1. **Descreva brevemente a função do HDFS e do MapReduce no ecossistema Hadoop e explique como eles contribuem para o processamento de grandes volumes de dados.**

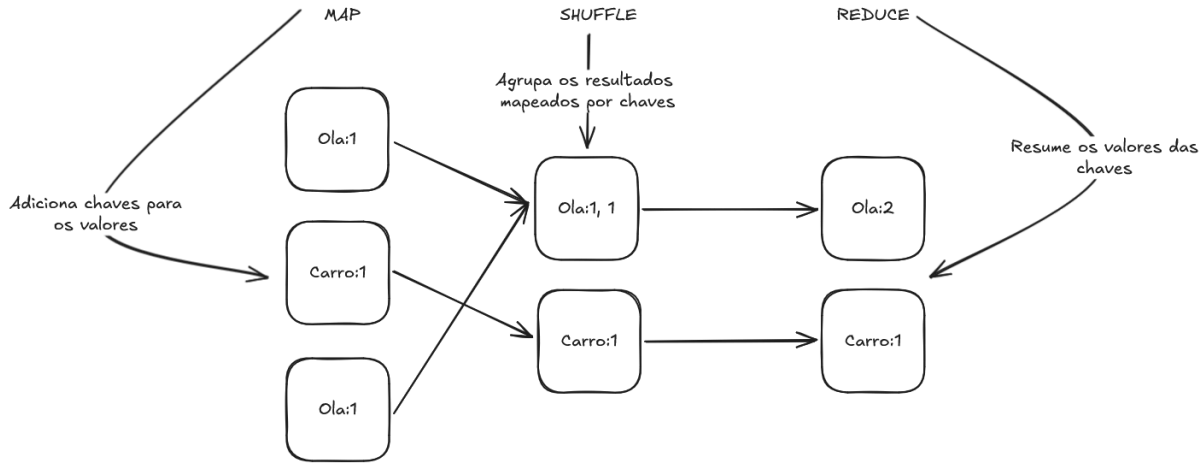
O Hadoop Distributed File System **(HDFS)** é a base do ecossistema Apache, projetado para o Hadoop e com suporte a diversas ferramentas de dados, como o Apache Spark. Sua arquitetura distribui arquivos em blocos de 128MB pelo cluster, permitindo ao Hadoop remontar o arquivo original sob demanda e escalar de forma quase ilimitada, sendo uma pedra fundamental do big data.

A dificuldade de processar conjuntos de dados massivos, na ordem de petabytes, impulsionou a criação do MapReduce. Métodos que exigem acesso completo aos dados, como contagem de palavras e algoritmos de clusterização, tornaram-se inviáveis devido à impossibilidade de carregar tudo em memória. O MapReduce, estruturado em três fases (map, shuffle e reduce), surgiu como solução.

map: A etapa inicial consiste em atribuir chaves aos valores identificados nos dados.

shuffle: Em seguida, os dados são reorganizados, agrupando os valores com base nas chaves definidas na fase de map.

reduce: Por fim, a etapa de reduce consolida todos os dados agrupados em uma única resposta final.



2. **Escolha uma das ferramentas de alto nível (Hive, Apache Kafka, Apache Mahout) e explique como ela simplifica o processamento de dados em Hadoop. Inclua um exemplo de operação que pode ser realizada com essa ferramenta.**

O apache hive, criado pelo facebook e cedido para a fundação apache, é uma ferramenta criada para permitir analistas de dados, que não tem conhecimento em java manipularem dados sem a necessidade de escrever um MapReduce do zero. O Hive se baseia em um dialeto semelhante ao **SQL**, o **HiveQL**. Um exemplo de operação que é frequentemente realizada pelo HiveQL é a manipulação de dados para criar um novo dataset.

Criando uma tabela com o Hive:

Create_Hive

```
CREATE TABLE sales (
  id INT,
  product STRING,
  price FLOAT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

Carregando um csv na tabela:

Load_Sales

