

Introdução

A computação em nuvem oferece recursos, como processamento, armazenamento e rede, através de máquinas virtuais (VMs), conforme a necessidade do usuário. O uso de recursos em nuvem tem crescido há alguns anos. Atualmente, a comunidade científica também tem apresentado interesse nas utilização de nuvens computacionais, pela disponibilidade de uma variedade grande de recursos, que podem ser utilizados sob demanda, pagando-se apenas o que for utilizado. Tipicamente, os recursos de infra estrutura da nuvem são utilizados através de máquinas virtuais. No entanto, mais recentemente a utilização de contêineres vem crescendo e ganhando popularidade devido às suas características.. Os contêineres são considerados ferramentas de virtualização mais leves do que as máquinas virtuais.

Nesse sentido, os provedores de serviços em nuvem de hoje também estão oferecendo a implantação de contêineres. (por exemplo, Docker, LXC, etc.). O conceito de contêiner é semelhante à VM, mas consome comparativamente menos tempo e recursos. Na containerização, o mesmo kernel está sendo compartilhado entre os contêineres e o sistema operacional da máquina física; daí estes serem mais leves em comparação com as VMs. Na containerização, os componentes de hardware e software são compartilhados entre a máquina física, o sistema operacional (SO) e os aplicativos dos contêineres. O sistema operacional da máquina física é o principal responsável para garantir o isolamento entre as aplicações dos contêineres.

O desempenho do ambiente não virtualizado difere do ambiente virtualizado devido às interações das máquinas virtuais com a camada de abstração chamada VMM (virtual manager monitor) e bibliotecas de virtualização por partes de interfaces de contêineres. Apesar de contêineres serem mais leves e fazerem o uso do espaço do usuário ao lado do espaço usuário padrão de um sistema operacional, podemos fazer uma comparação entre a execução de tarefas que rodam em sistemas operacionais diretamente sobre o hardware físico e contêineres sobre o mesmo sistema operacional, pois a arquitetura de acesso do kernel linux são diferentes; E seguindo esse raciocínio, através de diversos experimentos conheceremos o desempenho das interfaces de virtualização, ocasionando uma melhor alocação de recursos, e assim, melhorando também a eficiência na execução de tarefas.

Dessa forma, esse projeto propõe uma análise do desempenho de vários benchmarks quando executados em diversos ambientes de contêineres com o objetivo de auxiliar a compreensão do efeito da containerização no tempo de execução de aplicações dos mais diversos perfis.

Objetivo

A proposta deste trabalho é avaliar o desempenho de um conjunto de contêineres como docker, containerd, singularity e rkt utilizando aplicações representativas de diversas áreas, fazendo o uso de benchmarks bem consolidados no meio acadêmico, tanto na nuvem como em uma máquina física..

Dentre os benchmarks adotados, as categorias escolhidas são:

- FIO, o benchmark de I/O de um container e cluster de contêineres
- Paradnn, benchmark de deep learn com CPU, GPU e TPU
- BigDataBench, benchmark de big data o mais flexível que atende mais tipos de SGBD distribuídos
- TPC-C (Transaction Process Council) benchmark de banco de dados relacional
- NAS Parallel Benchmarks - benchmark de CPU criado pela nasa
- Iperf - Benchmark de redes entre containers

Com as aplicações para benchmark e os tipos de contêineres definidos, o próximo passo será testar o comportamento de cada uma dessas tecnologias na execução de benchmarks relacionados. Utilizaremos também os seguintes orquestradores de contêineres candidatos aos testes on-premise são kubernetes(K8S), mini-kube(k3s), docker swarm e em nuvem serão Elastic Container Service(ECS), Google Kubernetes Engine (GKE) e Azure Kubernetes Service (AKS)

Dessa maneira, serão avaliados diferentes aplicações e ambientes com contêineres e orquestradores, de forma a representar as principais cargas de trabalho científicas e utilização de softwares utilizados no mercado.

Metodologia

Os testes iniciais deverão ser realizados no [meu notebook](#), hoje com 32Gb de ram, com linux mint instalado de kernel 5.14.0-1027-oem.

