

Atividade em Laboratório Virtual 01

Disciplina	HDM – Desenvolvimento de Soluções com MapReduce utilizando Hadoop
-------------------	--

Objetivos

Exercitar os seguintes conceitos vistos em sala de aula:

- ✓ Compilar um programa Hadoop, gerando um arquivo jar para ser submetido a um job Hadoop.
- ✓ Executar o programa que foi compilado.
- ✓ Consultar os resultados.

Ao final desta atividade o aluno deverá ser capaz de compilar e executar um programa Hadoop.

Enunciado

Atenção: *primeiramente iremos formatar o HDFS e iniciar os serviços do Hadoop. Essas etapas podem parecer um pouco repetitivas para aqueles alunos que já são experientes com a ferramenta Hadoop. Essas atividades estão descritas passo a passo nesse documento. Se o aluno sentir que é capaz de realizar essas etapas sem o acompanhamento do tutorial, ele deverá saltar essas etapas de formatar o HDFS e iniciar os serviços do Hadoop por conta própria. Aqueles que não se lembram do procedimento, para a realização dessas tarefas devem continuar seguindo esse tutorial.*

No momento de iniciar esta atividade é esperado que o aluno já tenha realizado a importação da imagem de sua máquina virtual no VirtualBox.

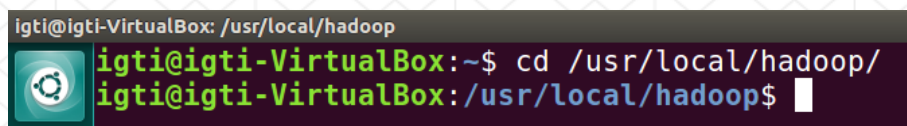
Após executar o login (*login: igti senha: igti*) na máquina virtual, inicie o Terminal do Linux, conforme demonstrado na Figura 1.

Figura 1 – Tela inicial da máquina virtual.

A ferramenta Hadoop já foi previamente instalada no ambiente virtual e encontra-se no diretório `/usr/local/hadoop`. Sendo assim, devemos ir até o diretório de instalação do Hadoop utilizando o seguinte comando:

```
cd /usr/local/hadoop
```

Após a digitação do comando acima, teremos a tela apresentada na Figura 2.

Figura 2 – Diretório de instalação do Hadoop.

1. Formatar o sistema de arquivos distribuídos do Hadoop:

Antes de tudo é necessário realizar a formatação do sistema de arquivos distribuídos do Hadoop (HDFS). Esse processo envolve dois passos, sendo: (i) eliminar os diretórios temporários; (ii) realizar a formatação do HDFS.

- (i) Delete todos os subdiretórios que estão dentro da pasta tmp localizada dentro do diretório padrão do Hadoop (`/usr/local/hadoop`). Utilize o comando abaixo:

```
rm -r /usr/local/hadoop/tmp/*
```

- (ii) Para formatar o HDFS, execute o comando abaixo:

`/usr/local/hadoop/bin/hdfs namenode -format`

Após alguns segundos, a tela apresentada na Figura 3 será exibida:

Figura 3 – Tela de formatação do Hadoop (resultado).

```

2018-12-06 15:39:26,326 INFO blockmanagement.BlockManager: maxNumBlocksToLog = 1000
2018-12-06 15:39:26,582 INFO util.GSet: Computing capacity for map INodeMap
2018-12-06 15:39:26,582 INFO util.GSet: VM type = 64-bit
2018-12-06 15:39:26,582 INFO util.GSet: 1.0% max memory 483.4 MB = 4.8 MB
2018-12-06 15:39:26,583 INFO util.GSet: capacity = 2^19 = 524288 entries
2018-12-06 15:39:26,584 INFO namenode.FSDirectory: ACLs enabled? false
2018-12-06 15:39:26,584 INFO namenode.FSDirectory: POSIX ACL inheritance enabled? true
2018-12-06 15:39:26,584 INFO namenode.FSDirectory: XAttrs enabled? true
2018-12-06 15:39:26,585 INFO namenode.NameNode: Caching file names occurring more than 10 times
2018-12-06 15:39:26,593 INFO snapshot.SnapshotManager: Loaded config captureOpenFiles: false, skipCaptureAccessTimeOnlyChange: false, snapshotFailLowSnapshotDescendant: true
2018-12-06 15:39:26,609 INFO util.GSet: Computing capacity for map cachedBlocks
2018-12-06 15:39:26,615 INFO util.GSet: VM type = 64-bit
2018-12-06 15:39:26,616 INFO util.GSet: 0.25% max memory 483.4 MB = 1.2 MB
2018-12-06 15:39:26,616 INFO util.GSet: capacity = 2^17 = 131072 entries
2018-12-06 15:39:26,663 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
2018-12-06 15:39:26,672 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
2018-12-06 15:39:26,672 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
2018-12-06 15:39:26,683 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
2018-12-06 15:39:26,688 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 milli
2018-12-06 15:39:26,696 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2018-12-06 15:39:26,697 INFO util.GSet: VM type = 64-bit
2018-12-06 15:39:26,697 INFO util.GSet: 0.0299999999329447746% max memory 483.4 MB = 148.5 KB
2018-12-06 15:39:26,698 INFO util.GSet: capacity = 2^14 = 16384 entries
2018-12-06 15:39:26,698 INFO namenode.FSImage: Allocated new BlockPoolId: BP-488760947-127.0.1.1-1544117966787
2018-12-06 15:39:26,851 INFO common.Storage: Storage directory /usr/local/hadoop/tmp/dfs/name has been successfully formatted.
2018-12-06 15:39:26,893 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/local/hadoop/tmp/dfs/name/current/fsimage.ckpt_0000000000
00000000 using no compression
2018-12-06 15:39:27,149 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/hadoop/tmp/dfs/name/current/fsimage.ckpt_0000000000000000
00 of size 389 bytes saved in 0 seconds
2018-12-06 15:39:27,212 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2018-12-06 15:39:27,264 INFO namenode.NameNode: SHUTDOWN_MSG:
=====
SHUTDOWN_MSG: Shutting down NameNode at igti-VirtualBox/127.0.1.1
=====
igti@igti-VirtualBox: /usr/local/hadoop$

```

Neste momento você apagou todos os arquivos e diretórios do sistema de arquivos distribuídos do Hadoop (HDFS). Esses comandos específicos do HDFS serão melhor trabalhados nas próximas atividades práticas e também durante nossas aulas. Por enquanto iremos utilizar apenas a formatação. Agora já estamos prontos para iniciar os serviços do Hadoop.

2. Iniciando os serviços do Hadoop:

Estamos dentro do diretório de instalação do Hadoop, e a próxima tarefa será iniciar os serviços da ferramenta. Iremos utilizar um arquivo que encontra-se dentro do diretório `/usr/local/hadoop/sbin` chamado `start-all.sh`. Para isso, execute o comando abaixo:

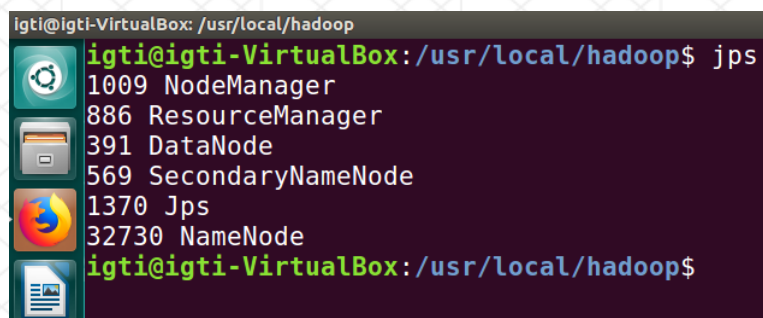
`/usr/local/hadoop/sbin/start-all.sh`

Observe que ao utilizar o comando `start-all` o Hadoop emite uma mensagem informando que esse comando foi descontinuado. Apesar disso, o comando funciona perfeitamente e, caso queira, você pode utilizar os comandos `start-yarn.sh` e `start-dfs.sh`. O efeito será exatamente o mesmo.

Neste momento é esperado que todos os serviços do Hadoop tenham sido iniciados. Para conferir se todos os serviços foram devidamente iniciados (DataNode, ResourceManager, NameNode, SecondaryNameNode e NodeManager), digite o comando `jps` no Terminal. Esse comando lista os serviços Java que estão sendo executados na

máquina. Após a execução do comando `jps`, se tudo estiver correto, a tela da Figura 4 será apresentada. Observe que todos os cinco serviços necessários para a execução do Hadoop encontram-se listados:

Figura 4 – Tela com os cinco serviços do Hadoop inicializados.



```
igti@igti-VirtualBox: /usr/local/hadoop$ jps
1009 NodeManager
886  ResourceManager
391  DataNode
569  SecondaryNameNode
1370 Jps
32730 NameNode
igti@igti-VirtualBox: /usr/local/hadoop$
```

