

Exploração de Elasticidade em Contêineres Docker

Gustavo Rafael De Paris Chiossi¹, Guilherme Galante¹

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

gustavo.chiossi@unioeste.br, guilherme.galante@unioeste.br

Resumo. *Várias soluções de elasticidade foram desenvolvidas e implementadas para ambientes baseados em máquinas virtuais, no entanto, a exploração de métodos de elasticidade para ambientes computacionais baseados em contêineres ainda não está solucionada. Nesse contexto, o presente projeto visa o estudo e a implementação de técnicas para permitir a alocação automática e dinâmica de recursos em contêineres Docker. Como resultado, espera-se validar o mecanismo implementado e analisar como essa característica pode impactar no desempenho e no comportamento das aplicações containerizadas.*

1. Introdução

A elasticidade em ambientes de computação em nuvem é uma característica fundamental que permite o dimensionamento dinâmico de recursos computacionais de acordo com a demanda de serviços. Em um cenário onde a carga de trabalho varia de forma imprevisível, a elasticidade oferece uma solução eficiente para ajustar automaticamente os recursos, garantindo que o desempenho do sistema seja mantido enquanto se otimiza o custo operacional [Herbst et al. 2013]. Esse conceito é particularmente relevante em contextos modernos de computação, como containers Docker, que combinam leveza, portabilidade e eficiência para a execução de aplicações em ambientes dinâmicos.

Containers Docker são uma das tecnologias mais adotadas para virtualização de processos e serviços, permitindo o empacotamento e a execução de aplicações de maneira isolada, com menor sobrecarga em comparação a outras abordagens de virtualização [Dogani et al. 2023].

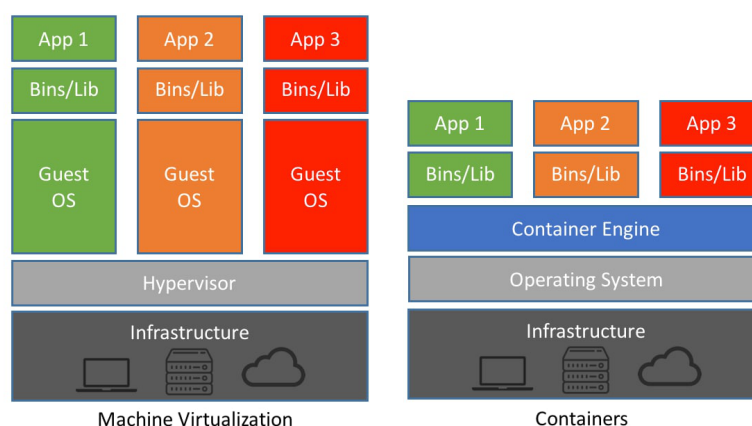


Figura 1. Containers Docker vs VM's.

Sua flexibilidade e facilidade de escalabilidade tornam-nos ideais para ambientes de computação em nuvem, onde a necessidade de ajustar recursos de forma automática é constante. A elasticidade de containers, portanto, refere-se à capacidade desses sistemas de ajustar os recursos computacionais — como CPU e memória — de maneira a responder a flutuações na carga de trabalho, otimizando tanto o desempenho quanto o uso dos recursos [Jiang and Duan 2024].

O presente trabalho visa analisar a elasticidade de containers Docker, com foco em como os recursos de CPU e memória são alocados e ajustados em resposta a variações de carga. A pesquisa busca entender o comportamento de containers em cenários de alta e baixa demanda, utilizando uma combinação de ferramentas e metodologias para medir e registrar esses dados. Essa análise se torna ainda mais relevante quando se considera que a elasticidade eficaz pode reduzir custos operacionais e melhorar o desempenho das aplicações [Rossi et al. 2019].

Tendo isso em vista, a análise da elasticidade de containers Docker é um passo crucial para o avanço de metodologias que busquem otimizar o uso de recursos em ambientes computacionais altamente dinâmicos e escaláveis, trazendo benefícios tanto para o desempenho quanto para a redução de custos operacionais [Sheganaku et al. 2023]. A partir da combinação de ferramentas de monitoramento e análise de dados, a pesquisa busca fornecer uma visão aprofundada sobre os fatores que influenciam a elasticidade e como ela pode ser melhorada para atender às necessidades dos sistemas modernos de computação em nuvem.

2. Objetivos

O principal objetivo desta pesquisa é entender o comportamento dos containers Docker em cenários elásticos, com um foco específico primeiramente na alocação e ajuste dinâmico de recursos como CPU e memória. Em um ambiente de computação em nuvem, onde as cargas de trabalho podem variar consideravelmente, a elasticidade dos containers é um fator crucial para garantir a eficiência operacional e o desempenho das aplicações. Com base nesse contexto, a pesquisa busca:

Avaliar a capacidade dos containers de se adaptar a variações de carga: A elasticidade é frequentemente medida pela capacidade de um sistema de responder adequadamente a flutuações na carga de trabalho, ajustando seus recursos de maneira eficiente [Herbst et al. 2013]. Este objetivo investiga como os containers Docker reagem a aumentos e diminuições repentinas na demanda, explorando a precisão e a rapidez do ajuste dinâmico em comparação com outras tecnologias de virtualização, além das técnicas de autoescalamento aplicáveis [Dogani et al. 2023].

