Implementação de Banco de Dados Distribuídos em MySQL para o portal (En)cena

Maria do Carmo Brito da Silva[1], Muriel Souza da Cruz[1], José Henrique Coelho Brandão[1], Pedro Sousa Silva[1], Madianita Bogo[1]

[1] Departamento de Computação - Centro Universitário Luterano de Palmas - CEULP/ULBRA, Av. Joaquim Teotônio Segurado, 1501 – 77000-900 – Plano Diretor Sul, Palmas – TO – Brasil

{caramellomari, jhbc007, pedrossc.88, madianitab} @gmail.com muriel-souza@live.com

Resumo. *O presente trabalho aborda conceitos de bancos de dados distribuídos, bem como apresenta uma proposta de banco de dados distribuído para o portal (En)Cena do CEULP/ULBRA. São mostrados os materiais e procedimentos necessários para disponibilizar o banco de dados de forma distribuída usando o MySQL.*

1. Introdução

Tradicionalmente, os Bancos de Dados se encontram em um único espaço físico, ou seja, possui uma arquitetura centralizada. Normalmente, nesta arquitetura acontecem alguns problemas como, por exemplo, uma falha no servidor do Banco de Dados pode ocasionar a inoperabilidade de todo o sistema (SILVA, 2012, p. 10).

Muitas empresas convivem com o problema de lentidão no momento de inserção e recuperação de dados no sistema gerenciador de bancos de dados. Normalmente, isso ocorre devido às dificuldades não vistas no instante da concepção do modelo de banco de dados (RIBEIRO, 2010).

O (En)Cena: a saúde mental em movimento é um portal voltado para produções textuais, imagéticas e sonoras referentes ao tema da loucura. O portal é idealizado pelos cursos de Comunicação Social, Psicologia e Sistemas de Informação do CEULP/ULBRA. Ele visa estimular essas produções,

> "em especial nos serviços de saúde, pois a proposta partiu do pressuposto que há muitas experiências vividas em serviços de saúde que condizem com a proposta da Reforma Psiquiátrica e da Luta Antimanicomial, mas que não são publicizadas" (Encena, 2018).

O portal é muito requisitado, tendo em média 45.000 acessos por mês, deixando o processo de consulta ao banco de dados lento e, até indisponível em algum momento. A fim de sanar problemas como a demora de acesso aos dados, o objetivo deste trabalho é otimizar o processo de consultas ao Banco MySQL do (En)Cena através de um sistema de Banco de Dados Distribuídos.

Eller (1997) diz que um banco de dados distribuído é uma coleção de dados que pertencem de forma lógica ao mesmo sistema, porém, distribuídos sobre pontos de uma rede de computadores.

Um banco de dados distribuído deve ter

> "Autonomia Local, Independência dos Sítios, Operação Contínua, Independência de Local, Independência de Fragmentação, Independência de Replicação, Processamento de Consultas Distribuídas, Gerenciamento de Transações Distribuídas, Independência de Máquina, Independência do Sistema Operacional, Independência de Rede de Comunicação e Independência de SGBD" (Euller, 1997).

Este trabalho está organizado da seguinte forma: no capítulo 2 é apresentado o referencial teórico referente à área de Banco de Dados Distribuído; no capítulo 3 são apresentados os materiais e métodos utilizados para a elaboração deste trabalho; no capítulo 4

ISSN: 2447-0767

são mostrados os resultados e discussões; no capítulo 5 serão apresentadas as considerações finais deste trabalho e, por fim, são listadas as referências bibliográficas

2. Referencial Teórico

O Banco de Dados Distribuído tem seus dados armazenados em diversos computadores interconectados, estes variam de tamanho ou de função e podem possuir uma administração separada. Oliveira (2007) afirma que a distribuição física depende de arquitetura de rede, se local ou de longa distância, ou se as estações são fixas ou móveis, entre outros fatores. "Resumidamente, o banco de dados distribuído distribui dados" (Oliveira, 2007, p. 28).

