

Gestão de Recursos para Computação em Nuvem usando Escolha Social Computacional

Felipe V. Furquim¹, Daniel Cordeiro^{1*}

¹Escola de Artes, Ciências e Humanidades - Universidade de São Paulo (USP)
São Paulo – SP – Brasil

fvfurq@usp.br, daniel.cordeiro@usp.br

Abstract. *Resource management in Cloud Computing platforms involves several fields of how computational resources are shared between users. A lot of decisions are made on both the provider and the user side, so both sides have an intrinsic sense of priority: there are some options that are better than others according to some criteria, but they all cover the needs. These preferences are not properly differentiated by traditional resource management algorithms. This research aims to design decision-making systems that aggregate the preferences of different actors, using concepts from Computational Social Choice Theory with strategyproof decision-making.*

Resumo. *A gestão de recursos em plataformas de Computação em Nuvem envolve vários campos de como recursos computacionais são compartilhados entre usuários. Diversas decisões são tomadas tanto no lado do provedor quanto no lado do usuário, assim ambos os lados têm um intrínseco senso de prioridade: tem algumas opções que são melhores que outras, de acordo com algum critério, mas todas elas cobrem as necessidades computacionais do usuário, respeitando as restrições de disponibilidade do provedor. Essas preferências não são propriamente diferenciadas por algoritmos tradicionais de gestão de recursos. Esta pesquisa tem como objetivo projetar sistemas de tomada de decisões que agreguem as preferências de diferentes atores, utilizando conceitos da Teoria da Escolha Social Computacional com tomada de decisão strategy-proof.*

1. Introdução

A Teoria da Escolha Social Computacional estuda como preferências individuais podem ser agregadas em uma escolha feita de forma coletiva [Brandt et al. 2016]. Aproveitando da teoria da Ciência da Computação, é possível notar aplicações da teoria da complexidade computacional e algoritmos de aproximação à escolha social. A Teoria da *Computational Social Choice* (COMSOC) pode providenciar ferramentas para fazer decisões conjuntas em um sistema multi-agentes formado por agentes de software heterogêneos e possivelmente egoístas [Duddy 2015].

A gestão de recursos em plataformas de Computação em Nuvem envolve vários campos de como recursos computacionais são compartilhados entre

*Este trabalho foi financiado pelos projetos n° 2019/26702-8 e 2021/06867-2, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

usuários [Cordeiro 2012]. Do ponto de vista de um provedor de Computação em Nuvem, Máquinas Virtuais precisam ser alocadas para o recurso físico apropriado. Essa escolha precisa levar em conta a carga de trabalho esperada, Máquinas Virtuais já alocadas nessa máquina, a CPU e memória disponíveis e etc. Além disso, a possibilidade de uma migração de uma Máquina Virtual cria a oportunidade de fazer um gerenciamento dinâmico de recursos, reconfigurando o escalonamento de tempos em tempos. Também há a gestão de recursos no lado dos usuários, que precisam escolher entre diferentes provedores de plataformas na nuvem, diferentes configurações de recursos e etc. Todas essas escolhas têm um impacto significativo na performance e no preço total a ser pago pelo usuário.

As escolhas de ambos os lados têm um intrínseco senso de prioridade: algumas opções são melhores que outras, de acordo com algum critério, mas todas elas cobrem as necessidades. Essas preferências não são propriamente diferenciadas por algoritmos tradicionais de gestão de recursos.

Idealmente, um processo de decisão colaborativo deve ser tal que os tomadores de decisão sempre votem pelas opções que são as mais vantajosas para si, independente das escolhas feitas pelos outros votantes. Na teoria da Escolha Social Computacional, diz-se que tais votantes não tem interesse em votar de forma *estratégica* (mudando seu voto para impedir que determinadas escolhas vençam). Sistemas de eleição que impedem esse comportamento são denominados *strategyproof* [Barberà and Jackson 1994].

A ideia é que os usuários da plataforma informem as características (tanto de hardware quanto de software) desejadas sobre os serviços e que a plataforma determine qual alocação de recursos deve fazer com base nessas escolhas. O usuário, por exemplo, informaria sua escolha com provedor de nuvem, diferentes configurações de recursos e capacidades, preços, região e etc. Pelo lado do provedor, a decisão a ser tomada em qual recurso físico apropriado uma máquina virtual deveria ser alocada, considerando a carga de trabalho, as VMs já alocadas nesse recurso, a quantidade de memória e CPU disponível, padrões de entrada e saída da VM e etc.

Porém, um usuário pode selecionar critérios que não refletem no seu cenário com o objetivo de conseguir um resultado melhor do que o ideal para seu caso. Isso pode ser prejudicial para outros usuários que farão uso do mesmo grupo de recursos físicos. Com o objetivo de impedir esse tipo de comportamento, estamos trabalhando em um sistema de votação *strategyproof*, mencionado anteriormente.

2. Sistema de Votação *Strategyproof*

Sistemas de votação adotam mecanismos (algoritmos) para a tomada de decisão de forma descentralizada. Há várias formas de agregar a preferência de diversos votantes. Um exemplo importante é o Orçamento Participativo (*Participatory Budgeting* [Peters et al. 2021]), um processo de eleição para determinar a alocação do orçamento de uma comunidade (como uma cidade ou região) dedicada ao financiamento de projetos aplicada pela primeira vez na cidade de Porto Alegre/RS e atualmente usada em diversos países do mundo.

Estamos interessados em estudar as propriedades dos mecanismos e algoritmos de votação [Gibbard 1973] que selecionam uma solução considerando a opinião dos eleitores e que, ao mesmo tempo, tornam a votação resiliente a manipulação por parte dos votantes (ou seja, que processos de eleição *strategyproof*).

