| **FGA 0211 *-* PSPD:** Turma 01 | |  | **Semestre:** | 2023.2 |
| --- | --- | --- | --- | --- |
| **Nome:** | João Gabriel de Campos de Matos  Victor Rayan Ferreira  Lucas Gomes Caldas  Maria Eduarda Dos Santos Abritta Ferreira  Lucas Braun Vieira Xavier | | **Matrícula:** | 180042238  190044390  212005426  202016945  190033088 |
| **Professor:** | Fernando W. Cruz | |  |  |



Projeto de pesquisa - Hadoop



# Introdução

O conceito de Big Data tem ganhado destaque na comunidade acadêmica e empresarial devido à crescente disponibilidade de grandes volumes de dados e à necessidade de extrair informações significativas desses dados para a tomada de decisões. O Apache Hadoop, uma plataforma derivada de iniciativas do Google, surge como uma ferramenta fundamental para lidar com os desafios do Big Data.

Esse projeto tem o objetivo de proporcionar a oportunidade de experimentar o Apache Hadoop, explorando seus benefícios e compreendendo as diferenças em relação aos métodos tradicionais de desenvolvimento de aplicações. O foco está na construção de um ambiente de experimentação com Hadoop.

As atividades de laboratório incluem a montagem de um cluster Hadoop básico, testes no framework Hadoop com alterações para avaliar seu desempenho, uma análise detalhada do comportamento do Hadoop na execução de aplicações MapReduce, e testes de tolerância a falhas e escalabilidade da aplicação em um ambiente distribuído.

# Metodologia utilizada e Encontros

A metodologia que utilizaremos no projeto será o Scrum, que é um framework ágil amplamente utilizado para gerenciar projetos e desenvolvimento de produtos. Ele é conhecido por sua flexibilidade e foco na entrega iterativa e incremental de resultados. No Scrum, há três papéis principais que dividiremos entre o grupo:

* **Product Owner**: Esse papel será realizado pelo aluno ***Lucas Gomes Caldas***, que será responsável por definir as funcionalidades e prioridades do projeto, representando os interesses solicitados pelo professor para a conclusão desse projeto. Ele tomará as decisões sobre o que será desenvolvido e em que ordem.
* **Scrum Master**: Esse papel será realizado pela aluna ***Maria Eduarda Dos Santos Abritta Ferreira***, que será a facilitadora do processo Scrum. Ela ajudará a equipe a entender e seguir as práticas do Scrum, remove obstáculos que possam impedir o progresso e promove um ambiente de trabalho colaborativo.
* **Equipe de Desenvolvimento**: Esse papel será realizado por todos os alunos. É quem realiza o trabalho de desenvolvimento.

No nosso projeto em Scrum, incluiremos esses eventos:

* **Daily Scrum**: É uma reunião diária curta em que a equipe compartilha o progresso, discute obstáculos e planeja o trabalho para o dia seguinte.

| **Encontro** | **Ata de decisões** |
| --- | --- |
| **12/09/2023** | **Horário**: 12:40 às 14h  **Local**: UnB - FGA  **Participantes**: João Gabriel, Victor Rayan, Lucas Gomes, Maria Eduarda e  Lucas Braun  **Pauta**:  - Discussão sobre os primeiros passos para instalar o Hadoop.  - Definição de responsabilidades iniciais.  **Decisões Tomadas**:  - Foi decidido que João Gabriel e Lucas Gomes, serão os responsáveis por realizar a pesquisa inicial sobre os requisitos de software necessários para a instalação do Hadoop.  - Maria ficará encarregada de identificar os melhores tutoriais e documentação online para a instalação do Hadoop.  - Victor e Lucas Braun irão verificar a disponibilidade de recursos de servidor e rede para o cluster Hadoop.  **Ações a Serem Tomadas:**  - João Gabriel e Lucas Gomes devem apresentar um relatório sobre os requisitos de software na próxima reunião, que acontecerá em 14 de setembro de 2023.  - Maria deve compartilhar os links para os tutoriais e documentação até 13 de agosto de 2023.  - Victor e Lucas Braun devem fornecer informações sobre os recursos de servidor e rede até 13 de agosto de 2023.  **Próxima Reunião:** 14 de setembro de 2023.  **Observações**: Nenhuma observação adicional neste momento. |

