Big Data: Análise e a avaliação de ferramentas para gerenciamento de grande volume de dados

Cristiano Gehring, Eder Pazinato

Ciência da Computação – Universidade de Passo Fundo (UPF) – Campus I
BR 285, São José | Passo Fundo/RS

132757@upf.br, ederpazinatto@upf.br

Abstract. *This article presents the analysis of the Big Data concept and the evaluation of the tools that work with large volume data management. The work was developed with the creation of two clusters in distinct environments, being they Amazon Web Service and Google Cloud Platform, installed in the platform Hadoop with use of tool Hive. The practical tests were carried out from a meteorological database of the Centro de Previsão de Tempo e Estudos Climáticos / Instituto Nacional de Pesquisas Espaciais. At the end of the work, with the completion of the tests, the Amazon Web Service cluster performed better.*

Resumo. *Este artigo apresenta a análise do conceito Big Data e a avaliação das ferramentas que trabalham com gerenciamento de grande volume de dados. O trabalho foi desenvolvido com a criação de dois clusters em ambientes distintos, sendo eles Amazon Web Service e Google Cloud Platform, instalado na plataforma Hadoop com uso da ferramenta Hive. Os testes práticos foram realizados a partir de uma base de dados meteorológicos do Centro de Previsão de Tempo e Estudos Climáticos / Instituto Nacional de Pesquisas Espaciais. Ao final do trabalho, com a finalização dos testes o cluster da Amazon Web Service obteve melhor desempenho.*

1. Introdução

Desde a invenção da escrita, a sociedade sempre buscou maneiras de gravar as informações de forma que pudessem ser recuperadas posteriormente. Mas foi apenas com o surgimento do banco de dados que isso se tornou mais fácil e organizado. Entretanto, com o surgimento e a popularização das mídias sociais, a quantidade de informações que transitam pela rede cresceu de forma exponencial, principalmente em arquivos como fotos, mensagens, vídeos, músicas e texto, sendo estes, gravados de forma semi estruturada ou não estruturada.

Embora o empenho para organizar este número excessivo de dados seja considerável, o grande desafio imposto atualmente é a grande dificuldade de armazenar estes arquivos de forma padronizada, visto que mais de 70% dos dados que circulam na internet estão dispostos de forma não estruturada [SALES 2016] e a tendência é de que eles aumentem ao longo dos anos. Com as redes sociais crescendo gradativamente e o número de usuários aumentando, torna-se cada vez mais intenso o armazenamento de dados estruturados, semi estruturados e não estruturados. Contudo também há o paradigma de Internet das Coisas (*Internet of Things – IoT*) onde além dos computadores, outros eletrônicos fazem parte da infraestrutura global de internet, gerando, consumindo e interagindo com outras pessoas e objetos, no mundo físico e

virtual. Temos como exemplo desses objetos as etiquetas RFID, os sensores, os vestíveis e os smartphones [MARQUESONE 2016].

Há uma projeção feita pela Cisco que o número de objetos inseridos no contexto de IoT será em torno de 50 bilhões até o ano de 2020. Dada essa quantidade, um relatório da International Data Corporation (IDC) prevê que, em 2020, os dados gerados por máquinas representarão 42% de todos os dados existentes [MARQUESONE 2016]. Com o grande volume de dados surge a possibilidade de análise e gerenciamento, para gerar informações úteis à tomada de decisão. O conceito Big Data é utilizado para caracterizar os dados que excedem a capacidade de processamento de sistemas de banco de dados convencionais [SCHNEIDER 2012].

Segundo Schneider (2012), o primeiro ponto a ser esclarecido é que Big Data não tem uma única definição. Na verdade, é um termo que descreve pelo menos três distintas, mas interligadas, tendências:

- Captar e gerenciar grandes lotes de informação.
- Trabalhar com muitos tipos de dados.
- Explorar as massas de informações e novos tipos de dados com novos estilos de aplicativos.

Entender como tratar os desafios de Big Data é mais difícil do que entender o que significa o termo e quando empregá-lo. O desafio do processamento dos grandes volumes de dados está relacionado a aspectos como o armazenamento dos dados na memória principal, a grande quantidade de iterações sobre os dados e as frequentes falhas, sendo que o processamento intensivo e iterativo dos dados excede a capacidade individual de uma máquina convencional. Sendo que quanto mais dados as empresas tiverem mais complexa será a tarefa de geri-los. Uma das dificuldades de fazer consultas sobre esses dados é de que nem sempre trazem aquilo que se estava procurando.

Para melhor armazenamento e gerenciamento dos dados, surgiram ferramentas como o Hadoop e o Spark. Ferramentas que buscam ser capazes de armazenar arquivos de formas padronizadas, além de registrar o maior número de informações possíveis.

Este trabalho tem como objetivo instalar, avaliar e comparar características e diferenças de performance entre ferramentas para manipulação de grandes volumes de dados, exibindo essas informações de forma clara e sucinta.

2. Revisão Bibliográfica

Nesta seção serão apresentadas às ferramentas e conceitos que serão necessários para o desenvolvimento deste trabalho.

2.1. Hadoop

O Hadoop é uma plataforma *open source* escrita na linguagem de programação Java para armazenamento e processamento distribuído, podendo armazenar qualquer tipo de dado, a um custo baixo e em grande escala, ou podendo realizar análises de dados complexos rapidamente.

A estrutura do Hadoop é constituída por dois componentes principais o HDFS e o YARN. O HDFS é o Hadoop *Distributed File System* (Sistema de Arquivos Distribuídos do Hadoop), responsável por gerenciar os dados armazenados em disco no cluster. O YARN (*Yet Another Resource Negotiator*) atua como um gerenciador de

recursos do cluster, alocando recursos computacionais para aplicações que precisam realizar um processamento distribuído [BENGFORT 2016].

Diferente das plataformas tradicionais, o Hadoop é capaz de armazenar qualquer tipo de dado no seu formato nativo e realizar em escala, uma variedade de análises e transformações sobre esses dados. Com uma grande vantagem de baixo custo, já que é um software de código aberto que roda em hardware comum [IDG NEWS SERVICES 2016].

2.2. Spark

Por outro lado, o Spark é um framework que implementa o conceito dos RDDs (*Resilient Distributed Datasets*), permitindo o reuso de dados distribuídos em uma variedade de aplicações, provendo um mecanismo eficiente para recuperação de falhas em clusters, porém não faz armazenamento distribuído [IDG NEWS SERVICES 2016]. Spark combina SQL, streaming e análises complexas de forma integrada na mesma aplicação para lidar com uma grande variedade de situações de processamento de dados. Spark roda em cima de Hadoop, Apache Mesos, Kubernetes, standalone ou na nuvem. Pode acessar diversas fontes de dados como HDFS, Cassandra, HBase ou S3 [APACHE SPARK 2017].

