

Para validar o sub-modelo de custo, uma alternativa seria a comparação do seu valor resultante de custo de uma dada infra-estrutura de TI com o valor obtido a partir da ferramenta *ISIDE* (*Infrastructure Systems Integrated Design Environment*). O *ISIDE* é uma implementação da metodologia proposta na referência [AFT04] para o projeto de infra-estruturas ótimas em custo, que possui um banco de dados de componentes de infra-estrutura de TI e de seus custos. Uma segunda alternativa seria a comparação com valores de infra-estruturas de TI semelhantes obtidas junto a fornecedores de equipamentos corporativos. Como terceira alternativa ter-se-ia a comparação com os valores obtidos utilizando-se a ferramenta *TPC Pricing Spec* [TPCa].

No tocante à validação do sub-modelo de impacto, poder-se-ia também confrontar os valores obtidos com valores constantes nos *logs* históricos de uma aplicação de comércio eletrônico. É bastante provável que não se obtenha facilmente tal informação, porém, muito provavelmente, uma empresa que implemente conceitos de *Business Intelligence* (BI) deve ter informações suficientes para permitir tal comparação.

Uma vez validado cada um dos sub-modelos, é possível se trabalhar em torno da validação da metodologia proposta. Sua validação passa basicamente pela resposta a duas perguntas: A atividade de projeto de infra-estruturas de TI segundo a perspectiva BDIM é realmente melhor do que as outras abordagens existentes? A atividade de projeto de SLAs segundo a perspectiva BDIM é realmente melhor do que as outras abordagens existentes? O critério de comparação adotado para definir se uma abordagem é melhor do que outra, logicamente, se refere ao atendimento dos objetivos gerais do negócio. Logo, uma infraestrutura é melhor que outra caso ofereça um maior suporte às operações de negócio. Um critério análogo é adotado para o caso de SLAs. Para quantificar tal comparação, sugere-se a adoção das métricas perda financeira e custo da infra-estrutura de TI. As referências constantes nos apêndices B e E caminharam neste sentido, quando realizaram comparações da abordagem proposta com a abordagem convencional, a qual fora previamente formalizada nos referidos trabalhos. Nestes estudos, ficou claro que o resultado final da metodologia proposta, quando comparado com o resultado da abordagem tradicional, é superior. Tal fato é constatado quando a aplicação da abordagem proposta incorre em um menor impacto financeiro (perda financeira + custo da infra-estrutura de TI) e melhores valores de lucro (para o caso do projeto de SLAs que envolvem serviços terceirizados).

Capítulo 5

Conclusões e Trabalhos Futuros

No decorrer deste capítulo serão tecidas as considerações finais sobre o presente trabalho de dissertação. A seção 5.1 sumariza as principais conclusões obtidas com os estudos realizados nesta dissertação. Por fim, a seção 5.2 discute idéias de possíveis próximos passos para este trabalho.

Nesta dissertação, realizou-se uma investigação formal de técnicas para (i) projetar infra-estruturas de Tecnologia da Informação e (ii) escolher parâmetros de Acordos de Níveis de Serviço de TI segundo uma perspectiva de negócio, de forma a otimizar métricas de negócios. Esta investigação resultou na proposta de uma abordagem que foi aplicada em diversos estudos realizados nos artigos constantes nos apêndices A, B, C, D e E. Como mostrado nos referidos apêndices, a principal diferença desta abordagem, quando comparada com abordagens convencionais, diz respeito às métricas consideradas. Tradicionalmente, as abordagens convencionais estão baseadas apenas em métricas técnicas. A abordagem proposta considera, além das métricas técnicas convencionais, o impacto negativo nos negócios, oriundo a partir de falhas na infra-estrutura de TI e de degradações de performance. Tal impacto negativo é obtido através de um modelo de impacto, conforme proposto por BDIM, cuja função é criar uma relação causa-efeito entre a TI. Este mapeamento (relação causa-efeito) entre eventos ocorridos na TI com o impacto quantitativo destes nos resultados do negócio, fornece uma base de informação que propicia a tomada de decisões que melhorem os resultados do negócio em si e também da TI. Neste sentido, tal informação permitiu que dois problemas em especial da gerência de TI (projeto de SLAs e de infra-estruturas) fossem reexaminados. Um modelo de impacto em particular foi apresentado e utilizado por este trabalho

nos diversos estudos realizados. Com ele é possível quantificar o impacto que eventos de TI têm nos negócios, o que serve como fonte de informação propiciando tomada de decisões que melhorem os resultados do negócio em si e também da TI. Note que a TI neste caso funciona como um elemento de apoio às operações de negócio.

A idéia básica por trás da abordagem proposta consiste em minimizar a soma do custo da infra-estrutura de TI com o impacto negativo no negócio, de forma a achar a infra-estrutura ótima segundo a perspectiva de negócio. Uma vez achada a configuração ótima, pode-se *calcular*, e não mais escolher de maneira *ad hoc*, os valores de SLOs, caso o objetivo seja projetar um SLA. Tem-se, portanto uma escolha *a posteriori* dos SLOs. Esta é outra característica fundamentalmente diferente das abordagens convencionais. Nestas os valores dos SLOs são escolhidos *a priori* para depois projetar uma infra-estrutura de TI que os suporte. Ainda com relação à negociação de SLAs, este trabalho abordou as especificações intra-empresa e inter-empresas de SLAs. Por fim, também se realizou um estudo sobre os efeitos que picos esperados de demanda têm nos negócios.

Como resultados diretos do presente trabalho tem-se diversos artigos científicos submetidos ou publicados em conferências internacionais. Os referidos artigos e seus respectivos veículos de publicação são enumerados a seguir:

- "SLA Design from a Business Perspective" publicado no 16th IFIP/IEEE Distributed Systems: Operations and Management Workshop (DSOM 2005), em Outubro de 2005;
- "Business-Oriented Capacity Planning of IT Infrastructure to handle Load Surges" publicado no 2006 IEEE/IFIP Network Operations & Management Symposium (NOMS 2006), em Abril de 2006;
- "Optimal Design of E-Commerce Site Infrastructure from a Business Perspective" publicado na 39th Hawaii International Conference on System Science (HICSS 2006), em Janeiro de 2006;
- "Optimal Choice of Service Level Objectives from a Business Perspective" publicado no 12th Annual Workshop of HP OpenView University Association (HPOVUA 2005);
- "Business-Driven Service Level Agreement Negotiation and Service Provisioning" submetido ao 2006 Conference on Measurement and Simulation of Computer and

Telecommunication Systems (MASCOTS 2006), em Abril de 2006;

- "Business-Driven Design of Infrastructures for IT Services" Submetido ao periódico Performance Evaluation, em Maio de 2006.

5.1 Conclusões

A presente dissertação mostra que desconsiderar a perspectiva de negócio durante as atividades de gerência de TI pode resultar em grandes perdas financeiras – a conclusão geral desta dissertação. O grande problema está relacionado à possível não existência de sintonia com as reais necessidades do negócio. Dentre as diversas atividades existentes na gerência de TI, duas em particular foram escolhidas – projeto de infra-estruturas de TI e de SLAs – e repensadas em termos de uma perspectiva de negócio por meios de BDIM. A seguir são apresentadas conclusões mais específicas que foram obtidas ao se repensar cada uma das referidas atividades em função de BDIM. Tais conclusões configuram-se como um sumário das conclusões obtidas nos artigos científicos presentes nos Apêndices A, B, C, D e E, que figuram entre os pioneiros de BDIM.

No tocante ao projeto de infra-estruturas de TI para o contexto de comércio eletrônico, enumeram-se, mais especificamente, as seguintes conclusões. Em primeiro lugar, evidenciou-se uma possível alta perda financeira incorrida quando a configuração da infra-estrutura subjacente, necessária para suportar as operações de negócio, foi definida sem a adoção de uma abordagem focada nos negócios. De forma análoga a esta situação pode-se ter *overdesign* da infra-estrutura de TI, isto é, a escolha de uma configuração que superdimensiona as necessidades do negócio (Apêndice A). Além disso, evidenciou-se também que a importância de um BP pode afetar o projeto da infra-estrutura de TI que o suporta. Em outras palavras, quanto mais importante for o BP, maior é a tendência de se necessitar de uma melhor infra-estrutura de TI. No contexto deste trabalho, o grau de importância de um BP está diretamente relacionado ao faturamento gerado por ele.

Na seqüência, foi realizado um estudo sobre os efeitos que variações de demanda têm nos negócios. Variações na demanda imposta aos serviços de TI são bastante comuns, principalmente no contexto de comércio eletrônico. Estas podem se dar em grande escala (chamadas de picos de demanda) ou dentro de uma pequena faixa de valores (pequenas variações). Es-

tas últimas são consideradas variações aceitáveis em torno do valor médio de demanda, e estão associadas à inerente característica estocástica da demanda imposta à infra-estrutura de TI. Por outro lado, os picos de demanda são bastante comuns nos períodos que antecedem datas comemorativas como, por exemplo, o dia das mães ou Natal. Nossos estudos sugerem que durante a ocorrência das referidas pequenas variações, o projeto de uma infra-estrutura estática de TI, segundo a perspectiva dos negócios, tem a tendência de permanecer ótimo ou muito perto do ótimo (Apêndice A). No entanto, na ocorrência de picos de demanda ou de variações de demanda fora da faixa aceitável de valores, o projeto de uma infra-estrutura estática de TI, segundo uma visão BDIM, tende torna-se rapidamente não-ótima (Apêndices A e B), gerando possíveis enormes perdas financeiras. Diante da iminência de perda financeira, há basicamente duas alternativas para evitá-la: considerar a priori os picos de demanda esperados ou planejados, durante o projeto da infra-estrutura ou utilizar uma infra-estrutura adaptativa. Adotar a primeira alternativa (Apêndice B) produz melhores resultados que a simples desconsideração de existência dos tais picos de demanda ou então, basear o projeto da infra-estrutura de TI nos picos de demanda. Neste último caso, ter-se-á muito provavelmente uma subutilização da infra-estrutura durante os períodos normais de demandas (Apêndice B). Além disso, foi constatado que, enquanto o projeto ótimo da infra-estrutura de TI tende a ser profundamente afetado por mudanças na intensidade do pico de demanda, a variação na duração, em geral, tem pouca influência no projeto ótimo. A segunda alternativa, utilização de infra-estruturas adaptativas segundo a perspectiva BDIM, é recomendada, pois se verificaram fortes indícios da aplicabilidade do modelo de impacto proposto e utilizado neste trabalho (Apêndice A).

Com relação ao projeto de SLAs, conclusões mais particulares são listadas a seguir. Primeiramente, a especificação de SLAs cujos valores de SLO não estão em conformidade com os objetivos gerais de negócio produz possíveis grandes perdas financeiras. Tais valores não necessariamente expressam a relação existente entre a TI e o negócio propriamente dito (Apêndice C). Quando o projeto de um SLA envolve serviços terceirizados, os resultados demonstram que a aplicação da abordagem proposta pode resultar em melhores valores de lucro tanto para o provedor quanto para o consumidor do serviço. Tais resultados são apresentados no Apêndice E. Convencionalmente, o provedor do serviço projeta uma infra-estrutura considerando as penalidades sob o seu ponto de vista, isto é, de forma a maximizar o

seu lucro. Desta forma, não necessariamente há uma expressão clara da correlação existente com os objetivos do negócio do cliente do serviço. Neste sentido, a abordagem proposta para a negociação de SLAs envolvendo serviços terceirizados considerou também a perda financeira do cliente do serviço.

Por fim, resta uma breve discussão sobre a aplicabilidade da abordagem proposta nesta dissertação. Como elencadas na próxima subseção, o presente trabalho possui diversas limitações. Porém, tais limitações estão relacionadas aos modelos analíticos usados pela abordagem e não se referem à abordagem propriamente dita. Por exemplo, o fato de os estudos desenvolvidos nesta dissertação terem sido realizados no contexto particular de comércio eletrônico (*e-commerce*) refere-se ao fato do modelo de impacto assumir total dependência dos BPs para com a infra-estrutura de TI. E, conforme menção anterior, a abordagem é perfeitamente genérica de forma a possibilitar que outros modelos de impacto mais gerais sejam adotados.

5.2 Trabalhos Futuros

Esta seção apresenta possíveis trabalhos futuros. Alguns destes possíveis trabalhos futuros endereçam limitações deste trabalho. A princípio, são abordados os aspectos relativos ao modelo de infra-estrutura de TI e, em seguida trata-se de questões relacionadas ao projeto de SLAs.

Um importante trabalho futuro diz respeito à realização dos possíveis passos, discutidos no capítulo 4, referentes à validação do presente trabalho. Tal tarefa é de extrema importância e contribui substancialmente para o aumento da validade, acurácia e confiabilidade dos modelos analíticos apresentados. Além disso, não são considerados elementos como rede, computadores pessoais dos clientes dos serviços e etc. A inexistência de tais elementos constitui uma limitação do presente trabalho e, dessa forma, modelá-los de forma que seja possível levá-los em consideração durante o projeto da infra-estrutura de TI representa um interessante trabalho futuro. Ainda, uma dada classe de recursos (*IT resource class*) possui apenas recursos de TI (*IT resources*) idênticos. Um segundo possível trabalho futuro seria considerar a existência de recursos de TI não-idênticos dentro de uma mesma classe de recursos de TI. Tal consideração implicaria em diferentes valores de custo, disponibilidade e

tempo de resposta entre os recursos de TI. Uma terceira limitação do presente trabalho está relacionada à observação do fato de que, por exemplo, é bastante comum as empresas já possuírem uma determinada infra-estrutura de TI. Neste caso, o projeto da infra-estrutura não é feito do zero, mas sim sob a forma de expansões incrementais da sua infra-estrutura de TI existente. Desta forma, evidencia-se como trabalho futuro investigar problemas mais reais de empresas de forma a endereçar situações como a descrita. Duas outras restrições dos modelos apresentados estão relacionadas à consideração de que os BPs dependem totalmente dos serviços de TI (*IT services*) e que estes, por sua vez, dependem de todas as classes de recursos. Dada a possibilidade de existirem BPs que não dependem totalmente dos serviços de TI e serviços de TI que não dependem de todas as classes de recursos de TI, um possível trabalho futuro seria considerar a existência de diferentes relações de dependência, e, principalmente, a existência de "*humans in the loop*" nos BPs. Além disso, uma vez que os projetos de um subsistema de armazenamento de dados e de um sistema organizado em n -camadas envolvem uma série de particularidades não tratadas no modelo apresentado, um interessante trabalho futuro seria considerar tais particularidades. Ainda com relação ao projeto de infra-estrutura de TI, enumera-se a seguinte lista contendo sugestões de trabalhos futuros: levar em conta a possibilidade de haver restrições com relação ao número de recursos de TI numa dada configuração em *cluster*. Por exemplo, pode ser que não seja possível configurar um dado *cluster* com 5 recursos, mas somente com um número múltiplo de 2; considerar virtualização da infra-estrutura de TI, de forma a possibilitar a alocação de apenas uma fração de componente de TI (*IT component*); investigar como utilizar a métrica impacto negativo nos negócios como um gatilho de re-otimização em um ambiente de infra-estrutura adaptativa; levar em conta a geografia dispersa dos usuários do serviço de TI, pois um serviço pode não estar fora do ar para todos os seus usuários; dessa forma, seria interessante redefinir a disponibilidade a partir de um ponto de vista geográfico [Coe00].

Por fim, no tocante ao projeto de SLA, enumeram-se as seguintes possibilidades: especificação de outros parâmetros do SLA, e não apenas seus SLOs. Recompensas e penalidades são exemplos de outros parâmetros que podem ser especificados; consideração de métricas de desempenho (SLIs) diferentes das consideradas neste trabalho. Atentar para a provável existência de restrições nos valores dos parâmetros de um SLA. Como exemplos de tais restrições tem-se a definição de um valor mínimo para um determinado SLO ou a especifi-

cação de uma faixa aceitável de valores para penalidades ou recompensas. Tais restrições são justificadas por razões políticas ou administrativas e podem ser facilmente acomodadas como restrições na otimização; e prever o acontecimento de uma determinada violação de SLA. Isto possibilitaria projetar o impacto negativo desta violação no negócio, permitindo assim a tomada de decisões quanto a viabilidade do emprego de ações para evitar tal violação [SB04].

Bibliografia

- [AAA06] Bruno Abrahão, Virgilio Almeida, and Jussara Almeida. Self-adaptive SLA-driven capacity management for internet services. In *17th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM 2006*, 2006.
- [AF02] Danilo Ardagna and Chiara Francalanci. A cost-oriented methodology for the design of web-based it architectures. In *SAC '02: Proceedings of the 2002 ACM symposium on Applied computing*, pages 1127–1133, New York, NY, USA, 2002. ACM Press.
- [AFT04] Danilo Ardagna, Chiara Francalanci, and Marco Trubian. A cost-oriented approach for infrastructural design. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 1431–1437, New York, NY, USA, 2004. ACM Press.
- [AGL⁺04] Sarel Aiber, Dagan Gilat, Ariel Landau, Natalia Razinkov, Aviad Sela, and Segev Wasserkrug. Autonomic self-optimization according to business objectives. In *1st International Conference on Autonomic Computing (ICAC 2004)*, pages 206–213, 2004.
- [ASBB04] Issam Abi, Mathias Sallé, Claudio Bartolini, and Abdel Boulmakoul. A business driven management framework for utility computing environments. Research Report HPL-2004-171, Hewlett-Packard, 2004.
- [BCL⁺03] Melissa Bucu, Rong Chang, Laura Luan, Chris Ward, Joel Wolf, Philip Yu, Tevfik Kosar, and Syed Umair Shah. Managing ebusiness on demand sla contracts in business terms using the cross-sla execution manager sam. In *ISADS*

- '03: *Proceedings of the The Sixth International Symposium on Autonomous Decentralized Systems (ISADS'03)*, page 157, Washington, DC, USA, 2003. IEEE Computer Society.
- [BCL⁺04] Melissa J. Bucu, Rong N. Chang, Laura Z. Luan, Christopher Ward, Joel L. Wolf, and Philip S. Yu. Utility computing sla management based upon business objectives. *IBM Systems Journal*, 43(1):159–178, 2004.
- [BKP02] Jan Van Bon, George Kemmerling, and Dick Pondman. *IT Service Management: An Introduction*. Van Haren Publishing, 2002.
- [BR03] Erik Beulen and Pieter M. A. Ribbers. It outsourcing contracts: Practical implications of the incomplete contract theory. In *HICSS*, pages 268–, 2003.
- [BS04a] Claudio Bartolini and Mathias Sallé. Business-driven prioritization of service incidents. In Sahai and Wu [dso04], pages 64–75.
- [BS04b] Claudio Bartolini and Mathias Sallé. Business-driven prioritization of service incidents. In Sahai and Wu [dso04], pages 64–75.
- [BTdZ99] Jacques Bouman, Jos Trienekens, and Mark Van der Zwan. Specification of service level agreements, clarifying concepts on the basis of practical research. *step*, 00:169, 1999.
- [CCD⁺02] Fabio Casati, Malu Castellanos, Umesh Dayal, Ming Hao, Ming-Chien Shan, and Mehmet Sayal. Business operation intelligence research at hp labs. In *Data Engineering Bulletin* 25(4), 2002.
- [CDD⁺04] Tod Courtney, David Daly, Salem Derisavi, Shravan Gaonkar, Mark Griffith, Vinh Vi Lam, and William H. Sanders. The Möbius modeling environment: Recent developments. In *QEST'04: Quantitative Evaluation of Systems*, pages 328–329, 2004.
- [Coe00] Flávia Estélia Silva Coelho. Avaliação de disponibilidade de redes de computadores baseadas na arquitetura cliente/servidor em N-camadas, orientador: Jacques philippe sauvé. Master's thesis, Universidade Federal de Campina Grande, 2000.

- [CSDS03] Fabio Casati, Eric Shan, Umeshwar Dayal, and Ming-Chien Shan. Business-oriented management of web services. *Commun. ACM*, 46(10):55–60, 2003.
- [DCR04] Budi Darmawan, Kimberly Cox, and Bahaeldin Ragab. *Business Service Management Best Practices*. IBM, 2004.
- [DHP01] Y. Diao, J.L. Hellerstein, and S. Parekh. A business-oriented approach to the design of feedback loops for performance management. In *12th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM*, 2001.
- [dso04] Utility computing: 15th ifip/ieee international workshop on distributed systems: Operations and management, dsom 2004, davis, ca, usa, november 15-17, 2004.proceedings. In Akhil Sahai and Felix Wu, editors, *15th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM 2004*, volume 3278 of *Lecture Notes in Computer Science*. Springer, 2004.
- [Dub02] Denise Dubie. Never-fail business services, <http://www.nwfusion.com/buzz/2002/bim.html>, 2002.
- [Ele03] Pink Elephant. Selling ITIL - how to prepare & build a case to build internal commitment to promote ITIL best practices within an organization, <http://www.pinkelephant.com>, April 2003.
- [Far03] A. D. H. Farrel. Logic-based formalisms for the representation of service-level agreements for utility computing. Msc thesis, Department of Computing, Imperial College of Science, Technology and Medicine, London, 2003.
- [GLS04] Dagan Gilat, Ariel Landau, and Aviad Sela. Autonomic self-optimization according to business objectives. In *ICAC '04: Proceedings of the First International Conference on Autonomic Computing (ICAC'04)*, pages 206–213, Washington, DC, USA, 2004. IEEE Computer Society.
- [HP] Hewlett-Packard. HP ITSM reference model, <http://h20219.www2.hp.com/services/cache/78360-0-0-225-121.aspx>.

- [HPS86] Paul Gerhard Hoel, Sidney C Port, and Charles J Stone. *Introduction to Stochastic Processes*. Waveland Press, 1986.
- [Ins01] IT Governance Institute. Board briefing on IT governance, <http://www.itgovernance.org/resources.htm>, 2001.
- [JST03] G. John Janakiraman, Jose Renato Santos, and Yoshio Turner. Automated multi-tier system design for service availability. In *First Workshop on Design of Self-Managing Systems*, 2003.
- [KC03] Jeffrey O. Kephart and David M. Chess. The vision of autonomic computing. *IEEE Computer*, 36(1):41–50, 2003.
- [KHW⁺04] Alexander Keller, Joseph Hellerstein, J.L. Wolf, K.Wu, and V. Krishnan. The champs system: Change management with planning and scheduling. In *NOMS 2004: IEEE/IFIP Network Operations and Management Symposium*. IEEE Computer Society, 2004.
- [Kle76a] Kleinrock. *Queuing Systems*, volume I. Wiley, 1976.
- [Kle76b] Leonard Kleinrock. *Queueing Systems, Volume I: Theory*. Wiley-Interscience, New York, 1976.
- [LCD⁺03] E. Lassetre, D. Coleman, Y. Diao, S. Froelich, J. Hellerstein, L. Hsiung, T. Mummert, M. Raghavachari, G. Parker, L. Russell, M. Surendra, V. Tseng, N. Wadia, and P. Ye. Dynamic surge protection: An approach to handling unexpected workload surges with resource actions that have lead times, 2003.
- [Lew99a] Lundy Lewis. *Service Level Management for Enterprise Networks*. Artech House Publishers, 1999.
- [Lew99b] Lundy Lewis. *Service Level Management for Enterprise Networks*. Artech House, 1999.
- [LR99] Lundy Lewis and Pradeep Kumar Ray. Service level management definition, architecture, and research challenges. In *GLOBECOM '99: Proceedings of Global Telecommunications Conference*, volume 3, pages 1974–1978, 1999.

- [LSE03] D. Davide Lamanna, James Skene, and Wolfgang Emmerich. Slang: A language for defining service level agreements. In *9th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS 2003)*, 28-30 May 2003, San Juan, Puerto Rico, *Proceedings*, pages 100–. IEEE Computer Society, 2003.
- [LSW01] Zhen Liu, Mark S. Squillante, and Joel L. Wolf. On maximizing service-level-agreement profits. In *ACM Conference on Electronic Commerce*, pages 213–223. ACM, 2001.
- [MA00] Daniel A. Menascé and Virgilio A. F. Almeida. *Scaling for E-Business*. Prentice Hall, 2000.
- [MAD04] Daniel A. Manascé, Virgilio A. F. Almeida, and Lawrence W. Dowdy. *Performance by Design: Computer Capacity Planning by Example*. Prentice-Hall PTR, 2004.
- [MAFM00] Daniel A. Menascé, Virgilio Almeida, Rodrigo C. Fonseca, and Marco A. Mendes. Business-oriented resource management policies for e-commerce servers. *Perform. Eval.*, 42(2-3):223–239, 2000.
- [MAR⁺03a] D. A. Menascé, V. A. F. Almeida, R. Riedi, F. Ribeiro, R. Fonseca, and W. Meira Jr. A hierarchical and multiscale approach to analyze E-business workloads. *Performance Evaluation*, 54(1):33–57, September 2003.
- [MAR⁺03b] Daniel A. Menascé;, Virgilio A. F. Almeida;, R. Riedi;, F. Ribeiro;, R. Fonseca;, and J. W. Meira. A hierarchical and multiscale approach to analyze ebusiness workloads. In *Performance Evaluation*, pages 224–234, New York, NY, USA, 2003. ACM Press.
- [MAR⁺03c] Danniell A. Menascé, Virgilio A. F. Almeida, R. Riedi, F. Ribeiro, R. Fonseca, and Jr. W. Meira. A hierarchical and multiscale approach to analyze e-business workloads. *Performance Evaluation*, 54(1):33–57, 2003.
- [Mas02] Paul Mason. A new culture for service-level management: Business impact management, <http://www->

- 900.ibm.com/cn/software/tivoli/products/download/analystreports/ar-service-level-mgt.pdf, 2002.
- [MBC04] Vijay Machiraju, Claudio Bartolini, and Fabio Casati. Technologies for business driven it management. In Lawrence Cavedon, Zakaria Maamar, David Martin, and Boualem Benatallah, editors, *Extending Web Services Technologies: The Use of Multi-Agent Approaches*, volume 13 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*. Springer, 2004.
- [MBD01] Daniel A. Menascé;, Daniel Barbará;, and Ronald Dodge. Preserving QoS of e-commerce sites through self-tuning: a performance model approach. In *EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 224–234, New York, NY, USA, 2001. ACM Press.
- [Mic] Microsoft. Microsoft operations framework (MOF), <http://www.microsoft.com/technet/itsolutions/cits/mo/mof/default.mspx>.
- [MV00] Daniel A. Menasce and A. F. Almeida Virgilio. *Scaling for E Business: Technologies, Models, Performance, and Capacity Planning*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.
- [Pat02] David A. Patterson. A simple way to estimate the cost of downtime. In *Proceedings of the 16th USENIX System Administration Conference — LISA 2002*. USENIX, 2002.
- [Pie] Jay St. Pierre. test_tools, <http://www.mathworks.com/matlabcentral>.
- [SB04] Mathias Sallé and Cláudio Bartolini. management by contract. In *NOMS 2004: IEEE/IFIP Network Operations and Management Symposium*, pages 787–800. IEEE Computer Society, 2004.
- [Sch00] Holger Schmidt. Service contracts based on workflow modeling. In *Services Management in Intelligent Networks, 11th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM 2000*, pages 132–144, 2000.

- [SDM01] A. Sahai, A. Durante, and V. Machiraju. Towards automated sla management for web services. Research Report HPL-2001-310, Hewlett-Packard, 2001.
- [SM00] Rick Sturm and Wayne Morris. *Foundations of Service Level Management*. Sams, 2000.
- [SMM05a] Jacques Philippe Sauvé, Filipe Teixeira Marques, and José Antão Beltrão Moura. Business-oriented capacity planning of it infrastructure to handle load surges. Technical report, Universidade Federal de Campina Grande, 2005.
- [SMM⁺05b] Jacques Philippe Sauvé, Filipe Teixeira Marques, José Antão Beltrão Moura, Marcus Costa Sampaio, João Francisco Homrich da Jornada, and Eduardo Radziuk. Optimal design of e-commerce site infrastructure from a business perspective. Technical report, Universidade Federal de Campina Grande, 2005.
- [SMM⁺05c] Jacques Philippe Sauvé, Filipe Teixeira Marques, José Antão Beltrão Moura, Marcus Costa Sampaio, João Francisco Homrich da Jornada, and Eduardo Radziuk. SLA design from a business perspective. In *16th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM 2005*, Lecture Notes in Computer Science. Springer, 2005.
- [SMM06a] Jacques Philippe Sauvé, Filipe Teixeira Marques, and José Antão Beltrão Moura. Business-oriented capacity planning of it infrastructure to handle load surges. In *Proceedings of the 2006 IEEE/IFIP Network Operations and Management Symposium (NOMS 2006)*, 2006.
- [SMM⁺06b] Jacques Philippe Sauvé, Filipe Teixeira Marques, José Antão Beltrão Moura, Marcus Costa Sampaio, João Francisco Homrich da Jornada, and Eduardo Radziuk. Optimal design of e-commerce site infrastructure from a business perspective. In *Proceedings of the 39th Hawaii International Conference on System Sciences (HICSS)*. IEEE Computer Society, 2006.
- [SMS⁺04] Jacques Philippe Sauvé, José Antão Beltrão Moura, Marcus Costa Sampaio, João Francisco Homrich da Jornada, and Eduardo Radziuk. An IT business impact management framework. RT-DSC 1/2004, Universidade Federal

- de Campina Grande, <http://jacques.dsc.ufcg.edu.br/projetos/bl-hp/reports/001-2004.pdf>, January 2004.
- [SMS⁺05] Jacques Philippe Sauvé, José Antão Beltrão Moura, Marcus Costa Sampaio, João Francisco Homrich da Jornada, and Eduardo Radziuk. Business Impact Management - characterization through scenarios, hp-ufcg bottom line project 2005 internal report. January 2005.
- [SMS⁺06] Jacques Sauvé, Antão Moura, Marcus Sampaio, João Jornada, and Eduardo Radziuk. An introductory overview and survey of businessdriven it management. In *1st IEEE/IFIP International Workshop on Business-Driven IT Management*, 2006.
- [SPEa] Standard Performance Evaluation Corporation SPEC. Specjappserver2004 benchmark, <http://www.spec.org/jappserver2004/>.
- [SPEb] Standard Performance Evaluation Corporation SPEC. Specweb2005 - standard performance evaluation corporation – (SPEC) benchmark, <http://www.spec.org/web2005/>.
- [ST87] R. A. Sahner and K. S. Trivedi. Reliability modeling using SHARPE. *IEEE Transactions on Reliability*, R-36(2):186–193, June 1987.
- [TPCa] Transaction Processing Performance Council TPC. Tpc pricing spec, <http://www.tpc.org/pricing/default.asp>.
- [TPCb] Transaction Processing Performance Council TPC. Transaction processing performance council – TPC, <http://www.tpc.org/>.
- [Tri82] Kishor Shridharbhai Trivedi. *Probability and statistics with reliability, queuing, and computer science applications*. Prentice-Hall PTR, 1982.
- [TT05a] R. Taylor and C. Tofts. Death by a thousand SLAs: A short study of commercial suicide pacts. Research Report HPL-2005-11, Hewlett-Packard, 2005.
- [TT05b] Richard Taylor and Chris Tofts. Death by a thousand SLAs: a short study of commercial suicide pacts. Technical Report HPL-2005-11, Hewlett-Packard

Company, Trusted Systems Laboratory, HP Laboratories Bristol, January 2005.

[UPS⁺05] Bhuvan Urgaonkar, Giovanni Pacifici, Prashant J. Shenoy, Mike Spreitzer, and Asser N. Tantawi. An analytical model for multi-tier internet services and its applications. In *SIGMETRICS*, 2005.

[Web] WebStone. Webstone benchmark, <http://www.mindcraft.com/webstone/>.

[Wor] Math Works. Matlab, <http://www.mathworks.com/>.

Appendix A

Optimal Design of e-Commerce Site Infrastructure from a Business Perspective

Jacques Sauvé¹, Filipe Marques¹, Antão Moura¹, Marcus Sampaio¹, João Jornada² and Eduardo Radziuk²

¹*Universidade Federal de Campina Grande*

{jacques,filipetm,antao,sampaio}@dsc.ufcg.edu.br

²*Hewlett-Packard-Brazil*

{joao.jornada,eduardo.radziuk}@hp.com

Abstract: A methodology for designing data center infrastructure for E-commerce sites is developed. It differs from existing methodologies in that it evaluates and compares alternative designs from a business perspective, that is, by evaluating the business impact (financial loss) imposed by imperfect infrastructure. The methodology provides the optimal infrastructure that minimizes the sum of provisioning costs and business losses incurred during failures and performance degradations. A full numerical example design is provided and results are analyzed. The use of the method for dynamically provisioning an adaptive infrastructure is briefly discussed. ¹

¹Publicado na 39th Hawaii International Conference on System Science (HICSS 2006), em Janeiro de 2006.
/ Accepted for publishing in the 39th Hawaii International Conference on System Science (HICSS 2006), in January of 2006.

