UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”

Faculdade de Ciências e Tecnologia – Campus de Presidente Prudente

Programa de Pós-Graduação em Ciência da Computação

Definir posteriormente

Definição e Revisão Bibliográfica do Trabalho da disciplina de Banco de Dados do Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual Paulista.

Bruno Santos de Lima
Leandro Ungari Cayres

Presidente Prudente - SP
Abril - 2019

1 Informações gerais

1.1 Título

- Definir posteriormente...

1.2 Pesquisadores

1. **Nome:** Bruno Santos de Lima
   E-mail: bruno.s.lima@unesp.br
   Titulação: Bacharel
   Instituição: Universidade Estadual Paulista - Unesp
   Página Pessoal: `https://brunoslima.github.io/`
   Currículo Lattes: `http://lattes.cnpq.br/2119168921461476`

2. **Nome:** Leandro Ungari Cayres
   E-mail: leandroungari@gmail.com
   Titulação: Bacharel
   Instituição: Universidade Estadual Paulista - Unesp
   Currículo Lattes: `http://lattes.cnpq.br/5996829502147029`

1.3 Objetivos

O objetivo do trabalho a ser desenvolvido consiste em realizar um estudo de caso comparando o desempenho de bancos de dados relacionais e não-relacionais efetuando operações CRUD implementados em um ambiente NodeJS.

1.4 Metodologia

Inicialmente selecionamos bases de dados já utilizadas em outros trabalhos presentes na literatura. Essas bases de dados são utilizados em nosso estudo de caso no qual realizamos uma comparação de desempenho entre banco de dados relacionais e não relacionais efetuando operações CRUD em um ambiente NodeJS. As bases selecionadas foram...

Definidas as bases de dados que subsidiam a realização do estudo de caso, o próximo passo foi delimitar quais bancos de dados relacionais e não relacionais seriam utilizados em nosso estudo.

2 Banco de Dados Relacional

Os Bases de dados relacionais são baseadas no conjunto de propriedades ACID (atomicidade, consistência, isolamento e durabilidade), entretanto não acomodam características pertencentes

ao Big Data. O principal motivo dessa situação ocorrer é a alta consistência presente nos bancos de dados relacionais. Contudo, no ambiente de Big Data, a alta consistência afeta diretamente os aspectos de disponibilidade e eficiência, que são importantes, devido ao alto volume, variedade e velocidade presente em Big Data (González-Aparicio et al., 2016).

Os Bancos de Dados Relacionais utilizam o modelo relacional em sua composição, foram projetados para atender ao processamento de dados corporativos e tornaram-se a melhor opção para armazenar informações que variam de registros financeiros, dados pessoais, entre outros. No entanto, os requisitos dos usuários e características de hardware têm evoluído desde então para incluir data warehouses, gerenciamento de texto e processamento de fluxo, nos quais tem requisitos diferentes do existentes em processamento de dados tradicionais para negócios. Além disso, a web 2.0 possui novas aplicações que dependem de armazenar e processar grande quantidade de dados e precisa de alta disponibilidade e escalabilidade que adicionou mais desafios para os bancos de dados relacionais. E por causa disso um número crescente de empresas adotou vários tipos de bancos de dados não relacionais, comumente referidos como bancos de dados NoSQL (Mohamed et al., 2014).

Deste modo, os Bancos de dados Relacionais funcionam perfeitamente para manipular um volume limitado de dados. Contudo, ao trabalhar com a análise de dados, dados relacionados a serviços sociais que possuem um volume demasiadamente alto de dados, os bancos de dados relacionais tornam-se muito caros e complexos (Ramesh et al., 2016).

3 Banco de Dados Não-Relacional

Os banco de dados NoSQL, "Não apenas SQL", foram desenvolvidos visando armazenar e processar grandes volumes de dados. Em linhas gerais os bancos de dados NoSQL são livres de esquematizações, lidam com dados não estruturados como e-mail, documentos e mídias sociais de maneira eficiente, suportam replicação e consistência eventual usando critérios de correção, como BASE (Básico, Disponibilidade, estado de consistência, consistência eventual) (Mohamed et al., 2014) (Ramesh et al., 2016).

O termo NoSQL é comumente utilizado para se referir a uma ampla variedade de armazenamentos de dados nos quais as restrições de transação ACID foram relaxadas para permitir melhor dimensionamento e desempenho horizontal (Rafique et al., 2018). Os recursos gerais presentes nos bancos de dados NoSQL são sumarizados em: esquemas menos estruturados, suporte a operações de junção, alta escalabilidade, modelagem de dados simples com linguagem de consulta simples (Ramesh et al., 2016). Os bancos de dados NoSQL foram categorizados em: armazenamento de documentos, famílias de colunas, chave/valor, gráficos e multimodais (González-Aparicio et al., 2016).

3.1 Orientado por Colunas

o modelo de banco de dados não-relacional orientado por colunas foi projetado especialmente visando processar grandes quantidades de dados espalhados por vários servidores. Eles armazenam dados usando um mapa distribuído, multidimensional e esparso, permitindo que um número variável de colunas possa ser armazenado em cada registro, em linhas gerais as colunas são extensões de dados relacionados, podem não suportam associação de tabelas, esse conceito não é utilizado (Patil et al., 2017).

Neste modelo ainda existe o conceito de chaves, entretanto elas apontam para várias colunas. Um conjunto de colunas pode ser formado pelo agrupamento de várias colunas, podendo ser acessadas como colunas únicas ou famílias de colunas (Patil et al., 2017). Dentre os exemplos de bancos de dados orientados a colunas citamos o Cassandra e o HBase.

3.2 Chave-Valor

Dentre os modelos utilizados para banco de dados não-relacionais o mais simples a ser implementado é o modelo de Chave-valor. O conceito utilizado por esse modelo é utilizar uma tabela de hash e fornecer a cada item de dados uma chave exclusiva e um ponteiro, os dados só podem ser consultados usando uma chave específica. Esse modelo é útil para representar dados polimórficos e não estruturados, mas é considerado ineficiente para aplicações voltadas apenas para consultas ou atualizações de partes de um determinado valor (Patil et al., 2017). Os exemplos de bancos de dados de chave-valor são o Amazon SimpleDB, OracleBDB, Redis e flare.

3.3 Armazenamento em Gráfico

O banco de dados não-relacional que utiliza como estrategia modelar uma rede de relacionamentos entre elementos específicos de estruturas gráficas como nós e arestas para representar dados é denominado Banco de Dados de Armazenamento Gráfico. Diferentemente do SQL, que utiliza linguagem de consulta declarativas de alto nível, neste tipo de banco de dados não-relacional, a consulta é específica do modelo de dados. A vantagem adquirida pelo uso do armazenado em gráfico é que o processo de modelagem de dados e relacionamentos entre entidades é bastante simplificado (Patil et al., 2017). Dentre os bancos de dados de armazenamento gráfico conhecidos podemos citar o Neo4j e o Giraph.

3.4 Orientado a Documentos

Os bancos de dados relacionais armazenam os dados em linhas e colunas, em contrapartida os bancos de dados não-relacionais orientados a documentos organizam e armazenam os dados como uma coleção de documentos, nos quais utilizam uma estrutura semelhante a JSON (JavaScript Object Notation) ou XML (Extensible Markup Language).

Uma característica desse tipo de Banco de dados é a possibilidade de naturalmente modelar dados que estão estreitamente relacionados com a programação orientada a objetos. Cada documento é considerado como um objeto, da mesma forma cada documento pode ser um JSON ou um XML no banco de dados orientado a documentos. O conceito de esquema nos bancos de dados de documentos é dinâmico, uma vez que, cada documento pode conter campos distintos um dos outros. Essa característica é útil na modelagem de dados não estruturados e polimórficos. Por fim, os bancos de dados orientados a documentos possibilitam consultas robustas, em que qualquer combinação de campos no documento pode ser realizada visando consultar dados (Patil et al., 2017). Dentre os bancos de dados não-relacionais orientados a documentos podemos citar como exemplo o MongoDB e o CouchDB.

3.4.1 MongoDB

4 Escalabilidade em Banco de Dados

Com o objetivo de garantia das propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade), é mais complexo e desafiador prover alta escalabilidade em um Sistema de Gerenciamento de Banco de Dados Relacional do que outras formas de armazenamento de dados (Fisher e Abbot, 2011). Os Sistemas Gerenciadores de Banco de Dados Relacionais oferecem muitas vantagens sobre o ACID, como operações transacionais, removendo os efeitos de transações de banco de dados parciais decorrentes de falha do sistema, situação inesperada ou uma transação interrompida. No entanto, os Sistemas Gerenciadores de Banco de Dados Relacionais também traz impactos na escalabilidade do sistema ao trabalhar com várias operações simultâneas no banco de dados (Silva et al., 2015).

Os serviços e plataformas mais populares presentes na Internet como Amazon, Google, Facebook, Twitter e E-bay são dependentes do armazenamento e processamento de grandes volumes de dados em uma escala que os Sistemas Gerenciadores de Banco de Dados Relacionais tradicionais tornam-se insuficientes (Pokorny, 2011) (Rafique et al., 2018).

Silva et al. (2015) cita outras soluções para prover escalabilidade em sistemas, como o agrupamento de instâncias de bancos de dados, o uso de bancos de dados não relacionais (Pokorny, 2011), sistemas baseados em MapReduce (Abouzeid et al., 2009), ajuste de desempenho disponível para cada Sistema Gerenciador de Banco de Dados Relacionais, replicação de banco de dados (Kemme e Alonso, 2010), atualização de hardware, entre outros.

4.1 Escalabilidade Vertical

4.2 Escalabilidade Horizontal

5 CRUD

6 Ferramentas de Benchmarking

7 Trabalhos Relacionados 1 - Escalabilidade Horizontal

Ao apresentar o framework YCSB, Cooper et al. (2010) submeteram a benchmarking as bases de dados não relacionais Cassandra, Hbase, Yahoo!'s PNUTS e o sharded MySQL para exemplificar seu uso de maneira prática. Ao realizarem os testes avaliando as camadas de performance e escalabilidade, Cooper et al. (2010) concluem que, assim como suposto pelas descrições dos desenvolvedores, Cassandra e Hbase apresentam maior latência para operações read e menor latência para operações update e write em relação ao PNUTS e MySQL; O PNUTS e Cassandra possuem uma escalabilidade melhor que o HBase quando o número de servidores no cluster aumenta proporcionalmente com a carga de trabalho. Escalabilidade esta, que daremos maior foco ao decorrer do artigo; Cassandra, Hbase e PNUTS são aptos a crescer elasticamente durante a execução de uma carga de trabalho, porém o PNUTS apresenta uma latência melhor e mais estável para tal.

Os autores Jogi e Sinha (2016) comparam o banco de dados relacional MySQL com os bancos não relacionais Cassandra e Hbase quanto a operações heavy write. O teste foi realizado por meio de uma aplicação web REST (Representacional State Transfer) em Java que recebe dados gerados pela ferramenta web nGrinder em formato JSON e os armazenava no banco. O desempenho de cada banco foi calculado pelo nGrinder em termos de TPS (Transações por segundo) ao longo de 10 minutos de testes. Chegou-se à conclusão de que o Cassandra tem o melhor desempenho entre os três bancos com a maior velocidade de escrita. O Hbase por sua vez, se mostrou aproximadamente duas vezes mais rápido que o banco relacional MySQL. Segundo Jogi e Sinha (2016) o Cassandra apresenta tamanho desempenho para operações heavy write por incorporar ao mesmo tempo características do Big Table do Google e do Dynamo da Amazon.

O trabalho de Swaminathan e Elmasri (2016), prioriza a análise da escalabilidade dos bancos de dados Hbase, Cassandra e MongoDB. Para obter os resultados foi utilizado o framework YCSB aplicando as cargas de trabalho 50% read – 50% write, 100% read, 100% blind write, 100% read–modify–write e 100% scan, variando o conjunto de dados a ser manipulado entre 1, 4, 10 e 40 GB a medida que foi aumentado o tamanho do cluster em 2, 3, 5, 6, 12 e 13 nós. O objetivo dos testes foi evidenciar as vantagens e desvantagens de cada ferramenta para um cenário específico dadas as suas diferenças de design.

De acordo com Waage e Wiese (2015), o fato de estas diversas ferramentas não relacionais existentes oferecerem a possibilidade de armazenagem “em nuvem” e, esses ambientes não garantem a confidencialidade dos dados armazenados, é um obstaculo para uma maior adoção das mesmas. Desta forma, Waage e Wiese (2015) propõem que os dados sejam criptografados antes de armazenados em bases de dados na nuvem e, realizam um estudo do impacto que essa alteração no conjunto de dados causa ao desempenho dos bancos de dados Cassandra e Hbase. Tal estudo foi realizado com o uso do framework YCSB onde as workloads foram aplicadas a dados não encriptados e, encriptados usando o algoritmo Advanced Encryption Standard (AES) com chaves de 128, 192 e 256 bits de comprimento, relatando uma redução no desempenho médio do cluster e que, esse custo é relativamente o mesmo independente do tamanho da chave de encriptação.

8 Trabalhos Relacionados 2 - Comparação de Desempenho

9 Bases de Dados

Bibliografia

ABOUZEID, A.; BAJDA-PAWLIKOWSKI, K.; ABADI, D.; SILBERSCHATZ, A.; RASIN, A. Hadoopdb: An architectural hybrid of mapreduce and dbms technologies for analytical workloads. *Proc. VLDB Endow.*, v. 2, n. 1, p. 922–933, 2009.

COOPER, B. F.; SILBERSTEIN, A.; TAM, E.; RAMAKRISHNAN, R.; SEARS, R. Benchmarking cloud serving systems with ycsb. In: *Proceedings of the 1st ACM Symposium on Cloud Computing*, SoCC '10, New York, NY, USA: ACM, 2010, p. 143–154 (*SoCC '10*,).

FISHER, M. T.; ABBOT, M. L. Scalability rules. In: *Boston: Pearson Education*, 2011.

GONZÁLEZ-APARICIO, M. T.; YOUNAS, M.; TUYA, J.; CASADO, R. A new model for testing crud operations in a nosql database. In: *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, 2016, p. 79–86.

JOGI, V. D.; SINHA, A. Performance evaluation of mysql, cassandra and hbase for heavy write operation. In: *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, 2016, p. 586–590.

KEMME, B.; ALONSO, G. Database replication: A tale of research across communities. *Proc. VLDB Endow.*, v. 3, n. 1-2, p. 5–12, 2010.

MOHAMED, M.; G. ALTRAFI, O.; O. ISMAIL, M. Relational vs. nosql databases: A survey. *International Journal of Computer and Information Technology (IJCIT)*, v. 03, p. 598, 2014.

PATIL, M. M.; HANNI, A.; TEJESHWAR, C. H.; PATIL, P. A qualitative analysis of the performance of mongodb vs mysql database based on insertion and retriewal operations using a web/android application to explore load balancing — sharding in mongodb and its advantages. In: *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2017, p. 325–330.

POKORNY, J. Nosql databases: A step to database scalability in web environment. In: *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services*, iiWAS '11, New York, NY, USA: ACM, 2011, p. 278–283 (*iiWAS '11*,).

RAFIQUE, A.; VAN LANDUYT, D.; LAGAISSE, B.; JOOSEN, W. On the performance impact of data access middleware for nosql data stores a study of the trade-off between performance and migration cost. *IEEE Transactions on Cloud Computing*, v. 6, n. 3, p. 843–856, 2018.

RAMESH, D.; KHOSLA, E.; BHUKYA, S. N. Inclusion of e-commerce workflow with nosql dbms: Mongodb document store. In: *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 2016, p. 1–5.

SILVA, L. J. G.; VASCONCELOS, L. G.; SILVA, G.; VASCONCELOS, L. E. G. Towards scalability in systems with write operations in relational databases. In: *2015 12th International Conference on Information Technology - New Generations*, 2015, p. 267–272.

SWAMINATHAN, S. N.; ELMASRI, R. Quantitative analysis of scalable nosql databases. In: *2016 IEEE International Congress on Big Data (BigData Congress)*, 2016, p. 323–326.

WAAGE, T.; WIESE, L. Benchmarking encrypted data storage in hbase and cassandra with ycsb. In: CUPPENS, F.; GARCIA-ALFARO, J.; ZINCIR HEYWOOD, N.; FONG, P. W. L., eds. *Foundations and Practice of Security*, Cham: Springer International Publishing, 2015, p. 311–325.