

Atividade de Aula – Trabalho Prático

| Disciplina | HDM – Desenvolvimento de Soluções com MapReduce utilizando Hadoop |
|---------------|---|
| Nome do aluno | THIAGO FERNANDES DA SILVA OLIVEIRA |

Objetivos

O Trabalho Prático em laboratório possui como objetivos principais:

- ✓ Criar um programa para executar no Hadoop/Hadoop que classifique clientes de acordo com o valor de suas compras.
- ✓ Compilar o código criado.
- ✓ Executar o programa.
- ✓ Analisar os resultados gerados.

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

Enunciado

1. Alterando o código fonte

Para esse trabalho prático deseja-se classificar e bonificar os clientes de acordo com a maior compra que ele realizou na empresa, aproveitando o código-fonte que foi criado na Atividade 02.

Caso a compra realizada seja maior que R\$50.000,00, deverá ser informado que o cliente possui um bônus de 15%. Em caso de compras entre R\$20.000,00 e R\$49.999,99, o bônus será de 8%. Caso a compra tenha sido menor que R\$20.000,00 o sistema deverá informar que não há bonificação a ser concedida.

A função Reduce deverá ser alterada para a inclusão da frase que destaca o bônus a ser concedido. Para isso, sugerimos a utilização do código abaixo:

```
double maior = 0.0;
Text value = new Text();
String classificacao;

while (values.hasNext()) {
    value = values.next();
    if (Double.parseDouble(value.toString()) > maior)
        maior = Double.parseDouble(value.toString());
}

if ( maior >= 50000 )
    classificacao = "Bonus 15%";
else if ((maior > 20000) && (maior < 50000))
    classificacao = "Bonus 8%";
else
    classificacao = "Nao ha bonus";

value.set(String.valueOf(maior) + "\t" + classificacao);
output.collect(key, value);
```

Como adicional, pede-se que após a mensagem de bonificação seja inserido o valor do bônus. Exemplo: caso a maior compra do cliente tenha sido de R\$60.000,00, o resultado deverá ser:

099 Bonus 15% 9000.00

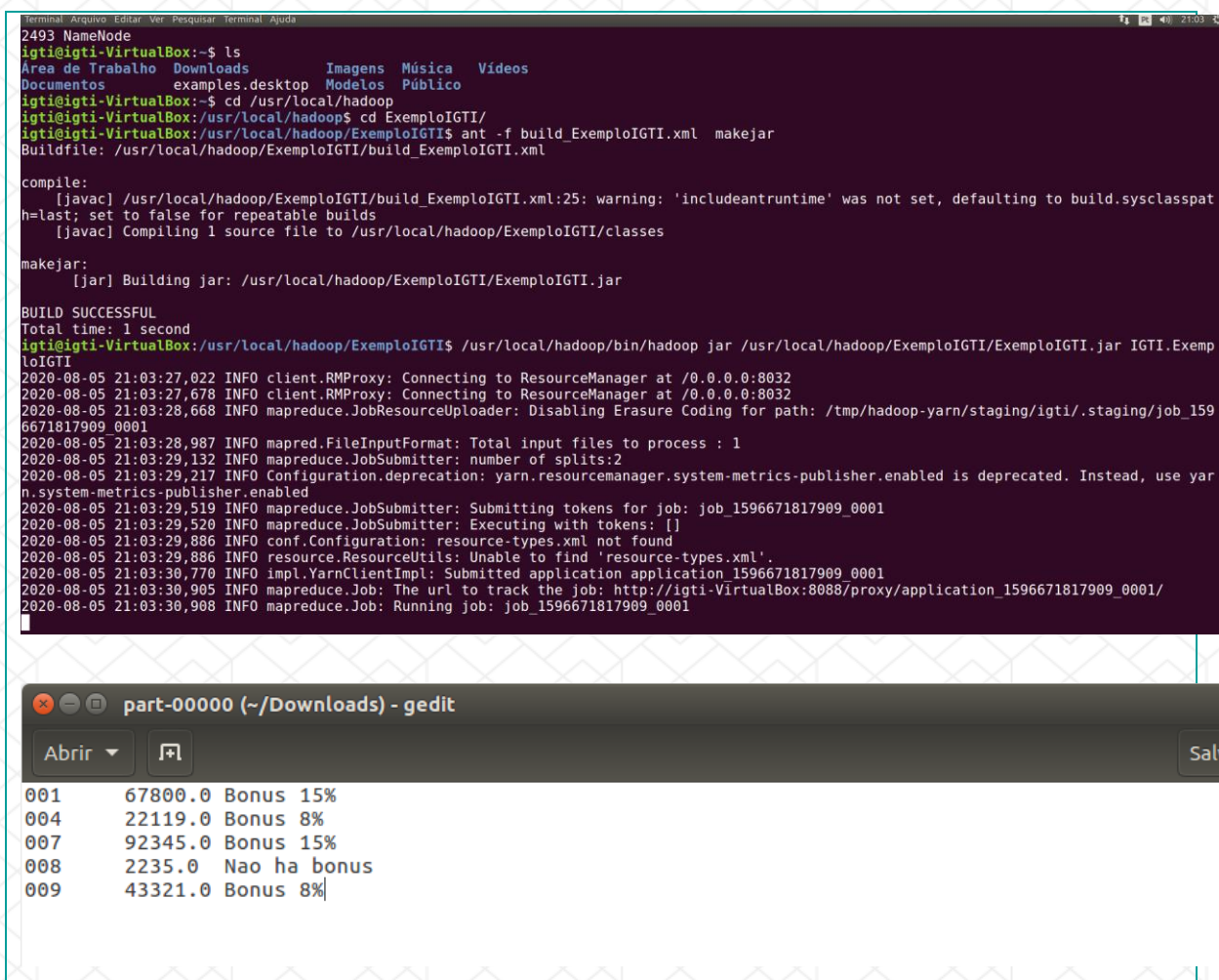
Após a alteração do código, você deverá realizar a compilação e execução. Para isso siga os mesmos passos definidos nas Atividades 01 e 02, sendo eles:

- Apague o conteúdo do diretório temp.
- Formate o HDFS (para apagar algum dado anterior que ainda esteja por lá).
- Inicie os serviços do Hadoop.
- Certifique-se que todos os serviços (05) foram iniciados corretamente.
- Compile o ExemploIGTI.java já com sua alteração.
- Execute o programa.

- Dê *print* na tela de execução e armazene a imagem.
- Verifique no HDFS (pelo browser ou linha de comando) o resultado gerado.
- Dê *print* no conteúdo do arquivo gerado no HDFS e armazene a imagem.

Atividades

1. Para o programa ExemploIGTI já com as alterações de bonificação realizadas, apresentar o print da tela de execução do programa (terminal) e o conteúdo do arquivo gerado no HDFS.



```

Terminal: Arquivo Editar Ver Pesquisar Terminal Ajuda
2493 NameNode
igti@igti-VirtualBox:~$ ls
Área de Trabalho Downloads Imagens Música Vídeos
Documentos examples.desktop Modelos Público
igti@igti-VirtualBox:~$ cd /usr/local/hadoop
igti@igti-VirtualBox:/usr/local/hadoop$ cd ExemploIGTI/
igti@igti-VirtualBox:/usr/local/hadoop/ExemploIGTI$ ant -f build_ExemploIGTI.xml makejar
Buildfile: /usr/local/hadoop/ExemploIGTI/build_ExemploIGTI.xml

compile:
[javac] /usr/local/hadoop/ExemploIGTI/build_ExemploIGTI.xml:25: warning: 'includeantruntime' was not set, defaulting to build.sysclasspath=last; set to false for repeatable builds
[javac] Compiling 1 source file to /usr/local/hadoop/ExemploIGTI/classes

makejar:
[jar] Building jar: /usr/local/hadoop/ExemploIGTI/ExemploIGTI.jar

BUILD SUCCESSFUL
Total time: 1 second
igti@igti-VirtualBox:/usr/local/hadoop/ExemploIGTI$ /usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/ExemploIGTI/ExemploIGTI.jar IGTI.ExemploIGTI
2020-08-05 21:03:27,022 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2020-08-05 21:03:27,678 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2020-08-05 21:03:28,668 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/igti/.staging/job_1596671817909_0001
2020-08-05 21:03:28,987 INFO mapred.FileInputFormat: Total input files to process : 1
2020-08-05 21:03:29,132 INFO mapreduce.JobSubmitter: number of splits:2
2020-08-05 21:03:29,217 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2020-08-05 21:03:29,519 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1596671817909_0001
2020-08-05 21:03:29,520 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-08-05 21:03:29,886 INFO conf.Configuration: resource-types.xml not found
2020-08-05 21:03:29,886 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-08-05 21:03:30,770 INFO impl.YarnClientImpl: Submitted application application_1596671817909_0001
2020-08-05 21:03:30,905 INFO mapreduce.Job: The url to track the job: http://igti-VirtualBox:8088/proxy/application_1596671817909_0001/
2020-08-05 21:03:30,908 INFO mapreduce.Job: Running job: job_1596671817909_0001

part-00000 (~/.Downloads) - gedit
Abrir + Salvar
001 67800.0 Bonus 15%
004 22119.0 Bonus 8%
007 92345.0 Bonus 15%
008 2235.0 Nao ha bonus
009 43321.0 Bonus 8%
  
```

2. No programa que você executou, quantas vezes a função Map foi “chamada”?

20 vezes pois existem 20 registros de entrada.

Map-Reduce Framework

```
Map input records=20
Map output records=20
Map output bytes=260
Map output materialized bytes=312
Total input data=220
```

3. No programa que você executou, quantas vezes a função Reduce foi “chamada”?

O Reduce foi executado 5 vezes, pois existem 5 chaves distintas ou 5 grupos.

```
Combine output records=0
Reduce input groups=5
Reduce shuffle bytes=312
Reduce input records=20
Reduce output records=5
```

4. Quantos grupos de registros (agrupados pela chave) temos na função Reduce?

Temos 5 grupos de registros.

5. Quanto tempo foi gasto pelas tarefas Map e Reduce?

O tempo total = tempo map + tempo reduce = 18897ms + 5514ms = 24411ms

```
Job Counters
  Launched map tasks=2
  Launched reduce tasks=1
  Data-local map tasks=2
  Total time spent by all maps in occupied slots (ms)=18897
  Total time spent by all reduces in occupied slots (ms)=5514
  Total time spent by all map tasks (ms)=18897
  Total time spent by all reduce tasks (ms)=5514
  Total vcore-milliseconds taken by all map tasks=18897
  Total vcore-milliseconds taken by all reduce tasks=5514
  Total megabyte-milliseconds taken by all map tasks=19350528
  Total megabyte-milliseconds taken by all reduce tasks=5646336
```

Obs.: Para as perguntas 2, 3, 4 e 5 os logs que são gerados na tela do Terminal podem ser úteis.