A.1 Introduction

The problem addressed in this paper is that of infrastructure design for Information Technology (IT) services that cater to business processes that are heavily dependent on IT. An example of such a business process is that supported by an e-commerce site: IT services are the main technology support in such a context and any failure or performance degradation in the IT infrastructure can profoundly affect business operations.

Within the general problem of designing infrastructure to provision IT services, the work reported here concentrates on the data center, that part of the infrastructure most easily controllable by the service provider. Current approaches in data center design usually either consider the problem from a reliability point of view, e.g. [JST03], from a response time point of view, e.g. [MBD01] or, more recently, from a business perspective, e.g. [GLS04]. The last approach is more novel and merits some discussion.

A new area of academic research – and also of the practitioner’s art – is termed Business-Driven IT Management (BDIM) [SB04; BS04a; SMM⁺05c]. BDIM takes Service Management (SM) to a new maturity level since metrics meaningful to the customer are used to gauge IT effectiveness rather than technical metrics such as availability and response time. This is the crucial departure that the present work takes on most past efforts.

In the present study, infrastructure design aims to decide how many and what kind of resource components should be used to provision IT services. Clearly, adding more fail-over servers will improve service availability and adding more load-balanced servers will lower response time. But what values of availability or of response time should the designer aim for? How does one combine requirements on availability and requirements on response time into coherent design decisions? BDIM answers this question as follows: the impact of any IT infrastructure imperfection should be gauged in terms of its impact on business as captured by *business metrics*. The design decisions should then be evaluated in terms of the business impact caused by the resulting design.

This paper provides a concrete business impact model that includes the impact of IT component failures on service availability and, in turn, on the business and the impact of load on performance (response time) and, in turn, on the business. Using this impact model, the problem of designing optimized IT infrastructure is formally defined and solved analytically.

The rest of the paper is structured as follows: section A.2 informally discusses the design problem using a BDIM approach while section A.3 formalizes it; section A.4 considers an application of the method through a full numerical example; section A.5 discusses related work; conclusions are provided in section A.6.

A.2 Informal Problem Description

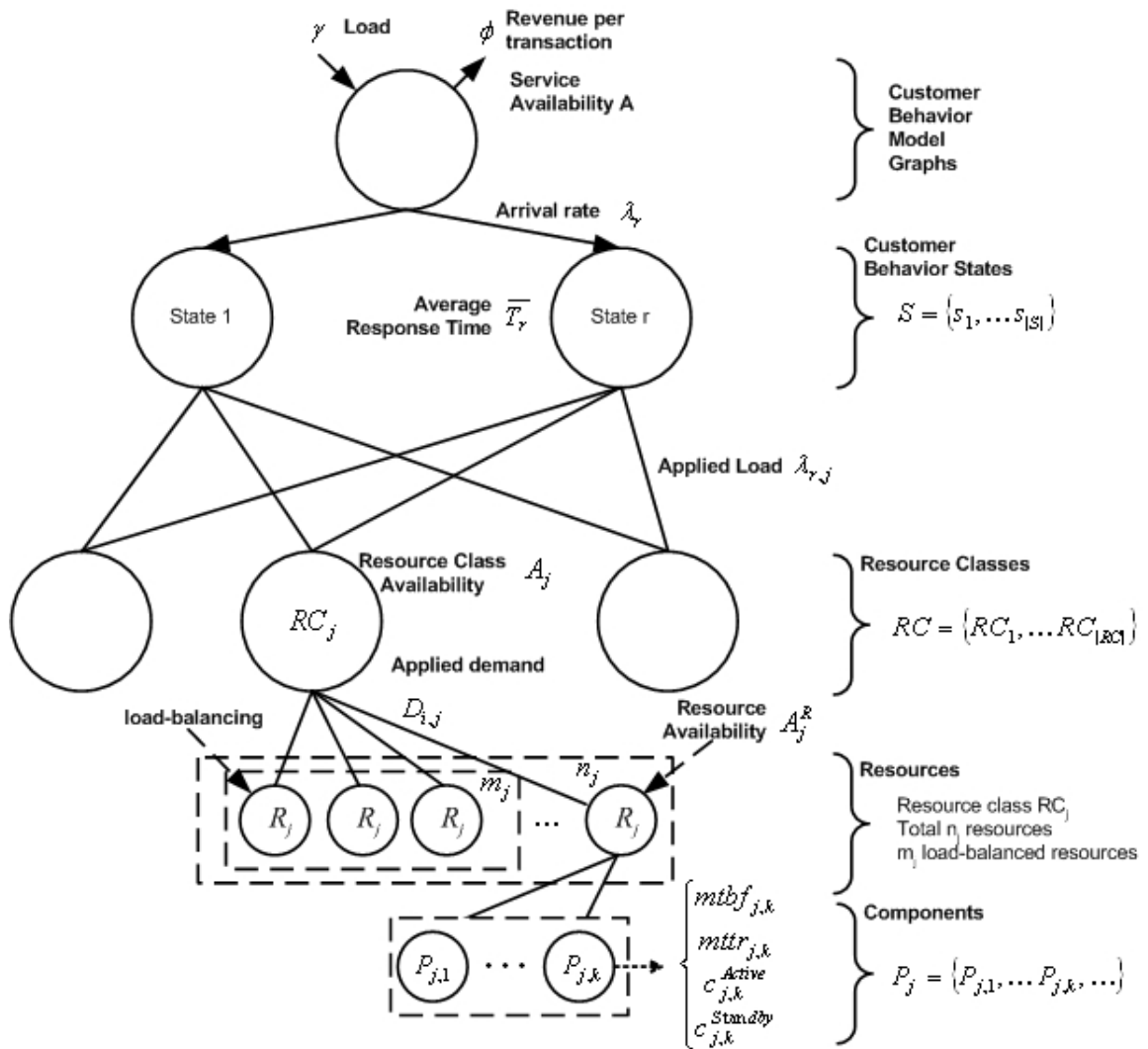


Figure A.1: Model entities

Infrastructure must be designed for an e-commerce site. The approach taken was first suggested in [SMM⁺05c] and aims to minimize monthly financial outlays as calculated by the infrastructure cost plus the business loss incurred due to the imperfect infrastructure.

Thus, the approach uses a business perspective in the design process through a business impact model. Two kinds of imperfections present in the IT infrastructure are considered, both generating business loss. The first is that components may fail, rendering the service unavailable part of the time. The second is that the load imposed on the infrastructure components results in delays, with the possibility of customers defecting due to overlarge delays.

Sessions visiting the site are divided into two types: revenue-generating sessions where, at some point during the visit, some revenue will accrue to the site's owner; in the second type of session, customers may visit pages on the site, may possibly even be adding items to a shopping cart, but end up desisting before generating revenue.

The infrastructure itself consists of several tiers, say a web tier, an application tier and a data tier. Each tier is served by a load-balanced cluster with a certain number of machines, sufficient to handle the applied load. Varying this number of machines affects response time and thus the business loss due to customer defections. Furthermore, additional machines are available in standby mode to improve site availability and hence reduce business losses due to service unavailability.

The problem studied here is to choose the best infrastructure configuration (number and type of machine in each tier's load-balanced cluster and the number of standby machines), that is, the configuration that minimizes monthly cost plus business losses.

A.3 Problem Formalization

This section formalizes the infrastructure design problem. The analysis uses results from reliability theory, queueing theory and extends a novel business impact model presented in [SMM⁺05c].

A.3.1 The Design Optimization Problem

Let us first define the design problem to be solved. Please refer to Table A.1 for a notational summary and to Figure A.1 for a summary of the entities involved.

The infrastructure provisioning the e-commerce site is made up of a set RC of resource classes. For example, the resource classes could correspond to tiers (web tier, application tier, data tier). Resource class RC_j is provisioned with a total of n_j machines, of which m_j

Table A.1: Notational summary for problem definition

Symbol	Meaning
RC	Set of resource classes in IT infrastructure (e.g. tiers)
RC_j	The j th resource class
n_j	The total number of resources (machines) in RC_j
m_j	The total number of load-balanced machines in RC_j
ΔT	Any time period over which cost and loss are evaluated. Typically a month
$C(\Delta T)$	The infrastructure cost over the time period ΔT
$L(\Delta T)$	The financial loss over the time period ΔT due to imperfections in the infrastructure

make up a load-balanced cluster while the rest are standby machines. The load-balanced machines enable the tier to handle the input load while the standby machines provide the required availability. The design problem can be posed as an optimization problem as shown in table A.2 .

Table A.2: Formal definition of design problem

Find:	For each resource class RC_j , the total number of machines n_j and the number of load-balanced machines m_j
By minimizing:	$C(\Delta T) + L(\Delta T)$, the total financial impact on the business over the time period ΔT
Subject to:	$n_j \geq m_j$ and $m_j \geq 1$

One must now derive expressions for $L(\Delta T)$ and $C(\Delta T)$, which we now proceed to do.

A.3.2 Characterizing the Infrastructure

In this section, expressions for the infrastructure cost $C(\Delta T)$ and for site availability, A , are developed. Site availability will be used in a later section to derive an expression for loss, $L(\Delta T)$. Please refer to Table A.3 for a notational summary.

Table A.3: Notational summary for problem definition

Symbol	Meaning
R_j	An individual resource in RC_j
P_j	The set of components that make up resource R_j
$P_{j,k}$	The k^{th} component in P_j
$c_{j,k}^{active}$	The cost rate of component $P_{j,k}$ if active
$c_{j,k}^{standby}$	The cost rate of component $P_{j,k}$ if on standby
A	The site availability
A_j	The availability of resource class RC_j
A_j^R	The availability of an individual resource R_j from class RC_j
$mtbf_{j,k}$	The Mean-Time-Between-Failures of component $P_{j,k}$
$mttr_{j,k}$	The Mean-Time-To-Repair of component $P_{j,k}$

As mentioned previously, the infrastructure used to provision the e-commerce site consists of a set of resource classes, $\{RC_1, \dots, RC_{|RC|}\}$. Class RC_j consists of a cluster of IT resources. This cluster has a total of n_j identical individual resources, up to m_j of which are load-balanced and are used to provide adequate processing power to handle incoming load. The resources that are not used in a load-balanced cluster are available in standby (fail-over) mode to improve availability.

An individual resource $R_j \in RC_j$ consists of a set $P_j = \{P_{j,1}, \dots, P_{j,k}, \dots\}$ of components, all of which must be operational for the resource to also be operational. As an example, a single Web server can be made up of the following components: server hardware, operating system software and Web software. Individual components are subject to faults as will be described later.

Determining infrastructure cost. Each infrastructure component $P_{j,k}$ has a cost rate $c_{j,k}^{active}$ when active (that is, used in a load-balanced server) and has a cost rate $c_{j,k}^{standby}$ when on standby. These values are cost per unit time for the component and may be calculated as its total cost of ownership (TCO) divided by the amortization period for the component. The cost of the infrastructure over a time period of duration ΔT can be calculated as the sum of

individual cost for all components.

$$C(\Delta T) = \Delta T \cdot \sum_{j=1}^{|RC|} \left(\sum_{l=1}^{m_j} \sum_{k=1}^{|P_j|} c_{j,k}^{active} + \sum_{l=1}^{n_j-m_j} \sum_{k=1}^{|P_j|} c_{j,k}^{standby} \right)$$

Determining service availability. Recall that IT components making up the infrastructure can fail, producing unavailability and hence business loss. In order to calculate business loss, one needs to evaluate the availability A of the site. This is done using standard reliability theory [Tri82]. For service to be available, all resource classes it uses must be available. Thus:

$$A = \prod_{j \in RC} A_j$$

where A_j is the availability of resource class RC_j . Since this resource class consists of a cluster of n_j individual resources, and since service will be available and able to handle the projected load when at least m_j resources are available for load-balancing, one has, from reliability theory:

$$A_j = \sum_{k=m_j}^{n_j} \left[\binom{n_j}{k} \cdot (A_j^R)^k \cdot (1 - A_j^R)^{n-k} \right]$$

where A_j^R is the availability of an individual resource R_j from class RC_j . This individual resource is made up of a set P_j of components, all of which must be operational for the resource to be operational. Thus:

$$A_j^R = \prod_{k \in P_j} \left[\frac{mtbf_{j,k}}{mtbf_{j,k} + mttr_{j,k}} \right]$$

where $mtbf_{j,k}$ and $mttr_{j,k}$ are, respectively, the Mean-Time-Between-Failures (MTBF) and Mean-Time-To-Repair (MTTR) of component $P_{j,k}$. Observe that values from MTBF can be obtained from component specifications or historical logs whereas values for MTTR will typically depend on the type of service contract available.

A.3.3 The Response Time Performance Model

Since business loss occurs for high values of response time – defection typically occurs when response time reaches 8 seconds [MA00] – this section uses queueing theory to obtain an expression for $B(T^{DEF})$, the probability that response time has exceeded T^{DEF} , the defection threshold, and that revenue-generating customers will therefore defect. Please refer to Table A.4 for a notational summary.

Table A.4: Notational summary for problem definition

Symbol	Meaning
T^{DEF}	The response time threshold after which customer defection occurs
$B(y)$	The probability that response time is greater than y
S	The set of states in the Customer Behavior Model Graph. Each state represents a particular interaction with the e-commerce site (browse, search, etc.)
γ	The rate at which sessions are initiated at the site
f	The fraction of sessions that generate revenue (type RG sessions)
$p_{i,r}^{RG}$	Probability of going from state i to state r in the RG CBMG
$p_{i,r}^{NRG}$	Probability of going from state i to state r in the NRG CBMG
V_r^{RG}	Average number of visits to state r in RG CBMG
V_r^{NRG}	Average number of visits to state r in NRG CBMG
λ_r	Arrival rate of requests to IT infrastructure in state r
α_j	Speedup factor for resources in resource class RC_j
$T_r(y)$	Cumulative distribution of response time for requests in state r
NZ^{RG}	Set of states from the RG CBMG that have non-zero average number of visits

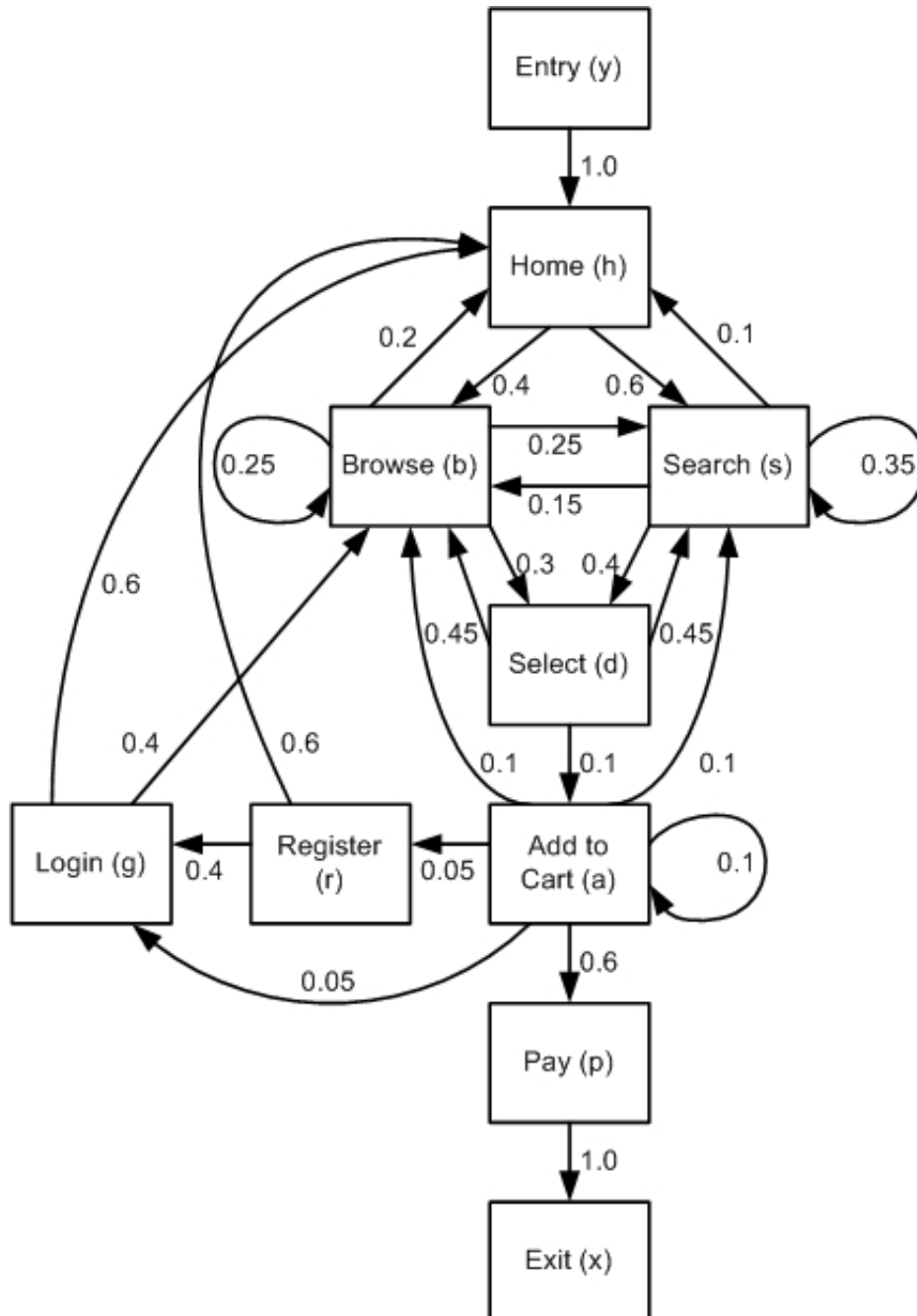


Figure A.2: CBMG for the e-commerce site

In order to assess response time performance, one must model the load applied to the IT resources. Access to the e-commerce site consists of sessions, each generating several visits to the site's pages. The mathematical development that follows is (initially) based on the Customer Behavior Model Graph (CBMG) [MA00], that allows one to accurately model how customer-initiated sessions accessing a web site impose load on the IT infrastructure. The use of the CBMG model will then be extended to include business impact.

A CBMG consists of a set S of states and transitions between states occurring with particular probabilities. Each state typically represents a web site page that can be visited and where a customer interacts with the e-commerce site. As an example, consider Figure A.2 that shows the states and the transition probabilities for a simple but typical e-commerce site. The customer always enters through the Home state and will then Browse (with probability 0.4) or Search (with probability 0.6). The Select state represents viewing the details of a product and the other states are self-explanatory.

Some of these states are revenue-generating (for example, a state "Pay" where the customer pays for items in a cart). Sessions are initiated at a rate of γ sessions per second. For our purposes, we divide the sessions into two types: type RG sessions generate revenue while type NRG sessions do not. Customer behavior for each session type is modeled by means of its own CBMG [MA00]. The particular CBMG shown in Figure A.2 is an example applicable to type RG sessions since the Pay state is visited with non-zero probability. For type NRG sessions, the CBMG will include the same states but with different probabilities. For example, there will be no path leading to the Pay state, the only revenue-generating state in this particular graph. The fraction of sessions that are revenue-generating is denoted by f . The transition probability matrices have elements $p_{i,r}^{RG}$, the probability of going from state i to state r in the RG CBMG, and $p_{i,r}^{NRG}$ for the NRG CBMG. Observe that S , f and all transition probabilities for these graphs can be obtained automatically from web server logs.

As shown in [MA00], flow equilibrium in the graph can be represented by a set of linear equations that can be solved to find the average number of visits per session to state r . The set of equations to be solved for the RG CBMG is:

$$V_1^{RG} = 1$$

$$V_r^{RG} = \sum_{i=1}^{|S|} (V_i \cdot p_{i,r}^{RG})$$

Table A.5: Transition probabilities in NRG CBMG

	y	h	b	s	g	p	r	a	d	x
Entry (y)		1.00								
Home (h)			0.55	0.40						0.05
Browse (b)		0.10	0.50	0.20					0.10	0.10
Search (s)		0.10	0.15	0.40					0.25	0.10
Login (g)		0.60	0.30							0.10
Pay (p)										1.00
Register (r)		0.50			0.40					0.10
Add to Cart (a)			0.40	0.30	0.05		0.05	0.05	0.10	0.05
Select (d)			0.45	0.40				0.05		0.10

In this set of equations, the average number of visits in the RG CBMG is V_r^{RG} . The situation for the NRG CBMG is similar and the average number of visits is V_r^{NRG} .

We now need to find $B(T^{DEF})$, the probability that customers will defect due to response time exceeding T^{DEF} while navigating. Customer defection will occur and cause business loss only in the revenue-generating sessions. Let NZ^{RG} represent the set of states from the RG CBMG that have non-zero average number of visits. The crucial fact to be understood is that if the response time in *any* of the states in NZ^{RG} exceeds the threshold T^{DEF} , then defection will occur; in other words, a customer defects when any page access becomes too slow. Let the cumulative distribution of response time in state r be $T_r(y) = \Pr[\tilde{T}_r \leq y]$, where \tilde{T}_r is the random variable corresponding to the response time seen by the customer in state r . $T_r(T^{DEF})$ is the probability that there will be no defection in a visit to state r . Thus, since defection will *not* occur if all response times are within the threshold, we can say:

$$B(T^{DEF}) = 1 - \prod_{r \in NZ^{RG}} T_r(T^{DEF})$$

In order to find $T_r(T^{DEF})$, the IT services are modeled using a multi-class open queueing model. Open queueing models are adequate when there is a large number of potential customers, a common situation for e-business. Since, in each state, the demands made on the IT infrastructure are different, each state in the CBMG represents a

traffic class in the queueing model. Standard queueing theory [Kle76b] can be used to solve this model by considering the arrival rate of requests corresponding to state r as $\lambda_r = \gamma \cdot (f \cdot V_r^{RG} + (1 - f) \cdot V_r^{NRG})$ transactions per second. Observe that, in this analysis, some simplifications are made to make mathematical treatment feasible. The assumptions are:

1. Poisson arrivals are assumed (this is a reasonable for stochastic processes with large population) and also exponentially distributed service times. This assumption is necessary to find the probability distribution function of response time using Laplace transforms [Kle76b]. This assumption is frequently made when analyzing performance [MAD04]. Also, since the numerical results will be used primarily to *compare* designs, one expects little sensitivity to particular distributions.
2. Since there are m_j identical load-balanced parallel servers used for processing in resource class RC_j , response time is calculated for an equivalent single server with input load reduced by a factor of m_j [MA00].

Details of the full mathematical development can be found in [SMM⁺05b].

A.3.4 The Business Impact Model

The key expressions to be used in estimating business loss have been determined in the last sections. These are site availability, A , and the defection probability for revenue-generating sessions, $B(T^{DEF})$. These are now combined to calculate business loss.

Revenue-generating sessions are initiated at a rate of $f \cdot \gamma$ sessions per second. If availability were perfect and response time always low, this would also be the revenue-generating throughput (sessions end without defection and produce revenue). However, due to IT imperfections (see Figure A.3), the actual throughput is X transactions per second, with $X < f \cdot \gamma$. Let the average revenue per completed revenue-generating session be ϕ . The lost throughput in transactions per second is ΔX . Thus, one may express the business loss over a time period ΔT as: $L(\Delta T) = \Delta X \cdot \phi \cdot \Delta T$.

Loss has two components: loss due to unavailability and loss due to high response time. Thus, we have: $L(\Delta T) = (\Delta X^A + \Delta X^T) \cdot \phi \cdot \Delta T$ where ΔX^A is the throughput lost

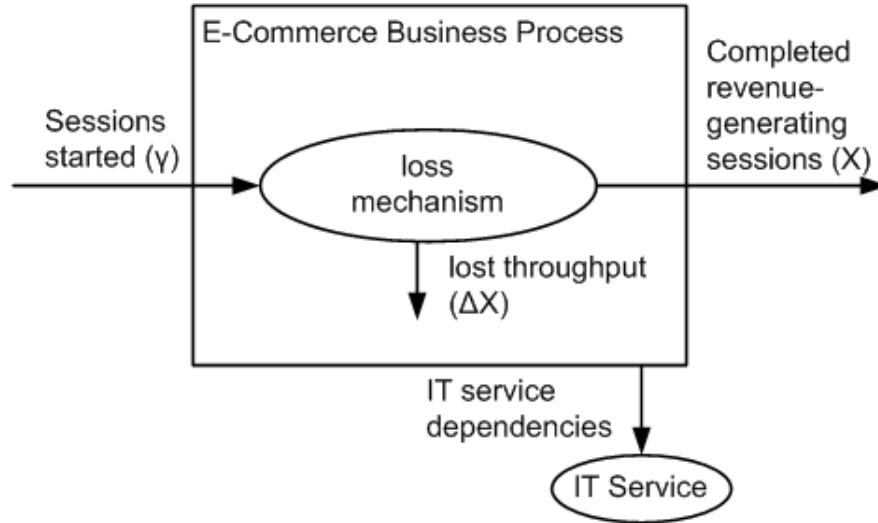


Figure A.3: E-commerce Business Loss

due service unavailability and ΔX^A is the throughput lost due to high response time (customer defections). When the site is unavailable, throughput loss is total and this occurs with probability $1 - A$:

$$\Delta X^A = f \cdot \gamma \cdot (1 - A)$$

On the other hand, when the site is available, loss occurs when response time is slow and this occurs with probability A :

$$\Delta X^T = f \cdot \gamma \cdot B(T^{DEF}) \cdot A$$

The above results are combined to yield:

$$L(\Delta T) = f \cdot \gamma \cdot (1 - A + B(T^{DEF}) \cdot A) \cdot \phi \cdot \Delta T$$

This concludes our analysis.

A.4 An Example E-Commerce Site Design

The purpose of this section is to use the above results and exercise the IT infrastructure design process for a representative e-commerce site.

The site has a revenue-generating CBMG as shown in Figure A.2. For the non-revenue-generating CBMG, we do not provide a figure similar to Figure A.2 but the transition probabilities are shown in Table A.5. In practice, these transition probabilities for a given site

Table A.6: Average number of visits to each state

State (r)	RG Session (V_r^{RG})	NRG Session (V_r^{NRG})
Entry	1.000	1.000
Home	1.579	1.780
Browse	2.325	4.248
Search	3.300	3.510
Login	0.167	0.005
Pay	1.000	0.000
Register	0.083	0.003
Add to cart	1.667	0.069
Select	2.250	1.309
Exit	1.000	1.000

can be gathered from web server log files. Solving these CBMGs as shown in section A.3.3 yields the average number of visits (V_r^{RG} and V_r^{NRG}) shown in Table A.6. Observe that, for RG sessions, the Pay state is always visited (probability 1.0) whereas it is never visited in NRG sessions (probability 0.0). The IT infrastructure consists of three resource classes: web tier, application tier and database tier. For our study, the parameters shown in Table A.7 and Table A.8 are used, except where otherwise noted. The values for all input parameters are meant to be typical for current technology and were obtained from [JST03; MA00; MAD04]. In Table A.7, tuples such as (a,b,c) represent parameter values for the three resource classes (web, application, database); furthermore, each resource is made up of three components: (hardware (hw), operating system (os), application software (as)). Table A.8 shows the average demand (in milliseconds) imposed by a transaction on the various tiers for each customer behavior state; recall that the actual service times are random variables with exponential distribution.

Let us first try to design the site infrastructure in an ad hoc fashion, without business considerations, as is typically performed by an infrastructure designer. This is done by trying to minimize cost while maintaining reasonable service availability and response time. In the discussion that follows, a particular design is represented by the tuple $(n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db})$ which indicates the number of machines (total and load-balanced)

Table A.7: Parameters for example site

Parameters	Values
T^{DEF}	8 seconds
ϕ	\$1 per transaction
γ	14 transactions per second
f	25%
ΔT	1 month
α_j	(1,1,3)
$C_{j,k}^{active}$ (\$/month)	hw =(1100, 1270, 4400) os=(165, 165, 165) as=(61, 35, 660)
$C_{j,k}^{standby}$ (\$/month)	hw =(1000, 1150, 4000) os=(150, 150, 150) as=(55, 30, 600)
$(A_{web}^R, A_{as}^R, A_{db}^R)$	(99.81%, 98.6%, 98.2%) (these values are calculated from appropriate MTBF and MTTR values)

Table A.8: Service demand in milliseconds in all tiers

Tier	CBMG state							
	h	b	s	g	p	r	a	d
Web tier	50	20	30	70	50	30	40	30
Application tier	0	30	40	35	150	70	40	25
Database tier	0	40	50	65	60	150	40	30

in each of the three tiers. The cheapest infrastructure here is $(n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db}) = (1,1,1,1,1,1)$. However, this design cannot handle the applied load (average response time is very high) due to saturation of the servers in all tiers. In order to handle the load and make sure that no server is saturated, the design must use $(n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db}) = (5,5,2,5,5,2)$. There are 5 servers in the web and application tiers and 2 servers in the database tier. This design has a monthly cost of \$24430, average response time of 1.76 s. and service availability of 84.32%. Since this value for availability is typically considered inadequate, the designer may add a single standby server in each tier, yielding a design with infrastructure $(6,6,3,5,5,2)$, monthly cost of \$31715, average response time of 1.76 s. and service availability of 99.38%. If this value of unavailability is still considered inadequate – and one may well ask how the designer is supposed to know what value to aim for – then an additional standby server may be added to each tier, yielding a design with infrastructure $(7,7,4,5,5,2)$, monthly cost of \$39000, average response time of 1.76 s. and service availability of 99.98%. There the designer may rest. We will shortly show that this is not an optimal design.

The problem is that none of the above design decisions take business loss into account. It is instructive to discover the values for loss for the above designs as well as for the optimal design which minimizes the sum of cost plus loss as shown in section A.3.4 (see Table A.9). In that table, each line represents a different infrastructure design alternative; the first column indicates the infrastructure design being considered; the second column is the cost of this infrastructure; the third column represents the business loss (in \$) due to customer defections; the fourth is the business loss due to service unavailability; the fifth is the total financial commitment (cost plus business losses); finally, the last column indicates how much the business loses by adopting that particular infrastructure compared to the optimal one (described in the last line). All financial figures are monthly values.

For the optimal design $(8,9,5,6,6,3)$, the average response time is 0.26 s., availability is 99.98%. It has lowest overall cost+loss, and the table clearly shows the high cost of designing in an ad hoc fashion: a wrong choice can cost millions of dollars per month (last column). Observe that an over-design can also be suboptimal. In this case, business loss can be quite low, but as a result of an over-expensive design.

It is interesting to note that the importance of the site revenue should (and does) affect

Table A.9: Comparing infrastructure designs

Infrastructure design	Cost (\$)	Response Time Loss (\$)	Unavailability Loss (\$)	Cost + Loss (\$)	Cost of choosing wrong (\$)
(5,5,2,5,5,2)	24,430	4,964,417	1,422,755	6,411,602	6,361,129
(6,6,3,5,5,2)	31,715	5,851,498	55,929	5,939,142	5,888,669
(7,7,4,5,5,2)	39,000	5,886,685	1,712	5,927,397	5,876,924
(8,9,5,6,6,3) (optimal)	48,351	754	1,368	50,473	0

infrastructure design. For example, by reducing per transaction revenue from \$1.00 to \$0.10, the optimal design is no longer (8,9,5,6,6,3) but (8,9,3,6,7,2), with monthly cost \$38516, total monthly loss \$2467, average response time 0.26 s. and availability 99.87%; as expected, a site generating less revenue merits less availability (99.87% rather than 99.98%). In other scenarios, response time rather than availability could be the main metric affected. Additional scenarios concerning the importance of per transaction revenue are discussed in [SMM⁺05c].

