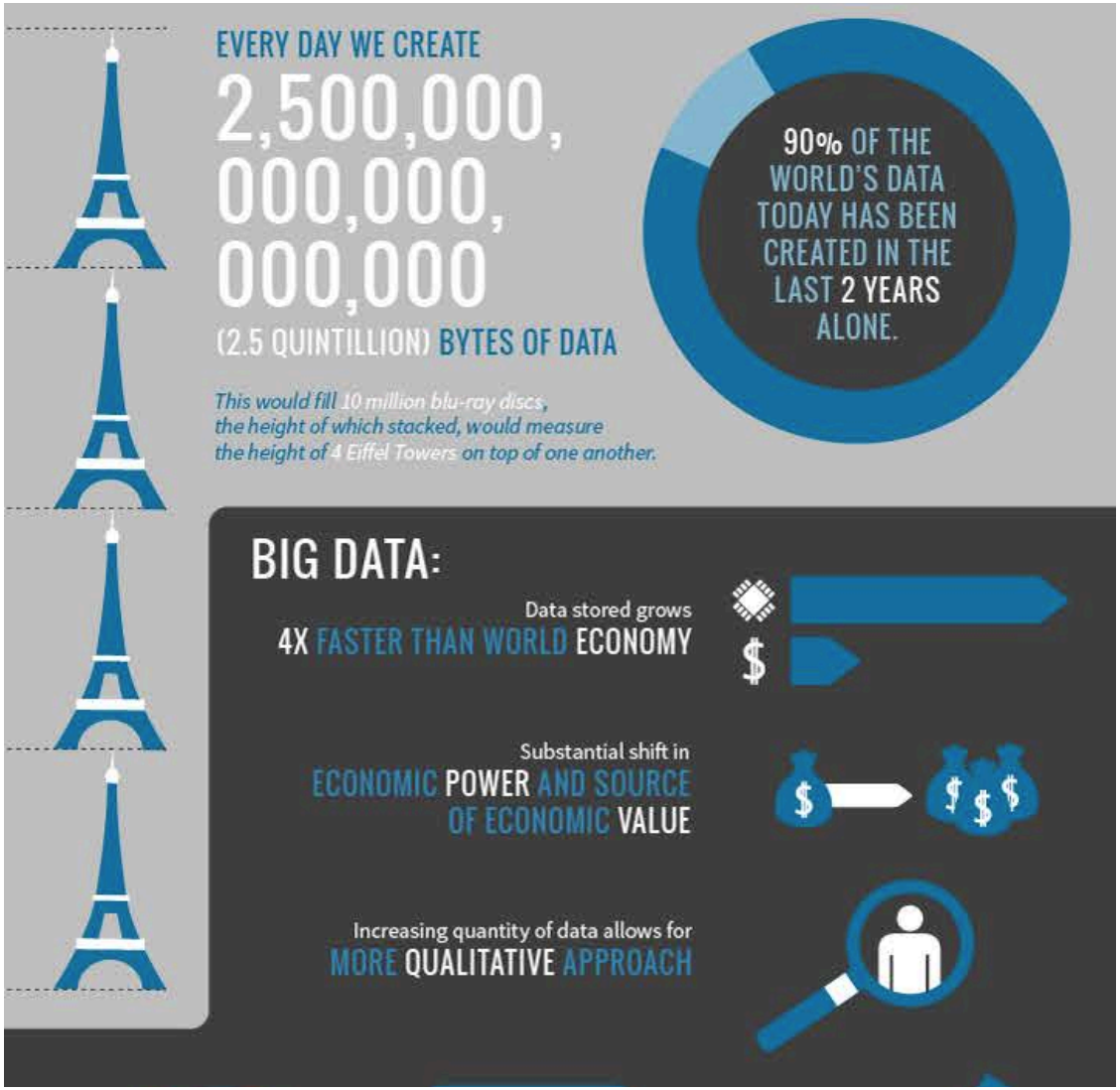
Computação em Nuvem

Fernando Antonio Mota Trinta

Map/Reduce

Contextualização



Fonte: vcloudnews

Contextualização e mais dados

- *2,7 bilhões de usuários na internet*
  - *5 bilhões de celulares no mundo*
  - *1 bilhões de smartphone vendidos em 2013*

90% dos dados no mundo hoje foram produzidos nos últimos dois anos

E mais dados...

- *Facebook*
  - *1B de usuários, 1,13 Trilhões de "likes", 219B de fotos e 140.3B de relacionamentos*
- *Youtube*
  - *100 horas de vídeos adicionado a cada minuto*
- *Yahoo!*
  - *+ de 650M de usuários, 11B visitas a páginas/mês*
- *Flick*
  - *+ de 5B de fotos*
- *Twitter*
  - *80 TB e 1B de tweets por dia*
- *Boeing*
  - *640 TB gerados em um voo transatlântico*
- *Wal-Mart*
  - *2,5 PB e 1 milhão de transações/hora*
- *LHC CERN*
  - *15 Petabytes por ano*

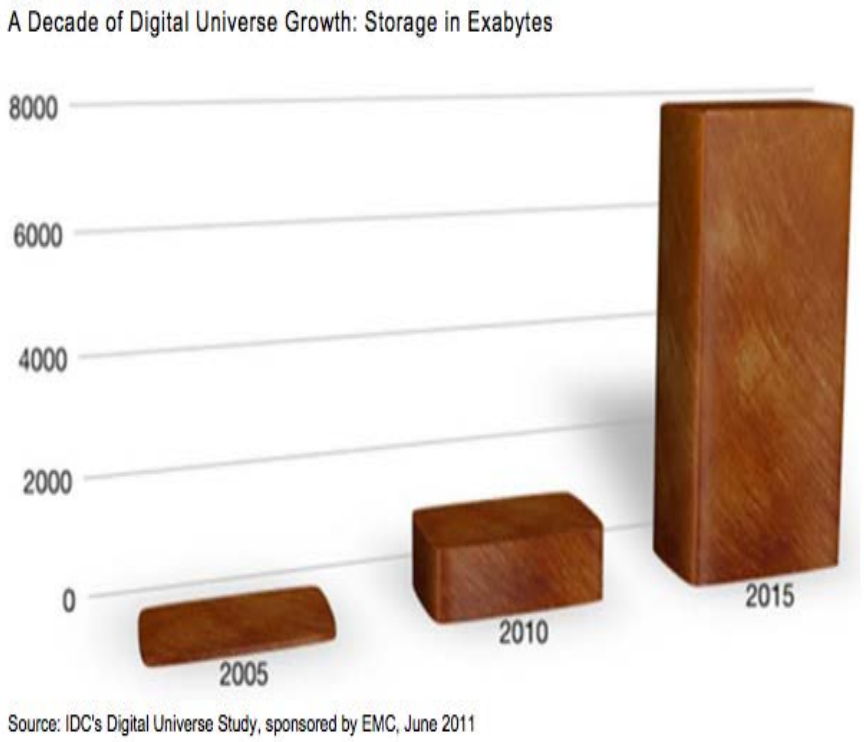
Comportamento

1990s

2010s

Dados x Informação

- "*Extracting Value from Chaos" - a informação mundial está dobrando a cada 2 anos - 1.8 zettabytes foram criados em 2011, crescendo mais que a lei de Moore.*



