Uma abordagem híbrida para armazenamento de dados de contexto no EXEHDA

Diórgenes Yuri Leal da Rosa[1], **Ivan José Rambo**[1], **Roger da Silva Machado**[1], **Ricardo Borges Almeida**[1], **Henrique de Vasconcellos Rippel**[1], **Adenauer Corrêa Yamin**[1], **Ana Marilza Pernas**[1]

[1]Universidade Federal de Pelotas (UFPEL)
Pelotas – RS – Brasil

{diorgenes,ijrambo,rdsmachado,rbalmeida,adenauer,marilza}@inf.ufpel.edu.br

hvrippel@gmail.com

Abstract. *This article presents a hybrid approach to storage contextual information present in EXEHDA middleware. The proposal consists of a layer of abstraction that coordinates access to context data and a hybrid repository of contextual data that uses various models of databases. The contribution was evaluated in order to demonstrate the benefits of its use, where it is emphasized that the results showed a performance gain considering the insertion time and also an improvement in disk storage space used.*

Resumo. *Este artigo apresenta uma abordagem híbrida para o armazenamento de informações contextuais presentes no middleware EXEHDA. A proposta consiste em uma camada de abstração que coordena o acesso aos dados de contexto e em um repositório híbrido de dados contextuais que conta com distintos modelos de bancos de dados. A contribuição foi avaliada de forma a demonstrar os benefícios de sua utilização, onde destaca-se que os resultados obtidos apresentaram um ganho de desempenho considerando o tempo de inserção e também uma melhora no espaço de armazenamento em disco utilizado.*

1. Introdução

Os avanços de diversas tecnologias que permeiam redes de computadores, especialmente aqueles obtidos pelas pesquisas envolvendo Computação Ubíqua (ubicomp), propiciam serviços, aplicações e informações aos usuários a qualquer hora e em qualquer lugar. Estes recursos distribuídos, por sua vez, acabam gerando volumes cada vez maiores de dados, de diferentes tipos e formatos, que precisam ser avaliados e tratados habilmente para garantir a natureza volátil, espontânea, heterogênea e distribuída das comunicações que são peculiares à Ubicomp [Langheinrich 2010].

Dessa forma, ao ambicionar as características computacionais da ubiquidade, toda a aplicação precisa estar preparada para atuar com: (a) um volume massivo de dados; (b) diferentes formatos de eventos e (c) eficácia no que diz respeito à velocidade no tratamento das informações contextuais. Para atender estas demandas, os conceitos de *Big Data* se mostram oportunos devido ao seu foco na utilização de funcionalidades analíticas capazes de lidar com variedades de formatos, velocidade e a volatilidade dos dados [Y Demchenko 2014].

Os bancos de dados não-relacionais vêm assumindo um papel de destaque no âmbito de *Big Data* justamente pelo seu desempenho no tratamento de grandes conjuntos de dados de formatos variados. Percebe-se assim o alinhamento entre os requisitos da Ubicomp e as potencialidades dos bancos de dados não-relacionais. [Sadalage and Fowler 2013]

O presente trabalho consiste em uma contribuição para com o *middleware* EXEHDA (*Execution Environment for Highly Distributed Applications*). Este middleware, proposto em [Yamin 2004] [Lopes et al. 2014], tem como objetivo definir a arquitetura para um ambiente de execução destinado às aplicações da UbiComp.

A proposta traz a concepção de uma abordagem híbrida de armazenamento para dados contextuais a qual foi incorporada no Servidor de Contexto do EXEHDA objetivando otimização no processo de interação com o repositório de dados. A contribuição consiste na concepção de um Repositório Híbrido de Dados Contextuais (RHDC) e de um Gerenciador de Dados Contextuais (GDC) atribuindo ao EXEHDA a possibilidade de um cenário de armazenamento misto.

Este artigo está organizado da seguinte forma: na Seção 2 são descritos os trabalhos relacionados; a Seção 3 introduz os conceitos referentes à *Big Data*, mais especificamente descreve as características dos bancos não-relacionais, juntamente com o *middleware* EXEHDA; na Seção 4 é discutida a concepção da abordagem proposta; a Seção 5 apresenta a avaliação do trabalho desenvolvido; e na Seção 6, são apresentadas as considerações finais.

2. Trabalhos Relacionados

[Carvalho 2014] visa promover a coexistência dos bancos de dados relacional e não-relacional oferecendo uma solução com base em uma abordagem híbrida. O trabalho destaca os desafios e tendências para o desenvolvimento de soluções de armazenamento híbridas.