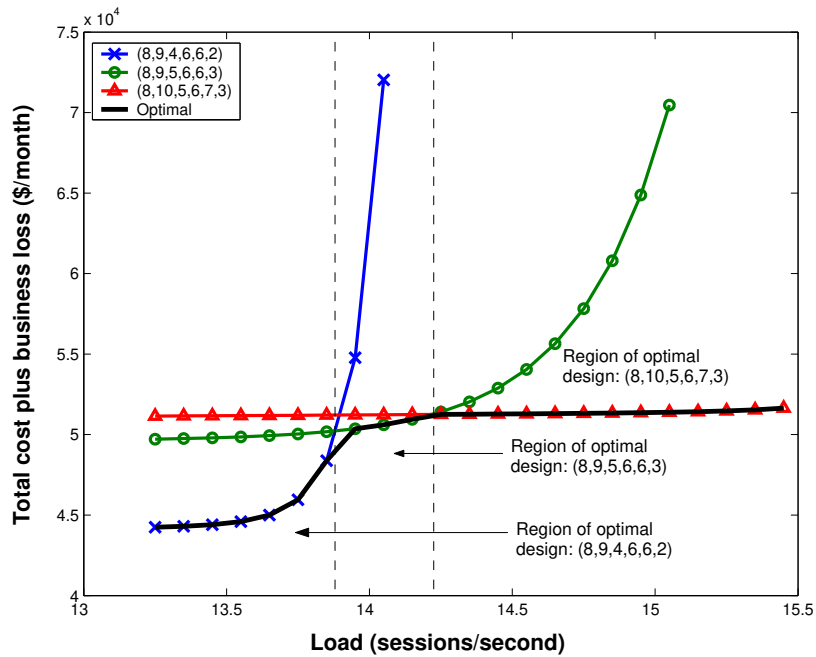


Figure A.4: Sensitivity of total cost plus loss due to load

Finally, we can show how sensitive the optimal design, IT metrics and business metrics are to variations in input load. This is an important consideration since the design procedure assumes a fixed value for input load (γ) while, in practice, this load varies over time. Consider Figure A.4 which shows the total cost plus loss (i.e., $C(\Delta T) + L(\Delta T)$) as load varies. The load values (γ) are divided in three regions: the first design is (8,9,4,6,6,2) and is optimal for all values of load in the left region ($\gamma=13.25$ to 13.85); the middle region ($\gamma=13.85$ to 14.15) has an optimal design of (8,9,5,6,6,3) with an additional database server; the right region ($\gamma=14.25$ to 15.40) has an optimal design of (8,10,5,6,7,3) with an additional application server. Four curves are shown in the figure; the first (blue, cross marker) shows cost plus loss when using the design that is optimal for the left region; similarly, the second curve (green, circle marker) shows cost plus loss when using the design that is optimal for the middle region; the third curve (red, triangle marker) shows the situation for the design that is optimal for the right region. Finally, the heavy black curve simply follows the bottom-most curve in any region and represents the optimal situation in all regions, using three different infrastructure designs, one for each region.

Three major conclusions can be reached from this figure. First, an optimal design remains optimal for a range of load. Although some of these ranges are wider than others, the width of the ranges lends some hope that a static infrastructure design may be optimal or close to optimal even in the presence of some variation in load. The second conclusion is that, in the presence of larger load variations, an infrastructure design can quickly become suboptimal; an example is the leftmost optimal design (8,9,4,6,6,2) which quickly accumulates heavy losses at loads greater than $\gamma=13.85$. In this case, dynamic provisioning can be used to introduce a new infrastructure configuration at appropriate times to reduce business losses (scaling up) or to reduce infrastructure costs (scaling down), as appropriate. The third major conclusion is that it appears that the business impact model described in this paper can be used as one of the mechanisms for dynamic provisioning since it captures appropriate load transition points for reprovisioning using a business perspective. Further investigations will be conducted concerning this point.

Additional interesting details can be seen in Figure A.5 which shows individual components of cost and business loss for the three data center designs described above. Costs clearly go up (from left to right) as designs use more resources, although the increase in cost

is more than offset by the reduction in loss offered by better designs.

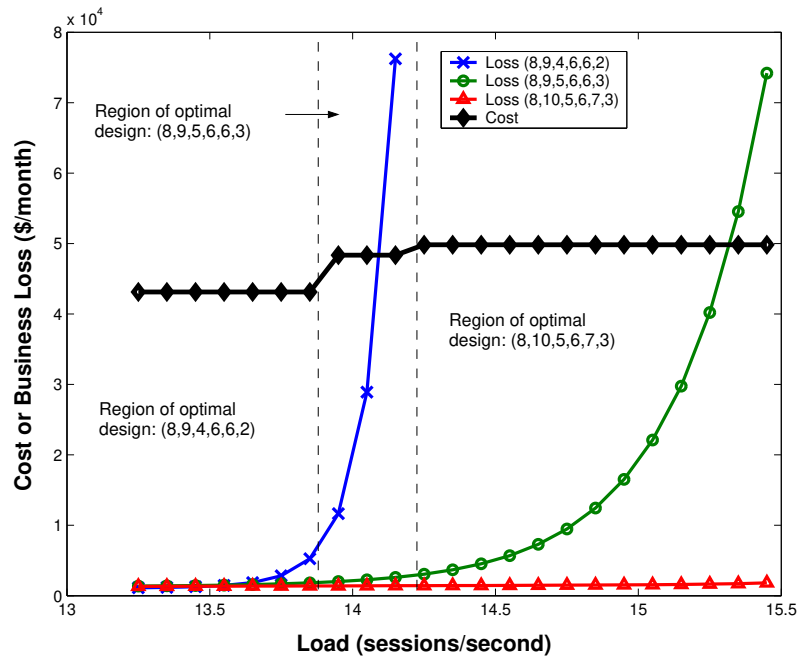


Figure A.5: Sensitivity of cost and loss due to load

Finally, Figure A.6 shows response time for the three designs as well as the optimal response time (heavy black line). Since the load applied to the system varies with time, two situations can occur: in a static provisioning scenario, one can use the data shown in Figure A.6 to choose the “best” design over the expected range of load. This will not be an optimal design for all load values but the designer has a tool to evaluate the cost of over-designing to handle load surges. On the other hand, in an adaptive infrastructure scenario, a dynamic provisioning algorithm can be used to trigger infrastructure changes to keep the design optimal at all load levels. The dashed lines in Figure A.6 show where dynamic provisioning must trigger and the heavy black line shows the response time that is attained, a low value for all load levels.

A.5 Related Work

In the area of infrastructure design, [JST03] describes a tool – AVED – used for capacity planning to meet performance and availability requirements and [AF02] describes a methodology for finding minimum-cost designs given a set of requirements. Simi-

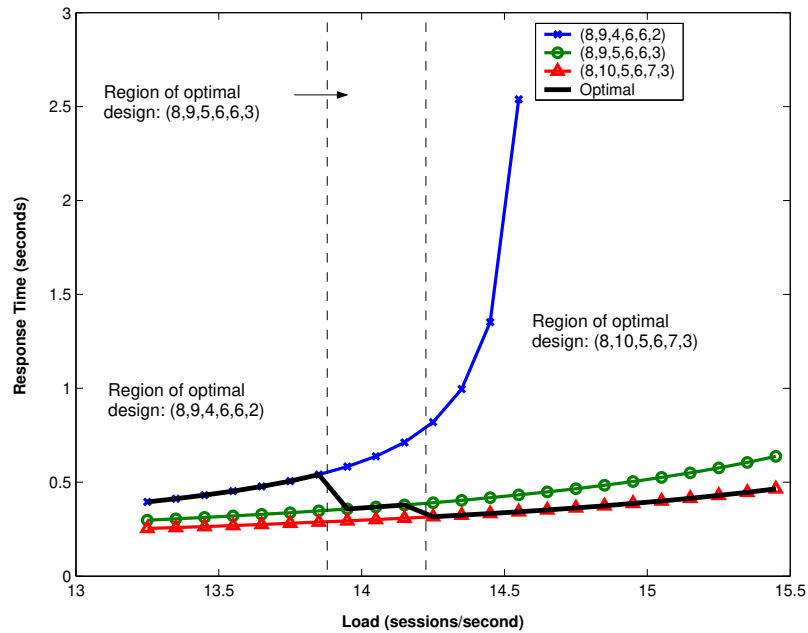


Figure A.6: Response time for various designs

larly, [MBD01] optimizes using IT level metrics. However, none of these references consider the problem of capacity planning from a business perspective, using business metrics. Furthermore, response time considerations are not directly taken into account in [JST03; AF02].

Finally, [GLS04] considers the dynamic optimization of infrastructure parameters (such as traffic priorities) with the view of optimizing high-level business objectives such as revenue. It is similar in spirit to the work reported here, although the details are quite different and so are the problems being solved (the paper considers policies for resource allocation rather than infrastructure design). The model is solved by simulation whereas our work is analytical.

An initial version of the business impact model presented here appeared in [SMM⁺05c]. The current work adds a different customer behavior model (CBMG [MA00]) and a new analysis of customer defection, as well as new conclusions concerning the sensitivity of the optimal design to changes in applied load. Furthermore, dynamic provisioning considerations are discussed here.

A.6 Conclusions

In summary, a method was discussed to design IT infrastructure from a business perspective. The method is novel in that three types of metrics are considered and combined – availability, response time and financial impact – whereas most studies consider only one of the first two in isolation. The three metrics are tied through a business impact model, one of the main contributions of the present work. The method itself finds optimal data center infrastructure configurations by minimizing the total cost of the infrastructure plus the financial losses suffered due to imperfections. It is important to note that a business impact model such as the one discussed here can be used in other contexts to solve other IT management-related problems such as incident management, Service Level Agreement (SLA) design [SMM⁺05c], etc.

We offer the following conclusions:

1. Ad hoc infrastructure design – a process in which business considerations are not formally taken into account – can yield suboptimal designs causing significant business loss.
2. Overdesign to satisfy very stringent SLA requirements can also yield suboptimal designs due to their high cost.
3. The approach taken here can also be used to choose appropriate Service Level Objectives (SLOs) when designing SLAs. Rather than choosing SLO values a priori, the values are simply *calculated* from the optimal design obtained by the process described here.
4. The optimal infrastructure depends on the importance of the business processes being serviced: a business process generating more revenue will merit larger outlays in infrastructure. The method presented here shows how much should be spent to provision services for each business process.
5. Infrastructure designs are optimal over a range of input load; however, we have found that this range is typically small and that response time and resulting business losses can quickly grow when load varies significantly.

6. The method shown provides clear trigger points for dynamic provisioning and also offers a way of calculating what the infrastructure design should be for a given load.

In the future, we plan to develop new impact models applicable to business processes other than e-commerce (say, manufacturing, CRM, etc.). In addition, the approach can be used to investigate more general enterprise architecture scenarios; this will require the development of more holistic models that include the network and other components outside the data center, as well as more detailed enterprise architecture components. Finally, a fuller study of the use of business impact models in adaptive environments can be undertaken; this will be an expansion of the initial comments given here concerning dynamic provisioning.

Acknowledgments

We would like to acknowledge and thank the Bottom Line Project team (<http://www.bottomlineproject.com/>). This work was developed in collaboration with HP Brazil R&D.

Appendix B

Business-Oriented Capacity Planning of IT Infrastructure to handle Load Surges

Jacques Sauvé, Filipe Marques, Antão Moura

Universidade Federal de Campina Grande
{jacques.filipetm, antao}@dsc.ufcg.edu.br

Abstract: This work proposes a business-oriented approach to designing IT infrastructure in an e-commerce context subject to load surges. The main difference between the proposed approach and conventional ones is that it includes the negative business impact – loss – incurred due to IT infrastructure failures and performance degradation. The approach minimizes the sum of infrastructure cost and business losses, rather than only considering infrastructure cost. A complete example scenario shows the value of the method. ¹

¹Aceito para publicação no 2006 IEEE/IFIP Network Operations & Management Symposium (NOMS 2006), em Abril de 2006. / Accepted for publishing in the 2006 IEEE/IFIP Network Operations & Management Symposium (NOMS 2006), in April of 2006.

B.1 Introduction

The goal of this work is to present and formalize a new business-oriented IT infrastructure capacity planning approach that considers load surges. Input load surges are frequent, especially during a time period that precedes special dates, such as the end-of-year buying season or during planned sales promotions. Failing to consider such input variations when designing IT infrastructure may lead to response time requirements that will not be satisfied (during the load surges), leading to business loss caused by customer defections resulting from high response time. Alternatively, it is possible to over-design the infrastructure to meet requirements during the highest expected load surge. This alternative obviously leads to higher-cost infrastructure that will be underutilized under normal load. Few infrastructure design approaches consider expected load surges, reference [LCD⁺03] being an exception.

In the remainder of this paper, section B.2 describes the capacity planning approach from a conventional cost perspective. The capacity planning problem from a business perspective is formalized in section B.3 while section B.4 applies the approach to an example scenario. Finally, section B.5 summarizes our approach, offers conclusions and discusses next steps.

B.2 Capacity Planning Using a Cost Perspective

In this section we formalize the capacity planning problem as it has been conventionally treated using a cost-oriented approach, but including load surges in the model. The analytical model adopted here extends the model in [SMM⁺05c] to handle expected load surges. An *expected surge in load* means a large change in load occurring either during a traditional high sales period – e.g. end-of-year season, Mother’s Day – or during planned sales promotions.

B.2.1 IT Infrastructure Abstraction

To make the model easier to understand, the case of a single IT service S is considered here. Extending the model to multiple services is straightforward.

Service S relies upon the set RC of IT resource classes. Database and web servers are examples of resource classes. A given resource class RC_j is made up of a cluster of n_j identical IT resources. Of this total, m_j resources are in load-balanced mode in order to deal

with incoming load and offer acceptable response time, and $n_j - m_j$ resources are spares running in standby mode to offer better service availability.

Additionally, service S is subject to a set G of load surges: γ^w is the average load applied during the w^{th} load state and its duration is r^w units of time; in the remaining sections, the superscript w will be used to refer to the w^{th} load state; the value $w = 0$ refers to normal load while values of w between 1 and $|G|$ refer to the w^{th} load surge.

Consider Figure B.1: the load applied to service S can be either in a normal state or in a surge state. The load is assumed to switch from the normal state to the surge state whenever there is an expected event such as a sale or a traditional commemorative date such as the end-of-year season that greatly increases the sales rate. Analogously, the load is assumed to change back to normal state as soon as the special occasion ends. Note that, in order to make mathematical treatment easier, we assume that transients that occur when average load changes are negligible; this is a reasonable assumption since the duration of this transient period is probably negligible when compared to the time period (PP) over which the IT infrastructure is planned – typically 1 year.

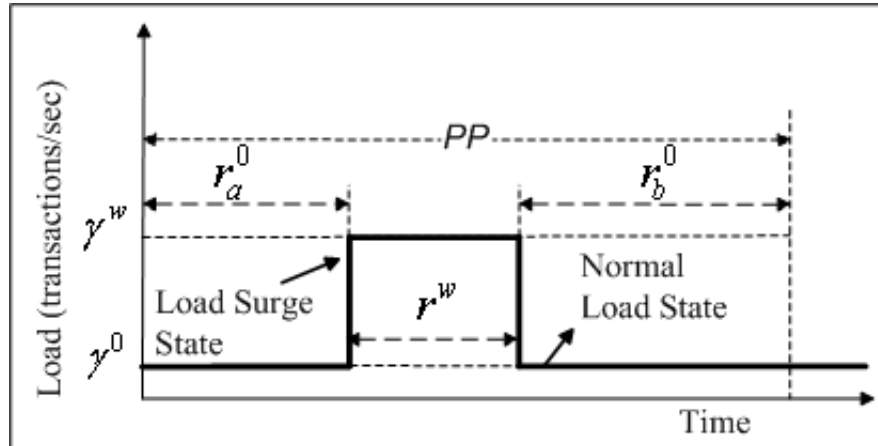


Figure B.1: Load States

Furthermore, a given IT resource $R_j \in RC_j$ is made up of a set $P_j = \{P_{j,1}, \dots, P_{j,k}, \dots\}$ of IT components. Operating system software, server hardware and application server middleware are examples of IT components that can be part of a resource. In the model, if one or more of the IT components fail, the respective IT resource will also fail.

The IT Infrastructure Cost: The IT infrastructure cost, $C(PP)$, over a time period PP , is simply the sum of all IT resources costs. Since IT resources can be configured

either in active, load-balanced mode, or in inactive, standby mode they may have different costs [JST03]. Since IT resources are composed of IT components, one can express their cost rates in terms of IT component cost rates; $c_{j,k}^{active}$ and $c_{j,k}^{standby}$ represent the cost rate of IT component $P_{j,k}$ when it is in active and standby modes respectively (see [SMM05a] for further details).

IT Service Availability: IT components are subject to failures that produce unavailability of the services relying upon them. Standard reliability theory is used to calculate the availability of individual IT resources, A_j^R . From this, the availability of IT resource classes, A_j , can be calculated and, finally, the availability A of service S itself. For further details, please refer to [SMM05a].

IT Response Time: With the purpose of estimating the response time distribution, $ResponseTimeDistribution^w(x)$, and also its mean, \bar{T}^w , over load state w , the load applied to IT resources must be modeled. The model is based on an open queueing model [Kle76a] suitable for contexts such as e-commerce where many potential customers may exist. In the present work we assume Poisson arrivals – arrivals are indeed well approximated by a Poisson process [MAR⁺03a] at small time scales which is our concern here – and exponentially distributed service times. The model considers expected input load surges and will be used to design IT infrastructure.

In order to find the response time distribution, one needs to calculate the cumulative distribution of response time, $T^w(y)$, during load state w . Since a transaction can possibly use resources from all resource classes, the response time for a transaction in load state w is the sum of $|RC|$ random variables, which can be evaluated through the product of their Laplace Transforms. Finally, the average response time can be found from the distribution. Space prevents us from providing full mathematical development; please see [SMM05a].

B.2.2 Cost-Oriented Capacity Planning Formalization

Informally, the problem is that of choosing the cheapest infrastructure configuration that satisfies certain requirements. Formally, we have (Table B.1):

Table B.1: Cost-Oriented Capacity Planning Problem Formalization

Find	n_j and m_j for each resource class RC_j
By Minimizing:	$C(PP)$, the infra cost over time period PP
Subject to:	$A \geq A_{MIN}, \bar{T}^w \leq T_{MAX}^w, n_j \geq m_j, m_j \geq 1$
Where	A_{MIN} is the minimum acceptable value for availability for service S ; T_{MAX}^w is the maximum acceptable value for the average response time for service S , and w indicates the load state

B.3 Adding a Business-Oriented View

In this section, the problem of designing IT infrastructure in order to handle expected load surges is reexamined from a business view. A new approach called Business-Driven IT Management (BDIM) [SMM⁺05c; SB04; MBC04] allows one to estimate the business losses incurred both by IT infrastructure imperfections as well as those caused by load surges.

To capture business losses, an impact model must be developed. This model – fully described in [SMM05a] – is used to create a cause-effect relationship between IT infrastructure events and adverse business impact. This is not a trivial task and business processes (BPs) play a crucial role: the impact model maps metrics such as service availability and response time – technical service metrics – into BP metrics such as business transaction throughput. Furthermore, a revenue model links BP throughput to business metrics such as revenue throughput and lost revenue throughput.

B.3.1 Handling Load Surges: The Challenge

Informally, the problem is that of choosing the ideal IT configuration that minimizes the sum of infrastructure cost plus business losses when expected load surges are considered. Formally, one may pose the problem as shown in Table B.2.

Now we argue how to evaluate business loss, $L(PP)$.

Table B.2: Business-Oriented Capacity Planning Problem Formalization

Find	n_j and m_j for each resource class RC_j
By Minimizing:	$C(PP) + L(PP)$, the total financial impact on the business over time period PP
Subject to:	$n_j \geq m_j$ and $m_j \geq 1$
Where	$C(PP)$ is the cost of the IT infrastructure over time period PP ; $L(PP)$ represents the business loss incurred either by IT infrastructure imperfections or load surges over time period PP

B.3.2 Loss Model Components

In the proposed model, business loss comes from two sources: service (un)availability, A , and customer defections. We assume that customers will defect, that is, not conclude their purchases, whenever response time is higher than a certain threshold; the value of 8 seconds is mentioned in the literature as an appropriate threshold value [MAD04]. Therefore, in order to estimate business loss, one must estimate the probability that response time is higher than this threshold during load state w , that is the customer defection probability, $B^w(T^{DEF})$. From the response time distribution, we have: $B^w(T^{DEF}) = 1 - ResponseTimeDistribution^w(T^{DEF})$.

B.4 Evaluating the Model: an Example

In this section we evaluate the above model through a complete example scenario for which an IT infrastructure must be designed in order to handle expected load surges.

The scenario chosen is an e-commerce site; this application satisfies the two main requirements for the proposed model to be applicable: there is a heavy IT dependency and the revenue model is typical of an e-commerce order-taking business process. Service S represents the e-commerce web site and relies upon three resource classes in a three-tier architecture: a web server resource class (RC_{web}), an application server resource class (RC_{as}) and a database resource class (RC_{db}). Each resource class is composed of three components:

Table B.3: Example scenario parameters

Parameters	Values
T^{DEF}	8 seconds
φ	\$ 3.00 per transaction
PP	1 year
r^0	10 months ($r^0 = PP - r^1 - r^2$)
r^1, r^2	1 month each (May and December)
γ_0	14.8 transactions per second
γ_1, γ_2	44.4 transactions/s ($\gamma^1 = \gamma^2 = \gamma^0 \cdot 3$)
α_j	(1,1.5,6)
$C_{j,k}^{active}$ (\$/month)	hw=(1100,1270,4400), os=(165,165,165), as=(61,35,660)
$C_{j,k}^{standby}$ (\$/month)	hw=(1000,1150,4000), os=(150,150,150), as=(55,30,600)
$(A_{web}^R, A_{as}^R, A_{db}^R)$	(99.81%, 98.6%, 98.2%) (these values are calculated from appropriate MTBF and MTTR values)
D_j	(0.040, 0.115, 0.130)
T_{MAX}^w	1.5 seconds
A_{MIN}	99.96%

hardware (hw), operating system (os) and either web server software (ws) in the web tier, application server software (as) in the application tier or DBMS (db) in the data tier.

The input parameters of the model receive typical values for current technology [SMM⁺05c; JST03; MAD04] as shown in Table B.3. In this table, tuples (a,b,c) represent the values for all resource classes ($RC_{web}, RC_{as}, RC_{db}$) in that order.

The first evaluation compares the two approaches to infrastructure design – cost-oriented and business-oriented. For the cost-oriented approach, the values for A_{MIN} – the minimum acceptable value for availability for service S – and T_{MAX}^w – the maximum acceptable value for the average response time for service S – were set at 99.96% and 1.5 seconds, respectively, typical values for service design.

Figure B.2 shows a comparison between the optimal designs of both approaches. In this figure two load surges were considered: one in May (Mother's Day) and the other in December (end-of-year season). Each demand surge is assumed to last 1 month and to demand n times the normal load value (X -axis). By considering only costs, one can hastily conclude that the conventional method yields better results than the one proposed here since it offers infrastructure with lower cost – the average yearly cost using the conventional approach is \$559,200, while it is \$640,100 using the BDIM method.

However, this hasty conclusion is erroneous and causes unnecessarily high financial outlays. The business losses incurred by IT failures and high response time have not been considered. By looking at Figure B.2 carefully, the Total Cost metric – infrastructure cost plus business losses – of the optimal designs found by the business-oriented approach is, on average, approximately 40% lower than the average Total Cost from the designs obtained by the conventional-approach. For large surges, the difference in total cost is over 100%.

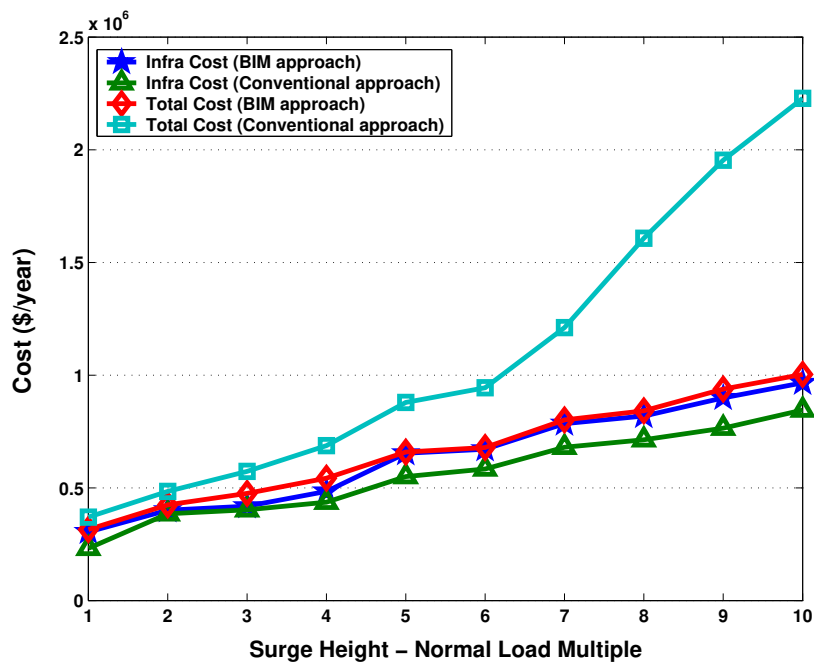


Figure B.2: Cost and BDIM Dimensions Comparison

Next, a sensitivity analysis investigates how surge duration and height influence optimal design and business metrics. Figure B.3 depicts how optimal design is influenced by variations in surge duration. In this scenario, there are two load surges whose total duration varies from 1 to 3 months (X -axis). A change in optimal design only occurs when the surges last 2

months: the optimal design changes from $(n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db}) = (4,6,4,2,4,2)$ to $(4,7,4,2,4,2)$. One may conclude that the optimal design is relatively insensitive to changes in surge duration, since business losses increase only linearly with surge duration. Note that even though this change causes an infrastructure cost increment (top curve) it leads to a greater loss reduction (bottom curve). This fact can be grasped when comparing the values shown by the middle curve – that shows how business loss grows when the original $(4,6,4,2,4,2)$ design is kept throughout – with the values of the bottom curve – the loss for the optimal design. The difference between both approaches is \$120,720/year, when the total duration of load surges is 2.5 months, and \$127,821/year, when the total duration of load surges is 3 months.

Another important conclusion about optimal design sensitivity is exemplified by Figure B.4 which shows how sensitive the optimal design is to variations in surge height. As can be seen, optimal design is profoundly affected by load surge height since surges saturate resources. This fact leads to an appreciable difference between the optimal designs (bottom curve) and the non-optimal one (top curve).

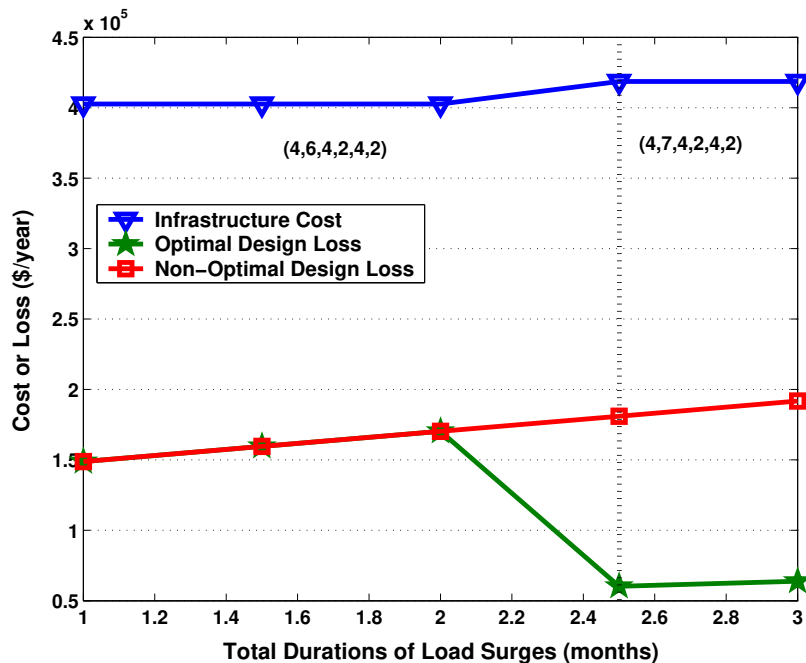


Figure B.3: Influence of Surge Duration on Optimal Design

The last aspect examined concerns the behavior of business losses over a period of a year considering three IT infrastructure design alternatives (Figure B.5). The first one (square

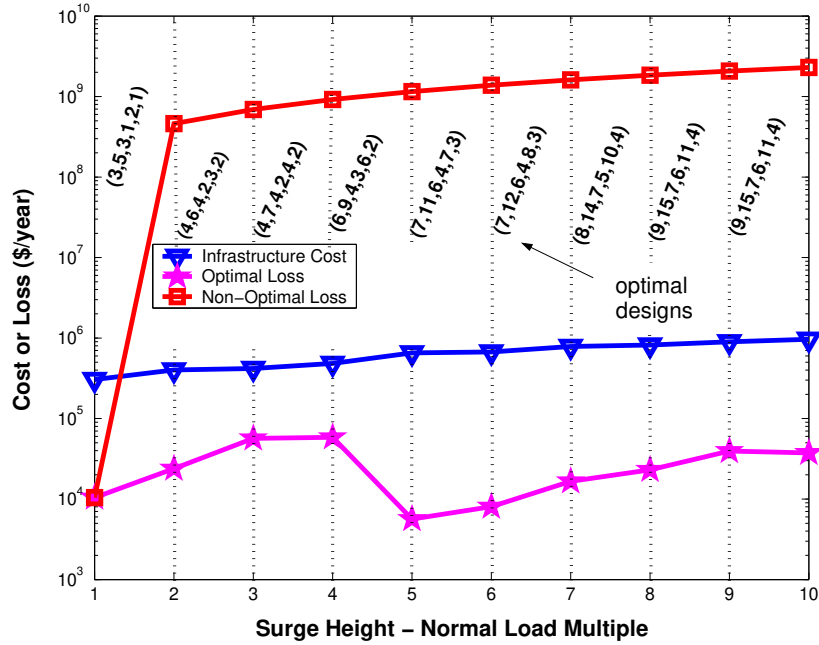


Figure B.4: Influence of Surge Height on Optimal Design

markers) is the optimal cost-oriented design considering only normal load; this design is $(n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db}) = (3,5,3,1,2,1)$. The second one (circle markers) is the over-design alternative $(6,8,4,3,5,2)$, and is obtained by optimally designing the infrastructure through the conventional approach, considering only the highest load surge. The last design alternative (cross markers) refers to the optimal business-oriented design $(4,7,4,2,4,2)$. The square marker curve assumes extremely high values during surges and has been truncated vertically. Furthermore, even though the business-oriented and over-design approaches are apparently close to one another, savings total nearly \$54,000 (one year).

B.5 Conclusions

In this paper we have formalized and proposed a business-oriented method to design IT infrastructure in order to properly handle expected load surges. We have shown that the resulting infrastructure design is superior to the one obtained from the strictly cost-oriented method. The value of the method was demonstrated by means of a complete numerical example scenario. For a discussion of related work, please refer to [SMM05a].

Three major conclusions ensue from the results presented in section B.4. The first one

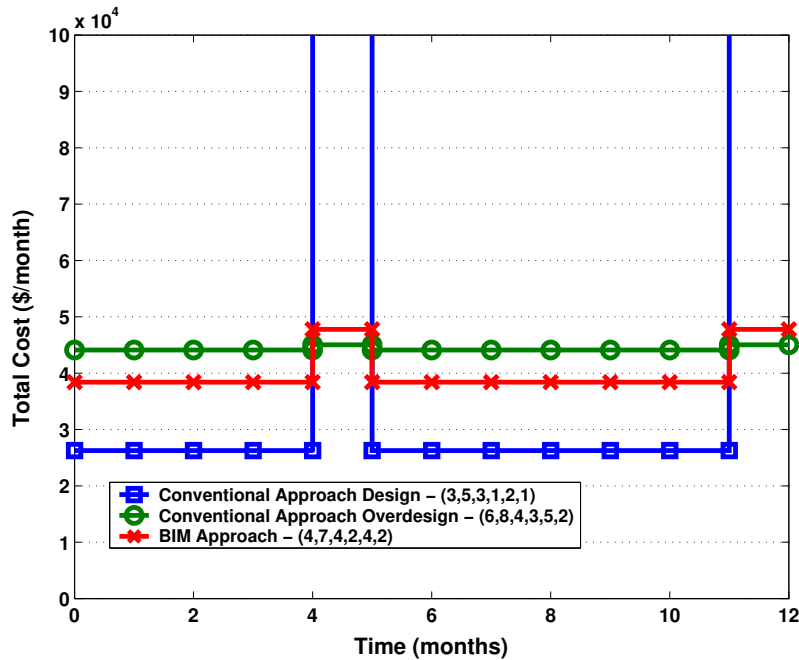


Figure B.5: Comparison of Three Design Alternatives

is that the proposed approach is superior to the conventional one since it considers the business losses incurred by IT failures and performance degradations and leads to more cost-effective designs. The second conclusion is that the optimal design is relatively insensitive to changes in surge duration, since business losses increase only linearly with surge duration. The last conclusion concerns the fact that optimal designs are profoundly affected by load surge heights since they easily saturate resources in a highly non-linear fashion typical of queuing systems.

