



Engenharia de software – 2011A

Trabalho sobre

NOSQL – Not only SQL

GRUPO - 9

- **Cléverton Heming**
- **Jardel Palagi**
- **Jonatam Gebing**
- **Marcos Wassem**

NOSQL

O Termo **NoSQL**, foi utilizado pela primeira vez em 1998, por **Carlo Strozzi**, como nome de seu SGBD – Sistema de gerenciamento de banco de dados, baseado no Modelo Relacional, sem interface SQL.



Início

- **NoSQL** são diferentes sistemas de armazenamento que vieram para suprir necessidades em demandas onde os bancos de dados tradicionais (relacionais) são ineficazes.
- Muitas dessas bases apresentam características muito interessantes como **alta performance**, **escalabilidade**, **replicação**, suporte à dados **estruturados** e **sub colunas**.

- Os atuais bancos de dados relacionais são muito restritos em relação à **escalabilidade**, pois utilizam **escalonamento vertical**. Nos servidores, quanto mais dados, mais espaço no servidor e memória necessita.
- A diferença do **NoSQL**, é que utiliza **escalonamento horizontal** e tem grande facilidade de distribuição do dados, ou seja, com um aumento de dados, aumenta-se o número de servidores, que podem ser ou não de alta performance, barateando e otimizando o armazenamento.

- Outra vantagem da utilização do **escalonamento horizontal**, é que com a divisão dos dados em vários servidores, o volume dos dados por servidor é minimizado, pois, conjuntos de dados menores são mais fáceis de serem **processados**, **armazenados** ou **gerenciados**.

BD Relacionais X BD NoSQL

	BD Relacional	BD NoSQL
Escalabilidade	<ul style="list-style-type: none">• Devido à estruturação do modelo, é possível, mas complexo.	<ul style="list-style-type: none">• É sem dúvida a principal vantagem do NoSQL.• Possui mais flexibilidade na inclusão de dados porem não possui estrutura.
Consistência	<ul style="list-style-type: none">• Ponto forte do Modelo Relacional.• A regra de consistência presente garante um maior rigor à consistência das informações.	<ul style="list-style-type: none">• Não garante a consistência da informação, caso nenhuma informação seja atualizada, retornará a todos os pedintes o mesmo valor.
Disponibilidade	<ul style="list-style-type: none">• Este modelo não suporta eficientemente grande demanda, dado a dificuldade e distribuição de dados.	<ul style="list-style-type: none">• Ponto forte do BD NoSQL.• Possui um alto grau de distribuição de dados, e garante um maior número de solicitações.

Técnicas de banco No-SQL

Existem técnicas que podem ajudar a melhorar a escalabilidade de bancos de dados NoSQL.

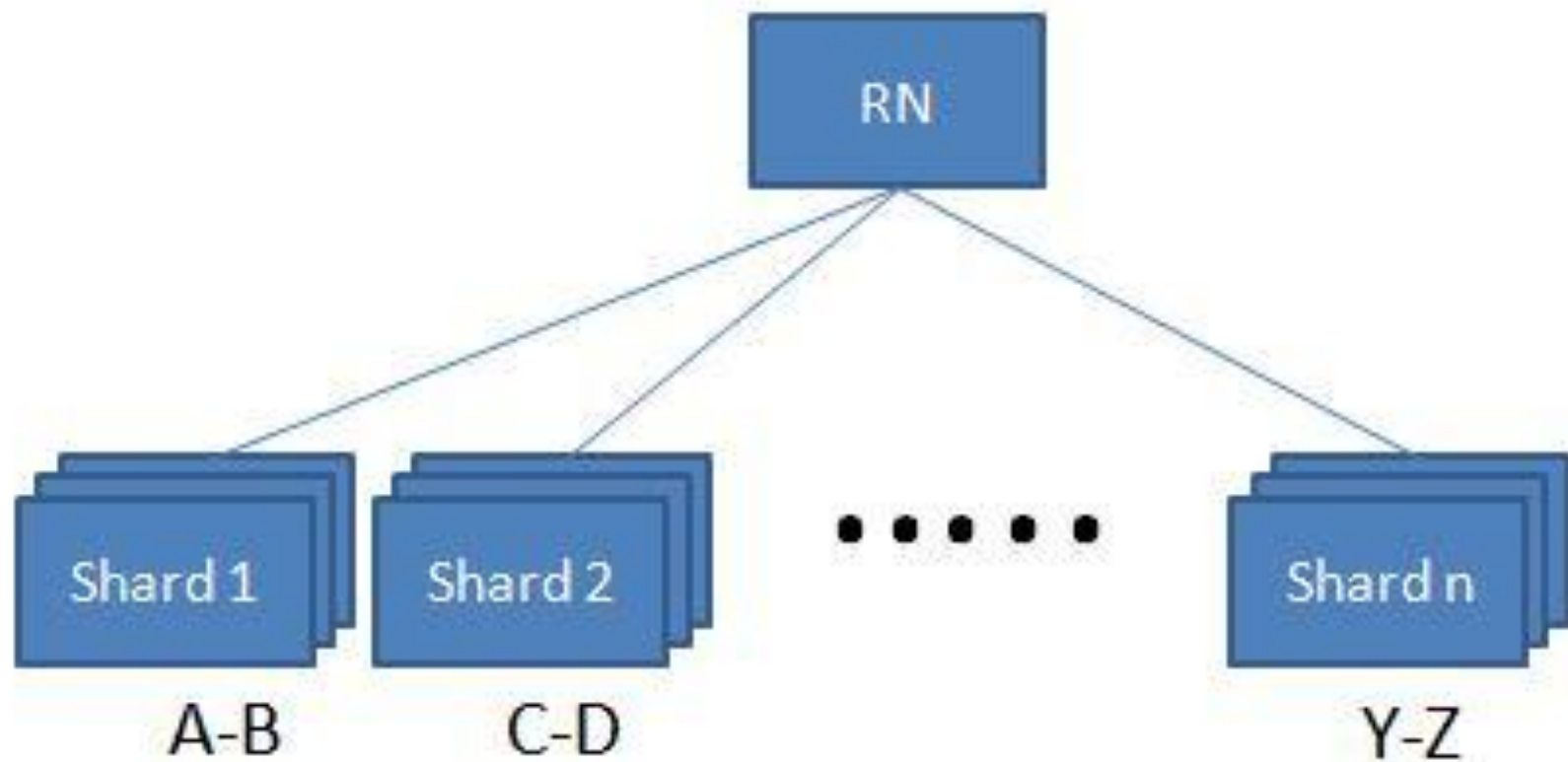
Algumas delas são requisitos para a utilização de alguns sistemas de bancos de dados No-SQL.

- Sharing
- MapReduce.

Sharding

- Sharding é um termo relacionado à Shared Nothing, que consiste em dividir os dados dados horizontalmente, ou seja, quebrar as tabelas, diminuindo o seu número de linhas e separando-as em ambientes diferentes (em outros servidores, por exemplo).
- Neste ponto todos os dados de uma partição não devem conter referências aos dados de outras partições, sendo que os dados em comum, deverão ser replicados entre as bases.

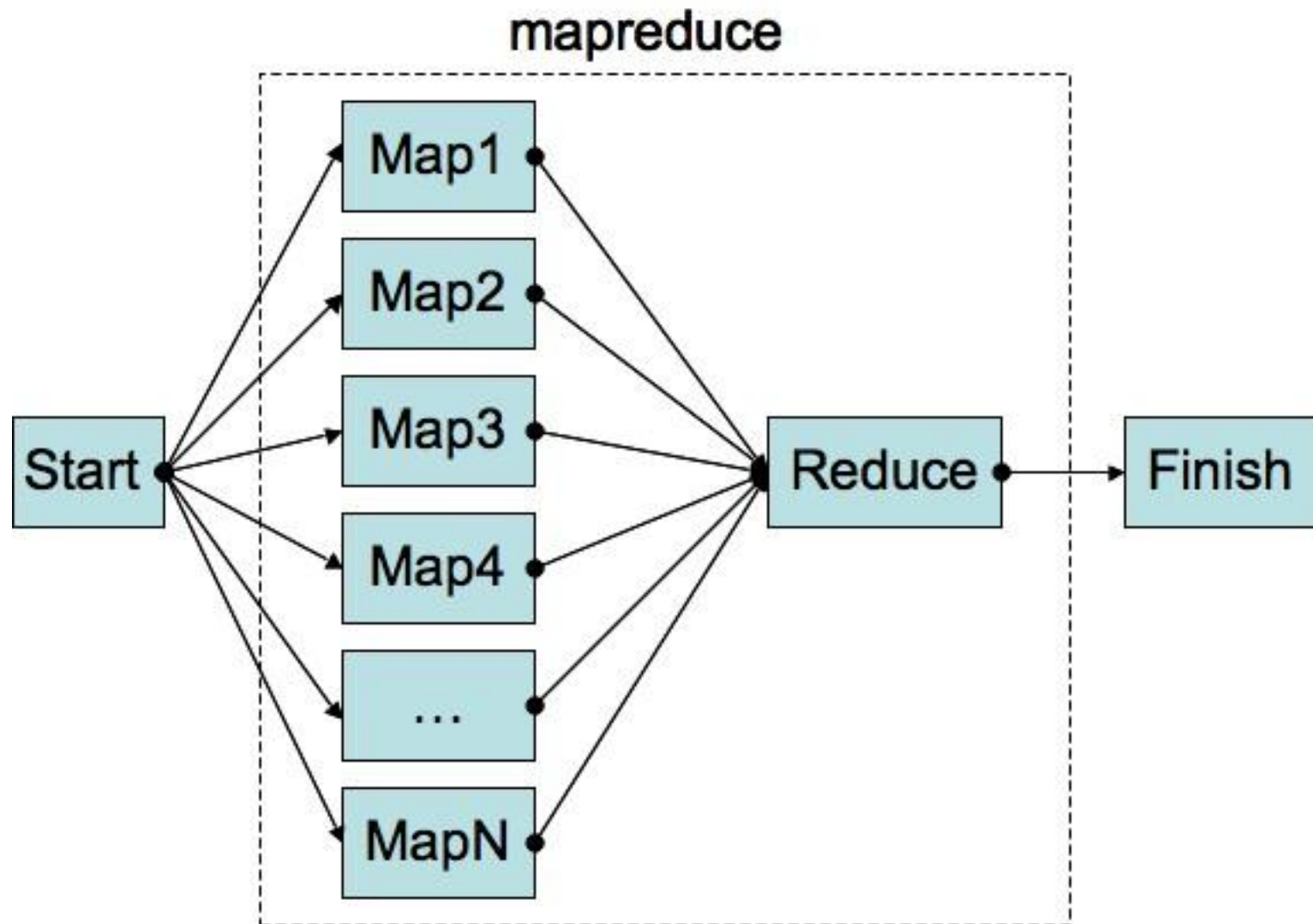
Sharding



MapReduce

- **MapReduce** é um algoritmo patenteado pelo Google para o processamento de dados de forma distribuída. Ele ocorre em duas fases:
- **Map:** O nó principal recebe os dados, divide e partes menores e as envia aos outros nós para serem processados. Ao final do processamento estes nós devolvem o resultado ao nó principal.
- **Reduce:** O nó principal combina as respostas obtidas pelos outros nós gerando o resultado final do processamento.

MapReduce



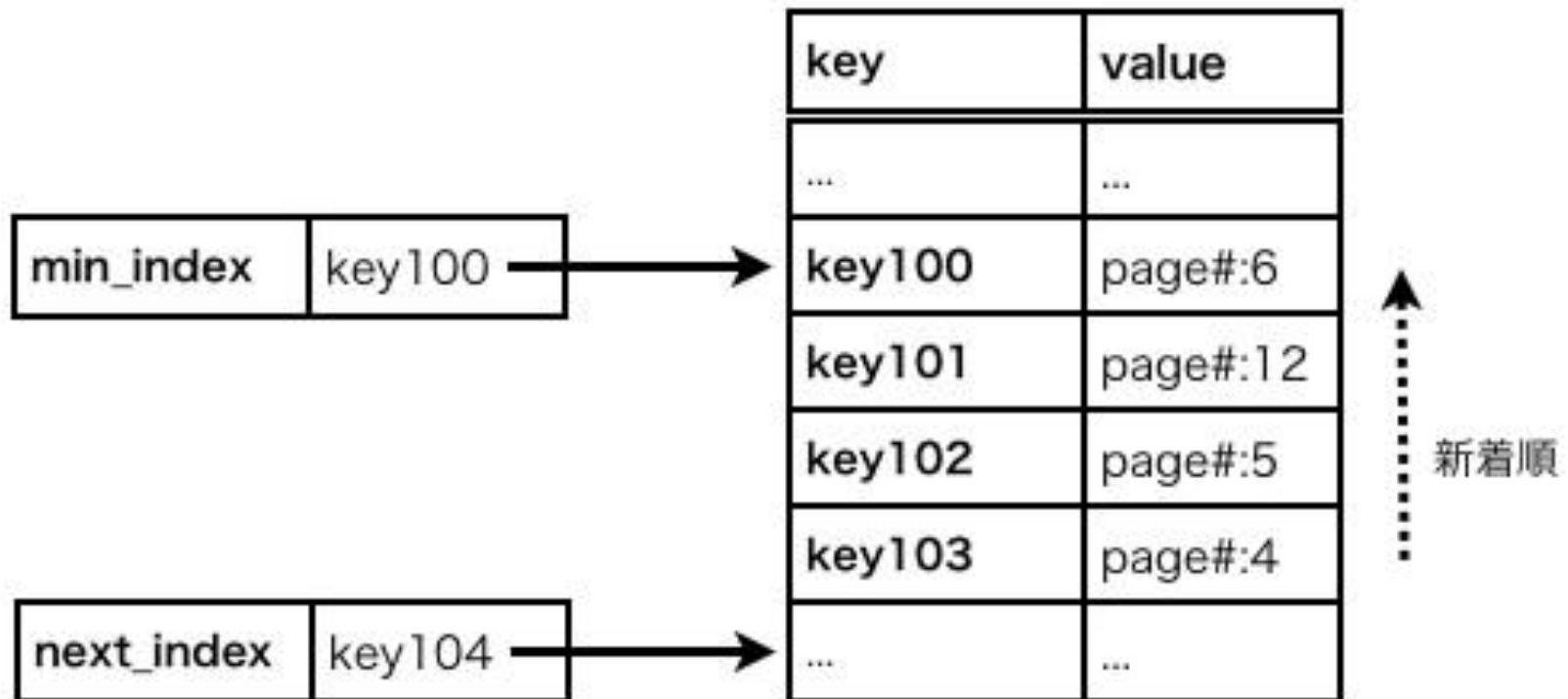
Principais tipos de banco No-SQL

- Key/Value
- Wide Columns – Column Families
- Document Store
- Graph Store
- Column Oriented Store

Key/Value Store (Chave/Valor)

- Esse é o tipo de banco de dados NoSQL mais simples.
- O conceito dele é uma chave e um valor para essa chave.
- Permite busca somente pela chave.
- É o que aguenta mais carga de dados.
- Esses tipos de bancos de dados, são o que tem a maior escalabilidade.

Key/Value Store (Chave/Valor)



Key/Value Store (Chave/Valor)



redis

- O **Redis**, é definido como advanced key-value store.
- Ele possui o código escrito em **C** sob a licença **BSD** e funciona em quase todos sistemas POSIX, como Linux ou Mac OS X.
- O **Redis** é um banco de dados **NOSQL** do tipo **chave: valor** persistente e distribuído, ou seja, uma ótima opção para armazenamento de cache.

Key/Value Store (Chave/Valor)



- **Memcached** é um sistema de **cache** em memória distribuído muito fácil de usar.
- A natureza do **Memcached** é extremamente genérica – trata-se de um serviço de rede e pode ser aplicado sem grandes mudanças a basicamente qualquer linguagem/ambiente de execução.

QUEM UTILIZA?



redis

- Boo-Box
 - Site: <http://www.boo-box.com/>

QUEM UTILIZA?



- Facebook
 - Site: <http://www.facebook.com/>
- WikiPedia
 - Site: <http://www.wikipedia.com/>
- YouTube
 - Site: <http://www.youtube.com>
- WikiPedia
 - Site: <http://www.wikipedia.com>
- Digg
 - Site: <http://www.digg.com>
- WordPress
 - Site: <http://www.wordpress.com>

Wide Columns Store

- Fortemente inspirados pelo BigTable do google.
- Suporta várias linhas e colunas, além disso ele permite subcolunas.

Wide Columns Store

1749217	<table><tr><td>text</td><td>user_id</td></tr><tr><td>tweet tweet tweet</td><td>4718</td></tr></table>	text	user_id	tweet tweet tweet	4718		
text	user_id						
tweet tweet tweet	4718						
1749218	<table><tr><td>text</td><td>user_id</td><td>in_reply</td></tr><tr><td>@nakhli blah blah</td><td>4718</td><td>1749216</td></tr></table>	text	user_id	in_reply	@nakhli blah blah	4718	1749216
text	user_id	in_reply					
@nakhli blah blah	4718	1749216					

Wide Columns Store



- O **Cassandra** é um repositório de dados leve feito em **Java**, que dispensa a sobrecarga de recursos dos bancos de dados relacionais convencionais.
- O objetivo é fornecer um ambiente **consistente**, **tolerante a falhas** e **alta disponibilidade** no momento do armazenamento de dados.

Wide Columns Store



- O **Hypertable** é um sistema de armazenamento **distribuído** de dados de **alto desempenho** projetado para suportar aplicações que exigem o máximo de desempenho, escalabilidade e eficiência.
- Foi concebido e modelado a partir do projeto **BigTable** Google e concentra-se principalmente conjuntos de dados em larga escala.

QUEM UTILIZA?



- Facebook
 - Site: <http://www.facebook.com/>
- Twitter
 - Site: <http://www.twitter.com/>
- Digg
 - Site: <http://www.digg.com>
- Cisco
 - Site: <http://>

QUEM UTILIZA?



- Rediff
 - Site: <http://www.rediff.com/>
- Baidu
 - Site: <http://www.baidu.com/>

Document Store

- Baseado em documentos XML ou JSON.
- Podem ser localizados pelo seu id único(Chave) ou por qualquer registro que tenha no documento.

Document Store

- Exemplo de um documento em JSON

```
{ "empinfo" :  
  {  
    "employees" : [  
      {  
        "name" : "Scott Philip",  
        "salary" : £44k,  
        "age" : 27,  
      },  
      {  
        "name" : "Tim Henn",  
        "salary" : £40k,  
        "age" : 27,  
      },  
      {  
        "name" : "Long Yong",  
        "salary" : £40k,  
        "age" : 28,  
      }  
    ]  
  }  
}
```

Document Store



- O **CouchDB** é um sistema de software livre de gerenciamento de banco de dados orientado a documentos que pode ser acessado através da API de JavaScript Object Notation (**JSON**) RESTful.
- O termo "Couch" é um acrônimo para "Cluster Of Unreliable Commodity Hardware", que reflete a meta do CouchDB de ser extremamente **escalável**, oferecendo **alta disponibilidade** e **confiabilidade**, mesmo ao executar em hardware que está geralmente sujeito à falhar.
- O CouchDB foi compilado originalmente em **C++**, mas em abril de 2008, o projeto foi movido para a plataforma **Erlang OTP** devido à sua ênfase em tolerância a falhas



- O [MongoDB](#) é um banco de dados orientado a documentos de alta performance, open source e schema-free, escrito em C++.
- Ele é uma mistura entre os repositórios escaláveis baseados em [chave/valor](#) e a tradicional riqueza de funcionalidades dos [bancos relacionais](#).
- Ele pode ser útil como um repositório de dados simples, rápido e não transacional para aplicações web ou para mecanismos de [cache](#).

QUEM UTILIZA?



- ShareThis
 - Site: <http://www.sharethis.com>
- Honda
 - Site: <http://www.honda.com>
- Adobe
 - Site: <http://www.adobe.com>
- PayPal
 - Site: <http://www.paypal.com>

QUEM UTILIZA?



- Amazon
 - Site: <http://www.amazon.com>
- Github
 - Site: <http://www.github.com>
- SourceForge
 - Site: <http://www.sourceforge.net>
- EA
 - Site: <http://www.ea.com>
- NY Times
 - Site: <http://www.nytimes.com>
- Justin.tv
 - Site: <http://www.justin.tv>

Graph Store

- Os dados não são dispostos em listas e sim em objetos.
- A busca destes itens são feitas pela navegação destes objetos.
- Muito utilizado em Redes sociais

Graph Store



Graph Store



- O **Neo4j** vem solucionar o problema de queda de performance entre **queries** que envolvem muitos **joins** em um SGBDR.
- Através da representação de dados utilizando **grafos** - leitura, escrita, e cruzamentos - , o Neo4j consegue navegar entre os nós e os relacionamentos com uma velocidade constante, independente da quantidade de dados que constituem o **grafo**. Isso nos trás alguns outros efeitos como algoritmos de grafos muito rápidos, sistema de recomendação e um estilo **OLAP** de análise o que atualmente não é possível com configuração normais de um SGBDR.

Graph Store



- É um banco de dados **gráfico** projetado especificamente para a **inteligência artificial** e **projetos de web semântica**, para que ele possa ser usado como um embutido de banco de dados orientado a objetos para projetos de todos os tamanhos.
- Praticamente não **existe limitações** de tamanho para o banco.

QUEM UTILIZA?



- ThingWorx
 - **Site:** <http://www.thingworx.com/>
- Evident Software
 - **Site:** <http://www.evidentsoftware.com/>
- MillionMind
 - **Site:** <http://millionmind.com/>
- INNOWEBTECH
 - **Site:** <http://www.inno-web-tech.com/>
- Jayway
 - **Site:** <http://www.jayway.com/>

Column Oriented Store

- Esses são bancos de dados relacionais, porem eles tem características do NoSQL.
- A principal diferença deles é que os dados são armazenados em colunas, ajudando na escalabilidade.

Graph Store



- Trata-se de um Banco de Dados Open Source, orientado a **colunas**, otimizado para **ferramentas analíticas**.
- Cada componente do Lucid foi projetado visando a flexibilidade, integração de dados de **alto desempenho** e **processamento de consultas sofisticadas**.
- Sua arquitetura é focada na simplicidade para o usuário, sendo assim, não é necessário DBA.



- É um Banco de Dados Open Source, orientado a **colunas**.
- Foi desenvolvido para lidar com consultas **complexas** de grandes **volumes de dados**.
- É utilizado em áreas de grande **massa de dados**, onde seria inviável utilizar um banco de dados tradicional.

QUEM UTILIZA?



- Actuate
 - **Site:** <http://www.actuate.com/home/>
- Marvelit
 - **Site:** <http://www.marvelit.com>

Gráfico – Comparação tipos de bancos NoSQL

Como apresentado no gráfico, os **key-value** aguentam maior quantidade de registros, enquanto os **grafos** são mais complexos.

