UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO

**Roteiro de Aula Prática – Apache Spark**

DISCIPLINA: DCA0132 – ENGENHARIA DE DADOS

PROFESSOR: Carlos Manuel Dias Viegas

Esta prática consiste em criar um cluster com o Apache Spark para executar o processamento de dados de maneira distribuída.

**ATENÇÃO**: Este roteiro é uma sugestão para a implantação do Apache Spark (sobre o Apache Hadoop) nativamente em uma máquina linux. Entretanto, nesta prática será utilizada uma solução baseada em containers Docker, na qual o Apache Hadoop e Apache Spark já estão devidamente configurados. Portanto, este roteiro não precisa ser seguido, bastando apenas fazer o download dos arquivos no repositório do github:

1. Os recursos Docker para a implantação do Apache Hadoop + Apache Spark estão disponíveis em:

<https://github.com/cmdviegas/docker-hadoop-cluster>

1. Ao inicializar o cluster, proceda com os testes descritos nos passos 7 e 8.

**Requisitos**

Hadoop Yarn previamente configurado.

**Configuração do sistema**

1. Baixar o Apache Spark em um terminal no **node-master:**

wget –c https://dlcdn.apache.org/spark/spark-3.3.1/spark-3.3.1-bin-hadoop3.tgz

1. Extrair para a pasta /home/user/hadoop-3.3.4/spark
2. Editar as variáveis do ambiente linux, adicionando/alterando:

Arquivo: **~/.bashrc**

|  |
| --- |
| SPARK\_HOME=$HADOOP\_HOME/spark  PATH="$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin:$HADOOP\_HOME/spark/bin:$HADOOP\_HOME/spark/sbin" |

Em seguida digite **bash** no terminal.

1. Configurar o ambiente do Spark para se integrar ao Yarn.

cd ~/hadoop-3.3.4/spark/conf/

mv spark-defaults.conf.template spark-defaults.conf

mv spark-env.sh.template spark-env.sh

Editar o arquivo abaixo e definindo o Spark Master como sendo o Yarn:

Arquivo: **~/hadoop-3.3.4/spark/conf/spark-defaults.conf**

|  |
| --- |
| spark.master yarn |
| spark.yarn.stagingDir hdfs://node-master:9000/user/ubuntu |

Configurar a alocação de memória RAM para o Spark:

|  |
| --- |
| spark.driver.memory 512m  spark.executor.memory 512m |

Editar o arquivo abaixo e acrescentar a variável do ambiente Hadoop...

Arquivo: **~/hadoop-3.3.4/spark/conf/spark-env.sh**

|  |
| --- |
| export HADOOP\_CONF\_DIR=${HADOOP\_HOME}/etc/hadoop |

1. Para monitorar a execução das tarefas com o Spark, é necessário acrescentar algumas definições ao arquivo de configuração:

Arquivo: **~/hadoop-3.3.4/spark/conf/spark-defaults.conf**

|  |
| --- |
| spark.eventLog.enabled true  spark.eventLog.dir hdfs://node-master:9000/spark-logs  spark.history.provider org.apache.spark.deploy.history.FsHistoryProvider  spark.history.fs.logDirectory hdfs://node-master:9000/spark-logs  spark.history.fs.update.interval 10s  spark.history.ui.port 18080 |

Porém, é necessário criar a pasta spark-logs dentro do HDFS

hdfs dfs -mkdir /spark-logs

Em seguida iniciar o processo:

start-history-server.sh

1. Como sugestão para otimizar a subsmissão de tarefas no Spark, envie os arquivos JAR, referentes às bibliotecas de execução do Spark, para o sistema HDFS de forma a agilizar a execução das tarefas (opcional)

- Criar os diretórios e enviar os arquivos:

|  |
| --- |
| hdfs dfs -mkdir /spark-libs  hdfs dfs -put ~/hadoop-3.3.4/spark/jars/\*.jar /spark-libs/ |

- Editar o arquivo spark-defaults.conf e adicionar:

|  |
| --- |
| spark.yarn.jars hdfs://node-master:9000/spark-libs/\*.jar |

1. Inicie o HDFS e o ResourceManager do Yarn e execute uma tarefa de exemplo com o Spark (cálculo de Pi):

spark-submit --deploy-mode client --class org.apache.spark.examples.SparkPi $SPARK\_HOME/examples/jars/spark-examples\_\*.jar 10

1. Acompanhe a execução da tarefa do item 6 abrindo o navegador no endereço:

http://node-master:18080