| **Encontro** | **Ata de decisões** |
| --- | --- |
| **14/09/2023** | **Horário**: 12:40 às 14h  **Local**: UnB - FGA  **Participantes**: João Gabriel, Victor, Lucas Gomes, Maria Eduarda e  Lucas Braun  **Pauta**:  - Instalação do Hadoop.  - Discussão sobre desafios encontrados durante a instalação.  **Decisões Tomadas**:  - Todos os participantes conseguiram concluir com sucesso a instalação do Hadoop em seus respectivos ambientes, apesar de alguns problemas enfrentados durante o processo.  - Foi decidido que, para melhorar o suporte mútuo, criaremos um grupo de comunicação no Discord, qual poderemos compartilhar problemas e soluções relacionados ao projeto.  **Ações a Serem Tomadas:**  - Cada participante deve se preparar para os próximos passos de testar o contador de palavras.  - Lucas Gomes ficará encarregada de criar o grupo de comunicação no Discord e compartilhar os detalhes de acesso com todos os participantes até o final do dia 14 de setembro de 2023.  **Próxima Reunião:** 19 de setembro de 2023.  **Observações**: Neste encontro, discutimos os desafios enfrentados durante a instalação do Hadoop, mas ficamos satisfeitos em relatar que todos conseguiram concluir com êxito o processo. A criação do grupo de comunicação visa facilitar o compartilhamento de experiências e soluções para futuros problemas relacionados ao Hadoop. |

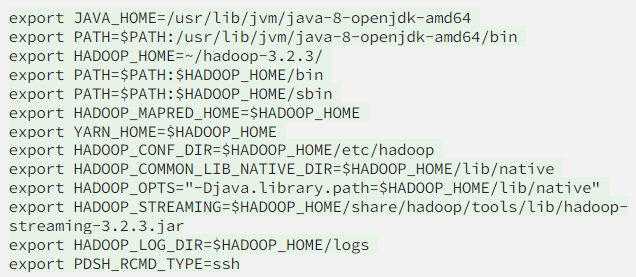
| **Encontro** | **Ata de decisões** |
| --- | --- |
| **19/09/2023** | **Horário**: 12:40 às 14h  **Local**: UnB - FGA  **Participantes**: João Gabriel, Victor, Lucas Gomes, Maria Eduarda e  Lucas Braun  **Pauta**:  - Conclusão dos últimos passos do projeto com Hadoop e contador de palavras.  - Avaliação do cumprimento de todos os requisitos do projeto.  **Decisões Tomadas**:  - Ficou decidido que todos os participantes conseguiram concluir com êxito os últimos passos do projeto, incluindo a implementação do contador de palavras com o Hadoop.  - A avaliação final do projeto revelou que todos os requisitos estabelecidos foram plenamente atendidos.  **Ações a Serem Tomadas:**  - Cada participante deve colocar na documentação descrevendo os detalhes da implementação do contador de palavras no Hadoop.  - Maria ficará responsável por finalizar e enviar a documentação do projeto.  **Próxima Reunião:** Não foi definida uma próxima reunião neste momento, pois o projeto foi concluído com êxito e não há etapas pendentes.  **Observações**: Esta reunião marca a conclusão bem-sucedida do projeto que envolveu a implementação do contador de palavras com o Hadoop. Todos os participantes estão satisfeitos com os resultados alcançados e prontos para compartilhar os detalhes da implementação e os resultados com o professor. |

# Instalando o Hadoop no sistema operacional Ubuntu

# 1. Primeiro, instalamos o java jdk 8



2. Para verificarmos se está lá, usamos o comando *cd /usr/lib/jvm.* Abrimos o arquivo *.bashrc* e colamos estes comandos:



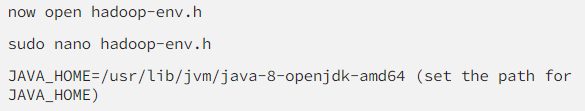
3. Utilizaremos *ssh — shell seguro*, que é um protocolo usado para se conectar com segurança ao servidor/sistema remoto, transfere dados em formato criptografado)



