

UMA PROPOSTA PARA GERENCIAMENTO HÍBRIDO DE DADOS DE CONTEXTO NO EXEHDA

DIÓRGENES YURI LEAL DA ROSA¹; IVAN JOSÉ RAMBO; RICARDO BORGES ALMEIDA¹; ROGER DA SILVA MACHADO¹; ADENAUER CORREA YAMIN¹; ANA MARILZA PERNAS¹

¹Universidade Federal de Pelotas – {diorgenes, ijrambo, rbalmeida, rdsmachado, adenauer, marilza}@inf.ufpel.edu.br

1. INTRODUÇÃO

Ao ambicionar as características computacionais da ubiquidade, é interessante que as aplicações considerem as demandas de Big Data, em especial os chamados 3V's de Big Data que são: volume, velocidade e variedade.

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 (SADALAGE; FOWLER, 2013). Percebe-se assim o alinhamento entre os requisitos da computação Ubíqua (UbiComp) e as potencialidades dos bancos de dados não-relacionais.

(CARVALHO, 2014) relaciona-se a este tema ao objetivar 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.

(FILHO, 2015) também compara os bancos PostgreSQL e MongoDB destacando o melhor desempenho do MongoDB.

Ao testar os gerenciadores PostgreSQL, MongoDB e Elasticsearch para detecção de APT (*Advanced Persistent Threat*) (MARWA, 2014) também atribui melhor desempenho ao MongoDB.

Analisando os trabalhos relacionados, pode-se notar que os bancos de dados não-relacionais tem proporcionado bons avanços aos sistemas. Por outro lado, a presença de relacionamentos para demandas específicas traz ao sistema a facilidade do tradicional uso de consultas de Structured Query Language (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.

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

A contribuição traz uma abordagem híbrida de armazenamento para dados contextuais no EXEHDA que ocorre por intermédio da concepção de um Repositório Híbrido de Dados Contextuais (RHDC) e de um Gerenciador de Dados Contextuais (GDC).

2. METODOLOGIA

A concepção da proposta híbrida de armazenamento teve por motivação as demandas de interação com os dados existentes no *middleware* EXEHDA. Ao unir os benefícios de modelos distintos de armazenamento o *middleware* pode atingir 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. As adaptações necessárias foram realizadas de forma a propiciar o funcionamento em paralelo das duas abordagens disponibilizadas pelo RHDC. 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 consulta 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.

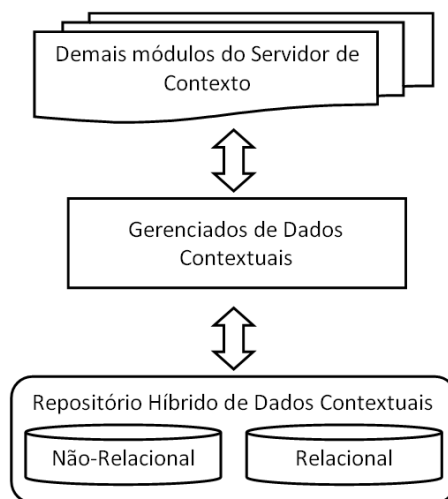


Figura 1 - Gerenciador de Dados de Contexto

3. RESULTADOS E DISCUSSÃO

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, considerando a gestão de armazenamento dos logs access.log e kern.log. 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 implementação da abordagem foram utilizados o sistema gerenciador 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; FOWLER, 2013). A consistência entre as situações e os eventos que o representam fica a cargo do GDC.

Para a 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 foi 0,02.

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

| | | Número de Eventos | | | | |
|------------|------------|-------------------|----------|----------|----------|----------|
| | | 10000 | 20000 | 40000 | 80000 | 100000 |
| access.log | MongoDB | 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 | MongoDB | 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 |

Tabela 1- Tempo de inserção

Analisando a Tabela 1 fica evidente a superioridade em relação ao processamento realizado pelo modelo não-relacional. 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.

| | | Número de Eventos | | | | |
|------------|------------|-------------------|--------|---------|---------|--------|
| | | 10000 | 20000 | 40000 | 80000 | 100000 |
| access.log | MongoDB | 7,53MB | 15MB | 30MB | 60MB | 75MB |
| | PostgreSQL | 3,84MB | 7,7MB | 15,39MB | 30,77MB | 38,5MB |
| kern.log | MongoDB | 2,13MB | 4,31MB | 9,42MB | 17,24MB | 21MB |
| | PostgreSQL | 1,78MB | 3,5MB | 6,96MB | 14MB | 17,2MB |

Tabela 2 - Espaço em disco

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.

4. CONCLUSÕES

Este trabalho apresentou a concepção de uma abordagem híbrida de armazenamento de dados contextuais, a qual foi empregada no 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. Em uma vertente deste trabalho pode-se expandir os testes realizados para cenários que contemplem o trabalho simultâneo dos modelos na gestão de dados contextuais. É previsto ainda o trabalho com o paradigma de programação MapReduce na busca de melhorar o desempenho das consultas no modelo não-relacional.

5. REFERÊNCIAS BIBLIOGRÁFICAS

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

FILHO, M. A. P. M. **Sql x nosql: Análise de desempenho do uso do mongodb em relação ao uso do postgresql**. 2015.

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.; FOWLER, M. **NoSQL Essencial, Um Guia Conciso para o Mundo Emergente da Persistência Poliglota**. 2013. Novatec.

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

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.