Contextualização

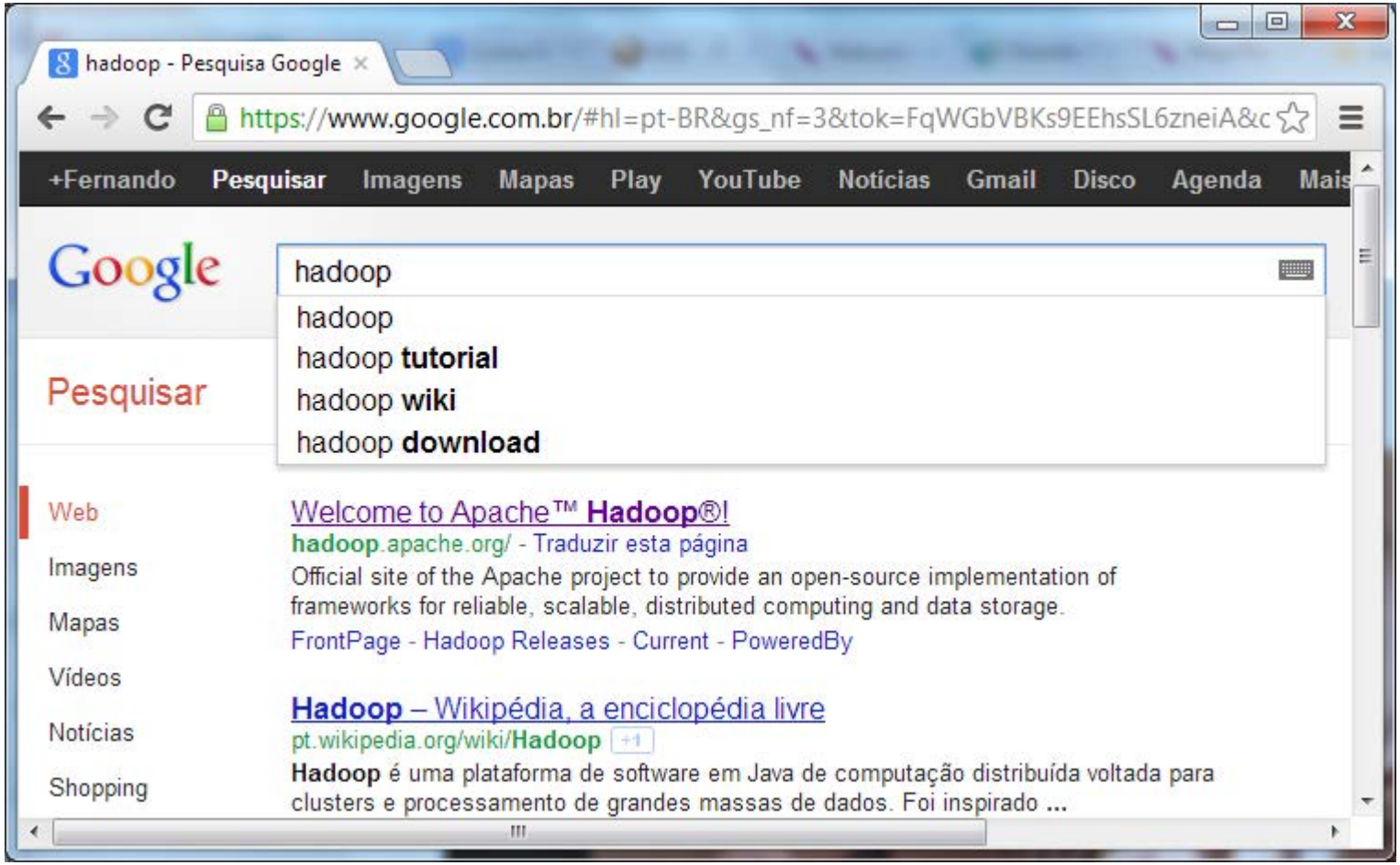
- *Grande quantidades de dados geram desafios*
  - *Armazenamento…*
  - *Processamento…*
  - *Análise…*

- *BigData & Data Analytics*
  - *ciência de examinar os dados brutos com a finalidade de tirar conclusões sobre essa informação…*
  - *Dados analisados são valiosos hoje*

Exemplos

LinkedIn

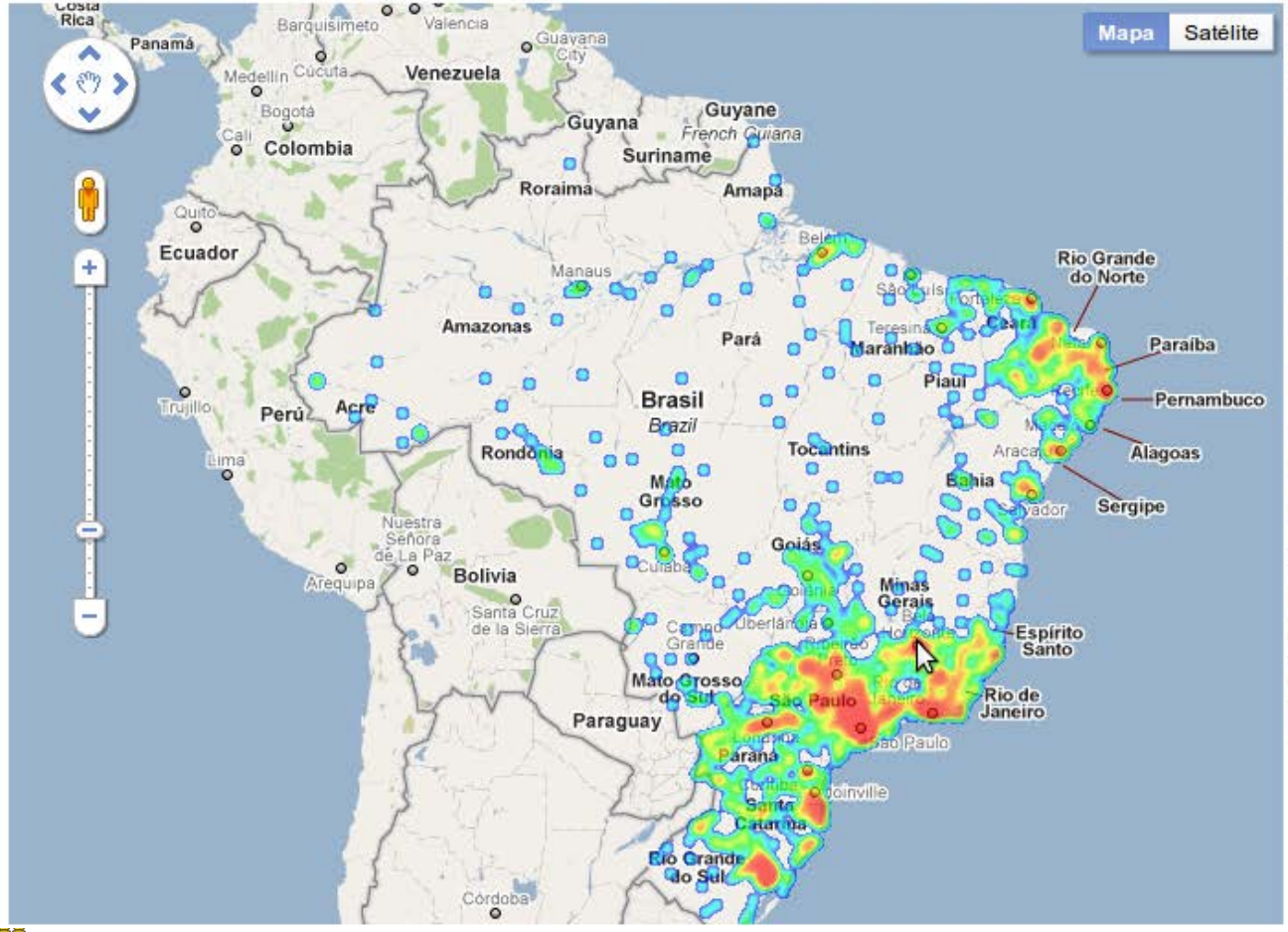
Exemplos


hadoop - Pesquisa Google
https://www.google.com.br/#hl=pt-BR&gs_nf=3&tok=FqWGbVBKs9EEhsSL6zneiA&c
+Fernando
Pesquisar
Imagens
Mapas
Play
YouTube
Notícias
Gmail
Disco
Agenda
Google
hadoop
hadoop
hadoop tutorial
hadoop wiki
hadoop download
Pesquisar
Web
Imagens
Mapas
Vídeos
Notícias
Shopping
Welcome to Apache™ Hadoop®!
hadoop.apache.org/ - Traduzir esta página
Official site of the Apache project to provide an open-source implementation of frameworks for reliable, scalable, distributed computing and data storage.
FrontPage - Hadoop Releases - Current - PoweredBy
Hadoop – Wikipédia, a enciclopédia livre
pt.wikipedia.org/wiki/Hadoop
Hadoop é uma plataforma de software em Java de computação distribuída voltada para clusters e processamento de grandes massas de dados. Foi inspirado ...

Dengue Watch: Heat Map

observatóriodadengue

Menções à dengue no Twitter no mês de fev/2011



Clique nos pontos do mapa para informações

Cidades: 11 Tweets: 59 População: 1925450 Tx. Inc. Méd.: 1.5334e-04

Cidade	Pop.	Tweets	Tx.Inc
Betim	377547	14	1.8230e-04
Brumadinho	34013	1	1.4289e-04
Contagem	603048	22	1.7922e-04
Ibirite	159026	4	1.2110e-04
Itabira	109551	6	2.7305e-04
Itauna	85396	1	5.2124e-05
Joao Monlevade	73451	4	2.7146e-04
Matozinhos	32973	1	1.4765e-04
Ribeirao das Neves	296376	2	2.6665e-05
Sabara	126219	3	1.1399e-04
Santa Barbara	27850	1	1.7627e-04

Fonte: Alberto Laender

Campos de Aplicação

- *Sistemas de Recomendação*
- *Processamento de Linguagem Natural*
- *Data Warehousing*
- *Pesquisa de Mercado*
- *Análise Financeira*
- *Máquinas de inferência*
- *Processamento de Vídeo/Imagens*
- *Análise de Logs*

Campos de Aplicação

- *Ciências da Saúde*
- *Gestão governamental*
- *Redes Sociais*
- *Telecomunicações*

Como armazenar e processar este grande volume de dados?

Computação em Nuvem

"*Grandes poderes trazem grandes responsabilidades!*"

- *O grande volume de dados demanda*
  - *Grande poder de processamento*
  - *Paralelizar e Distribuir tarefas*
  - *Facilidade de processamento*

Computação em Nuvem

- *Para tratar problemas de larga escala, idealmente não gostaríamos de se preocupar com:*
  - *Paralelização e distribuição automática*
  - *Tolerância a falhas*
  - *Escalonamento de I/O*
  - *Monitoramento de tarefas*
- *Então é necessário se adequar a um modo de facilitar o uso da nuvem para tarefas de manipulação de dados*

Modelo de Programação para Nuvem

- *Forma, abordagem ou maneira específica de como se programar, dentro do contexto de uma aplicação ou domínio*
  - *Abstrações adequadas*
  - *Eficiência*
- *Duas abordagens:*
  - *Estender um modelo existente*
  - *Propor um novo modelo*

Map/Reduce

- *Um modelo de programação e uma implementaçãoo associada para processamento e geração de grandes conjuntos de dados*
  - *Inspirado pelas primitivas map e reduce encontrados na Lisp e em outras linguagens funcionais*
- *Permite paralelizar grandes computações facilmente e usar re-execução como mecanismo para tolerância a falhas*

Map/Reduce

História

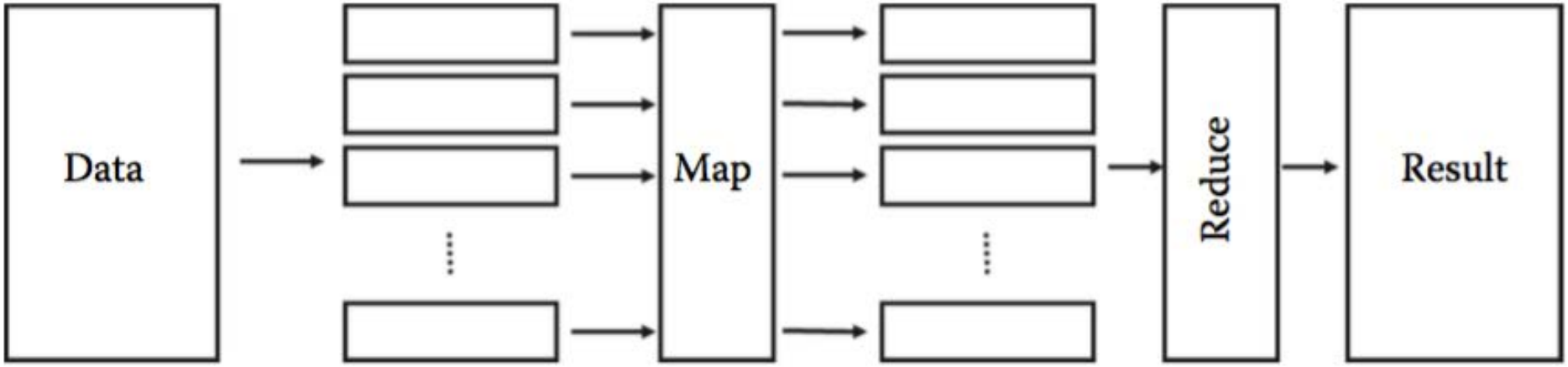
- *Originalmente desenvolvido pela Google*
  - *Jeffrey Dean e Sanjay Ghemawat*
  - *MapReduce: Simplified Data Processing on Large Clusters. OSDI'04: Sixth Symposium on Operating System Design and Implementation (December 2004)*
- *Usado no Search Engine para tratar a quantidade de dados a serem processados*

Map/Reduce

Idéia Geral

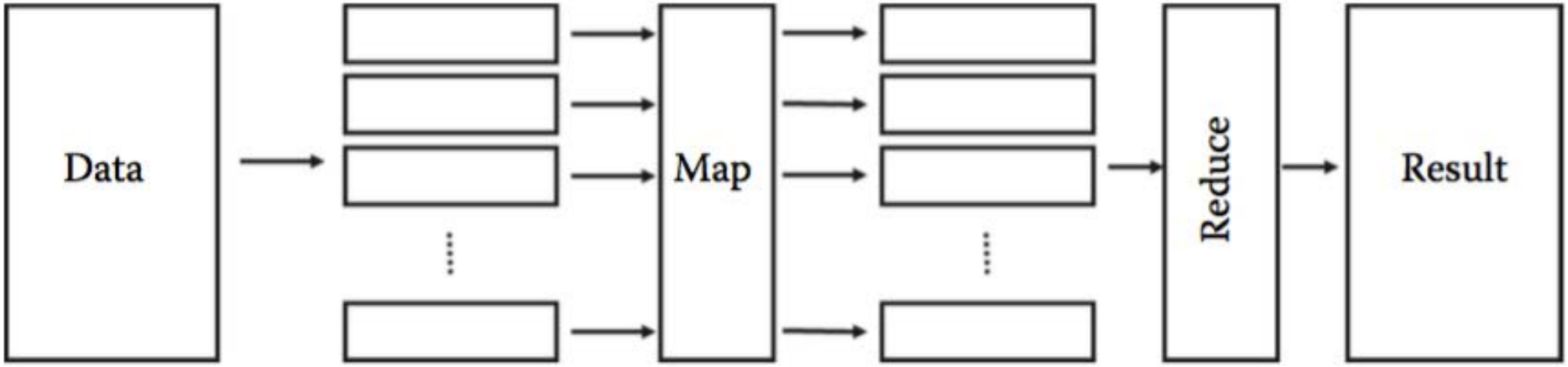
- *Divida uma tarefa que deve processar um grande conjunto de dados em partes*
- *Cada parte deve ser responsável por uma parte pequena do conjunto*
- *Cada parte é independente*
- *Após o término do processamento das partes individuais, junte os resultados das partes*

