



Curso 3: Administração de Dados Complexos em Larga Escala

-- Apache Hive --

Prof. Jose Fernando Rodrigues Junior

Objetivo: apresentar a solução de data warehouse Hive

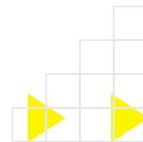




O que é exatamente?

“**Data Warehouse** é uma coleção de dados orientados por assunto, integrada, não-volátil, variante no tempo, que dá apoio às decisões de administração” (W.H. Inmon, 1992).

- **Orientados a transações consolidadas:** vendas, operações bancárias, acessos à informação...

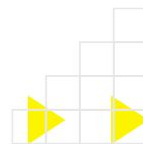




Apache Hive - Big Data Warehouse



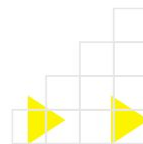
- Em Big Data, Data Warehousing pode ser feito sobre o sistema Apache **Hive**
 - Projeto originário do Facebook ⇒ [História](#);
 - **Custo bem menor** do que soluções comerciais, como o Oracle Exadata ou o IBM Netezza; funciona com computadores desktop, não necessitando de soluções *enterprise*;
 - Usa uma variação do SQL chamada **HiveQL**, cujo interpretador de consultas **compila Jobs MapReduce**;
 - **Modelo de dados robusto**: tabelas, rows, colunas, partições, arrays associativos, listas, e estruturas;
 - **Metastore**: dicionário de dados Hive.





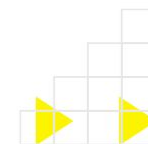
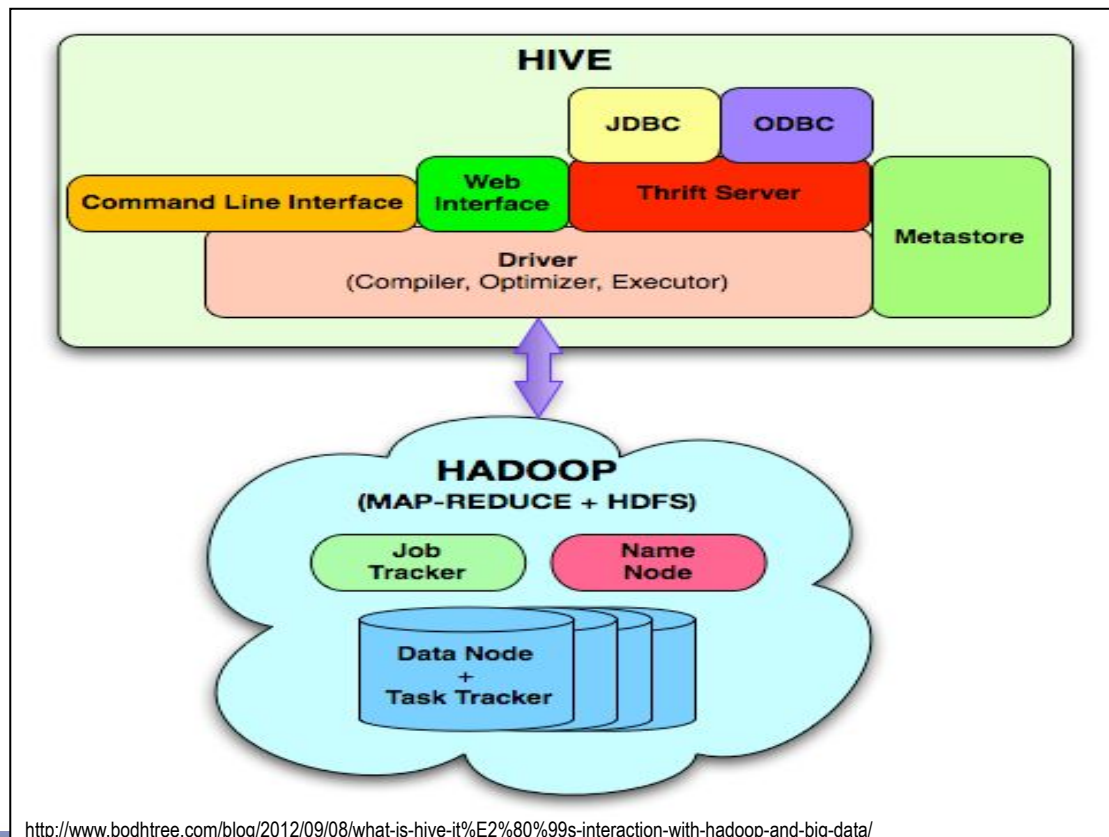
Apache Software Foundation

- Organização **sem fins lucrativos** criada para suportar projetos de **código aberto**;
- Maior projeto: servidor web **Apache HTTP Server**;
- Comunidade **descentralizada de desenvolvedores** colaborativos;
- *Open source* software sob a **licença Apache**;
- O **maior provedor** de software livre do mundo.





Visão geral da arquitetura Apache Hive





Visão geral da arquitetura Apache Hive

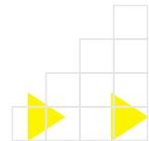
- **Em resumo:** o Apache Hive é um sistema que recebe comandos SQL e os executa sobre a infraestrutura Apache Hadoop;
- Mas, o que é o **Apache Hadoop**?
Trata-se de um arcabouço que executa processamento **MapReduce** sobre um sistema de armazenamento **HDFS**;
- **E porque isso é relevante?**
O Hadoop abstrai processamento paralelo distribuído escalável (MapReduce), sendo capaz de realizar operações sobre enormes conjuntos de dados (petabytes) de modo eficiente.

HDFS

(Hadoop Distributed File System)




- 📌 **Escalabilidade sobre clusters** com centenas e até milhares de nós computacionais abstraídos como um único sistema de arquivos;
- 📌 Dados quebrados em **blocos** (default de 128 MB) e distribuídos no cluster;
- 📌 **Abstração da distribuição física** dos dados;
- 📌 **Redundância** *default* de 3 cópias (tolerância a falhas);
- 📌 **Escalabilidade** facilitada (*on the fly*);
- 📌 **Qualquer máquina** pode ser usada - seu desktop, por exemplo;
- 📌 Possui um nó especial para gerenciamento: o ***namenode***, “onde está o que”;
- 📌 **Baixo custo!**






MapReduce

 **MapReduce:** um modelo de processamento que divide (Map) a tarefa (job) de processamento ao mesmo tempo em que prevê uma maneira de integrar os resultados (Reduce); originário da empresa Google, 2003;

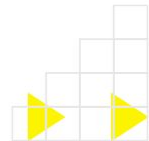
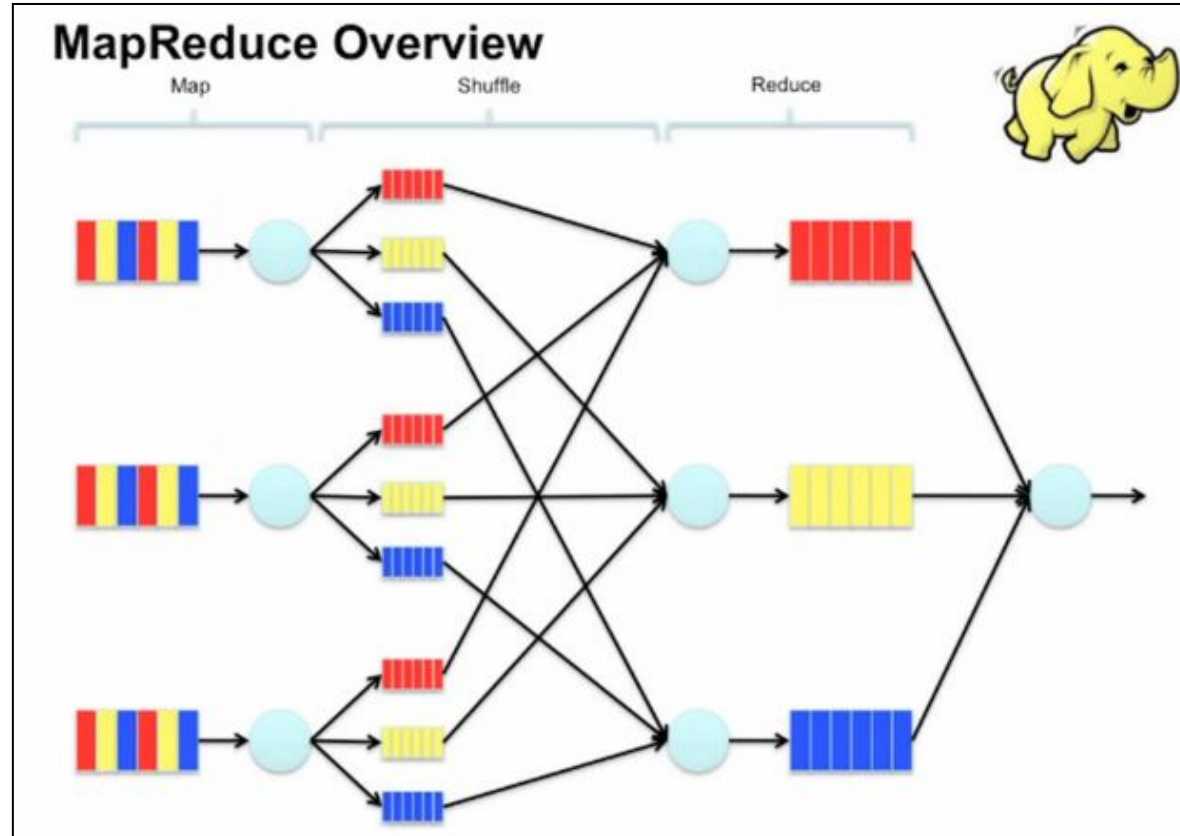
 **Abstração/simplificação de processamento distribuído** com pronta escalabilidade;

 **MapReduce é um modelo de processamento, não um software;**

 Nem todo tipo de processamento pode ser feito, **apenas os que podem ser divididos;**



Map-Shuffle-Reduce



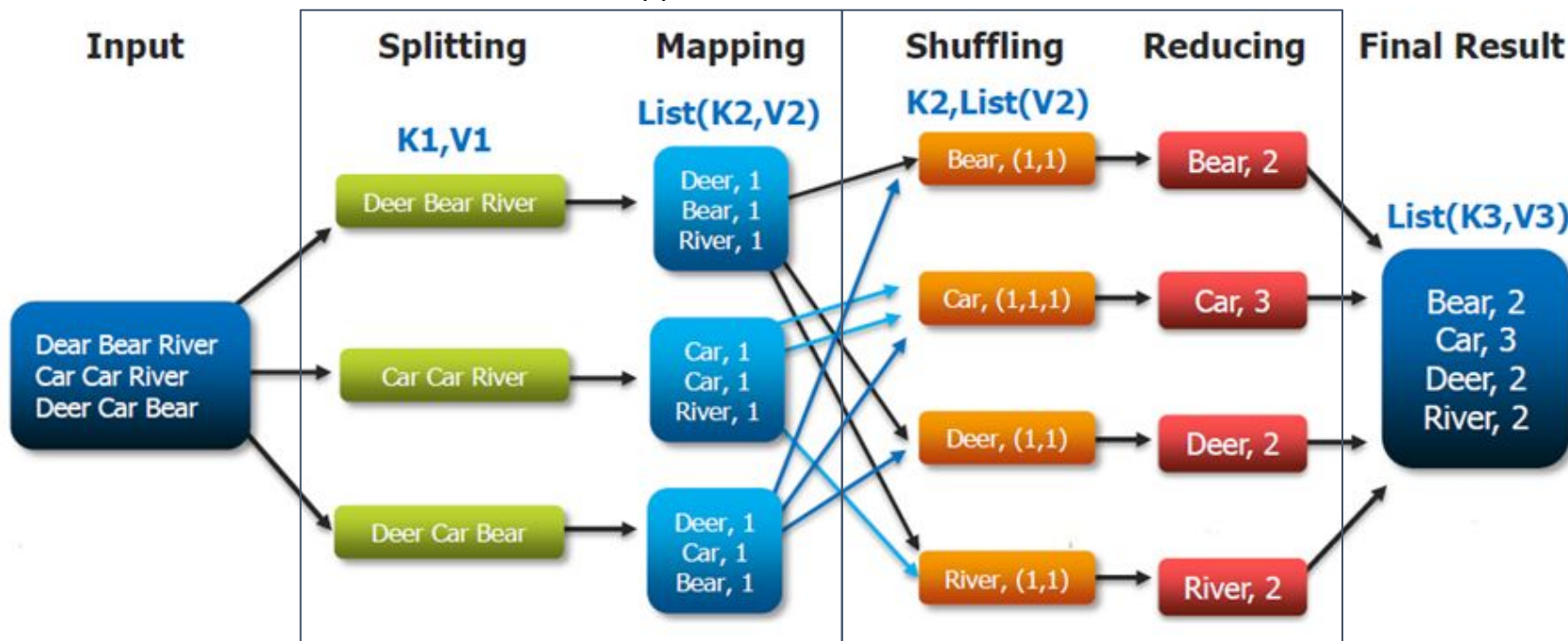
MapReduce



The Overall MapReduce Word Count Process

3 nós mapper

4 nós reducer





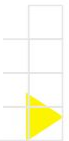
MapReduce

```
map(key, value):
```

```
// key: document name; value: text of the document  
  for each word w in value:  
    emit(w, 1)
```

```
reduce(key, values):
```

```
// key: a word; value: an iterator over counts  
  result = 0  
  for each count v in values:  
    result += v  
  emit(key, result)
```





MapReduce

map(^{k1,}key, ^{v1}value):

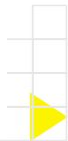
```
// key: document name; value: text of the document
for each word w in value:
    emit(w, 1)
```

^{k2,} ^{v2} \Rightarrow cada nó mapper produz uma lista de pares list(k2,v2)

após o shuffle (hash), cada nó reducer produz (k2, list(v2)), equivalente a um group by k2, input to reduce

reduce(key, values):

```
// key: a word; value: an iterator over counts
result = 0
for each count v in values:
    result += v
emit(key, result)
```





MapReduce-Hive

- O modelo MapReduce **suporta diversas operações computacionais**; dentre elas, várias operações SQL;
 - ⇒ [SQL to mapreduce](#) ⇒ Altamente complexo!
 - ⇒ Para saber mais: [Mining of Massive Datasets](#), cap. 2
- Para simplificar, é possível pensar em uma **camada de software que traduz SQL em processamento MapReduce**;
- Com efeito, o Apache Hive tem **duas funções** principais:
 - 1) Carregar dados no sistema HDFS;
 - 2) Traduzir SQL em operações MapReduce e executá-las em Apache Hadoop.

