

Arquitetura em cluster para aplicação em saúde com o modelo de banco de dados distribuído

Cluster Architecture for Health Application with the Distributed Database Model

José da Silva Azanha Neto¹
Débora Vigília Canne²
Antônio Andrade dos Santos³
Luís Antônio Coppa Ybarra⁴
Márcia Aparecida Sílvia Bissaco⁵
Jorge Rodolfo Beingolea Garay⁶

Resumo: Os modelos atuais de sistemas em saúde são imprescindíveis para a tomada de decisão médica e para a disseminação do conhecimento em saúde. O Prontuário Eletrônico do Paciente é um sistema que mantém grande volume de dados para acesso do paciente e de profissionais da saúde. O Prontuário Eletrônico Único do Paciente é um modelo distribuído de dados. Este modelo opera em grade que oferece maior disponibilidade, confiabilidade e a falhas, submetido a testes de prova de conceito que indicaram o sucesso na implementação. Entretanto, a instância do banco de dados em cada federação na grade é centralizada e deve sofrer com a concorrência. O objetivo é propor uma arquitetura em *cluster* para lidar com a concorrência no banco de dados. Foram obtidos melhores tempos de resposta com 73% para listar usuários e 35% para listar pacientes ao adotar o banco de dados em *cluster* em relação ao centralizado.

Palavras-chave: Prontuário do Paciente; Computação em Grade; Computação em *Cluster*; Balanceamento de Carga; Tolerância a Falhas.

Abstract: The current models of health systems are essential for medical decision-making and the dissemination of health knowledge. The Electronic Health Record is a system that maintains a large volume of data for patient and healthcare access. The Unique Electronic Health Record is a distributed model of data. This model operates on a grid that offers greater availability, reliability and fault tolerance, subject to proof of concept tests that indicate the success in the implementation. However, the database instance in each federation in the grid is centralized and must suffer concurrently. The goal is to propose a clustered architecture to handle the competition in the database. Better response times were obtained with 73% to list users and 35% to list patients when adopting the cluster database in relation to the centralized one.

Keywords: Electronic Health Records; Grid Computing; Cluster Computing; Load Balancing; Fault Tolerance.

Introdução

¹ Professor e gestor na Universidade Nove de Julho (UNINOVE). Rua Vergueiro, 235/249, São Paulo (SP), CEP 01504-000. E-mail: jose.azanha@gmail.com

² Cursando Tecnologia em Análise em Desenvolvimento de Sistemas na Universidade Nove de Julho (UNINOVE). Rua Vergueiro, 235/249, São Paulo (SP), CEP 01504-000. E-mail: debora.canne@uni9.pro.br

³ Professor na Universidade Nove de Julho (UNINOVE). Rua Vergueiro, 235/249, São Paulo (SP), CEP 01504-000. E-mail: uni9aas@gmail.com

⁴ Professor titular na Universidade Nove de Julho (UNINOVE). Rua Vergueiro, 235/249, São Paulo (SP), CEP 01504-000. E-mail: lybarra@uni9.pro.br

⁵ Professora e pesquisadora na Universidade de Mogi das Cruzes (UMC). Av. Dr. Cândido Xavier de Almeida Souza, 200, Mogi das Cruzes (SP), CEP 08780-911. E-mail: marciab@umc.br

⁶ Professor e pesquisador do Núcleo de Pesquisas Tecnológicas da Universidade de Mogi das Cruzes (NPT-UMC). Av. Dr. Cândido Xavier de Almeida Souza, 200, Mogi das Cruzes (SP), CEP 08780-911. E-mail: jorgegaray@umc.br

Os modelos de sistemas atuais em saúde são imprescindíveis para o acesso a tomada de decisão médica e para a disseminação do conhecimento em várias áreas de estudos. O Prontuário Eletrônico do Paciente (PEP) ou Registro Eletrônico em Saúde (RES) é um sistema de informação em saúde que mantém grande volume de dados para acesso do paciente e dos profissionais da área de saúde (COOREVITS *et al.*, 2013; KING, 2014). O volume de dados associado ao grande número de acessos pode gerar problemas de indisponibilidade do PEP causando atrasos em atendimentos de urgência, por exemplo. Devido ao grande volume de dados armazenado em PEPs que, segundo o trabalho de Azanha Neto (*et al.*, 2018), pode alcançar até 5GB de armazenamento em disco por dia/paciente. Esse volume inclui dados nos formatos de texto, imagens e exames.

É nesse sentido que na proposta do modelo de banco de dados para o Prontuário Eletrônico do Paciente (PEP) teve-se por objetivo sanar parte do problema relacionado ao volume de dados, composto por três níveis de informação. O primeiro nível, formado pela síntese do prontuário, contém um pequeno volume de dados (apenas acessos às informações básicas do paciente, como resultados de exames e consultas recentes e sem imagens de exames), pois tratam-se dos últimos atendimentos, medicamentos prescritos, alergias e doenças crônicas. No nível dois, tem-se a síntese do prontuário estendido com informações complementares do paciente. No nível três, tem-se o prontuário completo do paciente (Azanha Neto *et al.*, 2018).

A síntese está distribuída em todas as Organizações Virtuais (OV – grupo de computadores ou *data centers* integrados ou nuvens) e em instâncias de dados da grade computacional. Os demais níveis (dois e três) ficam em banco de dados (BD) locais da OV e são enviados para outras OVs sob demanda do médico. Assim, o modelo distribui os dados em uma arquitetura denominada *grade computacional* (GC) (Azanha Neto *et al.*, 2018). A GC distribui os modelos de dados em OVs, que estão geograficamente distantes e em domínios administrativos diferentes. Cada OV é responsável pela execução paralela de requisições do usuário através de computadores dedicados para o processamento de serviços web. Isto é, o modelo do PEUP pode ser implementado no modelo de nuvem para atender as demandas de serviços web.

