



UNIVERSIDADE FEDERAL DO CEARÁ  
CAMPUS QUIXADÁ  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

**FERNANDO HELTON LINHARES SOARES**

**UMA FERRAMENTA PARA O MONITORAMENTO DE SISTEMAS DE  
BANCO DE DADOS EM NUVEM**

**QUIXADÁ  
2011**

**FERNANDO HELTON LINHARES SOARES**

**UMA FERRAMENTA PARA O MONITORAMENTO DE SISTEMAS DE  
BANCO DE DADOS EM NUVEM**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso de Graduação em Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: Computação

Orientador: Prof.<sup>o</sup> Flávio R. C. Sousa

**QUIXADÁ  
2011**

S654f

Soares, Fernando Helton Linhares.

Uma ferramenta para o monitoramento de sistemas de banco de dados em nuvem / Fernando Helton Linhares Soares. – Quixadá, 2011.

40 f.: il.; color.; 31 cm.

Cópia de computador (printout(s)).

Orientador: Prof. MSc. Flávio Rubens de Carvalho Sousa  
Monografia (graduação em Sistemas de Informação) –

Universidade Federal do Ceará, Campus Quixadá, Quixadá, Ceará, 2011.

1. Computação em nuvem 2. Monitoramento 3. Controle de qualidade 4. Qualidade de serviço I. Sousa, Flávio Rubens de Carvalho (orient.) II. Universidade Federal do Ceará – Curso de Bacharelado em Sistemas de Informação III. Título

CDD 004.36

**FERNANDO HELTON LINHARES SOARES**

**UMA FERRAMENTA PARA O MONITORAMENTO DE SISTEMAS DE BANCO DE  
DADOS EM NUVEM**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso de Graduação em Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: Computação

Aprovado em: \_\_\_\_ / \_\_\_\_ / 2011.

BANCA EXAMINADORA

---

Prof.<sup>a</sup> MSc. Flávio R. C. Sousa (Orientador)  
Universidade Federal do Ceará-UFC

---

Prof. MSc. Vitor Almeida dos Santos  
Universidade Federal do Ceará-UFC

---

Prof. MSc. Wladimir Araújo Tavares  
Universidade Federal do Ceará-UFC

Dedico esta monografia a Heldo e Ledy, por estarem sempre ao meu lado me guiando pelos caminhos corretos sem nunca medir esforços para a realização dos meus sonhos. Estas duas pessoas me mostraram que honestidade e respeito são essenciais à vida, e que sempre devemos lutar pelo que queremos. Sou muito feliz e orgulhoso de poder chamá-los de pai e mãe. AMO VOCÊS!

## **AGRADECIMENTOS**

A Deus, meu refúgio e força, onde sempre encontrei soluções para os meus problemas.

Ao meu orientador, Flávio Sousa, pelo acompanhamento bem realizado ao longo desse trabalho, nunca se ausentando ou deixando de responder alguma dúvida.

Ao professor Davi Romero, meu tutor no PET por mais de dois anos, por sempre nos apoiar, nos guiar em nossas atividades, pela paciência e incentivo ao longo desses anos, por está sempre disponível a nos aconselhar.

A todos os professores que sempre se esforçaram para nos passar o máximo de conhecimento que tinham. Sempre presentes e dispostos a ajudar.

Aos meu colegas Adriano, André e Rainara, sempre amigos e companheiros durante toda a luta enfrentada durante o curso.

A minha família, por estar sempre unida e dedicada a ajudar uns aos outros. Em especial a minha tia Luciene, que foi essencial para minha permanência nesta cidade, lembra do nosso cafezinho?

A minha noiva, Donátilla, pela compreensão e incentivo durante todo esse trabalho. Por nunca me deixar sozinho nos momentos de desespero e sempre me ajudar quando mais preciso.

Enfim a todos que direta ou indiretamente fazem parte dessa história. O meu muito obrigado!