> "O que caracteriza um Banco de Dados Distribuído (BDD) é a existência de dois ou mais sites (ou nós), interconectados de forma permanente ou não, sendo que cada nó deve possuir um Sistema de Gerenciamento de Banco de Dados (SGBD) que compartilha um esquema global comum, embora algumas relações não precisem ser idênticas em todos os nós" (Oliveira, 2007).

2.1. Arquitetura de SBDDs

A arquitetura de um BDD apresenta os seus componentes e as suas atribuições, definindo sua estrutura. Também, mostra como acontece o relacionamento entre os componentes. Para um entendimento mais preciso de um BDD, a Figura 1 apresenta os componentes que fazem parte de sua arquitetura.

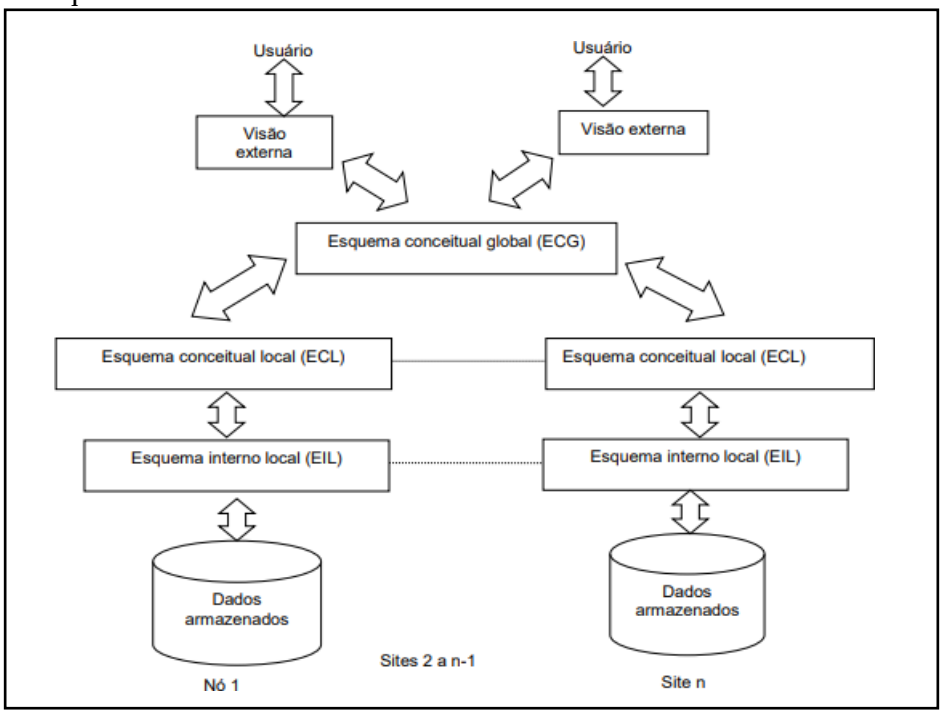


Figura 1: Arquitetura de um BDD (SILVA, 2012, p. 17)

Silva (2012) diz que o componente Esquema Conceitual Global (ECG) responde pela transparência de rede para os usuários, a partir daí é feito o direcionamento para vários componentes de Esquema Conceitual Local (ECL). "Cada ECL descreve o banco de dados local e este define o Esquema Interno Local (EIL) para cada nó" (Silva, 2012, p. 17).

Assim, nota-se que mesmo com dados do banco divididos em vários nós de uma rede de computadores, para o usuário fica a sensação de que os mesmos se encontram centralizados.

2.2. Distribuição dos Dados

Nesta seção são tratados os aspectos atribuídos à distribuição dos dados. No tópico Tipos de Fragmentação de Dados (2.2.1) é vista a fragmentação de forma horizontal, vertical e mista.

O tópico Transparência de Rede (2.2.2) trata da capacidade que o sistema tem não deixar visível ao usuário detalhes da distribuição.

