



PUC Minas

MapReduce

Diego Roberto Gonçalves de Pontes

O que é MapReduce?

- MapReduce é uma estrutura de execução distribuída baseada em Java que faz parte do ecossistema Apache Hadoop, simplificando a programação distribuída com duas etapas de processamento: Map e Reduce. Na etapa Map, os dados são divididos para processamento paralelo, e na etapa Reduce, os dados são agregados; normalmente, o Hadoop Distributed File System (HDFS) é usado para entrada e saída.

História do MapReduce

- MapReduce foi desenvolvido pelo Google em 2004 por Jeffery Dean e Sanjay Ghemawat. No artigo “MapReduce: Simplified Data Processing on Large Clusters”, eles explicam que o MapReduce é inspirado nas funções Map e Reduce da programação funcional. Originalmente, o sistema MapReduce do Google operava com o Google File System (GFS). Em 2014, o Google já não usava mais o MapReduce como principal modelo de processamento de big data. Inicialmente, o MapReduce era a principal forma de acessar dados no HDFS. Atualmente, existem outras tecnologias, como Hive e Pig, que permitem recuperar dados do HDFS utilizando comandos semelhantes ao SQL, além dos jobs tradicionais de MapReduce.

Como Funciona?

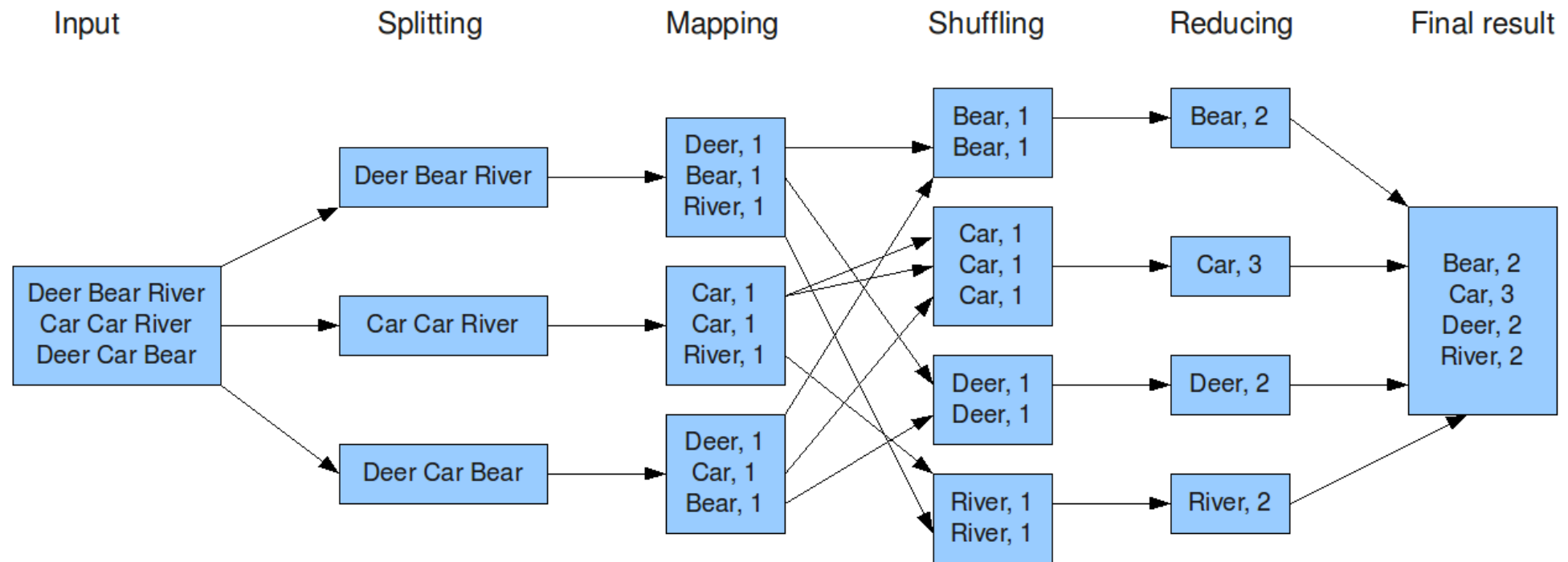
- Map: Na etapa Map, os dados de entrada são divididos em blocos menores e atribuídos a mappers para processamento paralelo, com cada nó aplicando a função Map aos seus dados locais e armazenando a saída temporariamente. O nó principal (master) assegura que apenas uma cópia dos dados redundantes seja processada.
- Shuffle, Combine e Partition: Na fase Shuffle, os nós do worker redistribuem os dados com base nas chaves geradas pela função Map, garantindo que todos os dados com a mesma chave sejam enviados ao mesmo nó. O processo Combine, opcional, pode ser executado para reduzir o volume de dados antes do embaralhamento, enquanto a Partition determina como os dados são organizados e atribuídos aos reducers.

Como Funciona?

- Reduce:
 - Os Reducers só começam a processar após a conclusão de todos os Mappers, agrupando e agregando pares de saída <chave, valor> com a mesma chave. Embora a etapa Map seja obrigatória para filtrar e classificar dados, a etapa Reduce é opcional e utilizada conforme necessário.

MapReduce

The overall MapReduce word count process



Considerações

- Leitura/Gravação Intensiva
 - Os jobs MapReduce não utilizam estruturas de memória distribuída, resultando em pouca retenção de dados na memória e exigindo leitura e gravação frequentes no HDFS. Jobs MapReduce mais complexos são compostos por múltiplos jobs menores encadeados, com dados compartilhados via HDFS, criando um gargalo de processamento.

Considerações

- Foco em Java
 - MapReduce é otimizado para Java, tornando a escrita de aplicativos MapReduce mais eficiente nessa linguagem, o que requer compilação e implantação em um cluster Hadoop, um processo não comum para analistas de dados que preferem SQL ou linguagens interpretadas como Python. Embora seja possível usar outras linguagens, como C ou Python, isso adiciona sobrecarga devido à necessidade de iniciar processos do sistema, afetando o desempenho do job.

Motivações

- Escalabilidade:
 - Processar petabytes de dados distribuindo a carga de trabalho entre milhares de máquinas.
 - Escala horizontal: Adicionar mais máquinas para aumentar a capacidade de processamento.
- Tolerância a Falhas
 - Reatribuição automática de tarefas falhadas a outras máquinas no cluster.
 - Garantia de continuidade no processamento sem interrupções significativas.

Motivações

- Flexibilidade
 - Aplicação do modelo MapReduce a uma ampla variedade de problemas de processamento de dados.
 - Desde indexação de páginas web até análise de grandes volumes de logs e dados de redes sociais.



PUC Minas