Importante: se ao listar os processos com o comando `jps`, não aparecerem como ativos os processos `DataNode`, `ResourceManager`, `NameNode`, `SecondaryNameNode` e `NodeManager`, você deverá seguir os seguintes passos:

- 1) Pare o serviço do Hadoop: `/usr/local/hadoop/sbin/stop-all.sh`.
- 2) Delete novamente os arquivos temporários: `rm -r /usr/local/hadoop/tmp/*`.
- 3) Veja se os arquivos e diretórios temporários foram realmente excluídos, usando o comando: `ls /usr/local/hadoop/tmp`.
- 4) Formate novamente o HDFS: `/usr/local/hadoop/bin/hdfs namenode -format`.
- 5) Reinicie os serviços do Hadoop: `/usr/local/hadoop/sbin/start-all.sh`.
- 6) Consulte novamente os serviços do Hadoop com o comando `jps`.

Obs.: você não deverá avançar no tutorial caso os cinco processos do Hadoop não estejam em execução.

3. Compilando um programa no Hadoop:

Nesse momento já estamos com os serviços do Hadoop executando plenamente e já podemos construir o nosso primeiro programa. Nesse programa iremos repetir o exemplo apresentado no Capítulo 4 de nossa apostila.

Inicialmente, dentro do diretório `/usr/local/hadoop` foi criado um diretório chamado `ExemploIGTI`. Você deverá navegar nesse diretório e verificar que o mesmo possui um

arquivo e um outro diretório.

O Arquivo build_ExemploIGTI.xml armazena os dados de compilação do nosso programa e o arquivo ExemploIGTI.java, que se encontra dentro do diretório src, possui o código fonte em Java referente ao nosso programa de exemplo.

Vá para o diretório ExemploIGTI utilizando o seguinte comando:

```
cd /usr/local/hadoop/ExemploIGTI/
```

Liste o conteúdo do diretório ExemploIGTI:

```
ls
```

Em seguida, vá para o diretório src utilizando o seguinte comando:

```
cd /usr/local/hadoop/ExemploIGTI/src
```

Liste o conteúdo do diretório src:

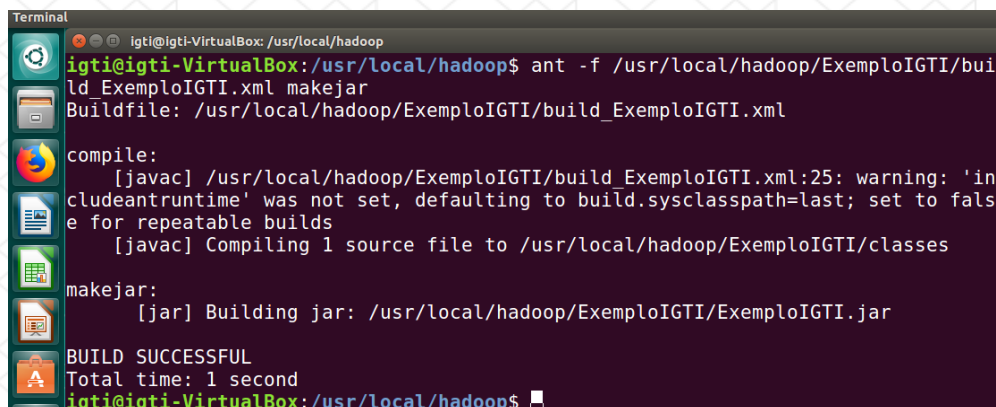
```
ls
```

Para compilar o nosso primeiro programa Hadoop iremos utilizar o Apache Ant, que deverá estar previamente instalado na máquina. No caso do nosso laboratório, o Ant já foi instalado. O comando para compilação é o seguinte:

```
ant -f /usr/local/hadoop/ExemploIGTI/build_ExemploIGTI.xml makejar
```

Após compilar o programa, a mensagem da Figura 5 irá aparecer na tela:

Figura 5 – Tela do resultado da compilação.

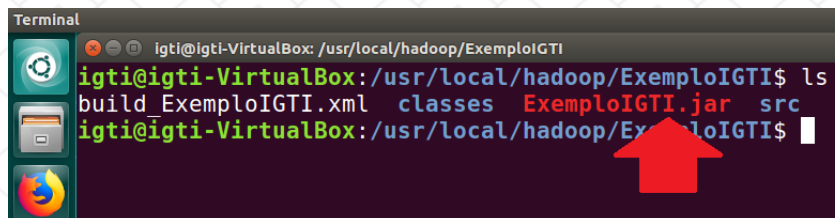


Em seguida, vamos verificar como ficou o conteúdo do nosso diretório ExemploIGTI, dando um comando ls. A Figura 6 apresenta o resultado.

```
cd /usr/local/hadoop/ExemploIGTI
```

```
ls
```

Figura 6 – Novo conteúdo do diretório ExemploIGTI.