Apesar do modelo do PEUP possuir um modelo de dados distribuído (MDD) e com suporte da arquitetura em GC, cada instância do BD é executada em um servidor centralizado em cada OV. Dessa forma, os servidores da GC processam paralelamente os serviços web, porém, cada instrução SQL é executada em fila no servidor BD da OV, ou seja, gerando um gargalo no modelo (VON WINCKLER, 2014; SHAMS *et al.*, 2010). O BD utilizado para esta pesquisa possui 1.700.000 registros, contendo dados fictícios em saúde para validação.

Os testes de prova de conceito realizados visam garantir a funcionalidade e o ganho em desempenho e segurança, ao se adotar um modelo como o PEUP, que propõe a viabilidade do acesso a um número grande de usuários em saúde. Isso implica lidar com a concorrência de recursos e manter o ambiente confiável e disponível. Além disso, questões ligadas à tolerância às falhas e ao desempenho devem ser consideradas (AOKI, CARVALHO, 2012). Assim, neste trabalho busca-se avaliar o desempenho do modelo atual do PEUP com um BD centralizado versus a arquitetura proposta com um *cluster* de BD.

Algumas propostas importantes foram encontradas na literatura sobre a aplicação da computação distribuída em sistemas para a saúde, nas quais se destaca a estrutura de nuvem voltada para a saúde pública com o Smarter Public Health Prevention System. O referido sistema disponibiliza relatórios em tempo real para a tomada de decisão médica, por meio de uma interface para dispositivos móveis, o que permite que pacientes tenham fácil acesso aos serviços e informações necessárias de saúde pessoal (DA SILVA, 2016).

Entretanto, é possível concluir que a Proposta pode trazer sérios problemas, quando se tratar da transmissão de volumes expressivos de dados em redes de longa distância, o que deve ser compensado com uma boa arquitetura e, em alguns casos, devem ser processados em horários alternativos. No entanto, a literatura apresenta alguns avanços, como o uso do Globus Toolkit, no qual são executados protocolos otimizados para a transferência de dados em redes de longa distância. O Globus Toolkit utiliza o GridFTP (Transferência de Dados em Grade Computacional), em que a autenticação ocorre via SSH instanciado pelo Grid Security Infrastructure, usando o conceito de *streaming* e protocolos de comunicação *peer-to-peer*, otimizados para a transmissão de dados biológicos como *bitorrents* (DA SILVA, 2016).

O E-Health Capacity Building é um ambiente com políticas e desenvolvimento

institucional, que permite participação da comunidade, formação de recursos humanos e grandes desafios para a capacitação em saúde eletrônica para os profissionais de saúde qualificados. A nuvem computacional permite que hospedeiros de serviços de saúde executem serviços remotos aos prontuários, para que qualquer usuário autorizado obtenha acesso a esses dados em qualquer lugar do mundo através apenas da internet. Essa proposta permite o compartilhamento de dados e colaboração entre profissionais da saúde, clínicas de imagem, radiologistas e outros centros clínicos (CHOWDHARY, 2011).

O Inter-Operability Using Universal Standards permite que aplicativos clínicos possam ser construídos utilizando as plataformas de imagens médicas heterogêneas, onde cada versão específica à plataforma de imagem foi gerada. O desafio da plataforma é incluir grande interoperabilidade do *design* do teste em múltiplas plataformas de sistemas operacionais e hardware, programação e linguagens de *script* (CHOWDHARY, 2011).

O objetivo do presente estudo é propor uma arquitetura em *cluster* para lidar com a concorrência no banco de dados.

Materiais e métodos

Os modelos de sistemas de saúde atuais são de extrema relevância para a tomada de decisão dos médicos e para a disseminação do conhecimento na área.

O MDD do PEUP proposto anteriormente está seguindo alguns requisitos, mesmo que parcialmente, do Manual de Certificação da Sociedade Brasileira em Informática em Saúde relacionados como Nível de Garantia de Segurança 1 e 2, Requisitos de Estrutura e Conteúdo, e Requisitos de Funcionalidades (AZANHA NETO *et al.*, 2018).

De acordo com Sousa (*et al.*, 2011), descentralização e o compartilhamento de recursos do ambiente computacional são estratégias importantes para lidar com questões de alta disponibilidade, desempenho, escalabilidade e confiabilidade. Alinhada com essas estratégias, a arquitetura distribuída favorece os sistemas em saúde no qual possuem altos volumes de acessos, principalmente àqueles que estão disponíveis pela internet ou em redes privadas de longa distância. Assim, a distribuição do PEUP depende de uma arquitetura sólida, que favoreça o acesso dos

usuários, compartilhe recursos de processamento e tenha réplicas para lidar com a concorrência. Para tanto, o modelo de distribuição deve viabilizar a descentralização do servidor de aplicação e do BD, por estarem operando em rede de longa distância, e permitir que os BDs se comuniquem para garantir a distribuição dos dados de pacientes e usuários em todos os modelos de dados.