Gerar dados que possam subsidiar a otimização de arquiteturas baseadas em containers: A coleta e análise de dados sobre o comportamento dos containers Docker são essenciais para melhorar a elasticidade e otimizar a infraestrutura de nuvem. As informações geradas podem subsidiar a criação de melhores práticas para a implementação de containers, com foco na eficiência no uso de recursos e no desempenho das aplicações.

3. Material e Métodos

A pesquisa foi estruturada em etapas claras e metodológicas para garantir a obtenção de dados confiáveis e relevantes. Primeiramente, foi realizada a configuração do ambiente, utilizando containers Docker para criar ambientes controlados e simulados. Cada container foi configurado para executar aplicações que geram cargas variáveis, possibilitando o estudo da elasticidade em diferentes cenários. O uso de containers Docker é amplamente reconhecido por sua leveza e eficiência, características que tornam essa tecnologia ideal para ambientes de computação dinâmica e escalabilidade [Dogani et al. 2023].

Em seguida, foi utilizado o Glances, uma ferramenta de monitoramento em tempo real, devido à sua capacidade de fornecer dados detalhados sobre o uso de CPU, memória e outros recursos. O Glances se destaca por sua versatilidade e por permitir o monitoramento remoto e a coleta contínua de métricas [Herbst et al. 2013], o que o torna uma escolha adequada para a coleta de dados em experimentos de elasticidade.

A coleta de dados foi realizada durante a execução dos containers que operavam sob diferentes cargas de trabalho. A API do Glances foi integrada para capturar esses dados programaticamente e registrá-los em planilhas, facilitando a análise posterior. Para automatizar esse processo, foi desenvolvido um script em Python que consulta a API em intervalos regulares e organiza as informações de forma estruturada, garantindo maior precisão, eficiência e relevância no processamento e na visualização dos resultados.



```
# Coleta as métricas do sistema (CPU e memória) usando o Glances

await data.get_metrics("cpu")
cpu_data = data.values
cpu_relevant = {
    'total': cpu_data.get('total'),
    'user': cpu_data.get('user'),
    'system': cpu_data.get('system'),
    'idle': cpu_data.get('idle')
}

await data.get_metrics("mem")
memory_data = data.values
memory_relevant = {
    'total': memory_data.get('total'),
    'available': memory_data.get('available'),
    'used': memory_data.get('used'),
    'percent': memory_data.get('percent')
}
```

Figura 2. Coleta de dados do script em Python.

Os dados registrados foram, então, processados e analisados para identificar padrões de consumo e avaliar como os containers respondem a mudanças na demanda. A partir dessa análise, métricas foram geradas para visualização e interpretação dos resultados. O uso de técnicas de visualização de dados, como gráficos e métricas de desempenho, é crucial para a compreensão dos efeitos da elasticidade e para a identificação de áreas de melhoria [Sheganaku et al. 2023].

4. Resultados Esperados

Espera-se que a coleta e análise de dados sobre o uso de CPU e memória nos contêineres Docker forneçam informações detalhadas sobre a elasticidade do sistema. A partir desses dados, será possível identificar padrões de consumo, avaliar a eficiência da alocação de recursos e compreender como os contêineres respondem a variações na carga de trabalho. Os experimentos devem demonstrar o impacto da elasticidade na otimização do uso de recursos, contribuindo para a redução de desperdícios e a melhoria do desempenho das aplicações containerizadas.

Com base nesses estudos e na pesquisa sobre técnicas de elasticidade, será possível desenvolver um protótipo funcional para alocação dinâmica de recursos na plataforma Docker. Esse protótipo será composto por três módulos principais: coleta de métricas, tomada de decisão e comunicação com o Docker. O módulo de coleta de métricas monitorará CPU e memória em tempo real, utilizando a API do Glances ou métricas nativas do Docker. O módulo de tomada de decisão processará as métricas e definirá estratégias de alocação de recursos. O módulo de comunicação com o Docker será responsável por executar comandos, via API do Docker, para modificar a alocação de CPU e memória conforme necessário, garantindo os recursos necessários para que aplicações containerizadas executem com bom desempenho.

O controlador de elasticidade será validado por meio de experimentos com aplicações containerizadas, incluindo cargas sintéticas e científicas, permitindo avaliar sua eficácia e impacto na eficiência operacional dos contêineres.

Como trabalhos futuros, pretende-se explorar cenários mais complexos, como ambientes de alta disponibilidade e sistemas distribuídos, o que permitirá uma aplicação mais direcionada e eficiente das técnicas desenvolvidas. Além disso, será avaliada a integração com outras plataformas de orquestração, como o Kubernetes. Também será feito a comparação, por meio de benchmarks, do protótipo criado com as demais ferramentas de elasticidade presentes no mercado.

Referências

- Dogani, J., Namvar, R., and Khunjush, F. (2023). Auto-scaling techniques in container-based cloud and edge/fog computing: Taxonomy and survey. *Computer Communications*, 209:120–150.
- 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.
- Jiang, C. and Duan, Y. (2024). Elasticity unleashed: Fine-grained cloud scaling through distributed three-way decision fusion with multi-head attention. *Information Sciences*, 660:120127.
- Rossi, F., Nardelli, M., and Cardellini, V. (2019). Horizontal and vertical scaling of container-based applications using reinforcement learning. In *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, pages 329–338.
- Sheganaku, G., Schulte, S., Waibel, P., and Weber, I. (2023). Cost-efficient auto-scaling of container-based elastic processes. *Future Generation Computer Systems*, 138:296–312.