2.3. Hive

O Apache Hive é um framework de "armazém de dados" (*data warehousing*) desenvolvido sobre o Hadoop. O Hive oferece uma interface baseada em SQL, que permite associar esquemas estruturados a dados no HDFS. O Hive oferece um dialeto próprio de SQL chamado HQL (Hive Query Language).

Os comandos do Hive e às consultas HQL são compilados em um plano de execução ou uma série de operações de HDFS e/ou jobs de MapReduce, que é então executado em um cluster Hadoop.

O lado positivo do Hive é oferecer a alta escalabilidade e por isso pode trabalhar com grande quantidade de dados, em escala de terabytes e petabytes [BENGFORT 2016].

3. Trabalhos Relacionados

Nesta seção serão apresentados alguns trabalhos relacionados à Análise de Big Data e de suas ferramentas de gerenciamento. Através de estudos com as ferramentas Hadoop e Spark.

No artigo Uso do Framework Hadoop e sua integração com a linguagem R [Donida, Hölbig 2015]. A ferramenta Hadoop é voltada para clusters e para a manipulação de grande quantidade de dados. Atualmente é muito utilizada por grandes empresas, como o Facebook e o Yahoo, por exemplo. O artigo visa, também, mostrar sua arquitetura e como é realizado o trabalho deste framework, o deixando interessante no campo da Big Data sendo utilizado por inúmeras empresas. Neste trabalho foi feita a integração do framework com a linguagem R por meio de alguns exemplos práticos, que pelo uso de pacotes e serviços da Linguagem R possibilitam a integração com o Hadoop. Com alguns estudos de casos esta integração com o R é explicitada e analisada.

Segundo Zaharia et al. (2010) no artigo Cluster Computing with Working Sets é apresentada a ferramenta Spark e também é realizada uma comparação entre Spark e Hadoop. Para esta comparação foi utilizado o algoritmo de regressão logística e de alternar os quadrados mínimos. Foi utilizado o tempo de execução como parâmetro de comparação. O resultado obtido mostra uma diferença de tempo de execução de até 10 vezes para o regressão logística e de aproximadamente 2.8 vezes para o algoritmo de alternar os quadrados mínimos.

4. Aplicação: Estudo de Caso

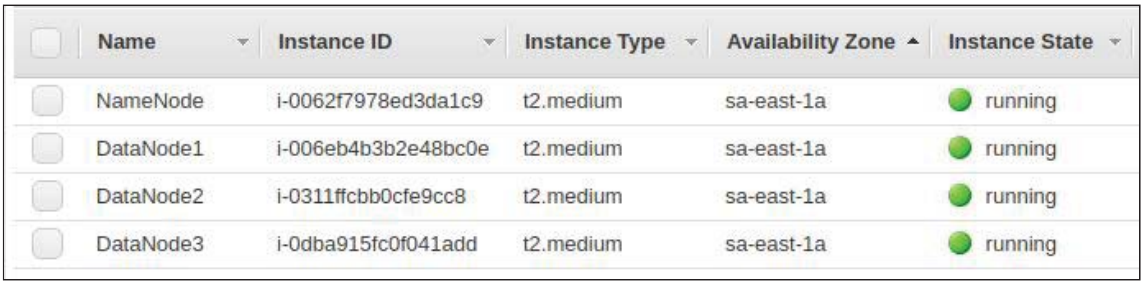
Para este trabalho foram preparados dois clusters com 4 máquinas para os testes, sendo estes: Amazon Web Services (AWS) e Google Cloud Platform, após a preparação dos mesmos, foram realizados os testes para verificar qual o ambiente obteria o melhor desempenho.

4.1. Configuração do Ambiente

Para este trabalho foram preparados dois ambientes para teste: Amazon Web Services (AWS) e Google Cloud Platform. Estes ambientes foram selecionado devido aos mesmos serem muito utilizados para computação em nuvem. Além disso, o primeiro foi escolhido por ser flexível e por permitir a criação de um cluster totalmente customizado. Por outro lado, o segundo foi escolhido por possibilitar maior facilidade para trabalhar e também por já ter um ambiente pronto.

4.1.1. Amazon Web Services (AWS)

Para a realização do trabalho foi criado um ambiente distribuído com 4 instâncias, conforme apresenta a figura 1, na plataforma ec2 da AWS (Amazon Web Services), seguindo o tutorial “Install Hadoop on Amazon AWS EC2 Instances” [DYCK 2018]. As instâncias criadas possuem às respectivas configurações 2 vCPUs, 2,5 GHz, família Intel Xeon, memória de 4 GiB (GibiByte). Nas máquinas criadas, foi instalado o sistema operacional Ubuntu Server 16.04.



Name	Instance ID	Instance Type	Availability Zone	Instance State
NameNode	i-0062f7978ed3da1c9	t2.medium	sa-east-1a	running
DataNode1	i-006eb4b3b2e48bc0e	t2.medium	sa-east-1a	running
DataNode2	i-0311ffcbb0cfe9cc8	t2.medium	sa-east-1a	running
DataNode3	i-0dba915fc0f041add	t2.medium	sa-east-1a	running

Figura 1. Instâncias na AWS

Para o funcionamento das trocas de informações entre as instâncias e entre o computador pessoal foram necessárias a geração dos pares de chaves públicas e privadas. Este processo foi executado na própria plataforma da Amazon. Com as chave configuradas é possível ter acesso às máquinas através do protocolo Secure Shell.

Após ter acesso aos servidores é necessário fazer a instalação do JDK (Java *Development Kit*), que pode ser obtido através do comando “*sudo apt-get install default-jdk*”.

Para facilitar a instalação da ferramenta foi criado o script "*bigdata.sh*" dentro do diretório "*/etc/profile.d/*". Este é um diretório que contém arquivos de script de shell que são executados durante os recursos de inicialização e instalação geral para todos os usuários. Este script cria algumas variáveis de ambiente úteis para instalação, configuração e execução do ecossistema Hadoop, como é visto na figura 2.

bigdata.sh ✖ hosts ✖

```
#!/bin/bash
# Variáveis de ambiente para ferramentas de Big Data

# Variáveis AmazonEC2 - Início
export NameNodeDNS="ec2-54-233-199-157.sa-east-1.compute.amazonaws.com"
export DataNode001DNS="ec2-18-231-171-25.sa-east-1.compute.amazonaws.com"
export DataNode002DNS="ec2-18-231-115-125.sa-east-1.compute.amazonaws.com"
export DataNode003DNS="ec2-18-231-176-227.sa-east-1.compute.amazonaws.com"

export NameNodeIP="172.31.1.168"
export DataNode001IP="172.31.11.40"
export DataNode002IP="172.31.3.77"
export DataNode003IP="172.31.6.122"
export IdentityFile="~/.ssh/hadoop-clusterkeypair.pem"
# Variáveis AmazonEC2 - Fim

# Variáveis JAVA - Início
export JAVA_HOME=/usr/lib/jvm/default-java
PATH=$PATH:$JAVA_HOME/bin
# Variáveis JAVA - Fim

# Variáveis HADOOP - Início
export HADOOP_HOME='/usr/local/hadoop'
export HADOOP_CONF_DIR="${HADOOP_HOME}/etc/hadoop"
export HADOOP_DATA_HOME="${HOME}/hadoop_data/hdfs"
PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
# Variáveis HADOOP - Fim
```

Figura 2. Script para a configuração das variáveis de ambiente

Para o funcionamento da comunicação entre os 4 nodos do cluster é necessário configurar o arquivo "*/etc/hosts*", que é onde fica indexado o ip de cada máquina com seu respectivo nome. Essa configuração pode ser vista na figura 3.

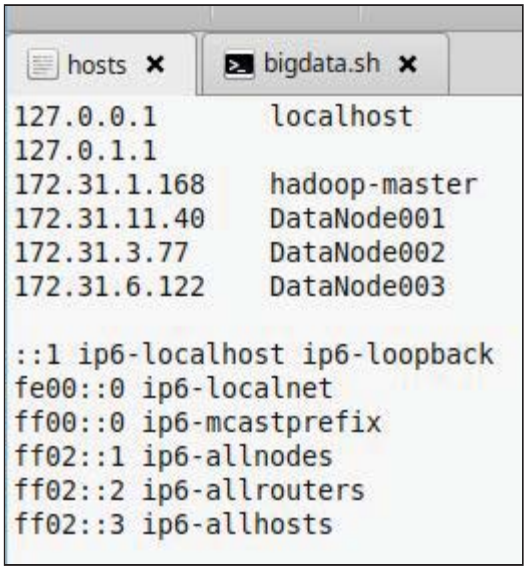


Figura 3. Indexação dos IPs dos nodos

Após a instalação dos programas necessários, deve-se baixar o Hadoop, descompactá-lo e mover o seu conteúdo para o diretório onde o mesmo será mantido. Os comandos para fazer este processo estão na figura 4.

```
wget http://ftp.unicamp.br/pub/apache/hadoop/common/hadoop-2.9.0/hadoop-2.9.0.tar.gz
sudo tar -zxvf ~/Downloads/Hadoop/hadoop-*.tar.gz -C /usr/local
sudo mv /usr/local/hadoop-* /usr/local/hadoop
```

Figura 4. Comandos para download e instalação do Hadoop

O penúltimo passo para configurar o Hadoop é editar os arquivos de configuração para o ambiente, o site do MapReduce, HDFS e do YARN. Os arquivos são apresentados na figura 5.

```
/usr/local/hadoop/etc/hadoop/core-site.xml
/usr/local/hadoop/etc/hadoop/mapred-site.xml
/usr/local/hadoop/etc/hadoop/yarn-site.xml
/usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

Figura 5. Arquivos de Configuração do Hadoop

Depois de toda a configuração, usa-se o comando "*hadoop namenode -format*" para formatar o sistema de arquivo do Hadoop, para depois iniciá-lo com o seguinte comando:

```
cd /usr/local/hadoop/sbin
start-all.sh
```

Figura 6. Comandos para inicialização do Hadoop

4.1.2. Google Cloud Platform

O Google Cloud Platform é uma plataforma de computação em nuvem oferecida pelo Google, que oferece uma série de serviços incluindo, computação, armazenamento de dados, análise de dados e aprendizagem de máquina.

Inicialmente foi criada uma conta na plataforma google cloud, após acessou-se o console onde existem todas as ferramentas da plataforma. No menu de navegação do painel Big Data foi selecionada a opção o “Dataproc” e dentro dela “Clusters”. Com isso, podemos visualizar um painel de gerenciamento de clusters. Dentro deste painel existe a opção de criar clusters a qual foi acessada, após abre-se um painel para criação e configuração de um cluster. Neste painel é possível escolher a região onde serão criadas às máquinas virtuais, sendo possível escolher a quantidade de máquinas, de memória RAM e de armazenamento. Para a criação do cluster foi selecionada a opção com 4 máquinas, sendo 1 máquina principal e 3 *workers*, como podemos conferir na figura 7 a seguir:

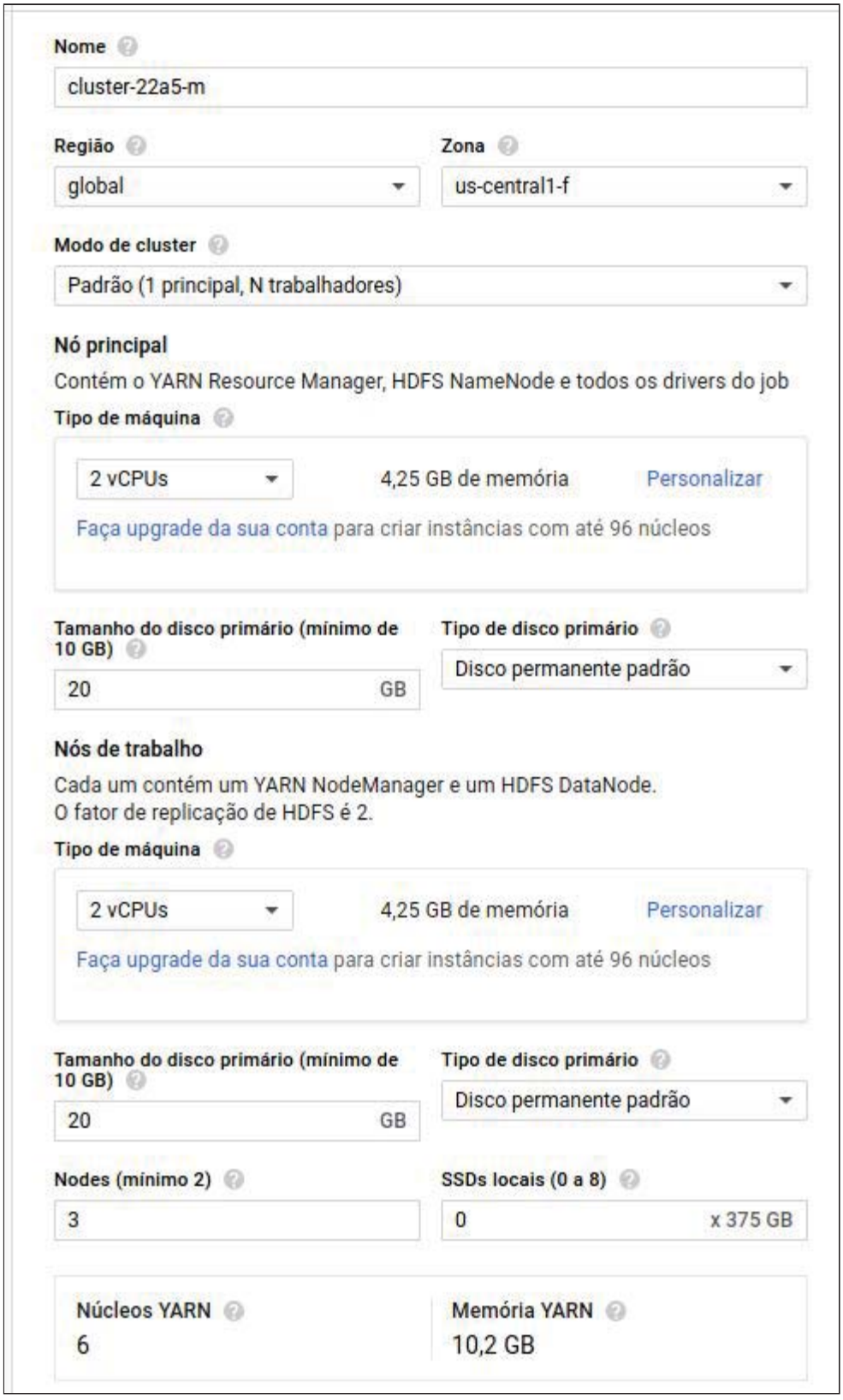


Figura 7. Interface de configuração do Cluster na Google Cloud Platform

As demais opções foram mantidas no formato padrão do painel para a criação do cluster e em seguida foi selecionada a opção criar. Assim o cluster é gerado de forma

automática pela plataforma. Após a criação do cluster é acessado no menu de navegação do painel a opção “Compute Engine” e dentro dele a opção “Instâncias de VMs” onde são listadas as máquinas virtuais criadas como podemos observar na figura 8:

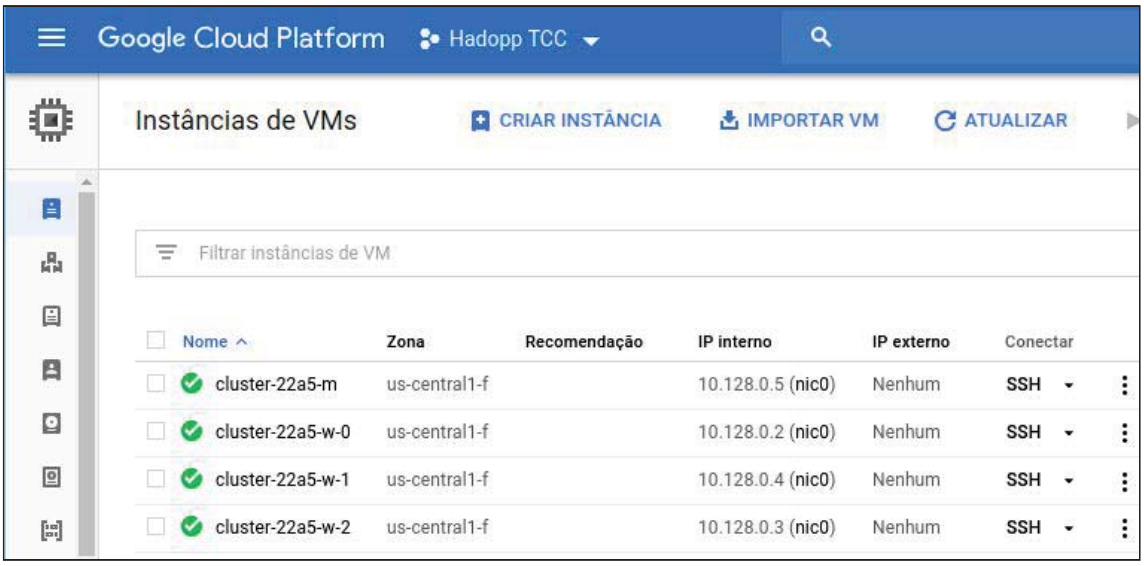


Figura 8. Painel onde são listadas as máquinas virtuais

Posteriormente para acessar a máquina virtual que foi criada vamos em conectar via SSH e selecionamos a opção abrir na janela do navegador para poder ter acesso a linha de comando da máquina, como podemos ver na figura 9 a seguir:

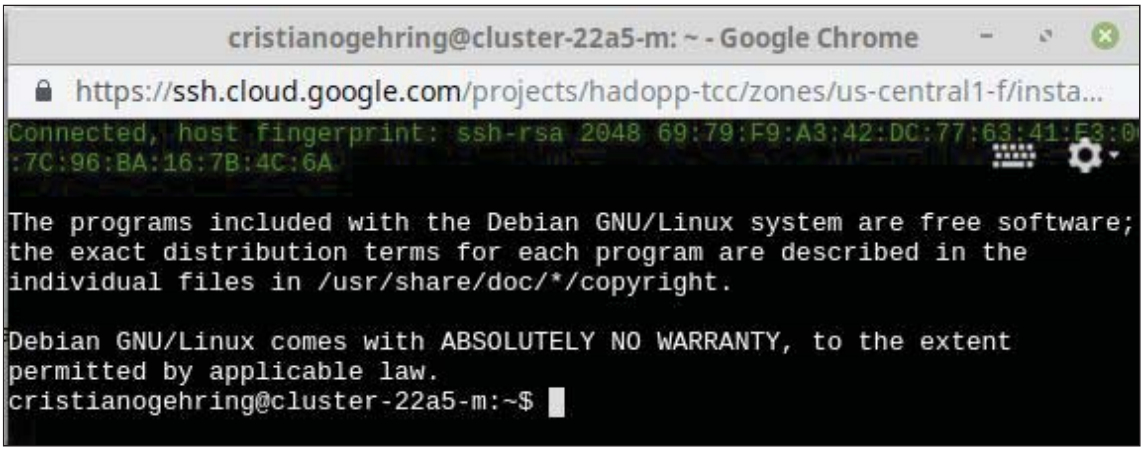


Figura 9. Ambiente para acesso SSH

É interessante ressaltar que nesta plataforma não foi necessária a instalação das ferramentas Hadoop e Hive, sendo que a própria plataforma já possui as mesmas instaladas e configuradas, facilitando o desenvolvimento do trabalho.

4.2. Testes

Os dados que foram utilizados nos testes realizados neste trabalho, são dados meteorológicos obtidos através do CPTEC/INPE (Centro de Previsão de Tempo e Estudos Climáticos / Instituto Nacional de Pesquisas Espaciais), e estão disponíveis para acesso via FTP (*file transfer protocol*) [CPTEC 2018].

Os dados estão armazenados em diretórios separados conforme sistema de data sendo esse respectivamente ano, mês e dia. Quando um desses diretórios é acessado pode-se verificar arquivos que estão em formato txt, cujo nome contém a descrição Eta15km (modelo atmosférico utilizado para fins de pesquisa cuja resolução horizontal é de 15km com 22 níveis verticais) além da data, abreviatura do nome do país e do estado seguido pelo nome da cidade. Além disso, cada diretório possui aproximadamente 559 arquivos, sendo que cada arquivo tem 18.9 kB com 169 linhas. Conforme percebido na figura 10:

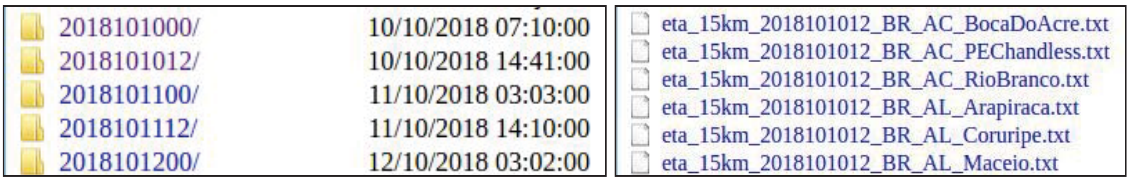


Figura 10. Diretórios e arquivos da base de dados

Após acessar o arquivo em txt podemos observar sua estrutura, que é composta por colunas identificadas por hora, tp2m (temperatura do ar a 2 metros), umrl (umidade relativa), lwnv (nebulosidade - fração de cobertura de nuvens baixas), mdnv (nebulosidade - fração de cobertura de nuvens médias), hinv (nebulosidade - fração de cobertura de nuvens altas), prec (precipitação total), ocis (radiação de onda curta incidente à superfície), vento (velocidade do vento), dirv (direção do vento), pslc (pressão à superfície) e tgsc (temperatura do solo na camada de 0-10 cm). Sendo esses dados capturados de hora em hora, conforme figura 11.

hora	tp2m	umrl	lwnv	mdnv	hinv	prec	ocis	vento	dirv	pslc	tgsc
1	27.78	81.44	0.00	0.00	0.00	0.00	0.00	3.85	88.50	991.70	-272.92
2	26.71	70.35	0.00	414.32	0.00	0.00	0.00	3.67	84.40	991.66	-272.92
3	26.42	64.18	0.00	399.55	0.00	0.00	0.00	3.38	67.67	991.52	-272.92
4	26.03	62.73	0.00	394.16	0.00	0.00	0.00	2.91	74.18	991.88	-272.92
5	25.53	62.89	0.00	392.95	0.00	0.00	0.00	2.92	95.60	991.50	-272.92
6	24.98	64.05	0.00	392.10	0.00	0.00	0.00	3.38	111.44	991.53	-272.92
7	24.27	62.67	0.00	391.17	0.00	0.02	0.00	3.74	113.82	991.00	-272.92
8	24.06	61.95	0.00	394.18	0.00	0.00	0.00	4.28	121.54	991.54	-272.92
9	23.88	61.04	0.00	391.87	0.00	0.00	0.00	4.67	123.78	991.89	-272.92
10	23.70	57.96	0.00	388.21	0.00	0.00	0.00	4.69	125.58	991.83	-272.92

Figura 11. Arquivo com a estrutura dos dados

Esses dados obtidos foram baixados para posteriormente trabalhar com os servidores da Amazon Web Service e Google Cloud Platform. Para a elaboração do trabalho foi utilizada a ferramenta Hive, para o uso dela foi necessário criar uma tabela com a estrutura necessária para trabalhar com os dados que seriam importados. A estrutura da tabela pode ser vista na figura 12.