Para atender aos requisitos acima descritos, no MDD do PEUP foi utilizada a arquitetura em grade computacional, que pode ser implementada juntamente com o modelo de nuvem computacional. Essas arquiteturas possuem alto grau de heterogeneidade, permitindo agregar sob o mesmo ambiente: linguagens de programação, redes, sistemas operacionais e hardwares diferentes. Assim, a grade adotada no MDD do PEUP foi o JAVA Parellel Processing Framework (JPPF) (XIONG, WANG, XU, 2010; SINHA, 2016). O JPPF suportou todas as requisições de Hyper Text Transfer Protocol (HTTP), que chegaram ao servidor de aplicação GlassFish Server com recursos de processamento. Para manter os dados do PEUP, foi implantado o servidor de BD MySQL 5.x, com replicação de dados Master/Master entre todas as instâncias de BD da grade. A aplicação do PEUP foi desenvolvida em linguagem de programação JAVA Web Servlets, e do *framework* JPPF para processamento de tarefas na grade. Conforme as **Tabelas 1 e 2**, foram utilizados sete computadores em Laboratório de Informática com o Debian 9 GNU/Linux executados em *virtual machines* (VM) com o VirtualBox 5.2 em dois computadores e os demais dedicados. É importante salientar que o fato do ambiente estar em VM, o desempenho não é o mesmo quando utilizamos um SO hospedeiro em hardware servidor. Este ambiente foi previamente concebido como proposta para prova de conceito e validação do MDD do PEUP (Azanha Neto *et al.*, 2018).

Tabela 1: Componentes do PEUP no laboratório de informática.

Servidores	Função Do Ambiente	Qtde. Recursos
GlassFish Server	Servidor de Aplicação	01
JPPF Driver Server	Nó Mestre (grade)	01
JPPF Node Server	Nó Escravo (grade)	03
JPPF Monitoring Admin	Admin (grade)	01
MySQL Server	Servidor de BD	01

Apesar da grade computacional do MDD do PEUP distribuir as requisições

dos usuários entre os quatro nós disponíveis da grade, o BD de cada OV é centralizado e, conseqüentemente, acarretará num gargalo no BD, produzindo aumento no tempo de resposta (TR) do ambiente ao usuário final. A proposta de um modelo de *cluster* do BD para o MDD do PEUP visa melhorar o balanceamento de carga de requisições SQL e melhorar o TR por OV. Na **Figura 1**, é possível visualizar a atual arquitetura de comunicação entre os servidores do PEUP.

Como o MySQL é o sistema gerenciador de BD (SGBD) adotado para o MDD do PEUP, o *cluster* de BD foi criado com o MySQL Cluster 5.x e os respectivos servidores que compõem seu ambiente. Assim, a solução conta com o MySQL Cluster Management (MGM), o MySQL Network DataBase (NDB) Server e o MySQL Data Node Server (DN). O MGM faz o gerenciamento dos nós do *cluster*, o NDB é o responsável pelo armazenamento das tabelas do BD do PEUP e o DN executa as transações SQL.

A análise de disponibilidade do DN para o recebimento das consultas configura-se em um eficiente método de balanceamento de carga. Além disso, a indisponibilidade de um DN, causada por questões de manutenção preventiva e/ou corretiva ou por alguma falha, faz com que o NDB dispense as consultas ao DN mais disponível. Esta ação permite maior tolerância a falhas do ambiente e confiabilidade ao PEUP. Além disso, com os DNs disponíveis, os testes comprovam maior desempenho.

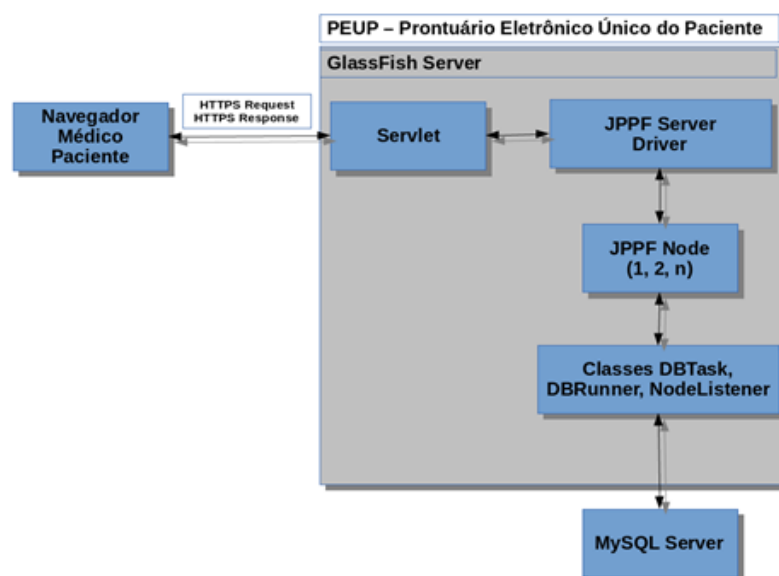


Figura 1: Arquitetura atual de comunicação dos servidores do PEUP. Os n JPPF Nodes comunicam-se com apenas um servidor de BD por OV. A métrica utilizada para avaliar o desempenho do *cluster* e do BD centralizado

foi o tempo de resposta (TR). Em ambos os casos, a validação da proposta atual versus a nova foi realizada em apenas uma OV. A medição do TR, considerando a simulação das requisições do usuário, foi feita com o auxílio da ferramenta JMeter. Esta ferramenta permite configurar as requisições como se estivessem saindo da interface e navegador do usuário e, em paralelo a isso, realiza a medição do TR em milissegundos (ms).