"Eu acredito demais na sorte. E tenho constatado que, quanto mais duro eu trabalho, mais sorte eu tenho." (Thomas Jefferson)

# UMA FERRAMENTA PARA O MONITORAMENTO DE SISTEMAS DE BANCO DE DADOS EM NUVEM

## RESUMO

Com o avanço da sociedade humana moderna, serviços básicos e essenciais são quase todos entregues como um serviço de forma completamente transparente e com pagamento baseado no uso. Recentemente, a mesma ideia tem sido aplicada no contexto da informática e uma mudança consistente neste sentido tem sido feita com a disseminação de *Cloud Computing* ou Computação em Nuvem. Sistemas de banco de dados em nuvem estão sendo utilizados e têm o potencial de atrair clientes de diversos setores do mercado, desde pequenas empresas com o objetivo de reduzir o custo total, por meio da utilização de infraestrutura e sistemas de terceiros, até grandes empresas que buscam soluções para gerenciar milhares de máquinas e atender um aumento inesperado de tráfego. Na nuvem, o monitoramento de serviços de dados é uma característica fundamental. Este trabalho teve como objetivo desenvolver o MDataCloud, uma ferramenta para o monitoramento de serviços de bancos de dados em ambientes de computação em nuvem. O MDataCloud é flexível, extensível e disponibilizado como um serviço. Com isso, os recursos monitorados são acessados de forma simples, facilitando o gerenciamento dos serviços de dados. Para validar a ferramenta proposta, um estudo de caso foi realizado e os resultados demonstraram que a ferramenta permite um gerenciamento simples e eficiente de serviços de banco de dados.

Palavras chave: Computação em Nuvem, Monitoramento, Qualidade de Serviço.

# **A TOOL FOR MONITORING CLOUD-BASED DATABANK SYSTEMS**

## **ABSTRACT**

With the advance of modern human society, basic and essential services are almost all delivered as a completely transparent form of service with payment on delivery. Recently the same idea has been applied in a computing context, and a consistent change in this sense has arisen with the dissemination of Cloud Computing. Cloud-based databanks are in use with the potential of attracting customers from several sectors, from small companies whose objective is cost reduction through the use of outsourced infrastructure and systems, to large enterprises seeking solutions to manage thousands of machines and to attend unexpected surges in traffic. Data service monitoring is a fundamental feature on the cloud. MDatacloud is flexible, extensible, and offered as a service. Thus, monitored resources are accessed in a simple fashion making data service management easier. In order to validate the proposed tool, a case study was conducted and the results show that the tool allows simple and efficient databank service management.

Key words: Cloud Computing; Monitoring; Quality of Service.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Serviços que Compõe o MDataCloud .....	21
Figura 2 – Arquitetura do MDataCloud .....	23
Figura 3 – Relacionamento Entre as Entidades .....	24
Figura 4 – Como as Regras são Definidas .....	25
Figura 5 – Exemplo de Configuração .....	26
Figura 6 – Relacionamento Entre os Serviços .....	27
Figura 7 – Diagrama de Classes .....	28
Figura 8 – Diagrama de Sequencia .....	30
Figura 9 – Adicionar Parâmetro/Adicionar Ação .....	31
Figura 10 – Adicionar Parâmetro/Adicionar Ação .....	32
Figura 11 – Adicionar Regra .....	32
Figura 12 – Listar Regras .....	33
Figura 13 – Iniciar Coleta .....	33
Figura 14 – Listar Parâmetros Valores .....	34
Figura 15 – Executar Ação de Forma Manual .....	34
Figura 16 – Ações Executadas .....	35

## SUMÁRIO

1 INTRODUÇÃO .....	11
2 FUNDAMENTAÇÃO TEÓRICA.....	13
2.1 Computação em Nuvem.....	13
2.1.1 Características Essenciais.....	14
2.1.2 Modelos de Serviço .....	14
2.1.3 Modelos de Implantação .....	15
2.2 Arquitetura Orientada a Serviços.....	16
2.3 Qualidade de Serviço e Monitoramento de Serviços de Dados em Nuvem .....	17
2.4 Trabalhos Relacionados .....	19
3 PROCEDIMENTOS METODOLÓGICOS .....	21
4 MDATA CLOUD .....	22
4.1 Arquitetura.....	23
4.2 Cenário de Utilização.....	27
4.3 Implementação.....	28
5 AVALIAÇÃO.....	32
6 CONSIDERAÇÕES FINAIS .....	38
REFERÊNCIAS .....	39

