UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

GABRIEL FELIPE CARDOSO GOMES

HEURÍSTICA ONLINE PARA AQUISIÇÃO DE INSTÂNCIAS EM CLOUD PÚBLICA: UM OLHAR SOBRE A DEMANDA DE E-COMMERCE

CAMPINA GRANDE - PB

2022

GABRIEL FELIPE CARDOSO GOMES

HEURÍSTICA ONLINE PARA AQUISIÇÃO DE INSTÂNCIAS EM CLOUD PÚBLICA: UM OLHAR SOBRE A DEMANDA DE E-COMMERCE

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Professor Dr. Thiago Emmanuel Pereira da Cunha Silva

CAMPINA GRANDE - PB

2022

GABRIEL FELIPE CARDOSO GOMES

Heurística online para aquisição de instâncias em cloud pública: Um olhar sobre a demanda de e-commerce

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Professor Dr. Thiago Emmanuel Pereira da Cunha Silva

Orientador – UASC/CEEI/UFCG

Professor Dr. Fábio Jorge Almeida Morais

Examinador – UASC/CEEI/UFCG

Professor Tiago Lima Massoni

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 06 de Abril de 2022.

CAMPINA GRANDE - PB

ABSTRACT

The spending that large enterprises are having on Infrastructure as a Service (IaaS) is large and growing. Public cloud providers make various purchasing options available for their services. These options are usually split into paying later for the time you have used the service or paying in advance, with an associated discount, for a reservation of the service to be used in the future. There are many studies that seek to minimize the cost of resource allocation in IaaS providers, based on the estimated future demand and the purchasing options made available by the providers. One of the possible solutions is the use of online algorithms that are able to decide when to reserve during the arrival of demand, this study seeks to evaluate the performance of an online heuristic taking into account the demand of a large e-commerce company.

Heurística online para aquisição de instâncias em *cloud* pública: Um olhar sobre a demanda de *e-commerce*

Gabriel Felipe Cardoso Gomes*
gabriel.gomes@ccc.ufcg.edu.br
Universidade Federal de Campina Grande (UFCG)
Campina Grande, Paraíba, BR

Thiago Emmanuel Pereira
temmanuel@computacao.ufcg.edu.br
Universidade Federal de Campina Grande (UFCG)
Campina Grande, Paraíba, BR

RESUMO

Os gastos que as grandes empresas estão tendo com Infraestrutura como serviço (IaaS) são grandes e estão crescendo cada vez mais. Provedores públicos de *clouds* disponibilizam diversas formas de compra para seus serviços. Essas opções geralmente são divididas em pagar posteriormente pelo tempo que você utilizou o serviço ou pagar antecipadamente, com um desconto associado, por uma reserva do serviço a ser usada no futuro. Existem muitos estudos que buscam minimizar o custo da alocação de recursos em provedores IaaS, com base na estimativa de demanda futura e nas opções de compra disponibilizadas pelos provedores. Uma das possíveis soluções é o uso de algoritmos online que são capazes de decidir quando reservar durante a chegada da demanda, esse estudo busca avaliar o desempenho de uma heurística online levando em consideração a demanda de uma grande companhia de *e-commerce*.

ACM Reference Format:
Gabriel Felipe Cardoso Gomes and Thiago Emmanuel Pereira. 2022. Heurística online para aquisição de instâncias em *cloud* pública: Um olhar sobre a demanda de *e-commerce*. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUÇÃO

O mercado de *clouds* públicas está em constante ascensão. Estima-se que o mercado de IaaS possui um crescimento anual de 41.7% [1]. Esse crescimento é devido à necessidade por recursos computacionais das empresas de tecnologia para que elas possam se hospedar e servir seus produtos. Por exemplo, a Netflix é uma das principais líderes em quantidade de conteúdo que é servido online, precisando lidar com uma enorme quantidade de clientes através do mundo e por isso contam com uma infraestrutura de mais de 100 mil instâncias [2]. Possuir uma infraestrutura desse tamanho exige um bom planejamento de custo, já que *clouds* públicas, tendo a AWS como exemplo, oferecem diversas formas de compra, para que o cliente possa customizar de acordo com sua demanda e minimizar o custo da infraestrutura.

