



Instituto Nokia de Tecnologia  
Wellington Ramos Chevreuil

# Agenda

- **Big Data**
  - **Surgimento e Ascensão**
  - **Cenários de BigData**
  - **Trabalhando com BigData**
- **Hadoop**
  - **Introdução**
  - **Detalhes**
  - **Componentes Principais**
  - **Persistência**
  - **HDFS Design**
  - **MapReduce**
- **Ecossistema Hadoop**

# Big Data

## Surgimento e Ascensão

- **Popularização da Internet**
- **Novos serviços oferecidos**
  - **Buscadores de conteúdo**
  - **Redes Sociais**
  - **Comércio eletrônico (cauda longa)**
- **Evolução da capacidade de armazenamento**
  - **Menores Custos**
    - **Toda atividade do usuário é persistida**
  - **Resultado: Centenas de TeraBytes, PetaBytes ou mesmo ExaBytes!!**

# Cenários de BigData

- **Recomendações**
  - **Produtos e itens para compra**
  - **Novos amigos em redes sociais**
  - **Novas músicas para serem ouvidas**
  - **Novos investimentos**
- **Publicidade direcionada**
- **Classificação de conteúdos**
- **Busca de páginas**
- **Localização**
- **Compras Coletiva**
- **Exemplos de casos de BigData**
  - **Google, Yahoo, Facebook, Linkedin, Twitter, Nokia, Navteq, Groupon, AOL Advertising, IBM, JP Morgan Chase, New York Times**

# Trabalhando com BigData

- **Hardware padrão**
  - **Max RAM: 64GB**
  - **Max Disco: 24TB**
- **Processamento médio do Google/mês: 400PB (2007)**
  - **Média de processamento: 180GB**
  - **Tempo processamento 180GB: 45 minutos**
- **Leitura/Escrita é muito lenta**
- **Solução**
  - **Leituras paralelas**
    - **1 HD = 75MB/segundo**
    - **1.000 HDs = 75 GB/segundo**
- **Grids?**
  - **Alto número de leituras/escritas de/para um único dispositivo não é eficiente**
- **Implantando um *cluster* de máquinas: Hadoop**