```
LOAD DATA INPATH '/dados/sales.csv' INTO TABLE sales;
```

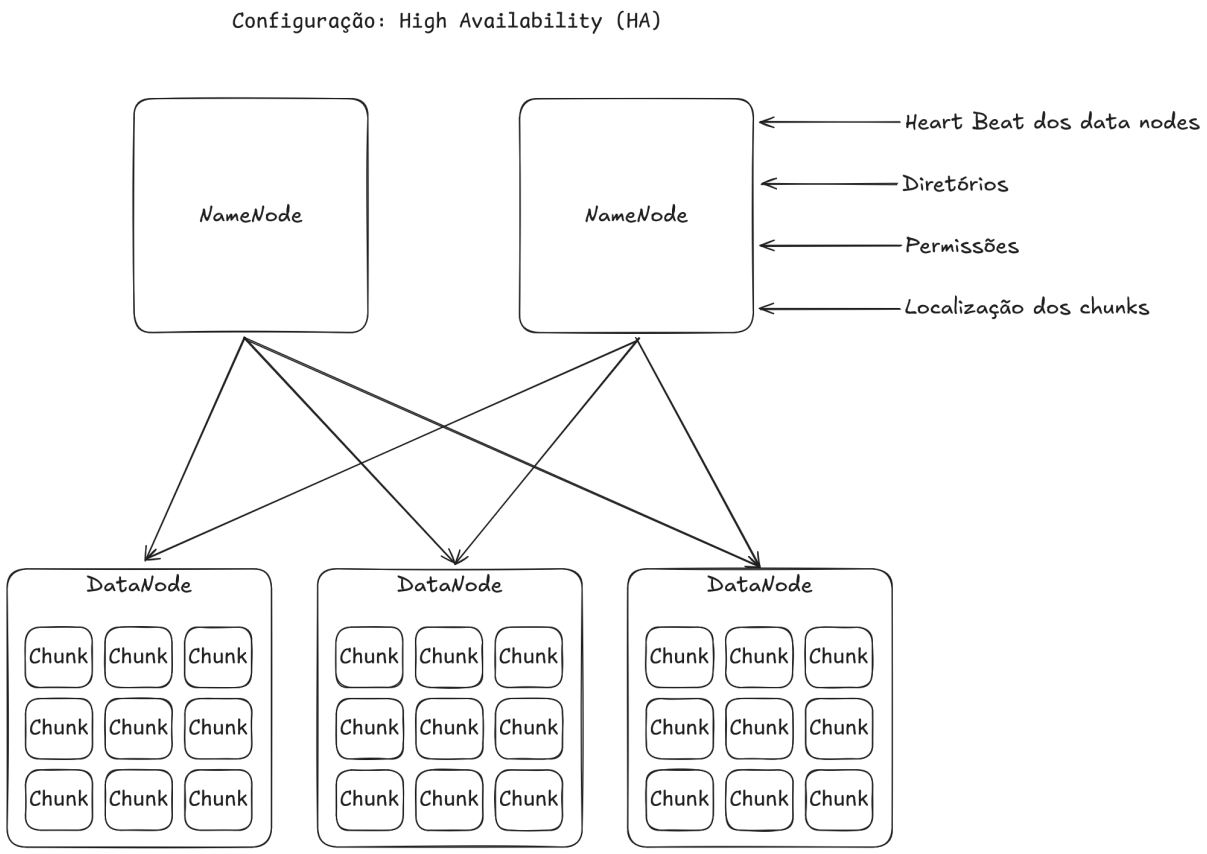
Obtendo o faturamento por produto de em um HDFS:

Select_Sales