Map/Reduce



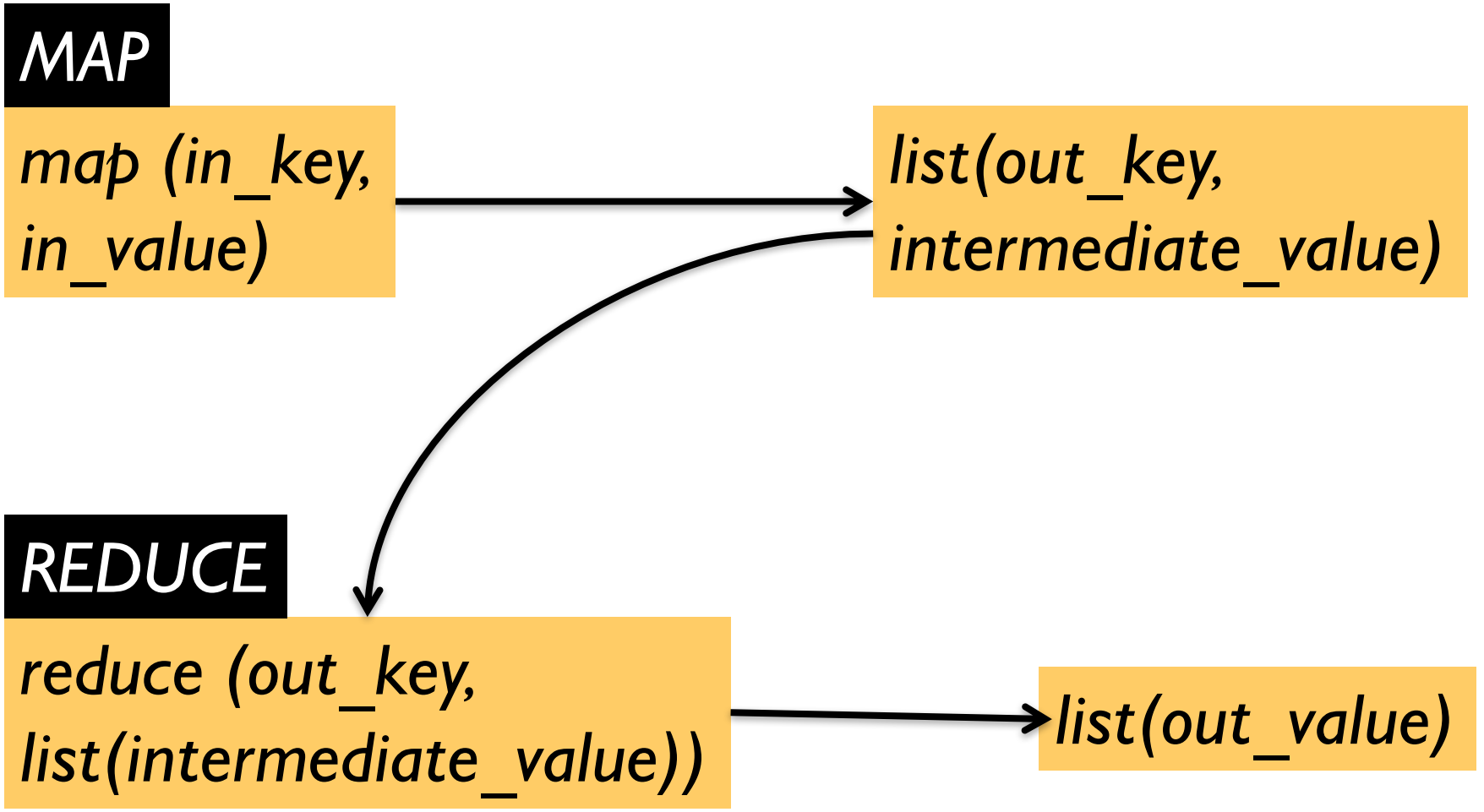
- *MAP:*
  - *Toma-se uma tarefa complexa ou custosa*
  - *Quebra tal tarefa em sub-problemas menores*
  - *Delega a resolução desta tarefas a nós distribuídos*

Map/Reduce



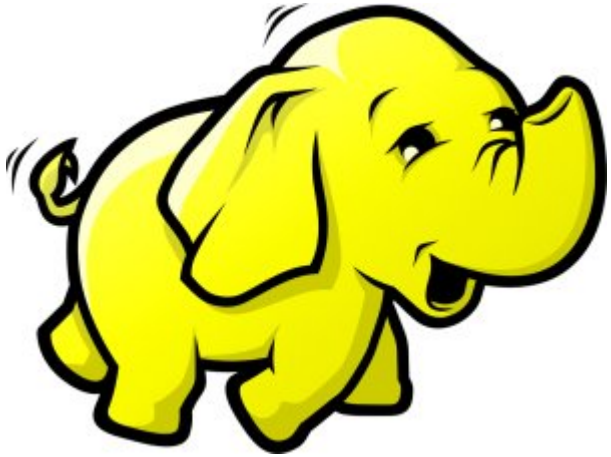
- *REDUCE:*
  - *Coleta as respostas dos nós distribuídos*
  - *Agrega tais respostas em uma saída que representa a solução do problema complexo*

Map/Reduce


MAP
map (in_key, in_value)
list(out_key, intermediate_value)
REDUCE
reduce (out_key, list(intermediate_value))
list(out_value)

Apache Hadoop

- *Inspirado em iniciativas da Google*
  - *BigTable e Map/Reduce*
- *Criado por Doug Cutting*
  - *Criador do Lucene*



Hadoop

Um framework open-source de propósito geral, orientado a processamento em lote/offline, de uso intensivo de dados (I/O intensive) utilizado para criação de aplicações que processem uma grande quantidade de dados.

O que seria grande?

- *25K máquinas*
- *Dezenas de clusters*
- *3 Pb de dados*
- *10000 jobs/semana*

Quem usa mais

http://wiki.apache.org/hadoop/PoweredBy

O Hadoop não é...