# Hadoop - Introdução

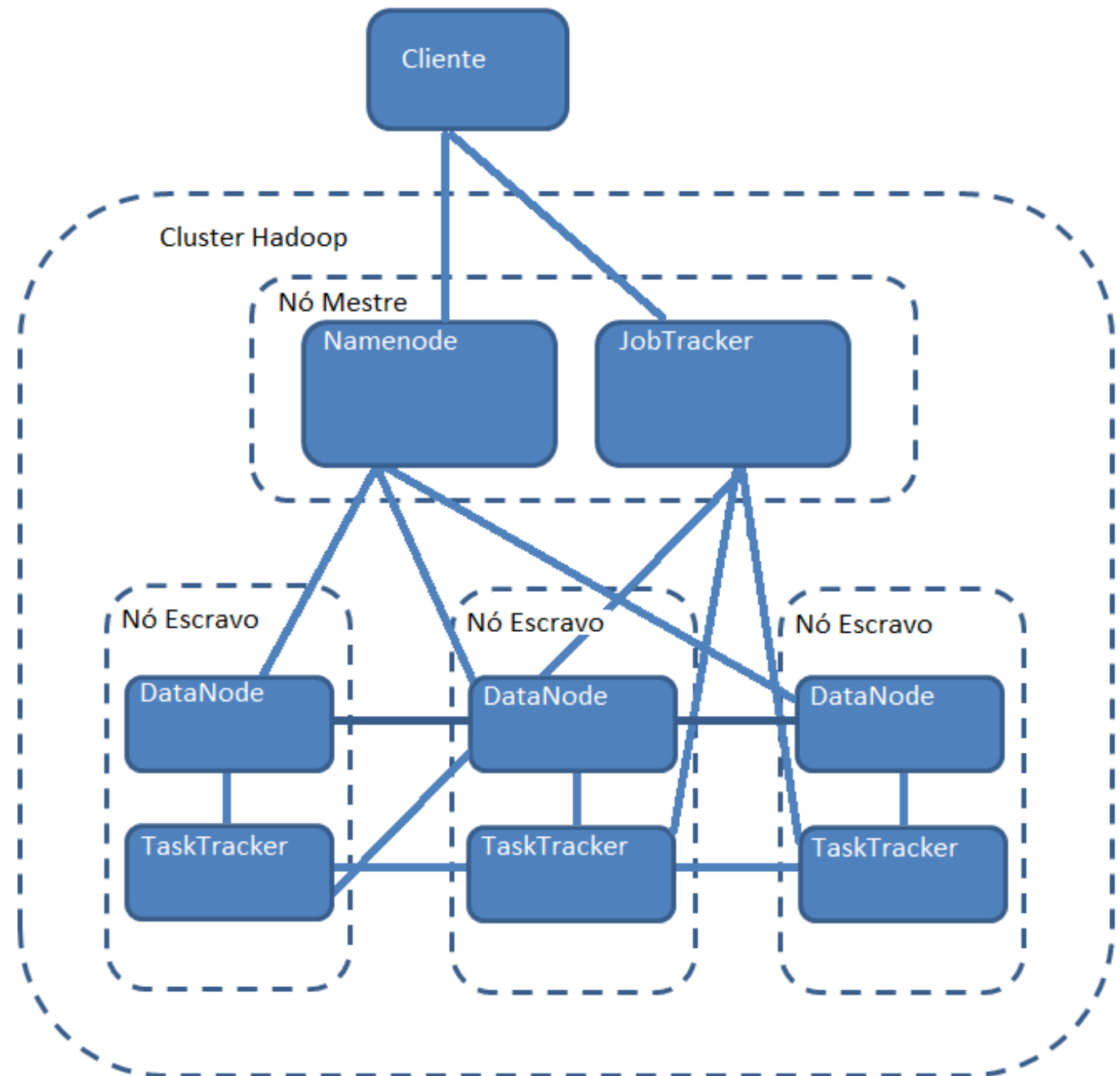
- **Sistema de Persistência e Processamento distribuído**
- **Google GFS**
- **Google MapReduce**
- **Apache Software Foundation**
- **Java**
- **Linux**
- **Hardware comódite**
  - **Relativamente barato**
  - **Baixa confiabilidade**
- **Requisitos**
  - **Suporte a falha parcial**
  - **Suporte a recuperação de dados**
  - **Suporte a recuperação individuais de nós**
  - **Consistência**
  - **Escalável**

# Hadoop em detalhes

- **Nós independentes entre si (shared nothing)**
- **Tolerância a falhas**
  - **Altas taxas de falha (HW commodity)**
  - **Falha de um nó, blocos são replicados para outro nó**
  - **Falha de uma tarefa, uma nova tentativa de execução é realizada (pelo mesmo nó ou por outro)**
- **HDFS = Namenode + SecondaryNamenode + Datanodes**
  - **Vários nós de dados (datanodes)**
  - **Arquivos divididos em blocos distribuídos através do cluster (tamanho padrão: 64 MB)**
  - **Replicação de blocos (padrão: 3 réplicas)**
  - **Poucos arquivos grandes**
  - **Não provê acesso randômico**

# Componentes Principais

- **HDFS (Hadoop File System)**
  - **Namenode**
  - **SecondaryNamenode**
  - **DataNodes**
- **MapReduce**
  - **JobTracker**
  - **TaskTrackers**