Entre os métodos de compra da AWS temos: instâncias sob demanda e as instâncias reservadas. As instâncias sob demanda cobra o cliente apenas pelas horas em que o recurso foi utilizado, mas o valor pago por hora é o mais alto entre os métodos de compra, já que não existe nenhum desconto. As instâncias reservadas são planos de um ou três anos em que o cliente recebe o recurso por todas as horas do período de tempo contratado, pagando pelas mesmas antecipadamente, havendo um desconto que pode variar em relação a quantidade de tempo contratado e ao adiantamento do pagamento.

O desconto entregue pelo modelo de reserva pode chamar a atenção dos clientes para essa opção de compra, mas podem ocorrer cenários onde há o desperdício de recurso caso a reserva seja maior que a demanda futura do serviço, além do método de pagamento adiantado não ser viável financeiramente para alguns clientes. Esse tipo de problema torna essencial o estudo e entendimento do comportamento da demanda dos seus serviços, para que a compra seja realizada da maneira mais barata possível.

Decidir como realizar a compra desses recursos já é um problema bastante explorado na literatura, tendo soluções com algoritmos onlines [3], modelos de predição estatístico [4] e rede neural [5]. O estudo apresentrado nesse trabalho é a avaliação de uma heurística online [3] usando a demanda de uma grande companhia de *e-commerce*.

2 CONTEXTO

2.1 Mercado de Instâncias

Clouds públicas costumam oferecer dois modelos principais de compra de instâncias 1) o modelo "pague pelo que usar"; 2) e o modelo de reserva. No caso da AWS esses modelos são chamados de on-demand e reserved instance, respectivamente. No modelo on-demand, o usuário paga um valor pela quantidade de horas que ele consumir de um determinado tipo de instância. Por exemplo, uma instância EC2 t2.medium (Linux, Norte da Virgínia) possui uma taxa de consumo por hora de $0,0464 dólares [6]. Nesse caso, paga-se $4,64 dólares para provisionar essa instância por 100 horas. Uma vantagem apresentada pelo mercado *on-demand* é que não há necessidade de firmar um vínculo de compromisso, isto é, o usuário pode parar de utilizar a instância a qualquer momento.

Enquanto isso, no mercado de instância reservada é necessário firmar um compromisso de tempo de uso da instância. A AWS possui as opções de um ou três anos; entretanto, ao utilizar esse tipo de mercado a taxa de consumo por hora recebe um desconto. Também existe a opção de pagar adiantado parte do custo total do vínculo. Caso esse pagamento adiantado seja realizado o desconto será ainda maior. Por exemplo, a instância EC2 t2.medium (Linux, Norte da Virgínia), discutida no exemplo anterior, caso seja reservada por um período de 1 ano com pagamento adiantado parcial de $120,00 tem um desconto na taxa de consumo por hora de quase 70%: sua taxa de consumo passa a ser $0,014. Com isso, o custo total da reserva

Conference acronym 'XX, March 03–05, 2022, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

é $120 + 0.014h$, onde h é a quantidade de horas da reserva. Para o exemplo de 1 ano, temos $h = 365 * 24$. Para generalizar, podemos dizer que o custo de reserva c_r é $c_{up} + o_c dh$, onde c_{up} é o valor pago adiantado, o_c é a taxa de consumo por hora e d é o desconto oferecido para a reserva.

2.2 O problema do planejamento de capacidade

Pode-se perceber que não é trivial a escolha de qual mercado deve ser utilizado, visto que pode se usar múltiplos mercados para comprar as instâncias que irão suprir a demanda. Por sorte, esse não é um problema novo na literatura.

Por exemplo, o *Bahncard Problem* pode ser visto como uma generalização do problema de compra de instâncias [7]. O Bahncard busca minimizar o custo de realizar múltiplas viagens ao decorrer de um longo período de tempo em uma linha ferroviária. Nessa linha é possível comprar uma passagem pelo preço p, porém, também é possível comprar um passe de validade T pelo custo C onde todas as passagens compradas passaram a ter preço βp, podendo ser definido como $BP(C, \beta, T)$. É fácil perceber como o *Bahncard problem* pode ser mapeado para a compra de instâncias no modelo de reserva: C pode ser utilizado como o custo do pagamento adiantado, β é o desconto da reserva para o custo o_c e T é o período de reserva.

Em $BP(C, \beta, T)$, caso saibamos as viagens que serão feitos ao longo do tempo, é possível chegar a uma resposta ótima através de técnicas de programação linear [8] ou programação dinâmica [3].