... Um banco de dados relacional

... Um sistema online de processamento de transações

... Um sistema de armazenamento estruturado de qualquer tipo

Hadoop x Relacional

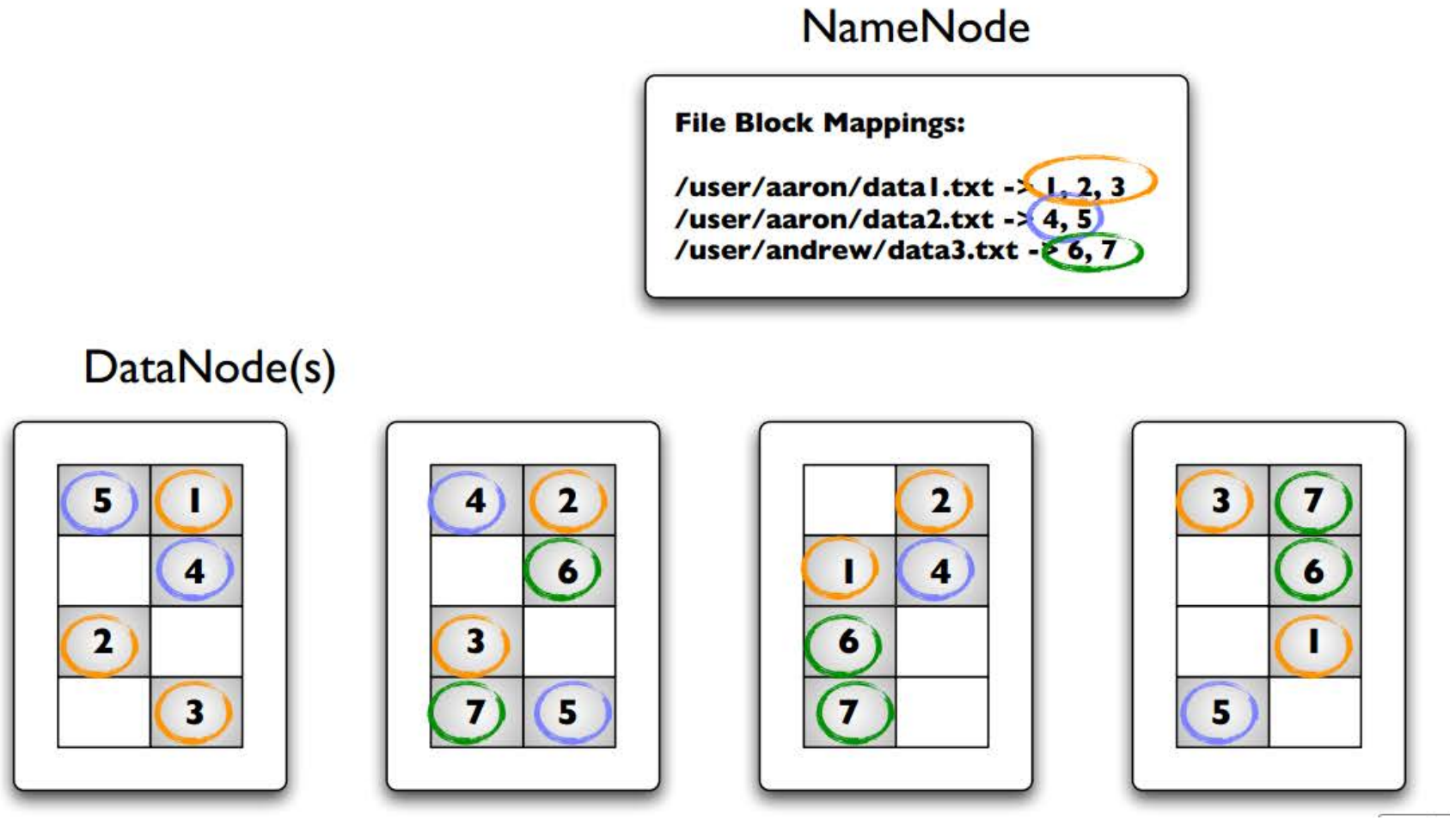
Hadoop	*Relacional*
Pares de chave/valor	*Tabelas*
Informa-se como processar dados	*Informa-se como se deseja obter os dados (SQL)*
Offline/Lote	*Online/Tempo Real*
Escalabilidade Horizontal	*Escalabilidade Vertical*

Hadoop

- *Componentes principais*
  - *Um sistema distribuído de arquivos*
    - *Haddop Distributed File System (HDFS)*
  - *Um sistema de Map/Reduce*

HDFS

- *Dados são replicados e distribuídos em vários nós*
  - *Fator de replicação: 3*
- *Projetado para arquivos grandes*
  - *Terabytes*
- *Orientado a blocos*
- *Comandos a lá linux*
  - *ls, cp, mv, rm, etc*


NameNode
File Block Mappings:
/user/aaron/data1.txt -> 1, 2, 3
/user/aaron/data2.txt -> 4, 5
/user/andrew/data3.txt -> 6, 7
DataNode(s)
5 1
4
2
3
4 2
6
3
7 5
2
1 4
6
7
3 7
6
1
5

Coisas Boas

- *Tolerância a falhas*
  - *Sistema ativo mesmo no caso de falha de alguns nós*
- *Self-Healing*
  - *Auto balanceamento dos arquivos*
- *Escalável*
  - *Adição de novos nós do cluster*

Coisas Boas

- *Código Aberto*
  - *Comunidade ativa*
  - *Apoio de grandes coorporações*
- *Economia*
  - *Software livre*
  - *Uso de máquinas convencionais*
- *Separação da Lógica de Negócios*
  - *Trabalho duro fica com o Hadoop*

Coisas não tão boas

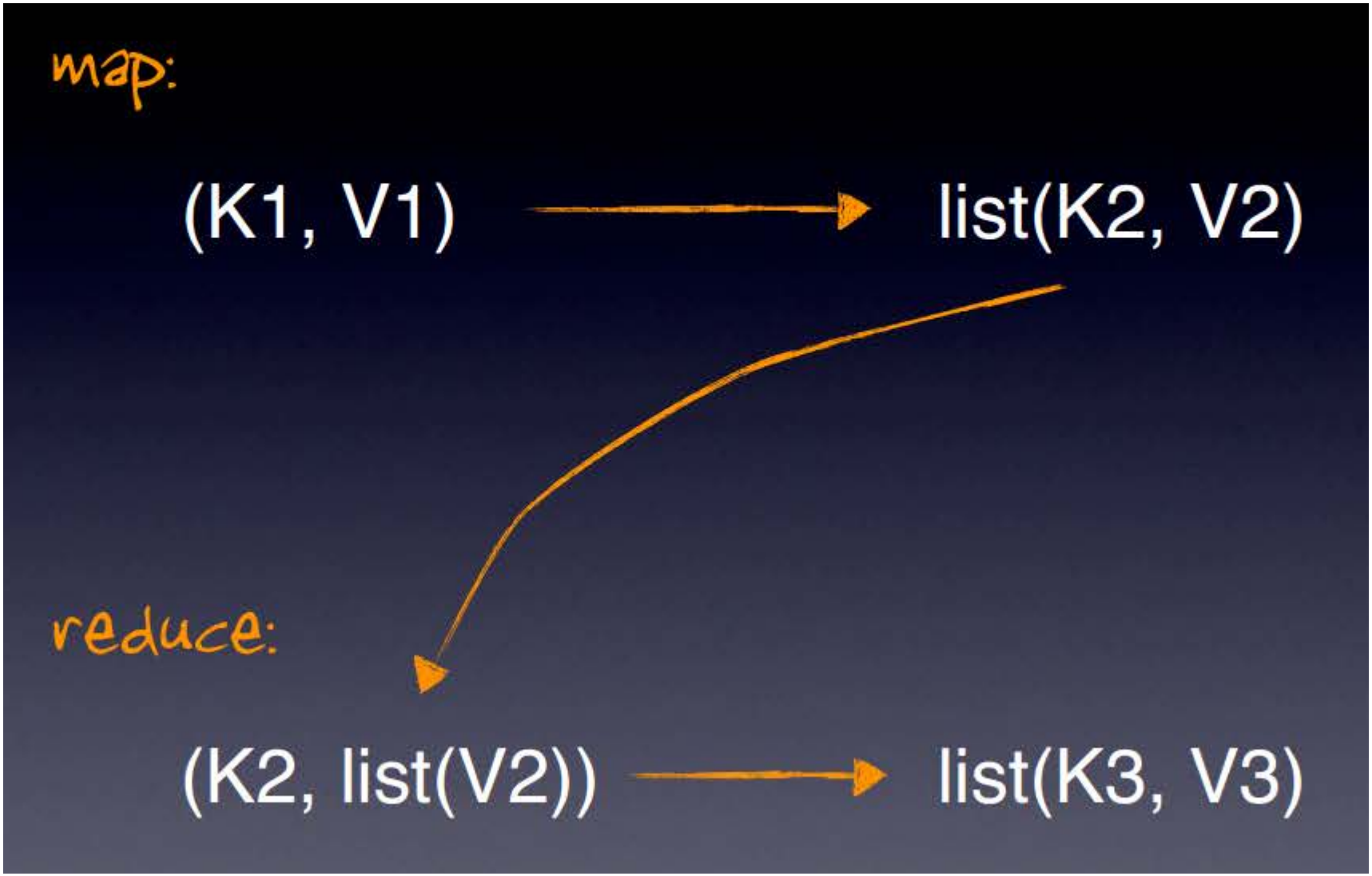
- *Único nó mestre*
  - *Ponto crítico de falha*
- *Paralelização de aplicações*
  - *Problemas não paralelizáveis*
  - *Processamento de arquivos pequenos*
    - *Muito processamento & poucos dados*