## 1 INTRODUÇÃO

Os avanços tecnológicos têm resultado em uma enorme quantidade de dados armazenados sobre os indivíduos e empresas, bem como em grande mudança em *hardware* e plataformas de computação. Esta mudança promove o desenvolvimento de um rico conjunto de aplicações emergentes na internet e em dispositivos móveis, que estão transformando a nossa forma de trabalhar e viver. Este contexto, aliado a fatores econômicos, está levando ao aumento de infraestruturas e instalações de fornecimento de computação como um serviço, conhecido como *cloud computing* ou computação em nuvem, através da qual empresas e indivíduos podem alugar capacidade de computação e armazenamento, em vez de fazer grandes investimentos de capital necessários para a instalação de equipamentos de computação em larga escala [Buyya et al. 2009]. Os serviços alugados são tipicamente hospedados em centros de dados, utilizando *hardware* compartilhado para processamento e armazenamento.

Embora estes serviços emergentes tenham reduzido o custo de armazenamento de dados e melhorado o acesso, há uma enorme complexidade envolvida na garantia de serviços de dados que podem escalar, quando é necessário garantir operações consistentes e confiáveis sob cargas máximas. Além disso, o ambiente baseado em nuvem tem requisitos técnicos específicos para controlar centros de dados virtualizados, reduzindo os custos e aumentando a confiabilidade por meio da consolidação de sistemas em nuvem. Desafios como consistência, segurança dos dados e autonomia são importantes para a computação em nuvem e acredita-se que futuras aplicações centradas em dados irão alavancar os serviços de dados em nuvem [Sousa et al. 2010].

Sistemas de Banco de Dados em nuvem estão começando a ser utilizados e têm o potencial de atrair clientes de diversos setores do mercado, desde pequenas empresas com o objetivo de reduzir o custo total, por meio da utilização de infraestrutura e sistemas de terceiros, até grandes empresas que buscam soluções para gerenciar milhares de máquinas e atender um aumento inesperado de tráfego [Sousa et al. 2009].

O monitoramento de ambientes de computação em nuvem é fundamental para o gerenciamento adequado destes ambientes [Sousa et al. 2011]. Tratando-se especificamente do monitoramento de serviços de banco de dados, as soluções relacionadas neste trabalho tenta resolver essa questão realizando monitoramento de características específicas do serviço de dados, tais como utilização de CPU e memória. Além disso, estas soluções não foram construídas para ser utilizadas por outros serviços de dados.

Para melhorar o monitoramento dos serviços de dados, este trabalho propõe o MDataCloud, uma ferramenta para monitorar serviços de banco de dados em ambientes de computação em nuvem. O MDataCloud é flexível, extensível e disponibilizado como um serviço. Com isso, os recursos monitorados são acessados de forma simples, facilitando o gerenciamento dos serviços de dados.

Para atingir este objetivo, definimos os seguintes objetivos específicos: estudar as atuais ferramentas para o monitoramento de ambiente neste contexto e analisar as técnicas de monitoramento utilizadas por estas ferramentas; construir um protótipo da ferramenta proposta e realizar um estudo de caso para avaliar a eficiência da ferramenta.

O texto está organizado como se segue. O Capítulo 2 destaca os principais conceitos utilizados nesse trabalho e a arquitetura orientada a serviços. No Capítulo 3 é discutido sobre os procedimentos metodológicos utilizados no trabalho. O Capítulo 4 detalha a ferramenta desenvolvida, sua arquitetura, um cenário de utilização e a implementação. O Capítulo 5 descreve um estudo de caso realizado com a ferramenta proposta. Por fim, o Capítulo 6 contém as conclusões e trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo discorre sobre a fundamentação teórica utilizada neste trabalho. É apresentada uma definição de computação em nuvem, suas características essenciais, seus modelos de serviço e seus modelos de implantação. A arquitetura orientada a serviços também é apresentada, assim como os conceitos envolvidos na qualidade de serviço e no monitoramento de serviços de banco de dados em nuvem. Por fim este capítulo contém uma descrição dos trabalhos relacionados.

### 2.1 *Computação em Nuvem*

A computação em nuvem está se tornando uma das palavras chaves da indústria de TI. A nuvem é uma metáfora para Internet ou infraestrutura de comunicação entre componentes arquiteturais, baseada em uma abstração que oculta a complexidade da infraestrutura. Cada parte desta infraestrutura é provida como um serviço e estes são normalmente alocados em centros de dados, utilizando *hardware* compartilhado para computação e armazenamento [Buyya et al. 2009].