Se os resultados forem satisfatórios, os novos testes ocorrerão em provedores de nuvem, caso os resultados não sejam satisfatórios, novos testes deverão ocorrer em outra máquina, podendo ser da UFF e provedor de nuvem. Se mesmo assim os resultados não forem satisfatórios, uma revisão bibliográfica será feita e os dados serão publicados no TCC indicando alguma divergência com a teoria.

Cada teste deve ser realizado por pelo menos 10 vezes com um cálculo de média e desvio padrão entre resultados para melhor precisão, e em caso de variância mais alta haverá mais uma rodada de teste. Todos os softwares e interfaces de container serão utilizados na sua versão mais recente possível, em caso de incompatibilidade de versão poderá haver testes de conceito em máquinas virtuais utilizando primeiramente o meu notebook e posteriormente outras máquinas que possuem uma versão menor de sistema operacional, bibliotecas e versão do kernel. Apesar disso, todas as comparações de software terão como o foco principal uma comparação com o mesmo sistema que irá rodar o benchmark para que a comparação seja mais justa possível com as interfaces de contêineres.

O teste em nuvem deverá ocorrer sobre máquinas com interfaces de containeres instalados e suas redes definidas, ou seja, vms em nuvem. Dessa forma, não serão utilizados serviços de contêiner oferecidos por provedores de nuvem, pois assim como ECS, AKS e entre outros, todos eles possuem um padrão baseado em kubernetes e não possuem um acesso remoto que permite configuração de dentro do container. Nesse sentido, configurar um benchmark neste tipo de serviço seria árduo, pois seria necessário o desenvolvimento de sistema que atende às requisições por API e retornando os resultados logo após.

Com isso, com todos os resultados prontos, o teste em nuvem poderá ser mais abrangente, com mais famílias distintas de máquinas virtuais oferecidas pelo seu provedor e assim, obter mais informações sobre o comportamento das máquinas conforme o tipo de benchmark.

Cronograma

Data de início: 28/03 (o mais breve possível ou a data de aprovação da proposta)

Testes em hosts locais

instalação das interfaces de contêineres e escrever sobre a parte teórica de cada uma das interfaces e construção de cluster de contêiner: 3 semanas (11/04)

Benchmark FIO e parte teórica: 2 semanas (29/04)

benchmark paradnn e escrita teórica: 2 semanas (13/05)

big data bench e escrita teórica: 3 semanas (3/06)

TPC-C e escrita parte teórica: 2 semanas (24/06)

NAS Parallel Benchmarks e escrita parte teórica: 3 semanas (15/07)

Iperf: 1 semana (22/07)

Testes na nuvem

Testes na aws: 2 semanas (05/08)

Testes em outro provedor de nuvem: 2 semanas (19/08)

Revisão e ajustes

Revisão de texto ou solução de problemas 4 meses (até 18/12)

Trabalhos Relacionados

Existe uma enorme gama de trabalhos relacionados sobre a comparação de tipos de virtualização e seus similares, que apresentam análise de desempenho através de benchmarks.

O trabalho de Arango et al. [1] fez uma comparação de desempenho de ferramentas de virtualização baseadas em contêiner (Docker, LXC, Singularity) contra bare-metal, e concluiu que os containers Singularity são geralmente mais adequados para implementação de HPC do que Docker ou LXC.

O trabalho de Chung et al. [2] compara Docker com VM utilizando KVM para aplicações intensivas em computação (Linpack) e intensivas em dados (Graph500) mostrando que o desempenho do Docker é superior para aplicações intensivas em dados.

Felter et al. [3] analisaram o desempenho de máquinas virtuais (KVM) e compararam com contêineres (Docker). Mostraram que os overheads de CPU e memória foram mínimos, tanto para máquinas virtuais quanto para contêineres. Porém, essas tecnologias encontraram dificuldades quando a carga trabalho foi intensiva em E/S, pelo menos em 2015..