```
CREATE TABLE bigdata (
  hora int, tp2m float, umrl float, lwnv float, mdnv float, hinv float,
  prec float, ocis float, vento float, dirv float, pslc float, tgsc float
)
PARTITIONED BY (data date, cidade string, uf varchar(2))
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH
  SERDEPROPERTIES ('input.regex'='^\\s*(\\d+)\\s*(-?\\d*.\\d+)\\s*(-?\\d*.\\d+)' ||
  '\\s*(-?\\d*.\\d+)\\s*(-?\\d*.\\d+)\\s*(-?\\d*.\\d+)\\s*(-?\\d*.\\d+)\\s*(-?\\d*.' ||
  '\\d+)\\s*(-?\\d*.\\d+)\\s*(-?\\d*.\\d+)\\s*(-?\\d*.\\d+)\\s*(-?\\d*.\\d+)')
  TBLPROPERTIES ("skip.header.line.count"="1");
```

Figura 12. Estrutura da tabela para onde serão importados os dados

Depois de gerar a estrutura da tabela foi necessário criar um script escrito em Python que faz a leitura dos dados e insere na mesma. Como pode-se observar na figura 13.

```
import os
from pyhive import hive
conn = hive.Connection(host="localhost", port=10000, username="cristianogehring")
cursor = conn.cursor()

for dirpath, dirnames, arquivos in os.walk('/home/cristianogehring/arquivos/'):
  print(dirpath)
  for arquivo in arquivos:
    data = arquivo[9:13]+'-'+arquivo[13:15]+'-'+arquivo[15:17]
    uf = arquivo[23:25]
    cidade = arquivo[26:-4]
    cursor.execute("LOAD DATA LOCAL INPATH '"+dirpath+"/"+arquivo+"' INTO TABLE
    bigdata PARTITION(data='"+data+"', cidade='"+cidade+"', uf='"+uf+"')")
```

Figura 13. Script de importação de dados para o Hive

4.2.1 Caso de Teste 1

O caso de teste 1 tem por objetivo apresentar às médias de precipitações registradas por estado no mês de setembro de 2018. Para isso foi elaborado o SQL conforme figura 14.

```
SELECT ROUND(AVG(prec), 2) AS media_prec, UPPER(uf) AS uf FROM bigdata
WHERE MONTH(data) = 9 AND YEAR(data) = 2018 AND hora < 25 GROUP BY UPPER(uf) ORDER BY media_prec
```

Figura 14. Comando SQL Caso de Teste 1

O SQL foi executado nas ferramentas, onde na AWS o tempo de execução obtido foi de 392.145 segundos e na Google Cloud o tempo de execução obtido foi de 510.353 segundos. Como resultado do SQL executado em ambas às ferramentas, foram obtidas as médias de precipitações registradas por estado no mês de setembro de 2018, conforme figura 15 a seguir:

media_prec	uf
0.0	CE
0.01	MA
0.01	RN
0.01	PI
0.01	TO
0.02	PE
0.02	PB
0.04	BA
0.06	AL
0.06	DF
0.06	SE
0.06	AP
0.07	PA
0.11	SP
0.12	GO
0.12	RJ
0.13	PR
0.14	MG
0.17	ES
0.18	PY
0.18	AC
0.18	MT
0.2	RO
0.21	MS
0.23	RS
0.23	SC
0.27	AM

27 rows selected

Figura 15. Médias de precipitações

Com o tempo médio de execução entre às ferramentas obtido, pode-se fazer um comparativo entre ambas, o qual observou-se que a ferramenta AWS foi 30,14% mais rápida na execução do SQL que a ferramenta Google Cloud.

4.2.2 Caso de Teste 2

O caso de teste 2 tem por objetivo apresentar, a maior temperatura registrada por estado no mês de setembro de 2018, essa análise também exibe a data, hora e a cidade em que ocorreu o fenômeno. Para isso foi elaborado o comando SQL conforme figura 16.

