Execução de uma Aplicação Apache Hadoop em Ambientes Virtualizados

Iago C. Corrêa[1], **Leonardo H. Steil**[1], **Elton Rasch**[1], **Andrea S. Charão**[1], **Patrícia Pitthan Barcelos**[1]

[1] Laboratório de Sistemas de Computação
Universidade Federal de Santa Maria – UFSM

{icorrea,lsteil,erasch,andrea,pitthan}@inf.ufsm.br

Resumo. *O presente trabalho visa estudar o desempenho na execução de aplicações para processamento de grandes quantidades de dados em ambientes virtualizados, utilizando o Apache Hadoop. Será analisado o tempo de execução entre uma máquina virtual comum e um container do software Docker onde, o segundo obteve um menor tempo de execução, o que permitirá uma breve discussão e análise sobre o tema proposto.*

1. Introdução

Com o crescimento constante do volume de dados produzidos pela sociedade, surgiu a necessidade de servidores manusearem esses registros de forma rápida e fluida. Para isso, comumente se utiliza ferramentas como o *Apache Hadoop* [Foundation 2014] para processar essa abundância de dados, que recebeu o nome de *Big Data*. E assim, para dispor de uma divisão mais eficiente dos recursos das máquinas e clusters para o processamento dos dados, as organizações acabam recorrendo à técnica de virtualização [IBM 2012].

Este estudo procura discutir técnicas de virtualização associadas ao processamento de grandes quantidades de dados, comparando os desempenhos desses processamentos, com o intento de identificar qual a melhor maneira de virtualização para lidar com a computação intensiva de dados.

2. Virtualização: Docker e Máquinas Virtuais

O processo de virtualização de um sistema baseia-se na criação de um ambiente virtual hóspede, parcial ou totalmente dependente de seu hospedeiro, para a execução de tarefas previamente definidas pelo usuário. A vantagem do conceito proposto pelo modelo é a versatilidade que o usuário dispõe de, ao particionar seu sistema computacional e torná-lo um hospedeiro, usufruir de vários ambientes virtuais que podem ou não ter atribuições e objetivos diferentes [Carissimi 2008].

Nesse contexto, frequentemente surgem novas tecnologias com diferentes propostas de virtualização. Apresentaremos nas seções seguintes duas formas distintas de virtualização às quais serão utilizadas no presente trabalho.

2.1. Máquinas Virtuais

O conceito de máquinas virtuais é antigo, proposto inicialmente na década de 70 por pesquisadores da IBM para ser utilizado em sua linha de *mainframes* [Tanenbaum 2009]. A estruturação das máquinas virtuais consiste na criação de uma cópia exata do hardware

do sistema hospedeiro em software, que será usada para atender ao sistema operacional hospedado.

Nessa abordagem, as máquinas virtuais são controladas por um supervisor que intercepta as chamadas de sistemas dos sistemas operacionais virtualizados e realiza o tratamento dessas chamadas simulando o comportamento de um hardware real.

2.2. Docker e containers

Docker [Inc 2015] é uma plataforma de código aberto, escrita em linguagem *GO*, que surge como uma proposta de virtualização atípica. Diferentemente das máquinas virtuais que trabalham em conjunto com um hardware virtualizado, o *Docker* e seus *containers* (estruturas computacionais empacotadas e isoladas) trabalham com o sistema operacional hospedeiro e, ainda, quando for propício, compartilham também rotinas de bibliotecas e partes do kernel. Os *containers* executam imagens feitas previamente pela comunidade que contêm softwares empacotados, ou seja, é possível criar apenas uma imagem para rodar em um *container* e replicá-la várias vezes conforme a necessidade. A Figura 1, mostra a diferença estrutural entre as tecnologias de virtualização.