Bhimani et al. [4] investigaram o desempenho de diferentes aplicações do Apache Spark usando máquinas virtuais e contêineres do Docker. Esse estudo compara as diferentes estruturas de virtualização para um ambiente de nuvem empresarial de big data usando o Apache Spark. Além do makespan e do tempo de execução, também são analisadas diferentes utilizações de recursos (CPU, disco, memória etc) pelas aplicações Spark. Os resultados mostram que, usando o Docker, o Spark pode obter uma aceleração de mais de 10 vezes em comparação com o uso da máquina virtual. Essa aceleração foi devida ao recurso copy-on-write (COW) disponibilizado pelo Docker e driver específico para armazenamento que melhoram o desempenho de algumas aplicações Spark. Porém, para interações massivas de E/S aleatória não é recomendável o uso de container.

Younge et al. [5] utilizam os benchmarks HPCG e IMB para avaliar os overheads e a escalabilidade dos contêineres, usando Singularity, no sistema Cray XC30. Além disso, utilizaram o ambiente EC2 da Amazon para comparações. Os autores chegaram à conclusão de que os contêineres utilizando o Singularity atingem desempenho próximo ao nativo quando utilizam as bibliotecas MPI da CRAY, enquanto o EC2 da Amazon seria útil para o desenvolvimento inicial e teste de aplicações HPC.

Kon et al. [6] avaliaram o impacto do uso de contêiner, no caso o Docker, para o caso de muitos contêineres serem executados em um mesmo hardware. Os autores revelaram que o contêiner poderia alcançar o desempenho semelhante ao do sistema operacional do host, exceto em casos de processamento de E/S intensivo.

Da Silva et al. [16] Indicaram que as virtualizações baseadas em contêineres tiveram um desempenho melhor que a gerenciada por hypervisor. Porém, de modo geral, todos os ambientes tiveram um desempenho próximo ao nativo com uma diferença de no máximo 3% melhor, comparando contêineres e máquina física.

Links das aplicações:

FIO, o benchmark de I/O de um container e cluster de contêineres

Link da aplicação: <https://github.com/axboe/fio>

Link do artigo:

<http://nucsr.coe.neu.edu/sites/nucsr.coe.neu.edu/files/Papers/NVMe%20Camera%20Ready.pdf>

Paradnn, benchmark de deep learn com CPU, GPU e TPU

Link da aplicação: <https://github.com/Emma926/paradnn>

Link do artigo: <https://arxiv.org/abs/1907.10701>

BigDataBench, benchmarks de big data o mais flexível que atende mais tipos de SGBD distribuídos

Links da aplicação: <https://www.benchcouncil.org/BigDataBench/download.html>

Link do artigo: <https://arxiv.org/pdf/1802.08254.pdf>

benchmark banco de dados sql TPC-C (Transaction Process Council)

Link da aplicação:

https://www.tpc.org/tpc_documents_current_versions/current_specifications5.asp

Link do artigo:

<https://ri.unipac.br/repositorio/wp-content/uploads/2019/07/Erick-Rodrigues-Ferreira.pdf>

CPU com benchmark da nasa - NAS Parallel Benchmarks

Link da aplicação: <https://www.nas.nasa.gov/software/npb.html>

Link do artigo: <https://www.nas.nasa.gov/assets/pdf/techreports/1994/rnr-94-007.pdf>

Benchmark de redes entre containers - Iperf

Link da aplicação: <https://iperf.fr/iperf-download.php>

Link do artigo: <https://sci-hub.hkvisa.net/https://dl.acm.org/doi/abs/10.1145/3409334.3452040>