A estratégia de simulação do ambiente visa apresentar os resultados para um comparativo de TR de algumas requisições enviadas ao BD centralizado por OV e das mesmas requisições enviadas ao BD em *cluster* por OV. Os testes no ambiente experimental buscam avaliar o TR da GC em relação ao BD centralizado e o BD em *cluster*. Além disso, foi avaliado o Coeficiente de Variação (CV), que mede a variabilidade de um grupo, ou seja, mede o quanto um grupo se afasta da média, e quanto menor a variabilidade, menor é a dispersão dos dados. Muitos autores indicam diferentes métodos para a sua classificação. Além disso, o CV é intrínseco a cada processo. Nessa análise, foi usada a classificação Estatística Experimental Aplicada (FERREIRA, 1991), que classifica com respeito à precisão do processo, conforme demonstra a **Figura 2**.

Faixa	precisão	cv
entre 10% e 15%	ótima	baixo
entre 15% e 20%	boa	médio
entre 20% e 30%	regular	alto
maior do que 30%	muito ruim (ou péssima)	muito alto

Figura 2: Quadro com as faixas de Coeficiente de Variação.

O próximo passo da análise foi verificar a existência de *outliers*. *Outlier* é descrito em estatística como um ponto que está muito distante das demais observações em uma série estatística, e que é denominado comumente como o "ponto fora da curva". *Outliers* podem ocorrer em qualquer distribuição e são, frequentemente, indicativos de uma distribuição com muita variação entre os seus componentes ou algum erro de mensuração, podendo ser identificados por meio dos valores máximos ou mínimos - já que são observações de desvios extremos.

Entretanto, isso não é uma regra, pois o máximo e o mínimo são *outliers*. Os *outliers* podem ser sinalizados quando se encontram fora de um intervalo da média para mais ou para menos. Esse intervalo para mais ou para menos é definido pelo desvio padrão. A mediana é uma função de estatística robusta, porque consegue desconsiderar os pontos fora da curva que enviesam a amostra, enquanto que a média não consegue (PARANHOS, 2014).

As requisições utilizadas na simulação para avaliar os resultados de tempo de resposta são as funcionalidades: “Listar Pacientes” e “Listar Usuários”, previamente desenvolvidas para validar o MDD do PEUP em outras publicações e reusadas para a validação dessa proposta.

Por meio do projeto, tem-se por objetivos futuros amenizar os gargalos para diminuir o tempo de resposta das requisições, acreditamos que a arquitetura de grade utilizada nesta proposta deverá também facilitar na implementação, pois é flexível em relação à sua integração com várias outras linguagens de programação, e infraestrutura de redes, não sendo necessário mudar SO ou linguagens de programas já utilizados.

Resultados e discussões

Os testes de prova de conceito realizados aportaram para a proposta de incluir a arquitetura em *cluster* para o PEUP e propor um novo ambiente destinando-se à melhoria do desempenho do atual modelo. Assim, tem-se como premissa que a simulação realizada contou com duas funcionalidades previamente desenvolvidas para testar o MDD do PEUP.

A **Tabela 2** apresenta os servidores do MySQL Server que compõem o ambiente de *cluster* para o PEUP.

Tabela 2: A configuração do MySQL Cluster e seus componentes propostos por OV.

Servidores	Função	Qtde Recursos
MySQL Cluster Manager Server	Gerenciador do MySQL Cluster Server	01
MySQL NDB Server	Daemon do MySQL Cluster Server	01
MySQL Data Node Server	Nó de dados do MySQL Cluster Server	02

Conforme discutido anteriormente, o modelo atual do PEUP conta com

uma arquitetura em grade com um BD centralizado por OV. Com os resultados aqui apresentados, propõe-se substituir o BD centralizado por um BD em *cluster*. Assim, as tabelas e gráficos, que são apresentados a seguir, foram gerados com a ativação do BD centralizado em alguns testes e a ativação do *cluster* em outros testes.

Após a realização da primeira bateria de testes, considerando-se duas tarefas como cenário de testes – “Listar Usuários” e “Listar Pacientes” –, esse cenário foi testado em mais duas novas baterias de testes, para maior aferição dos testes e melhor qualidade de análise. Em uma segunda bateria de testes, foram realizadas 25 requisições para cada cenário de testes.

A **Tabela 3** apresenta os valores do TR em ms na execução da funcionalidade “Listar Usuários” e “Listar Paciente” no BD centralizado e no BD em *cluster*. A simulação feita pelo JMeter permitiu realizar a simulação de 25 requisições de usuários (testes) em 60 segundos. Esse processo se repetiu mais quatro vezes totalizando 100 requisições em quatro minutos.

A amostra da **Tabela 3** apresenta *outliers* que representam que os dados estão fora da curva da normalidade. Os tempos máximos e mínimos de resposta no cenário BD centralizado para “Listar Paciente” e “Listar Usuário” apresentaram grande dispersão como mostra os CVs de cada um. Assim, conclui-se que essa amostra não tem um tamanho favorável para análise. Por isso, o próximo passo foi aumentar a quantidade de requisições para 100.

Tabela 3: Dados estatísticos do teste com 25 requisições com o TR em (ms).

TR (ms)	Média	Mediana	Range	Desvio Padrão	CV %
Usuário BD Central	1.799	1.713	3.271	527,0	29,3
Usuário <i>Cluster</i>	1.568	1.574	681	308,0	19,7
Paciente BD Central	1.877	1.815	5.535	657,1	35,0
Paciente <i>Cluster</i>	1.573	1.588	1.181	291,3	19,0

Apesar disso, como pode ser visto na **Figura 3**, houve uma redução no TR com o uso do BD em *cluster* comparado ao centralizado, conforme será discutido a seguir. Este gráfico não mostra a variabilidade dos dados que será mostrada noutro gráfico de distribuição normal.