Em [Marwa 2014] são realizados testes para avaliar o desempenho de três tecnologias de armazenamento de dados para detecção de APT (*Advanced Persistent Threats*): PostgreSQL; MongoDB; e Elasticsearch. A conclusão obtida foi que o MongoDB mostrou melhor desempenho ao monitorar grande volume de dados, e oferece diversos conceitos que podem otimizar ainda mais o processamento.

Em [Filho 2015] é feita uma análise comparativa de desempenho entre a abordagem de armazenamento relacional utilizando o PostgreSQL e não-relacional utilizando o MongoDB. Para analisar o comportamento dos testes e medir o desempenho, o autor utiliza a ferramenta JMeter. Para os testes foram utilizadas as quantidades de 2000, 4000 e 8000 usuários simultâneos realizando uma requisição na base de dados. Nessas três situações, apesar de ocupar maior espaço em disco, o MongoDB obteve um desempenho superior.

Analisando os trabalhos relacionados, pode-se notar que o bancos de dados não-relacionais têm proporcionado bons avanços ao sistemas. Outro fato interessante a ser observado é o trabalho de [Carvalho 2014], o qual propõe a coexistência de dois modelos e destaca que essa é a tendência para o desenvolvimento de novas soluções de armazenamento, de forma a se aproveitar dos pontos positivos de cada modelo.

Os resultados desses trabalhos motivaram a elaboração da proposta aqui estabelecida, contemplando o início dos esforços para atendimento das demandas de *Big Data* no *midleware* EXEHDA.

3. Referencial Teórico

Esta seção introduz a base conceitual associada à concepção da abordagem híbrida de armazenamento, sendo que estes conceitos também foram considerados para avaliação e testes da mesma. Na Sessão 3.1 será abordada a relação existente entre *Big Data* e bancos de dados não-relacionais bem como suas principais bases teóricas. Já na Sessão 3.2 serão tratados aspectos importantes sobre o middleware EXEHDA.

3.1. Big Data: bancos de dados não-relacionais

Big Data é o termo que engloba uma série de conceitos e tendências referentes ao grande volume de dados existentes hoje no contexto computacional, bem como a forma que interagimos com esses dados. Seu uso ocorre nas mais diversas áreas de negócio muitas vezes para proporcionar significado e estratégias por meio de informações coletadas dos mais diversos dispositivos e usuários. É possível afirmar que big data baseia-se, principalmente, nos aspectos Volume, Velocidade, Variedade (3 V's) [Y Demchenko 2014].

Impulsionada pelas demandas de *Big Data*, a utilização de bancos de dados não-relacionais vem ganhando espaço gradativamente. NoSQL, traduzido pela comunidade como 'Not only SQL', refere-se a um grupo cada vez mais familiar de sistemas de bancos de dados não-relacionais, nos quais a base de dados não é constituída de tabelas/esquemas e geralmente não são utilizadas funções em SQL para manipulação de dados. Estas soluções são utilizadas em aplicações que trabalham com enormes quantidades de dados e, também, quando não é possível representar a natureza dos dados no modelo relacional de banco de dados [Moniruzzaman and Hossain 2013]. NoSQL destaca-se também por apresentar capacidade de distribuição da solução de banco de dados como um todo, e trabalhar de forma eficiente em cluster's [Sadalage and Fowler 2013], [Han et al. 2011].

Pode-se identificar que NoSQL possui quatro diferentes categorias em seu ecossistema: chave-valor, documento, famílias de colunas e grafos. Os principais pontos a serem destacados em cada uma das categorias são:

- Chave-valor: apresenta a arquitetura mais simples em NoSQL que é composta apenas por uma chave seguida por um valor. O valor aceita qualquer tipo de dado ou objeto e não possibilita pesquisa através de sua estrutura. A recuperação de dados neste modelo pode ser feita apenas pela chave.
- Documentos: armazenam estruturas de dados independentes na forma de árvores hierárquicas e autodescritivas, constituídas de mapas, coleções e valores escalares. Neste modelo são admitidas pesquisas realizadas a partir da estrutura do documento. Outro aspecto importante, consiste no fato de que os campos vazios são ignorados, o que otimiza o espaço em disco.
- Famílias de colunas: permite que o armazenamento de dados ocorra com chaves mapeadas para valores, e os valores agrupados em múltiplas famílias de colunas, cada uma dessas famílias de colunas funcionando como um mapa de dados. É um modelo interessante quando os grupos de informações são acessados de maneira conjunta.

- Grafos: trata o conjunto de dados como uma densa estrutura de redes, onde os nodos são conectados entre si estabelecendo relações.

