

Explorando a Elasticidade em Nível de Sistema Operacional

Valquíria Prestes Belusso¹, Guilherme Galante¹

¹Ciência da Computação – Universidade Estadual do Oeste do Paraná (Unioeste)
Cascavel – PR – Brasil

valquiria.belusso@unioeste.br, guilherme.galante@unioeste.br

Resumo. Este trabalho tem como objetivo desenvolver uma camada de abstração integrado ao sistema operacional para o controle da elasticidade oferecida pela computação em nuvem. Considerando que o sistema operacional tem conhecimento sobre todas as aplicações que estão sendo executadas e suas demandas por recursos, pode-se coletar as informações necessárias para controlar a elasticidade de modo rápido e acurado e de modo transparente. Dessa forma, algumas limitações encontradas em soluções propostas no estado-da-arte podem ser mitigadas.

1. Introdução

A elasticidade é uma das principais características da computação em nuvem e pode ser definida como a capacidade de um sistema de adicionar ou remover dinamicamente recursos computacionais utilizados por uma determinada aplicação ou usuário de acordo com a demanda [Herbst et al. 2013]. Na prática, a elasticidade é implementada em nuvens utilizando tecnologias de virtualização, que permitem que recursos virtualizados possam ser fornecidos rapidamente e de acordo com a carga de trabalho. Os recursos podem incluir desde CPUs virtuais, memória, e armazenamento, até máquinas virtuais completas [Bauer and Adams 2012]. A Figura 1 ilustra o comportamento de um serviço com uma alocação de carga estática e outra dinâmica, ou elástica.

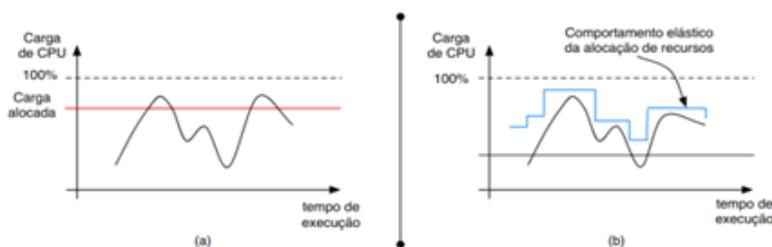


Figura 1. Provisionamento estático versus provisionamento dinâmico.

A alocação estática tenta provisionar recursos sempre para o pior caso (momentos de pico), correndo o risco de essa previsão ser subestimada, como mostra a Figura 1(a) em dois pontos (cargas acima da linha de provisionamento). Ainda, há a questão do desperdício de recursos, dado que momentos de pico não caracterizam a carga média no sistema. A Figura 1(b) mostra o provisionamento elástico de recursos computacionais. Como pode-se observar, recursos são adicionados ou liberados dinamicamente à máquina virtual, mantendo a capacidade do sistema compatível com a carga de trabalho, e minimizando a ociosidade de recursos e problemas de provisionamento.

De acordo com o estado-da-arte do assunto [Ullah A 2018, Singh et al. 2019, Arunkumar et al. 2020], a elasticidade oferecida pelos mecanismos atuais baseia-se em duas abordagens. A primeira abordagem baseia-se no monitoramento das solicitações externas ou no uso de recursos (carga do processador, uso de memória, solicitações de E/S) que podem variar amplamente ao longo do tempo. Os dados de monitoramento são empregados por um controlador de elasticidade que toma decisões sobre se os recursos devem ser escalados ou não, levando em consideração um conjunto de condições que, quando satisfeitas, acionam algumas ações sobre a nuvem subjacente. Na segunda abordagem, o controle da elasticidade é feito em nível de programação, ou seja, o controlador de elasticidade é incorporado ao código-fonte da aplicação, permitindo que as ações de alocação e desalocação de recursos partam da própria aplicação, sem a necessidade de mecanismos externos ou de interação com o usuário.

Uma questão relevante aqui para escolher uma das abordagens da elasticidade é: qual é o nível de abstração exigido pelo usuário para habilitar a elasticidade da nuvem em sua aplicação? Se o usuário deseja usar a elasticidade de uma forma transparente e sem esforço, a abordagem baseada em monitoramento parece a mais adequada. Caso o usuário pretenda obter o controle total da execução, incluindo métricas, parâmetros para inserir chamadas de elasticidade, a melhor abordagem é o nível de programação. No entanto, ambas as abordagens possuem limitações. O uso de monitores externos não é apropriado para algumas classes de aplicações e nem sempre conseguem coletar as informações necessárias de forma efetiva e rápida, ocasionando latência na alocação dos recursos necessários. Por outro lado, a abordagem em nível de programação, embora eficiente, necessita que a aplicação seja reestruturada para a inclusão das ações de elasticidade, exigindo conhecimento profundo da aplicação, muitas vezes complexa, e o acesso ao seu código-fonte, que nem sempre é possível.

Uma abordagem possível para mitigar as limitações das soluções atuais é mover o controle da elasticidade para o sistema operacional (SO). O sistema operacional é a camada de software que abstrai e coordena o uso dos recursos de um sistema computacional (processador, memória, armazenamento, etc.), bem como gerencia a execução de todas as aplicações do usuário [Tanenbaum and Bos 2014]. Assim, esse trabalho visa desenvolver uma camada de abstração da elasticidade que será integrada ao SO para o controle da elasticidade. Considerando que o SO tem informações sobre todas as aplicações que estão sendo executadas e suas demandas por recursos, pode-se coletar as informações necessárias para controlar a elasticidade de modo rápido e acurado, e gerenciar a alocação elástica de recursos de modo transparente.

A proposta tem como trabalhos mais próximos os desenvolvidos por [Imai et al. 2012] e [Ababneh et al. 2018]. No primeiro, os autores apresentam o *Cloud Operating System*, o qual baseia-se na instanciação de novas máquinas virtuais e migração de aplicações para garantir o desempenho. O segundo trabalho apresenta o ElasticOS, um conjunto de primitivas de sistema operacional que permitem que sejam usados recursos de diversos nós físicos conforme as necessidades de recursos vão além do disponível em uma única máquina. A abordagem proposta difere dos dois trabalhos na forma de alocação de recursos e em sua implementação.

2. Objetivos

Este trabalho tem como objetivo propor uma abordagem alternativa para a exploração de elasticidade para aplicações que não se adequam aos mecanismos encontrados no estado da arte atual. Nesta abordagem, o controle da elasticidade passa a ser um componente do sistema operacional, de modo que a coleta de informações e a execução das ações de elasticidade sejam gerenciadas de forma transparente ao usuário, e ao mesmo tempo, garantindo-se a eficiência e acurácia na alocação dos recursos.

Mais especificamente, será projetado e implementado um controlador de elasticidade para ambientes constituídos de máquinas virtuais utilizando o hipervisor Xen sobre sistemas operacionais Linux, como ilustrado na Figura 2. A camada de elasticidade coletará informações provenientes do Linux da máquina virtual, tais como, uso atual de CPU (%), quantidade de memória alocada/usada, informação de novas alocações de memória, etc., e baseada nesses dados, realizará a adequação dos recursos por meio de requisições ao hipervisor Xen.

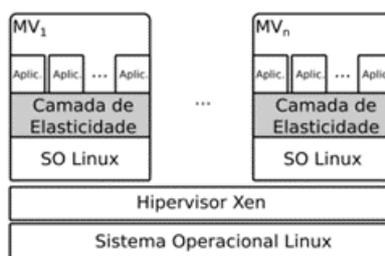


Figura 2. Arquitetura da solução proposta.

Nessa primeira etapa do projeto, os recursos a serem administrados serão memória e CPU. No caso da elasticidade de memória, ela é percebida de maneira transparente pelas aplicações. Por outro lado, as aplicações deverão ser instrumentadas para se beneficiarem da elasticidade de CPU.