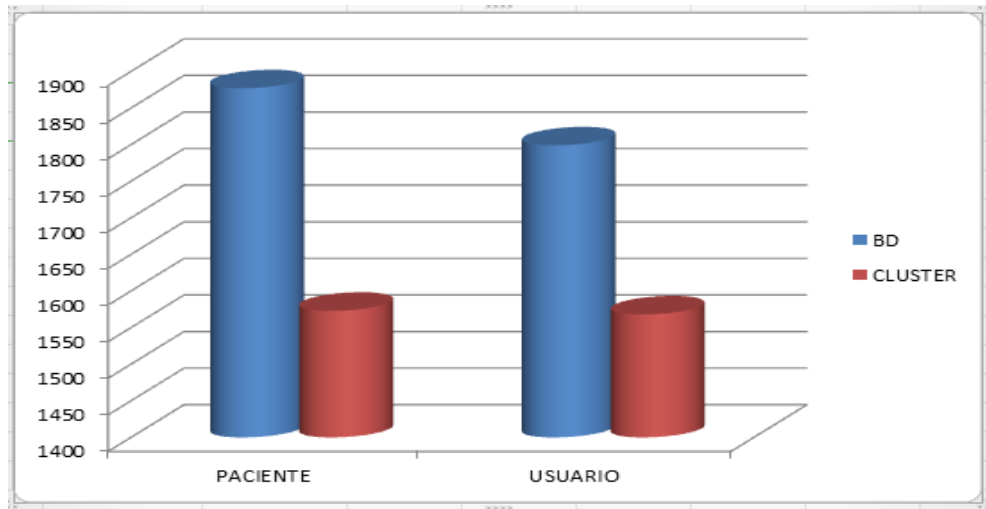


Figura 3: Gráfico com teste com 25 requisições com o TR em milissegundos no eixo Y. Paciente refere-se à funcionalidade Listar Pacientes e Usuário para Listar Usuários no eixo X.

Ao concluir a execução de 25 requisições em cada cenário (**Tabela 3**), totalizando 100 requisições em quatro minutos, foi possível fazer a análise: O cenário um (Listar Paciente – BD centralizado) versus cenário três (Listar Paciente – BD *cluster*), comparando-se os registros dos testes efetuados, o ganho não foi muito alto. Porém, houve um ganho de 16,19% no TR em relação ao BD centralizado. O cenário dois (Listar Usuários – BD centralizado) versus cenário quatro (Listar Usuários – BD *cluster*) apresentou também um ganho de aproximadamente 13% em relação ao BD centralizado. Apesar dos testes não apresentarem aumento significativo, com volume de acesso, esse TR em *cluster* sobressai-se em relação ao BD centralizado.

Em busca de melhores resultados, a terceira bateria de testes foi realizada, dessa vez com 100 requisições em 60 segundos para cada cenário de testes. Conforme apresentado na **Tabela 4**, a menor dispersão foi a do requisito “Listar Paciente – BD centralizado” e a maior dispersão apresentada foi a do requisito “Listar Paciente – BD *cluster*”. Isso não significa que o modelo centralizado seja a melhor opção, pois o TR nesse modelo foi bem maior.

Tabela 4: Resultados de TR da Consulta Usuários e Pacientes considerando o BD centralizado versus o BD em *cluster*.

TR (ms)	Média	Mediana	Desvio Padrão	CV %
Usuário BD Centralizado	13.123	13.148	1.772	13,51
Usuário BD <i>Cluster</i>	3.488	3.348	655	18,80
Paciente BD Centralizado	11.083	11.370	1.209	10,91
Paciente BD <i>Cluster</i>	7.156	7.371	1.360	19,01

Em todos os experimentos, os resultados demonstram um bom comportamento do ambiente e uma boa precisão dos dados, pois os CVs não ultrapassam os 20% sugeridos pelo autor (FERREIRA, 1991).

Nesta análise não foi encontrado *outliers* e, com isso, foi calculado o intervalo de confiança de 95% para cada média. A **Figura 4** de probabilidade de massa mostra que os dados não possuem muita dispersão. Na parte central de cada gráfico concentra-se o tempo médio de resposta que ocorreu com maior frequência durante o teste para o cenário “Listar Paciente”. A parte central da curva mostra a concentração de resultados que ocorreram com maior frequência próxima a da média em um teste, levando em consideração o desvio padrão da amostra. O modelo *cluster* apresentou um ganho de TR significativo em relação ao modelo centralizado.

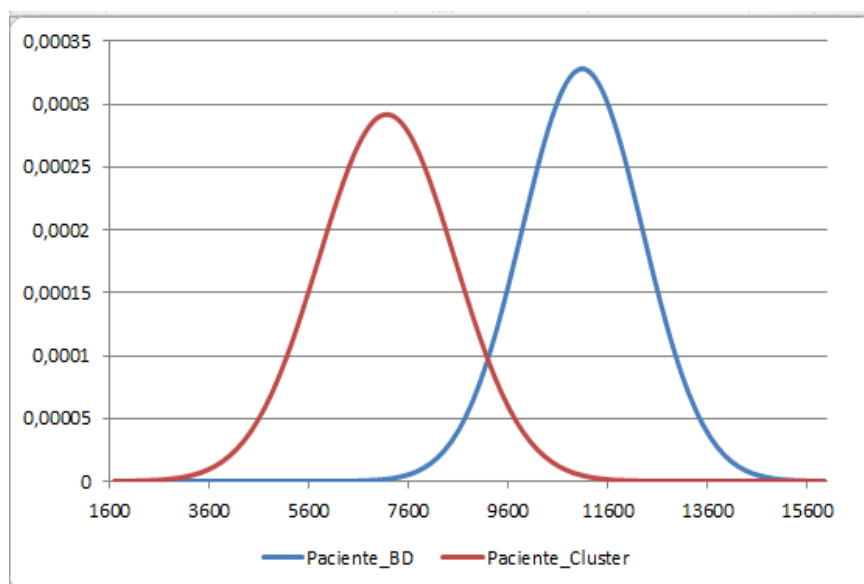


Figura 4: Gráfico de distribuição normal para a tarefa “Listar Paciente”.