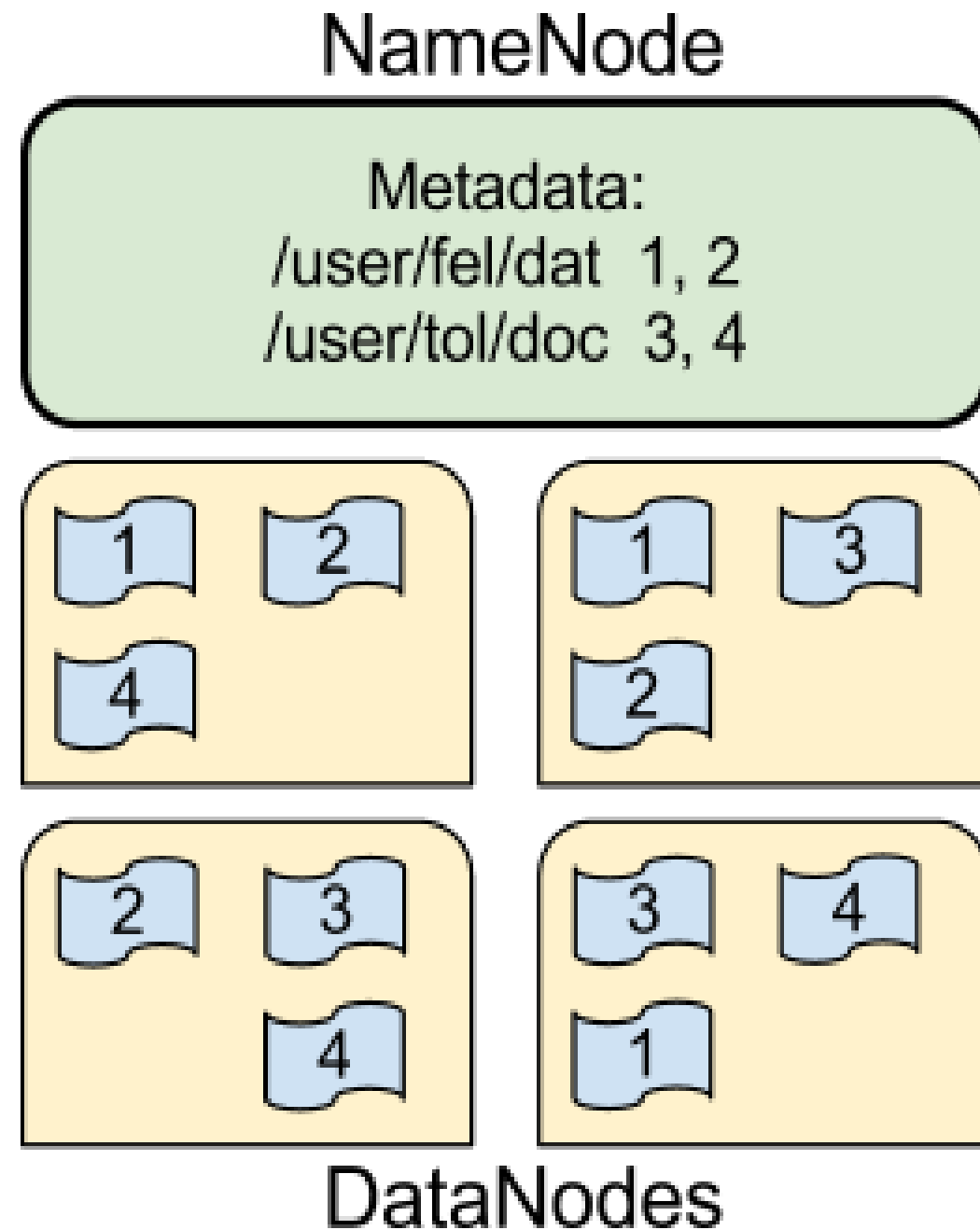




# Persistência com Hadoop

## HDFS

- **Hadoop File System**
  - Arquivos divididos em blocos
  - Blocos replicados ao longo do Cluster
  - *Rack awareness*