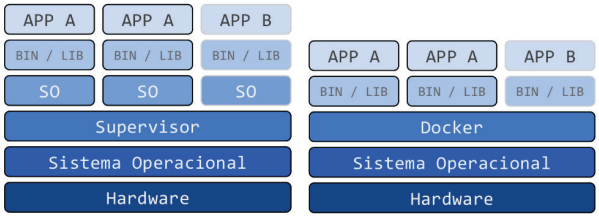

Figure 1. Máquina virtual (esquerda) e o Docker (direita).

3. Metodologia

Para avaliar o desempenho entre os ambientes virtualizados, utilizou-se o *Apache Hadoop*, um *framework* que implementa o paradigma de programação *MapReduce* [Dean and Ghemawat 2008]. A motivação na escolha dessa ferramenta reside no fato desta possibilitar um controle refinado sobre a carga nos ambientes virtuais, de modo que se torna possível submeter os ambientes a um nível de processamento onde fica clara a diferença entre os modelos estruturais de virtualização.

Os testes foram executados com o *Hadoop* na versão 2.7.3 utilizado o *Java* (versão 8), e o *Hadoop* foi executado em modo de operação pseudo-distribuído. A aplicação baseia-se na manipulação de dados a partir de satélites de monitoramento climático.

A entrada de dados, ou *input* da aplicação foi retirada do website da *National Aeronautics and Space Administration* [Administration 1958] que provê, de forma pública, medições feitas pelos satélites da instituição. O *output* da aplicação é composto por vários pares chave-valor onde a chave é uma coordenada geográfica, que tem por valor a média de ozônio observada naquela região.

3.1. Configuração da Máquina Virtual (VM)

A máquina virtual é composta por uma instalação mínima do *Debian* 8, cuja imagem contém apenas o necessário para executar o *Hadoop*. Assim, obtém-se uma virtualização leve para a execução das aplicações. O software utilizado para gerenciar, emular e monitorar a máquina virtual foi o *Oracle VM VirtualBox*, versão 5.1.8.

3.2. Configuração do *container*

A imagem utilizada para executar sobre o *container* tem o mesmo propósito da imagem da máquina virtual, i.e., ser o mais sucinta e leve possível. A imagem também é composta apenas por ferramentas necessárias para a execução do *Hadoop* e foi retirada do repositório de imagens para *containers* oficial do *Docker* [Hub 2015]. O *container* foi executado com o *Docker* na versão 1.12.3.

4. Experimentação

Foram executados dois experimentos para realizar a comparação entre os ambientes virtualizados. O primeiro, baseava-se em manter o mesmo tamanho de *input*, modificando a quantidade de execuções da aplicação. Como segundo experimento, foi modificado o tamanho do *input*, preservando apenas uma execução da aplicação para cada carga.

Para o primeiro experimento, foram executadas 1, 10, 100, e 1000 vezes a mesma aplicação. Conforme exibe a Figura 2, utilizando a máquina virtual, os tempos foram de aproximadamente 7 segundos para 1 execução, 30 para 10, 165 para 100 e 4037 para 1000 execuções. Já os tempos observados para o *container* foram melhores, sendo 4 segundos para 1 execução e, 16, 145, 2738 segundos para 10, 100, 1000 execuções, respectivamente.

Já para o segundo experimento, de acordo com a Figura 3, os testes registraram tempo de execução nas máquinas virtuais de 7 segundos para o tamanho original do arquivo (aproximadamente 3,5 MB), 12 para um arquivo de 10 MB, 28 para um arquivo de 100 MB e 175 para um arquivo de 1000 MB. Para o *container* foi observado os tempos de 4 segundos para o tamanho original, 6 para 10 MB, 18 para 100 MB e 149 para 1000 MB. É notável, novamente, que os tempos registrados para o *container* são menores.

