Pós-Graduação – Big Data
Disciplina: MapReduce e Spark
Prof. Alessandro Binhara

Por: ***Cristiane Fagundes***
Tiyomi Nakaba

Uma tradução do tutorial sobre Hadoop Streaming disponível em https://www.tutorialspoint.com/hadoop/hadoop_streaming.htm sobre Hadoop Streaming.

HADOOP STREAMING

O Hadoop Streaming (HS) é uma ferramenta distribuída pelo Hadoop. Ele permite criar e executar tarefas de Map (mapeamento) e de Reduce (redução) de qualquer script ou executável.

Exemplo do Python
Considerando a contagem de palavras usando a linguagem Python. No Hadoop, qualquer trabalho deve ter duas fases: Map e Reduce, essas fases são escritas no script do Python para executá-las no Hadoop. Da mesma forma, os scripts podem ser em Perl e Ruby.

Código da fase Mapper

```
!/usr/bin/python
import sys
# Input takes from standard input for myline in sys.stdin:
# Remove whitespace either side myline = myline.strip()
# Break the line into words words = myline.split()
# Iterate the words list for myword in words:
# Write the results to standard output print '%s\t%s' % (myword, 1)
```

É preciso garantir que o arquivo tenha permissão para executar. O script “chmod +x /home/ expert/hadoop−1.2.1/mapper.py” deve resolver qualquer bloqueio.

Código da fase Reducer

```
#!/usr/bin/python
from operator import itemgetter
import sys
current_word = ""
current_count = 0
word = ""
# Input takes from standard input for myline in sys.stdin:
# Remove whitespace either side myline = myline.strip()
# Split the input we got from mapper.py word, count = myline.split('\t', 1)
# Convert count variable to integer
   try:
      count = int(count)
except ValueError:
   # Count was not a number, so silently ignore this line continue
if current_word == word:
   current_count += count
else:
   if current_word:
      # Write result to standard output print '%s\t%s' % (current_word, current_count)
   current_count = count
   current_word = word
# Do not forget to output the last word if needed!
if current_word == word:
   print '%s\t%s' % (current_word, current_count)
```

Salve os códigos de Map e de Reducer com a extensão .py no diretório Hadoop. Garanta que os arquivos tenham permissão para execução. Os comandos “chmod +x mapper.py” e “chmod +x reducer.py” devem resolver erros por falta de permissão.

O Python é sensível a tabulação e por isso busque os códigos através do link original deste tutorial: https://www.tutorialspoint.com/hadoop/hadoop_streaming.htm.

Execução do WordCount

```
$ $HADOOP_HOME/bin/hadoop jar contrib/streaming/hadoop-streaming-1.
2.1.jar \
   -input input_dirs \
   -output output_dir \
   -mapper <path/mapper.py \
   -reducer <path/reducer.py
```

Onde “\” é usado para continuação da linha para uma leitura clara.

Por exemplo, o código abaixo deve ser escrito em uma única linha contínua:

```
./bin/hadoop jar contrib/streaming/hadoop-streaming-1.2.1.jar -input myinput
-output myoutput -mapper /home/expert/hadoop-1.2.1/mapper.py -reducer
/home/expert/hadoop-1.2.1/reducer.py
```

Execução do Streaming

No exemplo acima, tanto o Map quanto o Reducer estão em scripts do Python que são lidos como entrada padrão e emitem saída padrão. A utilidade irá criar um Map/Reducer, submetê-lo a um cluster apropriado, monitorar o progresso do

processamento até que seja concluído. Se um script for especificado por mapeadores, cada tarefa de mapeamento irá iniciar o script como um processo independente. À medida em que a tarefa do mapeador é executada, ela converte suas entradas em linhas e alimenta as linhas para a entrada padrão (STDIN) do processo. Enquanto isso, o mapeador coleta as saídas orientadas à linha da saída padrão (STDOUT) do processo e converte cada linha em um par chave/valor, que é coletado como a saída do mapeador. Por padrão, o prefixo de uma linha até o primeiro caractere de tabulação é a chave e o restante da linha excluindo o caractere de tabulação será o valor. Se não houver nenhum caractere de tabulação na linha, então a linha inteira será considerada como a chave e o valor será nulo. No entanto, isso pode ser personalizado, caso haja alguma necessidade.

Quando um script é especificado por redutores, cada tarefa redutor irá iniciar o script como um processo separado, e assim o redutor é inicializado. À medida que a tarefa do redutor é executada, ela converte seus pares chave/valores de entrada em linhas e alimenta as linhas para a entrada padrão (STDIN) do processo. Entretanto, o Reducer coleta as saídas orientadas a linha da saída padrão (STDOUT) do processo, converte cada linha em um par chave/valor, que é coletado como a saída do redutor. Por padrão, o prefixo de uma linha até o primeiro caractere de tabulação é a chave e o restante da linha excluindo o caractere de tabulação é o valor. No entanto, assim como no Mapper, isso pode ser personalizado de acordo com requisitos específicos.

Comandos Importantes

Parâmetros	**Descrição**
-input directory/file-name	Endereço do arquivo origem para ser mapeado. (Obrigatório)
-output directory-name	Endereço para salvar o resultado da redução. (Obrigatório)
-mapper executable or script or JavaClassName	Mapeador executável. (Obrigatório)
-reducer executable or script or JavaClassName	Redutor executável. (Obrigatório)
-file file-name	Torna os executáveis mapeador, redutor ou combinador disponíveis localmente em nós de computação.

-inputformat JavaClassName	A classe que você informar deve retornar pares chave/valor em classe de texto. Se não for informado, TextInputFormat é usado como padrão.
-outputformat JavaClassName	A classe que você informar deve retornar pares chave/valor da classe de texto. Se não for informado, TextOutputformat é usado como padrão.
-partitioner JavaClassName	Separador entre as chaves
-combiner streamingCommand or JavaClassName	Executável combiner para a saída do mapa.
-cmdenv name=value	Passa a variável de ambiente para comandos de streaming.
-inputreader	Para compatibilidade em passo anterior: especifica uma classe de leitor de registro (em lugar de uma classe de formato de entrada).
-verbose	Saída detalhada.
-lazyOutput	Cria a saída lentamente. Por exemplo, se o formato de saída é baseado em FileOutputFormat, o arquivo de saída é criado somente na primeira chamada para a coleta da saída (ou Context.write).
-numReduceTasks	Especifica o número de redutores.
-mapdebug	Script a ser adotado quando houver falha no Map.
-reducedebug	Script a ser adotado quando houver falha no Reduce

Fonte: repositório tutorialspoint.com, disponível através do link
https://www.tutorialspoint.com/hadoop/hadoop_streaming.htm