Não há dispersão alta entre os dois modelos, conforme valores mencionados na **Figura 5**, e possibilita melhor visualização do desempenho do TR para o cenário “Listar Usuário”. Há uma amplitude muito grande entre os picos de TR nos dois modelos. Sem dúvida, o melhor modelo para este cenário foi o *cluster*, por apresentar pouca dispersão de dados. A abertura da curva ilustra a dispersão dos dados, sendo que, quanto mais aberta for a curva, mais dispersos estão os dados.

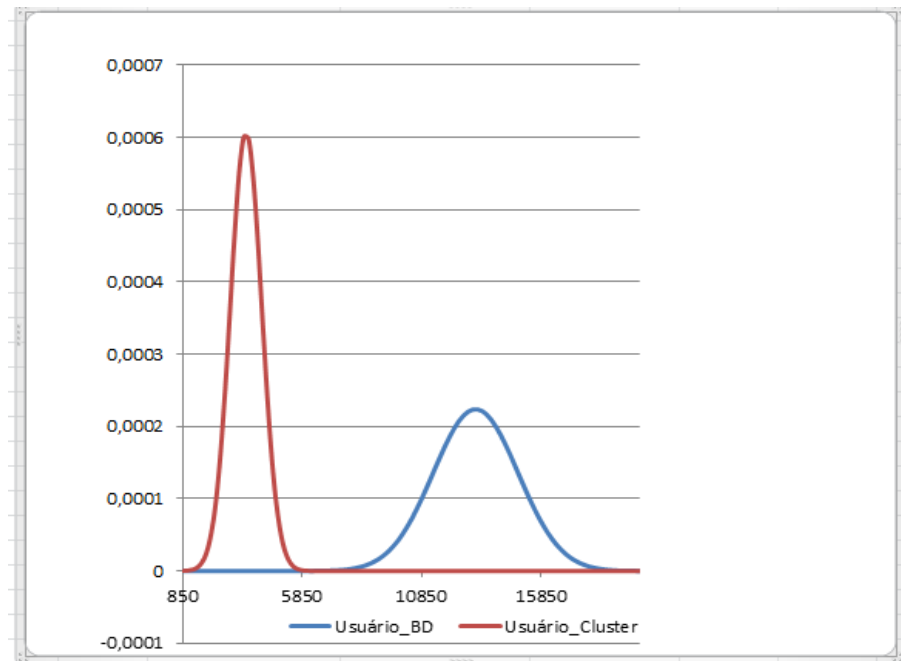


Figura 5: Gráfico de distribuição normal para a tarefa “Listar Usuário”.

Os resultados obtidos para um total de 400 testes, sendo 100 requisições (testes) para cada tarefa de 60 segundos, foram analisados dentro do conceito exposto, conforme a **Tabela 5** e a **Figura 6**, com intervalo de confiança de 95% para a média.

Tabela 5: Resultados de TR de Listar Pacientes e Usuários considerando o BD centralizado versus o BD em *cluster* com 100 requisições (testes) em 60 segundos.

TR (ms)	Valor Mínimo	Valor Máximo
Usuário BD Centralizado	12.775,69	13.470,31
Usuário BD <i>Cluster</i>	3.359,62	3.616,38
Paciente BD Centralizado	10.846,04	11.319,96
Paciente BD <i>Cluster</i>	6.889,44	7.422,56

Após calcular os intervalos de confiança da média, verificou-se que todos os valores estão dentro da média, tornando os testes com resultados confiáveis devido à boa representatividade da média em relação aos desvios padrão.

A avaliação do TR da grade computacional na execução das funcionalidades “Listar Paciente – BD centralizado” e “Listar Paciente – BD *cluster*” mostrou que não houve um ganho significativo no TR. O ganho do BD em *cluster* foi de aproximadamente 5% em relação ao BD centralizado.