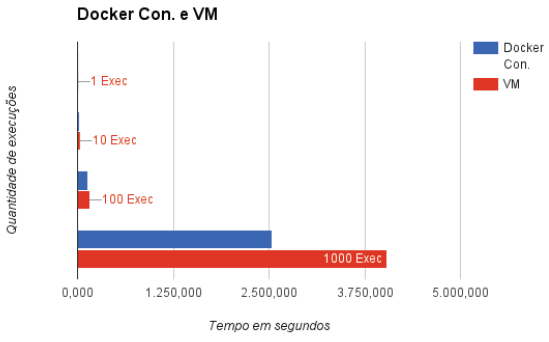

Figure 2. Tempo de execução vs quantidade de execuções.

5. Discussão

Analisando o gráfico apresentado no tópico anterior, ficam evidentes as diferenças de tempo de execução da mesma aplicação em ambientes virtuais distintos. Também se torna visível que o desempenho da aplicação no *container* foi expressivamente melhor.

Supõe-se que o principal motivo do *container* prover um desempenho melhor se deve ao fato de que existe uma camada a menos na estrutura de virtualização do *Docker* [Bernstein 2015]. Ao contrário das VMs, o *container* trabalha com o sistema operacional

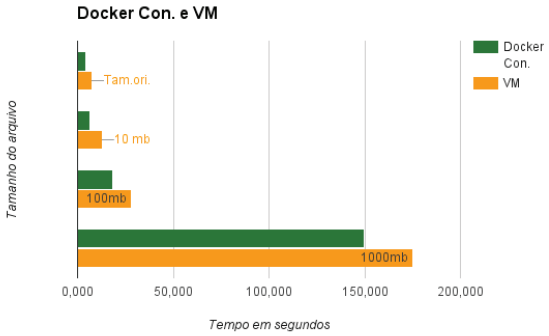

Figure 3. Tempo de execução vs tamanho da entrada.

hospedeiro e compartilha com o mesmo, partes do kernel e bibliotecas, o que reduz em muito o tempo necessário para a realização de chamadas de sistema, por exemplo. Enquanto as máquinas virtuais convencionais apresentam um sistema operacional hóspede que têm a perspectiva de estar atuando em conjunto com um hardware físico quando, na verdade, fazem interações com um hardware totalmente artificial. Ou seja, com as máquinas virtuais convencionais perde-se poder de processamento emulando hardwares virtuais para servir ao sistema operacional hóspede.

6. Conclusão

Através dos testes executados neste trabalho, é possível perceber a vantagem do Docker e seus containers frente às máquinas virtuais para a exeucação de aplicações de computação intensiva de dados. A versatilidade da tecnologia moderna otimiza, quando possível, a execução de tarefas, utilizando recursos do sistema operacional hospedeiro, resultando em um desempenho melhor e uma opção viável de virtualização.

References

Administration, N. A. S. (1958). *Ozone and air quality*. https://ozoneaq.gsfc.nasa.gov/.

Bernstein, D. (2015). *Containers and Cloud: From LXC to Docker to Kubernetes*. IEEE Cloud Computing, volume: 1, issue: 3, sept. 2014 edition.

Carissimi, A. (2008). *Virtualização: da teoria a soluções*. 26º simpósio brasileiro de redes de computadores e sistemas distribuídos. http://www.gta.ufrj.br/ensino/CPE758/artigos-basicos/cap4-v2.pdf.

Dean, J. and Ghemawat, S. (2008). *MapReduce: simplified data processing on large clusters*. Commun. ACM, 51(1):107–113.

Foundation, A. (2014). *What is the Apache Hadoop?* http://hadoop.apache.org/index.pdf.

Hub, D. I. (2015). *Docker Hub*. https://hub.docker.com/r/sergeygals/hbase-pseudo-distributed/.

IBM (2012). *For virtualization, it's all about control*. https://www.ibm.com/midmarket/us/en/article_Virtualization2_1209.html.

Inc, D. (2015). *Docker: The container engine*. https://github.com/docker/docker.

Tanenbaum, A. S. (2009). *Sistemas Operacionais Modernos*. São Paulo: Pearson Prentice Hall, 3th edition.