As future work we intend to concentrate on dynamic resource allocation from a business perspective, thus allowing the present model to deal with unexpected load surges (rather than expected ones). That is, we want to investigate how to use business impact metrics as a reoptimization trigger in an adaptive computing context. Other loss models involving gradual loss can also be investigated. Finally, we need to provide tools to help IT designers benefit from the approach.

We would like to thank the Bottom Line Project team. This work was developed in collaboration with HP Brazil R&D.

Appendix C

SLA Design from a Business Perspective

Jacques Sauv ¹, Filipe Marques¹, Ant o Moura¹, Marcus Sampaio¹, Jo o Jornada² and Eduardo Radziuk²

¹*Universidade Federal de Campina Grande*

{jacques,filipetm,antao,sampaio}@dsc.ufcg.edu.br

²*Hewlett-Packard-Brazil*

{joao.jornada,eduardo.radziuk}@hp.com

Abstract: A method is proposed whereby values for Service Level Objectives (SLOs) of an SLA can be chosen to reduce the sum IT infrastructure cost plus business financial loss. Business considerations are brought into the model by including the business losses sustained when IT components fail or performance is degraded. To this end, an impact model is fully developed in the paper. A numerical example consisting of an e-commerce business process using an IT service dependent on three infrastructure tiers (web tier, application tier, database tier) is used to show that the resulting choice of SLOs can be vastly superior to ad hoc design. A further conclusion is that infrastructure design and the resulting SLOs can be quite dependent on the “importance” of the business processes (BPs) being serviced: higher-revenue BPs deserve better infrastructure and the method presented shows exactly how much better the infrastructure should be. ¹

¹Publicado no 16th IFIP/IEEE Distributed Systems: Operations and Management Workshop (DSOM 2005), em Outubro de 2005. / Accepted for publishing in the 16th IFIP/IEEE Distributed Systems: Operations and Management Workshop (DSOM 2005), in October of 2005.

C.1 Introduction

Service Level Agreements (SLAs) are now commonly used to capture the performance requirements that business considerations make on information technology (IT) services. This is done both for services provided in-house and for outsourced services. An SLA defines certain Service Level Indicators (SLIs) and restrictions that such indicators should obey. Restrictions are frequently expressed in the form of Service Level Objectives (SLOs), threshold values that limit the value of SLIs. Some typical SLIs are service availability, service response time, and transaction throughput. The problem examined in this paper is that of designing SLAs; the SLA design problem is informally defined as that of choosing appropriate values for SLOs. For example, should service availability be 99.9%, 99.97%? How is one to choose adequate values? There are other aspects to SLA design (choosing SLIs, choosing measurement methods and periods, choosing penalties, etc.) but these are not considered here.

It is interesting to examine how choosing SLOs is typically done today. Naturally, since SLOs are chosen according to how important a service is to the business, the IT client (a senior business manager) is involved in choosing SLOs. However, as reference [TT05b] has vigorously shown, the methods used are almost always pure guesswork, frequently resulting in drastic loss or penalties. It is clear that one needs more mature and objective models to properly design SLAs. An approach based on Business Impact Management [Mas02; SB04] is presented in this paper.

The remainder of the paper is organized as follows: section C.2 informally discusses the approach while section C.3 formalizes it; section C.4 considers an application of the method through a full numerical example; section C.5 discusses related work; conclusions are provided in section C.6.

C.2 Gaining a Business Perspective on IT Operations

An informal discussion of the approach adopted here will help the reader follow the formal treatment presented in the next section.

C.2.1 Addressing IT Problems through Business Impact Management

SLOs must be chosen by taking into account the importance of the IT service on the business. In the approach being described here, this is done by capturing the impact of IT faults and performance degradations on numerical business metrics associated with the business. By considering business metrics, one may say that the approach is part of a new area of IT management called Business Impact Management (BIM) [Mas02; SB04]. BIM takes Service Management (SM) to a new maturity level since metrics meaningful to the customer such as financial or risk measures are used to gauge IT effectiveness rather than technical metrics such as availability and response time.

For BIM to be successfully applied to the problem at hand, one needs to construct an impact model. Since it is quite difficult to bridge the gap between events – such as outages – occurring in the IT infrastructure and their financial effect on the business, an intermediate level is considered: that of the business processes (BPs) using the IT services. Thus, an impact model is used to map technical service metrics to BP metrics such as BP throughput (in transactions per second) and a revenue model to map BP throughput to a final business metric such as revenue throughput.

Thus, this paper essentially investigates how BIM can be useful in addressing some common IT problems. SLA design was chosen as an example of an activity performed by IT personnel that can be rethought from a business perspective using BIM.

C.2.2 SLA Design: An Optimization Problem

The IT infrastructure used to provision IT services is designed to provide particular service levels and these are captured in SLAs. Intuitively, a weak infrastructure (with little redundancy or over-utilized resources) has the advantages of having low cost but may generate high business losses – as captured by the BIM impact model – resulting from low availability and customer defections due to high response times. An infrastructure with much better availability and lower response times will possibly generate lower business losses but may have a much higher total cost of ownership (TCO). Thus, in both cases, total financial outlay (TCO plus business losses) may be high. It thus appears that a middle ground can be found that will minimize this sum. Once this infrastructure yielding minimal financial outlay is

found, one may then calculate SLOs such as availability and response time. As a result, SLO thresholds will be *outputs* from the method rather than being chosen in an ad hoc way. These SLOs will be optimal in the sense that they will minimize total financial outlay.

C.3 Problem Formalization

The optimization problem considered aims to calculate the number of load-balanced resources and the number of fail-over resources to be used in provisioning IT services so as to minimize overall cost (TCO plus business losses). The model considers workloads with fixed averages and static resource allocation. Once this infrastructure is found, SLOs such as service availability, average response time, etc. can be calculated and inserted in the SLA. This section formalizes the SLA Design problem.

C.3.1 The Entities and their Relationships

Figure C.1 shows the entities and their relationships used in the problem formalization. It can also be useful to the reader as a quick reference to the notation employed. The model includes entities both from the IT world and the business world. The business (top) layer consists of several business processes. For simplicity, assume that there is a one-to-one relationship between business processes and IT services. Extension to several services is straightforward but would needlessly complicate the formalism for this presentation. We thus have a set BP of BPs and a set S of services: $S = \{s_1, \dots, s_{|S|}\}$. The infrastructure used to provision these services consists of a set RC of resource classes.

Service s_i depends upon a set RC_i^S of these resource classes. For example, a service could depend on three resource classes: a Web resource class, an application server resource class and a database resource class. Class RC_j consists of a cluster of IT resources. This cluster has a total of n_j identical individual resources, up to m_j of which are load-balanced and are used to provide adequate processing power to handle incoming load. The resources that are not used in a load-balanced cluster are available in standby (fail-over) mode to improve availability.

Finally, an individual resource $R_j \in RC_j$ consists of a set $P = \{P_{j,1}, \dots, P_{j,k}, \dots\}$ of components, all of which must be operational for the resource to also be operational. As

an example, a single Web server could be made up of the following components: server hardware, operating system software and Web server software. Individual components are subject to faults as will be described later.

An SLA is to be negotiated concerning these services. For service s_i , the SLA may specify Service Level Objectives (SLOs). The impact model to be presented assumes that BP throughput is lost if the service is unavailable or if response time exceeds a certain threshold. The following SLO parameters are considered for service s_i and will constitute the promise made to the customer in the SLA: A_i^{MIN} , the minimum service availability, \bar{T}_i , the average response time, T_i^{DEF} , the response time threshold causing customer defection and $B_i(T_i^{DEF}) = B_i^{MAX}$, the probability that response time is larger than the threshold.

One may thus summarize the SLA as the four sets: $A^{MIN} = \{\dots, A_i^{MIN}, \dots\}$, $T = \{\dots, \bar{T}_i, \dots\}$, $T^{DEF} = \{\dots, T_i^{DEF}, \dots\}$, $B^{MAX} = \{\dots, B_i^{MAX}, \dots\}$.

C.3.2 The Cost Model

Each infrastructure component $P_{j,k}$ has a cost rate $c_{j,k}^{Active}$ when active (that is, used in a load-balanced server) and has a cost rate $c_{j,k}^{Standby}$ when on standby. These values are cost per unit time for the component and may be calculated as its total cost of ownership (TCO) divided by the amortization period for the component. The cost of the infrastructure over a time period of duration ΔT can be calculated as the sum of individual cost for all components. In the equation below, j runs over resource classes, l runs over resources and k runs over components.

$$C(\Delta T) = \Delta T \cdot \sum_{j=1}^{|RC|} \left(\sum_{l=1}^{m_j} \sum_{k=1}^{|P_j|} c_{j,k}^{Active} + \sum_{l=1}^{n_j-m_j} \sum_{k=1}^{|P_j|} c_{j,k}^{Standby} \right) \quad (C.1)$$

C.3.3 Loss Considerations

A weak infrastructure costs little but may generate large financial losses due to low availability or high response time. The converse situation is an infrastructure that causes little loss but is expensive to provision. In order to evaluate this tradeoff, financial loss must be calculated. In general, the model used is that at time t , the imperfect infrastructure produces adverse impact on business – or simply business loss – at rate $l(t)$; the rate is expressed in

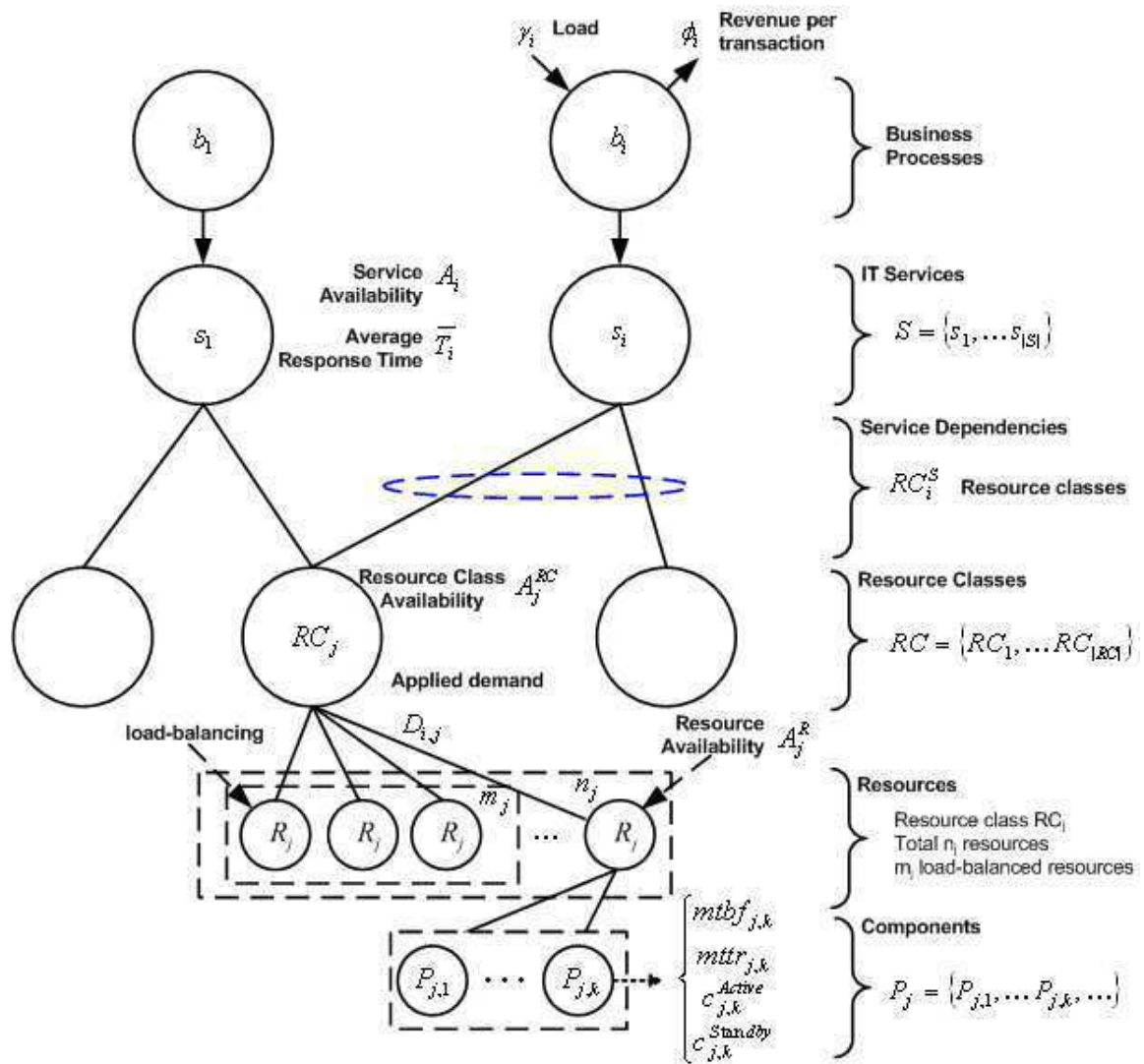


Figure C.1: Entities and their relationships

Find:	The SLA parameters, the sets A^{MIN} , T , B^{MAX}
By minimizing:	$C(\Delta T) + L(\Delta T)$, the total financial impact on the business over evaluation period ΔT
Over:	$\{n_1, \dots, n_{ RC }\}$ and $\{m_1, \dots, m_{ RC }\}$
Subject to:	$n_j \geq m_j$ and $m_j \geq 1$
Where:	$C(\Delta T)$ is the infrastructure cost over the SLA evaluation period ΔT ; $L(\Delta T)$ is the financial loss over the SLA evaluation period ΔT ; n_j is the number of resources in resource class RC_j ; m_j is the number of load-balanced resources in RC_j .

units appropriate to the business metric used per time unit. As an example, loss rate could be expressed in dollars per second when using dollar revenue as a business metric.

For simplicity, assume that all SLOs are evaluated at the same time and that the evaluation period is ΔT . Thus, the accumulated business impact over the evaluation period is $L(\Delta T) = \int_0^{\Delta T} l(t) dt$. Assuming a constant rate (l) of faults over time, we have $L(\Delta T) = \Delta T \cdot l$. A specific loss model will be discussed below.

C.3.4 The SLA Design Problem

The SLA Design problem may be stated informally as follows: one wishes to determine the number of servers – both total number of servers and number of load-balanced servers – that will minimize the financial impact on the enterprise coming from two sources: infrastructure cost and financial loss. Formally, a first SLA Design problem may be posed as follows:

The set $T^{DEF} = \{\dots, T_i^{DEF}, \dots\}$ which indicates the response time threshold from which defections start to occur is given as input. A typical value is 8 seconds for web-based e-commerce [MAD04]. As a result of the optimization, values for the three sets of SLA thresholds availability: $A^{MIN} = \{\dots, A_i^{MIN}, \dots\}$, average response time: $T = \{\dots, \bar{T}_i, \dots\}$, and defection probability: $B^{MAX} = \{\dots, B_i^{MAX}, \dots\}$ will be found. These are the values to be used in an SLA.

In order to complete the model, one needs to define an impact model and a way to calculate loss $L(\Delta T)$, and the SLOs A^{MIN} , T , and B^{MAX} . The next sections cover this.

C.3.5 A Specific Loss Model

When IT problems occur, the impact on business may be decreased revenue or increased costs or both. In this paper only decreased revenue is considered, a situation applicable to revenue-generating BPs typical in e-commerce. Each BP has an input load (in transactions per second). Some of this load is lost due to a loss mechanism with 2 causes: service unavailability and customer defection due to high response times. Subtracting lost load from the input load results in the BP transaction throughput (denoted by X). The revenue throughput due to any given business process is $V = X \cdot \phi$ where ϕ is the average revenue per transaction for the business process. The total loss rate, over all BPs is

$$l = \sum_{i=1}^{|BP|} l_i$$

where BP is the set of BPs and l_i is the loss rate due to BP b_i . In the above, we have $l_i = \Delta X_i \cdot \phi_i$. Here, ΔX_i is the loss in throughput (in transactions per second) for BP b_i and ϕ_i is the average revenue per transaction for process b_i .

We consider that the BP is heavily dependent on IT, and thus BP availability A_i is equivalent to the availability of the IT service (s_i) used by the BP. When service s_i is unavailable, throughput loss is total and this occurs with probability $1 - A_i$. We thus have $\Delta X_i^A = \gamma_i \cdot (1 - A_i)$ where ΔX_i^A is loss attributable to service unavailability, γ_i is the input load incident on BP b_i and A_i is the availability of service s_i . When service is available (this occurs with probability A_i), loss occurs when response time is slow. Thus, we have $\Delta X_i^T = \gamma_i \cdot B_i(T_i^{DEF}) \cdot A_i$ where ΔX_i^T is loss attributable to high response time, $B_i(T_i^{DEF}) = Pr[\tilde{T}_i > T_i^{DEF}]$ is the probability that the service response time (the random variable \tilde{T}_i) is larger than some threshold T_i^{DEF} . This models customer defection and assumes that a customer will always defect if response time is greater than the threshold (typically 8 seconds for an e-commerce BP).

The total loss in BP throughput is simply the sum of losses due to unavailability and losses due to high response time:

$$\Delta X_i = \Delta X_i^A + \Delta X_i^T = \gamma_i \cdot (1 - A_i) + \gamma_i \cdot B_i(T_i^{DEF}) \cdot A_i \quad (\text{C.2})$$

C.3.6 The Availability Model

In order to calculate lost throughput, one needs to evaluate the availability A_i of an IT service, s_i . This is done using standard reliability theory [Tri82]. Individual component availability may be found from Mean-Time-Between-Failures (MTBF) and Mean-Time-To-Repair (MTTR) values. Since all components must be available for a resource to be available, the component availabilities are combined using “series system reliability” to yield resource availability A_j^R . Combining resource availability to compute resource class availability (A_j^{RC}) uses “m-out-of-n reliability” since the resource class will be available and able to handle the projected load when at least m_j resources are available for load-balancing. Finally, for service s_i to be available, all resource classes it uses must be available and “series system reliability” is used to calculate service availability (A_i).

C.3.7 The Response Time Performance Model

The loss calculation depends on $B_i(T_i^{DEF})$, the probability that the service response time is larger than some threshold T_i^{DEF} . In order to find this probability, the IT services are modeled using an open queuing model. This is adequate for the case of a large number of potential customers, a common situation for e-commerce. Each resource class RC_j consists of a cluster of n_j resources, of which m_j are load-balanced. Let us examine service s_i . The input rate is γ_i transactions per second. Each transaction demands service from all resource classes in the set RC_i^S . Demand applied by each transaction from BP b_i on class RC_j is assumed to be $D_{i,j}$ seconds. In fact this is the service demand if a “standard” processing resource is used in the class RC_j resources. In order to handle the case of more powerful hardware, assume that a resource in class RC_j has a processing speedup of α_j compared to the standard resource. Thus, service time for a transaction is $D_{i,j}/\alpha_j$ and the service rate at a class RC_j resource for transactions from business process b_i is $\mu_{i,j} = \alpha_j/D_{i,j}$. Finally, since there are m_j identical load-balanced parallel servers used for processing in resource class RC_j , response time is calculated for an equivalent single server [MAD04] with input load $\lambda_{i,j} = \gamma_i/m_j$. Thus the utilization $\rho_{i,j}$ of class RC_j resources in processing transactions from business process b_i is:

$$\rho_{i,j} = \frac{\lambda_{i,j}}{\mu_{i,j}} = \frac{\gamma_i \cdot D_{i,j}}{m_j \cdot \alpha_j} \quad (\text{C.3})$$

The total utilization ρ_j of class RC_j resources due to transactions from all services is:

$$\rho_j = \sum_{i=1}^{|S|} \rho_{i,j} \quad (\text{C.4})$$

Observe that, when load is so large that any $\rho_j \geq 1$, then any service depending on that resource class will have $B_i(T_i^{DEF}) = 1$, since response time is very high for saturated resources.

Now, in order to find $B_i(T_i^{DEF})$ when $\rho_j < 1$, let us find the cumulative distribution of response time, $T_i(y) = Pr[\tilde{T}_i \leq y]$. In this case, the total response time for a transaction from BP b_i is the sum of $|RC_i^S|$ random variables, one for each resource class used by service s_i . In order to find the probability distribution of a sum of independent random variables, one may multiply their Laplace transforms [Kle76a]. In order to make mathematical treatment feasible, assume Poisson arrivals (this is a reasonable assumption for stochastic processes with large population) and exponentially distributed service times. (Observe that although service times may not be independent and exponentially distributed in practice, the optimization step *compares* design alternatives and that is probably insensitive to particular distributions – if they are the same when comparing results.) From queuing theory, the Laplace transform of response time (waiting time plus service time) for a single-server queue is $T^*(s) = a/(s + a)$ where $a = \mu \cdot (1 - \rho)$, μ is the service rate and ρ is the utilization. Recall that input load from several services is going to the same resource class. Thus, for the combination of resource classes used by service s_i , we have:

$$T^*(s) = \prod_{j \in RC_i^S} \frac{a_{i,j}}{s + a_{i,j}} \quad (\text{C.5})$$

where $a_{i,j} = \mu_{i,j} \cdot (1 - \rho_j)$. Inverting the transform yields the probability density function of response time, which is integrated to find the cumulative probability distribution function (PDF) of response time, $T_i(y)$. Finally:

$$B_i(T_i^{DEF}) = Pr[\tilde{T}_i > T_i^{DEF}] = 1 - T_i(T_i^{DEF}) \quad (\text{C.6})$$

Table C.1: Parameters for example

Parameters	Values	Parameters	Values
T^{DEF}	8 seconds	α_j	(1,1,3)
ϕ	\$1 per transaction	$c_{j,k}^{Active}$ (\$/month)	hw =(1100, 1100, 4400) os=(165, 165, 165) as=(61, 30, 660)
γ	14 transactions per second	$c_{j,k}^{Standby}$ (\$/month)	hw =(1000, 1000, 4000) os=(150, 150, 150) as=(55, 0, 600)
ΔT	1 month	D_j	(0.05, 0.1, 0.2) seconds
A_j^R (resource availability for R_j)	99.81% (this value is calculated from appropriate MTBF and MTTR values)		

Additionally, average response time is typically defined in an SLA and may be found from the Laplace transform as follows:

$$\bar{T}_i = - \left. \frac{dT_i^*(s)}{ds} \right|_{s=0} \quad (C.7)$$

C.4 A Numerical Example of SLA Design

The purpose of this section is to go through a complete example and verify the extent to which the method proposed can be useful in designing SLAs, i.e., choosing SLO values. Assume the existence of a single service (the index i is dropped) using three resource classes: a Web resource class (RC_{web}), an application server resource class (RC_{as}) and a database resource class (RC_{db}). In the example, the parameters shown in Table C.1 are used, typical for current technology [JST03]. In that table, tuples such as (a,b,c) represent parameter values for the three resource classes (web, application, database); furthermore, each resource is made up of three components: hardware (hw), operating system (os) and application software (as).

Let us now first get a feeling for the variation of some of these measures. Figure C.2 shows how the loss component due to response time (ΔX_i^T) indeed varies as response time rises with increased load. Similarly, one can get a feel for the loss component due to availability (ΔX_i^A) from Figure C.3. In that figure, availability is made to improve by changing the number of database machines from 2 to 6, while keeping other infrastructure components constant. The loss due to high response time is very low and is thus not shown in the figure. As one can see, cost increases, loss due to unavailability decreases while “cost + loss” reaches a minimum value for 4 machines.

It is now time to consider the main problem of interest in this paper: that of SLA design. If one were to design the SLA in an ad hoc way, one could approach the problem from the infrastructure side and try to minimize cost while maintaining reasonable service availability and response time. The cheapest infrastructure here is $(n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db})=(1,1,1,1,1,1)$. However, this design cannot handle the applied load (average response time is very high) due to saturation of the application server. A second try yields $(1,2,1,1,2,1)$ – more power in the application tier. This yields a monthly cost of \$9141, and SLOs of (average response time=1.5 s, service availability=95.32%). Since this availability is not typically considered adequate, the designer may increase the number of machines in other tiers yielding a design with infrastructure $(3,3,3,1,2,1)$, cost \$22201 and SLOs of (1.5 s, 99.96%). There the designer may rest. We will shortly show that this is not an optimal design.

Alternatively, the designer may base the design on the customer and over-design with $(5,5,5,2,3,1)$, cost \$37152 and SLOs (0.39 s, 99.998%). None of the above design decisions take loss into account. It is instructive to discover the values for loss for the above designs as well as for the design which minimizes the sum of cost plus loss as shown in section C.3.4 (see Table C.2).

For the best design, the SLOs are (average response time=0.625 s, availability=99.998%). It has lowest overall financial outlay, and the table clearly shows the high cost of choosing SLOs in an ad hoc fashion: a wrong choice can cost tens or even hundreds of thousands of dollars per month.

As a final experiment, it is instructive to see that the best design depends quite heavily on the importance of the business process being serviced. If one lessens the importance

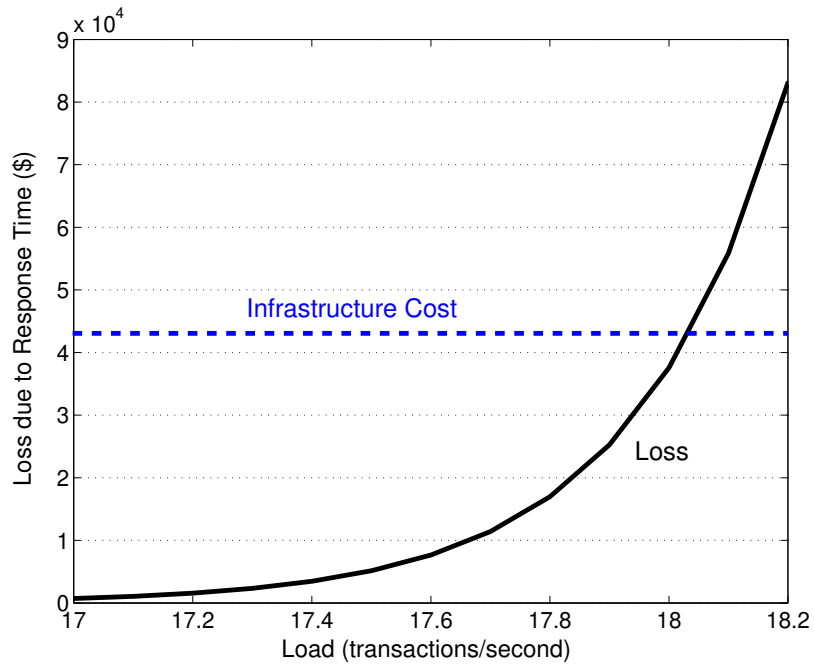


Figure C.2: Effect of Load on Loss

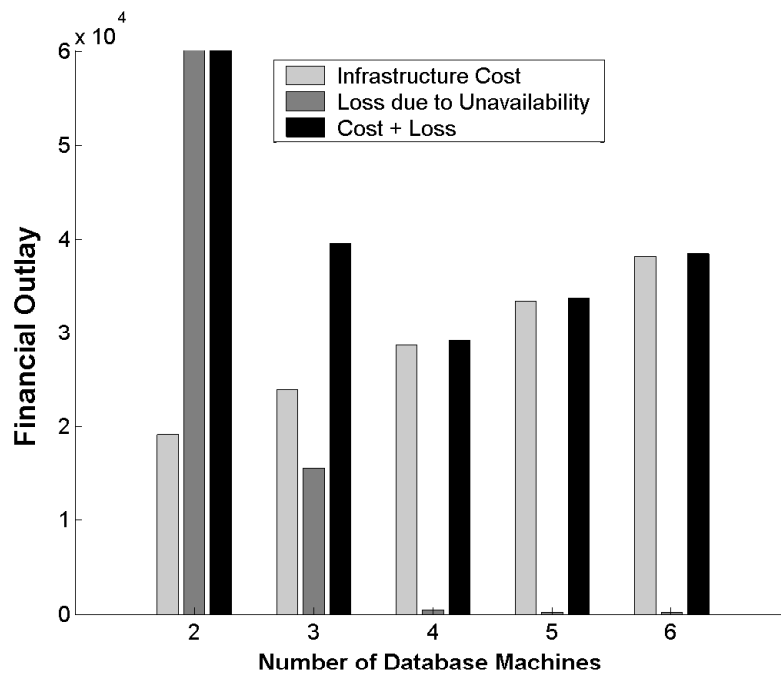


Figure C.3: Sensitivity of Loss due to Redundancy

Table C.2: Comparing designs

Infrastructure	Cost (\$)	Loss due to Response (\$)	Loss due to unavailability (\$)	Cost plus loss (\$)	The cost of choosing wrong (\$)
(1,2,1,1,2,1)	9141	20886	1697369	1727396	1698274
(3,3,3,1,2,1)	22201	21902	15428	59531	30409
(5,5,5,2,3,2)	37152	0	608	37760	8638
(3,4,4,1,2,2)(best)	28576	0	546	29122	0

of the BP by diminishing the average revenue per transaction by a factor of 10, the best design is (2,4,2,1,2,1), cost \$17396, total loss \$3243 and SLOs: (average response time=1.5 s, availability=99.97%). In this case, a much lower availability is best and the design is cheaper by \$11180 a month than if BP importance were not considered.

C.5 Related Work

Business Impact Management is a very new area of interest to researchers and practitioners that has not yet been consolidated. In the recent past, some problems typically faced in IT management are being studied through a business perspective [MBC04; BS04b; Mas02; CCD⁺02; CSDS03; LSW01; DHP01]. Some examples include incident prioritization [BS04b], management of Web Services [CSDS03], Business Process Management [CCD⁺02], etc. These references confirm a general tendency to view BIM as a promising way of better linking IT with business objectives. However, these references offer little in terms of formal business impact models to tie the IT layer to BP or business layers. This is one of our main contributions.

Although this paper stresses aspects of SLA Design, it is also licit to view the work as a method for IT infrastructure design (capacity planning). In this particular area, [JST03] describes a tool – AVED – used for capacity planning to meet performance and availability requirements and [AF02] describes a methodology for finding minimum-cost designs given a set of requirements. However, none of these references consider the problem of capacity planning from a business perspective, using business metrics. Furthermore, response time

considerations are not directly taken into account. Finally, [GLS04] considers the dynamic optimization of infrastructure parameters (such as traffic priorities) with the view of optimizing high-level business objectives such as revenue. It is similar in spirit to the work reported here, although the details are quite different and so are the problems being solved (SLA design is not the problem being considered). The model is solved by simulation whereas our work is analytical.

In the area of SLA design, HP's Open Analytics [TT05b] is a response to the downside of designing SLAs with current practices leading to a more formal approach as presented here. Open Analytics dictates that all assumptions leading to a performance decision must be made explicit and that all technical and financial consequences must be explained. "Open auditable mathematics, rather than wet finger in the air responses to requests [...]" must be used although details are not given.

Management by Contract [SB04] investigates how IT management can decide when it is better to violate an SLA or to keep compliance, according to a utility function that calculates the business impact of both alternatives. It is similar in spirit to our work, although it does not specifically address the problem of SLA design.

C.6 Conclusions

This paper has proposed a method whereby best values for Service Level Objectives of an SLA can be chosen through a business perspective. Business considerations are brought into the model by including the business losses sustained when IT components fail or performance is degraded. This is done through an impact model, fully developed in the paper. A numerical example consisting of a single e-commerce business process using a single IT service dependent on three infrastructure tiers (web tier, application tier, database tier) was used to show that the best choice of SLOs can be vastly superior to ad hoc design. A further conclusion is that infrastructure design and the resulting SLOs can be quite dependent on the "importance" of the BPs being serviced: higher-revenue BPs deserve better infrastructure and the method presented shows exactly how much better the infrastructure should be.

Much work can be undertaken to improve the results, among which the following are worth noting: a better availability model (such as presented in [JST03]) can be used to

approximate reality more faithfully; the load applied to the business process can be better modeled by following the Customer Behavior Model Graph approach [MAD04]; variations in the load applied to the BPs should be investigated; more complete impact models should be developed to be able to deal with any kind of BP, not only e-business BPs heavily dependent on IT; finally, the work should be extended to adaptive infrastructures and dynamic provisioning.

Acknowledgments.

We would like to acknowledge and thank the Bottom Line Project team. This work was developed in collaboration with HP Brazil R&D.