Map/Reduce

- *Base na programação funcional*
  - *Manipulação de dados (tipicamente listas)*
  - *Funções que transformam dados*
- *Modelo de Programação proposto pelo Google*
  - *Duas funções básicas: MAP e REDUCE*

Map/Reduce

- *MAP:*
  - *Toma-se uma tarefa complexa ou custosa*
  - *Quebra tal tarefa em sub-problemas menores*
  - *Delega a resolução desta tarefas a nós distribuídos (workers)*
- *REDUCE:*
  - *Coleta as respostas dos workers*
  - *Agrega tais respostas em uma saída que representa a solução do problema complexo*


map:
(K1, V1)
list(K2, V2)
reduce:
(K2, list(V2))
list(K3, V3)

Apache Hadoop

- *Divide arquivos*
  - *Em geral, blocos de 64Mb*
- *Usa pares de chave/valor*
- *Mappers*
  - *filtram e transformam o dado de entrada*
- *Reducers*
  - *Agregam a saída dos mappers*

Importante premissa

- *MOVE-SE CÓDIGO, NÃO DADOS*
  - *Arquivos são grandes, código não*
  - *Rede é um gargalo*

Clássico Exemplo

- *Word Count*
  - *Contar o número de ocorrência de uma palavra em um arquivo*

Fernando Antonio Mota Trinta
Ian Gabriel Braga Trinta
Ivana Régia Braga

Map (K1,V1)

- *(0, "Fernando Antonio Mota Trinta")*
- *(29, "Ian Gabriel Braga Trinta")*
- *(53, "Ivana Régia Braga")*

Map

list(K2,V2)

- *(“Fernando”, 1)*
- *(“ Antonio”, 1)*
- *(“ Mota ”, 1)*
- *(“ Trinta ”, 1)*
- *(“ Ian ”, 1)*
- *(“ Gabriel ”, 1)*
- *(“ Braga ”, 1)*
- *(“ Trinta ”, 1)*
- *(“ Ivana ”, 1)*
- *(“ Régia ”, 1)*
- *(“ Braga ”, 1)*

Reduce

(K2, list(V2))

- *(“Fernando”, 1)*
- *(“ Antonio”, 1)*
- *(“ Mota ”, 1)*
- *(“ Trinta ”, (1,1))*
- *(“ Ian ”, 1)*
- *(“ Gabriel ”, 1)*
- *(“ Braga ”, (1,1))*
- *(“ Ivana ”, 1)*
- *(“ Régia ”, 1)*

Reduce

list(V3,K3)

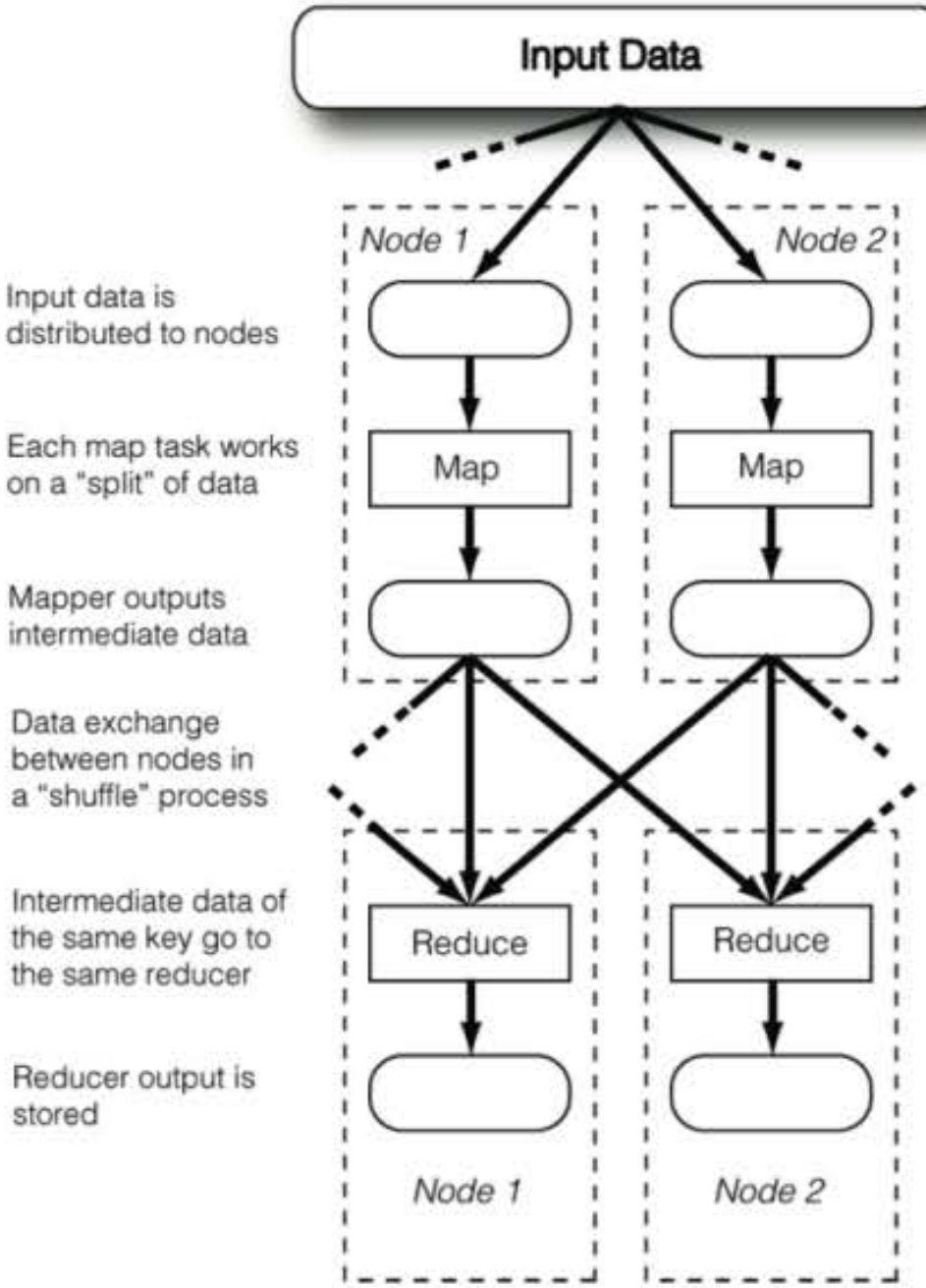
- *(“Fernando”, 1)*
- *(“ Antonio”, 1)*
- *(“ Mota ”, 1)*
- *(“ Trinta ”, 2)*
- *(“ Ian ”, 1)*
- *(“ Gabriel ”, 1)*
- *(“ Braga ”, 2)*
- *(“ Ivana ”, 1)*
- *(“ Régia ”, 1)*

```
public class SimpleWordCount
  extends Configured implements Tool {

  public static class MapClass
    extends Mapper<Object, Text, Text, IntWritable> {
    ...
  }

  public static class Reduce
    extends Reducer<Text, IntWritable, Text, IntWritable> {
    ...
  }

  public int run(String[] args) throws Exception { ... }

  public static void main(String[] args) { ... }
}
```

```
public static class MapClass
  extends Mapper<Object, Text, Text, IntWritable> {

  private static final IntWritable ONE = new IntWritable(1L);
  private Text word = new Text();

  @Override
  protected void map(Object key, Text value, Context context)
    throws IOException, InterruptedException {

    StringTokenizer st = new StringTokenizer(value.toString());
    while (st.hasMoreTokens()) {
      word.set(st.nextToken());
      context.write(word, ONE);
    }
  }
}
```