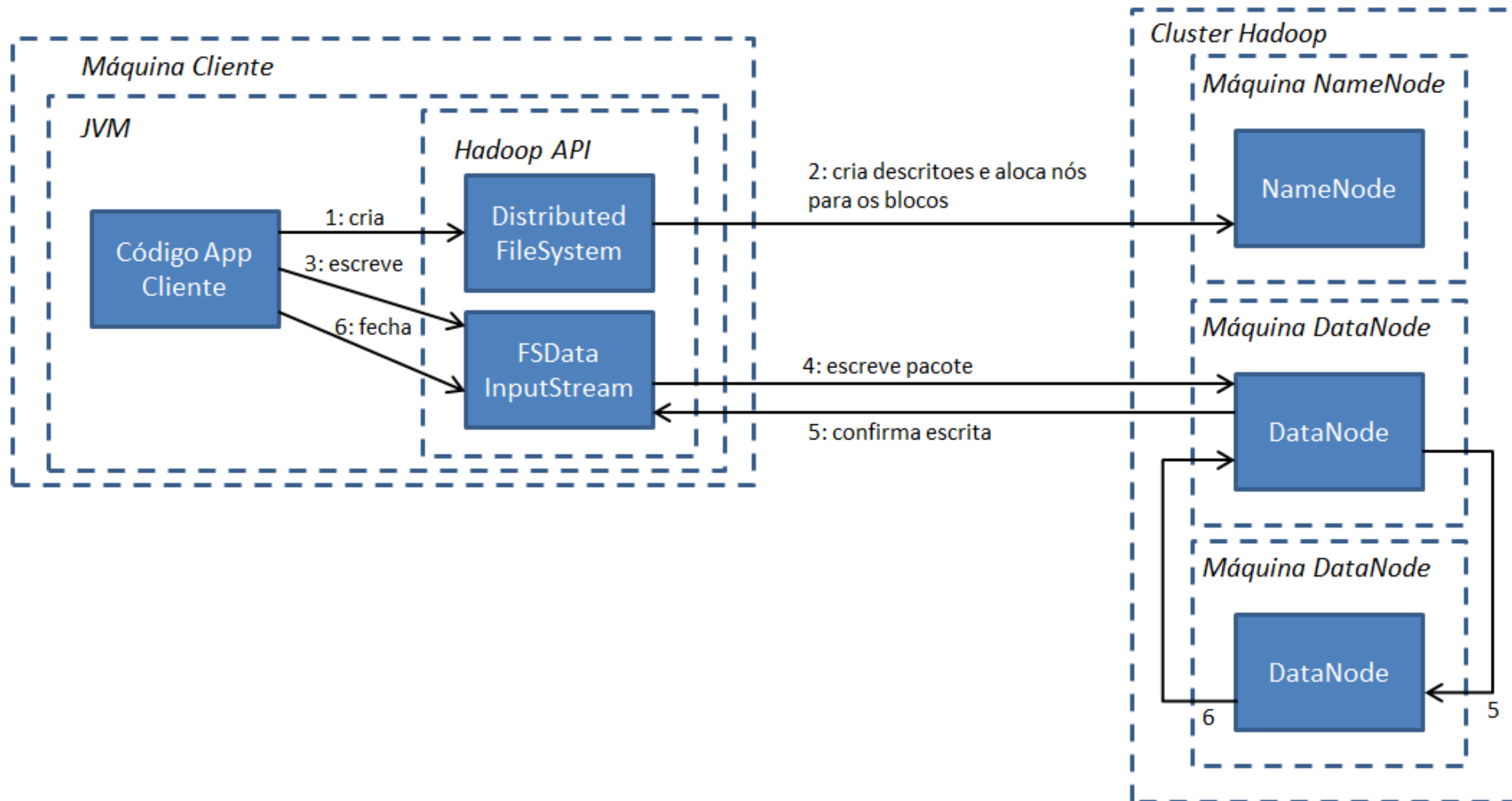
# Persistência com Hadoop

## Gerenciamento de Dados

- **HDFS = Namenode + SecondaryNamenode + Datanodes**
  - **Namenode**
    - Gerência dos blocos
    - Decide sobre replicação
    - Aloca nós para os blocos
    - Ponto de falha
  - **Datanode**
    - Contém os blocos de arquivos
    - Responsável pelos dados
  - **SecondaryNamenode**
    - Processo auxiliar ao Namenode
    - Não é backup do Namenode