Referências bibliográficas

- [1] Arango, C.; Dernat, R.; Sanabria, J. Performance evaluation of container-based virtualization for high performance computing environments. arXiv preprint arXiv:1709.10140 (2017).
- [2] Chung, M. T.; Quang-Hung, N.; Nguyen, M.-T.; Thoai, N. Using docker in high performance computing applications. In Communications and Electronics (ICCE), 2016 IEEE Sixth International Conference on (2016), IEEE, pp. 52–57.
- [3] Felter, W.; Ferreira, A.; Rajamony, R.; Rubio, J. An updated performance comparison of virtual machines and linux containers. In Performance Analysis of Systems and Software (ISPASS), 2015 IEEE International Symposium On (2015), IEEE, pp. 171–172.
- [4] Bhimani, J.; Yang, Z.; Leeser, M.; Mi, N. Accelerating big data applications using lightweight virtualization framework on enterprise cloud. In High Performance Extreme Computing Conference (HPEC), 2017 IEEE (2017), IEEE, pp. 1–7.
- [5] Younge, A. J.; Pedretti, K.; Grant, R. E.; Brightwell, R. A tale of two systems: Using containers to deploy hpc applications on supercomputers and clouds. In Cloud Computing Technology and Science (CloudCom), 2017 IEEE International Conference on (2017), IEEE, pp. 74–81.
- [6] Kon, J.; Mizusawa, N.; Umezawa, A.; Yamaguchi, S.; Tao, J. Highly consolidated servers with container-based virtualization. In Big Data (Big Data), 2017 IEEE International Conference on (2017), IEEE, pp. 2472–2479.
- [7] DA SILVA, Rodrigo Alves Prado et al. Vetorização e Análise de Algoritmos Paralelos para a Migração Kirchhoff Pré-empilhamento em Tempo. In: Anais do XVIII Simpósio em Sistemas Computacionais de Alto Desempenho. SBC, 2017. p. 304-315.
- Mohan Jain, Shashank M. Jain Linux Containers and Virtualization: A Kernel Perspective. Apress, 2020.
- Asif Raza Shah, Syed, A R. S. et al, Appl. Sci. 2021, 11(3), 993;
<https://doi.org/10.3390/app11030993>, Academic Editor: Fabrizio Marozzo
- Bhimani, JJanki B. Understanding Performance of I/O Intensive Containerized Applications for NVMe SSDs. nucsrl.coe.neu.edu.
<http://nucsrl.coe.neu.edu/sites/nucsrl.coe.neu.edu/files/Papers/NVMe%20Camera%20Ready.pdf>
- W. Gao, J. Zhan, L. Wang, C. Luo, D. Zheng, F. Tang, B. Xie, C. Zheng, X. Wen, X. He, H. Ye, and R. Ren, “Data motifs: A lens towards fully understanding big data and ai workloads,”

Parallel Architectures and Compilation Techniques (PACT), 2018 27th International Conference on, 2018.

Yu (Emma) Wang, Gu-Yeon Wei and David Brook. Benchmarking TPU, GPU, and CPU Platforms for Deep Learning. arxiv.org. <https://arxiv.org/pdf/1907.10701.pdf>

Erick Rodrigues Ferreira¹ , Sergio M. Trad Junior. Analise de desempenho de Bancos de Dados <https://ri.unipac.br/>,
<https://ri.unipac.br/repositorio/wp-content/uploads/2019/07/Erick-Rodrigues-Ferreira.pdf>
2022

D Bailey E Barszcz J Barton D Browning R Carter L Dagum R Fatoohi S Fineberg P Frederickson T Lasinski R Schreiber H Simon V Venkatakrishnan and S. Weeratunga, RNR Technical Report RNR-94-007, March 1994, www.nas.nasa.gov,
<https://www.nas.nasa.gov/assets/pdf/techreports/1994/rnr-94-007.pdf>

Kun Suo, Yong Shi, Ahyoung Lee, Sabur Baidya. Characterizing Networking Performance and Interrupt Overhead of Container Overlay Networks. ACM Southeast Conference – ACMSE 2021 – Session 1: Full Papers – ISBN: 978-1-4503-8068-3 Virtual Event, USA, April 15-17, 2021. <https://sci-hub.hkvisa.net/https://dl.acm.org/doi/abs/10.1145/3409334.3452040>

Syed Asif Raza Shah ¹ , Ahmad Waqas ¹ , Moon-Hyun Kim ² , Tae-Hyung Kim ³ , Heejun Yoon ⁴ and Seo-Young Noh ², Benchmarking and Performance Evaluations on Various Configurations of Virtual Machine and Containers for Cloud-Based Scientific Workloads. <https://www.mdpi.com/2076-3417/11/3/993/pdf> acessado em 2022