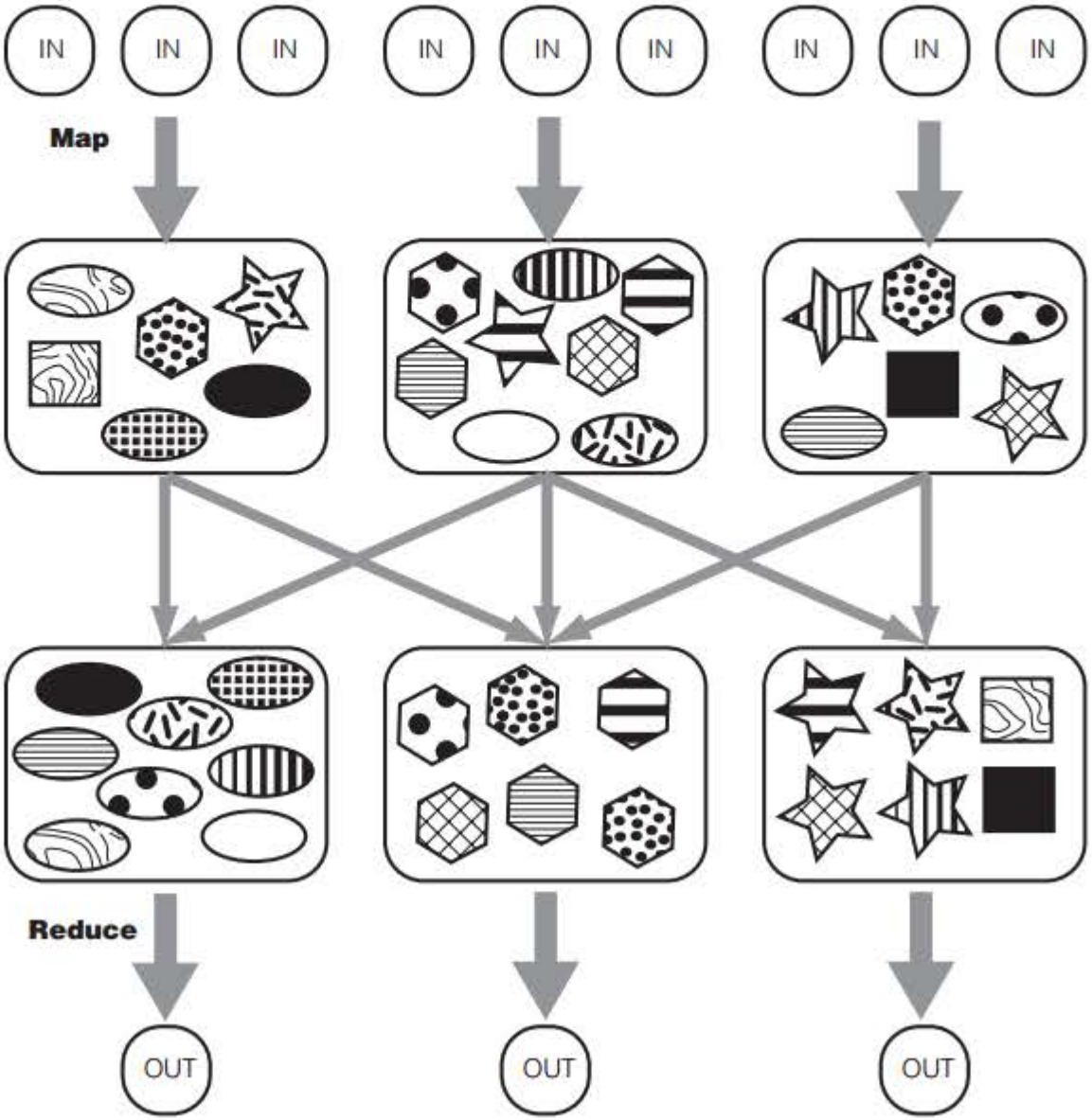
```
public static class Reduce
  extends Reducer<Text, IntWritable, Text, IntWritable> {

  private IntWritable count = new IntWritable();

  @Override
  protected void reduce(Text key, Iterable<IntWritable> values,
                        Context context)
    throws IOException, InterruptedException {

    int sum = 0;
    for (IntWritable value : values) {
      sum += value.get();
    }
    count.set(sum);
    context.write(key, count);
  }
}
```

```
public int run(String[] args) throws Exception {
  Configuration conf = getConf();

  Job job = new Job(conf, "Counting Words");
  job.setJarByClass(SimpleWordCount.class);
  job.setMapperClass(MapClass.class);
  job.setReducerClass(Reduce.class);
  job.setOutputKeyClass(Text.class);
  job.setOutputValueClass(Text.class);

  FileInputFormat.setInputPaths(job, new Path(args[0]));
  FileOutputFormat.setOutputPath(job, new Path(args[1]));

  return job.waitForCompletion(true) ? 0 : 1;
}
```


Input Data
Node 1
Node 2
Input data is distributed to nodes
Each map task works on a "split" of data
Map
Map
Mapper outputs intermediate data
Data exchange between nodes in a "shuffle" process
Intermediate data of the same key go to the same reducer
Reduce
Reduce
Reducer output is stored
Node 1
Node 2

Particionamento

- *Decisão de quais chaves vão para qual reducer*
- *Mundo ideal*
  - *Distribuição uniforme entre reducers*
- *Evitar a sobrecarga de um único reducer*

