Alunos:
Carla Rocha Cangussú - 170085023
Guilherme Keyti Cabral Kishimoto - 190088257
João Victor Correia de Oliveira - 190089792
Matheus Pimentel Leal - 150141629
Thalisson Alves G. De Jesus - 190117401

Relatório do Laboratório de Hadoop

1. Introdução

Neste laboratório, nosso objetivo foi montar um cluster Hadoop usando Docker e rodar uma aplicação MapReduce para contar palavras em um arquivo de texto grande. Queríamos explorar como o Hadoop lida com grandes volumes de dados e testar sua capacidade de recuperação em caso de falhas.

2. Metodologia

Organização do Grupo:

Nos reunimos por Discord para discutir o laboratório.

Decidimos dividir as tarefas, enquanto alguns de nós ficaram responsáveis pela configuração em Docker, outros ficaram responsáveis pela configuração em máquina virtual. Optaríamos por seguir a abordagem que tivesse um avanço mais significativo.

3. Atividades Realizadas

3.1. Montagem do Cluster Hadoop

Começamos escrevendo o script run.sh, que automatiza a criação de contêineres Docker para o Hadoop.

Criamos uma rede Docker para que os contêineres pudessem se comunicar.

Usamos o Dockerfile para construir a imagem base, instalando o Java, o Hadoop, e configurando o SSH para que os nós pudessem se comunicar.

Cada contêiner foi configurado com os arquivos de configuração do Hadoop, que incluíam detalhes como a lista de nós escravos e as configurações de rede.

3.2. Teste do Framework Hadoop

O que testamos:

Ajustamos o número de nós escravos e observamos como isso impactou o desempenho do cluster.

Resultados:

As mudanças no número de nós escravos mostraram que o Hadoop a quantidade de escravos influencia diretamente no desempenho do programa, quanto mais escravos mais rápido o programa é executado.

3.3. Teste de Tolerância a Faltas e Escalabilidade da Aplicação

Aplicação MapReduce:

Usamos o script generator.py para criar um arquivo de texto com milhões de palavras aleatórias.

Utilizamos o MapReduce com os scripts mapper.py e reducer.py fornecido pelo professor, que realizam a contagem de palavras.

Execução e Resultados:

Rodamos o MapReduce usando o script src/run/wc/run.sh.

Durante a execução, simulamos falhas desligando manualmente alguns nós escravos. Notamos que a continuação do programa ficou um pouco mais lenta devido à redução no número de nós escravos disponíveis.

O Hadoop conseguiu processar o arquivo mesmo com falhas simuladas.

3.4. Conclusão Geral

O que aprendemos:

Montar um cluster Hadoop usando Docker.

O Hadoop lida bem com falhas e consegue redistribuir tarefas sem que o job seja interrompido. Isso foi interessante de ver em ação, pois nos mostrou a robustez do sistema.

Desafios e Soluções:

Tivemos dificuldades em verificar se o programa estava rodando por completo após a falha de um slave. Para resolver isso, adicionamos um contador que soma o número total de palavras processadas e exibimos esse valor no final do programa, o que nos permitiu confirmar se a quantidade de palavras processadas estava de acordo com o esperado.