

Introdução aos Bancos de Dados Não-Relacionais

Mauricio De Diana (mestrando)
Marco Aurélio Gerosa (orientador)

Conteúdo

Histórico de BDs não-relacionais na Web

4 Categorias de bancos NoSQL

Exemplos de formatos e acesso a dados de cada categoria

Modelo relacional

Base na lógica e na matemática

Relações

Domínios

Operações (σ , π , \bowtie)

Restrições

Bancos não-relacionais

Bancos hierárquicos

Bancos orientados a objetos

Bancos de XML

Etc

Bancos não-relacionais

Bancos hierárquicos

Bancos orientados a objetos

Bancos de XML

Etc

Ou seja, não-relacional não define muito

Web 2.0

Data is the Next Intel Inside

Inteligência Coletiva

Grande volume de dados

Escala global (*Internet scale services*)

Web 2.0

Alto grau de paralelização

Hardware *commodity* e barato

Software livre, código aberto

Dados semi-estruturados e crús

Soluções para dados

Bigtable (Google)

Dynamo (Amazon)

PNUTS (Yahoo!)

Além de:

GFS, Hadoop, HDFS, MapReduce, Chubby, etc

Soluções para dados

Bigtable (Google)

Dynamo (Amazon)

PNUTS (Yahoo!)

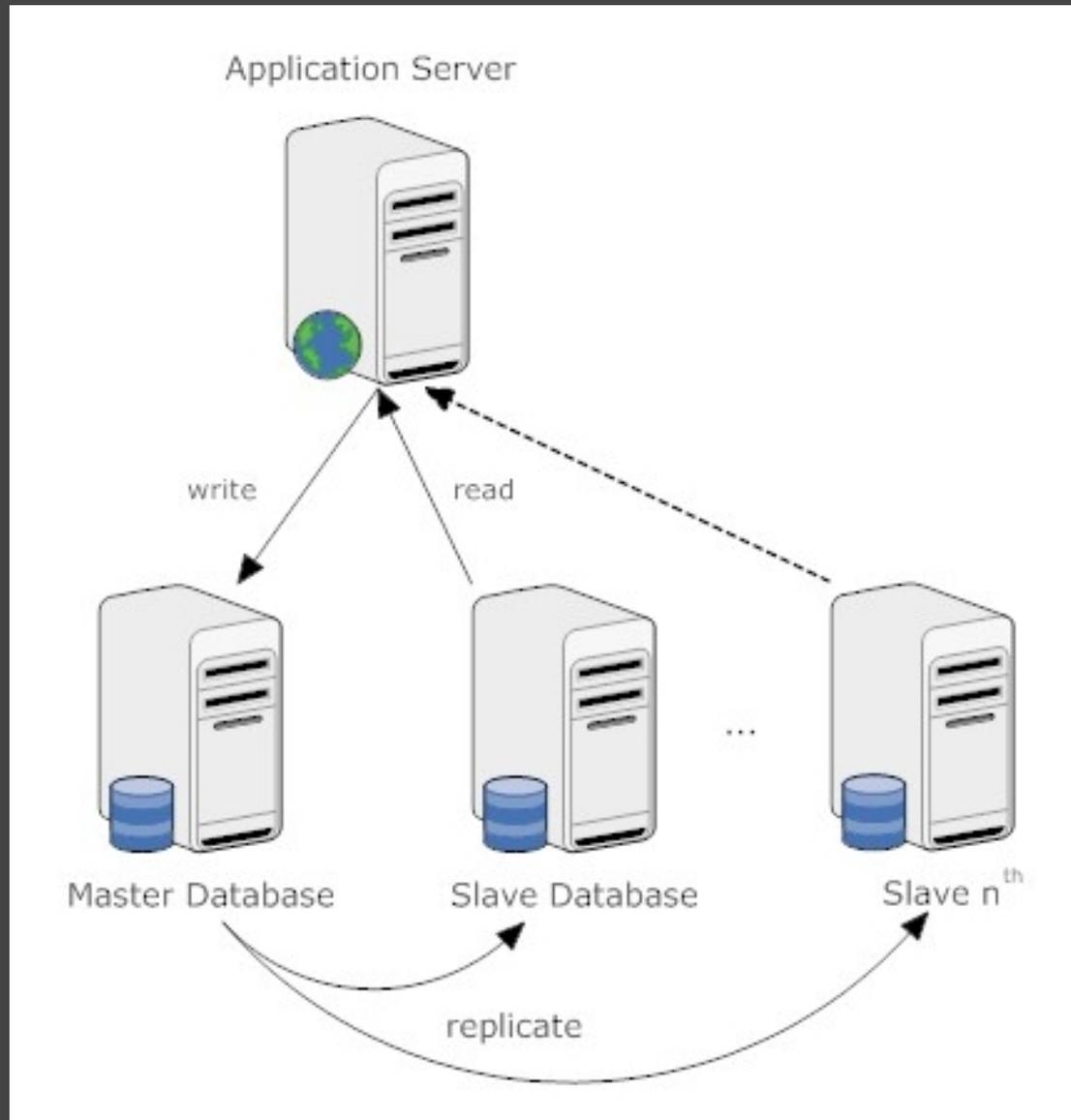
Além de:

GFS, Hadoop, HDFS, MapReduce, Chubby, etc

Enquanto isso, nas startups...

Escalando BDs relacionais

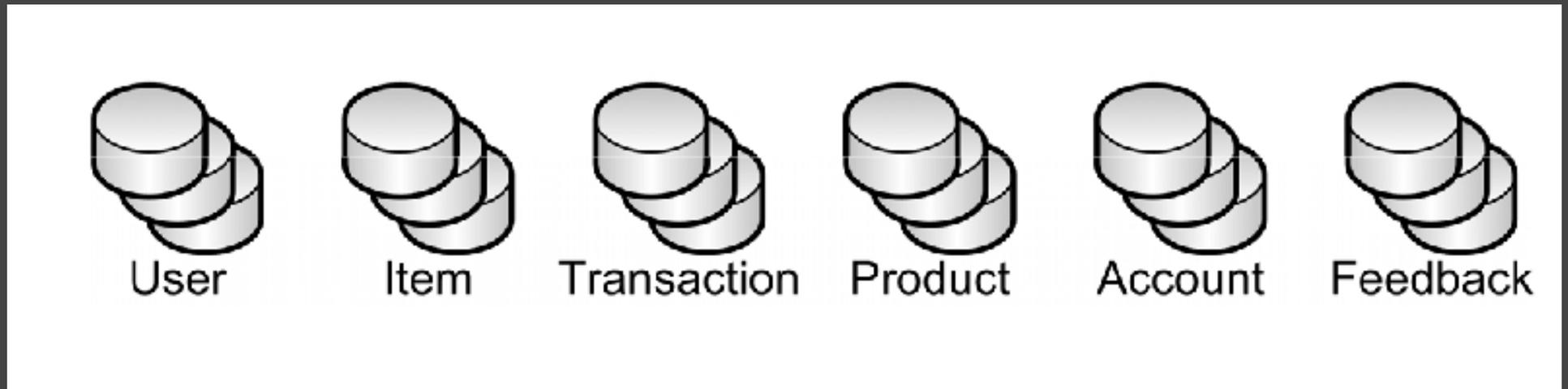
Master-slave



Fonte: <http://www.teonator.net/2008/10/23/mysql-replication/>

Escalando BDs relacionais

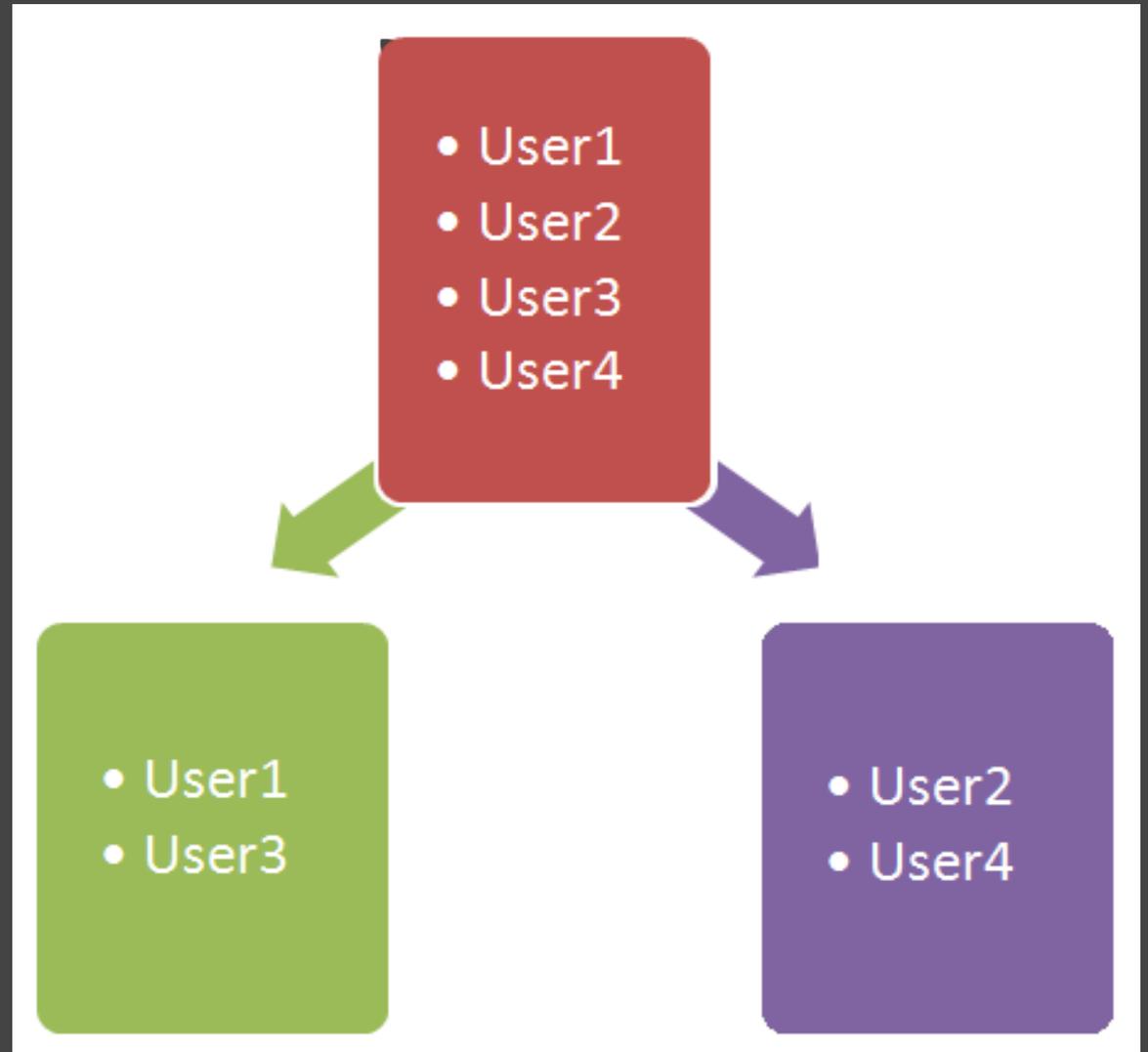
Particionamento funcional



Fonte: http://qconlondon.com/london-2008/file?path=/qcon-london-2008/slides/RandyShoup_eBaysArchitecturalPrinciples.pdf

Escalando BDs relacionais

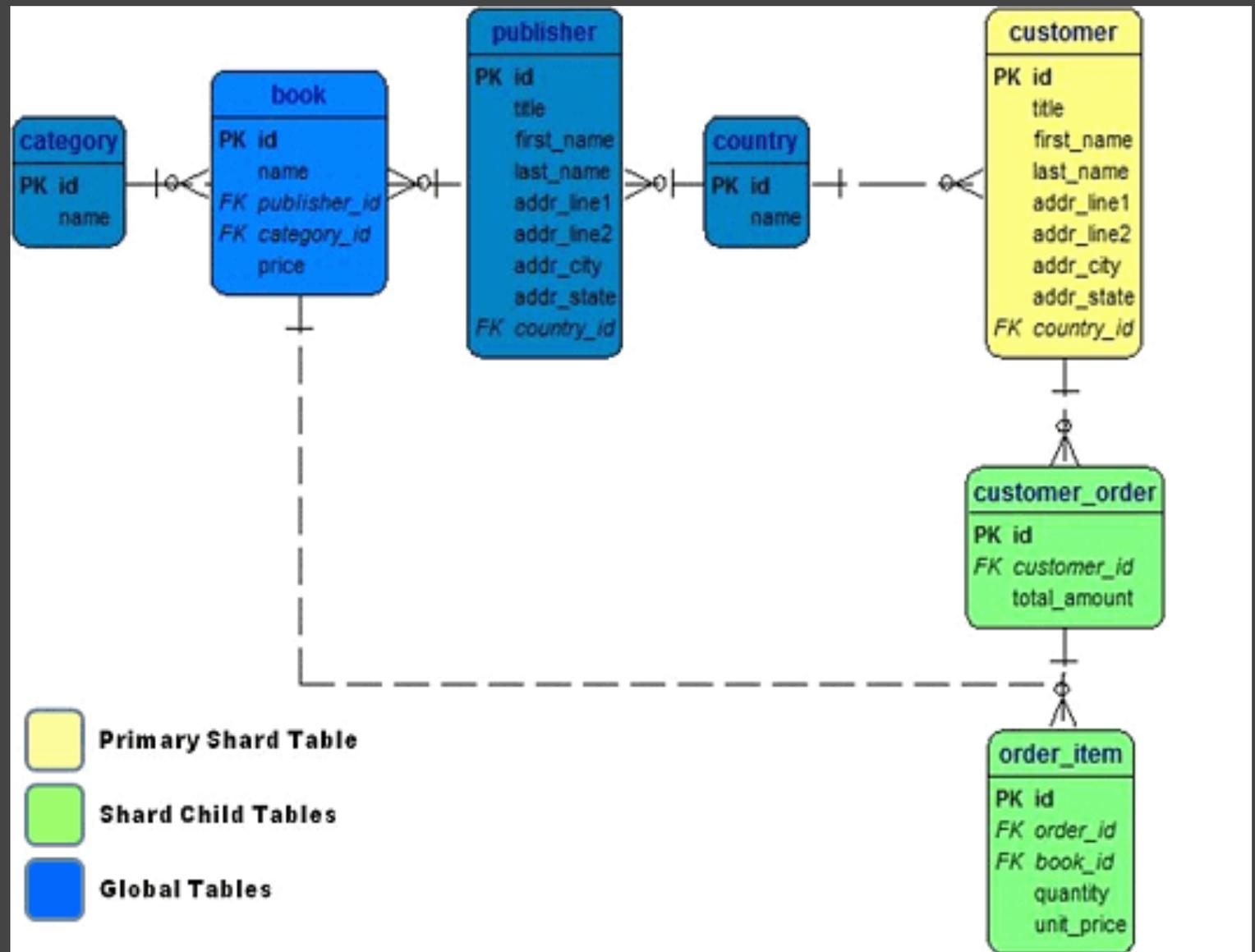
Sharding



Fonte: <http://itsfrosty.wordpress.com/2009/03/20/database-sharding-basics/>

Escalando BDs relacionais

Sharding



Fonte: <http://www.codefutures.com/database-sharding/>

Sharding - Problemas

Queries distribuídas

Cross-shard joins

Auto-increment

Integridade referencial

Lógica de *sharding* na camada de aplicação
(Opção: MySQL Proxy)

NoSQL

Definição fraca:

Bancos não-relacionais distribuídos muito usados na Web

"No SQL" x "Not only SQL"

NoSQL

Escalabilidade horizontal

Sem esquema

Replicação simples

API simples

Software livre / Código aberto

Consistência eventual

ACID x BASE

ACID

Atômico

Consistente

Isolado

Durável

BASE

Basicamente disponível
(Basically available)

Soft-state

Eventualmente consistente

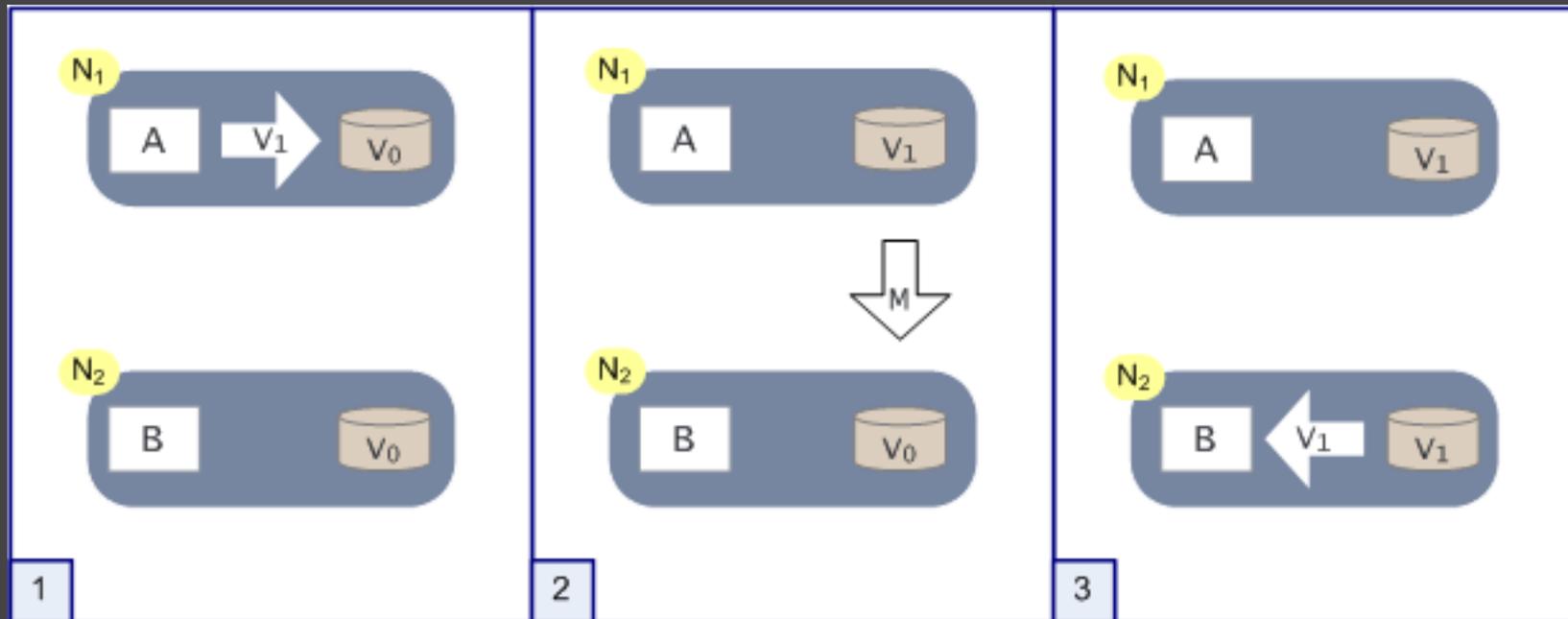
Teorema CAP

Consistência

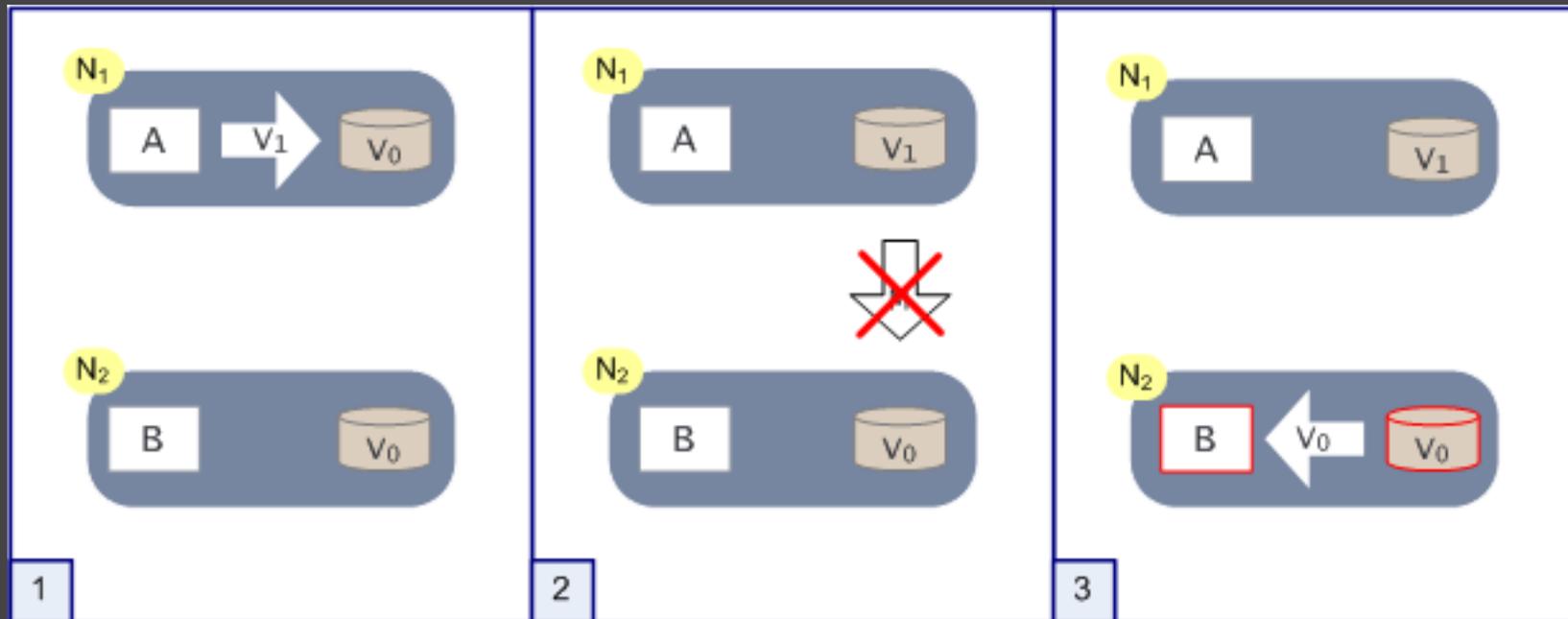
Disponibilidade (*Availability*)

Tolerância à partição (*Partition Tolerance*)

Cenário 1



Cenário 2



Fonte: <http://www.julianbrowne.com/article/viewer/brewers-cap-theorem>

NoSQL - Principais categorias

Bancos de dados orientados a documentos
(Document-oriented databases)

Armazéns chave-valor
(Key-value stores)

Bancos de dados orientados a colunas
(Column-oriented databases)

Bancos de dados de grafos
(Graph databases)

BDs orientados a documentos

Armazenam coleções de pares de chave-valor

Esquema flexível

Formatos: JSON, XML, propriedades, etc

Sharding simples: documentos sem referências explícitas

Exemplos: MongoDB, CouchDB, Riak

Documento (JSON)

```
{
  autor: 'joao',
  criado : new Date('03-28-2009'),
  titulo : 'Primeiro post',
  texto : 'Aqui está o texto...',
  tags : [ 'exemplo', 'joao' ],
  comentarios : [ { autor: 'edu', comentario: 'Ruim' },
                  { autor: 'maria', comentario: 'Legal' } ]
}
```

Documento (JSON)

```
{
  autor: 'joao',
  criado : new Date('03-28-2009'),
  titulo : 'Primeiro post',
  texto : 'Aqui está o texto...',
  modificado: new Date('05-06-2009'),
  modificado por: 'jose',
  tags : [ 'exemplo', 'joao', 'jose' ],
  comentarios : [ { autor: 'edu', comentario: 'Ruim' },
                  { autor: 'maria', comentario: 'Legal' } ]
}
```

Acesso

MongoDB (exemplo)

```
> j = { name : "mongo" };  
{ "name" : "mongo" }
```

```
> t = { x : 3 };  
{ "x" : 3 }
```

```
> db.things.save(j);
```

```
> db.things.save(t);
```

```
> db.things.find();
```

```
{ "name" : "mongo" ,  
  "_id" : ObjectId("497cf60751712cf7758fbdbb") }  
{ "x" : 3 , "_id" : ObjectId("497cf61651712cf7758fbdbc") }
```

Acesso

MongoDB (exemplo)

```
> db.things.find({name:"mongo"}).forEach(  
    function(x) { print(tojson(x));});  
{ "name" : "mongo" ,  
  "_id" : ObjectId("497cf60751712cf7758fbdbb") }  
  
> db.things.find({x:3}).forEach(  
    function(x) { print(tojson(x));});  
{ "x" : 3 , "_id" : ObjectId("497cf61651712cf7758fbdbc") }
```

Fonte: <http://www.mongodb.org/display/DOCS/Tutorial>

Armazéns chave-valor

Tabelas de hash distribuídas

Exemplos: Dynamo, Redis, Voldemort

Acesso: get / put

BDs orientados a colunas

Armazenam dados por colunas, não linhas

Mais eficiência manipulando todos os registros (OLAP)

Linhas não têm a mesma quantidade de colunas

Esquema flexível

Exemplos: Bigtable, Cassandra, HBase

Formato

EmpId, Sobrenome, Nome, Salário

Orientado a linha:

```
1, Silva, João, 600; 2, Dias, Maria, 500; 3, Matos, José, 440;
```

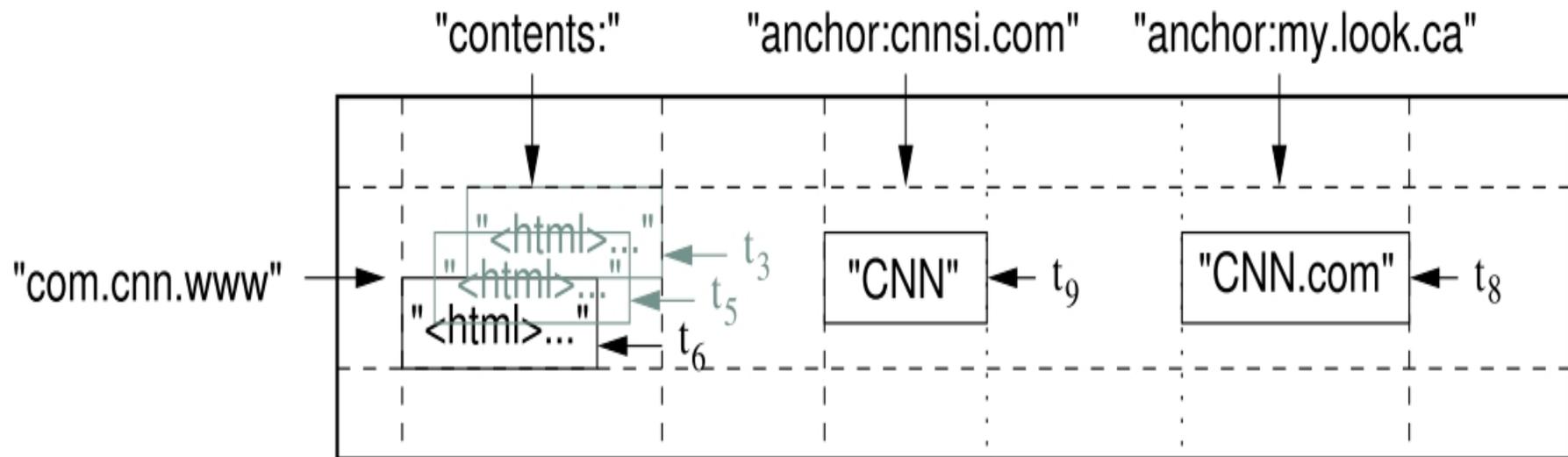
Orientado a coluna:

```
1, 2, 3; Silva, Dias, Matos; João, Maria, José; 600, 500, 440;
```

Acesso

Bigtable (exemplo)

Linha, coluna (família de coluna, qualificador), *timestamp*



Fonte: Bigtable: A Distributed Storage System for Structured Data

BDs de grafos

Armazenam vértices e arestas (e suas propriedades)

Representam interconectividade entre os dados

Operações sobre grafos: percorrer caminho

Exemplos: Infogrid, Neo4J

Acesso

Infogrid (exemplo)

```
customer.traverseToNeighbors().getSingleMember();  
order.traverseToNeighbors().getSingleMember();  
  
customer.unrelate(order);
```

Fonte: <http://infogrid.org/blog/2010/02/operations-on-a-graph-database-part-2/>

Tendências

Release 2.0 (fev/09)

Claremont Report (jun/09)

ThoughtWorks Radar (abr/10)

Conclusões

Dois focos:

Performance / Escalabilidade

Simplicidade

Ferramenta mais adequada para o problema

Futuro incerto (mas a volta ao modelo relacional é improvável)

Referências

<http://nosql-database.org/>

<http://nosql.mypopescu.com/>

Varley, Ian Thomas. No Relation: The Mixed Blessings of Non-Relational Databases (2009).

Orend, Kei. Analysis and Classification of NoSQL Databases and Evaluation of their Ability to Replace an Object-relational Persistence (2010).

Dúvidas?

Obrigado

mmediana@ime.usp.br