Mas, da mesma forma que é difícil para um viajante saber quais serão as viagens que ele fará no futuro, é também difícil antecipar a demanda futura de uma aplicação para um grupo de instâncias de maneira que possamos decidir o mercado que será comprado. Uma possível solução seria tentar prever qual seria o futuro e resolver o problema com o futuro predito, como utilizado em [4]. De toda forma, prever o futuro não é tão simples, principalmente para startups ou até mesmo novos serviços de grandes companhias que não possuam ou possuam de maneira limitada, o acesso a um histórico de demanda, ou ainda, que possuam comportamento altamente variável e não estacionário, dificultando a predição.

Uma outra alternativa, é encarar o problema de maneira online. Como a maioria das instâncias obtidas através de IaS são pagas em grão de hora, podemos imaginar que a cada hora t receberemos uma demanda d_t e precisaremos tomar a decisão de quantas o_t instâncias *on-demand* serão utilizadas, e por fim $d_t - o_t$ instâncias reservadas. Realizar a reserva sempre que recebemos uma nova demanda é o que caracteriza um algoritmo online, já que os resultados começam a ser processados a partir de cada nova entrada.

Essa mudança na especificação do problema o torna mais difícil de ser resolvido já que compras que parecem boas de serem realizadas no presente podem se tornar piores em possíveis futuros. Mas mesmo que o futuro se mostre o pior possível, somos capazes de medir qual será a maior diferença de custo que a solução heurística terá em relação a solução ótima que conhece o futuro. [3] calculou essa diferença, e sendo C_{opt} custo otimizado da demanda, o custo máximo que a heurística pode chegar é $(2 - C_{opt})$ onde é o desconto obtido no custo hora da reserva em relação a opção *on-demand*.

Apesar da solução não ter um desempenho tão elevado como opções que usam predição do futuro, ela é especialmente útil para startups e novas aplicações, visto que não é possível prever o comportamento desses cenários já que o passado é inexistente ou limitado.

Nesse estudo, iremos utilizar a heurística proposta por [3], o algoritmo procura o limiar de gasto com *on-demand* onde o gasto com instância reservada teria resultado no mesmo custo. De maneira mais formal, sendo o_c o custo *on-demand*, c_{up} o custo de pagamento adiantado e d a taxa de desconto da reserva, quando $o_c h = c_{up} + o_c dh$ significa que durante essas h horas não haveria diferença no custo entre servir a demanda com instâncias *on-demand* ou instâncias reservadas, o momento h que marca essa igualdade é chamado de limiar de quebra ou α. Sendo assim, a heurística tenta corrigir esse erro e começa a reservar instâncias. O pseudocódigo pode ser encontrado abaixo, t é o instante de tempo em que a demanda chega e r o período de reserva em horas.

Algoritmo 1 Heurística Online

1: x_i será o número de instâncias reservadas no tempo i, inicialmente $x_i = 0$, para todo i.
2: $I(X)$ será uma função, onde $I(X) = 1$, se X é verdadeiro; e $I(X) = 0$, caso contrário.
3: Para cada nova demanda d_t realize o seguinte loop:
4: **while do** $\sum_{i=t-r+1}^{t} I(d_i > x - i) > \alpha$
5: Reserve uma nova instância: $r_t \leftarrow r_t + 1$
6: Atualize as reservas que podem ser utilizadas no futuro: $x_i \leftarrow x_i + 1 | \forall i \in [t, t + r - 1]$
7: Adicione uma reserva fantasma no passado para sinalizar que já corrigimos uma reserva: $x_i \leftarrow x_i + 1 | \forall i \in [t - r + 1, t - 1]$
8: **end while**
9: Suba instâncias *on-demand*: $o_t \leftarrow (d_t - x_t)^+$
10: Espere próxima demanda: $t \leftarrow t + 1$, repita o passo 3.

3 METODOLOGIA

A fim de avaliar o desempenho da heurística citada na seção anterior, usaremos a demanda real de alguns produtos de uma companhia de *e-commerce*, o período analisado é todo o ano de 2020. Devido a demanda ser uma informação sensível e de difícil acesso, tanto pelo lado da AWS quanto pela companhia, realizamos aproximações com os gastos no período de tempo avaliado, que foram coletados através do AWS Cost Explorer. Tivemos acesso aos custos diários separados por mercado e tipo de instância, dividimos o custo diário pelo preço daquele tipo de instância naquele mercado e obtivemos a quantidade de instância/hora gastas durante o dia - essas instância/hora são atribuídas igualmente entre todas as horas do dia.