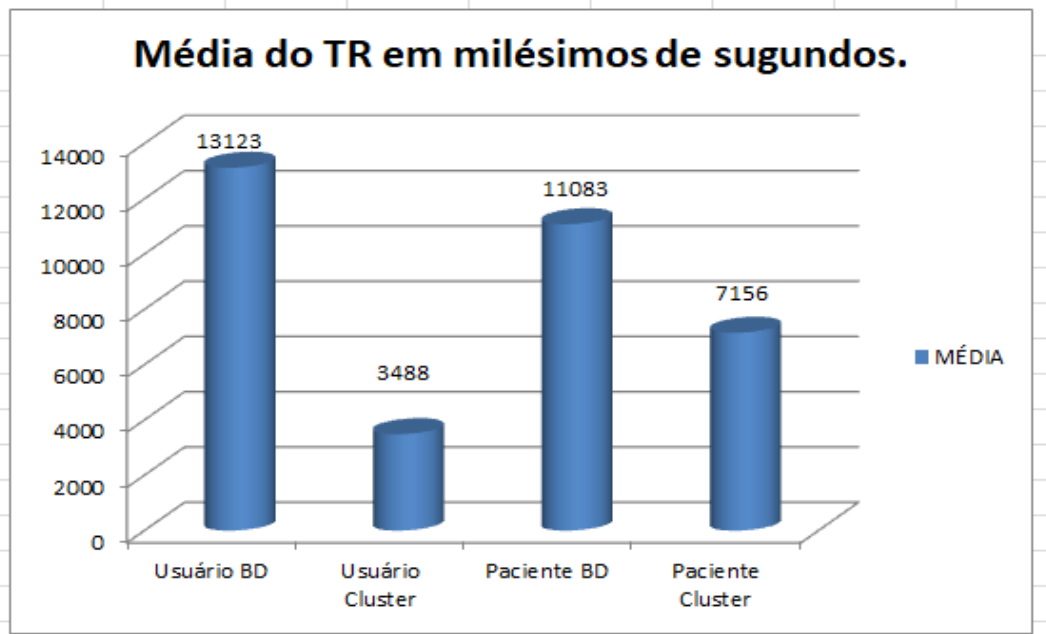


Figura 6: Gráfico da média (em milissegundos no eixo Y) do TR das tarefas Listar Usuário e Listar Paciente (no eixo X) entre BD centralizado e BD em *cluster*.

A mesma avaliação para a funcionalidade “Listar Usuário - BD centralizado” e “Listar Usuário – BD *cluster*”, apesar de parecer que houve ganho satisfatório, na realidade melhorou em 8% o desempenho do TR.

A **Tabela 6** e a **Figura 7** apresentam os resultados na visão da GC, e que não foram satisfatórios, pois temos uma grande dispersão dos dados também. Na GC, executando a funcionalidade “Listar Paciente – BD centralizado”, o range dos tempos máximos e mínimos do TR correspondem a 4.532 ms, enquanto a média desse mesmo requisito é de 2.030 ms.

Tabela 6: Resultados de Média de Listar Pacientes e Usuários considerando o BD centralizado versus em *cluster* com 100 testes em 60 segundos.

Média (ms)	Listar Paciente	Listar Usuário
BD Centralizado	2.030	1.949
BD <i>Cluster</i>	1.928	1.791

Ao executar 100 requisições para os dois tipos, Centralizado e *Cluster*, houve um ganho considerável de 5% no tempo de resposta dado pelo *Cluster*/Paciente e um ganho um pouco maior de 8% para *Cluster*/Usuário.

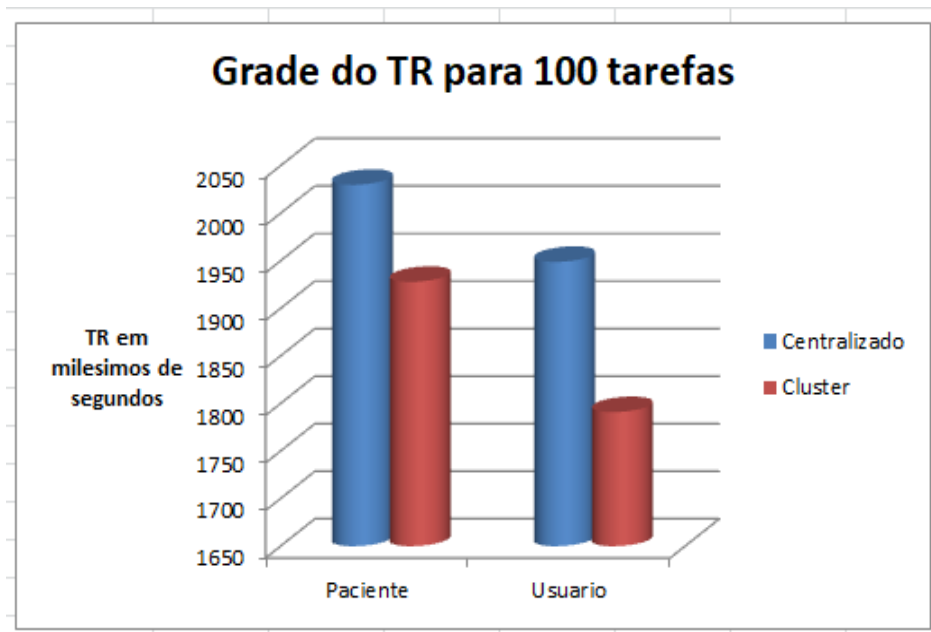


Figura 7: Gráfico com dados do teste com 100 requisições com o TR em milissegundos da GC. Paciente refere-se à funcionalidade Listar Pacientes e Usuário para Listar Usuários.

Os resultados foram coletados a partir da simulação de usuários no JMeter, utilizando-se dos ambientes com o BD centralizado e em *cluster* nos testes com as funcionalidades “Listar Usuários” e “Listar Pacientes”.

Na **Tabela 5**, os quadros comparativos com os TRs para Listar Pacientes – BD centralizado e *cluster*, após 100 testes para cada tarefa, mostraram que houve um ganho significativo de 35% no TR para Listar Pacientes. Quando comparados com os resultados do Banco de Dados. A mesma comparação com a funcionalidade “Listar Usuários” mostrou um ganho muito significativo de 73%, conforme demonstrado na **Figura 4**.

Com base nas análises e na **Figura 3**, pode-se afirmar que a utilização do BD em *cluster* aumenta o desempenho do TR dos dados. Ao aumentar a quantidade de testes, os resultados mostraram pouca dispersão com 95% de confiança.