2.2.1. Tipos de Fragmentação de Dados

Oliveira (2007) diz que a fragmentação orientada pelo conteúdo dos dados, define quais dados serão distribuídos e a como será feita a distribuição. Esse tipo de fragmentação divide o banco de dados em várias partes, distribuindo entres sites participantes do BDD. A fragmentação pode ser horizontal, vertical ou mista, conforme Figura 2.

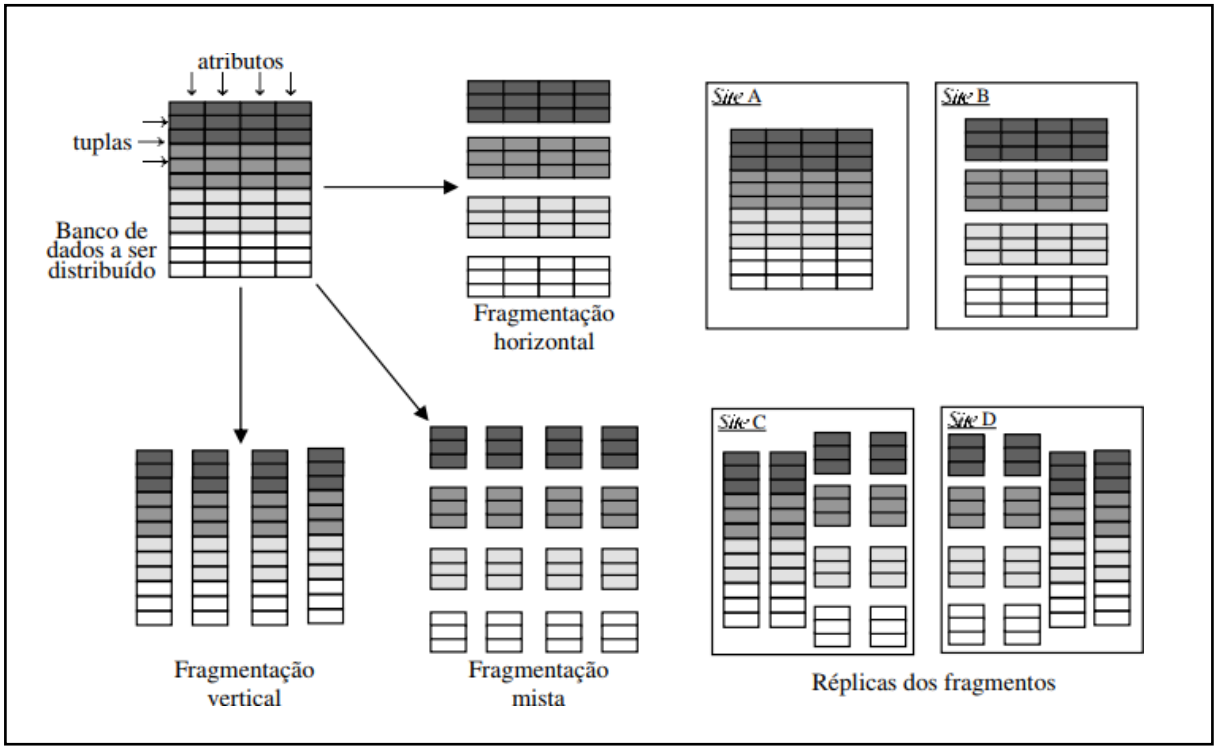


Figura 2: Tipos de Fragmentação de Dados (OLIVEIRA, 2007, p. 34)

Na definição de fragmentos, conforme Mesquita (1998), existem algumas regras que devem ser seguidas:

- Completude - Quando os dados são mapeados em fragmentos, isto é, quando um valor de atributo pertencente à relação global deve necessariamente pertencer a algum fragmento;
- Reconstrução - "A reconstrução da relação global a partir dos seus fragmentos deve ser possível" (MESQUITA, 1998, p. 34);
- Disjunção - é necessário que os fragmentos sejam disjuntos para evitar a repetição de dados.