Normalmente, um ambiente de computação em nuvem é composto por um grande número de máquinas físicas (nós físicos) de baixo custo, interligados por meio de uma infraestrutura de rede. Estas máquinas físicas possuem as mesmas configurações de *software*, porém diferem na sua capacidade de *hardware* [Soror et al. 2010].

A computação em nuvem é uma evolução da *Utility Computing* [Brantner et al. 2008], cujo objetivo é fornecer componentes básicos como armazenamento, processamento e largura de banda de uma rede, como uma "mercadoria" através de provedores especializados com um baixo custo por unidade utilizada. Os usuários de serviços baseados em *Utility Computing* não precisam se preocupar com escalabilidade, pois a capacidade de armazenamento fornecida é percebida como infinita por ser de responsabilidade do fornecedor do serviço.

O *National Institute of Standards and Technology* (NIST) define computação em nuvem como sendo: “um modelo que possibilita acesso, de modo conveniente e sob demanda, a um conjunto de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente adquiridos e liberados com mínimo esforço gerencial ou interação com o provedor de serviços” [Mell and Grance

2009]. O NIST descreve que a computação em nuvem é composta por cinco características essenciais, três modelos de serviço e quatro modelos de implantação, detalhados a seguir.

### 2.1.1 Características Essenciais

As características essenciais são vantagens que as soluções de computação em nuvem oferecem. Algumas destas características, em conjunto, definem exclusivamente a computação em nuvem e faz a distinção com outros paradigmas. Por exemplo, a elasticidade rápida de recursos, amplo acesso e medição de serviço são características básicas para compor uma solução de computação em nuvem [Mell and Grance 2009].

- **Self-service sob demanda** – Quando há necessidade de o usuário adquirir mais recursos, *hardware* e *software* dentro de uma nuvem podem ser automaticamente reconfigurados de forma transparente ao usuário.
- **Amplo acesso** – O acesso aos recursos é disponibilizado através de mecanismos padronizados, o que possibilita o uso de diversas plataformas como celulares, laptops e PDAs.
- **Pooling de recursos** – Os recursos computacionais do provedor são organizados em um pool para servir múltiplos usuários usando um modelo multi-tenant ou multiinquilino, com diferentes recursos físicos e virtuais, dinamicamente atribuídos e ajustados de acordo com a demanda dos usuários.
- **Elasticidade rápida** – Caso haja necessidade de escalar o serviço devido o aumento da demanda, ou então liberar recursos devido a retração dessa demanda, esses recursos podem ser adquiridos de forma rápida e elástica, em alguns casos até de forma automática.
- **Serviço medido** – Sistemas em nuvem automaticamente controlam e otimizam o uso de recursos por meio de uma capacidade de medição.

### 2.1.2 Modelos de Serviço

Os modelos de serviço na computação em nuvem definem um padrão de arquitetura para soluções de computação em nuvem. Os três modelos que compõem o ambiente de computação em nuvem são [Mell and Grance 2009]:

- **Software como Serviço (SaaS)** – O modelo de SaaS proporciona *softwares* com propósitos específicos que são disponíveis para os usuários

através da Internet. Os *softwares* são acessíveis a partir de vários dispositivos do usuário por meio de uma interface *thin client* como um navegador Web. No SaaS, o usuário não administra ou controla a infraestrutura subjacente, incluindo rede, servidores, sistemas operacionais, armazenamento, ou mesmo as características individuais da aplicação, exceto configurações específicas.

- **Plataforma como Serviço (PaaS)** – O modelo de PaaS fornece sistema operacional, linguagens de programação e ambientes de desenvolvimento para as aplicações, auxiliando a implementação de sistemas de *software*. Assim como no SaaS, o usuário não administra ou controla a infraestrutura subjacente, mas tem controle sobre as aplicações implantadas e, possivelmente, as configurações de aplicações hospedadas nesta infraestrutura.
- **Infraestrutura como Serviço (IaaS)** – A IaaS torna mais fácil e acessível o fornecimento de recursos, tais como servidores, rede, armazenamento e outros recursos de computação fundamentais para construir um ambiente de aplicação sob demanda, que podem incluir sistemas operacionais e aplicativos. Em geral, o usuário não administra ou controla a infraestrutura da nuvem, mas tem controle sobre os sistemas operacionais, armazenamento, aplicativos implantados e, eventualmente, seleciona componentes de rede, tais como firewalls.

