

Escalonamento Descentralizado Multiagente de Recursos em Nuvem Utilizando Blockchain

Wilton Jaciel Loch ¹, Guilherme Piêgas Koslovski ¹

¹Programa de Pós-Graduação em Computação Aplicada (PPGCA)
Universidade do Estado de Santa Catarina (UDESC) - Joinville, SC - Brasil

1. Introdução e definição do problema

O paradigma de computação em nuvem ganhou popularidade como uma escolha viável para as necessidades de recursos computacionais entre os mais diversos meios. Internamente, nos aglomerados dos provedores, os serviços de computação fornecidos de forma virtualizada devem ser escalonados ou alocados em estruturas físicas que os comportem. Esse processo de escalonamento é crucial tanto para a manutenção da qualidade do serviço, quanto para a diminuição dos custos por parte do provedor. Entretanto, muitos dos algoritmos de escalonamento atuais lidam apenas com um conjunto homogêneo de recursos computacionais ou falham em tratar as requisições em velocidades aceitáveis para cenários reais [Rodriguez and Buyya 2018]. Embora existam algoritmos eficientes para determinados contextos, a otimalidade da utilização de recursos heterogêneos dadas variadas restrições, formas de serviço e necessidades específicas dos clientes, com tomadas de decisão aceleradas ainda é um desafio.

2. Proposta

Nesta situação, a presente proposta busca desenvolver um algoritmo de escalonamento descentralizado e multiagente, primordialmente fundamentado no conceito de *Blockchain*. Ao invés do processo de tomada de decisão ser realizado de forma centralizada por um único agente controlador, as escolhas a respeito da alocação ou movimentação das requisições de serviço são feitas por um conjunto finito predeterminado de agentes independentes atuando em paralelo, cujo comportamento social integrado constitui um sistema emergente capaz de realizar as funções esperadas de maneira eficiente.

Primeiramente, a utilização de *Blockchain* é justificada por tratar-se de uma forma eficiente de manter um conjunto de dados íntegros em um contexto descentralizado, com ordenamento temporal coeso, bem como rapidez no tratamento das informações e resiliência complementar à própria descentralização [Antonopoulos 2014]. No cenário abordado pela proposta, a cadeia é utilizada como meio de compartilhamento e sedimentação de informações importantes para a tomada de decisão dos participantes e, de maneira geral, como forma de comunicação assíncrona entre os mesmos. Os agentes em suma, são manifestações ativas dos recursos físicos de um *data center* e naturalmente a sua quantidade é variável de acordo com o tamanho dos aglomerados ou a quantidade de máquinas ativas em um dado momento.

O conceito principal por trás da existência desses agentes é inverter a hierarquia da tomada de decisão nas alocações, e no lugar de trabalhos serem escolhidos para estruturas físicas passivas, as últimas são responsáveis pela escolha dos próprios trabalhos, utilizando como base informações estáticas e dinâmicas de outros agentes como alicerce para

um comportamento altruísta que visa a melhor divisão de trabalho entre a comunidade, dadas naturalmente as diferentes características individuais dos recursos.

Um aglomerado pode então ser visto como um conjunto de trabalhadores diversos que representam uma abstração das estruturas físicas componentes do mesmo. Cada trabalhador pode possuir características físicas distintas - que no paralelo com recursos computacionais podem ser vistas como um perfil de *hardware* - que o tornam mais ou menos adequado a certos tipos de trabalho, sendo essas informações compartilhadas com os outros trabalhadores. Um trabalhador é também um agente independente capaz de tomar decisões a respeito da aceitação ou não das tarefas que lhe são apresentadas, sem a existência de um mestre que determine-o em seu intermédio. Nesse sentido, todos os novos trabalhos são dispersados entre diferentes trabalhadores e cada um deve escolher se irá executá-lo ou se deixará a tarefa a encargo de outro agente conhecido. Tal escolha é feita apenas com base no conhecimento que um trabalhador possui sobre si mesmo e sobre os outros, de forma que se há outro indivíduo mais adequado para a realização daquele trabalho, então um trabalhador qualquer que não este irá recusar a execução prevendo que o melhor a aceitará. Conhecendo, por exemplo, o consumo energético próprio e dos outros trabalhadores e recebendo uma proposta de trabalho com foco em economia de energia, um trabalhador qualquer apenas executará o trabalho se possuir a maior eficiência energética dentre seus vizinhos, caso contrário o deixará para o trabalhador que a possui. Todo esse processo pode ser simplesmente entendido como uma contínua liberação dos trabalhos por parte dos agentes menos habilitados para a tomada dos mais habilitados em certo aspecto.

Como não há nenhuma forma de controle centralizado, a drástica redução na probabilidade de conflito entre as decisões de dois trabalhadores distintos é dada pela concordância a respeito do conjunto de informações compartilhado por ambos em seu grupo - ressaltando a importância da utilização de Blockchain - e pelo mesmo padrão de comportamento altruísta empregado por estes.

3. Considerações Finais

A utilização de uma modelo descentralizado e multiagente pode advir diversas vantagens sobre modelos tradicionais de escalonamento, entre elas a agilidade no tratamento das requisições e escalabilidade, decorrentes principalmente da propriedade distribuída do modelo; maior resiliência, dada a inexistência de um único ponto de falha central; e possivelmente a utilização de mais critérios específicos no processo de escalonamento, tanto das requisições quanto das estruturas físicas. O projeto encontra-se em fases iniciais e o desenvolvimento dos algoritmos bem como seus testes são trabalhos futuros.

Referências

- Antonopoulos, A. M. (2014). *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*. O'Reilly Media, Inc., 1st edition.
- Rodriguez, M. A. and Buyya, R. (2018). Containers orchestration with cost-efficient autoscaling in cloud computing environments. *CoRR*, abs/1812.00300.