De maneira mais formal, para cada dia D existe um custo diário total $dtc_{IT,MT}$ associado ao tipo de instância IT compradas através do mercado MT e para esse dia, a demanda por hora é igual a $\frac{dtc_{IT,MT}}{c_{IT,MT}} * \frac{1}{24}$, onde $hc_{IT,MT}$ é o custo por hora da instância de tipo IT no mercado MT. Nós não consideramos o custo tido com instâncias Spot e para realizar os cálculos levamos em consideração o preço das instâncias e seus respectivos mercados em Fevereiro de 2022. Segue algumas estatísticas dos nossos dados de demanda:

Como pode ser visto na diferenças das estatísticas da Tabela 1 e da Tabela 2, a segunda metade do ano possui uma demanda maior

Minimo	Máximo	Desvio Padrão	Média
2	4039	887.1	743.7

Tabela 1: Estatísticas da demanda durante todo o ano de 2020 (quantidade de instâncias por hora).

Minimo	Máximo	Desvio Padrão	Média
288	4039	880.6	1281.6

Tabela 2: Estatísticas da demanda entre os meses de junho e novembro no ano de 2020 (quantidade de instâncias por hora).

que a primeira metade, o motivo dessa grande variabilidade dos dados é devido a eventos que criaram picos na demanda, como a *Black Friday*.

Depois de calcular a alocação de compras da demanda usando a heurística, poderemos comparar o desempenho da estratégia em relação às compras adotadas pela companhia. Também utilizaremos um algoritmo offline para calcular a compra ótima,a alocação de compras com o menor custo, como citado em [8] e qual o custo necessário para prover toda a demanda apenas com o mercado on-demand, dessa forma poderemos saber quão distante a estratégia heurística e a da companhia estão do ótimo e do alocação mais trivial.

Para maximizar a lealdade de comparação e facilitar as análises, a heurística online e o alocador ótimo só poderão criar reservas do mercado de instância reservada com pagamento adiantado parcial e no caso da solução online só será considerado o custo das reservas que estiverem dentro do período de 2020, essa restrição é necessária visto que existe a possibilidade de uma reserva durar para além do período analisado e o valor de pagamento adiantado pode subir o custo dessa estratégia mais do que deveria. Sendo assim, o valor de pagamento adiantado será diluído nas horas que estiverem dentro do intervalo analisado, de maneira formal o custo da reserva cr é considerado como $c_{od}dh\frac{c_{up}}{8760}$.

4 RESULTADOS

Essa seção descreve os resultados obtidos pelas diferentes estratégias: heurística online, estratégia da companhia e estratégia ótima. Já que os valores de custo são dados sensíveis, todas as análises levarão em consideração os valores normalizados pela estratégia da companhia, isto é, o custo para prover toda a demanda da companhia durante o ano de 2020 será 1.

Como podemos observar na Figura 1, a heurística online obteve um desempenho pior que a companhia chegando a ser 38% mais custosa, aproximadamente o desempenho da estratégia trivial de comprar todas *on-demand*. O esquema de alocação adotado pela companhia ainda possui 13% de espaço para melhorar em relação a estratégia ótima.

Para entender melhor o desempenho da heurística temos na Figura 2, nele é possível ver que diferente da estratégia adotada pela companhia, as reservas demoram a acontecer, isso acontece principalmente devido a ideia central do algoritmo de esperar as

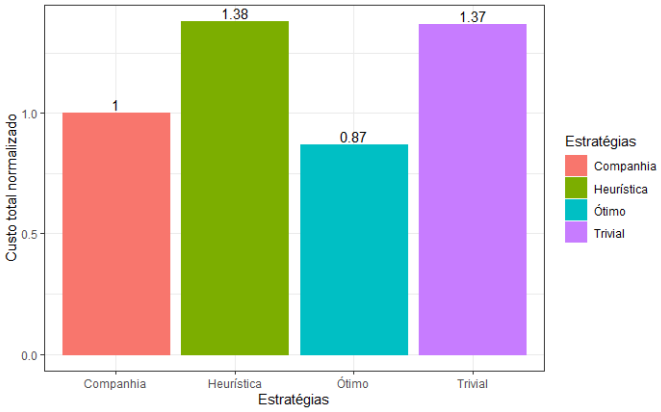

Figura 1: Custo total normalizado para prover a demanda.

instâncias on-demand serem totalmente mal utilizadas para iniciar as reservas.