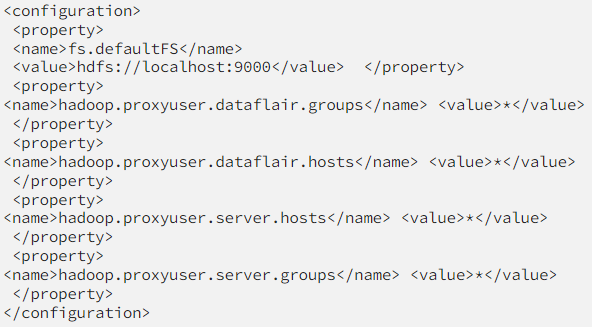
4. Acessamos o site hadoop.apache.org e baixamos o arquivo *tar (hadoop.apache.org)*



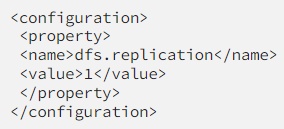
5. Extraimos o arquivo tar *cd hadoop-3.2.3/etc/hadoop*



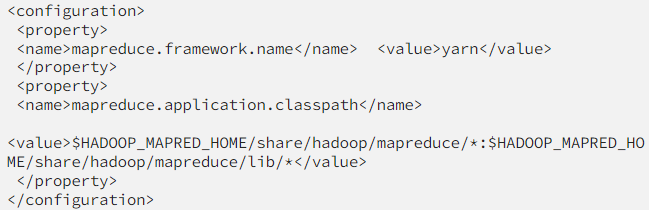
6. Utilizamos *core-site.xml* pois é uma parte fundamental da configuração do ambiente Hadoop, permitindo que você ajuste as configurações centrais do sistema para atender às necessidades específicas do seu cluster



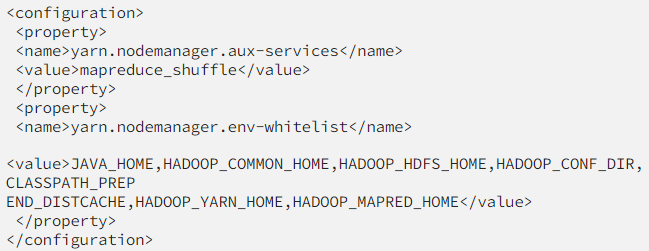
7. Utilizamos *hdfs-site.xml* para configurar o comportamento do HDFS em um ambiente Hadoop, permitindo que os administradores e desenvolvedores ajustem as configurações para atender às necessidades específicas de armazenamento, desempenho e segurança do cluster.



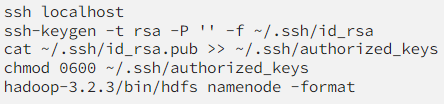
8. Utilizamos *mapred-site.xml* para configurar o comportamento do Apache MapReduce em um ambiente Hadoop, permitindo que administradores e desenvolvedores ajustem as configurações para atender às necessidades específicas de processamento.



9. Utilizamos *yarn-site.xml* para configurar o comportamento do serviço YARN em um ambiente Hadoop, permitindo que administradores e desenvolvedores ajustem as configurações para atender às necessidades específicas de gerenciamento de recursos.



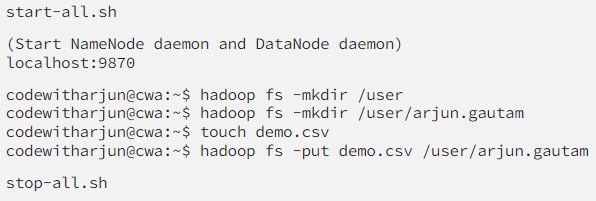
10. Passamos o *ssh* para desempenhar um papel na segurança e na administração de sistemas e redes, proporcionando uma maneira segura de acessar, gerenciar e transferir dados entre sistemas remotos



11. Formatamos o sistema de arquivos



12. E para dar o start, utilizamos:



Além disso na instalação para facilitar a replicação em todos os workers. Utilizamos esse repositório do Github:

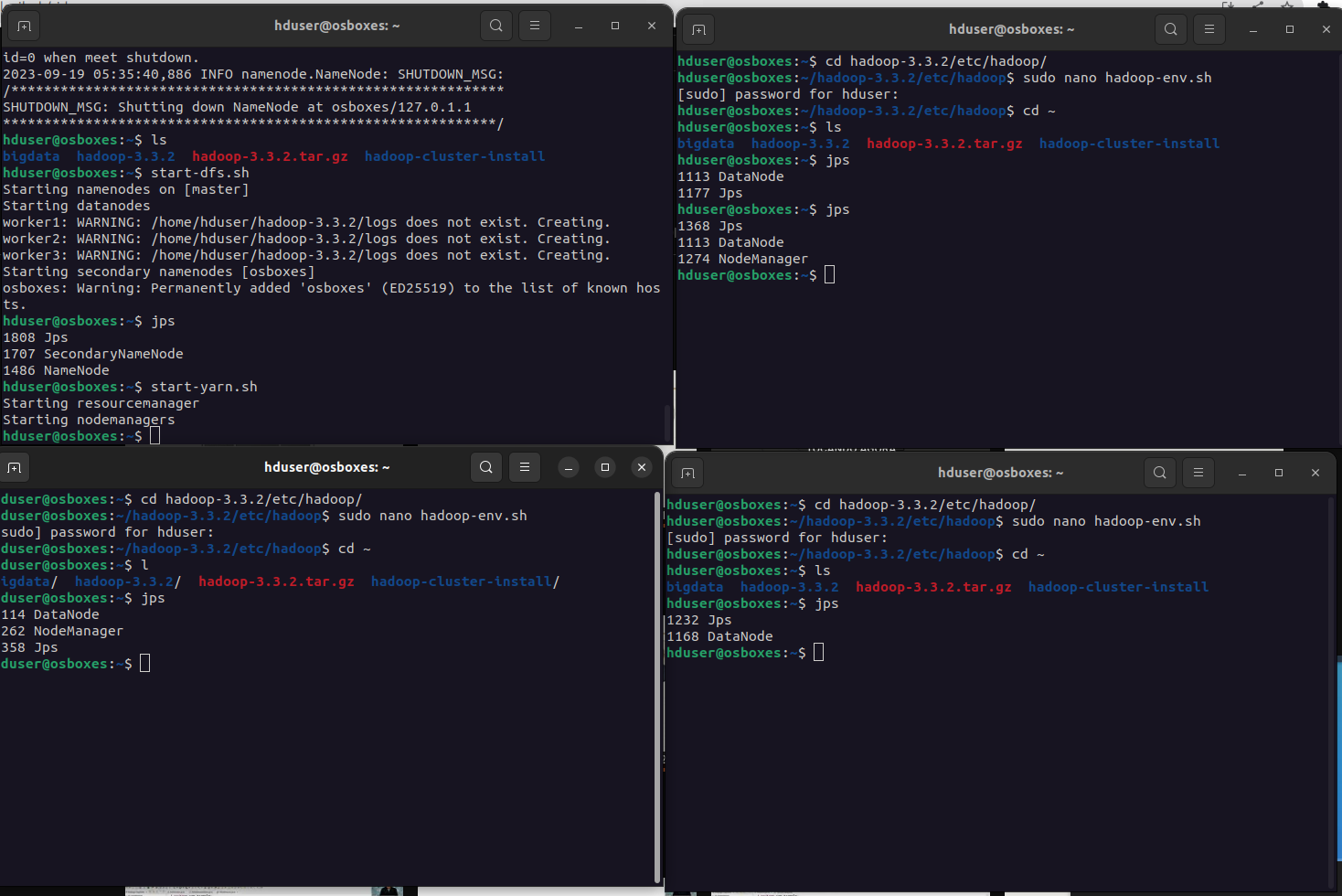
As demais instruções são encontradas no README desse repositório.

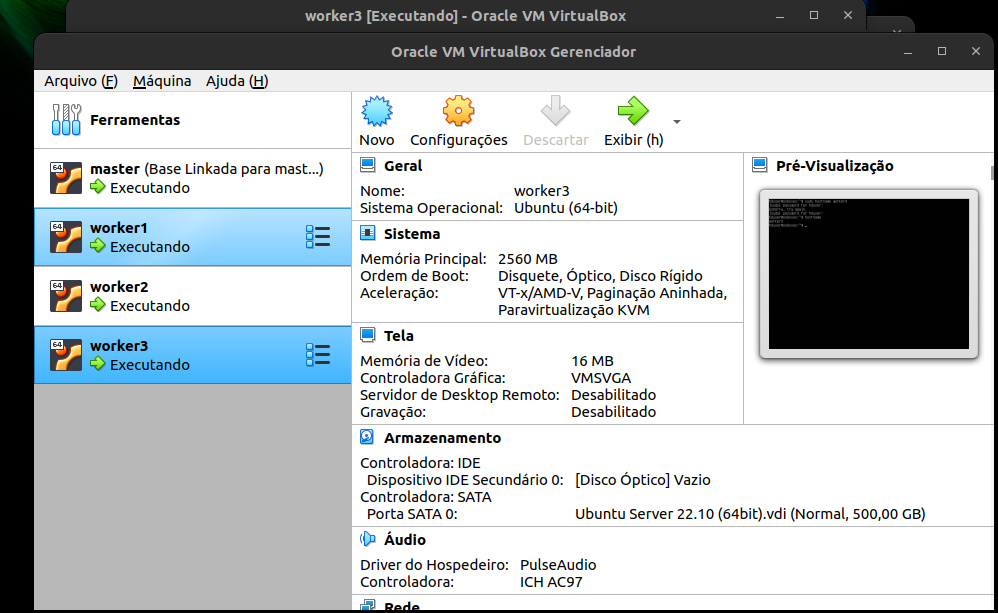
<https://github.com/nilesh-g/hadoop-cluster-install>