```
WITH with_maior_tp2m AS (
  SELECT max(tp2m) AS maior_tp2m, UPPER(uf) AS uf FROM bigdata
   WHERE MONTH(data) = 9 AND YEAR(data) = 2018 AND hora < 25
   GROUP BY UPPER(uf) ORDER BY maior_tp2m)
SELECT a.maior_tp2m, DATE_FORMAT(b.data, 'dd/MM/Y') AS data, b.hora, b.cidade, b.uf
  FROM with_maior_tp2m a
  JOIN bigdata b ON b.tp2m = a.maior_tp2m AND a.uf = b.uf AND b.hora < 25
 ORDER BY a.maior_tp2m desc;
```

Figura 16. Comando SQL Caso de Teste 2

O SQL foi executado nas ferramentas, onde na AWS o tempo de execução obtido foi de 682.101 segundos e na Google Cloud o tempo de execução obtido foi de 734.024 segundos. Como resultado do SQL executado em ambas às ferramentas, foi obtida a maior temperatura registrada por estado, conforme figura 17:

maior_tp2m	data	hora	cidade	uf
38.97	13/09/2018	21	Nhumirim	MS
38.97	13/09/2018	20	Nhumirim	MS
38.37	23/09/2018	21	SaoFelixDoAraguaia	MT
37.9	23/09/2018	21	Chortitzer_Las_Palmas	PY
37.81	23/09/2018	21	Parana	TO
37.16	18/09/2018	21	ConceicaoDoAraguaia	PA
36.78	12/09/2018	21	Cacoal	RO
36.28	08/09/2018	22	Ariquemes	GO
35.94	29/09/2018	21	Caxias	MA
35.89	24/09/2018	21	Parintins	AM
35.85	29/09/2018	21	Teresina	PI
35.26	10/09/2018	22	BocaDoAcre	AC
35.12	25/09/2018	20	Jales	SP
34.91	24/09/2018	20	DiamanteDoNorte	PR
34.84	24/09/2018	20	CentroUrbanoMg	MG
34.84	24/09/2018	19	CentroUrbanoMg	MG
34.05	24/09/2018	20	PortoGrande	AP
33.83	29/09/2018	20	Apodi	RN
33.74	23/09/2018	17	SaoBorja	RS
33.71	29/09/2018	20	MoradaNova	CE
32.51	29/09/2018	20	Floresta	PE
32.35	29/09/2018	20	SaoGoncalo	PB
32.13	17/09/2018	17	PaoDeAcucar	AL
32.13	17/09/2018	18	PaoDeAcucar	AL
32.0	28/09/2018	20	NovaVenecia	ES
31.76	27/09/2018	18	Campos	RJ
31.76	27/09/2018	19	Campos	RJ
31.3	24/09/2018	18	SaoMiguelDoOeste	SC
30.45	17/09/2018	18	Itabaianinha	SE
29.91	29/09/2018	20	AguasEmendadas	DF

30 rows selected

Figura 17. A maior temperatura registrada por estado

Neste teste com o tempo médio de execução entre às ferramentas obtido, pode-se fazer um comparativo entre ambas, o qual observou-se que a ferramenta AWS foi 7,61% mais rápida na execução do SQL que a ferramenta Google Cloud. Além do percentual também foi possível notar que a cidade com maior temperatura registrada no mês de Setembro de 2018 foi Nhumirim localizada no estado de Mato Grosso do Sul. Essa temperatura ficou registrada na data de 13 de setembro de 2018 às 20 e 21 horas, sendo de 38.97 graus celsius.

4.2.3 Caso de Teste 3

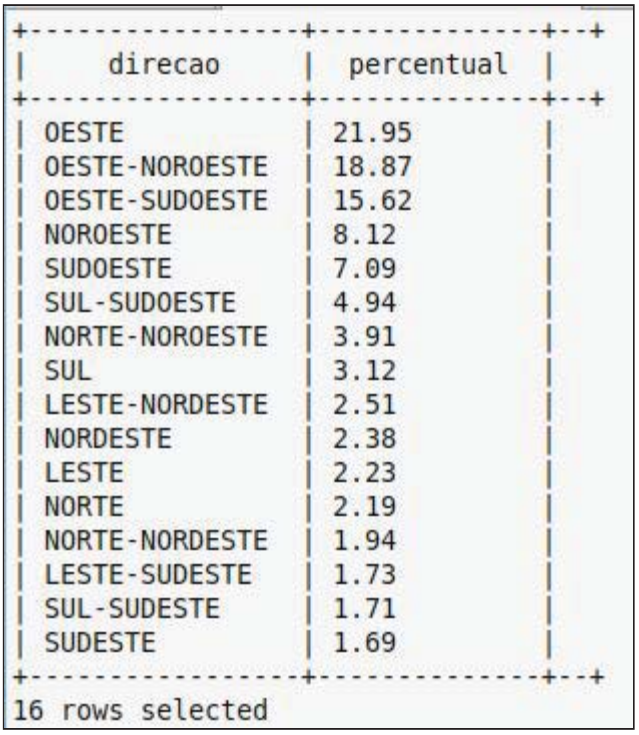
O caso de teste 3 tem o objetivo de apresentar, o percentual das coordenadas geográficas dos ventos. Para isso foi elaborado o seguinte SQL conforme figura 18.

