

Insper

Megadados

Spark

Engenharia

Maciel Vidal

Quanto dados processa o Google

Em **2008**, o Google processava **20 petabytes** de dados por dia.

Isso são **20,480 terabytes** ou **20,971,520 Gigabytes**.

Em 2008, eram produzidos menos de 1 zettabyte por dia.
(1,000,000,000 de terabytes)

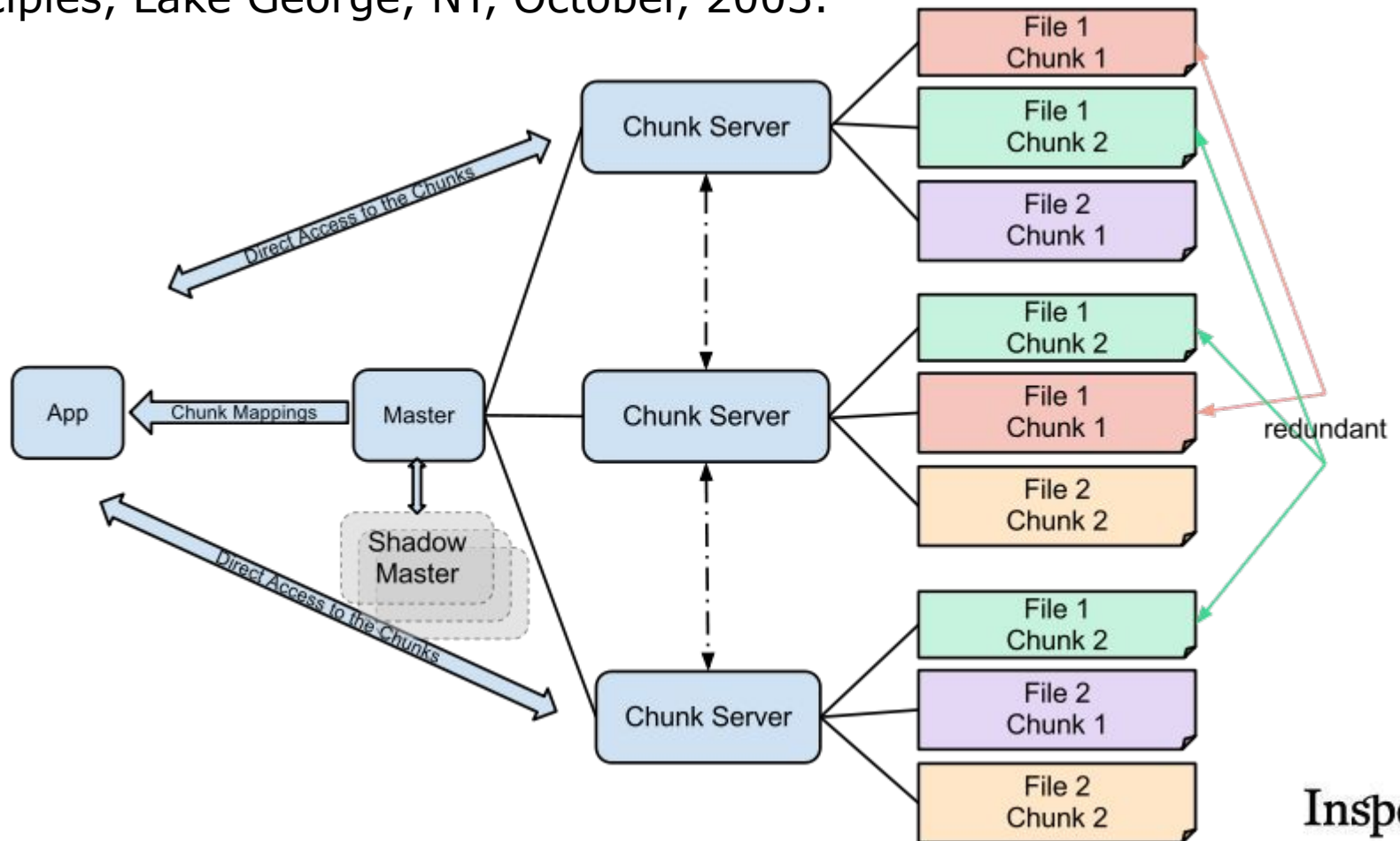
Em 2023, estima-se que são produzidos 120 zettabytes por dia.

Ou seja, **120,000,000,000 terabytes**.

Como processar isso?

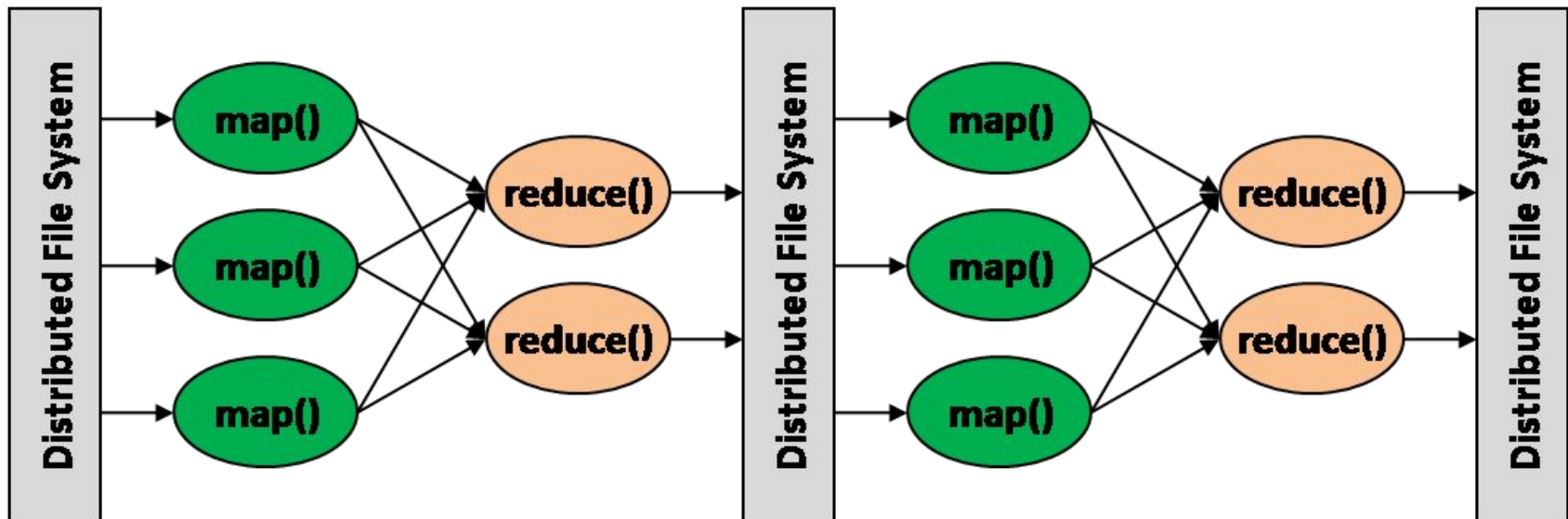
Google file system

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. "The Google File System", 19th ACM Symposium on Operating Systems Principles, Lake George, NY, October, 2003.



MapReduce

Jeffrey Dean and Sanjay Ghemawat. "MapReduce: Simplified Data Processing on Large Clusters", OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December, 2004.



<https://dzone.com/articles/how-hadoop-mapreduce-works>

MapReduce

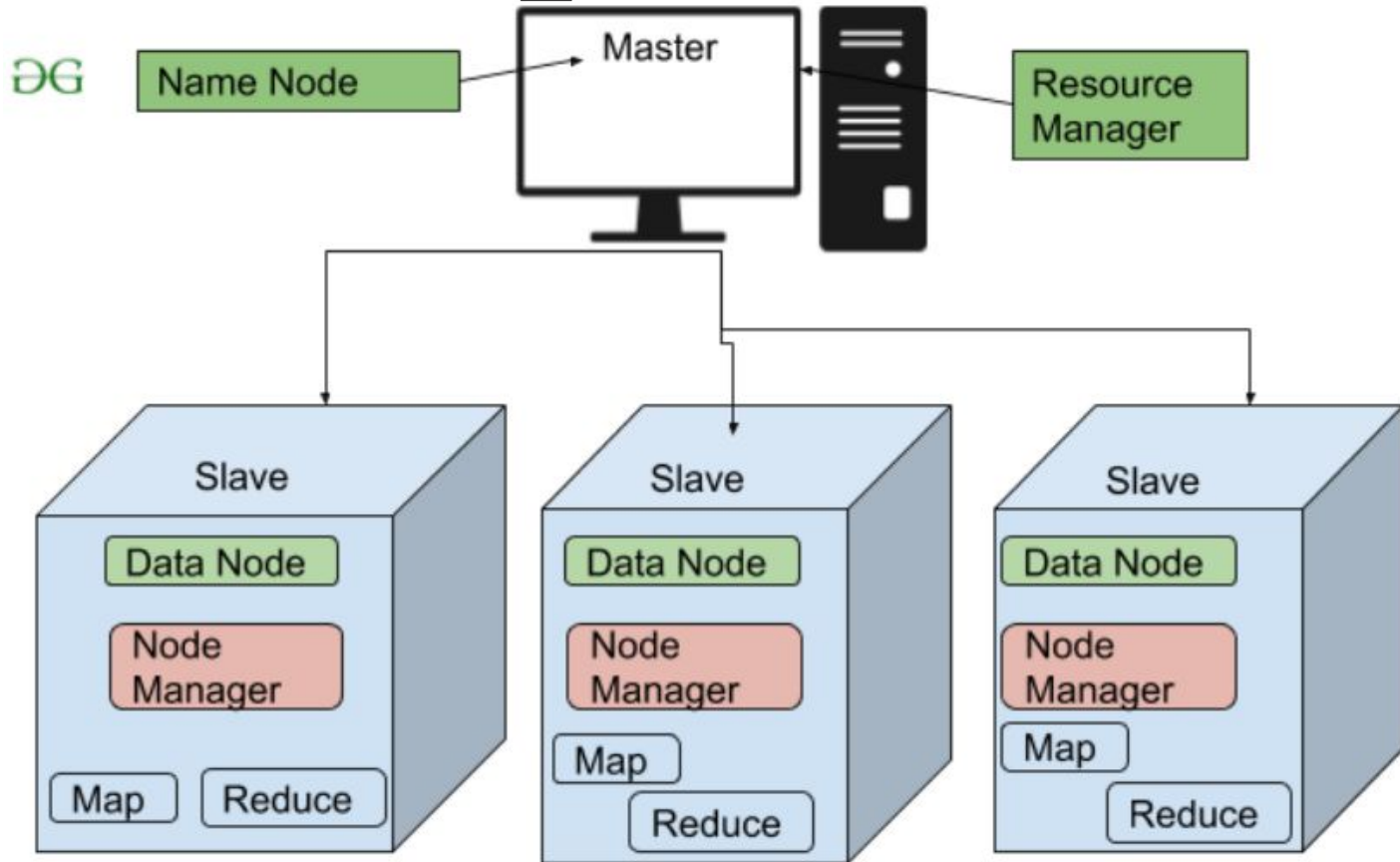
A idéia central do MapReduce é levar a computação até os dados, e não o contrário

- Muito mais eficiente quando a massa de dados é enorme!

MapReduce

```
function map(String name, String document):  
    // name: document name  
    // document: document contents  
    for each word w in document:  
        emit (w, 1)  
  
function reduce(String word, Iterator partialCounts):  
    // word: a word  
    // partialCounts: a list of aggregated partial counts  
    sum = 0  
    for each pc in partialCounts:  
        sum += pc  
    emit (word, sum)
```