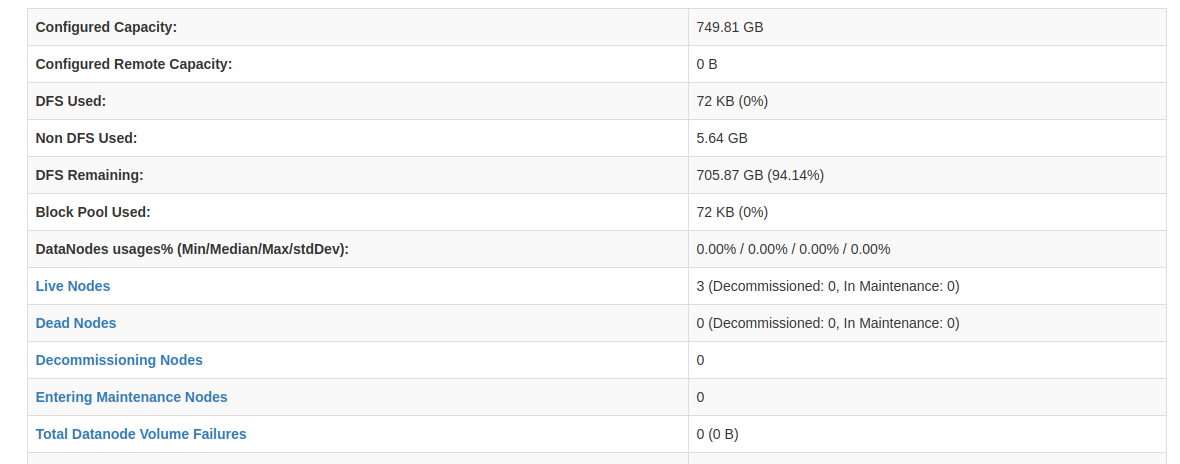
**a) Montar um cluster Hadoop**

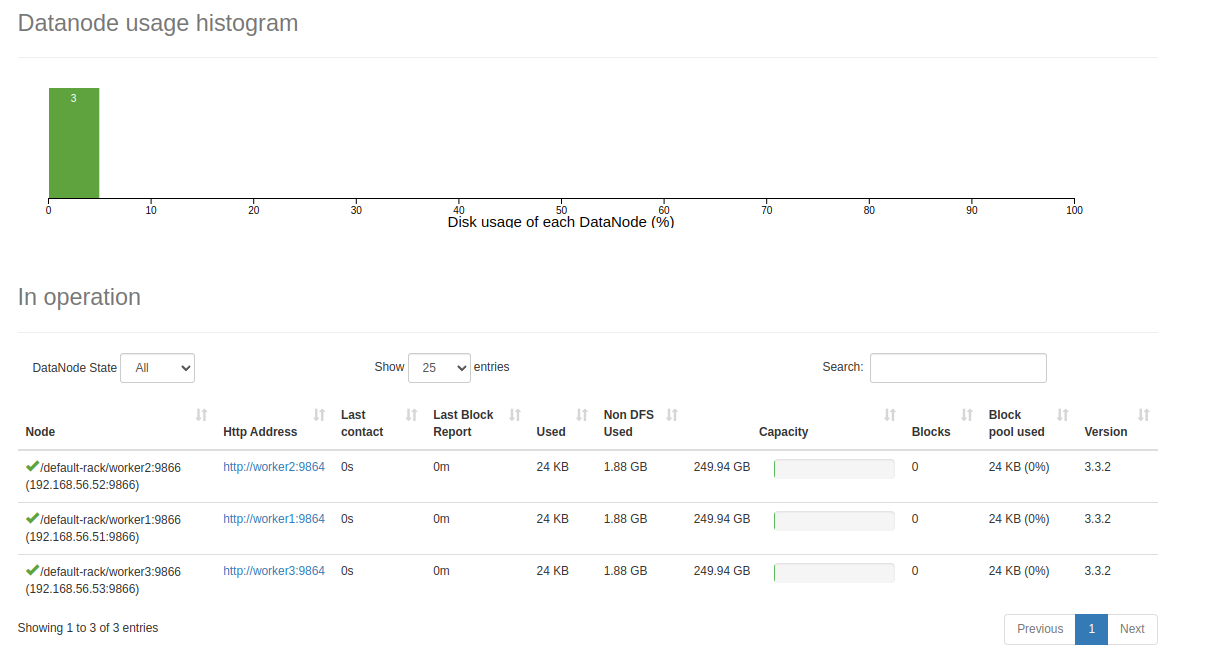
Foi criado um cluster com Hadoop Apache, utilizando o Oracle VirtualBox, composto por 1 master e 3 workers (slaves), a fim de distribuir e processar tarefas de processamento de dados de maneira eficiente e escalável.