### 2.1.3 Modelos de Implantação

Os modelos de implantação para computação em nuvem definem os tipos de restrições e a disponibilidade deste ambiente. A escolha do modelo de implantação a ser utilizado depende dos processos de negócio, do tipo de informação e do nível de visão desejado [Mell and Grance 2009].

- **Nuvem privada** – a infraestrutura de nuvem é utilizada exclusivamente por uma organização, sendo esta nuvem local ou remota e administrada pela própria empresa ou por terceiros.

- **Nuvem pública** – a infraestrutura de nuvem é disponibilizada para o público em geral, sendo acessada por qualquer usuário que conheça a localização do serviço.
- **Nuvem comunidade** – fornece uma infraestrutura compartilhada por uma comunidade de organizações com interesses em comum.
- **Nuvem Híbrida** – a infraestrutura é uma composição de duas ou mais nuvens, que podem ser do tipo privada, pública ou comunidade, e que continuam a ser entidades únicas apesar de conectadas por meio de tecnologia proprietária ou padronizada que permite a portabilidade de dados e aplicações.

## 2.2 Arquitetura Orientada a Serviços

A Arquitetura Orientada a Serviços (SOA) está sendo muito utilizada como forma de *software* na indústria. Esta arquitetura permite que serviços, já existentes ou desenvolvidos a partir de funcionalidades de aplicações já existentes, possam ser combinados para criar novas aplicações [Sousa 2010]. Uma vez criado um serviço, este pode ser reutilizado ou recombinado na construção de diferentes sistemas. Neste sentido, SOA foi desenvolvida no intuito de prover, ao *software*, agilidade e uma arquitetura flexível, adaptável e extensível.

Uma característica importante em uma SOA é sua capacidade de auxiliar no processo de migração do *software*. Com isso, os projetistas e desenvolvedores devem levar em consideração a identificação e mineração de serviços existentes nos sistemas legados, a integração de serviços individuais e/ou compartilhados, o custo para aquisição de uma infraestrutura para abrigar uma solução SOA, o desenvolvimento de aplicações para serviços, o desenvolvimento de políticas de segurança e a análise de desempenho, escalabilidade e outros requisitos de *Quality of Service* (QoS) [O'Brien et al. 2008].

Em uma SOA, um serviço é uma função de um sistema computacional que é disponibilizado para outro sistema. Um serviço pode operar de forma independente ou em conjunto com outros serviços por meio da composição de serviços e para que possam trabalhar em conjunto, serviços necessitam possuir uma interface de comunicação bem definida. Geralmente se utiliza de *WebServices* para realizar a comunicação entre o sistema cliente e aquele que disponibiliza o serviço. Os *WebServices* são componentes que permitem as aplicações enviar e receber dados em um formato padrão, como por exemplo, *eXtensible Markup Language* (XML).

O termo SOA expressa um conceito em que aplicativos são disponibilizados como serviços em uma rede de computadores de forma independente e se comunicando por meio de padrões. Portanto, uma implementação maior de SOA pode utilizar qualquer tecnologia padronizada baseada na Web. A maior parte das implementações de SOA se utilizam de WebServices, *Simple Object Access Protocol* (SOAP) e *Web Services Description Language* (WSDL). SOAP é um protocolo padronizado para troca de informações estruturadas entre plataforma distribuída usando como base o XML. Já WSDL trata-se de um documento XML que descreve o serviço, especifica como acessá-lo e quais operações ou métodos estão disponíveis. Uma *Universal Description, Discovery and Integration* (UDDI) atua como um meio de realizar uma especificação que define um serviço de registro para WebServices. Os provedores podem utilizar UDDI para publicar os serviços que eles oferecem.

### **2.3 Qualidade de Serviço e Monitoramento de Serviços de Dados em Nuvem**

Na nuvem, por definição, o usuário do serviço tem algumas garantias, tais como melhor desempenho e maior disponibilidade. Apesar das limitações de rede e segurança, as soluções em nuvem devem fornecer elevado desempenho, além de serem flexíveis para se adaptar diante de uma determinada quantidade de requisições [Sousa et al. 2010]. Como, em geral, os ambientes de computação em nuvem possuem acesso público, a quantidade de requisições realizadas é imprevisível e variável, o que dificulta fazer estimativas e fornecer garantias de qualidade do serviço. Estimar métricas de qualidade é um desafio nestes ambientes, pois os recursos gerenciados estão, frequentemente, sendo expandidos ou realocados com o objetivo de melhorar o desempenho.