```
WITH with_count_linhas AS (select count(*) AS qtd_linhas from bigdata),
with_direcao_vento AS (SELECT (CASE
     WHEN dirv < 11.25 OR dirv >= 348.75 THEN 'NORTE'
     WHEN dirv < 33.75 THEN 'NORTE-NORDESTE'
     WHEN dirv < 56.25 THEN 'NORDESTE'
     WHEN dirv < 78.75 THEN 'LESTE-NORDESTE'
     WHEN dirv < 101.25 THEN 'LESTE'
     WHEN dirv < 123.75 THEN 'LESTE-SUDESTE'
     WHEN dirv < 146.25 THEN 'SUDESTE'
     WHEN dirv < 168.75 THEN 'SUL-SUDESTE'
     WHEN dirv < 191.25 THEN 'SUL'
     WHEN dirv < 213.75 THEN 'SUL-SUDOESTE'
     WHEN dirv < 236.25 THEN 'SUDOESTE'
     WHEN dirv < 258.75 THEN 'OESTE-SUDOESTE'
     WHEN dirv < 281.25 THEN 'OESTE'
     WHEN dirv < 303.75 THEN 'OESTE-NOROESTE'
     WHEN dirv < 326.25 THEN 'NOROESTE'
     WHEN dirv < 348.75 THEN 'NORTE-NOROESTE' END) AS direcao
     FROM bigdata)
SELECT direcao, ROUND((COUNT(direcao) * 100) / qtd_linhas, 2) AS percentual
  FROM with_direcao_vento
  LEFT JOIN with_count_linhas ON TRUE
 WHERE hora < 25
 GROUP BY direcao, qtd_linhas
 ORDER BY percentual desc;
```

Figura 18. Comando SQL Caso de Teste 3

É importante ressaltar que às coordenadas geográficas dos ventos se originam da direção da rosa-dos-ventos com a direção em graus [Martins 2014].

O SQL foi executado nas ferramentas, onde na AWS o tempo de execução obtido foi de 664.507 segundos e na Google Cloud o tempo de execução obtido foi de 732.195 segundos. Como resultado do SQL executado em ambas às ferramentas, foi obtido o percentual das coordenadas geográficas dos ventos, conforme figura 19:



direcao	percentual
OESTE	21.95
OESTE-NOROESTE	18.87
OESTE-SUDOESTE	15.62
NOROESTE	8.12
SUDOESTE	7.09
SUL-SUDOESTE	4.94
NORTE-NOROESTE	3.91
SUL	3.12
LESTE-NORDESTE	2.51
NORDESTE	2.38
LESTE	2.23
NORTE	2.19
NORTE-NORDESTE	1.94
LESTE-SUDESTE	1.73
SUL-SUDESTE	1.71
SUDESTE	1.69

16 rows selected

Figura 19. Percentual das coordenadas geográficas dos ventos

Neste teste com o tempo médio de execução entre às ferramentas, pode-se fazer um comparativo entre ambas, o qual observou-se que a ferramenta AWS foi 10,18% mais rápida na execução do SQL que a ferramenta Google Cloud.

4.2.4 Caso de Teste 4

O caso do teste 4 tem por objetivo identificar se existe uma relação entre a variação de temperatura com a precipitação. Neste caso, foram analisados os dados da cidade de Passo Fundo, Rio Grande do Sul, no mês de Setembro de 2018, duas horas após o início da precipitação. Para isso foi elaborado o seguinte SQL conforme figura 20.