Observe que foi criado o arquivo ExemploIGTI.jar. Esse é o arquivo compilado e que será enviado para o Hadoop durante a execução. No nosso próximo tópico iremos utilizar esse arquivo para uma execução do Hadoop.

4. Executando nosso programa:

Após compilar nosso programa e gerar o nosso jar, iremos submetê-lo para a execução em um *job* Hadoop/MapReduce. A linha de comando que iremos utilizar está destacada abaixo:

```
/usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/ExemploIGTI/ExemploIGTI.jar  
IGTI.ExemploIGTI
```

O primeiro parâmetro que utilizamos é o `/usr/local/hadoop/bin/hadoop`. O arquivo Hadoop que fica dentro da pasta `bin` é o responsável por enviar os programas para a execução do *framework*. Em seguida é passado a palavra `jar`, indicando que iremos enviar um arquivo compilado do tipo `jar` para execução. O terceiro parâmetro é a localização desse `jar` no sistema de arquivos do Linux. Lembre-se que compilamos esse arquivo na pasta `ExemploIGTI`. Por último estamos informando a classe que possui o método *main* (`IGTI.ExemploIGTI`).

A Figura 7 apresenta o comando sendo enviado e o *job* sendo executado.

Figura 7 – Logs de execução de um *job* Hadoop/MapReduce.

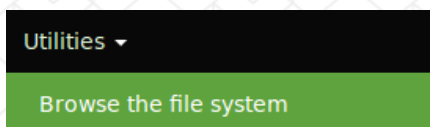
```

Terminal
igti@igti-VirtualBox: /usr/local/hadoop$ /usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/ExemploIGTI/ExemploIGTI.jar IGTI.ExemploIGTI
2019-01-02 14:06:11,186 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2019-01-02 14:06:11,194 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2019-01-02 14:06:12,580 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/igti/.staging/job_1546445115443_0001
2019-01-02 14:06:12,860 INFO mapred.FileInputFormat: Total input files to process : 1
2019-01-02 14:06:13,473 INFO mapreduce.JobSubmitter: number of splits:2
2019-01-02 14:06:13,660 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2019-01-02 14:06:14,412 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1546445115443_0001
2019-01-02 14:06:14,418 INFO mapreduce.JobSubmitter: Executing with tokens: []
2019-01-02 14:06:14,946 INFO conf.Configuration: resource-types.xml not found
2019-01-02 14:06:14,947 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2019-01-02 14:06:15,866 INFO impl.YarnClientImpl: Submitted application application_1546445115443_0001
2019-01-02 14:06:16,030 INFO mapreduce.Job: The url to track the job: http://igti-VirtualBox:8088/proxy/application_1546445115443_0001/
2019-01-02 14:06:16,035 INFO mapreduce.Job: Running job: job_1546445115443_0001
2019-01-02 14:06:36,841 INFO mapreduce.Job: Job job_1546445115443_0001 running in uber mode : false
2019-01-02 14:06:36,928 INFO mapreduce.Job: map 0% reduce 0%
2019-01-02 14:06:58,974 INFO mapreduce.Job: map 100% reduce 0%
2019-01-02 14:07:15,780 INFO mapreduce.Job: map 100% reduce 100%
2019-01-02 14:07:18,824 INFO mapreduce.Job: Job job_1546445115443_0001 completed successfully
2019-01-02 14:07:19,026 INFO mapreduce.Job: Counters: 53
File System Counters
  FILE: Number of bytes read=306
  FILE: Number of bytes written=605744
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=3306
  HDFS: Number of bytes written=71
  