Essas garantias de qualidade do serviço são definidas entre o provedor do serviço e o usuário, e expressas por meio de um acordo de nível de serviço (SLA), que consiste de contratos que especificam um nível de desempenho que deve ser atendido e penalidades em caso de falha [Sousa et al. 2010]. Questões de qualidade de serviço podem ser abordadas sob vários pontos de vista, tais como prestar um serviço sujeito a restrições de desempenho ou como descobrir e selecionar dinamicamente um serviço com requisitos de desempenho.

No caso específico de serviços de dados, embora seja relativamente fácil de controlar o SLA para um ou poucos bancos de dados pela adição de recursos, isto se torna um problema complexo de gerenciamento quando se necessita cumprir SLAs para milhares de aplicações que utilizam recursos compartilhados, visto que utilizar recursos dedicados não é

uma opção economicamente viável devido a grande quantidade de aplicações [Sousa et al 2010].

Muitas empresas, por exemplo, precisam exibir uma página web dentro de um determinado intervalo de tempo e esperam que os provedores de nuvem garantam a qualidade do serviço utilizando SLAs como base para características de desempenho. Contudo, em geral, os provedores baseiam seus SLAs apenas na disponibilidade dos serviços oferecidos, ao passo que os serviços em nuvem apresentam uma variabilidade de desempenho bastante elevada. Portanto, é importante que os provedores ofereçam SLAs baseados em desempenho para os usuários [Sousa et al 2010].

A computação em nuvem é um sistema autônomo gerenciado de forma transparente para os usuários. *Hardware* e *software* dentro de nuvens podem ser automaticamente reconfigurados, orquestrados e estas modificações são apresentadas ao usuário como uma imagem única. Essa autonomia é importante porque reduz o custo de equipe de monitoramento do sistema tanto no âmbito centralizado quanto distribuído. Comparados com sistemas tradicionais, é possível identificar três fatores complexos: intervenção humana limitada, alta alternância na carga de processamento e uma variedade de infraestruturas compartilhadas. Na maioria dos casos, não existem administradores de sistemas para ajudar os desenvolvedores que acessam a nuvem, fazendo com que a plataforma seja automatizada ao máximo e que os usuários possam variar a carga de trabalho habitual, necessitando de uma infraestrutura de virtualização eficaz.

O monitoramento do uso de recursos é essencial para o ambiente de computação em nuvem, visto que este ambiente está se alterando constantemente de forma a atingir os objetivos. Os recursos são monitorados constantemente para que se certifique que estão atingindo seus objetivos, e que armazenam informações em logs utilizadas para a sua adaptação, auto otimização e reconfiguração. Os recursos também são monitorados por outros serviços externos com o objetivo de verificar se os mesmos estão recebendo os serviços conforme estabelecido e se estão fornecendo os serviços conforme acertado.

No ambiente em nuvem, o objetivo é minimizar a quantidade de recursos necessários para garantir a qualidade do serviço, o que reduz os custos. Dessa forma, uma questão é como gerenciar de forma automática os recursos disponíveis e a carga de trabalho do sistema para garantir a qualidade do serviço e melhorar a utilização destes recursos. Técnicas de monitoramento adaptativas e dinâmicas são fundamentais para tratar esta questão.

O processo de monitoramento captura informações do ambiente, sejam elas físicas ou virtuais, relevantes para as propriedades do sistema. Para realizar este processo, agentes ou

sensores leem propriedades dos elementos geridos, tais como pedidos por segundo, consumo de energia, entre outros. Estes agentes podem monitorar outros elementos e emitir alertas, caso um elemento monitorado falhe. Juntamente com a correlação de eventos e outras formas de análise, o monitoramento será importante para determinar problemas e na recuperação quando se localizar ou suspeitar de uma falha. O monitoramento pode ser passivo, ou seja, sem alteração do sistema, por exemplo, processos no Unix, ou ativo, com alteração do sistema. No monitoramento ativo pode ser necessário adaptar a quantidade e o ritmo de coleta dos dados para não interferir no desempenho.

