

BANCO DE DADOS NOSQL

Um novo tipo de banco de dados não relacional

IREMAR NUNES DE LIMA ¹

LEONARDO CLARO DE AVILA ²

Resumo: Este trabalho analisa o novo tipo de Banco de Dados não relacional chamado NoSQL: ele é uma alternativa para resolver problemas de escalabilidade e desempenho dos Bancos de Dados tradicionais. O artigo procura discutir as características, vantagens e desvantagens de se usar esse tipo de Banco de Dados no mercado e aborda cases de sucesso.

Palavras-chave: Banco de Dados, SGBD não relacional, NoSQL

1 INTRODUÇÃO

Os Bancos de Dados Relacionais dominam o mercado por mais de 30 anos, mas com a crescente necessidade de cumprir requisitos não-funcionais de uma aplicação como as relacionadas a performance e escalabilidade, deu-se o surgimento de buscar novas soluções, sendo umas delas denominada NoSQL (Not Only SQL).

Antes de abordar o NoSQL, é necessário saber o porquê do desenvolvimento dessa

¹ DBA, mestre em informática e professor do centro universitário Newton Paiva (iremar.prof@uol.com.br).

² Pós graduando em Banco de dados e *Business Intelligence* no Centro universitário Newton Paiva (leo.claro@gmail.com).

tecnologia. Segundo Nascimento (2010), o NoSQL foi projetado para promover uma alternativa de armazenamento com velocidade e disponibilidade elevada. Grandes empresas de tecnologia atualmente investem nos seus próprios SGBDs baseados na idéia NoSQL.

De acordo com Steppat (2009), devido a necessidade de suprir demandas de aplicativos web altamente escaláveis onde os Banco de Dados Relacionais são ineficazes, o movimento NoSQL está cada vez mais ganhando força no mercado, onde grande maioria desses bancos são *open source*. Porém com a necessidade de diversas empresas investirem na criação e utilização dessa nova solução, estão surgindo implementações proprietárias como o Google Bigtable e Amazon Dynamo, entre outros.

Este artigo busca comprovar que com o uso da solução NoSQL consegue-se suprir as necessidades das grandes organizações em cenários nos quais o gerenciamento de volume tem se mostrado problemáticos, pela grande quantidade de dados, e por serem sistemas de alta escalabilidade, ganhando a desejada performance com simplicidade.

De acordo com Nascimento (2010), Bancos de Dados NoSQL são uma alternativa para resolver o problema de escalabilidade e desempenho dos bancos tradicionais, pois tem um modelo de concorrência simplificado e o custo e a complexidade para escalar é bem inferior ao do Banco de Dados Relacional.

2 A TECNOLOGIA NOSQL

Segundo Nascimento (2010), o termo NoSQL, foi utilizado pela primeira vez em 1998, por Carlo Strozzi, como nome de seu SGBD (Sistema Gerenciador de Banco de Dados), baseado

no Modelo Relacional, sem interface SQL. É um termo genérico para uma classe definida de banco de dados que rompe uma longa história de banco de dados baseados nos De acordo com Steppat (2009), o NoSQL começou a ser desenvolvido por grandes empresas para suprir a necessidade de alto armazenamento e escalabilidade, sendo atualmente tratado como o futuro de grande armazenamento de informações, conseguindo assim trabalhar com esse novo cenário que os bancos de dados relacionais não atendem como desejado mesmo com alto investimento.

Segundo Steppat (2009), esse modelo de Banco de Dados é de tal importância que as maiores empresas atualmente em tecnologia estão desenvolvendo e aperfeiçoando seus próprios Banco de Dados NoSQL. Como exemplos citam-se as empresas Google com o seu Google Bigtable, o Facebook com o seu Cassandra, e o Amazon com o Amazon Dynamo.

A seguir, são destacadas as principais características de um banco de dados NOSQL:

1 Escalabilidade Horizontal (Scale out)

Segundo Edlich (2011) é possível adicionar vários nós ao sistema, tais como adicionar um novo servidor e um sistema de software que possibilita a divisão do trabalho entre várias máquinas.

2 Replicação

Segundo Ferreira (2010) a cópia das informações é realizada em vários bancos para aumentar o desempenho na recuperação dos dados. Existem dois tipos de arquiteturas de replicação no NoSQL:

- Master-Slave: O banco "Master" replica a escrita nos slaves, com isso é aumentado a

velocidade da leitura, porém não melhora a capacidade de escrita.

- Master-Master: Quanto maior o número de masters em nosso sistema, maior será a capacidade da escrita.

3 Schema-free

Segundo Edlich (2011) o NoSQL não possui schema, sendo um dos fatores que contribui no alto escalonamento, pois grande parte dos novos bancos NoSQL salvam os dados de forma desnormalizada.

4 Clusterização

Segundo Edlich (2011) o gerenciamento de dados é realizado por diversos servidores a fim de prover alta disponibilidade, e um alto desempenho, diminuindo o tempo e até o risco do sistema ficar temporariamente indisponível.

5 Gerenciamento em larga escala

De acordo com Steppat (2009) o Google criou o Mapreduce para gerenciamento em larga escala que se divide em duas fases:

- Map: Nó raiz recebe os dados e subdivide em partes menores enviando para os nós filhos para serem processados e retornarem o resultado obtido para o nó raiz.
- Reduce: Nó raiz junta as respostas retornadas pelos nós filhos gerando o resultado o retorno do processamento inicial ao qual foi solicitado.

6 Sharding

De acordo com Ferreira (2010), o *sharding* é uma alternativa muito conhecida que divide os dados em múltiplas tabelas do banco para escalar tanto as leituras como as escritas. Isto traz o

grande problema de “quebrar” a lógica de relacionamentos o que é o forte dos bancos relacionais. A aplicação tem que resolver a complexidade gerada pela partição de informações como, por exemplo, a execução de joins e outros comandos.

Segundo Ferreira (2010), fazer *sharding* manualmente não é simples e exige considerável esforço da equipe de desenvolvimento. No caso dos bancos de dados NoSQL muitas destas mesmas técnicas são utilizadas. Porém, em geral, elas são invisíveis para o desenvolvedor o que facilita o desenvolvimento de aplicações para dados em larga escala. Porém o ecossistema¹ que está surgindo em torno do termo “NoSQL” é muito grande e diversificado tornando assim difícil a tarefa de fazer generalizações. Deve-se tentar utilizar algumas das características básicas dos bancos não relacionais para poder classificá-los.

Diversas são as vantagens e desvantagens na utilização de bancos de dados NoSQL, conforme descrito a seguir:

1 Escalonamento Horizontal nos Servidores

Segundo Nascimento (2010), ao contrário do Banco de Dados Relacional que utiliza o escalonamento vertical, que com o aumento de dados, mais espaço e memória necessita. De acordo com Ferreira (2010), escalar verticalmente o Banco de Dados Relacional é limitado pelo fato de que chegaria a um limite de capacidade com uma única máquina, ou a um limite de orçamento para conseguir uma máquina extremamente poderosa.

Segundo Ferreira e Nascimento (2010), uma solução seria escalar horizontalmente o banco, ou seja, colocar mais servidores que não necessita ser de alta performance rodando o banco.

¹ Ecossistema: conjunto de tópicos relacionadas ao termo NoSQL, tais como: arquitetura, armazenamento, modelo de dados.

No Banco de Dados Relacional escalar os servidores horizontalmente é muito complexo por conta que toda a informação necessária está em um relacionamento entre várias tabelas, sendo necessário junta-las para formar a informação. No Banco de Dados NOSQL a informação necessária está agrupada no mesmo registro, ou seja, na busca dessa informação, será apontado diretamente ao servidor (nó) que possui o registro correspondente a essa informação.

2 Alta disponibilidade

De acordo com Sousa e Rocha, o Banco de Dados NOSQL "Possui um alto grau de distribuição de dados, e garante um maior numero de solicitações." (SOUSA; ROCHA, 2010).

3 Menor tempo de resposta

De acordo com Steppat (2009), uma característica interessante que possibilita menor tempo de resposta no NOSQL, é a possibilidade da distribuição dos dados em *SHARD*, ou seja, ele consegue distribuir os dados entre vários servidores, onde cada servidor possui uma fatia dos dados, assim ficando mais fáceis de serem processados por estarem em pequenos volumes do que um pacote completo.

4 Alta escalabilidade

Segundo Steppat (2009) o NoSQL não possui schema e trabalha desnormalizado, porém não é apenas por isso que ele consegue escalar facilmente os dados. Os novos bancos NOSQL trabalham com key-value stores, ou seja, armazenam as informações em um array associativo, o valor é associado a uma chave, possibilitando a busca desse objeto (valor) pela sua chave, o valor pode ser um binário, ou uma string que está sendo salva de forma desnormalizado.

De acordo com Steppat (2009), outra característica interessante que possibilita a alta escalabilidade no NOSQL é a possibilidade da distribuição dos dados em *SHARD*, ou seja, ele consegue distribuir os dados entre vários servidores, onde cada servidor possui uma fatia dos dados.

5 Sistema em nuvem

O NoSQL é uma ótima solução para aplicações em nuvem, levando vantagens nesse tipo de aplicações comparado ao SGBD tradicional, pois segundo Hjort "[...] as funcionalidades que essa tecnologia provê são altamente convergentes com as propostas da computação em nuvem: performance, escalabilidade horizontal, alta disponibilidade e flexibilidade" (HJORT, 2011).

De acordo com a DBPédias, os Bancos de Dados NOSQL Azure Tables, Datastores, SimpleDB, e Vertica, trabalham com o tipo de licença *SaaS*¹, ou seja, são bancos focados exatamente para o uso em plataformas em nuvem.

3 TIPOS DE BANCO DE DADOS NOSQL

Segundo Nascimento (2011) os Bancos de Dados NoSQL são classificados pelo seu núcleo, ou seja, como ele trabalha com seus dados. Os núcleos são:

1 Sistemas baseados em armazenamento chave-valor (Key/Value Store)

Segundo Diana e Gerosa, “Sistemas distribuídos nessa categoria, também conhecidos como tabelas de hash distribuídas, armazenam objetos indexados por chaves, e possibilitam a busca

¹ SaaS: Software as a Service, segundo Wikipedia é o uso de um software em regime de utilização web.

por esses objetos a partir de suas chaves” (DIANA; GEROSA, 2010).

São exemplos de bancos de dados NoSQL nesta categoria:

- Dynamo
- Tokyo Cabinet
- Oracle Berkeley DB
- MemcacheDB
- Project Voldemort
- Redis
- Riak

2 Sistemas baseados em armazenamento colunas amplas (Wide Column Store / Column Families)

De acordo com DBPédias, os sistemas baseados em armazenamento colunas amplas são similares aos Bancos de Dados Relacionais, ele é constituído por várias tabelas, cada uma contendo um conjunto de linhas endereçáveis. Cada linha consiste de um conjunto de valores que são considerados colunas.

No entanto segundo DBPédias, devido à escalabilidade e requisitos de armazenamento impostos pelas aplicações para as quais eles são usados, esses bancos diferem das tabelas relacionais de várias maneiras importantes:

- Cada linha pode ter um conjunto diferente de colunas. Apesar de todas as linhas em uma determinada tabela podem ser obrigadas a ter um conjunto pré-definido de grupos de colunas, linhas individuais podem diferir em colunas específicas dentro destes grupos. Isto significa que uma tabela pode ter valores diferentes de linha para linha. Além disso, as colunas não precisam ser pré-definidas em um esquema, mas em vez

disso podem ser adicionados de forma dinâmica.

- As tabelas são destinadas a ter muitas colunas, mais do que em um Banco de Dados Relacional. Uma linha pode conter milhares de colunas. Alguns desses bancos podem suportar milhões de valores de colunas por linha.
- Todos os bancos dessa categoria suportam valores composto, ou seja, um registro pode possuir vários campos nele.
- As linhas são normalmente versões. Isto significa que a cópia múltipla da mesma linha pode existir (isto é, com o ID mesma linha). As linhas são normalmente versionado por um sistema atribuído *timestamp*.
- O armazenamento de dados é normalmente dividido em *shard* (fragmentos), que são gerenciados de forma independente.

São exemplos de bancos de dados NoSQL nesta categoria:

- Google Bigtable
- Apache Cassandra
- HyperTable
- Apache HBase
- SimpleBD
- Azure Tables

3 Sistemas baseados em grafos (Graph Store)

De acordo com DBPédias e Nascimento (2011), os sistemas baseados em grafos usam nós(vértice), arestas, e propriedades como elementos primários, proporcionando um forte contraste com as tabelas, linhas, e colunas. Além disso, eles enfatizam alto desempenho para acesso a dados associativo, evitando a necessidade de *joins*.

Segundo DBPédias, um recurso que o esses bancos possuem, é a capacidade de armazenar sozinho o valor de um campo de outra entidade (vértice) que foi percorrido, assim originando as arestas com essa outra entidade. Além disso, eles geralmente permitem coleções de IDs para ser armazenado como um valor único usando um *array* ou o *map layout*, originando assim o grafo estruturado com diversos caminhos que pode percorrer para chegar a um determinado nó. Consequentemente, a busca de uma entidade, digamos, um nó, de suas entidades relacionadas, tais como suas arestas, pode ser realizada rapidamente, sem a manutenção de índices secundários.

São exemplos de bancos de dados NoSQL nesta categoria:

- Neo4J
- InfoGrid
- HyperGraphDB
- BigData

4 Sistemas orientados a documentos (Document Store)

Segundo DBPédias e Nascimento (2011), esses bancos focam no armazenamento e acesso otimizado em documentos estruturados em vez de linhas ou registros. Nesse banco, os documentos (objetos) são mapeados facilmente aos tipos de dados das linguagens de programação. Esse banco enfatiza menos em escalabilidade e tempo de atividade (up-time) comparado com outros bancos NoSQL. Os sistemas orientados a documentos atuais suportam documentos em formato *JSON*¹, *XML*², ou *BSON*³.

São exemplos de bancos de dados NoSQL nesta categoria:

¹ JSON: é um subconjunto da notação de objeto de JavaScript, mas seu uso não requer JavaScript exclusivamente.

² XML: O XML é um formato para a criação de documentos com dados organizados de forma hierárquica.

³ BSON: um subconjunto de binário de JSON, para armazenamento mais fácil e indexação.

- Apache CouchDB
- MongoDB
- RavenDB

5 Sistemas orientados a colunas (Column Oriented Store)

De acordo com Nascimento, “[...] são bancos de dados relacionais, porém apresentam características do NoSQL. A principal diferença deles é que os dados são armazenados em colunas, ajudando na escalabilidade” (NASCIMENTO, 2011).

São exemplos de bancos de dados relacionais nesta categoria:

- Vertica
- MonetDB
- LucidDB
- Infobright
- Ingres/Vectorwise

4 Vantagens e desvantagens da utilização do NoSQL

São as seguintes as vantagens da utilização do NoSQL:

1. Escalonamento Horizontal nos Servidores: Segundo Ferreira (2010) ao contrário dos Bancos de Dados Relacionais que utilizam o escalonamento vertical o NoSQL faz o escalonamento horizontal e com isso tem grande facilidade na distribuição dos dados. Isto reduz o custo e otimiza o armazenamento, pois os dados serão subdivididos entre vários servidores.
2. Alta disponibilidade: Por seu grau de distribuição de dados ser alto, consegue garantir maior numero de solicitações realizada;
3. Menor tempo de resposta: Com a distribuição de dados no nó raiz com seus nós filhos,

o tempo de resposta consegue ser menor, pois os dados são mais fáceis de serem processados em pequenos volumes do que um pacote completo.

4. Alta escalabilidade: Segundo Steppat (2009), o NoSQL por não possuir schema, e trabalhar desnormalizado consegue ter uma alta escalabilidade.
5. Sistema em nuvem: o NoSQL é ótima solução para aplicações em sistemas em nuvem, pois segundo Hjort "[...] as funcionalidades que essa tecnologia provê são altamente convergentes com as propostas da computação em nuvem: performance, escalabilidade horizontal, alta disponibilidade e flexibilidade" (HJORT, 2011).

5 TIPOS DE BANCO DE DADOS NOSQL EXISTENTES

Segundo Nascimento (2011) os Bancos de Dados NoSQL são classificados pelo seu núcleo, ou seja, como ele trabalha com seus dados. Os núcleos são:

1. Sistemas baseados em armazenamento chave-valor (Key/Value Store): Baseia-se numa coleção de chaves únicas e de valores, os quais são associados com as chaves. São exemplos:
 - Dynamo
 - Tokyo Cabinet
 - Oracle Berkeley DB
 - MemcacheDB
 - Project Voldermort
 - Redis
2. Sistemas baseados em armazenamento colunas amplas (Wide Column Store / Column Families): De acordo com Nascimento (2011) esse núcleo é baseado no Bigtable do Google, eles suportam várias linhas e colunas, além de permitir subcolunas. São

exemplos:

- Google Bigtable
- Apache Cassandra
- HyperTable
- Apache HBase
- SimpleBD
- Azure Tables

3. Sistemas baseados em grafos (Graph Store) : Segundo Nascimento (2011) os dados são armazenados em nós de um grafo (objeto) cujas arestas representam o tipo de associação entre esses nós, e não registro como os outros tipos de NoSQL. São exemplos:

- Neo4J
- InfoGrid
- HyperGraphDB
- BigData

4. Sistemas orientados a documentos (Document Store): Segundo Nascimento (2011) basea-se em documentos JSON ou XML. Estas são as unidades básicas de armazenamento e estes não utilizam qualquer tipo de estruturação pré-definida, como é o caso das tabelas do modelo relacional. São exemplos:

- Apache CouchDB
- MongoDB
- Riak
- RavenDB

5. Sistemas orientados a colunas (Column Oriented Store): Segundo Nascimento (2011) a

orientação a registro mudou-se para orientação a colunas (atributos). São exemplos:

- Vertica
- MonetDB
- LucidDB
- Infobright
- Ingres/Vectorwise

Pode-se citar alguns cases que lidam com uma enorme quantidade de informações e que resolveram optar pelo NoSQL como uma solução para melhor desempenho e escalabilidade.

São exemplos:

- Amazon Dynamo: Amazon.
- Apache CouchDB: Apache.
- SimpleDB: Amazon.
- Voldemort: LinkedIn.
- Cassandra: Facebook, Twitter, e a Digg.
- Google Bigtable: Google, e a Caelum.
- Mixi: Facebook do Japão.
- MongoDB: New York Times.

6 CONCLUSÃO

A tecnologia NoSQL está presente em algumas grandes organizações, em cenários que necessitam de respostas instantâneas. Nesse cenário, as tecnologias tradicionais para gestão e gerenciamento dos dados apresentam limitações. Em particular, pode-se citar a escalabilidade.

Os Bancos de Dados NoSQL são ainda muito recentes, e estão cercados por polêmicas, mas há bons indícios de que será uma tecnologia inovadora e que pode ser aplicado em diversos cenários. No momento, estão em produção vários desses Bancos de Dados em aplicações de grande escala, como exemplo, o Cassandra no Facebook, Twitter e Digg.

REFERÊNCIAS

DBPEDIAS. Survey of Distributed Databases. Disponível em:

<http://dbpedias.com/wiki/NoSQL:Survey_of_Distributed_Databases>. Acesso em: 20 abr. 2012.

DIANA, Edmar de; GEROSA, Marco Aurélio. NOSQL na Web 2.0: Um Estudo Comparativo de Bancos Não-Relacionais para Armazenamento de Dados na Web 2.0. WTDBD 2010. 2010. Disponível em:

<http://www.lbd.dcc.ufmg.br/colecoes/wtdbd/2010/sbbd_wtd_12.pdf>. Acesso em: 20 abr. 2012.

EDLICH, Stefan. NOSQL Databases. 2011.

Disponível em: <<http://nosql-database.org/>>. Acesso em: 22 dez. 2011.

FERREIRA, Edmar. Introdução ao NoSQL parte I. ESCALABILIDADE. 2010. Disponível em: <<http://escalabilidade.com/2010/03/08/introducao-ao-nosql-parte-i>>. Acesso em: 23 nov. 2011.

FERREIRA, Edmar. Introdução ao NoSQL parte II. ESCALABILIDADE. 2010. Disponível em: <<http://escalabilidade.com/2010/04/06/introducao-ao-nosql-parte-ii>>. Acesso em: 23 nov. 2011.

HJORT, Rodrigo. Persistência nas Nuvens com NoSQL. AGAJORTE. 2011. Disponível em:

<<http://www.hjort.co/2011/06/persistencia-nas-nuvens-com-nosql.html>>. Acesso em: 10 mai. 2012.

LEMKE, Camilla. NoSQL na Nuvem. UNDER-LINUX. 2010. Disponível em: <<http://under-linux.org/nosql-na-nuvem-1372/>>. Acesso em: 10 mai. 2012.

NASCIMENTO, Jean. Introdução ao NoSQL. NoSQL Br. 2010. Disponível em:

<<http://www.nosqlbr.com.br/introducao-ao-nosql.html>>. Acesso em: 20 dez. 2011.

NASCIMENTO, Jean. NoSQL - você realmente sabe do que estamos falando?. IMASTERS. 2011. Disponível em:

<http://imasters.com.br/artigo/17043/bancodedados/nosql_voce_realmente_sabe_do_que_estamos_falando/>. Acesso em: 21 nov. 2011.

SOUSA, Thalles Ramon Pinheiro de; **ROCHA**, André Luiz de Santana Silva. **NoSQL**. CABEDELO-PB. 2010. Disponível em:

<<http://www.slideshare.net/andrerochajp/artigo-nosql>>. Acesso em: 15 fev. 2012.

STEPPAT, Nico. **Bancos de dados não relacionais e o movimento NoSQL**. CAELUM. 2009. Disponível em: <<http://blog.caelum.com.br/bancos-de-dados-nao-relacionais-e-o-movimento-nosql/>>. Acesso em: 25 nov. 2011.

WIKIPEDIA. **Computação em nuvem**. Disponível em:

<http://pt.wikipedia.org/wiki/Computa%C3%A7%C3%A3o_em_nuvem>. Acesso em: 15 mai. 2012.

WIKIPEDIA. **NoSQL**. Disponível em:

<<http://pt.wikipedia.org/wiki/NoSQL>>. Acesso em: 20 nov. 2011.