Os três fatores apontados pelo trabalho [Couchbase 2012] como os principais problemas de bancos de dados relacionais são a inflexibilidade/rigidez dos esquemas, seguido por baixa escalabilidade e alta latência/baixo desempenho. Estes indicativos demonstram a relevância no estudo de novas alternativas de bancos de dados. Por outro lado, a presença de relacionamentos para demandas específicas traz ao sistema a facilidade do tradicional uso de consultas SQL, desonerando a aplicação da implementação lógica do tratamento de dados. Sendo assim, a adoção de estratégias híbridas tem buscado a união dos benefícios de ambas alternativas.

3.2. EXEHDA

O EXEHDA é um *middleware* adaptativo ao contexto e baseado em serviços que visa criar e gerenciar um ambiente ubíquo. Sua arquitetura é distribuída e oferece suporte à aquisição, processamento e armazenamento de informações contextuais [Lopes et al. 2014].

Os recursos da infraestrutura física que formam o ambiente ubíquo são mapeados para três abstrações básicas [Yamin 2004]:

- EXEHDAcels: indica a área de atuação de uma EXEHDAbase, sendo composta por esta e por EXEHDAnodos;
- EXEHDAbase: é o ponto de convergência para os EXEHDAnodos, sendo responsável por todos os serviços básicos do ambiente ubíquo;
- EXEHDAnodo: são os dispositivos de processamento disponíveis no ambiente ubíquo, sendo responsáveis pela execução das aplicações. Um subcaso deste tipo de recurso é o EXEHDAnodo móvel. São os nodos do sistema com elevada portabilidade, tipicamente dotados de interface de rede para operação sem fio.

Dentro de cada célula podem existir inúmeros SB's (Servidores de Borda) que são responsáveis pela comunicação com o ambiente por meio de sensores e atuadores. Cada célula possui um EXEHDAbase no qual executa o SC (Servidor de Contexto), sendo este servidor responsável por armazenar as informações coletadas no RIC (Repositório de Informações Contextuais), bem como permitir a manipulação (processamento, visualização, etc.) destas informações.

4. Modelo Proposto

A concepção da proposta híbrida de armazenamento teve por motivação as demandas de interação com os dados existentes no *middleware* EXEHDA, no intuito de contemplar de maneira mais efetiva os 3V's de *Big Data*. Ao unir os benefícios de modelos distintos de armazenamento o *middleware* consegue um cenário mais adaptado aos dados em questão.

A proposta desenvolvida nesse trabalho partiu da implementação do modelo não-relacional de documentos em conjunto com o banco relacional anteriormente empregado pelo repositório de informações. As adaptações necessárias foram realizadas de forma a propiciar o funcionamento das duas abordagens disponibilizadas pelo repositório de dados contextuais em paralelo. A contribuição foi estabelecida no SC do EXEHDA.

Na abordagem proposta fica a cargo da aplicação determinar onde prefere armazenar seus dados contextuais, sendo indicada a utilização do banco relacional para dados que possuem relação entre eles aproveitando-se das características do modelo. E recomenda-se o armazenamento dos dados contextuais no modelo não-relacional, quando é necessário o tratamento de grande volume de dados e/ou ainda quando os dados armazenados possuem variedade de formatos, o que resultaria em colunas vazias no modelo relacional.

A Figura 1 apresenta a proposta híbrida de armazenamento oferecido, onde o GDC é responsável por disponibilizar métodos de inserção e consultas para acesso aos dados presentes nos dois modelos de armazenamento, representando uma abstração para acesso ao RHDC. Destaca-se que as aplicações não precisam se envolver com a interoperação entre as formas de armazenamento que estão sendo utilizadas no processamento dos vários contextos de seu interesse.

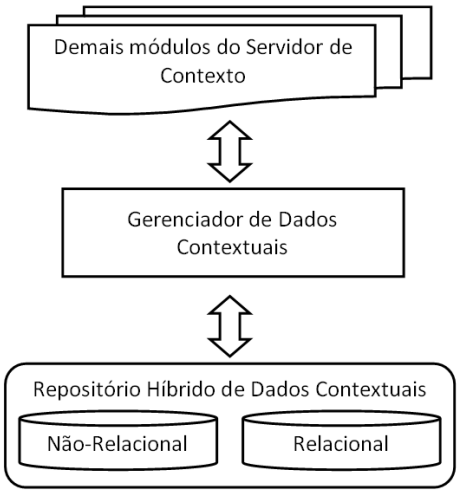

Figura 1. Gerenciador de Dados de Contexto

5. Estudo de Caso

As avaliações iniciais da proposta híbrida apresentada nesta sessão partiram dos trabalhos [Machado 2013] e [Almeida 2013] que foram baseados no EXEHDA. [Machado 2013] foi estruturado de forma a criar um analisador de registros de log. Já o trabalho [Almeida 2013] teve como foco o emprego da consciência de situação em soluções de SIEM (*Security Information and Event Management*), utilizando uma solução para processamento de eventos complexos.