```

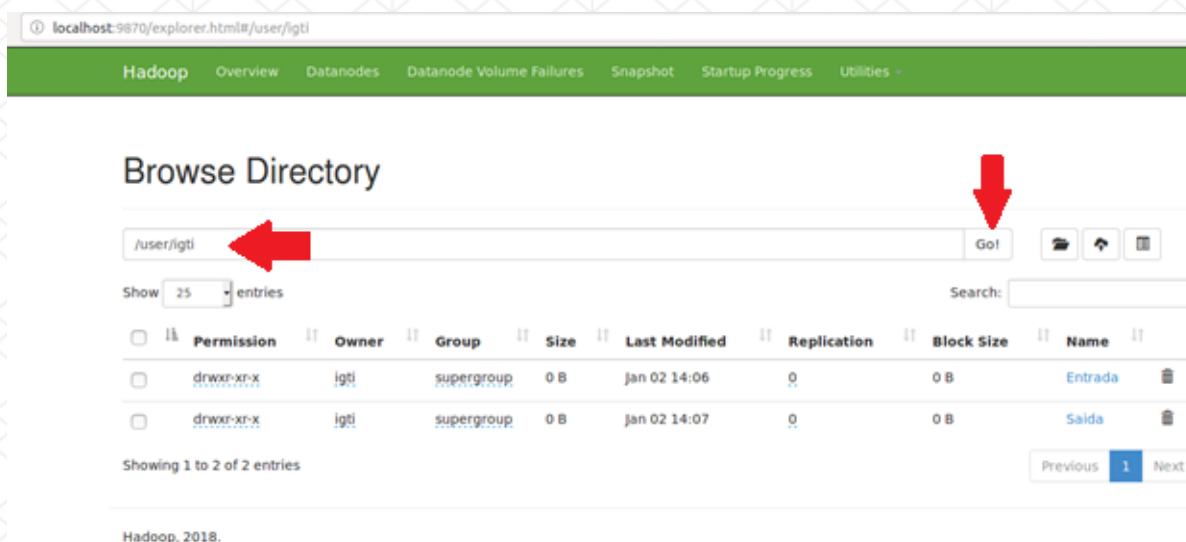
A Figura 8 apresenta os diretórios no HDFS após a execução do *job*.

Figura 8 – Diretórios criados no HDFS após a execução do *job*.

Para acessar os Diretórios criados no HDFS, acesse: <http://localhost:9870> clique em “Utilities” e depois em “Browse the file system”.

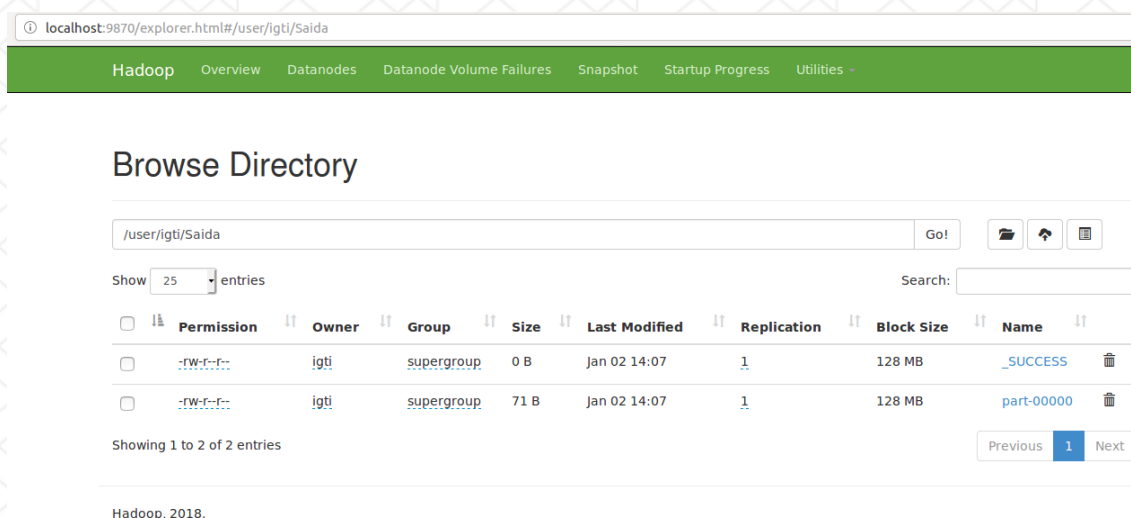


Inserir diretório “/user/igti” no campo abaixo do título “Browse Directory” e clique em “Go!”, conforme indicado na imagem abaixo:



Observe que foi criado um diretório chamado Saída. Nele está o resultado do *job*, conforme apresenta a Figura 9.

Figura 9 – Conteúdo do diretório Saída, no HDFS.



A Figura 10 apresenta o conteúdo do arquivo part-00000 com o resultado do nosso processamento (médias de itens vendidos por cliente).

Figura 10 – Resultado do processamento.



5. Conclusão:

Nesta atividade realizamos a compilação e execução de um programa Hadoop. Para o correto acompanhamento da atividade é necessário que o aluno acompanhe o capítulo 4, principalmente a aula 4.5.