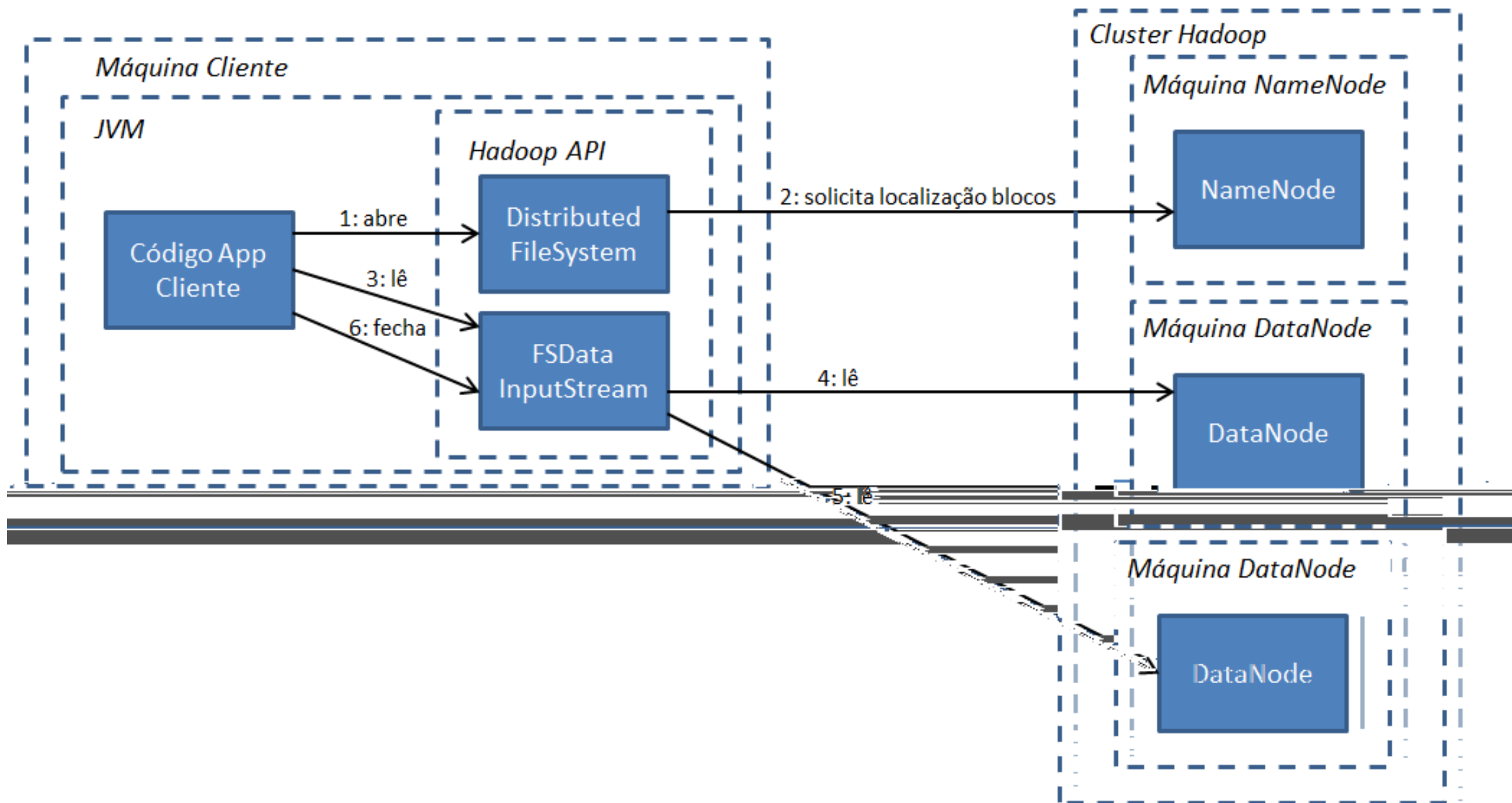
# HDFS Design

## Operação de Escrita



# HDFS Design

## Operação de Leitura



# Processando dados com Hadoop

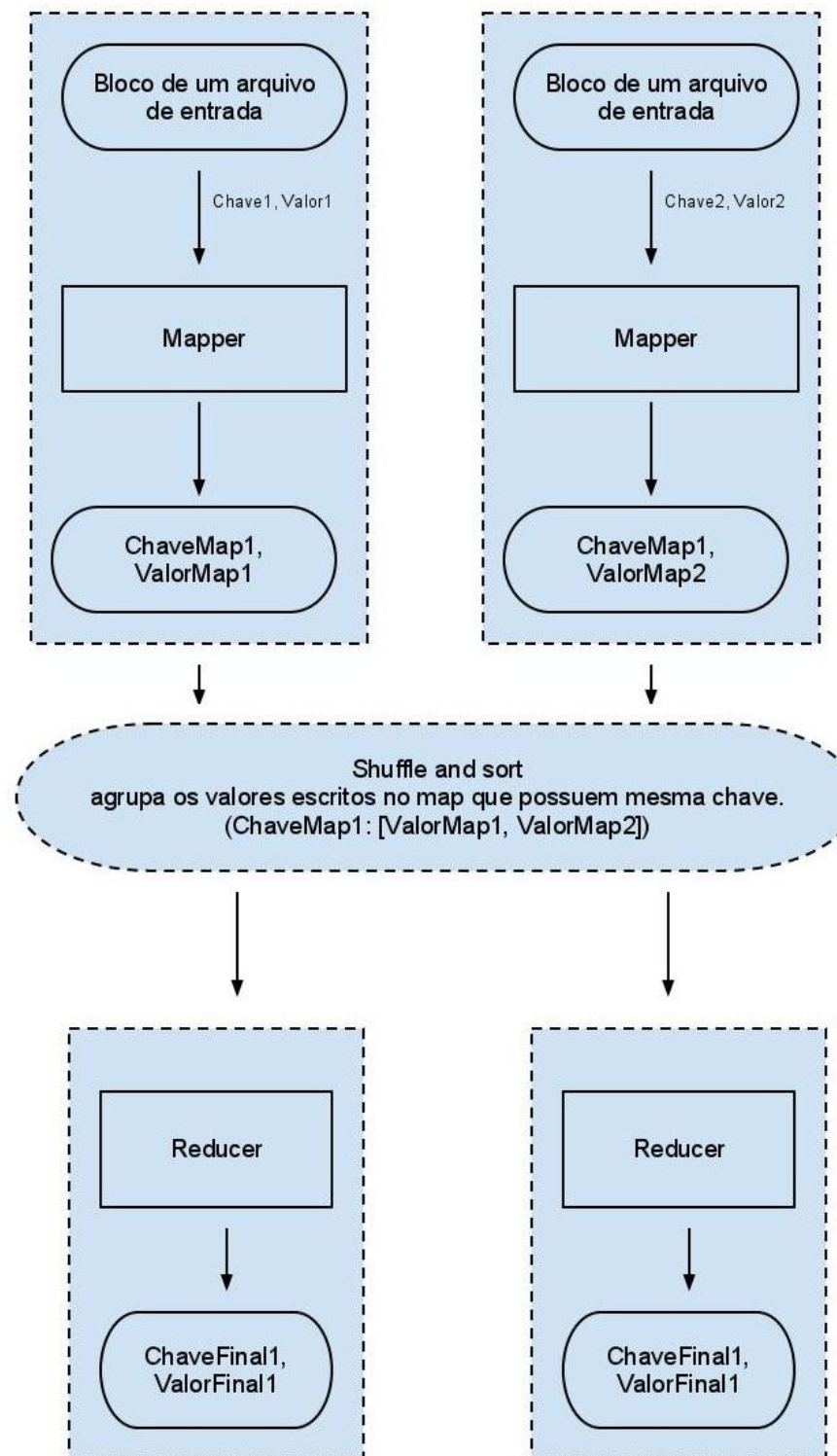
## MapReduce

- **MapReduce = JobTracker + TaskTrackers**
  - Leituras paralelas
  - Nós podem ser removidos/adicionados ao cluster sem necessidade de reinicialização
  - Processos sempre executam aonde os dados estão
  - Duas fases: Map + Reduce
  - Entre o Map e o Reduce: *Shuffle and Sort*
    - Envia os dados do *Mapper* para o *Reducer*
- Como o shell do linux:

```
~$ cat /my/log | grep '.*\.html' | sort | uniq -c > /my/output-file
```

