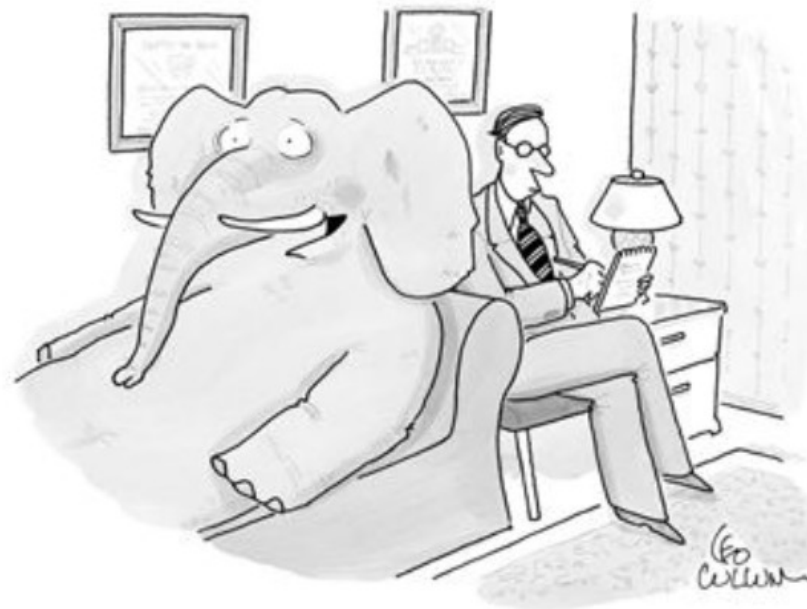


# Map Reduce



*"I'm right there in the room, and no-one even acknowledges me."*

# O que é

• *Processor computação de forma distribuída!*

Sistemas: Mecanismos  
Computação: Algoritmos

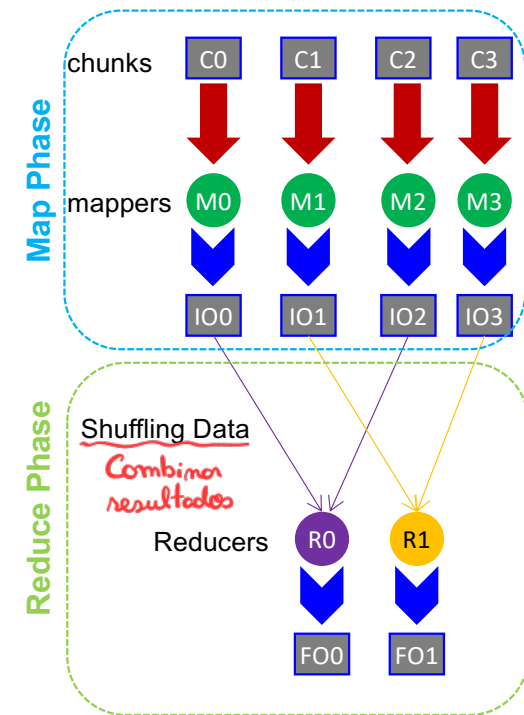
- Um paradigma de programação que permite processar grandes quantidades de informação de forma paralela.
- Divide-se em duas etapas:
  - “Map” responsável por seleccionar/ordenar/processar informação de forma independente/isolada. Resultado é um conjunto de pares ordenados “key-value”
  - “Reduce” responsável por combinar/agregar informação da etapa anterior num conjunto de dados muito reduzido (no limite 1)

*Cria o domínio de soluções*



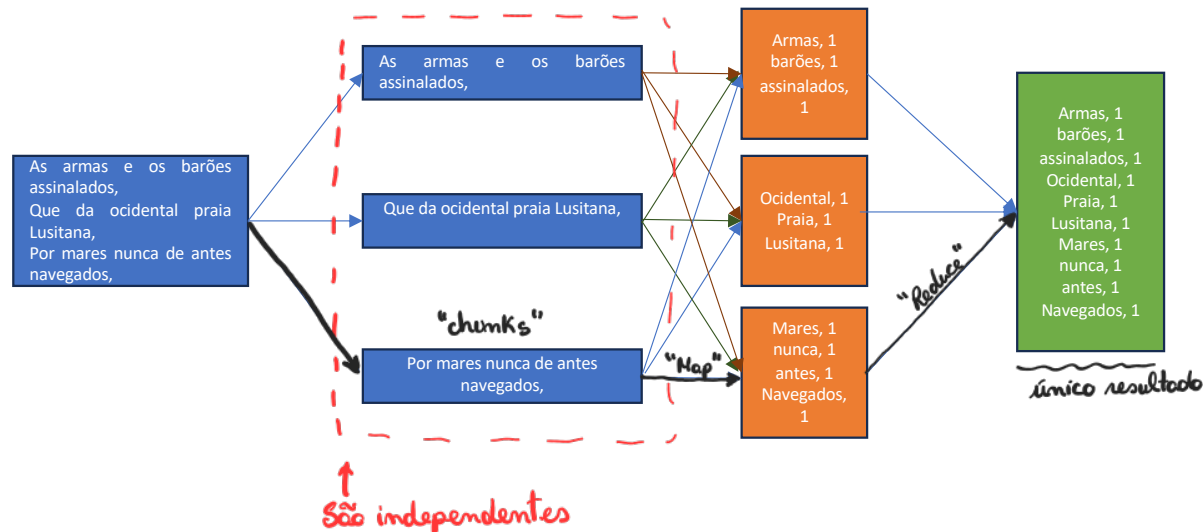
# Algoritmo

- Pedacos de informação ("chunks") são processados de forma isolada por "Mappers" que podem ser distribuidos numa rede pois tratam a **informação de forma isolada**.
- O resultados do trabalho dos "Mappers" é um sub-produto intermédio que é fornecido aos processos "Reducers" num processo denominado por "shuffling process".
- O resultado dos "Reducers" é também o resultado final do algoritmo



# Um exemplo

- Queremos fazer um histograma de palavras nos lusiadas.
  - Começamos por criar chunks – para simplificar cada chunk é uma estrofe
  - **Map**: Para cada estrofe vou obter uma lista de k,v com cada palavra e número de ocorrências.
  - **Reduce**: com os resultados anteriores vou somando e agregando
  - No final obtenho o histograma pretendido



# Vantagens

- Processamento Paralelo

- Cada tarefa é completamente independente, partimos o problema para o simplificar (“divide & conquer”)

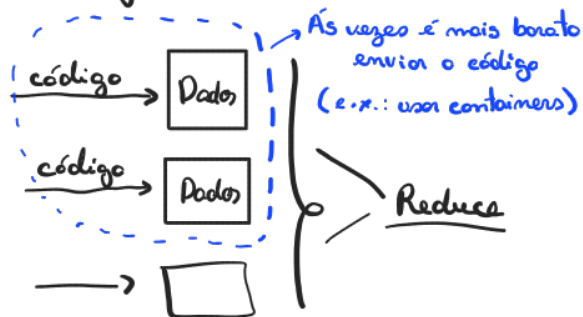
→ Paralelizável

- Localidade da Informação

- Os dados não estão centralizados, mas sim distribuídos por todos os nós de computação. ⇒ *Menor distribuição*

- Em vez de transmitir os dados entre nós, as funções “Map” e “Reduce” é que migram para a localização dos dados.

↳ *Menor código!*



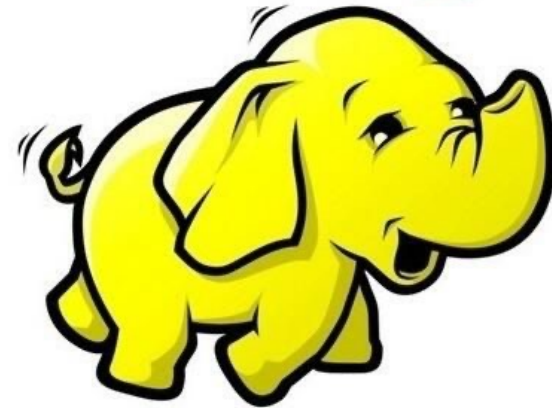
# Hadoop

Distribuído pelos nós  
da rede!

- Hadoop é uma implementação Open Source do MapReduce
- Hadoop é um motor de análise assente num sistema de ficheiros distribuído de seu nome HDFS (Hadoop File System) *↳ Garante que tudo está partilhado*
- HDFS é uma “cópia” do Google File System (GFS)