De acordo com o modelo de serviço (IaaS, PaaS ou SaaS) utilizado, o gerenciamento é diferente. Por exemplo, um provedor IaaS, em geral, gerencia apenas recursos da sua infraestrutura, ao passo que, um provedor de PaaS ou SaaS pode precisar gerenciar a utilização dos recursos em diferentes infraestruturas. Dessa forma, no modelo de PaaS e, principalmente, no modelo de SaaS deve-se ter soluções de monitoramento para melhorar a utilização dos recursos e verificar a garantia dos serviços considerando diferentes infraestruturas.

## **2.4 Trabalhos Relacionados**

Existem muitas ferramentas para gerenciar e monitorar serviços de banco de dados em nuvem. Neste trabalho consideramos apenas ferramentas para monitorar que estejam diretamente relacionadas a este trabalho. [Xeround 2011] é uma solução do banco de dados como um serviço e possui uma ferramenta web que permite ao usuário gerenciar suas instâncias de banco de dados. Ela permite ao usuário monitorar algumas informações sobre o seu banco de dados, como: uso de CPU e memória, e o número de conexões e operações por segundo. O usuário também tem a possibilidade de escalar os recursos reservados ao seu banco de dados a qualquer momento. Porém essa ferramenta permite apenas a utilização do SGBD MySQL e não é independente de plataforma, pois atualmente só é possível utilizar o serviço da Amazon como datacenter. Também não é possível adicionar mais informações que deseja monitorar do que as listadas anteriormente.

O Amazon Relational Database Service (RDS) é uma solução para a criação e acesso a um SGBD relacional em nuvem [Amazon RDS 2011]. Com isso, os usuários não precisam se preocupar em gerenciar a implantação, patches e atualizações de *software* e backups. Amazon RDS tem a opção de cinco tamanhos de instâncias diferentes e escala

verticalmente na camada de banco de dados. O Amazon RDS possui uma ferramenta de monitoramento similar ao Xeround e apresenta as mesmas limitações.

Estes trabalhos relacionados não apresentam resultados que comprovem a eficiência de suas ferramentas e as soluções por eles apresentadas não possuem portabilidade, já que são implementadas de forma específica para cada serviço.

### 3 PROCEDIMENTOS METODOLÓGICOS

Este projeto de pesquisa contemplou dois aspectos principais: a formação de aluno pesquisador e o controle e acompanhamento das fases do projeto a ser realizado por seu orientador. O projeto será dividido em subprojetos e esses são desenvolvidos pelo aluno. O controle das fases de cada subprojeto é feito através de uma ferramenta de gerência de projeto. As atividades desenvolvidas são atualizadas semanalmente na ferramenta e as metas são avaliadas e corrigidas quando ajustes de metas forem necessários. O acompanhamento de cada uma das fases dos subprojetos do aluno é feito pelo professor orientador, que realiza a análise dos relatórios gerados para cada fase e dos registros na ferramenta.

O projeto foi dividido nos seguintes subprojetos: revisão bibliográfica, onde foram levantados os conceitos e referências necessárias para o trabalho; pesquisa de trabalhos relacionados, que consistiu em pesquisar e analisar os trabalhos sobre o monitoramento de serviços de dados em nuvem; levantamento dos requisitos da aplicação, onde a ferramenta foi especificada; desenvolvimento do MDataCloud, onde foi necessário compreender aspectos relacionados a serviços web; e avaliação do MDataCloud por meio de um estudo de caso com objetivo de verificar o funcionamento da ferramenta proposta.

O MDataCloud foi implementado utilizando WebServices. A IDE utilizada no desenvolvimento foi o NetBeans 7.0 devido a sua facilidade de uso e pelo fato do aluno ter uma maior familiarização com a ferramenta. Como servidor de implantação dos serviços foi utilizado o Glassfish 3.1 devido este ser uma ferramenta livre e de manipulação simples. Também foi utilizado o *software* de virtualização Virtual Box 4.0 para configurar as máquinas virtuais utilizadas na implementação e na avaliação da ferramenta.

## 4 MDATA CLOUD

O MDataCloud, proposto neste trabalho, é uma ferramenta para o monitoramento de dados em ambiente de computação em nuvem independente de plataforma e construída com base em SOA. Esta ferramenta tem enfoque no monitoramento dos recursos de banco de dados da nuvem e, como é disponibilizada na forma de um serviço, possibilita acesso aos recursos monitorados em qualquer local. Por se encontrar no nível de Plataforma como Serviço (PaaS), esta ferramenta pode ser facilmente acoplada a outros serviços.