# MapReduce

## Execução



# MapReduce

## Passo a passo

- **API Java para programação**
- **Linguagens script via *streamjob***
- **Paralelização e distribuição automática**
  - **Desenvolvedores só precisam se concentrar no Map e no Reduce**
  - **Exemplo WordCount**
    - **Texto de input do *Mapper*:**

mapper1	("um", 1)	("prato", 1)	("de", 1)	("trigo", 1)	("para", 1)	("um", 1)	("tigre", 1)
mapper2	("dois", 1)	("pratos", 1)	("de", 1)	("trigo", 1)	("para", 1)	("dois", 1)	("tigres", 1)

- **Dados intermediários gerados:**

mapper1	("um", 1)	("prato", 1)	("de", 1)	("trigo", 1)	("para", 1)	("um", 1)	("tigre", 1)
mapper2	("dois", 1)	("pratos", 1)	("de", 1)	("trigo", 1)	("para", 1)	("dois", 1)	("tigres", 1)

# MapReduce

## Passo Intermediário

- Entre o Map e o Reduce: *shuffle and sort*
- Resultado parcial:

“de”	1	1
“dois”	1	
“para”	1	1
“prato”	1	
“pratos”	1	
“tigre”	1	
“tigres”	1	
“trigo”	1	1
“um”	1	1