Na fragmentação horizontal o banco de dados é dividido em tuplas de uma relação global de subconjuntos, assim cada conjunto resultante forma-se um BD. Ocorre o mesmo na fragmentação vertical, mas, em relação a um conjunto de atributos. "A fragmentação é realizada considerando-se que os atributos a serem agrupados têm características em comum" (MESQUITA, 1998, p. 35).

Segundo Mesquita (1998) a fragmentação mista é um tipo de fragmentação, nela os fragmentos são resultantes da aplicação de operações de fragmentação sobre fragmentos, e não sobre relações globais.

A replicação do dado ocorre quando um mesmo conjunto de tuplas ou atributos aparece repetido em dois ou mais banco de dados. Também, pode fragmentar e replicar a réplica do fragmento.

ISSN: 2447-0767

2.2.2. Transparência de Rede

Segundo Oliveira (2007), transparência de rede é a capacidade que o sistema tem de "esconder" do usuário detalhes da distribuição. Usuário do sistema ou desenvolvedor de aplicações devem serem isentos com preocupações relacionadas à forma como os dados foram fragmentados e replicados, como são identificados e onde localizados. O sistema deve ser configurado de forma que todos os serviços de consulta e atualizações sejam fornecidos, alcançando os dados onde estiverem.

Em sistemas com desconexão programada são difíceis de estabelecer a transparência de rede, em algumas situações chega a ser impossível. Isso pode se referir à coerência entre consulta e atualizações.

> "Por exemplo, um usuário desconectado pode fazer uma consulta em sua base de dados, enquanto outro site atualiza a réplica dessa base. Quando houver a conciliação, o resultado da mesma consulta seria outro e o usuário pode ter ficado com uma informação ultrapassada. Dependendo da situação, isso pode não ser tão grave, levando em conta que mesmo em um ambiente 'conectado', o usuário poderia fazer a consulta um pouco antes de outro site alterar o dado e, assim, ficaria com a informação desatualizada da mesma forma" (OLIVEIRA, 2007, p. 35).

As consultas globais também não podem serem atingidas em sistema com desconexão programada. O sistema submete a consulta a todos os sites em ambientes conectados, assim, gerando a união dos resultados e apresentando ao usuário. É impossível isso ser feito em ambientes desconectados. A única alternativa, não muito viável, seria disponibilizar uma réplica completa das bases de dados.

3. Metodologia

Nesta seção serão apresentadas ferramentas e tecnologias para o desenvolvimento deste projeto além dos métodos utilizados para alcançar o resultado obtido.

3.1. Materiais e Métodos

Para este projeto foi utilizado as seguintes ferramentas:

- MySQL: Um sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL como interface. Utilizado como o banco de dados distribuído do projeto.
- ProxySQL: Servidor proxy para atendimento de solicitações do cluster do banco de dados. Utilizado para gerenciar o fluxo de carga ao banco de dados distribuído.
- Ubuntu 18.4.1: Sistema Operacional de código aberto construído a partir do núcleo Linux. Utilizado em todas as instâncias do servidor.
- JMeter: Ferramenta utilizada para realizar testes de carga.

Quanto ao ambiente de testes, a seguinte arquitetura foi montada:

- Servidor Web: Responsável por hospedar o site e receber as solicitações http.
- Servidor ProxySQL: Responsável por receber as solicitações SQL do Servidor Web e dividir entres os servidores de banco de dados.
- Servidor Node Master: Hospeda o banco de dados MySQL, responsável pela armazenagem de dados e execução de instruções de escrita e leitura.
- Servidor Node Slave 2 e 3: Hospedam um espelho do banco do Servidor Node Master e recebe apenas solicitações de leitura.
- Servidor de Testes Diretos: Hospeda uma versão do Servidor Node Master que agirá sem a distribuição de dados, recebendo diretamente solicitações de leitura e escrita.