As imagens abaixo mostram o cluster Hadoop configurado.







****

**B) Teste do framework Hadoop**

**1. Ajuste as alocações de recursos YARN:**

- Abrimos o arquivo `yarn-site.xml` localizado em `HADOOP\_HOME/etc/hadoop/yarn-site.xml`.

- E fizemos mudanças no `yarn.nodemanager.resource.memory-mb` e `yarn.nodemanager.resource.cpu-vcores`.

**2. Modificamos replicação do HDFS:**

- Abrindo o arquivo `hdfs-site.xml` localizado em `HADOOP\_HOME/etc/hadoop/hdfs-site.xml`.

- Em `dfs.replication` e ajustamos o número de réplicas de blocos HDFS

*```xml*

*<property>*

*<name>dfs.replication</name>*

*<value>2</value>*

*</property>*

*```*

**3. Alterar a localização dos dados do HDFS:**

- Usando o comando `hadoop fs -mv` para mover um diretório ou arquivo HDFS para outra localização. Movemos um diretório chamado `/input` para `/new\_location/input`:

*```*

*$ hadoop fs -mv /input /new\_location/input*

*```*

**4. Modificar o escalonamento de jobs YARN:**

- Abrindo o arquivo de configuração do escalonador, como `capacity-scheduler.xml` ou `fair-scheduler.xml`, localizado em `HADOOP\_HOME/etc/hadoop/capacity-scheduler.xml`

- E fizemos ajustes nas configurações de filas e pools YARN para alterar as capacidades e políticas de escalonamento.

**5. Modificar configurações de JVM:**

- Abrindo o arquivo `hadoop-env.sh` localizado em `HADOOP\_HOME/etc/hadoop/hadoop-env.sh`.

- Ajustamos as configurações da JVM, como a quantidade máxima de memória alocada, definindo a variável `HADOOP\_HEAPSIZE`.

*```shell*

*export HADOOP\_HEAPSIZE=4096*

*```*

**Objetivos e conclusões que a equipe observou com esses testes.:**

**1. Alocação de Recursos YARN:**

Fizemos ajustes nas alocações de recursos no YARN, na quantidade de memória e núcleos disponíveis para containers. Isso foi feito com a intenção de analisar como essas mudanças afetariam o desempenho de nossas aplicações. Observamos que, ao aumentar a alocação de recursos, houve uma melhora perceptível no desempenho das aplicações, com tarefas sendo concluídas de forma mais rápida.

**2.Replicação do HDFS:**

Modificamos a configuração de replicação do HDFS, reduzindo o número de réplicas de blocos. Essa alteração teve como objetivo economizar espaço em disco e avaliar seu impacto no sistema.

**3. Mudança na Localização dos Dados do HDFS:**

Realizamos a reorganização de alguns diretórios no HDFS, movendo dados para diferentes locais. Isso teve o propósito de redistribuir a carga de trabalho no cluster e melhorar o desempenho de tarefas específicas.