```
SELECT product, SUM(price) FROM sales GROUP BY product;
```

3. **Explique o papel do NameNode e do DataNode dentro do Hadoop Distributed File System (HDFS). Como a arquitetura do HDFS contribui para a alta disponibilidade e escalabilidade no processamento de grandes volumes de dados? Justifique.**

   A arquitetura do hadoop foi criada com a intenção de ser extremamente resiliente, e por isso, foi tomada a decisão de separar armazenamento de dados e gerenciamento do banco em nodes separados. O nameNode foi criado com o intuito de armazenar metadados do banco, como em qual dataNode o chunk de um arquivo específico está armazenado,monitorar heartbeats de dataNodes, permissões de acesso e diretórios, para manter um cluster sempre funcionando, geralmente existem dois, ou mais nameNodes, dependendo da configuração escolhida. Os dataNodes são feitos para armazenar os chunks, geralmente os chunks são replicados em múltiplos dataNodes, que são usados para redundância de dados. O hadoop é extremamente resiliente a falhas, devido a sua redundância e altamente escalável, devido a sua arquitetura distribuída, que o fizeram um marco histórico, marcando o início do big data.



4. **Escolha um serviço de nuvem que suporte Hadoop (Amazon EMR, Google Cloud Dataproc, Microsoft Azure HDInsight) e descreva como ele pode ser utilizado para escalonar o processamento de big data.**

   O Amazon EMR (Elastic MapReduce) é um serviço da AWS que simplifica a execução de frameworks de big data como Hadoop e Spark em clusters na nuvem. Ele permite escalonar tanto o armazenamento quanto o processamento automaticamente.

O EMR possui três camadas:

Camada de Armazenamento: Aqui podemos ter uma S3, ou um HDFS

Gerenciamento de Recursos: YARN, sendo responsável por alocar CPU e RAM, além de lidar com falhas

Frameworks: Aqui temos os responsáveis por armazenar e manipular dados refinados, como: Hadoop, Spark, Hbase, etc.

5. **Descreva o fluxo de dados em um job MapReduce, do início ao fim. O que acontece durante as fases de splitting e shuffle?**

Nota: *Quase tudo foi respondido na primeira pergunta, então vou tentar me aprofundar ainda mais nos tópicos.*

Entrada:
A entrada consiste em um grande arquivo, ou um conjunto de grandes arquivos que estão armazenados em um HDFS ou outra forma de armazenamento distribuído, como uma S3

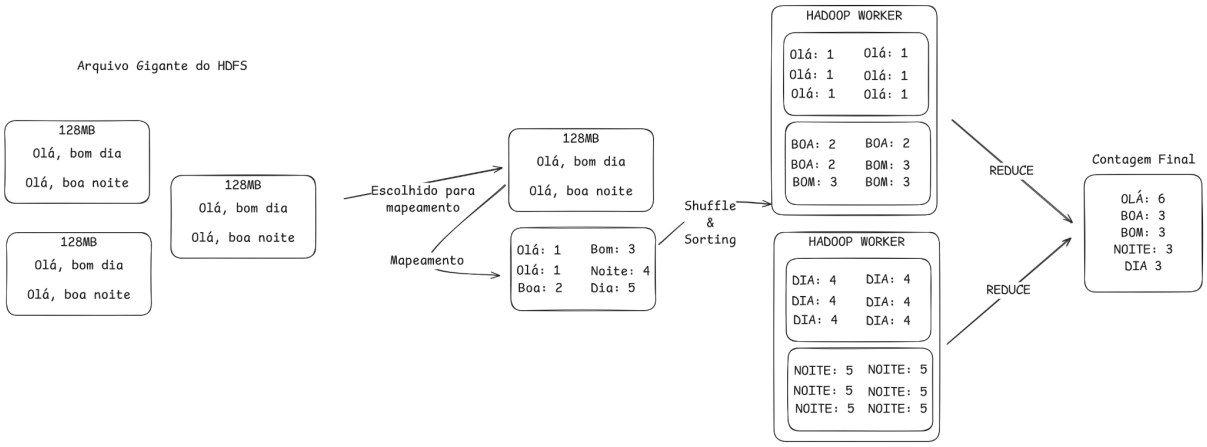
Splitting:
Essa é uma tarefa interna do Hadoop, se o arquivo for maior que um chunk, o arquivo será repartido, geralmente alinhadas com os blocos do HDFS e haverá um mapeamento para cada bloco.

Mapping:
O mapeamento irá transformar cada bloco do splitting em um conjunto de registro dentro de uma estrutura de chave e valor.

Shuffle:
No Shuffle, os dados irão ser redistribuídos pelo cluster, onde cada node terá um conjunto de chaves únicas em todo o cluster.

Sort:
Um algoritmo de sorteio irá ordenar os conjuntos de forma sequencial de acordo com o valor da chave.

Reducer:
As chaves serão agregadas de forma a produzir o resultado final.



6. **Como o Hive executa uma consulta SQL-like internamente?**

   Parser:

   Aqui Tudo é transformado em uma abstract sintax tree (AST) para futuramente ser analisado pelo hive, todas as palavras são transformadas em tokens, para futuramente serem lidas por uma análise semântica

   Análise Semântica:

   Depois, a sintaxe da query é testada, vemos se sua estrutura lógica está correta, se as funções que são chamadas são válidas entre outras validações.

   Otimização do Plano Lógico:

   Para garantir a execução sólida das queries, a otimização é garantida, removendo expressões booleanas irrelevantes, ordenando joints e realizando outros processos para garantir que as queries estão altamente otimizadas.

   Geração do Plano Físico:

   O plano lógico otimizado é traduzido para um conjunto de tarefas executáveis. Tradicionalmente, essas tarefas eram mapeadas para Jobs MapReduce, ou tarefas de manipulação de RDD via spark.

   Execução no Cluster:

   O plano físico é convertido para um DAG (grafo acíclico direcionado) de tarefas. Esse DAG é distribuído e executado no cluster Hadoop.

7. **Quando importando dados de um banco de dados relacional para o HDFS usando Sqoop, quais são as considerações importantes para garantir uma importação eficiente?**

   Número de mappers: O Sqoop inicia com 4 mappers por padrão. É possível acelerar a importação de dados, aumentando esse número. No entanto, certifique-se de que o banco de dados consiga suportar essa quantidade de conexões simultâneas para evitar sobrecarga do servidor ou falhas de conexão.

   Particionamento de dados: Para melhorar a importação de dados, é recomendado usar uma coluna indexada pelo banco e bem distribuída para dividir os dados, evitando assim a sobrecarga de alguns mappers. Além disso, para prevenir a leitura completa da tabela, use a opção `--split-by` com filtros SQL na cláusula `--where` ou utilize a opção `--query` para importar apenas os dados necessários.

   Evitando arquvivos pequenos:

   O número de arquivos em HDFS é igual ao número de mappers; muitos mappers geram muitos arquivos pequenos, que poluem o NameNode. Para resolver isso, ou use um numero menor de mappers, ou faça um merge no final da operação.

Habilite o modo direct:
habilite `--direct` em bancos MySQL, PostgreSQL e outros suportados para bypass do connector genérico, acelerando importações até em 2 vezes.

8. **Descreva como o Apache Kafka pode ser utilizado para construir uma arquitetura de processamento de dados em tempo real.**

   O apache kafka, assim como o rabbitMQ, são filas distribuídas, as filas são amplamente usadas para a comunicação entre sistemas distribuídos, podendo ser usadas para disponibilizar conteúdo produzido entre consumidores, ou consumir o que foi publicado. O Kafka é fundamental para uma arquitetura de processamento em tempo real, pois é usado para enviar o que foi processado diretamente para os consumidores de dados, seguindo o diagrama abaixo: