– Computação com grandes volumes de dados

Grandes volumes de dados pertenciam ao domínio das aplicações científicas até há 15 anos atrás.

Com o crescimento exponencial das aplicações Web o volume de dados que é produzido e processado tornou-se imenso.

As empresas que fornecem serviços através da Internet como pesquisa, publicidade, e redes sociais processam diariamente gigantescas quantidades de dados.

– Computação com grandes volumes de dados

Surge o termo **“Big data”**

Coleção de dados tão grande e complexa que é difícil o seu processamento através das aplicações tradicionais.

Big data surge em áreas como:

- weblogs, - aplicações com RFID (radio frequency identification),
- redes de sensores, - redes sociais, - motores de busca,
- serviços de vigilância militar, - registos médicos,
- arquivos de fotos e vídeo, - grandes aplicações comerciais, …

– Computação com grandes volumes de dados

Não há muito tempo, uma aplicação ter 1000 utilizadores por dia era muito e 10,000 era um caso extremo.

Hoje, imensas aplicações estão hospedadas na cloud,
disponíveis pela internet,
suportam utilizadores 24 horas por dia, 365 dias por ano.

Mais de 3 biliões (1) de pessoas estão ligadas à net em todo o nundo

(1) http://www.internetlivestats.com/internet-users/

– Computação com grandes volumes de dados

- O número de utilizadores de uma aplicação é muito difícil de prever.

- Uma nova app pode ir de 0 a um milhão de acessos num só dia.

- Alguns utilizadores acedem um vez e nunca mais voltam;

- Outros acedem várias vezes ao dia.

– Computação com grandes volumes de dados

As tecnologias de **cloud** suportam computação intensiva de dados através de:

- Grande quantidade de instâncias de computação criadas a pedido;
- Sistemas de armazenamento otimizados para guardar grandes blobs (binary large objects);
- Fornecendo frameworks e APIs otimizadas para processar grandes quantidades de dados.

– Computação com grandes volumes de dados

Alguns fornecedores de cloud foram criando tecnologias para processar grandes quantidades de dados:

Hadoop system: Hadoop Distributed File System + Map Reduce;

Google Map Reduce: Google File System + Map Reduce;

…

[Sistema de armazenamento + Sistema de Computação]

Big Data e Bases de Dados

Tradicionalmente as bases de dados relacionais distribuídas foram consideradas a evolução natural para o aumento do volume de dados.

– Computação com grandes volumes de dados

Big Data e Bases de Dados

Uma base de dados distribuída é construída dividindo os dados por várias localizações. São sistemas robustos, com suporte para transações distribuídas, mas são pouco eficientes para:

- enormes quantidades de dados;
- dados não estruturados;
- dados cuja estrutura evolui ao longo do tempo;

– Computação com grandes volumes de dados

Empresas como Google, Facebook, Amazon e LinkedIn concluíram que o modelo relacional de base de dados não servia para suportar as necessidades das suas aplicações.

Surgiram as bases de dados NoSQL

- Modelo de dados mais flexível;

- Capacidade de escalar dinamicamente para suportar um maior número de utilizadores;

- Pesquisas mais eficientes;

– Computação com grandes volumes de dados

A necessidade de novas formas de armazenar grandes quantidades de dados levou ao surgimento de Sistemas de Ficheiros Distribuídos e de novos Sistemas de Bases de Dados.

1 - Sistemas de Ficheiros Distribuídos de Elevado Desempenho

Google File System (GFS)

Infra-estrutura de armazenamento que suporta a execução de aplicações distribuídas.

Desenhado para ser tolerante a falhas, com elevada disponibilidade, para hardware comum e construído em Linux standard.

– Computação com grandes volumes de dados

Google File System (GFS) - O desenho do GFS assumiu que:

- Iria ser construído em hardware comum que por vezes falha;

- O sistema armazena um número razoável de grandes ficheiros; ficheiros com vários GB são comuns e devem ser tratados eficientemente;

- Os workloads consistem principalmente em dois tipos de leitura: large streaming reads; small random reads; Tem também escritas sequenciais que acrescentam dados a ficheiros já existentes;

- Grande e sustentada largura de banda é mais importante que baixa latência;

– Computação com grandes volumes de dados

Google File System (GFS)

A arquitetura do GFS é organizada como:

- Um master que contém os metadados de todo o file system e uma coleção de servidores (chunk servers) que fornecem o espaço de armazenamento; (um ficheiro é dividido em conjuntos de chunks);

- Chunks são replicados pelos vários servidores para tolerar falhas;

- Uma coleção de processos que implementam o master e os chunks servers;

– Computação com grandes volumes de dados

Google File System (GFS)

- Os processos clientes acedem ao master, identificam o servidor que contém os dados que pretendem e comunicam diretamente com o chunk server;

- Aplicações interactuam com o file system através de uma interface que possui as operações comuns de criar, apagar, ler ou escrever no ficheiro, e ainda “snapshots” e “record append”;

- É possível criar cópias do master e de qualquer dos chunk servers.

– Computação com grandes volumes de dados

Amazon Simple Storage Service (S3)

A forma de implementação não é pública.

- Fornece um serviço de armazenamento organizado em buckets.

- Cada bucket pode armazenar vários objectos, cada um identificado univocamente.

- Objetos são identificados por um URL e expostos através de HTTP.

O sistema é publicitado como sendo escalável, tolerante a falhas e com elevada disponibilidade,

– Computação com grandes volumes de dados

NoSQL systems

NoSQL – Not Only SQL – termo criado em 1998 para identificar uma base de dados que não oferecia uma interface SQL para manipular e aceder aos dados.

- Não são sistemas de bases de dados relacionais, mas uma coleção de scripts que permitem executar as operações mais comuns em base de dados, usando ficheiros de texto para guardar a informação.

- A filosofia é ultrapassar as limitações impostas pelo modelo relacional e criar sistemas mais eficientes;

– Computação com grandes volumes de dados

NoSQL systems

Para isso são usadas tabelas sem esquemas fixos, que permitem:

- conter um maior número de tipos de dados;

- evitar operações de junção para aumentar o desempenho e permitir escalar o sistema horizontalmente.

- Existem muito modelos:

- Document stores; Graphs; Key-value stores; Multivalue databases;
- Object databases; Tabular stores; Tuple stores;

– Computação com grandes volumes de dados

Bases de Dados SQL Vs NoSQL

Características de uma Base de Dados SQL:

- Dados são armazenados em tabelas;
- Associações entre os dados são representadas por dados;
- Linguagem de Manipulação de Dados (DML);
- Linguagem de Definição de Dados (DDL);
- Transações;

– Computação com grandes volumes de dados

Bases de Dados SQL Vs NoSQL

SQL

- Aplicações especificam o que querem e não “como querem”.
- Existe um motor de otimização dos queries;
- Abstração do nível físico; a camada física pode mudar sem ser necessário mudar as aplicações;
- Criação de índices para suportar queries;

– Computação com grandes volumes de dados

Bases de Dados SQL Vs NoSQL

SQL

Uma transação deve respeitar as propriedades **ACID**:

Atomic – Todas as operações de uma transação são completadas (commit) ou nenhuma delas.

Consistent – Um transação transforma uma base de dados de um estado consistente em outro estado consistente. A consistência é definida através de restrições de consistência.

– **Computação com grandes volumes de dados**

Bases de Dados SQL Vs NoSQL

SQL - propriedades **ACID**:

Isolated – Os resultados das alterações efetuadas durante uma transação não são visíveis até que a transação termine com sucesso (committed).

Durable – Os resultados de uma transação terminada com sucesso (commited) sobrevivem a falhas..

– Computação com grandes volumes de dados

Bases de Dados SQL Vs NoSQL

NoSQL definição (1):

Next Generation Databases mostly addressing some of the points: being ***non-relational, distributed, open-source*** *and* ***horizontally scalable****.*

The original intention has been **modern web-scale databases**. The movement began early 2009 and is growing rapidly. Often more characteristics apply such as: **schema-free, easy replication support, simple API, eventually consistent** / **BASE** (not ACID), a **huge amount of data**,…

(1) http://www.nosql-database.org/

– **Computação com grandes volumes de dados**

Bases de Dados SQL Vs NoSQL

NoSQL - caraterísticas

- Grandes volumes de dados;

- Replicação e distribuição escaláveis;
  - potencialmente milhares de máquinas;
  - potencialmente distribuída por todo o mundo;

- Respostas rápida aos queries;

- Principalmente queries, poucos updates;

– Computação com grandes volumes de dados

Bases de Dados SQL Vs NoSQL

NoSQL – caraterísticas distintivas

- Sem um esquema fixo;
- Transações não são ACID – são BASE (!)
- CAP teorema;
- Open-Source;

– Computação com grandes volumes de dados

Bases de Dados SQL Vs NoSQL

NoSQL

Transações **BASE** **(oposto de ACID)**

- **B**asically **A**vailable;
- **S**oft state; (o estado do sistema pode mudar ao longo do tempo, mesmo sem input)
- **E**ventually consistent (o sistema vai tornar-se consistente ao longo do tempo)

– Computação com grandes volumes de dados

Bases de Dados SQL Vs NoSQL
NoSQL - caraterísticas

- Consistência fraca;
- Disponibilidade em primeiro lugar;
- “Best effort”;
- Respostas aproximadas – OK;
- Agressivo (Otimista);
- Fácil e rápido;

– Computação com grandes volumes de dados

Bases de Dados SQL Vs NoSQL

NoSQL - caraterísticas

CAP teorema (Brewer, 2000)
(na verdade, não um teorema mas uma conjetura)

Um sistema distribuído pode apenas suportar duas das seguintes características:

- **Consistência;**
- **Disponibilidade;**
- **Tolerância a partições;**

– Computação com grandes volumes de dados

Bases de Dados SQL Vs NoSQL

NoSQL - CAP teorema

- **Consistência;**

- Todos os nós vêm os mesmos dados ao mesmo tempo;

- O cliente percebe que um conjunto de operações ocorreu como um todo;

(equivale ao conceito de atomicidade nas transações ACID)

– Computação com grandes volumes de dados

Bases de Dados SQL Vs NoSQL

NoSQL - CAP teorema

- Disponibilidade;

- A avaria de um nó não faz com que os outros deixem de funcionar.
- Cada operação deve terminar e dar a resposta pretendida.

– Computação com grandes volumes de dados

Bases de Dados SQL Vs NoSQL

NoSQL - CAP teorema

- Tolerância a partições;

- O sistema continua mesmo que haja mensagens perdidas;
- As operações serão concluídas mesmo que algum componente individual não esteja disponível.

– Computação com grandes volumes de dados

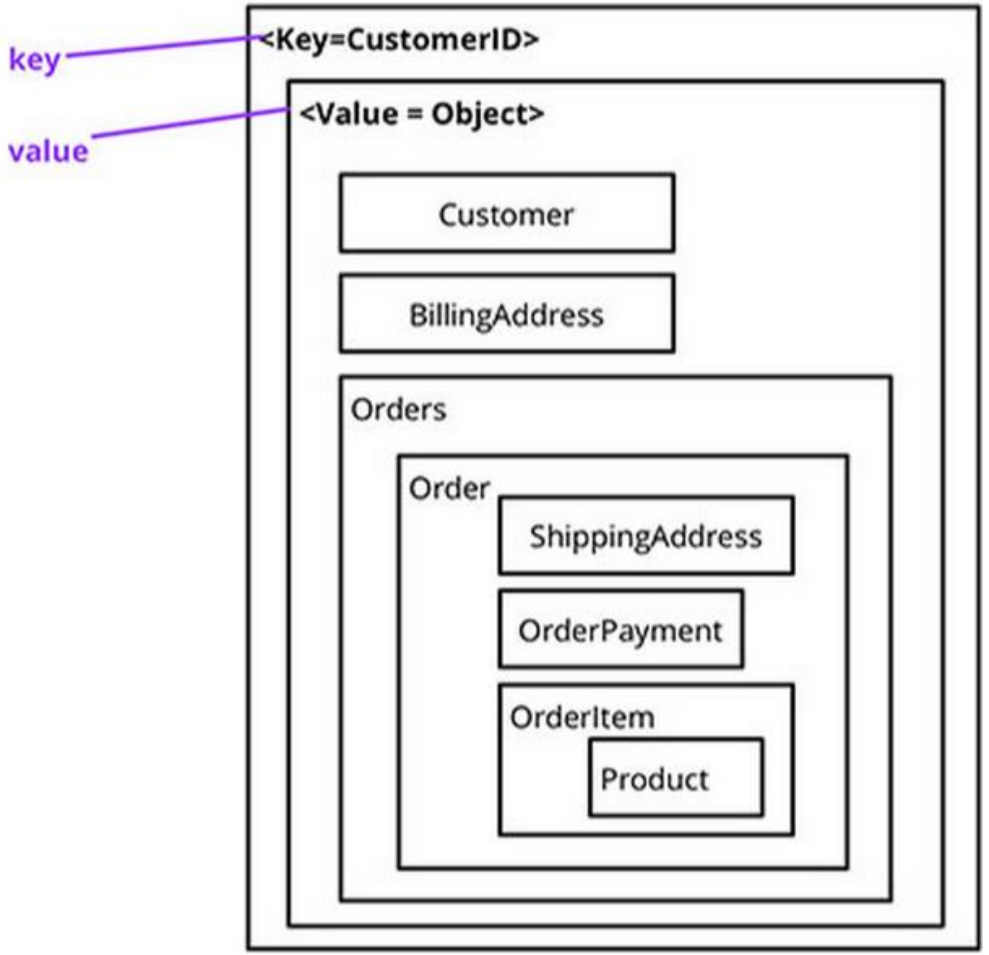
NoSQL – tipos de Bases de Dados NoSQL mais importantes

- **Key-Value databases**
- **Document databases**
- **Column family stores**
- **Graph Databases**
- **…**

(1) http://www.thoughtworks.com/insights/blog/nosql-databases-overview

– Computação com grandes volumes de dados

NoSQL - Key-Value databases



Modelo NoSQL
mais simples de usar.

O cliente pode:
- obter o valor para uma chave:
- pode atribuir um valor para uma chave;
- apagar uma chave da base de dados;

– Computação com grandes volumes de dados

NoSQL - Key-Value databases

O valor é um blob armazenado na base de dados;
O sistema de BD não conhece o que está dentro do blob;
É da responsabilidade da aplicação conhecer o que foi armazenado.

Os valores são acedidos pela única chave (primária) o acesso é rápido e facilmente escalável.

Exemplos de sistemas de bases de dados key-value:

Riak, Redis, Memcached, Berkeley DB, HamsterDB, Couchbase;

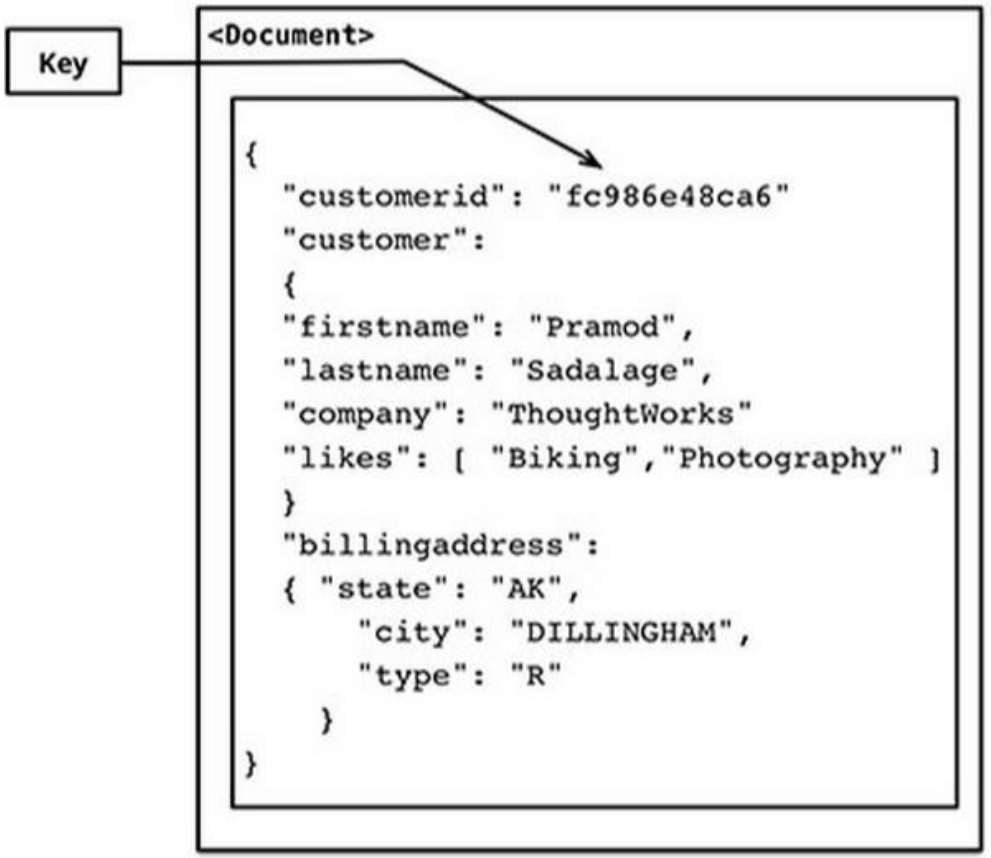
– Computação com grandes volumes de dados

NoSQL - Document databases

- O Documento é o conceito base destas bases de dados.

- A BD armazena e permite aceder a documentos.

- Os documentos são armazenados em XML, JSON, BSON, …



– Computação com grandes volumes de dados

NoSQL - Document databases

- Os documentos são auto-descritivos. Os documentos são similares mas não necessariamente iguais.

- Os documentos são conjuntos de elementos etiquetados (tagged) e que podem ser pesquisados.

- Cada documento tem um campo chave;

- Podem ser criados índices para acelerar as pesquisas.

– Computação com grandes volumes de dados

NoSQL - Document databases

Exemplos:

- MongoDB;
- CouchDB;
- Terrastore;
- OrientDB;
- RavenDB;

– **Computação com grandes volumes de dados**

NoSQL - Column family databases

- Cada bloco de armazenamento contém dados de uma coluna de uma tabela/registo/documento.

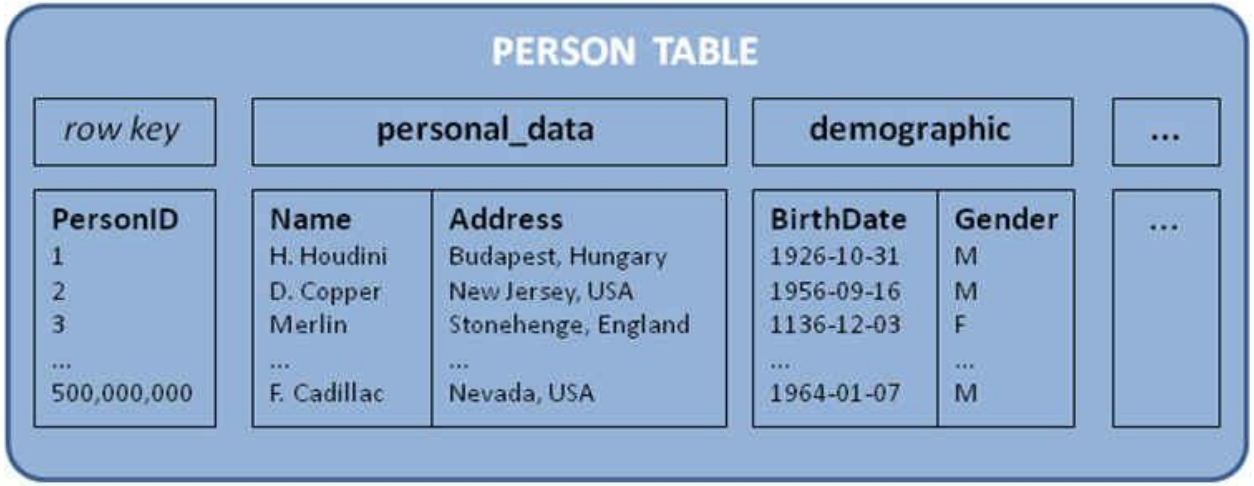


Figure 2 - Census Data in Column Families

– Computação com grandes volumes de dados

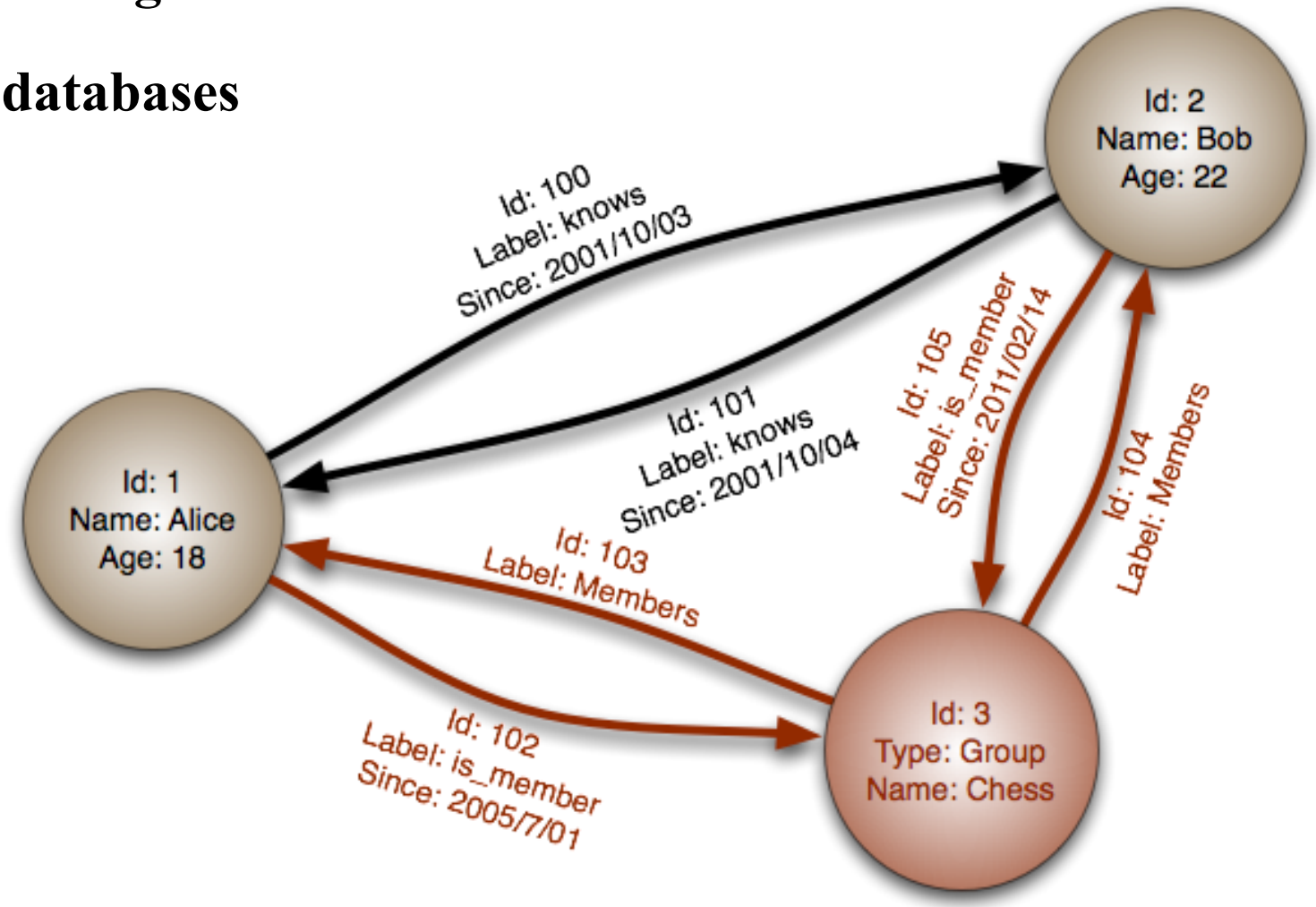
NoSQL - Column databases

- Mais eficiente se múltiplas linhas (ou documentos) são armazenadas em simultâneo, assim as atualizações podem ser agregadas.

- Exemplos:
  - Cassandra (usado por Facebook, Twitter)
  - Hbase;
  - Hypertable (baseado no modelo da Google Big Table);
  - Amazon DynamoDB (não é open source);

– **Computação com grandes volumes de dados**

NoSQL - Graph databases



– Computação com grandes volumes de dados

NoSQL - Graph databases

- Entidades são nós de um grafo.
- Associações entre entidades são representadas pelas arestas.

- Entidades e associações podem ter propriedades.

- A organização do grafo permite que seja interpretado de diferentes formas baseadas nas associações entre os nós.

Exemplos: Neo4J, Infinite Graph, OrientDB;

– Computação com grandes volumes de dados

SQL Vs NoSQL - escalabilidade

Para enfrentar o aumento do número de utilizadores e o aumento do volume de dados podem considerar-se duas hipóteses.

Scale up – usar servidores cada vez maiores (solução centralizada)

Scale out – distribuir os dados e a computação por múltiplos servidores

– Computação com grandes volumes de dados

SQL Vs NoSQL - escalabilidade

O **modelo relacional** por omissão escala usando servidores cada vez mais poderosos. Isso implica planear, dimensionar, adquirir o servidor e eventualmente parar a aplicação para ser migrada. (**Scale up**)

Bases de dados **NoSQL** foram desenhadas para ser distribuídas e tolerantes a falhas. São open source, usam máquinas comuns (mais baratas).

Se o volume de dados aumenta acrescentam-se servidores.

– Computação com grandes volumes de dados

NoSQL - escalabilidade e desempenho

Auto-sharding

Uma base de dados NoSQL automaticamente divide os seus dados por vários servidores sem parar a aplicação.

Servidores podem ser adicionados ou removidos

Muitos sistemas NoSQL replicam os dados pelos servidores para garantir a disponibilidade e tolerância a falhas.

– Computação com grandes volumes de dados

NoSQL - escalabilidade e desempenho

Distributed query support

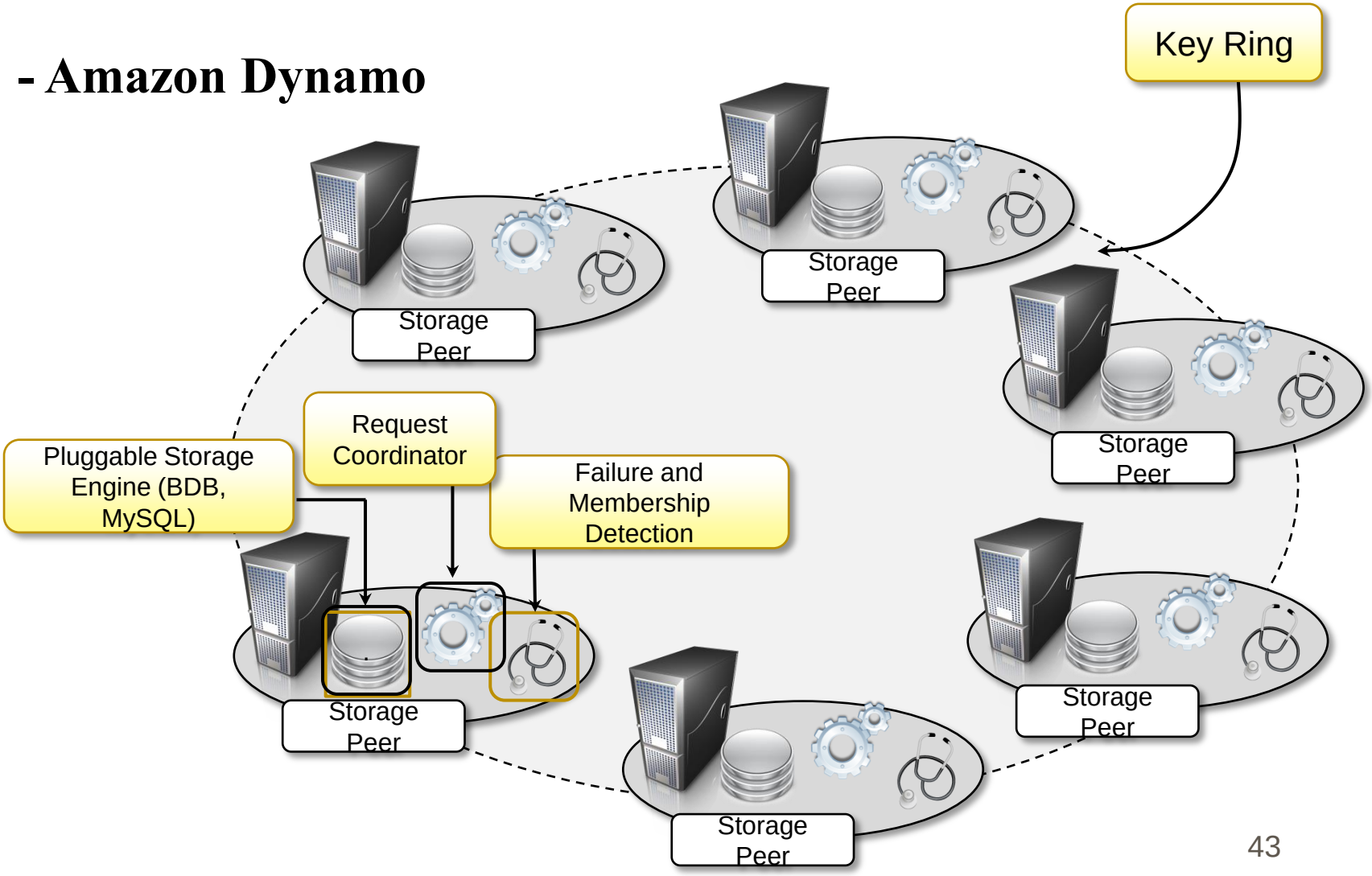
Sistemas NoSQL fornecem suporte para queries distribuídos

Integrated caching

Para aumentar o desempenho, colocam os dados em memória de forma transparente para a aplicação.

– Computação com grandes volumes de dados

NoSQL - Amazon Dynamo



– Computação com grandes volumes de dados

NoSQL - Amazon Dynamo

- Milhares de servidores servem 10 milhões de acessos por dia;

- Interface simplificada baseada em operações de get/put;

- Objetos são armazenados com um identificador único;

- Eventualmente consistente (não ACID);
(todos os utilizadores verão os mesmos dados a longo prazo)

- Coleção de "storage peers" organizados em anel.

– Computação com grandes volumes de dados

NoSQL - Amazon Dynamo

- Não há restrições de integridade;
- Não há associações embutidas na base de dados;
- Operações de “join” não são suportadas;
- A aplicação constrói o seu modelo de dados sobre a estrutura disponibilizada.

– Computação com grandes volumes de dados

2 – Modelo de programação Map-Reduce

Plataformas para programação de aplicações com computação intensiva de dados fornecem

➔ abstrações para exprimir computações que processem grandes volumes de dados,

´

➔ passam para o sistema de execução a gestão das transferências de dados.

– Computação com grandes volumes de dados

2 – Modelo de programação Map-Reduce

A plataforma MapReduce introduzida pela Google exprime a computação com duas primitivas:

map e *reduce.*

A transferência e gestão dos dados é feita pelo infraestrututa de armazenamento (Google file system no caso da Google)

– Computação com grandes volumes de dados

O modelo MapReduce é expresso na forma de duas funções:

map (k1, v1) → list (k2, v2)

reduce (k2, list (v2)) → list (v2)

Exemplo: procurar a ocorrência de palavras em documentos:

A função **map** lê um par key-value e produz uma lista de pares key-value de tipo diferente.

*- a função **map** para cada documento <documentoID, documento> gera uma lista com <palavra, documentoID>*

– Computação com grandes volumes de dados

O modelo MapReduce é expresso na forma de duas funções:

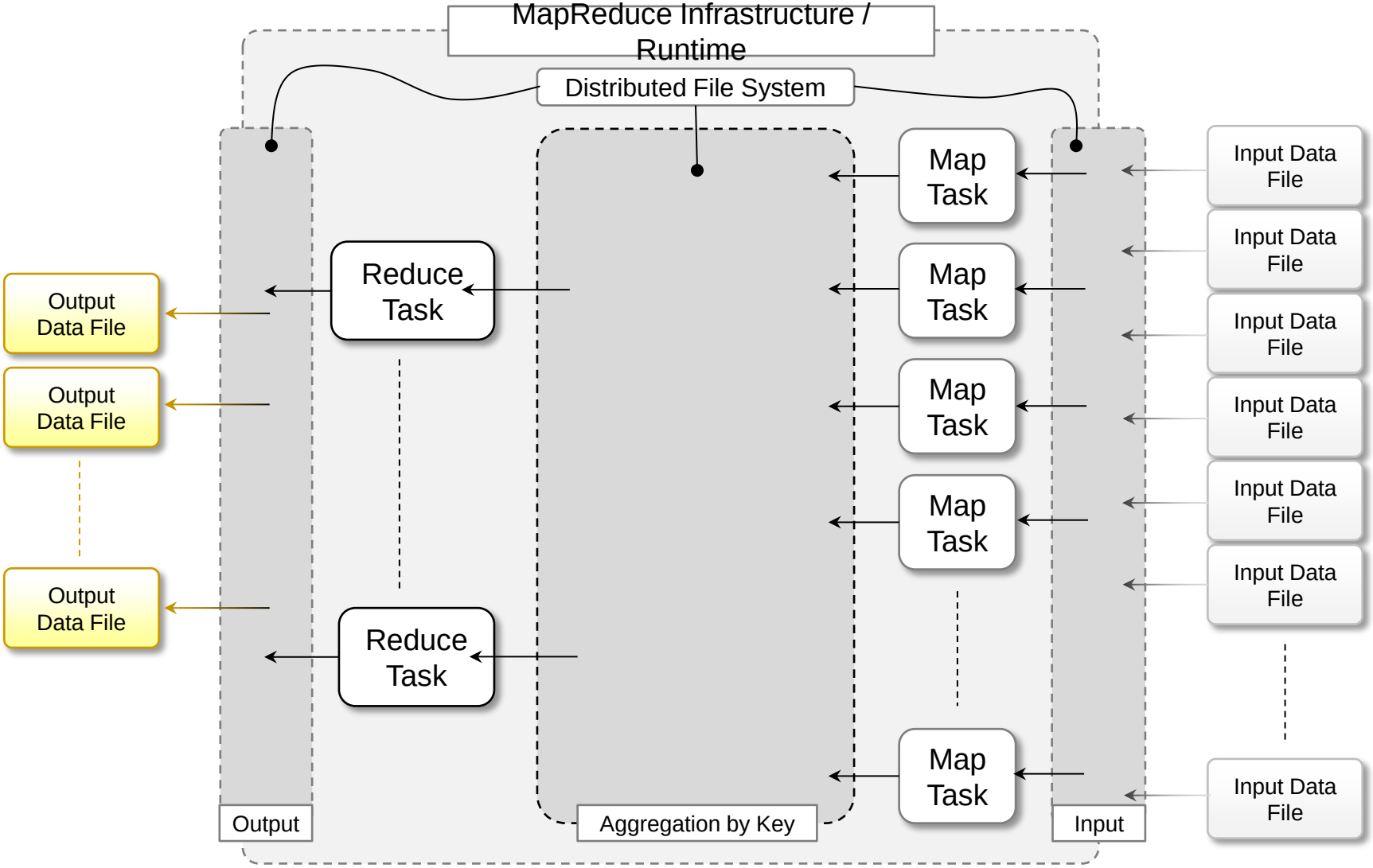
map (k1, v1) → list (k2, v2)

reduce (k2, list (v2)) → list (v2)

A função **reduce** lê um par composto por uma key e uma lista de valores e produz uma lista de valores do mesmo tipo.

- a função **reduce** vai agregar as ocorrências da mesma palavra, **<palavra, list (documentoID>** produzindo a lista de documentos onde aparece cada palavra.

Sistemas Distribuídos



– Computação com grandes volumes de dados

O utilizador submete uma coleção de ficheiros expressos na forma de uma lista de pares <key, value> e define as funções map e reduce.

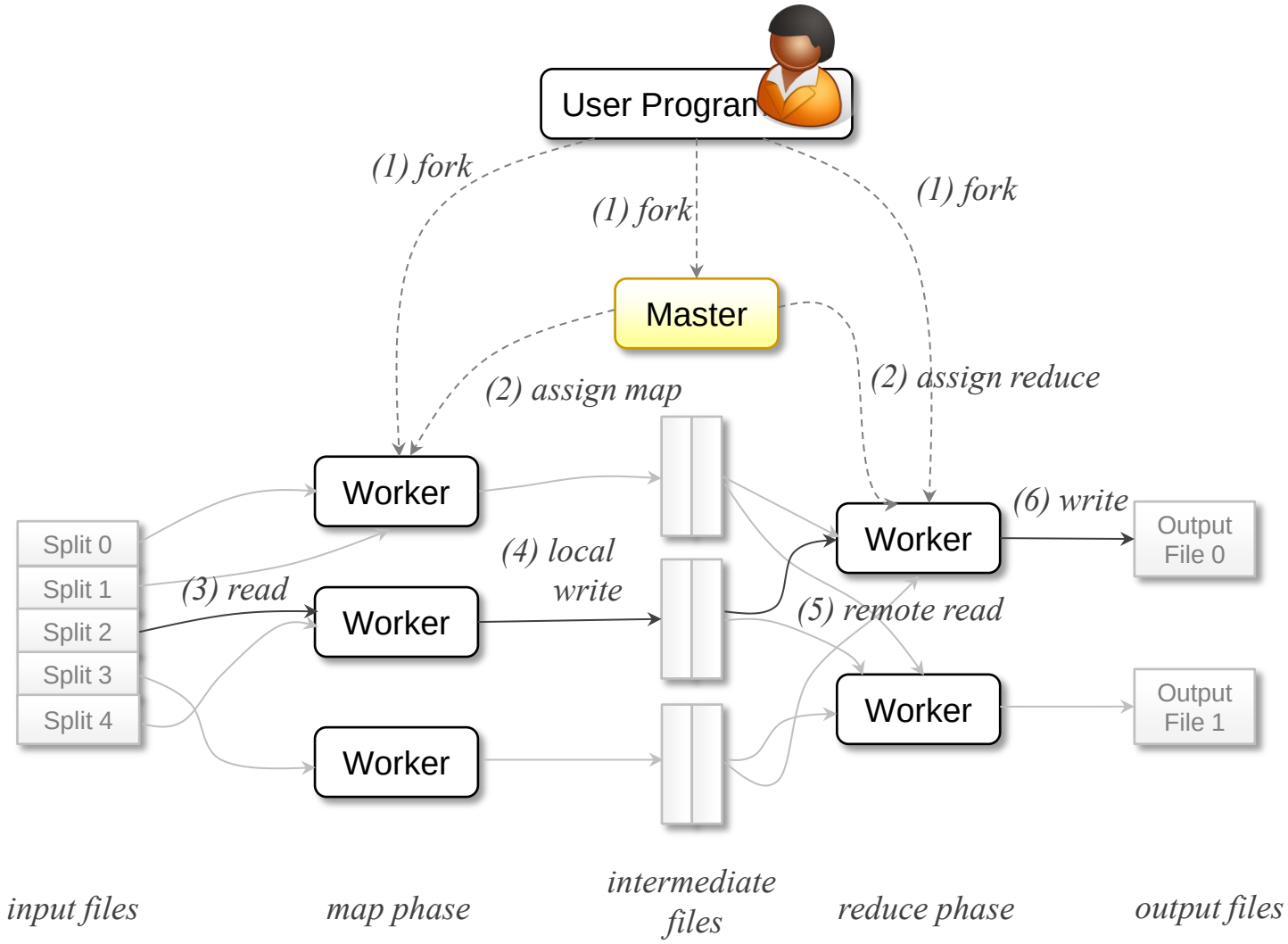
O sistema de suporte ao MapReduce usa os ficheiro, eventualmente depois de divididos, como input da função Map.

Os processos que executam funções Map geram ficheiros intermédios que contêm listas na forma de pares <k2, list(v2)).

Ficheiros com a mesma chave podem eventualmente ser agregados.

Estes ficheiros serão o input para os processos que executam a função Reduce, que irão produzir ficheiros de output na forma list(v2).

Google MapReduce infraestrutura



– Computação com grandes volumes de dados

Dois tipos de processo executam numa infraestrutura de MapReduce:

O Master e os Workers.

O master controla a execução das funções Map e Reduce, particionando os dados, reorganizando os resultados intermédios, da função Map, para input da função Reduce.

Os workers executam as funções de Map e Reduce.

Os utilizadores podem configurar o número de processos Map e Reduce, mas todo o processo de transferência de dados é da responsabilidade da plataforma de execução.

11 – Computação com grandes volumes de dados

Map-Reduce

Adicionalmente a plataforma assegura uma execução fiável.

Falhas nos processo workers são tratadas redirecionando o trabalho correspondente para outra máquinas.

Falhas no master são tratadas usando checkpoints periódicos.