**9. Escalonamento de Jobs YARN:**

Efetuamos alterações nas configurações de escalonamento de jobs YARN para priorizar determinados trabalhos sobre outros. Isso visava a atender às necessidades de tarefas críticas mais rapidamente

**10. Configurações de JVM:**

Realizamos ajustes nas configurações da JVM para nossos serviços Hadoop, alocando mais memória para melhorar o desempenho e a estabilidade. Essas modificações resultaram em um processamento de dados mais rápido e evitaram erros de falta de memória nas vms.

**c) Teste do MapReduce no Hadoop**

Utilizamos um programa WordCount feito em Java com a lógica do MapReduce. (Enviado zipado no aprender).

Para testar o código basta seguir os passos:  
Contador de Palavras

1. Compilar o WordCount.java

*make compila*

1. Transformar .jar

*make jar*

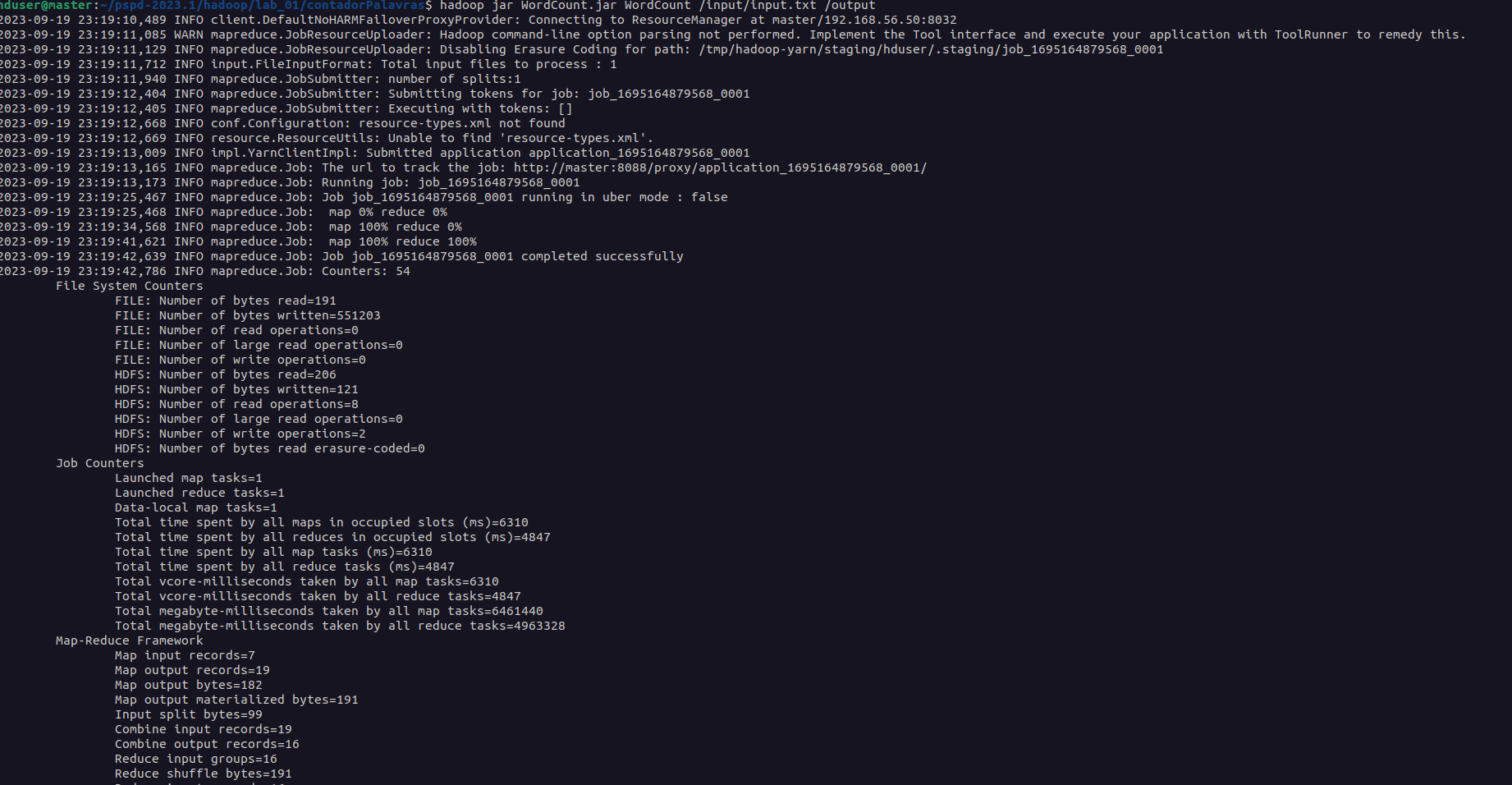
1. Rodar o Hadoop

hadoop jar WordCount.jar WordCount <diretorio da entrada> <diretorio do output>

1. Visualizar saída

hdfs dfs -cat <diretorio do output>/part-r-00000

Abaixo podemos ver a execução do WordCount feito em Java. Nesse teste utilizamos um arquivo input.txt pequeno, apenas para testar o Hadoop.