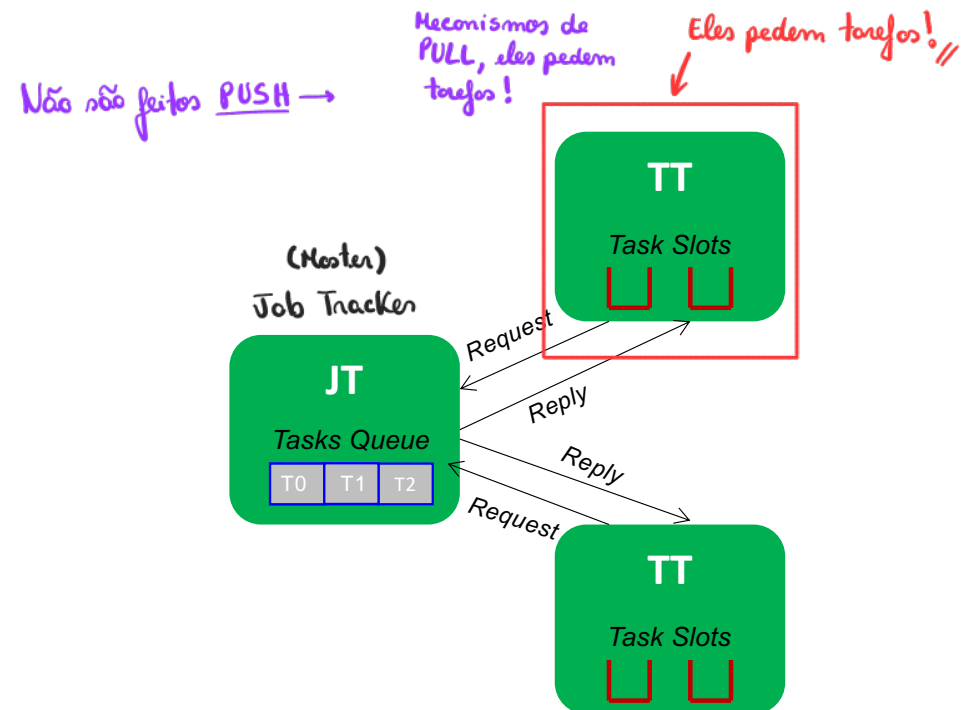
**hadoop**

Open Source



# Arquitectura de Software

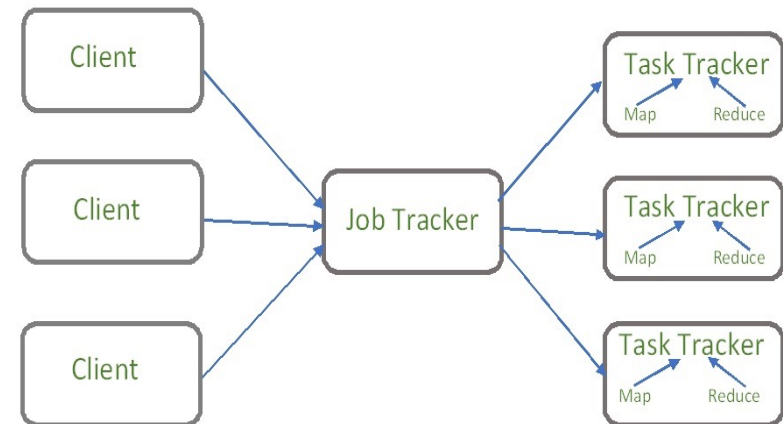
- Arquitectura *master-slave*
- *Master* tem o nome de **Job Tracker (JT)**
- *Slave* tem o nome de **Task Tracker (TT)**
- Estratégia de comunicação **é pull scheduling**
  - Não é o master quem atribui trabalho, mas sim os slaves que pedem tarefas



## Task Scheduling *(pull scheduling)*

Cada Task Tracker envia um heartbeat periodicamente ao Job Tracker com um pedido para realizar trabalho

- Job Tracker satisfaz o pedido atribuindo tarefas de mapping aos nós que tem informação por processar
- Job Tracker pode também atribuir tarefas de reduce independentemente da localização da informação



Hadoop 1.0 architecture



# Sumário

- O seu modelo de funcionamento simples permite ao utilizador escrever e testar rapidamente sistemas distribuídos
- Distribui de forma automática e eficiente carga de processamento entre máquinas *// → código em java !*
- Apresenta uma curva de escalabilidade quase plana (10 nós ou 1000 nós com o mesmo esforço)

*↳ Rede full Mesh ...  
hadoop escala automático !*

# Comparação com Modelos Tradicionais

Aspecto	Memória Partilhada	Mensagens	MapReduce
Comunicação	Implícita (via <u>load/store</u> ) <i>registos</i>	Explícita	<u>Limitada e Implícita</u> → <i>escrevo no filesystem...</i>
Sincronização	Explícita	Implícita (via mensagens)	Imutável (K, V)
Suporte em Hardware	Necessário	Nenhum	Nenhum
Esforço de Desenvolvimento	<u>Baixo</u> <i>Fácil nos é core</i>	<u>Alto</u> <i>Debug é chato...</i>	<u>Baixo</u> <i>nos são iguais...</i>
Esforço de Aprfeiçãoamento	Alto	Baixo	Baixo