Appendix D

Business-Driven Design of Infrastructures for IT Services

Jacques Sauvé, Filipe Marques, Antão Moura

Universidade Federal de Campina Grande

{jacques,filipetm,antao}@dsc.ufcg.edu.br

Abstract: A methodology for designing data center infrastructure for Information Technology (IT) services is developed. The main departure from existing methodologies is that it evaluates and compares alternative designs using business metrics rather than purely technical metrics. Specifically, the methodology evaluates the business impact (financial loss) imposed by imperfect infrastructure. The methodology provides the optimal infrastructure that minimizes the sum of provisioning costs and business losses incurred during failures and performance degradations. Several full numerical example scenarios are provided and results are analyzed. The use of the method for dynamically provisioning an adaptive infrastructure is briefly discussed.¹

D.1 Introduction: The Problem

The infrastructure needed to provision an enterprise's Information Technology (IT) services is ever more complex. This infrastructure takes the form of large distributed multi-tier sys-

¹Submetido ao periódico Performance Evaluation, em Maio de 2006. / Submitted for Performance Evaluation Journal, in May of 2006.

tems with load-balanced and redundant server farms, together with load-balancers, firewalls and other associated network components. This paper investigates methodologies for designing such infrastructure. The standard way to perform this design is to select technical performance metrics such as availability, mean response time and perhaps throughput, choose values for these metrics (say 99,9% availability, 1.5 sec. mean response time, etc.) and strive to achieve a design that minimizes cost and meets the performance metrics. Current design methods assume particular architectural decisions such as the use of tiers, the use of clusters with fail-over functionality to meet availability restrictions, load balancing and caching to handle load and achieve scale. If we restrict the discussion of infrastructure design to the data center *server farm*, one of the most important component, then the design space (number and type of machines in the load-balanced clusters, number of redundant machines, and so on) is explored in the search of minimum-cost solutions.

A problem with current approaches is that business requirements are not well captured. Since IT is but a means to help the business meet its challenges, not formally taking business requirements into account is a major weakness – just how major will be seen later. Today’s approach in capturing business requirements is to talk to business managers and ask them what the business needs in terms of availability, response time, etc. These “requirements” are then expressed formally in a document called a Service Level Agreement (SLA), whereby the IT provider makes certain promises regarding the performance that the service consumer can expect when using IT services. The SLA defines certain Service Level Indicators (SLIs, such as availability) and sets thresholds for them (Service Level Objectives or SLOs). The problem, as eloquently described in [TT05a], is that business managers simply do not know how to translate their business needs into technical metrics; they have a feeling for business and *business* metrics, but not for *technical* metrics. Telling a business manager that availability for e-mail service is 99.95% or that mean response time for order entry is 1.2 sec. is next to meaningless, at least with respect to what impact these numbers imply on the business. As a result, SLOs are chosen using a “finger in the air”, ad hoc approach. Needless to say, this cannot be said to capture business requirements in a meaningful way. As will be seen in a later section, the problem just described can be said to be *major* since hundreds of thousands of dollars in financial outlay can easily be the difference between an ad hoc solution and one that formally takes business considerations into account, even for

medium-sized infrastructures.

As can be seen from the above discussion, infrastructure design and SLA design (choosing SLA parameter values) are really two faces of the same coin. By designing the infrastructure, one essentially *sets* the SLA thresholds, at least the ones associated with service performance. These design problems (infrastructure design and SLA design) are multiplied when the two parties signing the SLA are different businesses, in other words, when IT has been outsourced and the service provider is not an internal IT department belonging to the same company as the service consumers. In that case, there are two sets of business requirements to be evaluated and taken into account: the service provider's business requirements and the service consumer's business requirements. Also, in this case, other SLA parameters, such as penalties, must be agreed upon, and there is little in terms of formal methods to help in choosing such SLA parameters.

We therefore contend that one needs more mature and objective methods and models to design IT service infrastructure and SLAs. We present an approach based on a new area of research – Business-Driven IT Management (BDIM) – that formalizes the linkages between IT infrastructure, the services it provides and the business it serves. BDIM can be used to design infrastructure by taking those linkages into account. In essence, the models developed and presented further on can map both availability and response time into business process metrics, and from these to business metrics, enabling a formal approach to business-driven IT service provisioning and SLA negotiation. BDIM can be considered an evolution of Service Management. In the past, the IT department was closed in on itself, an attitude that was forced to change in the 1990's when Service Management and, particularly, Service Level Management (SLM) were introduced [Lew99b]. Through these practices, the IT department was forced to look at itself from the outside, using the client's point of view. It is from that period that the idea of IT offering a defined service with quality guarantees was born. However, even with SLM, the promises (described in SLAs) are still couched in technical language. BDIM takes these ideas further and aims to use the client's language – business metrics – when discussing service quality and business impact.

Having described the problem of interest in this section, the paper describes our proposal for a BDIM solution in high-level terms in section D.2. Section D.3 fully describes the performance models while section D.4 presents applications to several design scenarios.

Related work is described and compared to ours in section D.5. A summary of the paper contents, conclusions and discussions of future directions for work in the area of BDIM are offered in section D.6.

D.2 A Method to Capture the Business Perspective in IT Infrastructure Design

This section describes – at a high level of abstraction – the method we have devised to include a business perspective in infrastructure design.

D.2.1 A Layered Model

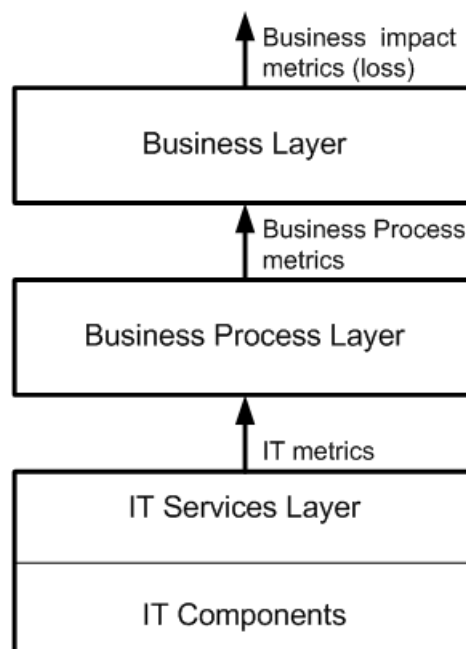


Figure D.1: Model Layers

In order to capture business requirements while designing IT infrastructure, one must include entities outside the IT world in the models. Figure D.1 shows the three layers that are used here to that effect. The bottom-most layer is the IT services layer, from which one can obtain IT metrics such as availability and response time for IT services. At the top, one can observe the business layer where business metrics meaningful to non-IT personnel are

estimated. Thus, to get a *business perspective*, one must relate what happens at the business layer as a result of events occurring at the IT layer. In between these two sits the business process (BP) layer where the actual users of the IT services are modeled. The BP layer is used to cross the gap between the IT and business layers, whenever that is deemed useful. In other words, it may sometimes be difficult to estimate the impact of IT events on the business directly but this may be made easier if one first estimates the effect on the BPs, and then from the BPs to the business. This is what is done in this paper. However, it is conceivably possible to go straight from IT to the business and forego the BP layer entirely. A few general comments may now be made concerning the models used in each layer; full details are given in section D.3.

IT layer models. These models calculate IT service metrics such as availability and response time. The input to the models includes details of arrival processes of work to be performed (web sessions initiation inter-arrival times, for example), details of the infrastructure components, their attributes (failure rates, speed, etc.) and relationships, and how they are combined to offer services. As a result, IT service metrics can be calculated. The models are typically reliability models (to calculate availability) and queuing theory models (to calculate response time statistics). Observe that the infrastructure considered here is assumed to be *static*. The use of the business-driven method in dynamic provisioning scenarios is certainly possible and very interesting but is not the focus of the work presented here; more on this in the conclusions.

BP layer models. At the BP layer, the models are meant to provide *linkage* (mapping) between the IT layer and the business processes, from which BP metrics can be calculated. Business metrics can, for instance, be throughput (number of business transactions per second), number of people affected by a service fault, or any other value associated with entities that make sense to a business manager. Although BP layer models can be more general, in this paper, we concentrate on the BP throughput metric. Our scenarios are thus applicable to any business process whose *health* can be gauged from its throughput. This is the case, for example, for an e-commerce sales business process where BP throughput is simply the rate of sales transactions performed through the web site. In particular, we are interested in evaluating:

- How is BP throughput affected when IT services are unavailable?

- How is BP throughput affected when IT services are slow?

As a concrete example, it is clear that an e-commerce site will have zero throughput whenever the underlying IT services (web service or database service, say) are unavailable. When, on the other hand, IT services are slow, customers will defect and not conclude their purchases, thus causing a drop in sales throughput. The models developed below estimate the value of this loss of BP throughput.

Business layer models. Here also, the models provide linkage between the layer below (BP) and the outgoing metrics. In this case, one is interested in business metrics. These are different from BP metrics since they are usually monetized and meant for consumption by top executives that may not be familiar with specific BP metrics. These metrics are meant to capture the *bottom line* in terms of the impact of IT events on the business. Monetized metrics have the advantage of allowing easy numerical comparison between alternative scenarios and, furthermore, are directly understandable by managers and executives. For this reason, the impact models used at the business layer are referred to as *revenue models*. Continuing the concrete example given above, a loss in BP throughput can be mapped to *financial business loss* in this layer.

With models of the types described above, one can answer questions such as the following:

- “What is the (negative) impact of IT faults and performance degradations on numerical metrics associated with the business?”
- “How much does it cost the business per minute if server XYZ is down?”
- “How much does it cost the business per minute if mean access time to the web site pages exceeds 5 sec. per page?”

Notice that in this case the business metric is implied to be *financial loss* due to IT infrastructure imperfections.

The observant reader will have noticed that the models allow one to monetize the impact of (un)availability and also the impact of high response time *separately*. The contribution of each factor can now be combined, offering a way of joining availability and response time considerations under a *single design problem*: it is the use of monetized business metrics that allows the two factors, typically handled separately, to be amenable to joint treatment.

D.2.2 Optimization problems dealing with infrastructure design

The fact that business metrics are numerical and can be ordered (a lower value is better, say) allows us to pose *optimization problems* using these business metrics as objective functions. Consider the following informal example (a formal version is given later). A company incurs a monthly financial cost for its IT infrastructure. Due to imperfections in this infrastructure (components fail, resources saturate), IT services are not always available and services may be slow, both of which cause the business to suffer *financial loss*. Let us now compare several points of the infrastructure design space. Intuitively, one expects that an infrastructure with lower monthly cost will be more imperfect (components will fail more, or will have less capacity), causing higher business losses; the other extreme is a very expensive but very high-quality infrastructure (much redundancy, much load balancing to handle load) that will cause less business loss. Thus, if we *sum* monthly cost and monthly business loss, these two extremes seem to be suboptimal since the overall financial outlay will be high, either due to high cost or to high loss; there likely will exist a midpoint with minimal financial outlay.

The main point to be understood here is that this is the point in design space that really represents the business requirements: the business wishes to operate at the lowest point in financial impact (cost + loss) to receive IT services, whatever the actual availability and mean response time figures happen to be. Having designed an infrastructure that operates at this optimal point in design space, one can now consider the SLA: SLO values (minimum desired availability, maximum mean response time, ...) are simply calculated *after* infrastructure design, and not imposed *before* design, as is currently done.

We can now proceed to formalize the main optimization problem whose solution comprises infrastructure design according to our method. At this point, only top-level details are given; section D.3 will complete the exposition. IT infrastructure must be designed to provide IT services. Although the method is general, the present paper concerns itself only with the server farm. Table D.1 summarizes the symbols needed in this section. The IT services are implemented in a multi-tier architecture and the servers are thus divided in tiers. The resources in a tier make up a *resource class*. For example, there could be a resource class made up of web servers (for the web tier), another resource class made up of application servers and a third made up of database and ERP servers (for the data tier). There are a total of RC resource classes. Resource class RC_j consists of a total of n_j servers, of which m_j

Table D.1: Notational summary for main optimization problem

Symbol	Meaning
RC	Set of resource classes in IT infrastructure (e.g. tiers)
RC_j	The j th resource class
n_j	The total number of resources (machines) in RC_j
m_j	The total number of load-balanced machines in RC_j
ΔT	Any time period over which cost and loss are evaluated, typically a month.
$C(\Delta T)$	The infrastructure cost over the time period ΔT
$L(\Delta T)$	The financial loss over the time period ΔT due to imperfections in the infrastructure

are organized as a load-balanced cluster. The m_j load-balanced machines enable the tier to handle the input load while the $n_j - m_j$ standby machines provide the required availability. The cost of the infrastructure over an evaluation period ΔT (typically one month) is $C(\Delta T)$ and the monthly financial loss suffered over this period when using this infrastructure to provision the IT services is $L(\Delta T)$. The design problem can be posed as an optimization problem as follows:

Find:	For each resource class RC_j , the total number of machines n_j and the number of load-balanced machines m_j
By minimizing:	$C(\Delta T) + L(\Delta T)$, the total financial impact on the business over the time period ΔT
Subject to:	$n_j \geq m_j$ and $m_j \geq 1$

We will turn our attention to the calculation of $C(\Delta T)$ and $L(\Delta T)$ in section D.3.

D.3 Performance Models

In this section, we derive analytical expressions for cost $C(\Delta T)$ and loss $L(\Delta T)$, after which the optimization problem presented previously can be resolved. Many submodels are

Table D.2: Notational summary for infrastructure cost

Symbol	Meaning
RC_j	The j th resource class
R_j	An individual resource in RC_j
P_j	The set of components that make up resource R_j
$P_{j,k}$	The k th component in P_j
$c_{j,k}^{active}$	The cost rate of component $P_{j,k}$ if active
$c_{j,k}^{standby}$	The cost rate of component $P_{j,k}$ if on standby

needed to complete the overall picture. These are the cost model (section D.3.1), the loss model (section D.3.2), which, in turn, includes an availability model based on reliability theory (section D.3.3) and a queuing model to calculate response time (section D.3.4). In fact the merging of these several dimensions into a single design problem is one of the contributions of this work.

D.3.1 Calculating infrastructure cost

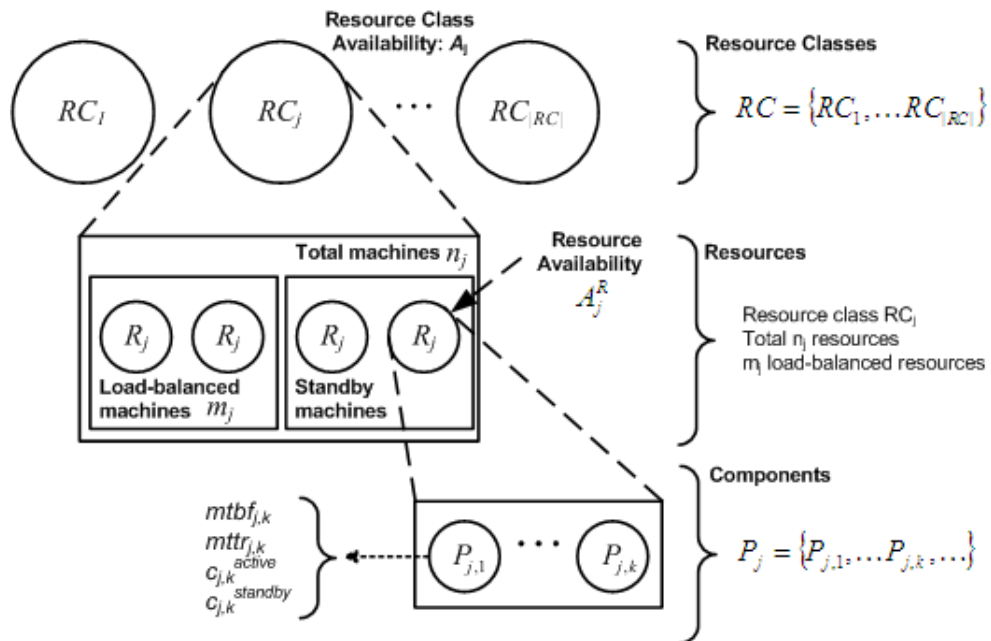


Figure D.2: Cost and Availability Model Entities

In order to determine infrastructure cost, one needs a model of the physical entities (see

Figure D.2 and the notational summary in Table D.2). Each resource class (tier) consists of n_j identical resources (server machines). An individual resource $R_j \in RC_j$ consists of a set $P_j = \{P_{j,1}, \dots, P_{j,k}, \dots\}$ of components. As an example, a database server can consist of three components: hardware, operating system software and DBMS software. Each infrastructure component $P_{j,k}$ has a cost rate $c_{j,k}^{active}$ when actively used (that is, used in a load-balanced server) and has a cost rate $c_{j,k}^{standby}$ when on standby. These values are cost rates (cost per unit time) for the component and may be calculated by dividing the total cost of ownership (TCO) by the amortization period for the component. The standby cost is typically lower since no licensing fees, refrigeration and electricity costs need be paid, for example. The cost of the infrastructure over a time period of duration ΔT can be calculated as the sum of individual cost for all components.

$$C(\Delta T) = \Delta T \sum_{j=1}^{|RC|} \left(\sum_{l=1}^{m_j} \sum_{k=1}^{|P_j|} c_{j,k}^{active} + \sum_{l=1}^{n_j-m_j} \sum_{k=1}^{|P_j|} c_{j,k}^{standby} \right) \quad (D.1)$$

In this equation, the index j runs over resource classes, l runs over resources and k runs over components.

D.3.2 Calculating business loss from a general business impact model

We now turn our attention to the calculation of loss, $L(\Delta T)$. This subsection considers a general business impact model while details are given in the next two subsection. Please refer to Figure D.3 and the notational summary in Table D.3. For simplicity, we assume the existence of a single IT service; generalization to several services is straightforward. Users access IT services through *sessions*. Sessions are started at a rate of γ sessions per second. Of these sessions, a fraction, f , end up generating revenue for the business, while others do not. This is typical of e-commerce which may be taken as the prototypical scenario for the analysis that follows. Thus, the net revenue-generating business process throughput is ideally $\gamma \cdot f$ business transactions per second.

BP throughput may be lost due to two causes: loss due to service unavailability and loss due to customer defection caused by high service response times. Let A be the service availability – this value will be calculated in a further section. The lost BP throughput due to service unavailability, is thus $\Delta X^A = \gamma \cdot f \cdot (1 - A)$. Now, when service is available, it may be too slow. Let $B(y)$ be the probability that, during a session, any visit to a site page

Table D.3: Notational summary for business impact model

Symbol	Meaning
γ	The rate at which sessions are initiated at the site
f	The fraction of sessions that generate revenue (type RG sessions)
A	The service availability
$B(y)$	The probability that, during a session, any visit to a site page resulted in response time greater than y
T^{DEF}	The response time threshold after which customer defection occurs
ϕ	The average revenue per business transaction
ΔT	Any time period over which cost and loss are evaluated, typically a month.

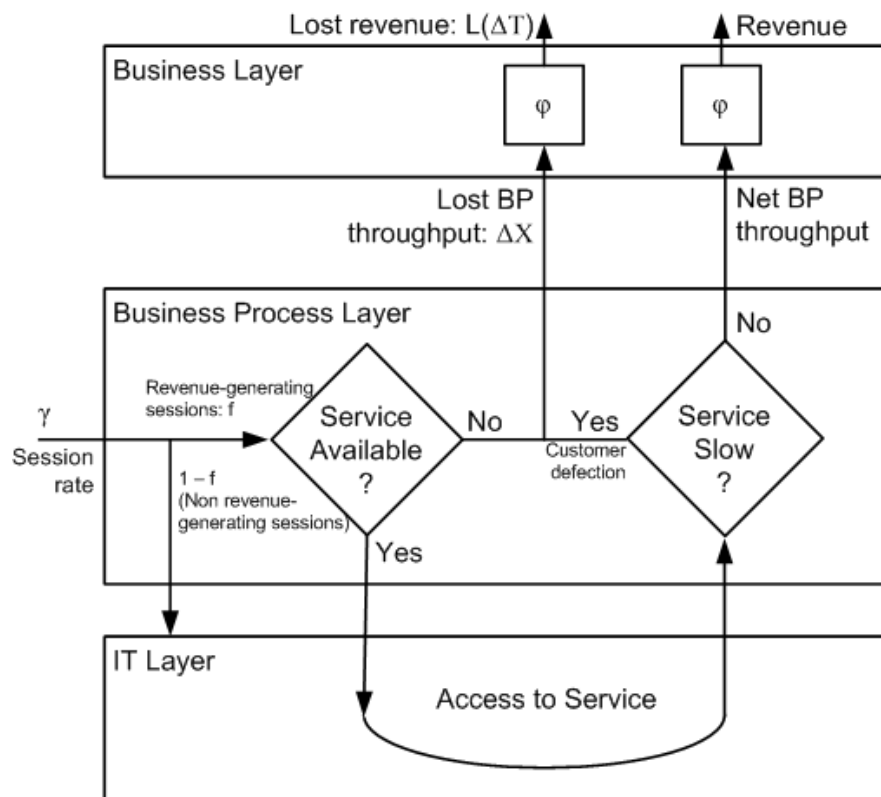


Figure D.3: Business Impact Model – Crossing the IT-Business Gap

– or any interaction with the service – results in response time greater than y . Further let T^{DEF} be the response time threshold after which customer defection occurs (the literature frequently mentions 8 seconds as a typical value). Thus the lost BP throughput due to customer defection, is $\Delta X^T = \gamma \cdot f \cdot A \cdot B(T^{DEF})$. Total lost throughput due to both causes is $\Delta X = \Delta X^A + \Delta X^T$.

Having found the effect of IT faults and performance degradations on BP throughput, we may now move up to the next layer and consider business loss. In the business layer, assume that ϕ is the average revenue per completed business transaction. Thus the lost revenue over a time period ΔT is $L(\Delta T) = \Delta X \cdot \phi \cdot \Delta T$, leading to the final result for business loss:

$$L(\Delta T) = \gamma \cdot f \cdot (1 - A + A \cdot B(T^{DEF})) \cdot \phi \cdot \Delta T \quad (\text{D.2})$$

We have thus arrived at the result we were seeking, the calculation of business loss, as long as expressions can be found for service availability and response time distribution for BP sessions; this will be done in the next two subsections.

Some final words are in order. First, observe how the several dimensions (cost, availability, response time) can all be treated in a single optimization problem, due to the common BP metric used (lost BP throughput). Secondly, the impact model used implies that, when service is unavailable, BP throughput stops entirely; this is applicable to BPs that are heavily dependent on IT, such as e-commerce. Different impact models would need to be developed for other types of business processes.

We now turn our attention to the calculation of service availability, A , and customer defection probability, $B(T^{DEF})$.

D.3.3 Calculating service availability

This section derives an expression for A , the service availability. The notational summary in Table D.4 will be helpful to the reader.

A single resource $R_j \in RC_j$ is made up of a set P_j of components $P_{j,k}$. Each such component is subject to failure. Let $mtbf_{j,k}$ and $mtrr_{j,k}$ be, respectively, the Mean-Time-Between-Failures (MTBF) and Mean-Time-To-Repair (MTTR) of component $P_{j,k}$. Values from MTBF can be obtained from component specifications or historical logs whereas values for MTTR will typically depend on the type of service contract available. Now, we assume

Table D.4: Notational summary for availability model

Symbol	Meaning
R_j	An individual resource in RC_j
P_j	The set of components that make up resource R_j
$P_{j,k}$	The k^{th} component in P_j
A	The service availability
A_j	The availability of resource class RC_j
A_j^R	The availability of an individual resource R_j from class RC_j
$mtbf_{j,k}$	The Mean-Time-Between-Failures of component $P_{j,k}$
$mttr_{j,k}$	The Mean-Time-To-Repair of component $P_{j,k}$

that all resource components must be operational for the resource itself to be operational; this is readily seen if one remembers that the resources are typically hardware, operating system software, and some kind of middleware or application software. If desired, different assumptions can easily be accommodated. From these values and reliability theory [Tri82], we have the availability of resource R_j :

$$A_j^R = \prod_{k=1}^{|P_j|} \left[\frac{mtbf_{j,k}}{mtbf_{j,k} + mttr_{j,k}} \right]$$

We may now combine resources into a load-balanced cluster with standby machines. In order to properly handle load, all m_j load-balanced machines in resource class RC_j must be operational; since there are n_j machines overall, we have the following availability for the whole RC_j cluster (this is called “m-out-of-n availability”):

$$A_j = \sum_{k=m_j}^{n_j} \left[\binom{n_j}{k} \cdot (A_j^R)^k \cdot (1 - A_j^R)^{n-k} \right]$$

Finally, all resource classes (tiers) must be operational for the service to be operational, from which:

$$A = \prod_{j=1}^{|RC|} A_j$$

D.3.4 Calculating customer defection probability

Customer defection probability ($B(T^{DEF})$) can be obtained from the response time *distribution* for requests to the IT infrastructure. This is a departure from most models based on queuing theory that usually obtain only the mean response time. In the following model, we assume a fixed average input load and static infrastructure; section D.4 relaxes the first assumption and dynamic infrastructures are left for future work in self-configuring autonomic computing. The notational summary in Table D.5 will help the reader follow the mathematical development in this section. The discussion that follows is couched in terms of an e-commerce site, although, as mentioned previously, the method is more generally applicable and the models used apply to scenarios in which customer defection affects BP throughput.

In order to assess response time performance, one must model the load applied to the IT resources. Access to the e-commerce site consists of sessions, each generating several visits to the site's pages. The mathematical development that follows is (initially) based on the Customer Behavior Model Graph (CBMG) [MA00], that allows one to model how customer-initiated sessions accessing a web site impose load on the IT infrastructure. The use of the CBMG model will then be extended to include business impact. A CBMG consists of a set S of states and transitions between states occurring with particular probabilities. Each state typically represents a web site page that can be visited and where a customer interacts with the e-commerce site. As an example, consider Figure D.4 that shows the states and the transition probabilities for a simple but typical e-commerce site. The customer always enters through the Home state and will then Browse (with probability 0.4) or Search (with probability 0.6). The Details state represents viewing the details of a product and the other states are self-explanatory.

Some of these states are revenue-generating (for example, a "Pay" state where the customer pays for items in a cart). Sessions are initiated at a rate of γ sessions per second. For our purposes, we divide the sessions into two types: type RG sessions are revenue-generating while type NRG sessions are non-revenue-generating. Customer behavior for each session type is modeled by means of its own CBMG [MA00]. The particular CBMG shown in Figure D.4 is an example applicable to type RG sessions since the Pay state is visited with non-zero probability. For type NRG sessions, the CBMG will include the same states but with different transition probabilities; for example, there will be no path leading

Table D.5: Notational summary for response time analysis

Symbol	Meaning
S	The set of states in the Customer Behavior Model Graph (CBMG). Each state represents a particular interaction with the e-commerce site (browse, search, etc.)
S_t	The set of transient states in the CBMG
T^{DEF}	The response time threshold after which customer defection occurs
$T_r(y)$	The cumulative distribution of response time for requests of the CBMG
$B_r(y)$	The probability that response time has exceeded y during a single visit to state r
$B_r^{mult}(y)$	The probability that, during a session, any of the multiple visits to state r resulted in response time greater than y
$B(y)$	The probability that, during a session, any visit to a site page resulted in response time greater than y
\bar{T}_r	The average response time for each state r for the RG CBMG
\bar{T}^{RG}	The average response time seen by revenue-generating customers
\bar{T}^{NRG}	The average response time seen by non-revenue-generating customers
γ	The rate at which sessions are initiated at the site
f	The fraction of sessions that generate revenue (type RG sessions)
$p_{i,r}^{RG}$	The probability of going from state i to state r in the RG CBMG
$p_{i,r}^{NRG}$	The probability of going from state i to state r in the NRG CBMG
V_r^{RG}	The average number of visits to state r in RG CBMG
V_r^{NRG}	The average number of visits to state r in NRG CBMG
α_j	The speedup factor for resources in resource class RC_j
λ_r	The arrival rate of requests to the IT infrastructure in state r
$\mu_{r,j}$	The service rate at resource class RC_j for transactions from state r
$\lambda_{r,j}$	The arrival rate of requests to a resource class RC_j for transactions from state r
$\rho_{r,j}$	The utilization of resources in resource class RC_j in processing transactions from state r
ρ_j	The total utilization of resources in resource class RC_j

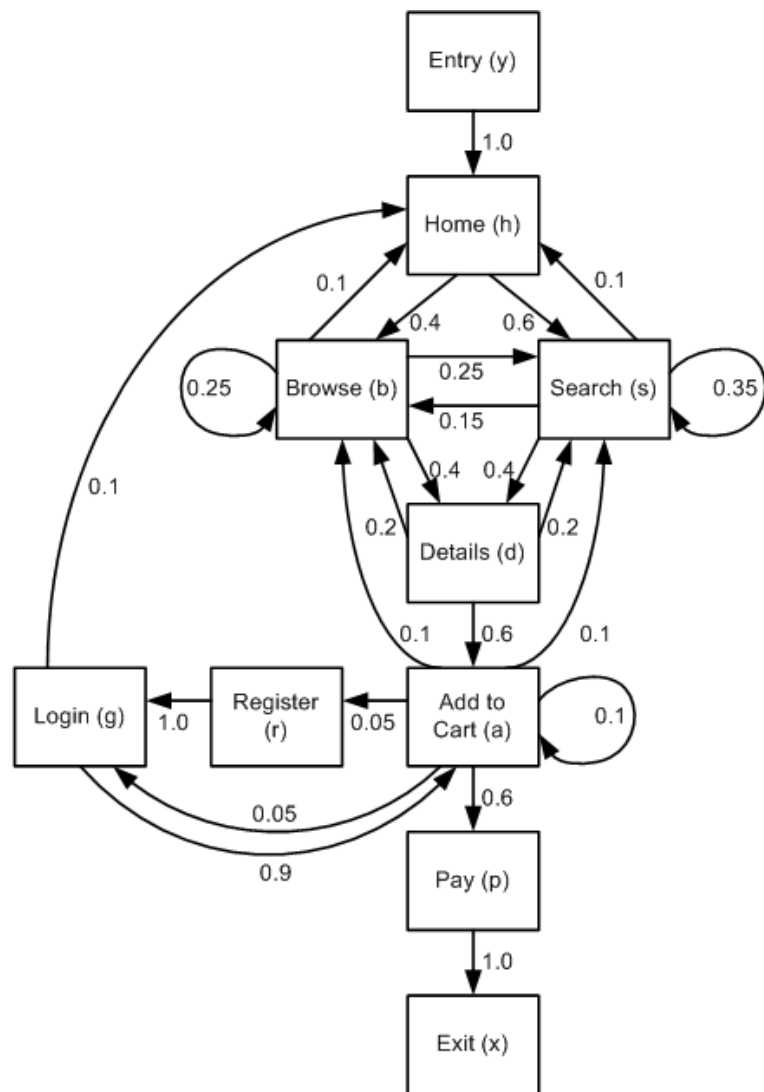


Figure D.4: CBMG for the e-commerce site

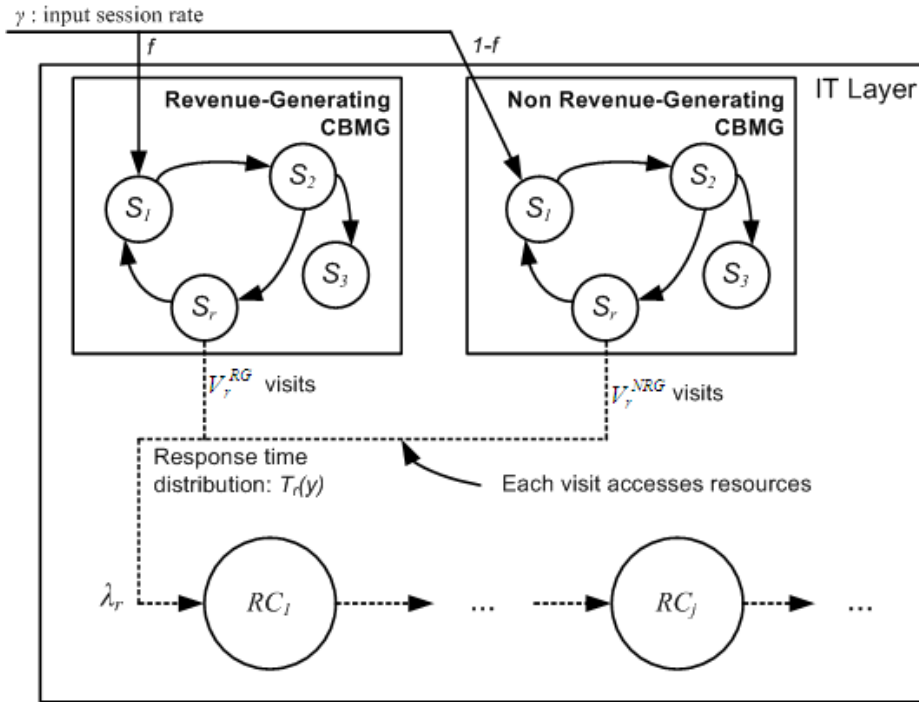


Figure D.5: Customer behavior model

to the Pay state, the only revenue-generating state in this particular graph. Figure D.5 shows how CBMGs are inserted into the overall performance model. The fraction of sessions that are revenue-generating is denoted by f . The transition probability matrices have elements $p_{i,r}^{RG}$, the probability of going from state i directly to state r in the RG CBMG, and $p_{i,r}^{NRG}$ for the NRG CBMG. Observe that S , f and all transition probabilities for these graphs can be obtained automatically from web server logs, as shown in [MA00].

