BANCOS DE DADOS NÃO-RELACIONAIS: UM NOVO PARADIGMA PARA ARMAZENAMENTO DE DADOS EM SISTEMAS DE ENSINO COLABORATIVO

Samuel Silva de Oliveira[1]

Resumo

A popularização do uso da internet e a grande quantidade de ferramentas nela disponíveis criaram uma nova forma de aprendizagem: o ensino colaborativo. O número de usuários ativos de internet no Brasil chegou a 60,7 milhões no mês de março de 2014, segundo levantamento divulgado pela Nielsen Ibope. Ainda segundo a companhia, houve um crescimento de 6,5% nesse número comparado a fevereiro. A quantidade de usuários que acessa a rede de casa chegou a 51,6 milhões. Esses números deixam claro o perfil do internauta brasileiro, quais sejam pessoas interessadas em trocar conhecimento e informações.

Muitas empresas e entidades diversas apostam no ensino colaborativo investindo em sites de troca de conteúdo, onde a informação é compartilhada por cada um dos usuários.

Com o advento do crescimento da web e de novas soluções desenvolvidas através do avanço tecnológico baseado em sistemas distribuídos, um enorme volume de informações passou a ser gerado por pessoas e entidades em todo o mundo. Todavia, constatou-se que os modelos de bancos de dados relacionais popularmente utilizados, apresentam limitações ao trabalhar com grandes volumes de dados. Consequentemente, surgiu a necessidade de criar um modelo de banco de dados dotado de escalabilidade, capaz de manipular uma crescente quantidade de dados de maneira uniforme. A partir de então, de acordo com estudos realizados sobre bancos de dados distribuídos e possíveis melhorias para alcançar maior nível de escalabilidade, e alta disponibilidade, novas aplicações não-relacionais foram desenvolvidas criando uma nova tendência chamada de NoSQL.

Estudos sobre conceitos, características e casos de uso de bancos de dados desenvolvidos sob esta perspectiva são apresentados neste artigo com o intuito de mostrar como o modelo não-relacional lida com a necessidade de escalabilidade e qualidade de serviço.

Palavras-chave: Banco de Dados, NoSQL, Escalabilidade

---

[1] Desenvolvedor de software. Atualmente trabalhando do PRODAP - Processamento de Dados do Amapá. Tem conhecimentos em Ruby, PHP, Javascript/jQuery, Metodologias Ágeis e tudo relacionado ao desenvolvimento de software.

Abstract

The popularization of the internet and the large quantity of tools available into it, created a new way of learning: collaborative teaching. The number of active users at Brasil reached 60,7 millions of users in March, 2014, according to a research made by Nielsen Ibope. According to this company, there was a growth of 6,5% of this number compared to February. The quantity of users who access the internet from their houses reached 51,6 millions of users. This make clear the profile of the brazilians users: People interested in exchange of knowledge and informations.

Many companies and entities use collaborative teaching, investing in websites of content's exchange where the knowledge is shared by every single user.

According to the growth of the web and the new solutions developed throught the technological progress based in distributed systems, a large volume of informations became to be created by people and companies worldwide. Nevertheless, the models of the databases used until then had some difficulties when working with a large volume of data. Thereafter, the requirement arose of creating a database model provided with scalability, being able to manipulate a crescent quantity of data of a uniform way. From then on, according to the researches conducted about distributed databases and likely improvements to reach a bigger level of scalability and high-availability, new non-relationals applications were developed creating a movement called NoSQL.

Researches about concepts, features and use cases of database built under this view are presented in this article in order to show how the non-relational model works with the requirement to scalability and quality of service.

Keywords: Database, NoSQL, Scalability

1. Introdução

Com a globalização, a expansão virtual se tornou efetiva e após o ano 2000 houve uma crescente de dados exponencial, preocupando vários especialistas em razão da falta de espaço em armazenamento. Segundo a IBM, em 2008 foram produzidos cerca de 2,5 quintilhões de bytes todos os dias e surpreendentemente 90% dos dados no mundo foram criados nos dois anos anteriores. Este fato é decorrente da adesão de grandes empresas à internet, como redes sociais, companhias de telefonia móvel, dentre outras.

A tendência de utilizar a internet como um canal de estudo já é real pela possibilidade de troca de conhecimento pelos usuários. Os sites de conteúdo colaborativo permitem essa troca, gerando um grande volume de dados a ser armazenado. O Wikipedia é um exemplo de sistema web estabelecido como uma enciclopédia livre, construído por milhares de colaboradores de todas as partes do mundo, sendo baseado no conceito de wiki, o que significa que qualquer internauta pode editar o conteúdo de quase todos os artigos. "O projeto Wikipédia foi iniciado em 15 de janeiro de 2001, na versão em língua inglesa. Em apenas um ano de existência, esta versão já possuía quase 10 mil artigos. Até hoje já foram criados mais de 14 milhões de artigos em centenas de línguas e dialetos (834.085 artigos na versão em português). Todos os dias, centenas de colaboradores de todas as partes do mundo editam milhares de artigos e criam muitos verbetes inteiramente novos." (Em: http://pt.wikipedia.org/wiki/Wikipédia:Sobre_a_Wikipédia. Acesso em: 05 agosto 2014.)

O paradigma sobre o armazenamento de dados em sistemas como esse, trazem à tona estudos sobre as características necessárias em modelos de banco de dados para que esse crescente volume de informações seja gerenciado de forma adequada.

Segundo Heuser (1998, p. 16) "Um modelo de banco de dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados". Um dos modelos de bancos de dados amplamente usados até os dias de hoje é o modelo relacional. Porém, constatou-se que os modelos de bancos de dados relacionais apresentam limitações ao trabalhar com grandes volumes de dados.

A disseminação do tema "modelos de bancos de dados não-relacionais" provém desse crescente volume de dados gerados na web, e da percepção de que o modelo relacional pode se mostrar ineficiente quando utilizado para gerenciar tão grande quantidade de informações.

O termo NoSQL foi primeiramente utilizado em 1998 como o nome de um banco de dados não-relacional de código aberto criado por Carol Strozzi.

Em 2006, o artigo "BigTable: A Distributed Storage System for Structured Data" publicado pelo Google resgatou o termo NoSQL como um conceito de gerenciamento de megadados. O BigTable foi descrito como um banco extremamente escalável e tolerante a falhas onde os dados inseridos já entram indexados, tornando assim mais rápida a consulta aos dados armazenados.

A grande motivação para o movimento NoSQL foi de resolver o problema de escalabilidade dos bancos tradicionais, tendo em vista que pode ser muito caro e/ou complexo escalar um banco de dados relacional.

Inspirada nesse novo conceito disseminado, a comunidade de software livre e código aberto em geral desenvolveram diversas soluções de bancos de dados não-relacionais seguindo diferentes vertentes.

2. Bancos de Dados Relacionais

O modelo relacional em bancos de dados é fundamentado no princípio de que dados são guardados em tabelas. Toda sua definição é teórica e baseada na teoria dos conjuntos, ramo da matemática que estuda conjuntos, que são uma coleção de elementos. O modelo relacional foi idealizado por Edgar Frank Codd, que o descreveu no artigo "Relational Model of Data for Large Shared Data Banks" ("Modelo de dados relacional para grandes bancos de dados compartilhados") quando era pesquisador da IBM em San José. Com o passar do tempo, o modelo relacional se tornou o sucessor do modelo hierárquico e do modelo em rede, amplamente utilizados anteriormente.

No aprimoramento do modelo relacional, novas funcionalidades foram adicionadas, como o tratamento de orientação a objetos sem comprometer os seus princípios fundamentais, chamado modelo objeto-relacional. Contudo, segundo Dias Neto (2013. p.12)

> Apesar de o armazenamento em um banco de dados relacional parecer algo simples, na prática não é. Isso porque não existe uma tradução perfeita e automática entre as tecnologias de objeto e relacional, pois essas tecnologias são baseadas em teorias diferentes.

Nos dias de hoje, esse modelo ainda é amplamente utilizado pelo fato de prover acesso facilitado aos dados, possibilitando aos usuários utilizar uma grande variedade de abordagens no tratamento das informações, além da possibilidade de uso dos sistemas gerenciadores de bancos de dados, que executam comandos na linguagem SQL (Structured Query Language) e têm a responsabilidade de gerenciar o acesso, a manipular e a organizar os dados, principalmente no que diz respeito à segurança.

Com o passar do tempo e com o crescente volume de dados gerados a partir da expansão virtual, identificou-se que o modelo relacional não é tão escalável quanto necessário. Quando utilizado para gerenciar um grande volume de informações e cargas de trabalhos típicas de operações modernas de grande carga, incluindo o dimensionamento de conjuntos de dados, o banco de dados relacional perde sua performance consideravelmente.

Neste sentido, quanto mais dados forem gerados, mais recursos de hardware serão necessários, como memórias e discos, para que a qualidade do serviço seja mantida.

3. Bancos de dados não-relacionais

Para evitar o custo da escalabilidade em ambientes relacionais, iniciou-se, ao longo do tempo, o desenvolvimento de bancos distribuídos capazes de gerenciar dados semi-estruturados provenientes de diversas origens e que possibilitavam escalabilidade mais barata e menos complexa, não necessitando de servidores muito robustos e nem um grande numero de profissionais para o gerenciar.

De acordo com Moura e Casanova (1999, p. 14)

> A criação de Sistemas de Gerenciamento de Bancos de Dados Distribuídos contribui de forma significativa para o aumento da produtividade em desenvolvimento de aplicações, um fator importante desde longa data. De fato, tais sistemas simplificam a tarefa de se definir aplicações que requerem o compartilhamento de informação entre usuários, programas ou organizações onde os usuários da informação, ou mesmo as fontes de informação, estão geograficamente dispersas.

Dessa forma, os bancos de dados não-relacionais ficaram muito populares entre as grandes empresas geradoras de conteúdo, e são amplamente difundidos na comunidade open source e software livre.

Uma das primeiras aplicações maduras de bancos de dados baseada no modelo não-relacional surgiu em 2004 quando o Google lançou o BigTable, descrito como um banco de dados de alta performance com o objetivo de alcançar um alto nível de escalabilidade e disponibilidade e ser tolerante à falhas afim de gerenciar Petabytes de informações.[2]

No ano de 2005, foi lançado um release inicial de um banco de dados não-relacional open source chamado CouchDB. Este banco de dados usa JSON (JavaScript Object Notation) para armazenar dados. JSON é um formato leve para intercâmbio de dados computacionais. Além disso, o CouchDB usa Javascript como linguagem de consulta com o MapReduce, um modelo de programação para processamento de grandes volumes de dados com um algorítimo paralelo e distribuído em cluster. O CouchDB é mantido pela Fundação Apache.

Em 2007, a Amazon publicou um artigo chamado “Dynamo: Amazon's Highly Available Key-value Store” descrevendo o Dynamo como um banco de dados de alta disponibilidade baseado no armazenamento de chave-valor (Key-value) usado nos servidores

[2] CHANG, Fay et al. Bigtable: A Distributed Storage System for Structured Data. 2006. Disponível em: <http://static.googleusercontent.com/media/research.google.com/pt-BR//archive/bigtable-osdi06.pdf>. Acesso em: 04 maio 2014.

da Amazon para prover uma experiência "always-on" (sempre ativo).

No ano de 2008, o Facebook iniciou o desenvolvimento do "Cassandra", um banco de dados distribuído não-relacional escrito em Java.

Com o grande volume de dados criado a partir da popularização do Facebook, surgiu a necessidade de criar um banco de dados com alto nível de escalabilidade, alta disponibilidade e tolerante à falhas baseado na computação em nuvem. O Cassandra é escrito em Java, utiliza a arquitetura do Dynamo, da Amazon e o modelo de dados é baseado no BigTable do Google. Ainda em 2008 o Facebook abriu o código fonte do Cassandra que passou a ser mantido pelos desenvolvedores da Fundação Apache a partir de 2009.[3]

A empresa 10Gen, na mesma época, lançou publicamente a primeira versão do MongoDB em fevereiro de 2009. O MongoDB é uma aplicação de código aberto de alta performance, sem esquemas e orientado à documentos. Foi escrito na linguagem C++. Além de orientado à documentos, é formado por um conjunto de documentos JSON. Este banco de dados possui muitas características semelhantes ao CouchDB, desenvolvido pela Fundação Apache.

Todos esses bancos de dados foram construídos baseados em algumas classificações. O próximo capítulo descreve algumas especificações sobre classificações de bancos de dados não-relacionais.

4. Classificações de bancos de dados não-relacionais

Os bancos de dados não relacionais são classificados em Bancos de esquema Chave/Valor (Key/Value Store), Bancos de dados orientados à documentos, Bancos de dados de Colunas e Bancos de dados de Grafos.

4.1. Bancos de dados de esquema Chave/Valor

Bancos de dados desta classificação trabalham com tabelas de hash Distribuídos (DHT).

Tratam-se de um conjunto de algoritmos ou matrizes programado para buscar em todos os dados dos arquivos compartilhados. É comumente usado por programas de compartilhamento conhecidos por mudanças frequentes. Os nós são programados para encontrar assuntos específicos em arquivos e trazê-los como resultado da busca.

[3]LAKSHMAN, Avinash; MALIK, Prashant. Cassandra - A Decentralized Structured Storage System. 2008. Disponível em: <https://www.cs.cornell.edu/projects/ladis2009/papers/lakshman-ladis2009.pdf>. Acesso em: 11 maio 2014.

Bancos de dados Chave/Valor são bem simplificados. Eles armazenam objetos indexados por chaves e possibilitam a busca por esses objetos a partir de suas chaves.

Este modelo, por ser de fácil implementação, permite que os dados sejam rapidamente acessados pela chave, principalmente em sistemas que possuem alta escalabilidade, contribuindo também para aumentar a disponibilidade de acesso aos dados. As operações disponíveis para manipulação de dados são bem simples, como o get( ) e o set( ), que permitem retornar e capturar valores, respectivamente. A desvantagem deste modelo é que não permite a recuperação de objetos por meio de consultas mais complexas. Alguns bancos que utilizam esse padrão são: DynamoDb, Couchbase, Riak, Azure Table Storage, Redis, Tokyo Cabinet, Berkeley DB, dentre outros.

4.2. Bancos de dados orientados à documentos

Bancos de dados orientados a documentos são baseados no armazenamento de pares de chave-valor, tendo um esquema altamente flexível. Esta característica torna os bancos de dados orientados à documentos ótimas opções para dados semi-estruturados, como os utilizados em ferramentas web colaborativas.

No modelo orientado a documentos temos um conjunto de documentos e em cada documento temos um conjunto de campos (chaves) e o valor deste campo. Outra característica importante é que este modelo não depende de um esquema rígido, ou seja, não exige uma estrutura fixa como ocorre nos bancos relacionais. Assim, é possível que ocorra uma atualização na estrutura do documento, com a adição de novos campos, por exemplo, sem causar problemas ao banco de dados.

O formato utilizado nesta classificação de bancos de dados não-relacional é o JSON, como pode ser visto na figura 4.1.

```
{
 autor: 'samuel',
 criado : new Date('10-08-2014'),
 titulo : 'Titulo da Postagem',
 texto : 'Aqui está o texto...',
 tags : [ 'exemplo', 'samuel' ],
 comentarios : [ { autor: 'joao', comentario: 'Ruim' },
 { autor: 'maria', comentario: 'Legal' }]
}
```

Figura 4.1. Exemplo de formato JSON

Como principais soluções que adotam o modelo orientado a documentos, se destacam o CouchDB e o MongoDB.

Um caso de sucesso do uso do MongoDB é o CartolaFC, um Fantasy Game criado pela Globo.com. Essa é a maior aplicação dinâmica do portal baseada em NoSQL, com mais de 2 milhões de usuários cadastrados e quase 90 milhões de pageviews somente no mês de Junho de 2011. (informação verbal)[4].

4.3. Bancos de dados orientados a colunas

Este modelo é mais complexo que o chave-valor, e neste caso, muda-se o paradigma da orientação à registros para a orientação à colunas (modelo não-relacional). Neste caso, nem todas as linhas têm a mesma quantidade de colunas. Nesse sentido, a escrita de um novo registro é bem mais custosa do que em um banco de dados tradicional. Assim, num primeiro momento, os bancos tradicionais são mais adequados a processamento de transações online (OLTP) enquanto os bancos de dados de famílias de colunas são mais interessantes para processamento analítico online (OLAP).

Bancos de dados orientados à colunas têm esquema flexibilizado.

Quando se quer otimizar a leitura de dados estruturados, bancos de dados de orientados à colunas são mais interessantes, pois eles guardam os dados contiguamente por coluna.

Este modelo foi fortemente inspirados pelo BigTable, do Google. Ele suporta várias linhas e colunas, além de permitir subcolunas. Além do BigTable, outros bancos de dados também se baseiam neste princípio, como: Hadoop, Cassanda, Hypertable, Amazon SimpleDB, dentre outros.

4.4. Bancos de dados de Grafos

Com uma complexidade maior, esses bancos de dados guardam objetos, e não registros como os outros tipos de NoSQL. A busca desses itens é feita pela navegação desses objetos. Bancos de dados dessa classificação armazenam vértices e arestas, representando interconectividade entre os dados.

O modelo de grafos é mais interessante que outros quando informações sobre a interconectividade ou a topologia dos dados são mais importantes, ou tão importante quanto os dados propriamente ditos.

O modelo orientado à grafos possui três componentes básicos: os nós (são os vértices do grafo), os relacionamentos (são as arestas) e as propriedades (ou atributos) dos nós e

[4] Informação fornecida por Franklin Amorim na Conferência de Usuários de MongoDB, em São Paulo, em Julho de 2011

relacionamentos.

Neste caso, o banco de dados pode ser visto como um multigrafo rotulado e direcionado, onde cada par de nós pode ser conectado por mais de uma aresta.

Comparado ao modelo relacional, que para estas situações pode ser muito custoso, o modelo orientado a grafos tem um ganho de performance, permitindo um melhor desempenho das aplicações.

5. Características comuns em bancos de dados não-relacionais

Todos os bancos de dados não-relacionais possuem características e requerimentos que os diferenciam dos bancos de dados relacionais convencionais. Essas características os tornam capazes de manipular grandes volumes de dados não estruturados ou semi-estruturados: Escalabilidade, Alta disponibilidade, esquema flexível e simples manipulação.

5.1. Escalabilidade

Escalabilidade é a possibilidade de crescimento de qualquer sistema de armazenamento de dados, com o menor custo possível. Nesse sentido, ser escalável significa ter a habilidade de manipular uma porção crescente de trabalho de forma uniforme, ou estar preparado para crescer.

Escalar horizontalmente um sistema significa adicionar mais nós ao sistema, tais como um novo computador com a aplicação específica usando técnicas de clustering.

Escalar verticalmente um sistema significa adicionar recursos em um único nó do sistema adicionando mais memória ou um disco rígido mais rápido.

O Amazon Dynamo se destacou por causa da forma como o sistema escala. Cada nó no cluster comunica com outros nós e faz ativamente parte da partição/replicação.

5.2 Alta disponibilidade

Um sistema de alta disponibilidade é um sistema capaz de resistir à falhas, cujo objetivo é manter os serviços ativos o máximo de tempo possível. Além de ser tolerante à falhas de software, um sistema de banco de dados não relacional requer alto gerenciamento de memória e processador para estar apto a responder à todas as requisições com o menor tempo de resposta possível.

5.3. Esquema Flexível

Um esquema de um banco de dados é a descrição de sua estrutura em uma linguagem formal suportada pelo sistema de gerenciamento de banco de dados (SGBD) e refere-se à organização de dados como um diagrama de como um banco de dados é construído (dividido em tabelas de banco de dados no caso de bancos de dados relacionais).

Os bancos de dados não-relacionais não possuem esse esquema determinístico. Diferentemente dos bancos SQL, não existe uma esquema forte. Essa abordagem facilita a distribuição dos dados entre vários servidores onde cada servidor possui apenas uma fatia dos dados.

5.4. Simples manipulação

Os bancos de dados não-relacionais em geral demonstram grande simplicidade na sua manipulação e configuração. Um exemplo está no banco de dados MongoDB que utiliza o formato JSON, trazendo mais facilidade ao desenvolvedor.

Dependendo da política de cada sistema, não é necessário manter especialistas em bancos de dados para gerenciar bancos de dados não-relacionais. Devido à sua simplicidade, os próprios desenvolvedores podem executar esta tarefa.

6. Conclusão

Geralmente, entidades que decidem utilizar bancos de dados não-relacionais buscam por um alto nível de escalabilidade para trabalhar com grandes volumes de dados e alta disponibilidade afim de oferecer o menor tempo de resposta aos seus usuários.

Em sistemas de ensino colaborativos, onde novos dados são criados à todo instante e necessitam destas características, como portais e comunidades online e fóruns, abordagens NoSQL podem ser utilizadas como solução para armazenamento de dados. Devido ao gargalo causado pelos problemas encontrados nos modelos tradicionais de bancos de dados. Várias empresas como Google, Facebook e Amazon já aderiram a soluções NoSQL, cada uma de acordo com as necessidades dos serviços prestados.

Embora estes modelos de bancos de dados demonstrem grande desempenho e melhorias em relação ao modelo relacional, é importante lembrar que nem sempre será possível garantir a consistência dos dados, controle de concorrência, dentre outras características fundamentais dos bancos de dados convencionais.

Pesquisas afirmam que a quantidade de dados gerados na web tende a aumentar em sistemas colaborativos devido à possibilidade que os usuários encontram em compartilhar suas experiências e aprender com outros usuários que praticam a mesma ação.

É importante ressaltar que soluções não-relacionais não foram construídas com a finalidade de substituir o modelo relacional, que ainda é amplamente utilizado com eficácia nos dias de hoje, mas permitir que aplicações possam gerenciar os seus grandes volumes de dados de forma mais eficiente, o que nem sempre é possível utilizando bancos de dados relacionais.

O ensino colaborativo na internet têm se destacado como um novo paradigma na educação atual e os bancos de dados não-relacionais são uma das diversas criações frente à esse desafio que é a educação no século XXI.

REFERÊNCIAS

ANDERSON, J. Chris; LEHNARDT, Jan; SLATER, Noah. **CouchDB: The Definitive Guide.** 2009. 1ª Edição. O'Reilly Media. Disponível em: <http://guide.couchdb.org/>. Acesso em: 26 jun. 2014.

CASANOVA, Marco Antonio; MOURA, Arnaldo Vieira. **Princípios de Sistemas de Gerência de Bancos de Dados Distribuídos:** Edição Revisada. 1999. Disponível em: <http://www.inf.puc-rio.br/~casanova/Publications/Books/1985-BDD.pdf>. Acesso em: 05 ago. 2014.

DIAS NETO, Arilo Cláudio. Banco de Dados Relacionais - Artigo Revista SQL Magazine 86. 2013. Disponível em :<http://www.devmedia.com.br/bancos-de-dados-relacionais-artigo-revista-sql-magazine-86/20401>. Acesso em 19 mai. 2014

HEUER, Carlos Alberto. **Projeto de Bancos de Dados:** Série Livros Didáticos. 2. ed. Rio Grande do Sul: Sagra Luzzato, 1998. 206 p.

LEAVITT, Neal. **Will NoSQL Databases Live Up to Their Promise?**. IEEE Computer (COMPUTER), Fallbrook - CA, v.43 n.2, p:12-14, 26 Jan. 2010. Mensal

STONEBRAKER, Michael. SQL Databases v. NoSQL Databases. **Communications Of The Acm,** New York, v. 53, n. 4, p.10-11, 01 abr. 2010. Mensal.