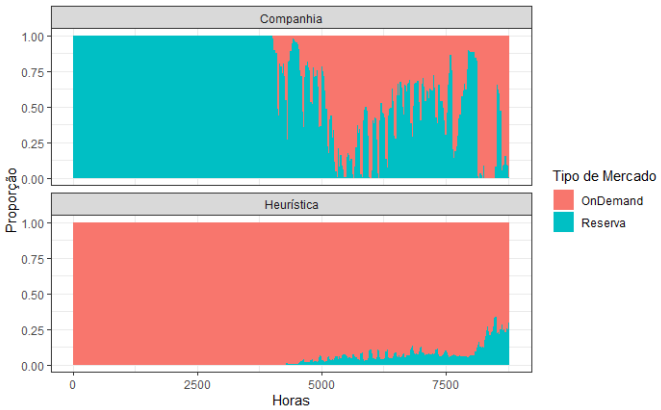

Figura 2: Proporção de instâncias por mercado de compra entre a estratégia heurística e da companhia.

Caso o período de análise fosse maior, a heurística poderia ter mais tempo para se adaptar com a demanda e apresentaria resultados melhores. Vale ressaltar que Wang [3] recomenda a utilização da estratégia para situações onde não se conhece o futuro nem o comportamento da demanda e que o algoritmo também suporta parâmetros adicionais como pequenas previsões do futuro e aumento da agressividade de reserva que não foram considerados nesse estudo.

5 TRABALHOS RELACIONADOS

Em 2013, Wang et al [3] lançou o artigo "*To Reserve or Not to Reserve: Optimal Online Multi-Instance Acquisition in IaaS Cloud*", este trabalho mostrou que o problema da reserva de instância era uma generalização do Bahncard e ao mesmo tempo sugeriu uma heurística online para resolvê-lo. Um grande diferencial desse trabalho é evitar a solução de preditar o futuro para resolver a alocação da reserva.

Por outro lado, trabalhos como Zhu et al [9] e Calheiros et al[4] atacam o mesmo problema de alocação de reserva, porém, procurando maneiras eficientes de preditar o futuro da demanda. Nossos próximos estudos focará na avaliação dessas diferentes abordagens para os diferentes tipos de demanda.

6 CONCLUSÃO

Nesse estudo nós avaliamos o desempenho de como uma heurística online se comporta dentro da demanda de uma grande companhia de ecommerce, como dito por Wang [3], não é o cenário onde o algoritmo mais se destaca, já que o diferencial online se sobressai em startups e serviços onde há uma dificuldade em se prever a demanda. Contudo, existem adaptações na heurística que podem melhorar seu desempenho como pequenas predições do futuro e a variabilidade da agressividade de reserva. Esses resultados preliminares servem como base para um aprofundamento maior na escolha de parâmetros e diferentes estratégias para o planejamento de capacidade levando em consideração a demanda de uma grande companhia de ecommerce.

REFERÊNCIAS

[1] Xath Cruz. The future of cloud adoption **Cloud Times**, 2012. Avaliable in: http://cloudtimes.org/2012/07/14/the-future-of-cloud-adoption/, visited on: 03-14-2022.

[2] Netflix on aws netflix **Amazon**, 2022. Avaliable in: http://aws.amazon.com/solutions/case-studies/netflix/, visited on: 03-14-2022.

[3] Wei Wang, Baochun Li, and Ben Liang. To reserve or not to reserve: Optimal online multi-instance acquisition in iaas clouds. In *10th International Conference on Autonomic Computing (ICAC 13)*, pages 13–22, 2013.

[4] Rodrigo N Calheiros, Enayat Masoumi, Rajiv Ranjan, and Rajkumar Buyya. Workload prediction using arima model and its impact on cloud applications' qos. ***IEEE transactions on cloud computing***, 3(4):449–458, 2014.

[5] Huamin Zhu, Jun Luo, and Hongyao Deng. Optimizing the procurement of iaas reservation contracts via workload predicting and integer programming. ***Mathematical Problems in Engineering***, 2020, 2020.

[6] On-demand pricing **Amazon**, 2022. Avaliable in: https://aws.amazon.com/ec2/pricing/on-demand/?nc1=h_ls, visited on: 03-14-2022.

[7] R. Fleischer. On the bahncard problem. page 161–174, 2001.

[8] Woo-Chan Kim and Ohyun Jo. Cost-optimized configuration of computing instances for large sized cloud systems. *ICT Express*, 3(3):107–110, 2017.

[9] C. Kenyon A. Karlin and D. Randall. Dynamic tcp acknowledgment and other stories about e/(e-1). page 209– 224, 2003.