Since a CBMG is a Markov chain with absorbing states (the Exit state, in figure D.4), the average number of visits per session to a transient state r , where $r \in S_t$ and $S_t \subset S$ can be obtained by solving the following set of linear equations [HPS86]:

$$V_1^{RG} = 1$$

$$V_r^{RG} = \sum_{i \in S_t} (V_i^{RG} \cdot p_{i,r}^{RG})$$

where V_r^{RG} is the average number of visits to state r in the RG CBMG. The situation for the NRG CBMG is similar and the average number of visits to state r is V_r^{NRG} . Each visit to a CBMG state results in a request being made to the IT resources. In other words, the CBMG shows how business transactions are transformed into IT service transactions (or requests).

We now need to find $B(T^{DEF})$, the probability that customers will defect due to response time exceeding T^{DEF} while navigating during a session. In order to find this probability, the IT services are modeled using a multi-class open queuing model. Open queuing models are adequate when there is a large number of potential customers, a common situation for e-commerce. Since, in each CBMG state, the demand made on the IT infrastructure is different, each state in the CBMG represents a traffic class in the queuing model. Let us examine state r . The arrival rate of requests corresponding to this state is $\lambda_r = \gamma \cdot (f \cdot V_r^{RG} + (1 - f) \cdot V_r^{NRG})$ requests per second; these requests arrive at the first resource class (say the web tier). Requests may demand service from all resource classes, as detailed below. Demand applied by each request from state r on resources from resource class RC_j is assumed to be $D_{r,j}$ seconds. In fact this is the service demand if class RC_j consists of “standard” processing resources. In order to handle the case of more powerful hardware, assume that a resource in class RC_j has a processing speedup of α_j compared to the standard resource. Thus, service time for a request is $D_{r,j}/\alpha_j$ and the service rate at a class RC_j resource for requests from state r is:

$$\mu_{r,j} = \frac{\alpha_j}{D_{r,j}}$$

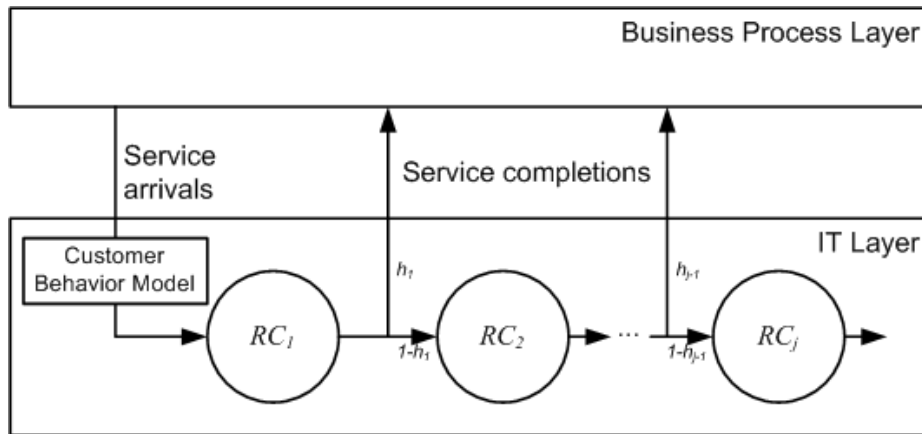


Figure D.6: Flow of a request between tiers and cache model

Now, in order to more accurately model large-scale infrastructures, cache effects must be considered. Requests access resource classes in a predetermined order, as shown in Figure D.6; for example, resource classes are typically accessed in the sequence: web tier \rightarrow application tier \rightarrow data tier. In order to capture cache effects we assume that each resource

class RC_j has a cache with hit ratio, h_j , which is the probability of a request hitting the RC_j cache and, as a result, not having to access the resource classes down the line. For the last resource class, we have $h_{|RC|} = 1$.

Since a request may not visit all the resource classes, we now need to find p_j^{access} , the probability of a request accessing resource class RC_j . Recalling that h_j is the cache hit ratio for resource class RC_j , we have: $p_1^{access} = 1$, $p_2^{access} = (1 - h_1)$ and, in general:

$$p_j^{access} = \prod_{k=0}^{j-1} (1 - h_k)$$

where, to simplify the equation, $k = 0$ represents a dummy resource class with hit ratio $h_0 = 0$.

Therefore, the load applied to resource class RC_j is $\lambda_r \cdot p_j^{access}$. Finally, since there are m_j identical load-balanced parallel servers used for processing in resource class RC_j , we use the equivalent server technique shown in [MAD04] to find response time; the equivalent single server has input load:

$$\lambda_{r,j} = \frac{\lambda_r \cdot p_j^{access}}{m_j}$$

Thus the utilization $\rho_{r,j}$ of class RC_j resources in processing requests from state r is:

$$\rho_{r,j} = \frac{\lambda_{r,j}}{\mu_{r,j}} = \frac{\lambda_r \cdot p_j^{access}}{m_j} \cdot \frac{D_{r,j}}{\alpha_j}$$

The total utilization ρ_j of class RC_j resources due to requests from all states is:

$$\rho_j = \sum_{r=1}^{|S|} \rho_{r,j}$$

Observe that, when load is so large that any $\rho_j \geq 1$, then we have $B(T^{DEF}) = 1$, since response time is very high for saturated resources. Now, in order to find $B(T^{DEF})$ when $\rho_j < 1$, three subproblems must be considered: i) how to combine the effects of all resource classes for a single visit to a CBMG state r , including cache effects; ii) how to combine the effects of the several visits made to this state; and iii) how to combine results for all CBMG states.

Combining resource classes and considering cache effects. Let us find the cumulative distribution of response time, $T_r(y) = \Pr[\tilde{T}_r \leq y]$, where \tilde{T}_r is the continuous random variable corresponding to the response time seen by the customer making a single visit to state

r . In order to find this distribution, our analysis must capture the effect of having several resource classes, each with its own cache. Response time \tilde{T}_r is made up of several components, one for each resource class accessed. The resource classes actually accessed by a request depend on cache behavior and, consequently, \tilde{T}_r is a random sum of non-identically distributed random variables. The distribution of the random variable representing the number of resource class accessed by a request in state r is given by $p_k^{stop} = h_k \cdot p_k^{access}$, which is the probability that a request accesses resource class RC_k , hits the cache and stops further progression to resource classes down the line. Knowing this distribution of the probability of accessing resource classes, we can proceed to capture the cache effects on \tilde{T}_r .

In order to make mathematical treatment feasible, assume Poisson arrivals (this is reasonable for stochastic processes with large population and has been shown to be a good approximation in [MAR⁺03b]) and exponentially distributed service times. From queuing theory, the Laplace transform of response time (waiting time plus service time) for a single-server queue is:

$$T^*(s) = \frac{a}{s + a}$$

where $a = \mu \cdot (1 - \rho)$, μ is the service rate and ρ is the utilization. Assuming for a moment that a request stops at resource class RC_k , then \tilde{T}_r is a sum of k random variables and the Laplace transform of its distribution is simply the product of the Laplace transforms of its k components, since we assume independence in response time between resource classes. Letting now k vary, and recalling that the probability that a request stops at the k^{th} resource class is p_k^{stop} , we can use the theorem of total Laplace transforms to write:

$$T_r^*(s) = \sum_{k=1}^{|RC|} \left[\left[\prod_{j=1}^k T_{r,j}^*(s) \right] \cdot p_k^{stop} \right]$$

where

$$T_{r,j}^*(s) = \frac{a_{r,j}}{s + a_{r,j}},$$

represents the Laplace transform of the probability distribution function of the resource class RC_j response time while processing requests from state r and $a_{r,j} = \mu_{r,j} \cdot (1 - \rho_j)$.

Inverting the above transform yields the probability density function of response time, which is integrated to find the cumulative probability distribution function (PDF) of response time $T_r(y)$. In its turn,

$$B_r(T^{DEF}) = 1 - T_r(T^{DEF}) \quad (D.3)$$

is the probability that response time has exceeded T^{DEF} during a single visit to state r and that the customer will defect.

Taking multiple visits to a state into account. Let us now capture the effects of the several visits made to CBMG states. Since a state r can be visited many times, we need to calculate the discrete density function of the number of visits to each state of the RG CBMG in order to work out $B_r^{mult}(T^{DEF})$, the probability that response time has exceeded T^{DEF} in any of the multiple visits to state r . Once we have $B_r^{mult}(T^{DEF})$, it is possible to obtain $B(T^{DEF})$ as will be shown below.

Let \tilde{Q} be a discrete random variable denoting the number of visits to transient state r . Since a CBMG is a Markov chain with absorbing states, the discrete density function of the number of visits to each transient state of the RG CBMG can be obtained from standard Markov chain theory [HPS86]:

$$q_{r,m} = \tau_{1,r}^{RG} \cdot (\tau_{r,r}^{RG})^{m-1} \cdot (1 - \tau_{r,r}^{RG}), \quad m \geq 1 \quad (D.4)$$

where $q_{r,m}$ is the probability that state r receives m visits, $\tau_{i,r}^{RG}$ denotes the probability that, starting at state i , the RG CBMG will be in state r at some positive time. This formula can be intuitively appreciated as follows: starting at state 1 (the initial state), a single visit is made to state r , this state is revisited $m - 1$ times and never returned-to again. $\tau_{i,r}^{RG}$ is found by solving the following relations [HPS86]:

$$V_r^{RG} = V_{1,r}^{RG} = \frac{\tau_{1,r}^{RG}}{1 - \tau_{r,r}^{RG}} \quad (D.5)$$

and

$$\tau_{i,r}^{RG} = p_{i,r}^{RG} + \sum_{k \neq r} (p_{i,k}^{RG} \cdot \tau_{k,r}^{RG}) \quad (D.6)$$

With the probability density function of the number of visits to each state r of RG CBMG, one finds $B_r^{mult}(y)$, the probability that response time has exceeded y during multiple visits to state r :

$$B_r^{mult}(y) = \sum_{m=1}^{\infty} T_r(y)^{m-1} \cdot B_r(y) \cdot q_{r,m} \quad (D.7)$$

The above equation is interpreted as follows: there will be defection after m visits if there is no defection during the first $m - 1$ visits and defection during the last visit. As an aside, the above equation is an infinite sum but can be calculated to a desired accuracy through a finite sum.

Combining results for all states. We are now ready to find $B(T^{DEF})$, our final goal in this analytical development. Customer defection will occur and cause business loss only in the revenue-generating sessions. The crucial fact to be understood is that, if the response time during any visit in any state r exceeds the threshold T^{DEF} , then defection will occur; in other words, a customer defects when any page access becomes too slow. Put differently, defection will *not* occur if all response times are within the threshold during multiple visits to state r . Of course, this is only an approximation to customer behavior: a customer may simply give up or defect due to high prices, say. Since we only wish to model IT-business linkage, defection behavior is only tied to service quality. We can thus say:

$$B(T^{DEF}) = 1 - \prod_{r \in S_t} [1 - B_r^{mult}(T^{DEF})] \quad (D.8)$$

Finally, one may desire to evaluate the average response time for each state, possibly to be included in an SLA. Average response time for may be found from the Laplace transform as follows:

$$\bar{T}_r = - \left. \frac{dT_r^*(s)}{ds} \right|_{s=0}$$

Average response time to the site (over all states) is the weighted average of \bar{T}_r using the average number of visits as weights. Letting \bar{T}^{RG} be the average response time for the RG CBMG and \bar{T}^{NRG} for the NRG CBMG we have:

$$\bar{T}^{RG} = \frac{\sum_{r \in S_t} [\bar{T}_r \cdot V_r^{RG}]}{\sum_{r \in S_t} V_r^{RG}}$$

$$\bar{T}^{NRG} = \frac{\sum_{r \in S_t} [\bar{T}_r \cdot V_r^{NRG}]}{\sum_{r \in S_t} V_r^{NRG}}$$

With the above equations for cost $C(\Delta T)$ and loss $L(\Delta T)$, one may now solve the optimization problem discussed in section D.2.2. We have used simple hill-climbing techniques to perform the optimization. Since we have not shown that the objective function is convex, we cannot claim that the minimum value found is a global minimum; however, repeated optimizations from different starting points in design space have always converged to the same final value.

Table D.6: Transition probabilities in NRG CBMG

	y	h	b	s	g	p	r	a	d	x
Entry (y)		1.00								
Home (h)			0.55	0.40						0.05
Browse (b)		0.10	0.50	0.20					0.10	0.10
Search (s)		0.10	0.15	0.40					0.25	0.10
Login (g)		0.60	0.30							0.10
Pay (p)										1.00
Register (r)		0.50			0.40					0.10
Add to Cart (a)			0.40	0.30	0.05		0.05	0.05	0.10	0.05
Select (d)			0.45	0.40				0.05		0.10

D.4 Application of the Method and Models

In this section the mathematical analysis derived above is used to exercise the IT infrastructure design process for an e-commerce site. The values for input parameters are typical for current technology [JST03] [MA00] [MAD04].

The e-commerce site has a revenue-generating CBMG as shown in Figure D.4. Table D.6 shows the transition probabilities for the non-revenue-generating CBMG. Transition probabilities for a given site can be obtained from web server log files. From these CBMGs, one can calculate the average number of visits shown in Table D.7. For RG sessions, the Pay state is always visited whereas it is never visited in NRG sessions. The IT infrastructure includes three resource classes: web tier, application tier and database tier. Table D.8 and Table D.9 show the parameters used, except where otherwise noted. In Table D.8, tuples such as (a,b,c) represent parameter values for the three resource classes (web, application, database); also, each resource is made up of three components: (hardware (hw), operating system (os), application software (as)).

D.4.1 Behavior of the loss metric

Before delving into infrastructure and SLA design, let us examine the behavior of the loss metric as availability and response time change. This is shown in Figure D.7 and Figure D.8,

Table D.7: Average number of visits to each state

State (r)	RG Session (V_r^{RG})	NRG Session (V_r^{NRG})
Entry	1.000	1.000
Home	2.456	1.780
Browse	4.159	4.248
Search	5.417	3.510
Login	0.155	0.005
Pay	1.000	0.000
Register	0.086	0.003
Add to cart	1.724	0.069
Select	3.685	1.309
Exit	1.000	1.000

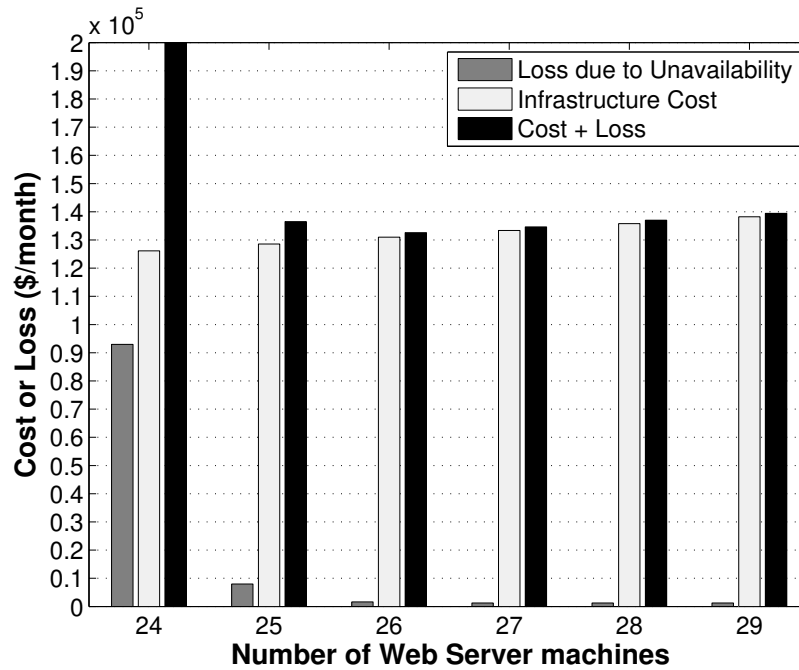


Figure D.7: Sensitivity of Loss due to Redundancy

Table D.8: Parameters for examples

Parameters	Values
T^{DEF}	8 seconds
ϕ	\$1 per transaction
γ	50 transactions per second
f	25%
ΔT	1 month
α_j	(1,1,3)
$C_{j,k}^{active}$ (\$/month)	hw =(2750, 3175, 11000) os=(412.5, 412.5, 412.5) as=(152.5, 87.5, 1650)
$C_{j,k}^{standby}$ (\$/month)	hw =(2000, 2300, 8000) os=(300, 300, 300) as=(110, 60, 1200)
$(A_{web}^R, A_{as}^R, A_{db}^R)$	(99.81%, 98.6%, 98.2%) (these values are calculated from appropriate MTBF and MTTR values)
h_{web}	0.90
h_{app}	0.85
h_{db}	1.00

Table D.9: Service demand in milliseconds in all tiers

Tier	CBMG state							
	h	b	s	g	p	r	a	d
Web tier	50	20	30	70	50	30	40	30
Application tier	0	30	40	35	150	70	40	25
Database tier	0	40	50	65	60	150	40	30

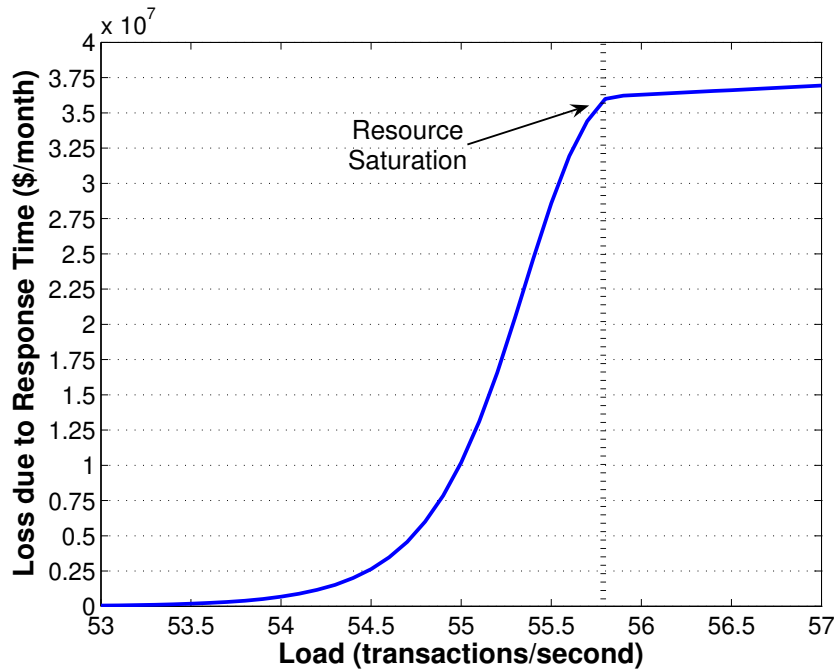


Figure D.8: Effect of Load on Loss

respectively. Figure D.7 shows how loss due to service unavailability is affected by the redundancy of web server machines while the rest of the infrastructure remains unchanged. As can readily be seen, upgrading from 24 machine through to 29 machines decreases loss due to unavailability while infrastructure cost increases. Figure D.8 shows how the increase in loss as response time – and thus defection – varies with increasing load. Note that after the resource saturation (the database machines saturate first in this scenario), response time loss increases linearly since ΔX^T is a linear function of γ .

D.4.2 Comparing business-oriented design with ad hoc design

An infrastructure designer typically designs site infrastructure without business considerations. The methodology aims to minimize cost while maintaining reasonable (or targeted) service availability and response time. In the following discussion, a particular design is represented by the 6-tuple $(n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db})$ which indicates the total number of machines in each tier and the number of load-balanced machines in each of the three tiers. The cheapest infrastructure is clearly $(n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db}) = (1, 1, 1, 1, 1, 1)$. This design cannot handle the applied load (average response time grows without bound) due to

saturation of the servers in all tiers. The cheapest design that can handle the load (keeping all server utilizations below 1) is $(n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db})=(22,3,2,20,2,1)$. There are 22 servers in the web tier, 3 in the application tier and 2 servers in the database tier. This design has a monthly cost of \$103692, average response time of 4.08 s. and service availability of 99.68%. Since this value for response time is typically considered inadequate, the designer may add a single load-balanced server in the web and application tiers (the most saturated), yielding a design with infrastructure $(22,3,2,21,3,1)$, monthly cost of \$105612, average response time of 0.60 s. and service availability of 92.90%. This seems to solve the response time problem but exacerbates the availability, since fewer standby servers are available. The designer now adds a standby server in the web and application tiers, yielding 99.59% availability for the design $(23,4,2,21,3,1)$. Assuming that this value of availability is considered adequate – and one may ask how the designer is supposed to choose what value to aim for – then the designer may rest. This is not an optimal design, as will be shown shortly.

None of the above design decisions take business loss into account. Let us examine the values for loss for the above designs as well as for the optimal design – that which minimizes the cost plus loss as shown in section D.2 (see Table D.10). In this table, each line represents a point in the infrastructure design space; the first column indicates the design being considered; the second column is the infrastructure cost; the third column is the business loss (in \$) due to customer defections; the fourth is the business loss due to service unavailability; the fifth is the total monthly financial commitment (cost plus business losses); finally, the last column shows how much the business loses by adopting that particular infrastructure compared to the optimal one (as given in the last line). All financial figures are monthly values.

The average response time for the optimal design $(26,5,3,22,3,1)$ is 0.40 s., availability is 99.99%. It has lowest overall financial impact (cost+loss), and the last column clearly shows the high cost of designing in an ad hoc fashion: a wrong choice can cost millions of dollars per month. It can also be observed that an over-design can quite easily be suboptimal. In this case, business loss can be very low, but as a result of high cost design.

Table D.10: Comparing infrastructure designs

Infrastructure design	Cost (\$)	Response Time Loss (\$)	Unavailability Loss (\$)	Cost + Loss (\$)	Cost of choosing wrong (\$)
(22,3,2,20,2,1)	103,692	25,102,025	102,458	25,308,176	25,175,510
(22,3,2,21,3,1)	105,612	9,317	2,297,800	2,412,729	2,280,063
(23,4,2,21,3,1)	110,682	9,987	133,134	253,803	121,137
(26,5,3,22,3,1) (optimal)	130,977	55	1,633	132,666	0

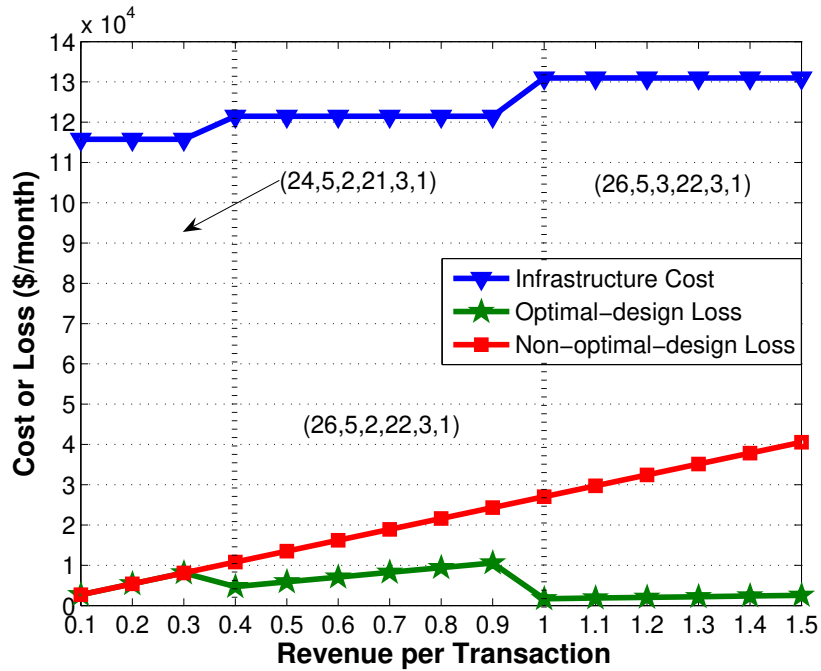


Figure D.9: Influence of BP importance on optimal design

D.4.3 The influence of business process importance

Another study concerns the influence of BP importance (revenue per transaction) on infrastructure design. Figure D.9 shows some results. Revenue per transaction (i.e., the BP importance) runs horizontally. The top curve shows infrastructure cost for the optimal design as revenue per transaction changes. One can see two transitions in optimal design: the first design is (24,5,2,21,3,1) and runs from $\phi = 0.1$ to $\phi = 0.4$; then, between $\phi = 0.4$ and $\phi = 1.0$ the optimal design changes to (26,5,2,22,3,1); finally, at $\phi = 1.0$, it goes to (26,5,3,22,3,1). Since these designs progressively add machines to the infrastructure, cost rises as depicted in the top curve. The bottom curve shows the business loss for the optimal design. One can see that design changes at $\phi = 0.4$ and $\phi = 1.0$ lower the business loss. Finally, the middle curve exhibits the situation for the business loss if the same design (24,5,2,21,3,1) were kept throughout: monthly loss increases constantly and reaches \$40,000/month. The difference between the bottom two curves captures the financial gain obtained by designing IT infrastructure from a business perspective. These results clearly show how important the aggregate business value of the business process itself is when designing IT infrastructure.

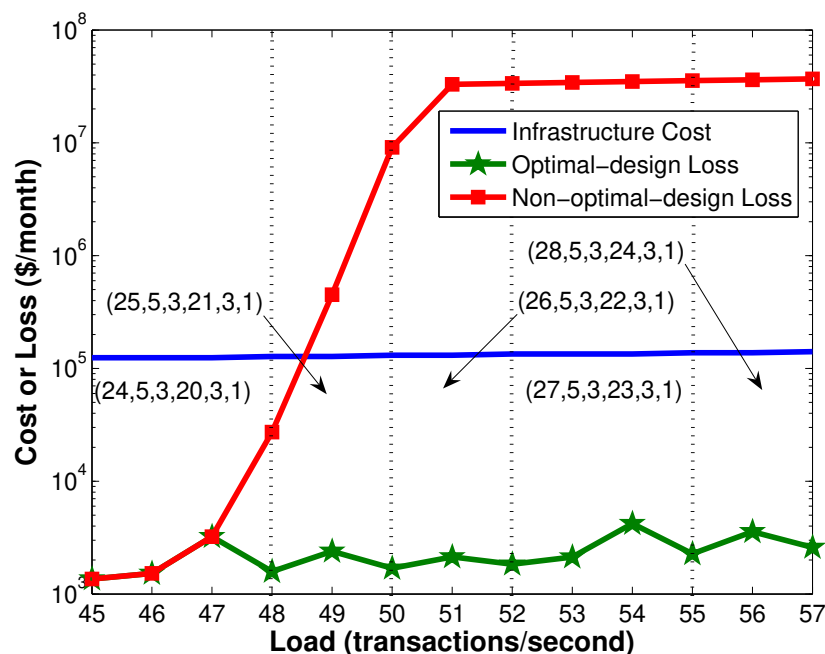


Figure D.10: Influence of load on optimal design

The next study varies the applied load. Figure D.10 shows that there is a considerable –

possibly huge – difference between the non-optimal design (the top curve) and the optimal one (the bottom curve) when the load increases.

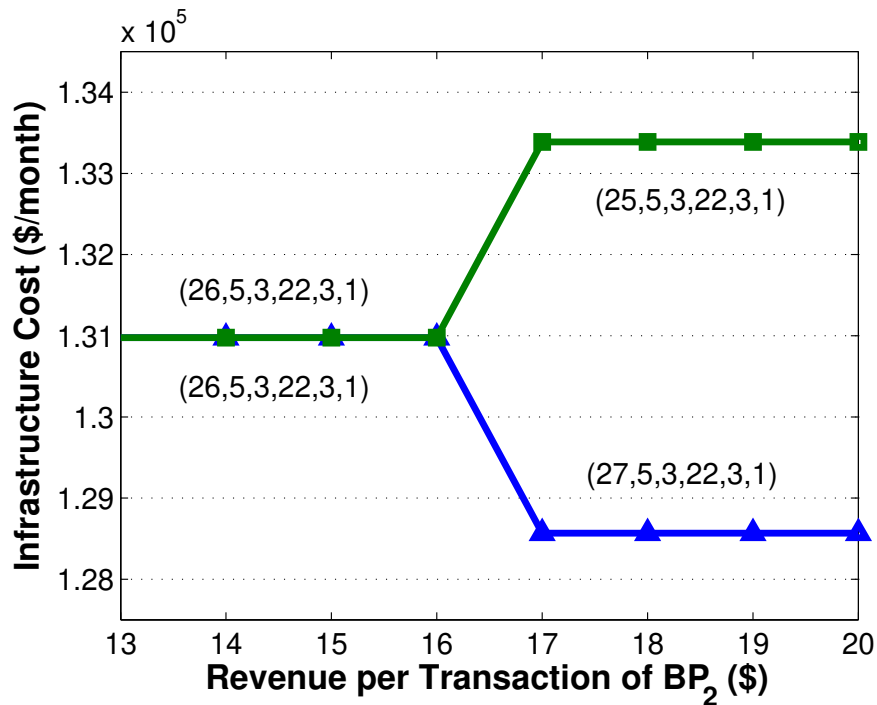


Figure D.11: Variation of Optimal Design due to Different BPs Importance

D.4.4 Provisioning two services

Business process importance can be examined from another perspective. Suppose the existence of a *second* BP, using a second IT service, similar to the one described in Table D.8, and assume that budget restrictions limit monthly provisioning costs for both services to \$262,000. Each service depends on distinct resource classes (that is, there are 2 resource classes of each type, one for each service). Figure D.11 shows how the resources are divided between the infrastructures of both services as the importance of BP *b2* increases while maintaining the importance of BP *b1* constant. As BP *b2* becomes more important, it soaks up a larger percentage of the infrastructure budget, as it should since it can cause greater loss. This is another clear example showing how infrastructure design can be heavily influenced by business considerations.

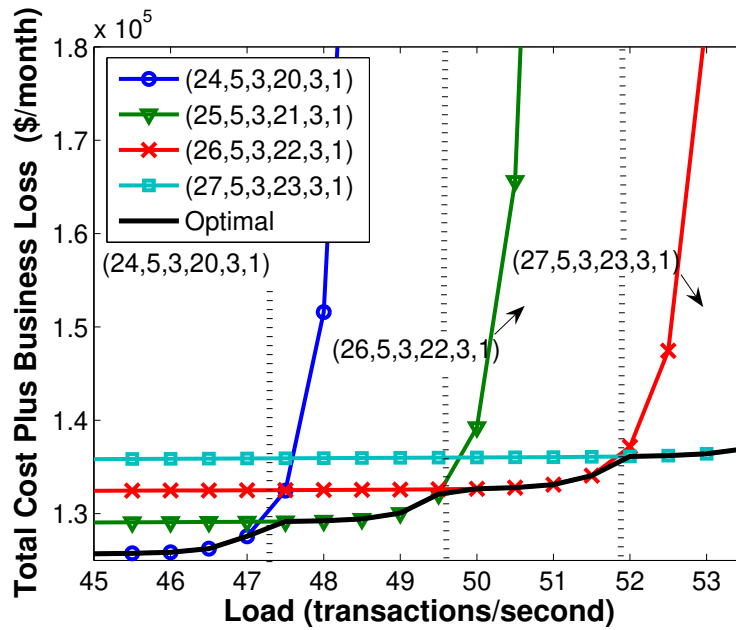


Figure D.12: Sensitivity of Total Cost Plus Loss due to Load

D.4.5 The effect of varying load on optimal design

Recall from the mathematical development that input load has a constant average (γ); in practice, this average typically changes slowly over time. Let us investigate the sensitivity of the optimal design, IT metrics and business metrics to time variations of the input load. Figure D.12 shows the total cost plus loss (i.e., $C(\Delta T) + L(\Delta T)$) as input load varies. The figure can be seen to be divided into four regions: the first design is (24,5,3,20,3,1) and is optimal for all load values in the leftmost region ($\gamma=45.0$ to 47.3); the second region ($\gamma=47.3$ to 49.6) has an optimal design of (25,5,3,21,3,1) with an additional web server; the third region ($\gamma=49.6$ to 51.9) has an optimal design of (26,5,3,22,3,1) with yet another web server; the rightmost region ($\gamma=51.9$ to 53.5) has an optimal design of (27,5,3,23,3,1) with a third additional web server. Five curves are shown in the figure; the first (blue, circle marker) shows cost plus loss when using the design that is optimal for the first region but for all values of load; similarly, the second curve (green, triangle marker) shows cost plus loss when using the design that is optimal for the second region; two more curves (cross and square markers) are shown for the optimal design in the remaining two regions. Finally, the heavy black curve simply follows the bottommost curve in any region and represents the optimal situation in all regions, using four different infrastructure designs, one for each region.