# MapReduce

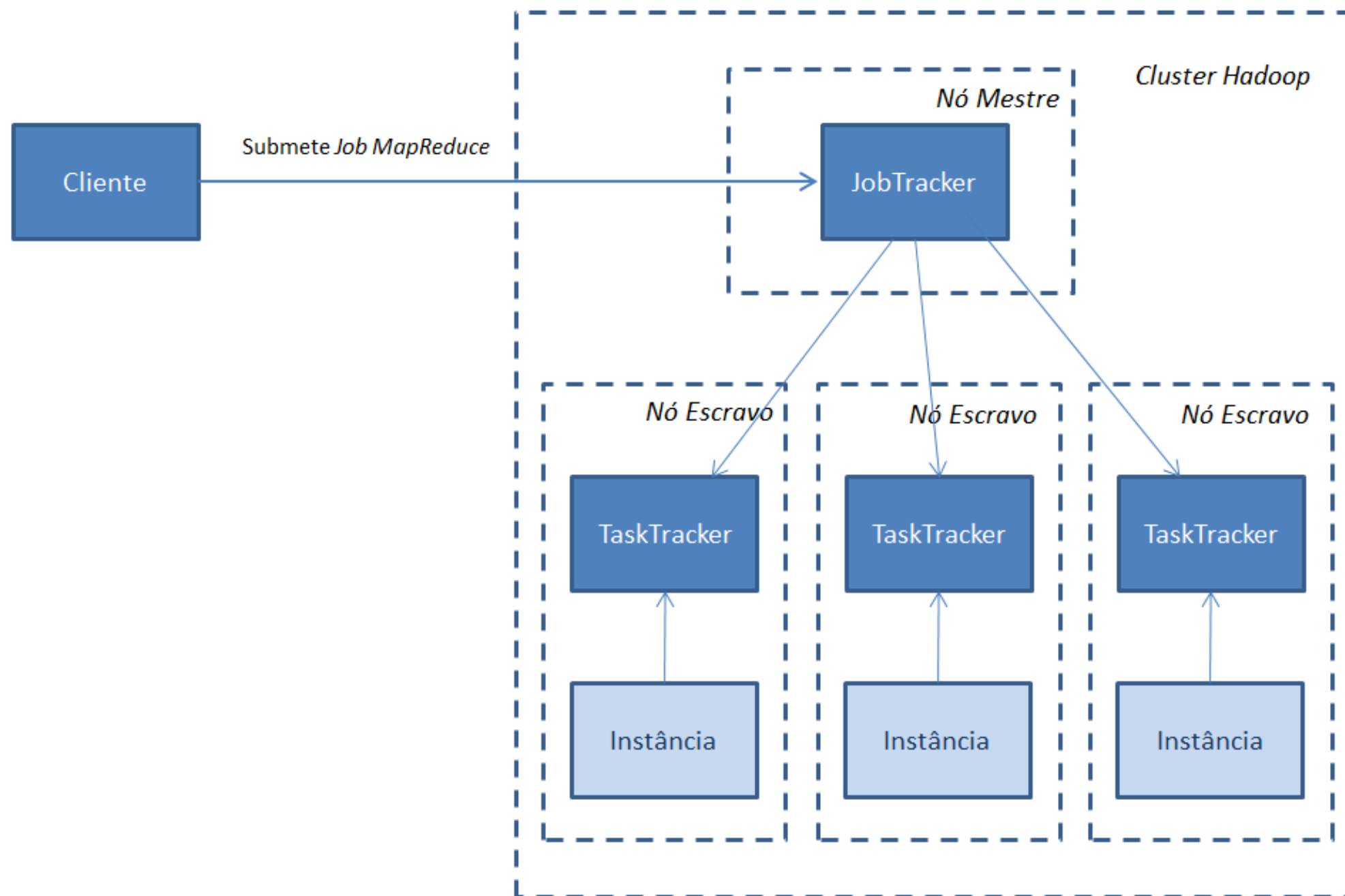
## Passo Final

- **Reduce - resumando tudo**
- **Resultado final:**

"de"	2
"dois"	1
"para"	2
"prato"	1
"pratos"	1
"tigre"	1
"tigres"	1
"trigo"	2
"um"	2

# MapReduce

## Execução



# Integrando Hadoop

- **Hadoop *facts***
  - Não é Banco de Dados
  - Processamento background
  - Precisa ser “alimentado” com dados
  - Não provê leitura randômica
  - API MapReduce Java
- **Desafios**
  - Integração com sistemas
    - Ler dados de outros sistemas
    - Disponibilizar resultados MapReduces
  - Time de BI e desenvolvedores não Java
- **Diversas Soluções “em cima” do Hadoop**
  - Ecossistema Hadoop

# **Ecosystem Hadoop**

- **Hive**
  - **Converte queries SQL em MapReduces**
- **Pig**
  - **Criado pelo Yahoo**
  - **Define uma linguagem de script para MapReduces denominada *PigLatin***
- **Hbase**
  - **“O banco de dados do Hadoop”**
  - **BD NoSQL orientado a colunas que funciona sobre o HDFS**
  - **Provê acesso randômico de leitura e escrita em tempo real a grandes quantidades de dados**
- **Flume**
  - **Sistema de exportação de logs contendo grande quantidade de dados para o HDFS**

# **Ecossistema Hadoop**

- **Sqoop**
  - **Ferramenta de exportação de dados de SGBDS para o Hadoop**
  - **Usa JDBC, gera uma classe Java de exportação de dados pra cada tabela no esquema relacional**
- **SCM**
  - **Ferramenta de auxílio para instalação e configuração de clusters Hadoop**
- **Ganglia**
  - **Monitoramento de Clusters Hadoop**

# Referências

- **Hadoop WebSite Oficial**
  - <http://hadoop.apache.org/>
- **Livros sobre Hadoop**
  - Hadoop: The Definitive Guide, Tom White, O'Reilly 2009
  - Hadoop In Action, Chuck Lam, Manning 2011
- **Site da Cloudera** - Contém informações, tutoriais e vídeo aulas sobre Hadoop
  - <http://www.cloudera.com>
- **Lista de discussão sobre Hadoop**
  - [mapreduce-user@hadoop.apache.org](mailto:mapreduce-user@hadoop.apache.org)
- **Artigos Técnicos sobre Hadoop:**
  - Processando Dados com Hadoop – MundoJ ed. 52
  - MapReduce Detalhado – MundoJ ed. 53
  - Usando o HDFS – MundoJ ed. 54
  - Big Data com Hadoop – Java Magazine ed 103

# Contatos

- **Email**
  - [Wellington.chevreuil@gmail.com](mailto:Wellington.chevreuil@gmail.com)
- **Linkedin**
  - <http://www.linkedin.com/profile/view?id=23184915>
- **Twitter**
  - @WChevreuil