O primeiro passo foi criar o cluster, ou seja, um sistema de redundância do banco de dados. Para isso, três servidores são utilizados. O primeiro, chamado de Master, é um banco de dados já existente, os outros dois, Slave1 e Slave2, são utilizados em sua replicação.

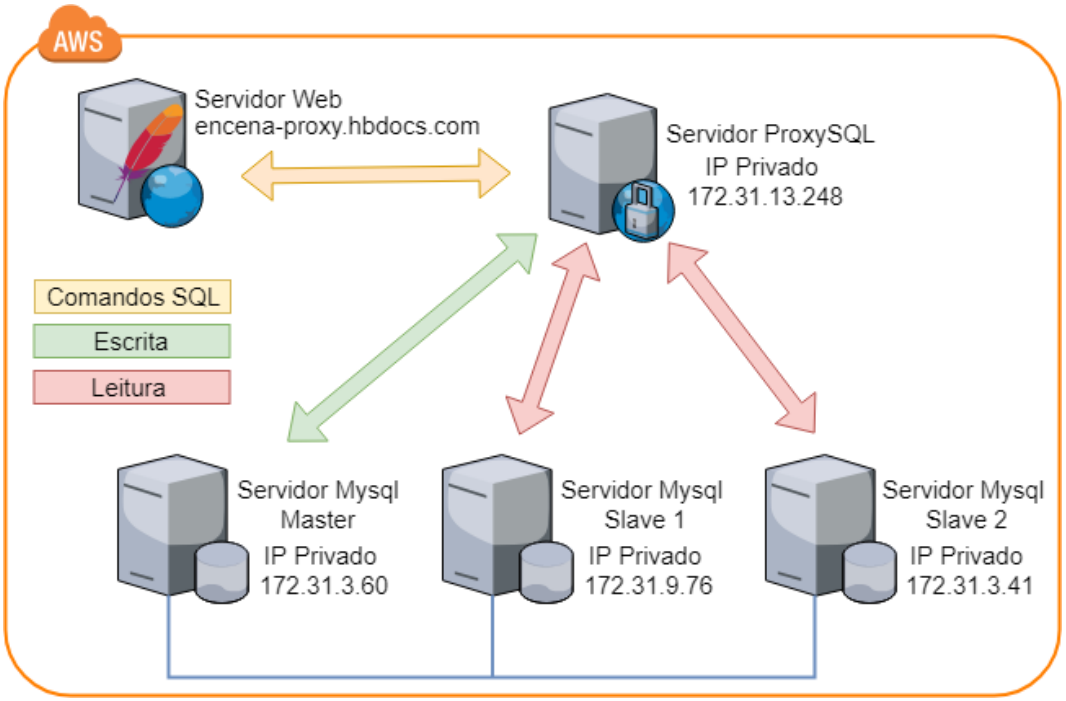
Esta arquitetura em grupo provê replicação, consistência de dados e ordenação de buscas, de modo que qualquer operação de escrita é feita apenas sobre o banco de dados

Master que atualiza os servidores secundários, utilizados para leitura apenas, além de garantir a transparência, onde o cliente não percebe as interações entre os diversos servidores.

4. Desenvolvimento

Este trabalho teve o objetivo de verificar a eficiência ao se utilizar uma arquitetura distribuída sobre uma aplicação web quanto a questão de desempenho sob alta demanda. Pela origem da aplicação, um website, é esperada uma grande quantidade de solicitações de leitura no banco de dados, o que pode causar lentidão e instabilidade. Por tanto é necessário garantir que os servidores sejam acessados de forma equilibrada.

4.1. Desenvolvimento



Neste ponto um quarto servidor utiliza a ferramenta ProxySQL funcionando como intermediário entre a aplicação e os servidores. Sua função é receber os comandos SQL e dividir entre os servidores, assim comandos de escrita são enviadas ao servidor master e os comandos de leitura para os servidores slave, levando em consideração sua capacidade e carga no momento.

```
proxysql> SELECT hostgroup_id, hostname, status FROM mysql_servers ORDER BY hostgroup_id;
+--------------+-------------+--------+
| hostgroup_id | hostname    | status |
+--------------+-------------+--------+
| 15           | 172.31.3.60 | ONLINE |
| 20           | 172.31.3.41 | ONLINE |
| 20           | 172.31.3.60 | ONLINE |
| 20           | 172.31.9.76 | ONLINE |
+--------------+-------------+--------+
4 rows in set (0.00 sec)
```

Finalmente, um quinto servidor contém a aplicação que é disponibilizada via web aos clientes. As solicitações deste servidor são encaminhadas ao servidor de proxy que envia para os seus servidores distintos e apresenta os resultados aos usuários pelo link encena-proxy.hbdocs.com.

Para o cliente, que envia a solicitação para o servidor Web, não é possível identificar as conexões internas do sistema distribuído, recebendo suas solicitações pelo mesmo servidor Web, garantindo sua transparência.

ISSN: 2447-0767

A figura a seguir apresenta a configuração final do cluster e os relacionamentos entre os seus servidores. Um último servidor é instalado para criar um comparativo, neste as solicitações não serão distribuídas, e processadas por uma única máquina, acessada pelo link encena.hbdocs.com.

4.2. Testes de Desempenho

Com a arquitetura montada foi possível iniciar os testes de acesso e analisar os resultados com a ferramenta JMeter. O principal objetivo deste teste é identificar se a arquitetura distribuída, apesar de apresentar mais passos para responder um cliente, é realmente mais rápida que a direta, onde todas as requisições são recebidas, executadas e respondidas por um mesmo servidor.

Para realizar os testes de velocidade, o JMeter foi configurado para realizar conexões em ambos os servidores, distribuído e não-distribuído. Por conta da memória limitada dos servidores de teste, foram realizadas simulações de 20 usuários realizando 2 requisições em 1 segundos, totalizando 40 requisições simultâneas.

4.3. Resultados

Os resultados são apresentados pelo JMeter, com a latência, ou o tempo de resposta de cada solicitação, em milissegundos, e ao final, a média da latência, também em milissegundos.

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum
Requisição ...	40	9341	8953	13494	14360	15363	4098	15363
TOTAL	40	9341	8953	13494	14360	15363	4098	15363

Figura 5: Resultado agregado das requisições do encena.hbdocs.com.

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum
Requisição ...	40	6667	6441	10063	10355	11310	2071	11310
TOTAL	40	6667	6441	10063	10355	11310	2071	11310

Figura 6: Resultado agregado das requisições do encena-proxy.hbdocs.com.

A Figura 5 apresenta os resultados das requisições sem a utilização de um sistema de Banco de Dados Distribuídos. Nela é possível verificar que a média do tempo de resposta às quarenta requisições é de 8.953ms, que em números aproximados, seriam nove segundos. Já a aplicação com a distribuição de Banco de Dados (Figura 6) apresenta uma média de 6.441ms para atender as quarenta requisições, que em termos aproximados seriam, seis segundos e meio. Uma diferença de 2.512ms, que em termos de porcentagem representa um desempenho superior de 28% do Banco de Dados distribuídos em relação ao Banco de Dados sem a distribuição de consultas.

Para um alguns usuários finais, com base nesses dados, uma diferença média aproximada de dois segundos e meio poderiam não ser tão perceptível, mas a medida que as requisições são aumentadas, o "gargalo" que o Banco de Dados não distribuído apresenta(devido a centralização de todas as requisições) poderiam impactar de forma significativa o tempo para execução de consultas no Banco de Dados.

Além disso, outro dado importante fornecido pelo JMeter é o tempo mínimo (requisição entregue em menor tempo) e tempo máximo (requisição com entrega mais demorada). Com base nos dados fornecidos para quarenta requisições, no Banco de Dados Distribuídos (Figura 6) a execução e entrega de requisição mais rápida foi de 2.071ms, enquanto no Banco de Dados não distribuído (Figura 5) foi de 4.098ms, uma diferença de 49,5%, em relação ao primeiro.

Já em relação ao tempo máximo de respostas às solicitações, o banco de dados não distribuído (Figura 5) levou 15.363ms para entrega mais demorada, uma diferença de 4.030ms em relação ao Banco de Dados Distribuídos (Figura 6) que demorou 11.310ms. Em termos percentuais, essa diferença de maior tempo para resposta a requisição foi de 26,3%. Indicando que o Banco de Dados não distribuído demora aproximadamente quatro

segundos a mais na resposta mais demorada.

Com isso, é possível identificar que a distribuição de requisições no Banco de Dados permite respostas mais rápidas, uma vez que as requisições estão distribuídas entre diversos servidores. A partir dessas informações é possível concluir que a implantação de um Banco de Dados Distribuídos no portal (En)Cena é muito positivo, levando em consideração a carga e a eficiências das requisições e entregas, respectivamente. Essa positividade será refletida tanto para os usuários que receberão respostas mais rápidas às suas requisições, tanto, o sistema que ficará mais otimizado para consultas e manipulação dos dados.

5. Desenvolvimento

Este trabalho apresentou os conceitos necessários e as práticas realizadas para a implementação do Banco de Dados Distribuídos para plataforma online (En)Cena, bem como, suas principais características e desenvolvimento da estrutura. Tendo como base o conhecimento teórico obtido por meio das bibliografias estudadas, foi possível compreender o funcionamento e a necessidade dos Bancos de Dados Distribuídos para agilizar consultas e inserções de dados no Banco.

Para a implementação do banco foi necessário a definição de uma estrutura de servidores e suas relações. Pois, para que fosse realizada consultas em Bancos de Dados distribuídos de maneira transparente e decidisse quais servidores MySQL receberiam às requisições de consultas ou inserções, por meio de um servidor proxySQL, por exemplo.

Com resultados promissores, em alguns momentos chegando a uma diferença de aproximadamente 50% a favor do Banco de Dados Distribuídos. A distribuição de consultas mostrou eficiente em relação a questões de balanceamento de carga que consequentemente acaba retornando consultas mais rápidas aos usuários. Porém, o custo para manutenção de um Banco de Dados Distribuídos é relativamente grande o que pode não ser muito aprazível financeiramente para o (En)Cena atualmente.

Referências

MESQUITA, Eduardo José Soler. Projeto de Dados em Bancos de Dados Distribuídos. 1998. 87 p. Dissertação (Matemática Aplicada)- Instituto de Matemática e Estatística da USP, São Paulo - SP, 1998.

OLIVEIRA, Vinicius Fernandes de. Especificação e implementação de um modelo assíncrono para replicação, propagação e conciliação de bases de dados distribuídas. 2007. 138 p. Dissertação de Mestrado (Pós-Graduação em Engenharia Elétrica)- UFRN, Natal-RN, 2007.

RIBEIRO, Robson Afrânio. Performance do banco de dados MySQL. 2010. Disponível em: <devmedia.com.br/performance-do-banco-de-dados-mysql/18508>. Acesso em: 24 nov. 2018.

SAÚDE MENTAL EM MOVIMENTO, ENCENA. (En)Cena: a saúde mental em movimento. 2018. Disponível em: <http://encenasaudemental.net/percurso/>. Acesso em: 11 nov. 2018.

SILVA, Marcos Paulo Honorato da Silva. Criação e Implantação de Bancos de Dados Distribuído para a Fábrica de Software do CEULP/ULBRA. 2012. 64 p. TCC (Sistemas de Informação) - CEULP/ULBRA, Palmas-TO, 2012.

ISSN: 2447-0767