Para estudar o problema do gerenciamento de recursos em plataformas de computação em nuvem de modo que os próprios usuários ou gestores da plataforma possam tomar as decisões de forma individual, foi desenvolvido um modelo de sistema de votação por pontuação (*Score Voting System*). Este modelo permite a introdução da correlação entre objetos: a escolha por um objeto (opção de voto) irá impactar na escolha de outro objeto. Ou seja, cada vez que um objeto for selecionado, isso pode favorecer outro objeto por meio de um processo de pontuação.

Para tanto, criou-se um sistema de votação por pontuação (*Score Voting Rule* [Cohen et al. 2022]). Um *Score Voting Rule* é um algoritmo que define uma regra de votação criada a partir de uma matriz quadrada para calcular o vencedor de uma eleição, dada um perfil de votação (visto como um vetor coluna de elementos nessa matriz). Ou seja, o algoritmo recebe um perfil de votação como entrada e retorna o conjunto de objetos vencedor.

Score Voting Systems podem ser usados para modelar sistemas eleitorais utilitários [Briest 2005] onde o eleitor deve escolher várias opções entre um conjunto de escolhas possíveis. O modelo é semelhante ao *Knapsack Voting* [Goel et al. 2019], onde cada cédula que contém um objeto dá um ponto para o mesmo, e o vencedor é o objeto com mais pontos. A ideia da votação por pontuação é generalizar os votos, permitindo que os eleitores atribuam qualquer quantidade de pontos a qualquer objeto.

3. Estudo do Modelo na Prática

O uso do modelo de *Score Voting* [Cohen et al. 2022] permite estudar e desenvolver mecanismos de escolha que sejam *strategyproof*. A fim de avaliar o uso prático desse modelo, utilizamos os dados disponibilizados pelo projeto *PARTicipatory BUDgeting LIBrary* (Pabulib [Stolicki et al. 2020]), um projeto que tem como objetivo coletar e compartilhar publicamente dados de votações de Orçamento Participativo.

Dado que esse repositório contém informações sobre eleições reais de orçamento participativo, incluindo os projetos sendo votados e as próprias cédulas de votação, nós mapeamos os *datasets* para uma matriz que representam uma *Score Voting Rule*. Desenvolvemos um algoritmo para construir uma matriz quadrada para o *score voting*, com linhas e colunas representando cada projeto a ser votado, e os valores representando o peso (negativo) das categorias em comum entre os projetos. Dito isso, considerando uma matriz quadrada M , os valores $M_{i,j}$ e $M_{j,i}$ terão a soma dos pesos das categorias compartilhadas entre o Projeto I e Projeto J .

O algoritmo que cria matrizes que representam uma *Score Voting Rule* a partir dos dados de eleições reais disponibilizados pela Pabulib funciona da seguinte forma. Será gerada uma matriz como descrito anteriormente que então será multiplicada por um vetor contendo a quantidade de votos para cada projeto. Portanto, é importante que $M_{i,i}$ seja maior que o valor absoluto de qualquer peso da categoria:

1. Conte de quantas aparições da categoria em cada projeto, definindo como peso negativo da categoria;
2. Some dos pesos de cada categoria em comum que os projetos têm entre si e o preencha da matriz com esses valores;
3. Tendo Min como o menor peso de categoria (ou seja, o peso da categoria com mais aparições), preencha a diagonal da matriz com $|Min| + 1$.

Exemplo:

| | |
|--|--------------------------|
| Projeto $I = \{\text{cultura, educação}\}$ | Pesos: cultura = -3 |
| Projeto $J = \{\text{cultura, esporte}\}$ | esporte = -2 |
| Projeto $K = \{\text{cultura, esporte}\}$ | educação = -1 |

$$M = \begin{pmatrix} 6 & -3 & -3 \\ -3 & 6 & -5 \\ -3 & -5 & 6 \end{pmatrix}$$

Como trabalho futuro, temos a intenção de estudar diferentes estratégias de algoritmos para criar automaticamente *Score Voting Systems* baseados em dados reais de eleições presentes em repositórios como Pabulib.

Referências

- Barberà, S. and Jackson, M. (1994). A characterization of strategy-proof social choice functions for economies with pure public goods. *Social Choice and Welfare*, 11(3):241–252.
- Brandt, F., Conitzer, V., Endriss, U., Lang, J., and Procaccia, A. (2016). *Introduction to Computational Social Choice. Handbook of Computational Social Choice*. New York, USA: Cambridge University.
- Briest, Krysta, V. (2005). Approximation techniques for utilitarian mechanism design. *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 39–48.
- Cohen, J., Cordeiro, D., Dardilhac, V., and Glaser, V. (2022). Designing strategyproof election systems with score voting. arXiv:2210.02496 [cs.GT].
- Cordeiro, D. (2012). *The impact of cooperation on new high performance computing platforms*. PhD thesis, Université de Grenoble.
- Duddy, C. (2015). Fair sharing under dichotomous preferences. *Mathematical Social Sciences*, 73:1–5.
- Gibbard, A. (1973). Manipulation of voting schemes: a general result. *Econometrica: journal of the Econometric Society*, pages 587–601.
- Goel, A., Krishnaswamy, A. K., Sakshuwong, S., and Aitamurto, T. (2019). Knapsack voting for participatory budgeting. *ACM Transactions on Economics and Computation (TEAC)*, 7.
- Peters, D., Pierczyński, G., and Skowron, P. (2021). Proportional participatory budgeting with additive utilities. *Advances in Neural Information Processing Systems*, 34:12726–12737.
- Stolicki, D., Szufa, S., and Talmon, N. (2020). Pabulib: A participatory budgeting library. *arXiv preprint arXiv:2012.06539*.