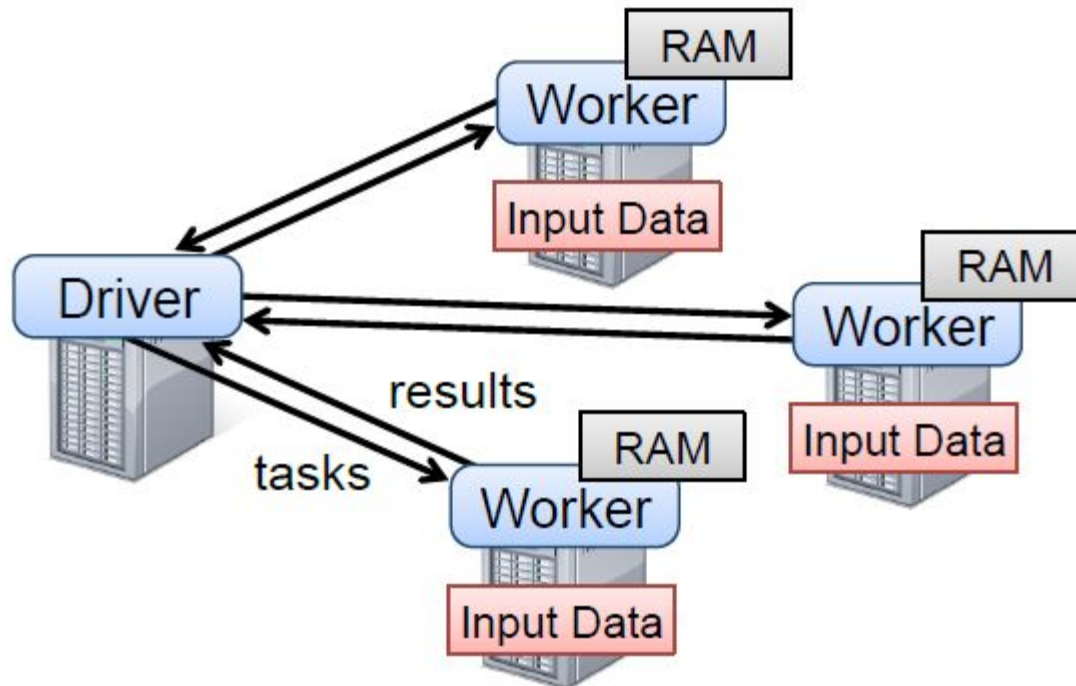
Hadoop



Spark

Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, Ion Stoica. “Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing”, NSDI 2012. April 2012.

<https://spark.apache.org/>



Spark

- Um framework para processamento distribuído
- Roda sobre a Java Virtual Machine
- Implementado em Scala
- Tem interfaces de programação nas linguagens:
 - Java
 - Scala
 - Python
- Baseado no paradigma de programação funcional

Praticando Spark

Abra a documentação de programação do Spark:

<https://spark.apache.org/docs/latest/rdd-programming-guide.html>

O conceito fundamental do Spark é o Resilient Distributed Dataset.

- **Distributed:** O dataset é distribuído em blocos ao longo de várias máquinas
- **Resilient:** Se uma máquina cai, os blocos que estavam sendo calculados nesta máquina são realocados e recalculados em outras máquinas. Ou seja, o sistema é resiliente à falhas.

Transformations and actions

Os comandos em Spark (ou seja, as operações no RDD) se dividem em:

- Transformations: Atuam sobre um RDD e retornam um RDD. Uma transformation não causa nenhuma computação no momento em que ocorre. A computação é escalonada para acontecer depois, no momento das actions. Esta estratégia (de só calcular as coisas no momento em que precisamos dela) é geralmente conhecida como *lazy computing* ou *lazy evaluation*
- Actions: Atuam sobre um RDD e retornam um elemento concreto de dados, ou realizam algum *side effect* final (como gravar os dados resultantes em disco)

Transformations

- map: aplica uma função em todo elemento de um RDD.
 - Mapeamento um para um
- flat_map: aplica uma função em todo elemento de um RDD
 - Mapeamento um para muitos
- reduce: “resume” um RDD em um só número
- reduce_by_key: agrupa os elementos por uma chave e resume todos os elementos da mesma chave.

Insper

www.insper.edu.br