Estes projetos caracterizam-se por possuírem em comum, a necessidade de um bom desempenho para o tratamento de um elevado volume de eventos que possuem diversidade nos formatos. Estes eventos são provenientes dos logs gerados pelo sistema operacional e por aplicações que operam em ambientes ubíquos.

Para implementação da abordagem foram utilizados o banco de dados PostgreSQL para o modelo relacional, e o MongoDB para o modelo não-relacional. Este último

consiste de um modelo não-relacional orientado a documentos, o qual admite pesquisas por intermédio de sua estrutura básica, sendo recomendado para o tratamento de logs [Sadalage and Fowler 2013]. Desta forma, os eventos registrados em logs passaram a ser armazenados no MongoDB, enquanto as configurações dos dispositivos monitorados e situações identificadas foram mantidas no PostgreSQL. A consistência entre as situações e os eventos que o representam fica a cargo do GDC que faz uso de funções python.

Como parte dos esforços de avaliação da abordagem híbrida proposta foram desenvolvidos dois cenários em [Rambo 2015] com o intuito de verificar o ganho na utilização de um modelo não-relacional comparado com o modelo relacional, sendo estes baseados respectivamente nos itens de coleta denominados access.log e kern.log.

O access.log é um log do servidor apache, que registra todas as solicitações processadas pelo mesmo. Já o kern.log fornece um registro detalhado das mensagens do kernel do Linux, que pode ser útil, por exemplo, para encontrar problemas no sistema operacional e analisar mensagens do *firewall*.

A escolha por estes logs é justificada por estes possuírem características distintas em relação ao formato de seus eventos e ainda por serem utilizados para identificação de situações normalmente encontradas em uma infraestrutura de redes de computadores [Swift 2010], sendo sua análise oportuna em ambientes ubíquos como os gerenciados pelo *middleware* EXEHDA.

Para realização dos testes com a abordagem híbrida desenvolvida, foram simulados localmente os ambientes necessários para a avaliação. A máquina responsável pelas simulações possui um processador Intel Core i5 com 2.27GHz de frequência, 4GB de memória e disco rígido de 500GB.

Para cada cenário foram coletados cinco diferentes quantidades de registros de logs (10000, 20000, 40000, 80000 e 100000) e para cada quantidade foi feita a análise de tempo de inserção e espaço em disco utilizado pelos modelos relacional e não-relacional. Para cada valor representado nas situações descritas nos cenários, a execução foi repetida quatro vezes e foi realizada a média dos valores coletados. Importante destacar que o desvio padrão máximo, considerando as diferentes medições dos testes de inserção foi de 0,02 segundos.

A Tabela 1 apresenta o tempo de inserção em ambas estratégias de armazenemanto com os logs kern e access, onde o tempo é representado no formato de (horas:minutos:segundos).

Tabela 1. Tempos de inserção.

		Número de Eventos				
		1000	2000	4000	8000	10000
access.log	MongoBD	00:00:45	00:01:08	00:02:17	00:04:34	00:05:57
	PostgreSQL	00:01:41	00:03:16	00:06:41	00:13:20	00:16:55
kern.log	MongoBD	00:00:36	00:01:04	00:02:08	00:04:12	00:05:50
	PostgreSQL	00:01:40	00:03:17	00:06:54	00:13:16	00:16:31

Analisando a Tabela 1 fica evidente a superioridade em relação ao processamento realizado pelo modelo não-relacional, comprovando as características de desempenho do modelo. Dentre as características que proporcionam o ganho de desempenho, é possível citar a geração do identificador único (id) para cada documento de uma coleção, onde o id é gerado por um algoritmo que utiliza 12-bytes, fazendo com que os registros sejam inseridos simultaneamente. Já no modelo relacional, é necessário que os registros sejam salvos um após o outro.

A Tabela 2 apresenta os valores de espaço em disco para armazenar os diferentes valores de logs coletados. Pode-se notar que em ambos logs a estratégia não-relacional ocupa uma menor quantidade de espaço em disco, mostrando-se apta a ser utilizada para o tratamento do grande volume de dados gerado pelo monitoramento de logs.

Tabela 2. Espaço ocupado em disco.