Several conclusions can be reached from this figure. First, an optimal design remains optimal for a range of input load. The width of the ranges lends some hope that a static infrastructure design may be optimal or close to optimal even in the presence of some variation in load. The second conclusion is that, in the presence of larger load variations, a particular infrastructure design quickly becomes suboptimal; an example is the leftmost optimal design (24,5,3,20,3,1) which quickly accumulates heavy losses at loads greater than $\gamma=47.3$. In this case, dynamic provisioning can be used to kick in – at appropriate times – either a better infrastructure configuration (scaling up) to reduce business losses or a cheaper configuration (scaling down) to reduce infrastructure costs, as is deemed appropriate. The third major conclusion is that the business impact model described in this paper appears to be useful as a mechanism for dynamic provisioning since it captures appropriate load transition points for reprovisioning using a business perspective. Further investigations need to be conducted concerning this point.

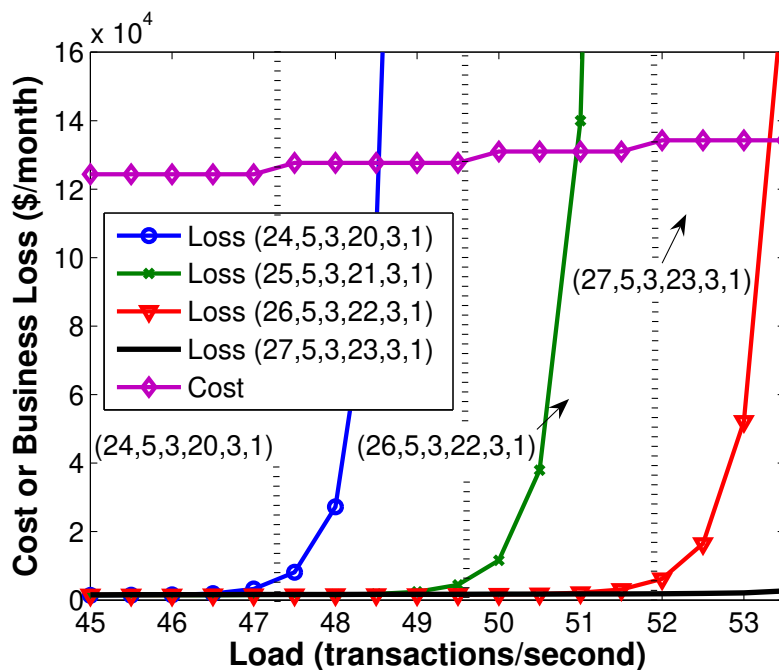


Figure D.13: Sensitivity of cost and loss due to load

Additional interesting details can be seen in Figure D.13 which shows cost and business loss separately for the first three data center designs described above. Costs clearly go up (from left to right) as designs use more resources, although the cost increase is more than offset by the reduction in loss provided by better designs.

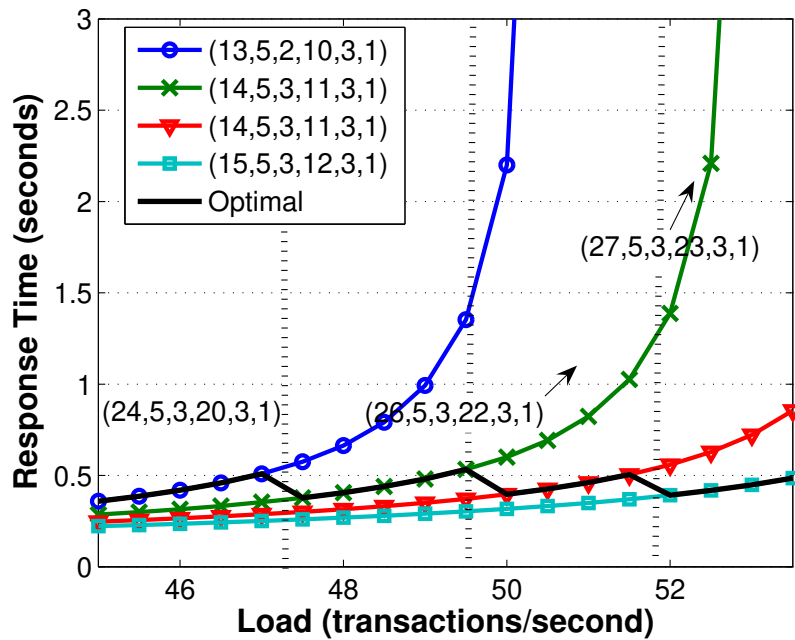


Figure D.14: Response time for various designs

Finally, Figure D.14 shows response time for the four designs as well as the optimal response time (heavy black line); by *optimal response time*, we mean the response time obtained by using the optimal design in each region. Since the load applied to the system varies with time, two situations can occur: in a static provisioning scenario, one can use the data shown in Figure D.14 to choose the "best" design over the expected load range. This will not be an optimal design for all load values but the approach provides the designer with a tool to evaluate the cost of over-designing to handle load surges. In an adaptive infrastructure scenario, on the other hand, a dynamic provisioning algorithm can be used to trigger infrastructure changes to keep the design optimal at all load levels. The dashed lines in Figure D.14 show where dynamic provisioning must trigger and the heavy black line shows the response time that is attained, a low value for all load levels.

D.4.6 Designing infrastructure for load surges

One more dimension will be explored in this subsection: designing infrastructure in the presence of load surges, as can occur, for example, before special events such as Mother's Day or end-of-year shopping season. First, two approaches to infrastructure design – cost-

oriented and business-oriented – are compared; this last considers adverse business impact as the main design metric. Whereas the conventional, cost-oriented approach only considers the cost dimension, the business-oriented approach takes both dimensions into account. For the cost-oriented approach, since there is no automatic way of choosing performance targets, the values for A_{MIN} – the minimum acceptable value for availability for service S – and T_{MAX}^w – the maximum acceptable value for the average response time for service S – were set at 99.99% and 1.5 seconds, respectively, typical values for service design. This is not required for business-oriented design, since performance metrics are *outputs* rather than *inputs* to the design process.

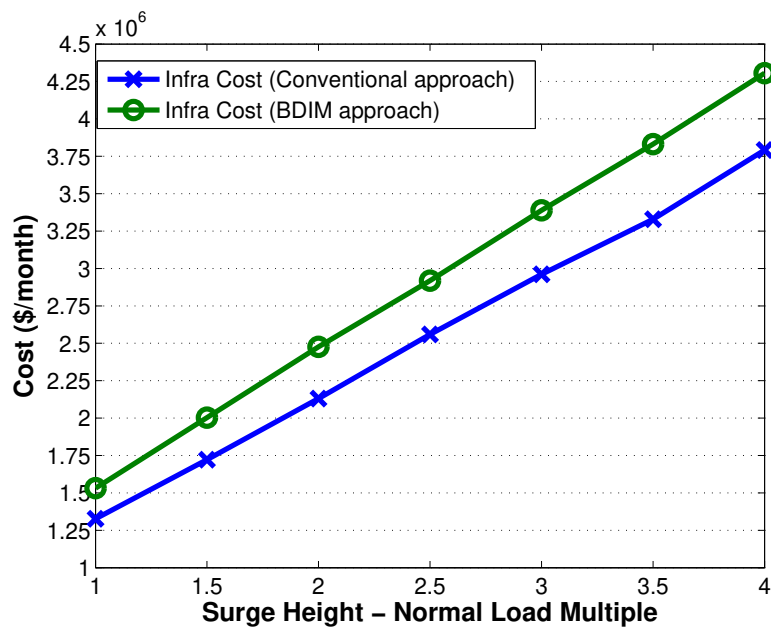


Figure D.15: Cost Dimension Comparison

Figure D.15 compares the optimal designs of both approaches but only depicts the cost dimension. Two load surges were considered: one in May (Mother’s Day) and the other in December (end-of-year season). Each demand surge lasts 1 month and load is n times the normal value (n is plotted on the horizontal axis). From this figure one can hastily conclude that the conventional method yields better (lower cost) results than the one proposed here – the average yearly cost using the conventional approach is \$2,546,000, while it is \$2,922,000 using the business perspective.

This hasty conclusion is erroneous and causes unnecessarily high financial outlays. The

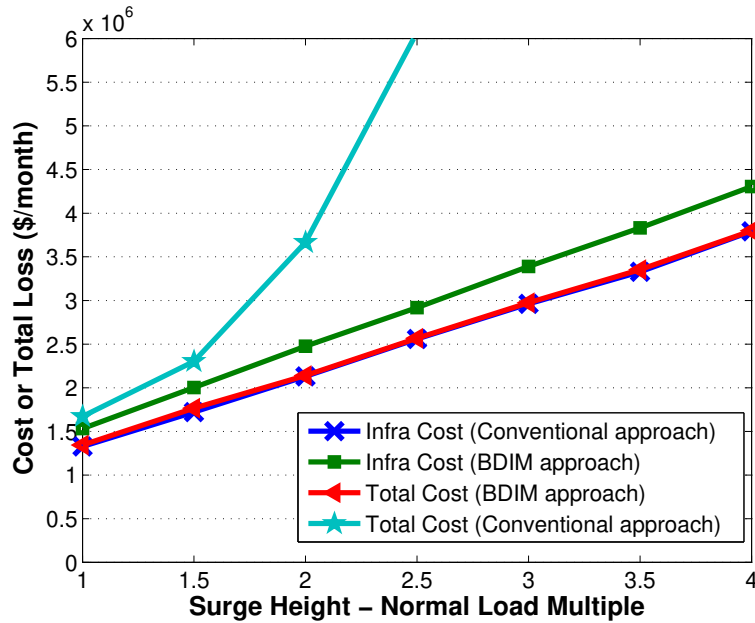


Figure D.16: Cost and BIM Dimensions Comparison

business losses incurred by IT failures and high response time have not been shown in the figure. Figure D.16 shows these losses. The Total Cost metric – infrastructure cost plus business losses – of the optimal designs found by the business-oriented approach is, on average, approximately 65% lower than the average Total Cost from the designs obtained by the conventional approach. For large surges, the difference in total cost is over 100%.

A sensitivity analysis is now undertaken; how do surge duration and height influence optimal design and business metrics? Figure D.17 depicts how optimal design is influenced by surge duration. In this scenario, there are two load surges with a total duration varying from 1 to 3 months (horizontal axis). A change in optimal design only occurs when the surges last 1.5 month: the optimal design changes from $(n_{web}, n_{as}, n_{db}, m_{web}, m_{as}, m_{db}) = (47, 8, 3, 42, 5, 1)$ to $(48, 8, 3, 43, 5, 1)$. Thus the optimal design is relatively insensitive to changes in surge durations, since business losses increase only linearly with surge duration, as can be gathered from Equation D.2. Note that although this change causes an increment in infrastructure cost (top curve), it also leads to greater loss reduction (bottom curve). This fact can be seen when comparing the values shown by the middle curve – it shows how business loss grows when the original $(47, 8, 3, 42, 5, 1)$ design is kept throughout – with the values of the bottom curve – the loss for the optimal design. The difference between both approaches is \$77,020/year,

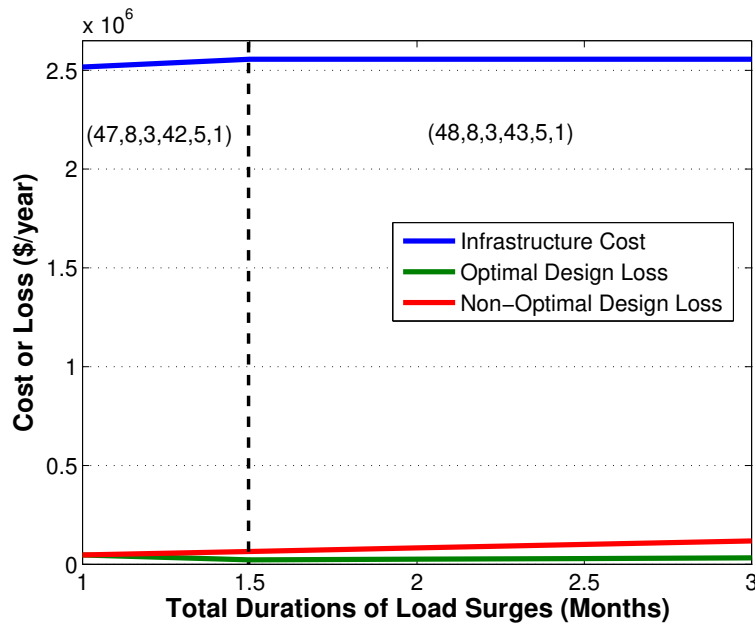


Figure D.17: Influence of Surge Duration on Optimal Design

when the total duration of load surges is 2 months, \$91,029/year, when the total duration of load surges is 2.5 months, and \$106,041/year for a 3-month total surge duration.

A final conclusion about optimal design sensitivity is exemplified by Figure D.18 which shows the sensitivity of optimal design to surge height variations. Again, in this scenario, there are two 1-month load surges with height varying from 1 to 4 times the normal load height (horizontal axis). The optimal design is profoundly affected by load surge height since surges saturate resources and lead to highly non-linear behavior in the loss metric. This fact leads to an appreciable difference between the optimal (bottom curve) and the non-optimal designs (top curve).

D.5 Related Work

The design approach proposed in this paper relates to other independent work in three IT R&D areas: BDIM, infrastructure design and SLA design.

A BDIM method for choosing values for SLOs of an SLA that minimize lost revenue due to an unavailable or a slow e-commerce site is introduced in [SMM⁺05c]. The method is then applied to the capacity planning of an e-commerce site when subject to invariant work-

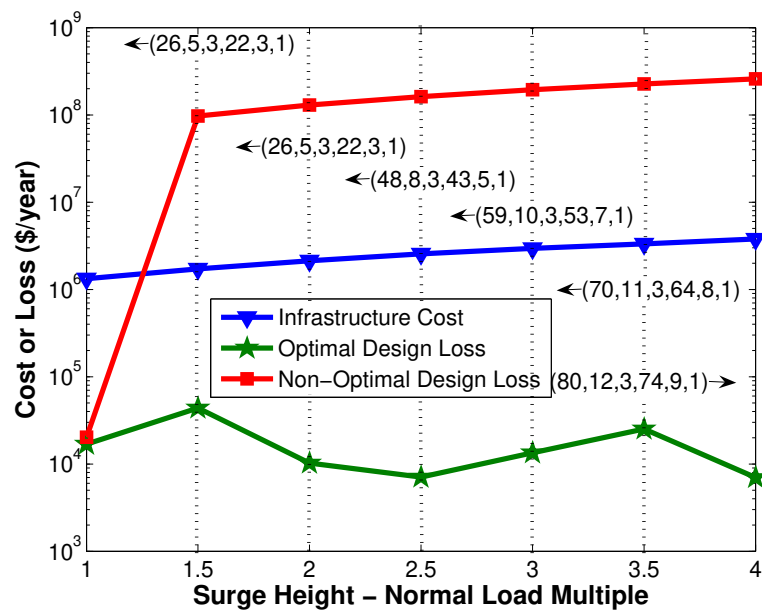


Figure D.18: Influence of Surge Height on Optimal Design

loads in [SMM⁺06b]. Comparison of the resulting BDIM designs against those provided by conventional, technical measure-based methods indicate financial advantage of the proposed BDIM approach. Variations in workload to the e-commerce site, in the form of load surges typical of periods of special sales promotions which occur along the year (Mother's day and end-of-year buying season being two such periods), are considered in [SMM06a]. In terms of expected financial gains, the BDIM method is again shown to prevail over conventional ones. In fact, references [SMM⁺05c] [SMM⁺06b] [SMM06a] present earlier versions of the present work. This paper consolidates and generalizes the results obtained there and extends the models to consider very large scale infrastructures through a cache mechanism.

BDIM has been attracting mounting research efforts recently. Results from these efforts have contributed to different IT management domains. The application of BDIM concepts to ITIL processes has yielded notable results for incident prioritization [BS04a] and change management [KHW⁺04]. The work here adds to these results since the proposed approach is applicable to ITIL capacity management.

The studies on capacity management in [MBD01] [JST03] [AFT04] [AF02] use no business-oriented metrics other than cost of infrastructure and treat availability separately from response time and throughput. The work in [JST03] describes a tool to automate the

capacity planning process in order to meet performance targets (such as the number of transactions per second) and downtime limits. The methodologies in [AFT04] [AF02] design minimum-cost infrastructures given a set of requirements. None of these references however, consider response time directly. The approach in [MBD01] carries out a cost-oriented capacity planning optimization based on IT-level metrics only. The problem of capacity planning from a business perspective, using business metrics, was not addressed in these studies. The approach in this paper considers business metrics and manages to join reliability and response time in a single optimization function, through the use of a monetized business impact function. Further, the approach also derives response time distributions rather than only mean values, as is standard in the field. Response time distributions allow one to estimate the probability of customer defection and, hence, the reduction in throughput when response time rises.

Customer defection avoidance is an interesting BDIM goal and has been considered in [MAFM00] [AGL⁺04]. As we have also done here, the authors of these two references consider (potential loss of) revenue and use a Customer Behavior Model Graph (CBMG) with parameters set from http logs, to describe customer behavior at e-commerce sites. Their work consists in defining resource management policies (in terms of traffic handling priorities) for the servers in order to keep customers satisfied with response time levels. The work in [MAFM00] attributes higher service priority to customers with higher shopping cart values and presents a solution obtained through simulation for the case of an electronic bookstore. Also by simulation, the optimizing algorithm in [AGL⁺04] searches over the space of possible IT infrastructure parameters (such as policies that attribute traffic handling priorities) and was used to tune the operation of a stock-trading web site. The site was built with IBM technology incorporating the simulator and it was used to demonstrate gains of the proposed BDIM solution when multiple business transactions were processed. The work in these two references, however, does not address capacity planning (server capacity is never changed) and the results are not meant for the formal design of SLAs. The models are solved by simulation whereas our work is entirely analytical.

The need for more formal methods in dealing with SLAs is argued for in [TT05a]. Negotiating an SLA is not a trivial task either for in-house or outsourced IT services, especially because it is difficult to relate the technical metrics specified in the SLA to the business it-

self. SLA modeling is the focus of several authors. Some of them concentrate on defining a language for the implementation and verification of the objectives and conditions in the contract [SDM01] [LSE03]. Schmidt [Sch00] combines service contracts to workflow concepts. Workflow concepts lead to non-ambiguous contract specification and facilitate management of services by the customer. Transaction Cost Theory and Incomplete Contract Theory were used in [BR03] to study outsourced SLA contracts. The study offers insight into a) the identification of scenarios for inclusion in contracts (this is difficult due to the incompleteness of the contract); b) cost analysis of outsourcing versus internal service provision alternatives and, c) costs of changing the provider. The paper considers the business measure of cost but not in a quantitative manner and its scope does not include defining SLOs. Our interest here is in setting SLO clauses in SLAs to reduce business loss. Our work complements all these references as support for SLA specification from a business perspective.

Although references [ASBB04] [BCL⁺04] [SB04] do not address the SLA design problem directly, they investigate SLA linkage to business metrics. The paper [ASBB04] proposes reviewing SLAs and the enterprise IT balanced scorecard in order to keep the cost of eliciting knowledge about the business value of the service low (information on SLAs may be obtained from the Configuration Management Data Base – CMDB as defined in the ITIL Configuration Management process). SLA penalties in an e-Business context with autonomic computing is considered in [BCL⁺04]: task scheduling is incremented by taking into account some business metrics when performing scheduling decisions. Management by Contract [SB04] investigates when it is better to violate an SLA rather than to keep compliance. Instead of analyzing pure financial loss as done in our work, a utility function is used to estimate risk associated with available options.

Performance optimization of a shared utility computing environment which supports multiple third party-applications subject to SLA performance targets – in terms of maximum throughput and minimum response times are considered in [AAA06]. The resulting model is solved using mathematical and simulation techniques. Performance monitoring at run-time in a Utility Computing setting is discussed by Farrel [Far03] but does not focus on business measures such as profit, costs or losses. The management of IT resources to maximize provider SLA revenue is addressed in [LSW01] by using a queuing theory model (dynamic workload characteristics were not modeled) and in [DHP01] by applying feedback

loops as control mechanisms for the number of users accessing an IT service (which is subject to an SLA). The emphasis in [DHP01] is on automated SLA enforcement, instead of SLA negotiation; revenue is achieved as a percentage of completed transactions and penalties are paid in the form of rebates to customers who experience bad QoS. The business being modeled in these last references is that of an independent service provider. Our work applies to a more generic business, where revenue is generated by sales of goods or services and it aims at assisting negotiation of SLA clauses.

D.6 Conclusions

Summary. This paper has investigated the design of IT infrastructure from a business perspective. At this stage of investigation, ‘infrastructure’ means the server farm used to provision IT services. The study is part of a new area of research called *Business-Driven IT Management* (BDIM). BDIM aims to reexamine IT management processes by considering how management decisions can be taken by using business metrics – as opposed to technical metrics – to compare alternative courses of action. This paper examines the capacity planning IT management process as well as the Service Level Management (SLM) process. BDIM approaches use a *business impact model* to gauge the effects of IT decisions on the business. In this study, the impact model has estimated the financial business loss accrued from IT component failure and performance degradation (high response time) of IT services. Since capacity planning is performed a priori, that is, when service provisioning has not yet been done, business loss was evaluated mathematically through queuing theory and reliability theory models. The full model is layered and includes entities such as IT components, business processes and business units generating revenue. An optimization problem was posed to find the best infrastructure design leading to the lowest sum of provisioning costs plus business losses over a time period.

Conclusions. Many design scenarios were investigated using the proposed models and several conclusions can be drawn from the results. First, business loss has two causes (service unavailability and service performance degradations), but neither one is always a dominant cause; scenarios can easily be built where one or the other of these two causes dominate. However, one can conclude that, whenever resources are not saturated, service unavailability

tends to be the more crucial source of problems.

Second, business-oriented design can be superior to ad hoc design or to approaches that minimize cost, in the sense that it can provide a design that has lower overall financial impact on the business. The difference can easily reach hundreds of thousands of dollars a month for medium-sized infrastructures.

Third, an important conclusion is that the approach solves a current problem in SLA design: how is one to choose SLA parameters such as availability and response time targets? The answer is to *not* choose them a priori but to simply *calculate* them from the final optimal design arrived at through business considerations. By definition, a design that minimizes business impact is the correct point in design space, regardless of the particular availability and response time measures provided by the design.

Fourth, if one measures the importance of a business process (BP) by the revenue it generates, then BP importance heavily affects design choices; the best design for a more important BP can be significantly different from a design for a less important BP.

Fifth, when several services are provisioned at the same time under a single optimization problem, the optimal design will allocate better infrastructure to services that yield more revenue, so as to minimize overall business impact (cost+loss).

Sixth, an optimal design stays optimal for a relatively small range of input load. Changing input load by a few percent (less than 5%) is enough to make a particular design non-optimal. This indicates the importance of dynamic provisioning. On the positive side, the same approach used here can provide trigger points for infrastructure reprovisioning decisions. Combined with techniques to estimate future load, one can calculate the new design to be used at these reprovisioning trigger points.

Finally, when investigating the effect of load surges, the main conclusions are that the optimal design is relatively insensitive to changes in surge *durations*, since business losses increase only linearly with surge duration; on the other hand, optimal designs are profoundly affected by load surge *heights* since they easily saturate resources in a highly nonlinear fashion.

Contributions. The work proposed here includes a number of crucial departures from past effort in the area of infrastructure provisioning. One may mention the following: the approach is based on business considerations and is thus part of a new research area called

BDIM. Second, the models successfully *combine* several performance measures such as availability and response time in a common performance model. The various metrics are tied through a business impact model, one of the main contributions of the present work; most studies consider only one of the metrics in isolation. Finally, whereas most performance evaluation work concentrates on response time averages, the mathematical development presented here develops expressions for the full response time distribution, enabling one to assess the effects of customer defections.

Future work. In the future, one may investigate new impact models applicable to business processes other than e-commerce (say, manufacturing, CRM, etc.); additionally, more complete models that include the network and other components outside the data center may be considered. Third, a fuller study of the use of business impact models in adaptive environments can be undertaken; this will expand the initial comments given here concerning dynamic provisioning. As an example, one may examine feedback controller design for dynamic provisioning based on business metrics. As a further effort, business impact models such as the one discussed here can be used in other contexts to solve other IT-management-related problems such as incident management, change management, etc.

Acknowledgments

We would like to acknowledge and thank the Bottom Line Project team (<http://www.bottomlineproject.com/>). This work was developed in collaboration with HP Brazil R&D.

Appendix E

Business-Driven Service Level Agreement Negotiation and Service Provisioning

Filipe Marques, Jacques Sauvé, Antão Moura

Departamento de Sistemas e Computação

Universidade Federal de Campina Grande

Campina Grande, Brazil {jacques,filipetm,antao}@dsc.ufcg.edu.br

Abstract: A business-driven approach to designing and negotiating Service Level Agreements (SLAs) in an e-commerce environment is proposed. In contrast to conventional SLA negotiation approaches, the one proposed better captures the linkage between service provider and service client by considering the negative business impact (business loss), originated from IT infrastructure failures and performance degradation and factors such knowledge in the SLA itself. A complete example scenario shows the value of the proposed approach. A main conclusion is that the SLA established using the business-driven perspective is superior to the one using a conventional approach since both service provider and client obtain higher profit. ¹

¹Submetido ao 2006 Conference on Measurement and Simulation of Computer and Telecommunication Systems (MASCOTS 2006), em Abril de 2006. / Submitted for publishing in the 2006 Conference on Measurement and Simulation of Computer and Telecommunication Systems (MASCOTS 2006), in April of 2006.

E.1 Introduction

Information Technology (IT) services are used to support business processes in today's enterprises to an extent that makes many IT services mission-critical. This high level of criticality has significantly raised the visibility of IT services and these must be designed and provisioned ever more carefully to support business needs. To that effect, the needs of the business must somehow be captured and linked to the IT world so that adequate service be designed and deployed. Now, how is one to express business needs in a way understandable to the technical IT world? In the last few years, this has been done through Service Level Agreements (SLAs) that contain Service Level Objectives (SLOs) for technical service metrics such as availability, throughput and response time. The reason that IT metrics are used for that purpose is that i) they are easily understood by IT personnel; and ii) they are easily measured and are thus more trusted than, say, human opinions on the quality of service. However, as was depicted in [TT05b], setting SLO values (quality thresholds) is not a straightforward task and has been done in an ad hoc, "finger-in-the-air" manner. The reason is that while technical metrics are understandable to technical people, they are not readily related to business results and are thus next to meaningless to business people. Just how much availability do we need for a particular service? 99.9%? 99.95%? Why? How is a business person supposed to choose adequate values to reflect business needs? The same goes for other technical metrics such as response time and throughput.

In [SMM⁺05c], we have shown how to use a business-driven approach to design SLAs, contributing to a new area of research called Business-Driven IT Management (BDIM) [SMM⁺05c; SB04; MBC04]. The term "designing SLAs" here means negotiating values for SLA parameters and also designing the infrastructure to meet the service objectives. In summary, new models estimate the *business loss* suffered by a business due to imperfect infrastructure that may fail or provide slow service. The business loss is used in an overall model to design the most cost-effective infrastructure that optimizes the sum of infrastructure cost and business losses. One of the novelties of this work is to use *business metrics*, rather than technical IT metrics to express linkage between IT and the business; this is what characterizes the approach as being BDIM.

However, this approach optimizes business results from the service client's point of view

and is thus only adequate for in-house services – those that have not been outsourced. In that case, the same enterprise both pays for the infrastructure and suffers the business losses so that it makes sense to sum the *two* values and minimize it. Also, when the service is not outsourced, there are no penalties specified in the SLA. In the very common scenario where service is outsourced to a service provider, two business results must be considered, penalties are a common SLA clause and the previous approach fails. Since the infrastructure is under the control of the service provider, it does not make sense for this infrastructure to be designed to optimize the *client's* business results; the provider wants to optimize its own business results and must take penalties into account.

The present paper thus seeks to extend models and approaches to the SLA design problem to consider outsourced services but maintaining the business perspective captured by recent models. To that end, one must properly model the linkage between IT and the business that is captured by an SLA. The model must maintain SLO thresholds but must factor in reward and penalty clauses common in SLAs underpinning outsourced services. The approach we describe here yields a multidimensional design problem, some of whose dimensions will be explored here. The particular dimensions explored are:

1. How to choose the SLA parameters "minimum service availability" and "maximum mean response time" (or percentiles of the response time distribution);
2. How to design a server farm to provision the service.

The remainder of this paper is organized as follows. An overview of related work is given in section E.2. Section E.3 describes the SLA negotiation problem from a conventional approach. The SLA negotiation problem from a business-driven perspective is presented and formalized in section E.4 while section E.5 applies the proposed approach to an example scenario. Finally, section E.6 summarizes our approach, offers conclusions and discusses next steps.

E.2 Related Work

As discussed in the introduction, the work in [SMM⁺05c] is closely related to the present one; reference [SMM⁺05c] proposed a formal method useful to choose optimal SLO values

for *in-house services*, whereas we here consider the completely different scenario of outsourced services for which the approach in [SMM⁺05c] fails. SLAs for outsourced services have also been studied in [BR03] using Transaction Cost Theory and Incomplete Contract Theory on six case studies and interviews with contract managers and legal experts. The results are three-fold: identification of possible future scenarios for inclusion in contracts (this is difficult due to the incompleteness of the contract); cost analysis concerning the decision about whether or not to outsource a service and the impact of changing the service provider. This study, however, does not quantify costs nor does it intend to define SLOs values.

The particular models for service availability and response time distribution used here are from [SMM⁺05b]. They are briefly summarized here for completeness and legibility.

Although Management by Contract [SB04] does not address the SLA design problem, it is very close to our work. It investigates when it is better to violate an SLA rather than to keep compliance. A utility function is used to estimate the losses associated with both options. Furthermore, reference [DHP01] attempts to maximize profit based on IT-level feedback loops. The focus is automated SLA enforcement, instead of SLA negotiation. Profit is defined as the revenue as a percentage of completed transactions minus rebates to customers who experience bad QoS. Analogously, [AAA06] tries to increase revenue through the reduction of user defection incurred by site performance degradation; the context is web server resource management.

A very interesting study that emphasizes the need for more formal methods in dealing with SLAs is shown in [TT05b]. Suitably negotiating an SLA is not a trivial task either for in-house or outsourced IT services, especially because it is extremely difficult to relate the technical metrics specified in the SLA to the business itself.

An IT service or resource provider may do BDIM-based dynamic allocation of server resources in order to maximize client revenue and hence, its own bonus commission – in fact, such an "autonomic self-optimization according to business objectives" web site scenario is the focus in [MAD04]. Reference [AAA06] considers outsourced applications subject to performance requirements, such as maximum throughput and minimum response time, by presenting optimization policies for a shared utility computing environment. In this work mathematical and simulation techniques are used to solve the proposed model. Also, [BCL⁺03] investigates SLA penalties in an e-Business context with autonomic computing. It improves

the task scheduling problem by adding a business perspective, that is, by taking into account some business metrics when performing scheduling decisions. Note that the main concern here is not SLA negotiation and only the provider side is considered.

The work presented in [Sch00] merges workflow concepts and service contracts. Basically, the contract is based on customer's business processes, for which the adoption of workflow provides a clear specification of the contract and instructions for usage and management of services by the customer.

All these papers consider business performance either from the provider's or from the client's point-of-view. By contrast, our work factors in both the provider and the client perspectives.

E.3 Conventional SLA Negotiation Approach

In this section we formalize the SLA negotiation problem as it has been traditionally approached (Figure E.1). The terminology used includes a service *provider* offering a service on behalf of a service *client* and *customers* that actually use the service. An SLA is established between provider and client. The service provider may or may not rely on a resource provider and this is invisible to the service client; for our purposes, it is quite adequate to think of the service provider as also encapsulating the resource provider.

Informally, the conventional approach consists of maximizing the provider profit, or conversely minimizing the infrastructure cost incurred by the service provider in order to satisfy the service requirements specified by the service client. This section briefly describes the technical and business metrics applied to provider and client, some of which are negotiable items in an SLA (response time, availability, service cost, and SLA penalty). The analytical model adopted here, the technical metrics, the service cost and the client loss derived from it have previously been described in [SMM⁺05b] but are here recast to clearly identify and segregate the provider perspective from the client perspective.

