Curso: Business Intelligence com ênfase em Big Data

Disciplina: Processamento Massivo Paralelo Hadoop e MapReduce

Profa. Gabriela Mota

[gbrlamota@gmail.com](mailto:gbrlamota@gmail.com)

Relatório da Atividade 2

# Equipe:

Adailton Nascimento de Oliveira Silva

Bárbara Leandra do Espírito Santo Ramos

Taynã Heron do Espírito Santo Ramos

Josué Souza Nascimento

# Requisitos/Tarefa(s):

Tarefa 1: Rodar o exemplo WordCount tal e qual foi fornecido pela distribuição do

Hadoop, observar os dados de saída.

Tarefa 2: Fazer as alterações solicitadas para gerar o WordCount personalizado

Tarefa 2.1: Escreva um código que execute esse filtro no método map da classe TozenizerMapper.

Tarefa 2.2: Crie um combiner que agregue os animais pela chave antes de executar o Shuffle.

# Plataforma de desenvolvimento da atividade:

* Foi utilizado o Docker Tools Box que é uma ferramenta que usa uma VM por trás para poder instalar o docker no Windows.
* Para usar o Hadoop usei algumas imagens docker(repositório das imagenns <https://github.com/big-data-europe/docker-hadoop>), formando um docker-compose que continha os recursos necessários para instalar o hadoop e suas dependências .
* Subir os “containers” com o docker-compose up -d, habilitando o serviço do hadoop podendo ser acessado através da interface no browser <http://192.168.99.100:9870/>

# Desenvolvimento da atividade:

1. Acessei o contêiner através do comando “docker exec -it namenode bash”.
2. Instalei o git no contêiner docker.
3. Criamos um repositório no github(<https://github.com/Adailton90/Hadoop-WordCount> ) onde subimos os arquivos e comandos executados, para no caso se algum integrante da equipe precisasse fazer alterações ou realizar testes feitos por outros membros, bastava dar um git pull ou push para acompanhar ou subir as mudanças.
4. Baixamos o repositório com o arquivo intup.txt e os arquivos “.jar”.
5. Subimos o arquivo input.txt para o hdfs do hadoop através “hdfs dfs -put input.txt input”
6. O arquivo [wordcountjava-1.0-SNAPSHOT.jar](https://github.com/Adailton90/Hadoop-WordCount/blob/master/wordcountjava-1.0-SNAPSHOT.jar) é com o “word count” fazendo o mapReducer normal,

# Código-fonte

# O trecho de código que foi alterado no metódo map

public void map(Object key, Text value, Context context

) throws IOException, InterruptedException {

StringTokenizer itr = new StringTokenizer(value.toString());

while (itr.hasMoreTokens()) {

if(**!itr.nextToken().equalsIgnoreCase("Wolf")**){

word.set(itr.nextToken());

context.write(word, one);

}

}

}

}

}

}

}

# Log de execução

[Cole aqui o log final de execução da atividade com sucesso.]

# Conclusão

[Registre aqui as conclusões finais da equipe. Reflita sobre o nível de dificuldade, como vocês gerenciaram o trabalho de forma remota, problemas e pontos positivos.]