		Número de Eventos				
		1000	2000	4000	8000	10000
access.log	MongoBD	3,84MB	7,7MB	15,39MB	30,77MB	38,5MB
	PostgreSQL	7,53MB	15MB	30MB	60MB	75MB
kern.log	MongoBD	1,78MB	3,5MB	6,96MB	14MB	17,2MB
	PostgreSQL	2,13MB	4,31MB	9,42MB	17,24MB	21MB

Analisando a Tabela 2 destaca-se que a diferença entre as estratégias de armazenamento deve-se ao formato dos eventos, onde os do access.log variam consideravelmente, o que gerava colunas em branco no modelo relacional. Em relação ao kern.log observa-se que não foi obtida tanta diferença, mas isso deve-se ao fato dos eventos deste log manterem um padrão e assim não gerarem colunas em branco no modelo relacional.

6. Considerações Finais

Este trabalho apresentou a concepção de uma abordagem híbrida de armazenamento de informações contextuais, a qual foi empregada no *midleware* EXEHDA. A contribuição estabeleceu-se pela concepção do RHDC e pelo GDC. Desta forma, foi possível contribuir para o EXEHDA, fornecendo os benefícios das duas abordagens de banco de dados (relacional e não-relacional), em conjunto com uma camada para abstração na comunicação com os bancos.

A avaliação da proposta já pode quantificar as melhorias resultantes da adesão de um novo modelo de banco de dados por parte do *middleware* EXEHDA, obtendo resultados positivos.

Destaca-se que este é um trabalho inicial nos esforços para atender as demandas de *Big Data* do EXEHDA. Como trabalhos futuros pode-se considerar:

- ampliar a amostragem de testes e a avaliação do impacto da abordagem Híbrida nos diferentes tipos de aplicações que o *middleware* EXEHDA oferece suporte;
- fazer uso de estratégias envolvendo MapReduce.
- adequar os serviços da arquitetura de software do EXEHDA visando as características de *Big Data*;

Referências

Almeida, R. B. (2013). Segurança da informação e gerenciamento de eventos: Uma abordagem explorando consciência de situação. Monografia de graduação em ciência da computação, Universidade Federal de Pelotas, Pelotas, RS, Brasil.

Carvalho, A. G. (2014). Interface nosql integrada a banco relacional para gerenciamento de dados em nuvem privada. Monografia bacharelado em engenharia da computação, Centro Universitário de Brasília Faculdade de Tecnologia e Ciências Sociais Aplicadas.

Couchbase (2012). Acesso em: 7 dez 2014. Couchbase Survey Shows Accelerated Adoption of NoSQL in 2012. Disponível em: <http://www.couchbase.com/press-releases/couchbase-survey-shows-accelerated-adoption-nosql-2012>.

Filho, M. A. P. M. (2015). Sql x nosql: Análise de desempenho do uso do mongodb em relaçãoo ao uso do postgresql.

Han, J., Haihong, E., Guan, L., and Jian, D. (2011). Survey on nosql database. *Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on*, pages 363 – 366.

Langheinrich, M. (2010). *Privacy in Ubiquitous Computing*. J. Krumm, ed., CRC Press.

Lopes, J., Souza, R., Geyer, C., Costa, C., Barbosa, J., Pernas, A., and Yamin, A. (2014). A middleware architecture for dynamic adaptation in ubiquitous computing. *j-jucs*, 20(9):1327–1351.

Machado, R. d. S. (2013). Loga-dm: uma abordagem de análise dinâmica de log com base em mineração de dados. Monografia de graduação em ciência da computação, Universidade Federal de Pelotas, Pelotas, RS, Brasil.

Marwa, A. (2014). Comparison of data base technologies for apt detection. Phd thesis, Royal Military Academy.

Moniruzzaman, A. B. M. and Hossain, S. A. (2013). Nosql database: New era of databases for big data analytics - classification, characteristics and comparison. *CoRR*, abs/1307.0191.

Rambo, I. J. (2015). Ricnr2: Uma proposta não-relacional para tratamento de dados de contexto no exehda. Monografia de graduação em ciência da computação, Universidade Federal de Pelotas, Pelotas, RS, Brasil.

Sadalage, P. J. and Fowler, M. (2013). *NoSQL Essencial, Um Guia Conciso para o Mundo Emergente da Persistência Poliglota*. Novatec.

Swift, D. (2010). Successful siem and log management strategies for audit and compliance. Technical report, SANS Institute - InfoSec Reading Room.

Y Demchenko, C Laat Dee, P. M. (2014). Defining architecture components of the big data ecosystem. *Collaboration Technologies and Systems (CTS), 2014 International Conference on*, pages 104 – 112.

Yamin, A. C. (2004). *Arquitetura para um Ambiente de Grade Computacional direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva*. PhD thesis, Universidade Federal do Rio Grande do Sul.