E.3.1 IT Infrastructure Basics

This subsection delineates the IT infrastructure model. For the purpose of making the model easier to understand, a single IT service S is considered. Extending the model to multiple

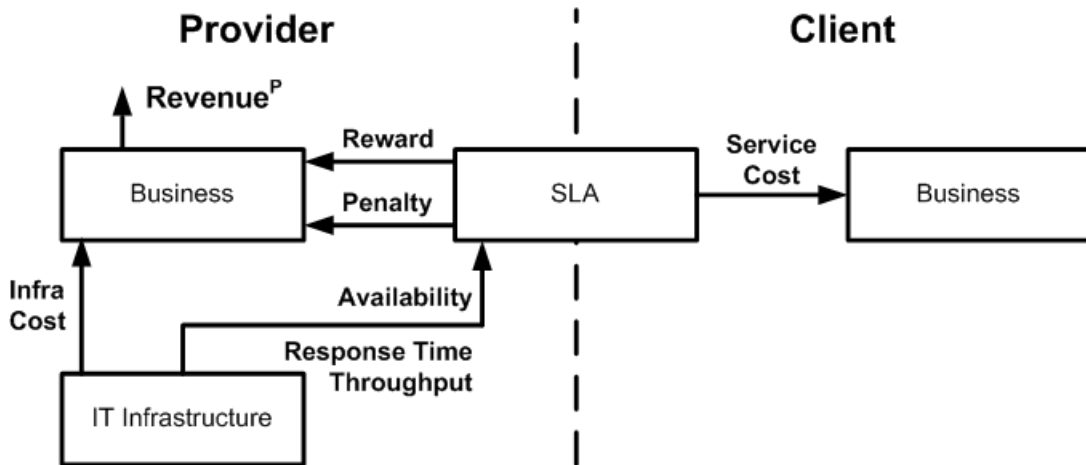


Figure E.1: Conventional SLA Negotiation Approach

services is straightforward.

Service S uses a set RC of IT resource classes. Typically, a resource class corresponds to an infrastructure tier. Database and web tiers are examples of resource classes. In its turn, resource class RC_j is composed of a cluster of n_j identical IT resources operating in fail-over mode. Of this total, m_j resources are load-balanced in order to deal with incoming load and offer acceptable response time, and $n_j - m_j$ resources are spares running in standby mode to offer better service availability. A given IT resource $R_j \in RC_j$ is made up of a set $P_j = \{P_{j,1}, \dots, P_{j,k}, \dots\}$ of IT components. Operating system software, server hardware and application server middleware are examples of IT components that can be part of a resource. In the model, the IT resource will also fail if one or more of its IT components fail.

Service Provider Side

The next subsections describe how to evaluate technical and business metrics involved in the conventional SLA negotiation approach from the provider's point of view. For clarity, we are assuming a single service client; the model easily extends to multiple service clients.

E.3.2 Defining and evaluating IT metrics

The two most common IT metrics used in SLAs are the object of this section: they are service availability and service response time. We show how they are defined and indicate how their values can be calculated. This will be used further on to formalize several SLA negotiation options. These metrics are considered to be "provider-side metrics" since they are defined

by the infrastructure made available by the provider and are thus under his control.

Service Availability

IT components can suffer failures that produce unavailability of the services relying upon them. This subsection shows how to evaluate the availability, A , of service S . Standard reliability theory [Tri82] uses Mean-Time-to-Repair (MTTR) and Mean-Time-Between-Failures (MTBF) values to calculate the availability, A_j^R , of individual IT resources:

$$A_j^R = \prod_{k \in P_j} \left[\frac{mtbf_{j,k}}{(mtbf_{j,k} + mttr_{j,k})} \right]$$

Values for MTBF can be obtained from IT resource specifications or historical logs, while values for MTTR typically depend on the terms of the service contract available (gold, silver, etc.). Once A_j^R has been calculated for all resources R_j , one combines IT resource availability to obtain the IT resource class availability, A_j . Since the clusters being used in a IT resource class will be available and ready to handle the imposed load when at least m_j resources are available for load-balancing, resource class availability, A_j , uses "m-out-of-n reliability" [Tri82]. Finally, for a service to be available, we need all required component classes (all tiers) to be available, that is:

$$A = \prod_{j \in RC} A_j$$

Service Response Time

Several metrics related to response time are typically used in SLAs. The two most important are the mean response time and some percentile of response time distribution. For example, an SLA may state that "95% of requests must be serviced within 3 seconds". In order to capture such SLA terms, we now need to find not merely the mean response time, as in most other performance evaluation work, but the whole distribution. As we shall see later on, having the distribution is also key to establishing business metrics caused by customer defection.

The technique used to estimate response time distribution has been reported elsewhere [SMM⁺05b] and full details are not repeated here. We limit ourselves to a brief description of the queuing-theoretic approach.

With the purpose of estimating response time distribution, $ResponseTimeDistribution(x)$, and also its mean, \bar{T} , the load applied to the IT resources must be modeled. The load model adopted is based on the Customer Behavior Model Graph (CBMG) [MV00] that permits a representation of how a given session initiated by a final customer applies load on the IT infrastructure.

A CBMG includes a set of states with associated transition probabilities; to make things more concrete, one can view a CBMG state as an e-commerce site page being visited. Some CBMG states generate revenue for the client; this is the case, for example, of an e-commerce site page where the customer pays for items in a shopping cart. There are two kinds of sessions: revenue-generating (RG) sessions and non-revenue-generating (NRG) sessions. RG sessions generate revenue since they visit a revenue-generating CBMG state with non-zero probability, while NRG sessions do not visit such states and do not generate any revenue (the client may be browsing without buying or may buy but never reach the pay page); the fraction of RG sessions is defined by f . In order to find $ResponseTimeDistribution(x)$, IT services are modeled by means of a multi-class open queuing model; each CBMG class represents a queuing model class [Kle76b]. The use of an open queuing model is justified in the presence of a large customer population, a common situation in an e-commerce context. Now, in order to find $ResponseTimeDistribution(x)$, one needs to find the distribution of response time $T_r(x)$ in all CBMG states, r , and combine them. Since a transaction can possibly use resources from all resource classes – that is, since a transaction may need service from all infrastructure tiers – response time for a transaction in state r is the sum of $|RC|$ random variables. The distribution of each of these random variables (the response time at a particular IT resource) can be found by means of Laplace transforms.

E.3.3 Defining and evaluating business metrics

SLA negotiation involves financial matters. These are considered in the present section.

Service Cost

Each IT resource (a server in the server farm) can be either in standby – or inactive – mode or in load-balanced mode. Being inactive implies that the IT resource has a lower cost rate

than when active, since, for example, no license fees or electricity costs apply [JST03]. Remembering that a given IT resource $R_j \in RC_j$ is made up of a set $P_j = \{P_{j,1}, \dots, P_{j,k}, \dots\}$ of IT components, one can find the cost of providing the service by considering the cost rates of individual components.

Thus, if $c_{j,k}^{active}$ is the active cost rate and $c_{j,k}^{standby}$ the standby cost rate of component $P_{j,k}$, one can calculate the IT service cost seen by the provider, $Cost^P(E)$, over a time period E as follows:

$$Cost^P(E) = E \cdot \sum_{j=1}^{|RC|} \left(\sum_{l=1}^{m_j} \sum_{k=1}^{|P_j|} c_{j,k}^{active} + \sum_{l=1}^{n_j-m_j} \sum_{k=1}^{|P_j|} c_{j,k}^{standby} \right) \quad (E.1)$$

In the above equation, j runs over resource classes, l runs over resources and k runs over components.

In estimating values for $c_{j,k}^{active}$ and $c_{j,k}^{standby}$, one must factor in all costs involved, not only the acquisition cost; this is typically done by dividing the expected Total Cost of Ownership (TCO) by the infrastructure amortization period.

Service Price

The amount to be paid by the service client to obtain the service is here modeled by applying a fixed factor, a , over the service cost (as defined in subsection E.3.3). Thus, the service price, $Price(E)$, over time period E is $Price(E) = Cost^P(E) \cdot a$. The value for a depends on the service provider pricing policies and represents the desired markup.

Service Provider Revenue

Service provider revenue can originate from the service price paid by the client, but also from alternative sources; it can also be subject to penalties as defined in the SLA. SLA rewards are examples of an alternative service provider revenue source; such rewards may be fixed or may be variable, depending, for example, on transaction load successfully processed or even on client revenue or profit. It is important to observe that the present work intends neither to specify optimal reward values nor to exhaust all service provider revenue alternatives. Rather, we intend to show the multidimensional "SLA negotiation space" that exists. For our purposes, service provider revenue over a time period E , $Revenue^P(E)$, can be expressed

as follows:

$$\begin{aligned} Revenue^P(E) = Price(E) + Bonus(Profit^C(E)) \\ - Penalty(E) \end{aligned} \quad (E.2)$$

where $Profit^C(E)$ represents the service client profit, $Bonus(Profit^C(E))$ represents an alternative, reward-based service provider revenue source and $Penalty(E)$ represents the SLA penalty function. Penalties are tied to the IT metrics: whenever IT metrics such as availability and response time do not meet targets specified in the SLA, penalties are incurred.

Observe that linkage between provider and client is made through the SLA penalty function and the alternative service provider revenue source. The SLA penalty function is usually defined in an ad hoc fashion as emphasized in [TT05b]. On the other hand, the alternative provider revenue source is subject to negotiation between provider and client and may sometimes not apply. It may be advantageous to both provider and client to negotiate a lower fixed price in order for the provider to receive a larger variable revenue. We do not consider these negotiation dimensions in this paper, although they are part of the negotiation space.

Service Provider Profit

Service provider profit is simply its total revenue minus costs:

$$Profit^P(E) = Revenue^P(E) - Cost^P(E) \quad (E.3)$$

Finally, the profit margin over time period E will be of particular interest in the rest of the paper and can be defined as:

$$ProfitMargin^P(E) = Profit^P(E) / Revenue^P(E) \quad (E.4)$$

The next subsections describe how to evaluate service client technical and business metrics concerning the conventional SLA negotiation approach.

E.3.4 Conventional SLA Negotiation Formalization

The conventional approach to SLA negotiation is ad hoc [TT05b]. The main items to be decided upon are thresholds for the mean response time, T_{MAX} (or some percentile of the

response time distribution) and for service availability, A_{MIN} , as well as financial reward and penalty functions. These items represent the provider-client linkage or, said another way, the linkage between IT and the business, i.e., the linkage between what the IT infrastructure must do and the penalties applied when things go wrong or possible rewards when expectations are exceeded. The values used for these items are obviously tied to service pricing, since the thresholds set on the IT metrics can only be satisfied with a properly designed infrastructure with its corresponding cost. One thus concludes that SLA negotiation is directly tied to the design of the IT infrastructure.

As one can see from the preceding sections, only provider metrics are really of interest when performing infrastructure design. The wishes from the client perspective are expressed by means of SLA clauses tied to IT metrics for the service being offered. Given this state of affairs, the provider now proceeds to optimize its own financial measures and design infrastructure (the server farm, in this case), as described below:

Table E.1: Design Problem Associated with Conventional SLA Negotiation

Find	n_j and m_j for each resource class RC_j .
By Maximizing:	$Profit^P(E)$, the provider profit over time period E
Subject to:	$n_j \geq m_j, m_j \geq 1, A \geq A_{MIN}$ and $\bar{T} < T_{MAX}$.
Where:	n_j is the total number of servers in resource class RC_j ; m_j is the number of load-balanced servers A_{MIN} is the minimum acceptable value for the final customer availability of service S ; T_{MAX} is the maximum acceptable value for the final customer average response time of service S ; E is the SLA evaluation period.

E.4 Business-Driven SLA Negotiation Approach

In this section, the SLA negotiation problem is reconsidered from a business perspective (Figure E.2). In other words, one now wishes to establish linkage between provider and client

in a clearer, more direct way, thus allowing SLA negotiations to be performed in a less ad hoc manner. The main technique used is to estimate and factor in *business loss* sustained by the client due to service quality or lack thereof. This technique falls under a new IT management research area called Business-Driven IT Management (BDIM) [SMM⁺05c; SB04; MBC04]. The new approach to SLA negotiations is termed the *business-driven approach*.

In order to capture business losses, an impact (loss) model must be developed. This model will be summarized below and is fully described in [SMM⁺05c; SMM⁺05b]; it is used to create a cause-effect relationship between IT infrastructure failure events and business loss. We use the loss model in the sections that follow to show how the SLA negotiation problem can be reexamined using a more direct business perspective.

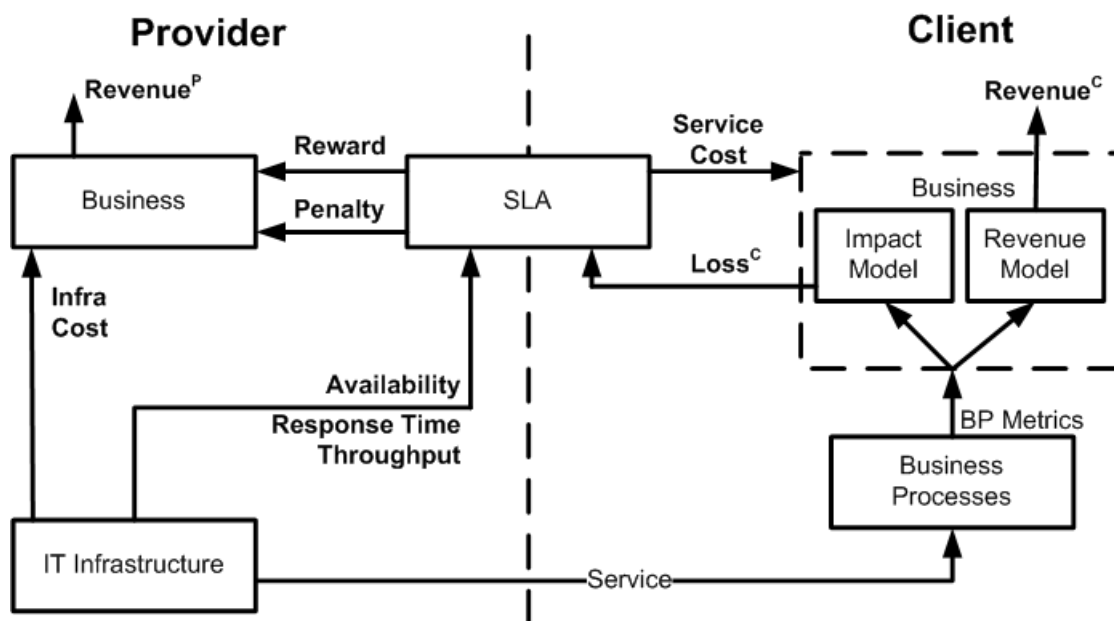


Figure E.2: Business-Driven SLA Negotiation Approach

E.4.1 Improving the Service Customer-Client Linkage

In the conventional SLA negotiation approach IT-business linkage is expressed weakly since performance requirements and penalty values are set in an ad hoc way, without capturing the true effect of service imperfections on the business. This fact motivates our work: based on the argument developed in [SMM⁺05c; SMM⁺06b] the optimal infrastructure that maximizes the service provider and service client profit margins is found; after such an optimal

infrastructure design is found, the SLO thresholds are calculated and can be inserted in the SLA, rather than guessed at. Moreover, the penalty imposed by the service client on the provider can be made a function of service client loss (see subsection E.4.1). This section describes technical and business metrics that are taken into account by the proposed approach while negotiating SLAs and designing the infrastructure required to provide the service.

The main new feature of the approach in this paper is to combine both the provider business perspective and the client business perspective since previous work does not address an outsourced service scenario with fidelity as far as modeling provider and client aspects of an SLA. We have already seen provider-side metrics; we now need to briefly discuss client-side metrics.

Service Client Side

Service Client Loss

The service client loss, that is, the business loss due to either IT infrastructure failures or performance degradations come from two sources: service unavailability and customer defections. Let us couch the argument in an e-commerce context; a customer accessing the service site will defect – that is, not conclude its purchases – when response time is superior to a given upper bound, T^{DEF} ; the value of 8 seconds is frequently mentioned in the literature as an appropriate threshold value [MAD04]. Consequently, one needs to calculate $B(T^{DEF}) = 1 - ResponseTimeDistribution(T^{DEF})$, the probability that response time exceeds this upper bound. Service availability, A , and customer defection probability, $B(T^{DEF})$, are combined to yield the client business loss over time period E as follows:

$$Loss^C(E) = \phi \cdot E \cdot \Delta X = \phi \cdot E \cdot (\Delta X^A + \Delta X^T)$$

where ϕ represents the average revenue per completed revenue-generating session from RG CBMG, ΔX represents the lost throughput in transactions per second, $\Delta X^A = f \cdot \gamma \cdot (1 - A)$ represents the lost throughput, in transactions per second, attributable to service client site unavailability and $\Delta X^T = f \cdot \gamma \cdot B(T^{DEF}) \cdot A$ represents the lost throughput, in transactions per second, attributable to high response time, γ represents input load rate the service client is subject to by its final consumers and f represents the fraction of revenue-generating sessions.

Service Client Cost

Since there is only a single service, the amount paid by the client to obtain service, $Cost^C(E)$, over time period E is the price charged by the provider:

$$Cost^C(E) = Price(E) \quad (E.5)$$

Service Client Revenue

The business revenue model assumes that, if the provided service were perfect, the service client revenue would be $Revenue_{potential}^C(E) = \gamma \cdot \phi$. In fact, this is only potential revenue since $\Delta(X)$ transaction/seconds are lost due to unavailability or final customer defections; therefore, client revenue is:

$$\begin{aligned} Revenue^C(E) &= Revenue_{potential}^C(E) - Loss^C(E) \\ &= \gamma \cdot \phi - Loss^C(E) \end{aligned} \quad (E.6)$$

Service Client Profit

The client's net income for time period E is:

$$Profit^C(E) = Revenue^C(E) - Cost^C(E) \quad (E.7)$$

and the profit margin is:

$$ProfitMargin^C(E) = Profit^C(E) / Revenue^C(E) \quad (E.8)$$

E.4.2 Business-Driven SLA Negotiation

We now have a very different scenario when compared to the conventional approach. We have achieved two things:

1. We have captured direct IT-business linkage through the notion of *business loss*;
2. We have used the concept of business loss to formally tie provider and client business results (profit margins) to each other. This tie can be directly expressed in SLA reward and penalty functions, if desired.

We are now in a position to rethink the infrastructure design problem and the problem of choosing SLO values (such as T_{MAX} and A_{MIN}). Informally, the business-driven SLA negotiation problem consists of maximizing service provider and service client profit margins as a multi-objective optimization problem, considering the business loss suffered by the client. The problem can be formally stated as follows:

Table E.2: Design Problem Associated with Business-Driven SLA Negotiation

Find	n_j and m_j for each resource class RC_j
By Maximizing:	$ProfitMargin^P(E)$, the provider profit margin over time period E and $ProfitMargin^C(E)$, the client profit margin over the same period
Subject to:	$n_j \geq m_j, m_j \geq 1, T < T_{MAX}$ and $A \geq A_{MIN}$
Where:	n_j is the total number of servers in resource class RC_j ; m_j is the number of load-balanced servers in resource class RC_j ; A_{MIN} is the minimum acceptable value for the availability of service S ; T_{MAX} is the maximum acceptable value for the mean response time of service S ; E is the SLA evaluation period.

Having solved this optimization problem, the infrastructure used to provision the service will be known and the SLO values to be used in the SLA can be calculated.

E.5 Evaluating the Approach: an Example

The purpose of this section is to evaluate the proposed method through a complete example scenario for which an SLA must be designed to underpin an outsourced service. Despite being hypothetical, it is suitable to show the value of the proposed approach in comparison with the conventional one.

In order to satisfy the requirements of the analytical model, the chosen scenario refers to an e-commerce service obtained through a web site. In this scenario, the e-commerce order-taking business process is heavily IT-dependent and is suited to the revenue model

used here. There are 3 resource classes in a three-tier architecture: a Web Tier resource class (RC_{web}), an Application Tier resource class (RC_{as}) and a Data Tier resource class (RC_{db}). Each resource class is composed of three components: hardware (hw), operating system (os) and either Web Server Software (ws) in the web tier, Application Server software (as) in the application tier or Database Management System (db) in the data tier. The site has 2 CBMGs: a revenue-generating and a non-revenue generating one. The average number of visit to each state of each CBMG is shown in Table E.3. These values are obtained from the CBMG transition probabilities as shown in [SMM⁺05b].

Table E.3: Average Number of visits to each CBMG state during a customer session

State (r)	RG Session (V_r^{RG})	NRG Session (V_r^{NRG})
Entry	1.000	1.000
Home	1.579	1.780
Browse	2.325	4.248
Search	3.300	3.510
Login	0.167	0.005
Pay	1.000	0.000
Register	0.083	0.003
Add to cart	1.667	0.069
Select	2.250	1.309
Exit	1.000	1.000

The parameters used to feed the example scenario are shown in Table E.4 and are typical for recent technology [SMM⁺05c; JST03; MAD04]. In this table, tuples (a,b,c) represent the values for all resource classes (RC_{web} , RC_{as} , RC_{db}) in that sequence. Service demand on resources follows an exponential distribution with mean values given in Table E.5.

The first step toward the demonstration of the proposed approach consists of a comparison between the two approaches to SLA design – conventional and business-driven. This is done by comparing two financial dimensions: the cost dimension and the business results dimension. On the one hand, the conventional approach tries to find the cheapest infrastructure that meets the performance requirements. On the other hand, the business-driven approach attempts to maximize both service provider and service client profit margins. The

Table E.4: Parameters for example scenario

Parameters	Values
T^{DEF}	8 seconds
ϕ	\$1 per transaction
E	1 month
γ	14 transactions per second
α_j The relative processing power of servers in class R_j	(1,2,5)
$c_{j,k}^{active}$ (\$/month)	hw =(1430, 1651, 5720) os=(214.5, 214.5, 214.5) as=(79.3, 45.5, 858)
$c_{j,k}^{standby}$ (\$/month)	hw =(1000, 1150, 4000) os=(150, 150, 150) as=(55, 30, 600)
$(A_{web}^R, A_{as}^R, A_{db}^R)$	(99.81%, 98.6%, 98.2%) (these values are calculated from appropriate MTBF and MTTR values)
a	1.5
T_{MAX}	1.55 seconds
A_{MIN}	99.96%

Table E.5: Resource demand in milliseconds

Tier	CBMG state							
	h	b	s	g	p	r	a	d
Web tier	50	20	30	70	50	30	40	30
Application tier	0	30	40	35	150	70	40	25
Database tier	0	40	50	65	60	150	40	30

performance requirements, T_{MAX} – the maximum acceptable value for the average service response time – and A_{MIN} – the minimum acceptable value for service availability – were established at 1.5 seconds and 99.96%, respectively, common values for service design chosen in an ad hoc fashion [TT05b].

Figure E.3 presents a comparison between optimal designs from the service provider perspective using both approaches; only the cost dimension is shown. In this study, the load varies from 5 transactions/sec to 9 transactions/sec and optimal values for IT Infrastructure cost are plotted on the vertical axis. Based only on this information, one sees that the conventional method yields better results than the new one being proposed since it yields a cheaper infrastructure; for a particular load value, the average yearly cost using the conventional technique is \$324,600, while it is \$345,400 using the business-driven approach.

However, when the profit metric is considered, one realizes that the previous hasty conclusion leads to lower profit values. Figure E.4 shows provider profit. The difference between the earned profit is greater than the extra infrastructure cost, and it is thus better for the provider to offer better service and collect higher profits. On the average, the provider will spend approximately 6% more with the business-driven approach while raising profits by almost 12%.

Finally, we show in Figure E.5 that the client also benefits from the proposed approach. The figure applies to the business-driven approach (which optimizes profit margin). As load increases, IT infrastructure cost (and quality) also increases in order to avoid service response time degradation. However this also causes service price to increase, but profits increase by an even larger amount; as a result, the difference between business losses and service cost decreases. On the average, the client will spend, on the average, approximately \$38,550 more with the business-driven approach while raising profits by almost \$562,630, on the average. Furthermore, with the business-driven approach, the average loss value is \$11,780 while the conventional approach yields an average loss of \$17,930, 52.2% higher.

With the scenario optimized from the business angle, one can now simply calculate – from the optimal infrastructure – the SLO values to be used: in this case, for a load value of 7 transactions per second, the values are $T_{MAX} = 0.112s$ and $A_{MIN} = 99.968\%$. These values are obtained from the detailed reliability and queuing-theoretic models developed in [SMM⁺05b]. Observe that these values were not chosen in an ad hoc way but calculated

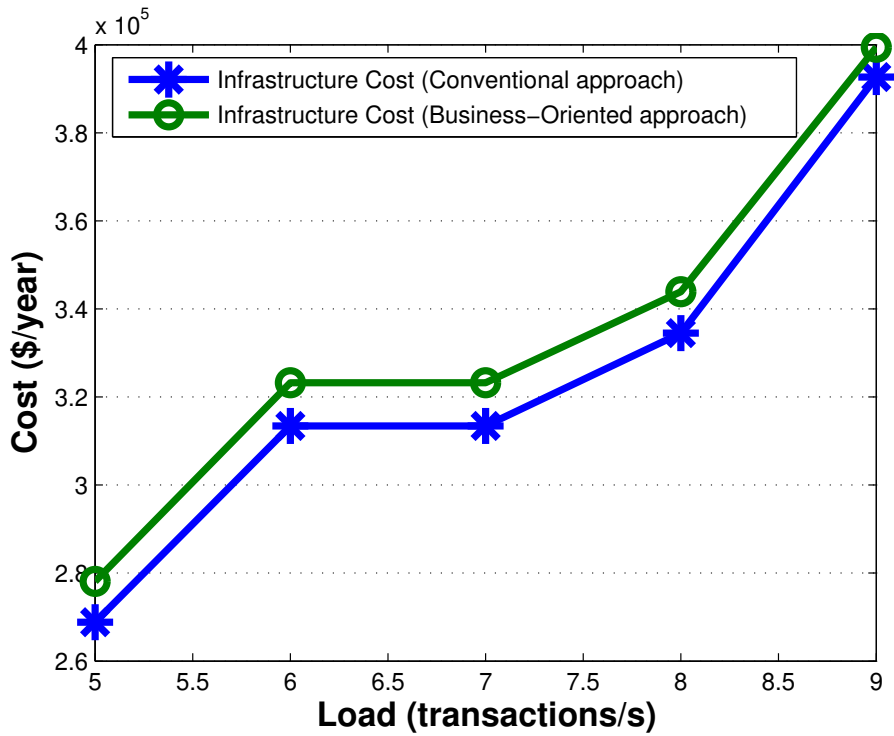


Figure E.3: Service Provider Cost Comparison

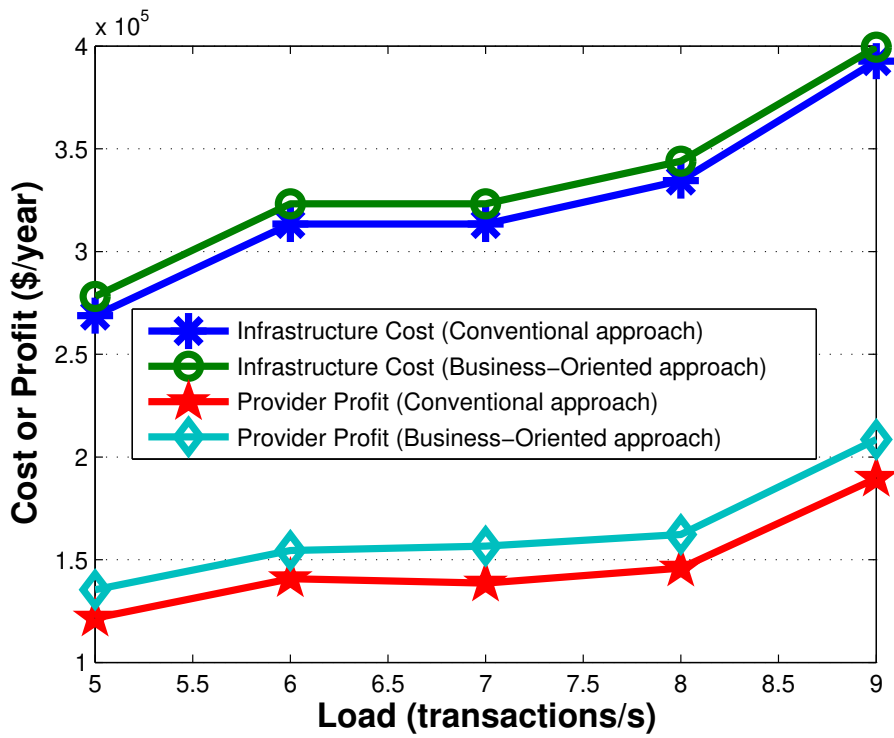


Figure E.4: Service Provider Cost and Profit Comparison

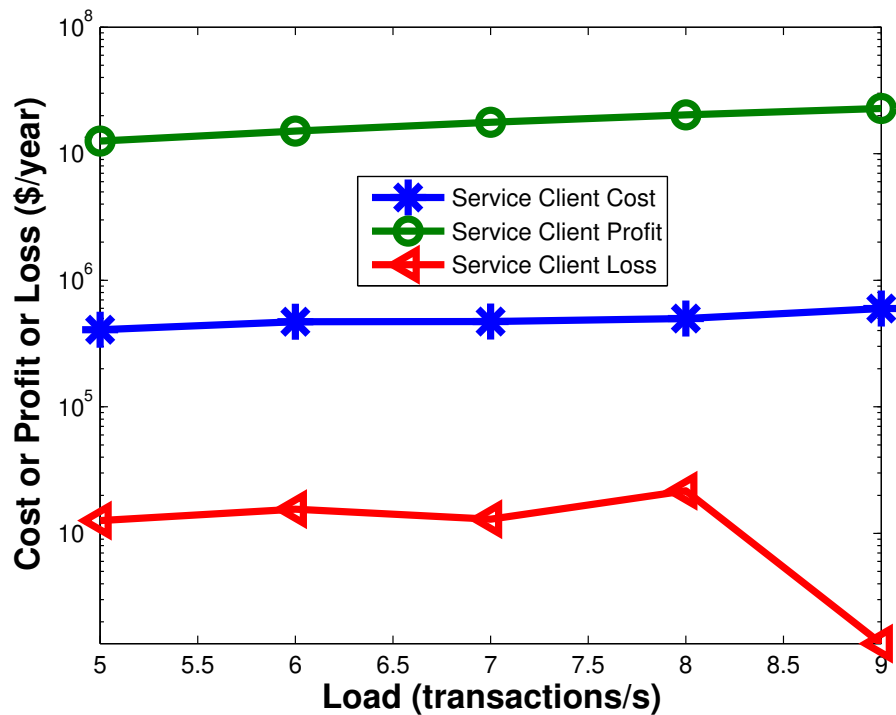


Figure E.5: Client Loss, Cost and Profit Comparison (Business-driven Approach)

after the optimal infrastructure was found.

E.6 Conclusions

This paper has proposed and formalized a business-driven method to design SLAs for outsourced IT services. Recall that "designing SLAs" here means negotiating values for SLA parameters and designing infrastructure to meet the service objectives. Negotiation must occur since two parties with conflicting business concerns are involved: the service provider and the service client. For the purpose of comparison with the proposed approach, the conventional SLA negotiation approach has also been formalized. The crucial difference between these methods is related to the way in which the IT-business linkage (or, equivalently, the provider-client linkage) is captured and defined. In the conventional approach, performance objectives for availability and response time, say, as well as penalty value are defined in an ad hoc way, that is, without taking into account the true impact that IT imperfections have on the client's business. On the other hand, the approach suggested here improves the

SLA design process by considering client business losses incurred from either IT infrastructure failures or service performance degradation in order to improve the provider-client linkage. The client business loss is evaluated by means of a business impact model, a new kind of model used in the area of Business-Driven IT Management. The aim is to design SLAs that are more beneficial to both provider and client. The value of the business-driven approach was demonstrated through a complete numerical scenario. The scenario clearly showed that the objectives were met: the business-driven approach yielded better profits for both provider and clients through the judicious design of the infrastructure and consequent choice of SLA parameters.

In the future we plan to (i) investigate other dimensions in the SLA design space such as how to define optimal penalty and reward functions; (ii) improve the infrastructure model in order to consider cache effects and thus much higher scales, other parts of the infrastructure such as the network and the use of legacy IT infrastructure; (iii) reconsider the approach in the light of dynamic infrastructures proportioned by autonomic computing.

Acknowledgment

We would like to acknowledge and thank the Bottom Line Project team – special thanks to Rodrigo Rebouças – for their support and discussions. This work was developed in collaboration with HP Brazil R&D.