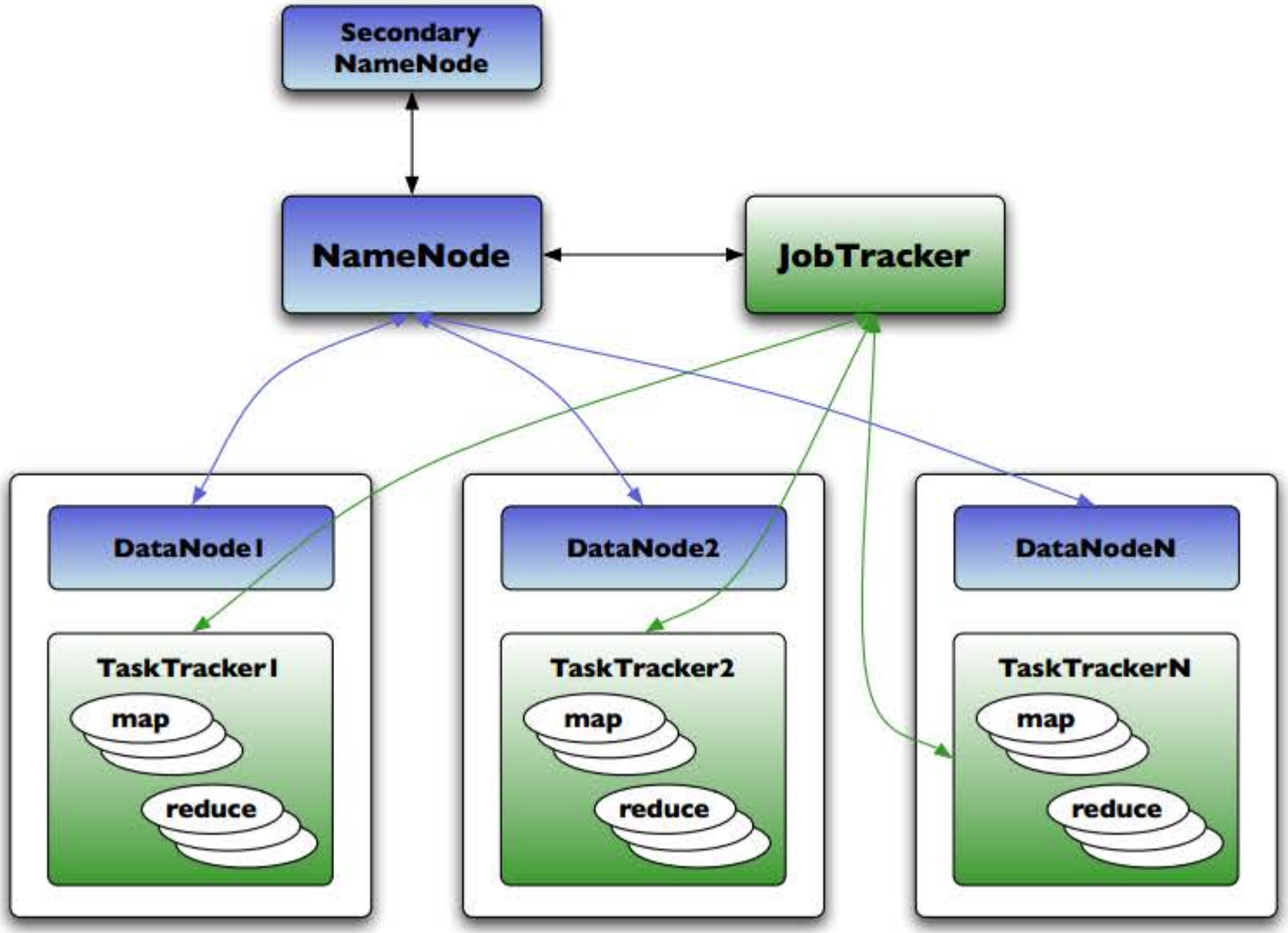

IN
IN
IN
IN
IN
IN
IN
IN
IN
Map
Reduce
OUT
OUT
OUT

Melhorando desempenho

- *Combinando localmente*
- *Reduzir o processo de agregação nos mappers*
  - *Aka "Local Reduce"*

	Dado	# pares k/v agregados
Sem combinação	*("Maria", 1)*	*1000*
Com combinação	*("Maria", 1000)*	*1*

Arquitetura Hadoop



NameNode

- *Gerente do HDFS*
  - *Gerencia os DataNodes*
- *Ponto central de falha*
- *Não deve armazenar dados ou rodar processos*

NameNode 'localhost:9000'

Started: Tue Sep 07 16:58:53 EDT 2010
Version: 0.20.1, r810220
Compiled: Tue Sep 1 20:55:56 UTC 2009 by oom
Upgrades: There are no upgrades in progress.

Browse the filesystem
Namenode Logs

Cluster Summary

786 files and directories, 390 blocks = 1176 total. Heap Size is 81.06 MB / 995.88 MB (8%)

Configured Capacity	:	465.44 GB
DFS Used	:	1.67 GB
Non DFS Used	:	201.38 GB
DFS Remaining	:	262.39 GB
DFS Used%	:	0.36 %
DFS Remaining%	:	56.37 %
Live Nodes	:	1
Dead Nodes	:	0

NameNode Storage:

Storage Directory	Type	State
/data/hadoop-pseudo/dfs/name	IMAGE_AND_EDITS	Active

DataNode

- *Armazena os blocos de arquivos*
- *Não armazena arquivos contíguos*
- *Informa dados do bloco ao NameNode*
- *Recebe tarefas do NameNode*

Secondary NameNode

- *Snapshot do NameNode*
- *Não é um servidor automático em caso de falhas*
  - *Apenas minimiza o tempo de queda/perda de dados em caso de falha do NameNode*

JobTracker

- *Particiona as tarefas através do HDFS*
- *Rastreia as tarefas map/reduce*
  - *Reexecuta tarefas em diferentes nós em caso de falha*

localhost Hadoop Map/Reduce Administration

State: RUNNING
Started: Tue Sep 07 16:58:58 EDT 2010
Version: 0.20.1, r810220
Compiled: Tue Sep 1 20:55:56 UTC 2009 by oom
Identifier: 201009071658

Cluster Summary (Heap Size is 81.06 MB/995.88 MB)

Maps	Reduces	Total Submissions	Nodes	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes
0	0	26	1	2	2	4.00	0

Scheduling Information

Queue Name	Scheduling Information
default	N/A

Filter (Jobid, Priority, User, Name)
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information
job_201009071658_0034	NORMAL	sleberkn	CitationHistogram	0.00%	1	0	0.00%	1	0	NA

Completed Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information
job_201009071658_0001	NORMAL	sleberkn	com.nearinfinity.hadoop.wordcount.SimpleWordCount	100.00%	10	10	100.00%	1	1	NA
job_201009071658_0003	NORMAL	sleberkn	BetterWordCount	100.00%	10	10	100.00%	1	1	NA
job_201009071658_0005	NORMAL	sleberkn	BetterWordCount	100.00%	10	10	100.00%	1	1	NA

Task Tracker

- *Rastreia tarefas de map & reduce individualmente*
- *Relata o progresso de uma tarefa ao JobTracker*

Hadoop map task list for job_201009071658_0035 on localhost

All Tasks

Task	Complete	Status	Start Time	Finish Time	Errors	Cou
task_201009071658_0035_m_000000	100.00%	65300 lines processed from: hdfs://localhost:9000/user/sleberkn /wordcount/input/WarAndPeace.txt	7-Sep-2010 22:56:08	7-Sep-2010 22:56:20 (12sec)		11
task_201009071658_0035_m_000001	100.00%	54500 lines processed from: hdfs://localhost:9000/user/sleberkn /wordcount/input/TheCountOfMonteCristo.txt	7-Sep-2010 22:56:08	7-Sep-2010 22:56:17 (9sec)		11
task_201009071658_0035_m_000002	100.00%	33000 lines processed from: hdfs://localhost:9000/user/sleberkn /wordcount/input/Ulysses.txt	7-Sep-2010 22:56:17	7-Sep-2010 22:56:26 (9sec)		11
task_201009071658_0035_m_000003	100.00%	22100 lines processed from: hdfs://localhost:9000/user/sleberkn /wordcount/input/MobyDick.txt	7-Sep-2010 22:56:20	7-Sep-2010 22:56:29 (9sec)		10
task_201009071658_0035_m_000004	100.00%	16600 lines processed from: hdfs://localhost:9000/user/sleberkn /wordcount/input/Dracula.txt	7-Sep-2010 22:56:26	7-Sep-2010 22:56:32 (6sec)		10
task_201009071658_0035_m_000005	100.00%	11700 lines processed from: hdfs://localhost:9000/user/sleberkn /wordcount/input/AdventuresOfHuckleberryFinn.txt	7-Sep-2010 22:56:29	7-Sep-2010 22:56:35 (6sec)		10
task_201009071658_0035_m_000006	100.00%	13000 lines processed from: hdfs://localhost:9000/user/sleberkn /wordcount/input/AdventuresOfSherlockHolmes.txt	7-Sep-2010 22:56:32	7-Sep-2010 22:56:38 (6sec)		10
task_201009071658_0035_m_000007	100.00%	3600 lines processed from: hdfs://localhost:9000/user/sleberkn /wordcount/input/TheTimeMachine.txt	7-Sep-2010 22:56:35	7-Sep-2010 22:56:38 (3sec)		10
task_201009071658_0035_m_000008	0.00%		7-Sep-2010 22:56:38			0
task_201009071658_0035_m_000009	0.00%		7-Sep-2010 22:56:38			0