6ª FEIRA DE INICIAÇÃO CIENTÍFICA E EXTENSÃO
FICE
05 e 06 de setembro

ANÁLISE COMPARATIVA ENTRE MODELOS DISTINTOS DE SGBD UTILIZANDO PROCESSAMENTO EM GPU E NOSQL

José Luiz Bermudez[1] ; Iury Krieger[2] ; Tiago Heineck[3]

INTRODUÇÃO

Segundo previsões de renomadas empresas de consultoria como Gartner, em 2017 cerca de 8,4 bilhões de dispositivos estarão conectados à rede, dados esses que representam um aumento de 31% em relação a 2016 e chegará a 20,4 bilhões até 2020 (GARTNER, 2016). Devido a essa grande quantidade de dispositivos, os números de dados armazenados aumentaram de forma exponencial.

Bancos de dados que apresentam uma escalabilidade horizontal ou seja, permitem a adição de novos “nós” (BEAUMONT, 2014) estão sendo cada vez mais utilizados, característica essa fornecida por bancos de dados não relacionais. Conhecidos como NoSQL (Not Only SQL) essa nova classe de banco de dados apresenta tempos de resposta superiores aos antigos modelos relacionais.

Com o objetivo de mostrar um comparativo entre as tecnologias de banco relacionais e não relacionais, testes salvando cálculos gerados a partir do fractal de Julia foram feitos. Visando aprimorar ainda mais o procedimento, utilizamos do processamento em GPU para a realização dos cálculos, após o retorno dos resultados os dados são gravados em arquivos para serem convertidos em bitmap e em seguida adicionados em seus respectivos bancos para a análise de tempo.

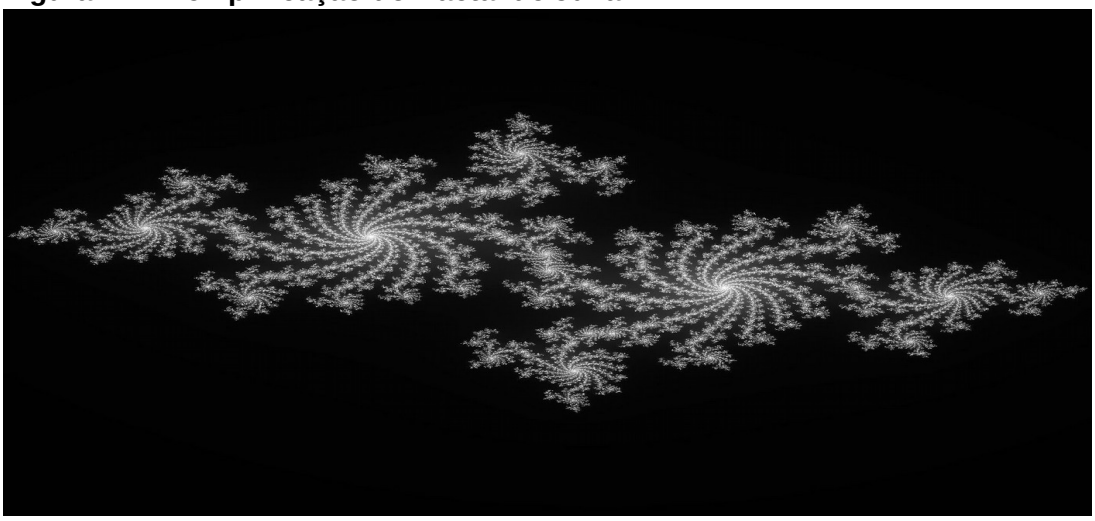
PROCEDIMENTOS METODOLÓGICOS

Durante a realização do processo comparativo diversas tecnologias foram utilizadas. O preenchimento de ambas as bases de dados foram feitos a partir da conversão de imagens geradas pelo fractal de Julia e transformadas em *bitmap*.

[1] Aluno do Instituto Federal Catarinense, Videira. Bacharelado em Ciência da Computação. E-mail: luiz.bermudez27@gmail.com
[2] Aluno do Instituto Federal Catarinense, Videira. Bacharelado em Ciência da Computação. E-mail: iurykrieger96@gmail.com
[3] Professor orientador do Instituto Federal Catarinense, Videira. Bacharelado em Ciência da Computação. E-mail: tiago.heineck@ifc-videira.edu.br

Fractais são figuras geométricas não-euclidianas dotada de auto similaridade, recursividade, holismo e amplificação (KANSO, 2013), ou seja, são figuras que apresentam dimensões distintas de 1, 2 e 3. Devido a sua característica de auto similaridade e recursividade apresentam um padrão em suas figuras e seus fragmentos são similares ao todo, por holismo temos que a soma de seu todo sempre será maior que suas partes, sua característica de amplificação permite que ampliemos a figura com diversas iterações, utilizando assim a computação para o procedimento.

Figura 1 – Exemplificação do fractal de Julia.



Fonte: O autor.

Em razão a grande quantidade de recursos necessários para a criação das imagens utilizamos juntamente do código Python o processamento em GPU. Antes utilizadas como processadores gráficos, hoje são processadores paralelos de propósito geral (NICKOLLS; DALLY, 2010). Devido a grande quantidade de processadores e uma plataforma projetada para ser extremamente rápida no processamento de grandes conjuntos de dados (FAN, 2004) obtivemos resultados significativos para a análise comparativa.

Graças a tecnologia CUDA e a utilização de sua biblioteca para Python, preenchemos as bases de dados analisadas com dados referentes a conversão de dez mil imagens de resolução 8K (7680 x 4620) para *bitmap*. Após a conversão foram escolhidos dois bancos com paradigmas distintos, como representante dos

SGBDs (Sistema Gerenciador de Banco de Dados) utilizamos o MySQL como banco relacional e para representar o não relacional utilizamos MongoDB.

Com o intuito de explorar as diferenças presentes em ambos os bancos, utilizamos características exclusivas no MongoDB como por exemplo a especificação GridFS e o método *Bulk*. Por se tratar de um banco de dados orientado a documentos, alguns fatores importantes precisam ser levado em consideração, inserções no banco são feitas através de *collections*, com um tamanho pré definido de 16MB, era necessário um limite superior para a realização dos testes.

A especificação GridFS tem como função armazenar e recuperar arquivos que ultrapassem esse tamanho limite (PYMONGO,2015). Subdividindo os dados em pequenas chunks, armazenam as informações referentes às demais partes, completando assim o arquivo e ultrapassando o limite imposto.

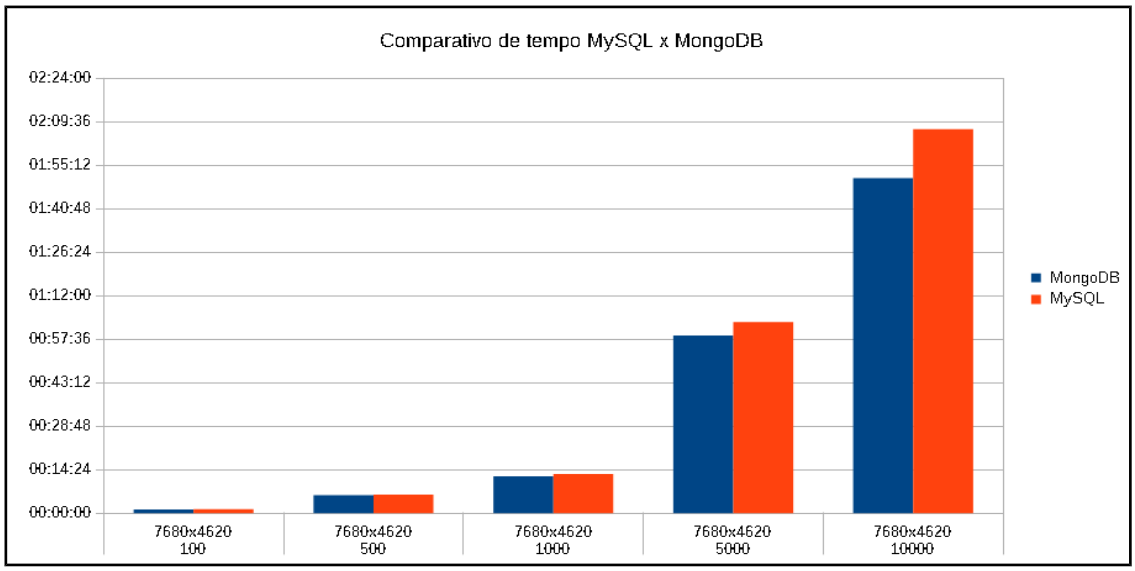
Em virtude da utilização do método Bulk, inclusões em massa são melhor administradas pois afetam apenas uma collection e suportam inserções através de uma passagem de matriz de documentos, utilizando o método "*db.collection.insert()"*. (MONGODB,2017)

Devido a grande quantidade de dados enviados para armazenamento, algumas configurações foram alteradas antes da realização dos testes com o MySQL. Após a alteração na variável responsável por impor um limite máximo no tamanho de pacote transmitido (*max_allowed_packet*), pode-se então dar início aos testes de performance nos dois ambientes.

RESULTADOS E DISCUSSÕES

Com o intuito de apresentar dados verossímeis, os códigos em python tanto para CPU quanto GPU foram executados utilizando a IDE Spyder, mantendo-se sempre as mesmas configurações de sistema operacional e hardware. Inicialmente, por apresentar uma arquitetura orientada a arquivos, a diferença de desempenho não apresenta grande potencial, no entanto, conforme a quantidade de dados aumentam, pode-se visualizar facilmente a crescente melhoria conforme exibido na figura 2.

Figura 2 – Gráfico comparativo de tempo.



Fonte: O autor.

Com um potencial exponencial de melhoria a partir da inclusão de mil imagens em resolução 8k, o banco com características não relacionais apresenta um tempo 45 segundos menor para a realização do armazenamento dos dados, valores esses que se distanciam cada vez mais, apresentando um total de dezesseis minutos de diferença quando armazenado dez vezes o valor inicial, dez mil imagens utilizando MongoDB.

CONSIDERAÇÕES FINAIS

Após a análise dos dados apresentados, pode-se constatar que graças aos diferentes paradigmas encontrados na computação, existem inúmeras soluções que visam a obtenção de melhores resultados. Uma vez que exista um empecilho, outras tecnologias podem lhe auxiliar, inicialmente apresentando dificuldades para a realização dos cálculos devido a baixa capacidade de processamento encontrado na CPU, a utilização de uma GPU nos auxiliou na criação de imagens geradas a partir de fractais, sendo elas utilizadas para a avaliação de desempenho dos diferentes SGBDs.

Através do gráfico exibido no artigo, pode-se concluir alguns pontos bem específicos. Uma vez que necessite realizar grandes transações é inquestionável, neste caso a eficácia do banco com o paradigma NoSQL, a utilização do MongoDB, por se tratar de um estrutura orientada a arquivos, apresenta resposta com eficiência exponencial conforme o aumento da base. Devido a popularização das bases relacionais, muitas vezes a preferência em se manter os costumes podem impedir resultados ainda mais favoráveis, apesar de não apresentar grandes variações de tempo em pequenas bases, o MongoDB apresenta uma estrutura de fácil compreensão e simplicidade, sendo assim uma ótima escolha para fins não só didáticos mas também comerciais.

REFERÊNCIAS

BEAUMONT, David. **How to explain vertical and horizontal scaling in the cloud**. Disponível em: <https://www.ibm.com/blogs/cloud-computing/2014/04/explain-vertical-horizontal-scaling-cloud/>. Acesso em: 22 maio 2017.

FAN, Zhe et al. **GPU cluster for high performance computing**. In: Supercomputing, 2004. Proceedings of the ACM/IEEE SC2004 Conference. IEEE, 2004. p. 47-47

GARTNER. **Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016**. Disponível em: <http://www.gartner.com/newsroom/id/3598917>. Acesso em: 22 maio 2017

KANSO, Mustafá Ali. **Fractais: afinal, o que são?** Disponível em: <http://hypescience.com/fractais-o-que-sao/>. Acesso em: 23 maio 2017.

MONGODB. **Bulk Write Operations**. Disponível em: <https://docs.mongodb.com/v3.0/core/bulk-write-operations/>. Acesso em: 24 maio 2017.

NICKOLLS, John; DALLY, William J. **The GPU computing era**. IEEE micro, v. 30, n. 2, 2010.

PYMONGO. **Gridfs: Tools for working with GridFS**. Disponível em: <http://api.mongodb.com/python/current/api/gridfs/index.html#module-gridfs>. Acesso em: 24 maio 2017.