**Resultados Observados com Monitoramento.**

**Fase de Mapeamento (Map):**

Usando ferramentas de monitoramento de CPU da VM, observamos que os nós Worker estavam usando recursos de CPU intensivamente durante a fase de mapeamento, conforme várias tarefas de mapeamento eram executadas em paralelo.

O uso de memória RAM também aumentou à medida que os mapeadores processavam os blocos de entrada.

As estatísticas de disco mostraram que os arquivos intermediários produzidos pelos mapeadores eram gravados no sistema de arquivos distribuído (HDFS) com alta taxa de transferência.

**Fase de Shuffle/Sort:**

Deduzimos que a partir das estatísticas de disco que os arquivos intermediários criados durante o mapeamento eram lidos e escritos durante esta fase. Essa etapa é intensiva em I/O devido à transferência de dados entre os nós.

**Fase de Redução (Reduce):**

Observamos que o uso de CPU nos Reducers aumentou à medida que eles processavam os dados agrupados e realizavam a redução.

**Monitoramento de Disco e Rede:**

A ferramenta de monitoramento de disco da Oracle VirtualBox confirmaram que o Hadoop gerenciava eficientemente os arquivos intermediários, minimizando a sobrecarga de leitura/gravação.

**D) Teste de tolerância a faltas e escalabilidade da aplicação**

Utilizamos de uma aplicação de leitura de dados que lê um arquivo txt de 2GB. E assim fizemos alguns testes simples.

**Teste de Escalabilidade:**

Iniciamos a aplicação com um número fixo de nós Worker e monitoramos o tempo de resposta.

Gradualmente, aumentamos o número de nós Worker e observamos como o tempo de resposta variou.

Concluímos que o acréscimo de nós aumentou a capacidade de processamento, melhorando o desempenho da aplicação.

**Teste de Tolerância a Falhas - Remoção de Nós:**

Enquanto a aplicação estava em execução, simulamos a remoção de um nó Worker do cluster.

Monitoramos a saúde da aplicação e observamos que o Hadoop automaticamente redistribuiu as tarefas para os nós restantes.

A aplicação manteve seu funcionamento normal, demonstrando a capacidade do Hadoop de lidar com falhas.

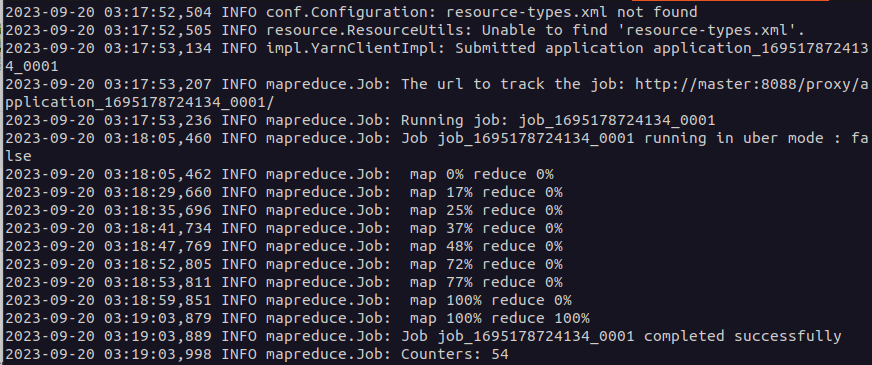
**Testes Futuros (Não Concluídos):**

Devido a limitações de tempo, não conseguimos realizar todos os testes planejados. No entanto, estão planejados testes adicionais que iremos fazer posteriormente.

Abaixo as imagens de alguns testes.

Fazendo com 3 Workers

Tempo total: 1 min e 12 segundos



Fazendo com 2 Workers.

Tempo total: 1 minuto e 42 segundos