```
WITH
  with_data_hora_precipitacao AS (
    SELECT data, min(hora) AS hora
      FROM bigdata
     WHERE prec > 0
       AND hora < 25
       AND data BETWEEN '2018-09-01' AND '2018-09-30'
       AND cidade = 'PassoFundo'
     GROUP BY data),
  with_precipitacao AS (
    SELECT w2.*, b1.tp2m, b1.prec
      FROM with_data_hora_precipitacao w2
      JOIN bigdata b1 ON b1.hora = w2.hora
       AND b1.data = w2.data
       AND b1.prec > 0
       AND b1.hora < 25
       AND b1.data BETWEEN '2018-09-01' AND '2018-09-30'
       AND b1.cidade = 'PassoFundo')
SELECT p.data,
       p.hora AS hora_inicial,
       p.tp2m AS temp_inicial,
       p.prec AS prec_hora_inicial,
       b.hora AS hora_final,
       b.tp2m AS temp_final,
       b.prec AS prec_hora_final,
       round((b.tp2m - p.tp2m), 2) AS dif_temp
  FROM with_precipitacao p
  JOIN bigdata b ON from_unixtime(unix_timestamp(concat(b.data,' ',b.hora),
                                                 'MM-dd-yyyy HH'))
                  = from_unixtime(unix_timestamp(concat(p.data,' ',p.hora),
                                                 'MM-dd-yyyy HH')+(2*3600))
WHERE b.hora < 25
  AND b.data BETWEEN '2018-09-01' AND '2018-09-30'
  AND b.cidade = 'PassoFundo';
```

Figura 20. Comando SQL Caso de Teste 4

O SQL foi executado nas ferramentas, onde na AWS o tempo de execução obtido foi de 58.098 segundos e na Google Cloud o tempo de execução obtido foi de 69.282 segundos. Como resultado do SQL executado em ambas às ferramentas, foram obtidos os dados da cidade de Passo Fundo e a relação entre a variação de temperatura com a precipitação, conforme figura 21 abaixo:

p.data	hora_inicial	temp_inicial	prec_hora_inicial	hora_final	temp_final	prec_hora_final	dif_temp
2018-09-01	1	17.08	1.78	3	17.04	2.03	-0.04
2018-09-02	2	15.3	1.05	4	14.27	0.0	-1.03
2018-09-03	15	13.87	2.54	17	14.43	2.14	0.56
2018-09-14	2	14.43	2.39	4	14.87	0.2	0.44
2018-09-15	2	13.59	1.59	4	13.84	1.21	0.25
2018-09-16	2	13.76	0.02	4	13.36	0.0	-0.4
2018-09-17	13	13.71	0.23	15	13.74	0.84	0.03
2018-09-20	5	16.46	1.13	7	16.55	0.73	0.09
2018-09-23	19	29.31	0.03	21	27.3	1.3	-2.01
2018-09-24	15	25.58	1.0	17	24.7	1.87	-0.88
2018-09-25	19	27.0	0.88	21	26.52	6.36	-0.48
2018-09-26	7	16.23	1.36	9	16.55	0.71	0.32
2018-09-27	2	18.53	1.09	4	18.08	0.71	-0.45
2018-09-30	11	19.24	0.01	13	20.1	1.76	0.86

14 rows selected

Figura 21. Dados da Cidade de Passo Fundo e a relação entre a variação de temperatura com a precipitação

Com base neste teste pode-se observar que a relação entre a variação de temperatura com a precipitação existiu no mês de Setembro. Neste caso, pode-se identificar que existe uma relação entre a variação de temperatura com a precipitação. Porém, não pode-se afirmar que a temperatura será maior ou menor do que a temperatura inicial registrada, como pode-se observar na figura 21 acima.

Também com o tempo médio de execução entre às ferramentas obtido, pode-se fazer um comparativo entre ambas, o qual observou-se que a ferramenta AWS foi 19,25% mais rápida na execução do SQL que a ferramenta Google Cloud.

4.3 Casos de Teste: comparação

O gráfico a seguir faz uma comparação entre o tempo de execução dos testes elaborados com as ferramentas Amazon Web Service e Google Cloud Platform. Bem como, a ferramenta Amazon Web Service quando comparada com a Google Cloud Platform obteve vantagem em tempo de execução, como conseguimos observar na figura 22.

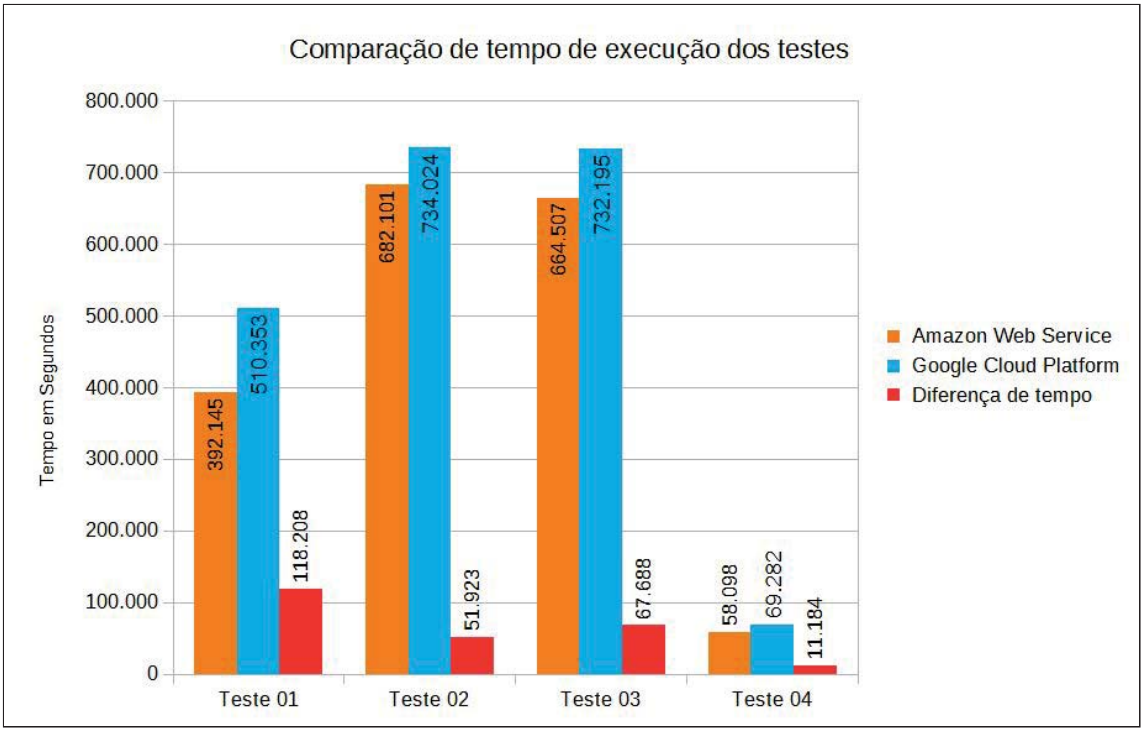


Figura 22. Gráfico de comparação de tempo de execução dos testes

5. Conclusão

Através da elaboração desta pesquisa pode-se analisar e avaliar às ferramentas de gerenciamento de dados. Bem como o Big Data vem sendo cada vez mais importante para a tomada de decisões em diversos setores como negócios, marketing, financeiro, e-commerce, entre outros. Também hoje em dia existe uma gama de dispositivos que coletam grandes volumes de dados, como por exemplo, o CPTEC/INPE que foi utilizado neste trabalho, que faz a coleta de dados meteorológicos de hora em hora.

Atualmente as ferramentas para trabalhar com o Big Data estão em constante evolução. Visto que através delas é possível que a manipulação dos dados possa ser feita, com menor tempo e de maneira mais efetiva. Além dessas, terem um tempo de

resposta considerado bom, do mesmo modo, elas podem manipular uma grande quantidade de dados e com isso gerar valor.

Nos testes realizados neste trabalho com as plataformas concluiu-se que, mesmo com os clusters possuindo configurações similares, o Amazon Web Service foi superior em desempenho ao da Google Cloud Platform. Visto que, o cluster da Amazon Web Service conseguiu realizar todos os testes deste trabalho com um menor tempo de execução.

Durante o estudo e instalação das plataformas foi possível perceber que a Amazon Web Service é uma plataforma mais flexível que possibilita instalar somente o necessário oferecendo um desempenho sólido nas execuções. Enquanto, a Google Cloud Platform abrange uma gama maior de ferramentas já instaladas, que auxiliam em uma rápida instalação, mas que no entanto, comprometem o desempenho das execuções.

5.1 Trabalhos Futuros

Como complemento deste trabalho, seria importante o usos de dados não estruturados. Nos testes realizados não foi possível usar dados com essas características, devido a grande dificuldade de encontrar uma base neste formato.

Outra atividade que pode ser realizada é a comparação entre as ferramentas Spark e Hadoop. Para tanto será necessário o uso e análise de ambientes com a ferramenta Spark.

6. Referência Bibliográfica

SCHNEIDER, R. D. Hadoop For Dummies, Special Edition. Mississauga, CAN: John Wiley & Sons Canada, 2012. 41 p.

SALES, Marcelo. Big data: seus dados perderam a estrutura, e agora?. Disponível em: <https://corporate.canaltech.com.br/noticia/negocios/Seus-dados-perderam-a-estrutura-e-agora/>. Acesso em: 30 out. 2016.

IDG NEWS SERVICES. Cinco coisas que você precisa saber sobre Hadoop e Apache Spark. Disponível em: <http://computerworld.com.br/cinco-coisas-que-voce-precisa-saber-sobre-hadoop-e-apache-spark>. Acesso em: 01 nov. 2016.

MARQUESONE, Rosangela. Big Data: Técnicas e tecnologias para extração de valor dos dados. São Paulo: Casa do Código. 2016.

APACHE SPARK. Spark. Disponível em <https://spark.apache.org/>. Acesso em: 08 set. 2017.

DYCK, Chris. Install Hadoop on Amazon AWS EC2 Instances. Disponível em: <http://klasserom.azurewebsites.net/Lessons/Binder/1960#CourseStrand_3422>. Acesso em: 03 set. 2018.

BENGFORT, Benjamin; KIM, Jenny. Analítica de dados com Hadoop: Uma Introdução para cientistas de dados. 1. ed. São Paulo: Novatec, 2016. 352 p.

DONIDA, Eliunes; HÖLBIG, Carlos Amaral. Uso do framework Hadoop e sua integração com a linguagem R. 2015. 16 p. Artigo Científico (Graduando Ciências da Computação)- Universidade de Passo Fundo, Passo Fundo, 2015.

ZAHARIA, Matei et al. Spark: Cluster Computing with Working Sets. 2010. 7 p. Artigo Cientifico (Graduate Computer Science)- University of California, Berkeley, 2010. Disponível em: <https://www.usenix.org/legacy/event/hotcloud10/tech/full_papers/Zaharia.pdf>. Acesso em: 15 nov. 2018.

CPTEC. Centro de Previsão de Tempo e Estudos Climáticos. Disponível em <ftp://ftp1.cptec.inpe.br/etamdl/Products/Eta_tempo/Eta15km/sisalert/>. Acesso em: 30 set. 2018.

MARTINS, Samantha. METEORÓPOLE: Uma meteorologista além da Meteorologia. 2014. Disponível em: <http://meteoropole.com.br/2014/01/vento-conversao-da-direcao/>. Acesso em: 04 nov. 2018.