Estes valores são significativos, pois reduzem a fila de transações SQL, quando comparados ao modelo de BD centralizado. Além disso, na ocorrência de falha de alguns dos nós, sejam eles do JPPF Server ou do MySQL Server, as requisições do usuário seriam atendidas, mesmo que ocorresse um aumento no TR nestes casos.

Conclusão

Após os testes realizados entre o acesso aos dados do PEUP centralizado e em *cluster*, nota-se que a proposta de uma arquitetura em *cluster* para BD é uma alternativa importante para o modelo de BD do PEUP mesmo operando em GC. Na arquitetura proposta nos objetivos, a GC executa paralelamente as requisições do servidor de aplicação e o *cluster*. Da mesma maneira, irá potencializar o desempenho do BD. Dessa forma, os resultados mostraram que a arquitetura de BD em *cluster* é eficiente em lidar com problemas de gargalo que o MDD do PEUP poderá sofrer com aumento do volume de acessos de usuários e a concorrência, principalmente quando operado em conexão através da internet ou em redes privadas de longo alcance.

É importante enfatizar que os *clusters* de computadores possuem inúmeras vantagens. Entretanto, como todo sistema computacional, também possui desvantagens. Portanto é de responsabilidade do especialista analisar cautelosamente as opções e escolher a melhor tecnologia para resolver sua tarefa de forma eficaz. Também é importante ressaltar que os *clusters* de computadores não são bons para resolver problemas que exijam constante troca de informações, pois o tempo de resposta pode variar, dependendo das condições da rede. Nesse ponto, a largura de banda e a qualidade da rede são cruciais para evitar gargalos no *cluster*. A disponibilidade dos serviços, tolerância a falhas e a escalabilidade também são vantagens presentes em *clusters* de computadores, uma vez que sistemas em *clusters* são formados por micros subsistemas independentes.

Portanto, no presente trabalho aproveita-se para mostrar que realmente é necessário investir em sistemas para a área da saúde e temos tecnologias que podem contribuir para o avanço e para a melhoria do acompanhamento do paciente. Nesse sentido, seria importante uma extensão desta pesquisa para um futuro trabalho com enfoque na avaliação do TR em caso de falhas em um ou mais recursos do PEUP operando em rede de longa distância e/ou em nuvens federadas.

Referências

AOKI, Eric Komiyama; DE CARVALHO, Alan Henrique Pardo. Práticas de segurança para o desenvolvimento de sistemas Web. **FaSci-Tech**, v.1, n.5, 2016.

AZANHA NETO, José da Silva; CANNE, Débora Virgília; YBARRA, Luis Antonio Coppa; GARAY, Jorge Rodolfo Beingolea; KOFUJI, Sergio Takeo. Proposta de Modelo de Banco de Dados Distribuído para o Prontuário Eletrônico Único do Paciente. **South American Development Society Journal**, v.4, n.11, DOI: 10.24325/ISSN: 2446-5763, p.266-280, 2018.

CHOWDHARY, Sunil Kumar; YADAV, Ajit; GARG, Naveen. Cloud Computing: Future Prospect for E-health. In: **Electronics Computer Technology (ICECT), 2011 3rd International Conference on**. IEEE, p.297-299, 2011.

COOREVITS, Pascal *et al.* Electronic Health Records: New Opportunities for Clinical Research. **Journal of Internal Medicine**, v.274, n.6, p.547-560, 2013.

DA SILVA, Fabricio Alves Barbosa. Big data e nuvens computacionais: aplicações em saúde pública e genômica. **Journal of health Informatics**, v.8, n.2, 2016.

FERREIRA, Paulo Vanderlei. Estatística experimental aplicada à agronomia. **EDUFAL**, Maceió, 437p., 1991. Disponível em http://www.scielo.br/scielo.php?script=sci_nlinks&ref=000065&pid=S0034-737X201100020000600006&lng=en. Acesso: 2 nov. 2017.

KING, Jennifer; PATEL, Vaishali; JAMOOM, Eric; FURUKAWA, Michael F. Clinical Benefits of Electronic Health Record Use: National Findings. **Health Services Research**, v. 49, n.1, pt.2, p. 92-404, 2014.

PARANHOS, Ranulfo *et al.* Desvendando os mistérios do coeficiente de correlação de Pearson: o Retorno. **Leviathan**. São Paulo, n.8, p.66-95, 2014.

SHAMS, Ramtin; SADEGHI, Parastoo; KENNEDY, Rodney; HARTLEY, Richard. A Survey of Medical Image Registration on Multicore and the GPU. **IEEE Signal Processing Magazine**, v.27, n.2, p.50-60, 2010.

SINHA, Subrata; HAZARIKA, Abinash; BORA, Bishwajit; HAZARIKA, Gopal. A Comprehensive Study on Grid Performance Enhancement and Server Fault Penalty Minimization on JPPF Grid for E. Coli Genome Sequence Alignment Problem. **Journal of Science**, v.1, n.3, p. 28-33, 2016.

SOUSA, Flavio; MOREIRA, Leonardo O.; DE MACÊDO, José Antonio Fernandes; MACHADO, Javam. **Gerenciamento de dados em nuvem: conceitos, sistemas e desafios**. Universidade Federal do Ceará, 2011.

WINCKLER, Gabriel Araujo von. **Proposta de arquitetura para federações de nuvens computacionais acadêmicas**. 2014. Tese de Doutorado. Universidade de São Paulo.

XIONG, Jing; WANG, Jianliang; XU, Jianliang. Research of Distributed Parallel Information Retrieval Based on JJPPF. In: **Information Science and Management Engineering (ISME), 2010 International Conference of**. IEEE, p.109-111, 2010.