3. Material e Métodos

Como apresentado nos objetivos, o controlador de elasticidade baseia-se no hipervisor Xen e no sistema operacional Linux. O Xen é um hipervisor gratuito e de código aberto que permite que um computador execute vários sistemas operacionais simultaneamente no mesmo hardware na forma de máquinas virtuais [Xen 2021]. O Xen foi escolhido por possuir o recurso de *hotplug* de vCPU, o qual permite aumentar dinamicamente o número de CPUs virtuais atribuídas a uma VM Linux em execução, sem precisar reiniciar a VM. Ele permite também que a memória seja redimensionada (para mais ou menos) por meio da técnica de *balloning*.

Para a construção do controlador de elasticidade proposto, foi necessário compreender como o sistema operacional fornece as informações dos recursos a serem administrados (CPU e memória). Já para a análise da comunicação entre as máquinas virtuais (chamadas de DomU, na nomenclatura do Xen) e o hipervisor (Dom0), foi decidido pela utilização do XenStore, que é um sistema de armazenamento que é compartilhado entre os *Guest Domains* (GDs) e é mantido pelo Dom0. Para se comunicar, os GDs e o Dom0 leem e escrevem no XenStore [XenStore 2021].

Após compreender os detalhes técnicos, será realizado o projeto da camada de elasticidade a ser integrada no Sistema Operacional. Nessa etapa, a arquitetura será refinada e os protocolos de comunicação entre as partes serão definidos, bem como os protocolos de alocação elástica de recursos. Detalhes de implementação, tais como, definição da linguagem de programação, uso de bibliotecas e frameworks também serão definidos nesta etapa.

Com o projeto elaborado, será implementado de acordo com as especificações. Deverão ser implementados mecanismos para a coleta das informações do SO, processamento das informações e comunicação MV-hipervisor para a requisição de recursos. Por fim, a exploração da elasticidade em nível de sistema operacional será validada por um conjunto de experimentos, nos quais a abordagem proposta será usada na alocação dinâmica de processadores virtuais e memória.

Referências

- Ababneh, E., Al-Ali, Z., Ha, S., Han, R., and Keller, E. (2018). Elasticizing linux via joint disaggregation of memory and computation. *CoRR*, abs/1806.00885.
- Arunkumar, S., Kumar, B. V., and Pandi, M. (2020). Artificial bee colony optimization based energy-efficient wireless network interface selection for industrial mobile devices. *Comput. Commun.*, 154:111–117.
- Bauer, E. and Adams, R. (2012). *Reliability and Availability of Cloud Computing*. Wiley-IEEE Press, 1st edition.
- Herbst, N. R., Kounev, S., and Reussner, R. (2013). Elasticity in cloud computing: What it is, and what it is not. In *10th International Conference on Autonomic Computing (ICAC 13)*, pages 23–27, San Jose, CA. USENIX Association.
- Imai, S., Chestna, T., and Varela, C. A. (2012). Elastic scalable cloud computing using application-level migration. In *2012 IEEE Fifth International Conference on Utility and Cloud Computing*, pages 91–98.
- Singh, P., Gupta, P., Jyoti, K., and Nayyar, A. (2019). Research on auto-scaling of web applications in cloud: Survey, trends and future directions. *Scalable Computing: Practice and Experience*, 20:399–432.
- Tanenbaum, A. S. and Bos, H. (2014). *Modern Operating Systems*. Pearson, Boston, MA, 4 edition.
- Ullah A, Li J, S. Y. . H. A. (2018). A control theoretical view of cloud elasticity: taxonomy, survey and challenges. *Cluster Computing 21*, pages 1735–1764.
- Xen (2021). Xen project. <https://xenproject.org/>, April.
- XenStore (2021). Xen store reference. <https://wiki.xenproject.org/wiki/XenStore.Reference>, April.