Segundo [Monteiro et al. 2011], as ações que podem ser aplicadas em um serviço de banco de dados em nuvem podem ser classificadas em três níveis:

- Nível do SGBD: ações nesse nível buscam alterar os parâmetros e a configuração do SGBD com o objetivo de melhorar o seu desempenho. Algumas ações que podem ser realizadas neste nível são: a criação de índices, visões materializadas, particionamento de tabelas, alteração do tamanho da área de buffer de memória, alteração no nível de multiprogramação (*MultiProgramming Level* - MPL), dentre outras.
- Nível de Máquina Virtual: a virtualização permite que máquinas virtuais compartilhem um pool de recursos físicos (memória, disco, CPU) de uma mesma máquina física. Por meio do monitor de máquinas virtuais (VMM – Virtual Machine Monitor) é possível gerenciar as VMs e, dinamicamente, alocar ou desalocar recursos de *hardware*. Este cenário torna possível um conjunto de ações que envolvem a adição de recursos físicos (ainda disponíveis) com a finalidade de assegurar o cumprimento dos SLAs.
- Nível de Nuvem: apesar da elasticidade dos recursos de *hardware*, uma máquina física possui recursos limitados, os quais podem não ser suficientes para um bom desempenho de todas as VMs. Neste caso, a infraestrutura da computação em nuvem pode ser ajustada. As ações possíveis neste nível incluem alocação de novas máquinas físicas, instanciação de novas VMs ou migração de VMs para outra máquina com uma maior quantidade de recursos físicos disponíveis.

O MDataCloud foi definido para tomar ações pertencentes ao nível de SGBD.

A seção 4.1 deste capítulo descreve aspectos relacionados a arquitetura da ferramenta. Um cenário de utilização da ferramenta é descrito na seção 4.2. Por fim, a seção 4.3 descreve como ocorreu a implementação da ferramenta.

#### 4.1 Arquitetura

O MDataCloud é composto de três serviços básicos como mostra a Figura 1: coleta, monitoramento e visualização. O serviço de coleta é responsável por obter informações sobre os recursos gerenciados. Estas informações são utilizadas pelo serviço de monitoramento, que permite ao usuário definir valores limites para os recursos gerenciados, e através do serviço de visualização, acompanhar visualmente os valores recuperados para cada recurso. Os usuários podem assim, escolher de forma simples como pretendem fazer o monitoramento do serviço de banco de dados em nuvem, especificando os critérios para o monitoramento.



Figura 1 – Serviços que Compõe o MDataCloud.

O serviço de coleta de dados é configurado em cada nó a ser monitorado. Esse serviço de coleta recebe do serviço de monitoramento uma lista de recursos que devem ser monitorados no serviço de banco de dados da máquina em que ele se encontra. Os dados coletados são armazenados temporariamente e repassados ao serviço de monitoramento a cada período de tempo definido pelo usuário. O serviço de coleta também é responsável por executar ações na máquina em que ele se encontra, ações que são enviadas pelo serviço de monitoramento, o que pode ocorrer de forma manual ou automática.

O serviço de monitoramento é responsável por coordenar os serviços de coleta, analisar os dados coletados e tomar decisões a partir deles. Para cada recurso monitorado há intervalos de valores válidos definidos por regras configuradas pelo usuário. O serviço de monitoramento toma decisões baseadas nesse conjunto de regras definidas pelo usuário.

Por fim, o serviço de visualização é responsável por facilitar a interação entre o usuário e o serviço de monitoramento. Este serviço permite ao usuário visualizar os dados coletados dos recursos que estão sendo monitorado na nuvem, incluindo o histórico de coleta e ações que foram tomadas. Este serviço também é a interface para que o usuário defina quais recursos serão monitorados e em quais máquinas, as regras que a ferramenta tomará como base para realizar monitoramento e o conjunto de ações que o serviço de monitoramento poderá tomar de forma automática.

Os três serviços que compõe o MDataCloud se comunicam e coordenam suas ações através de WebServices. A Figura 2 ilustra como ocorre a comunicação entre os serviços.

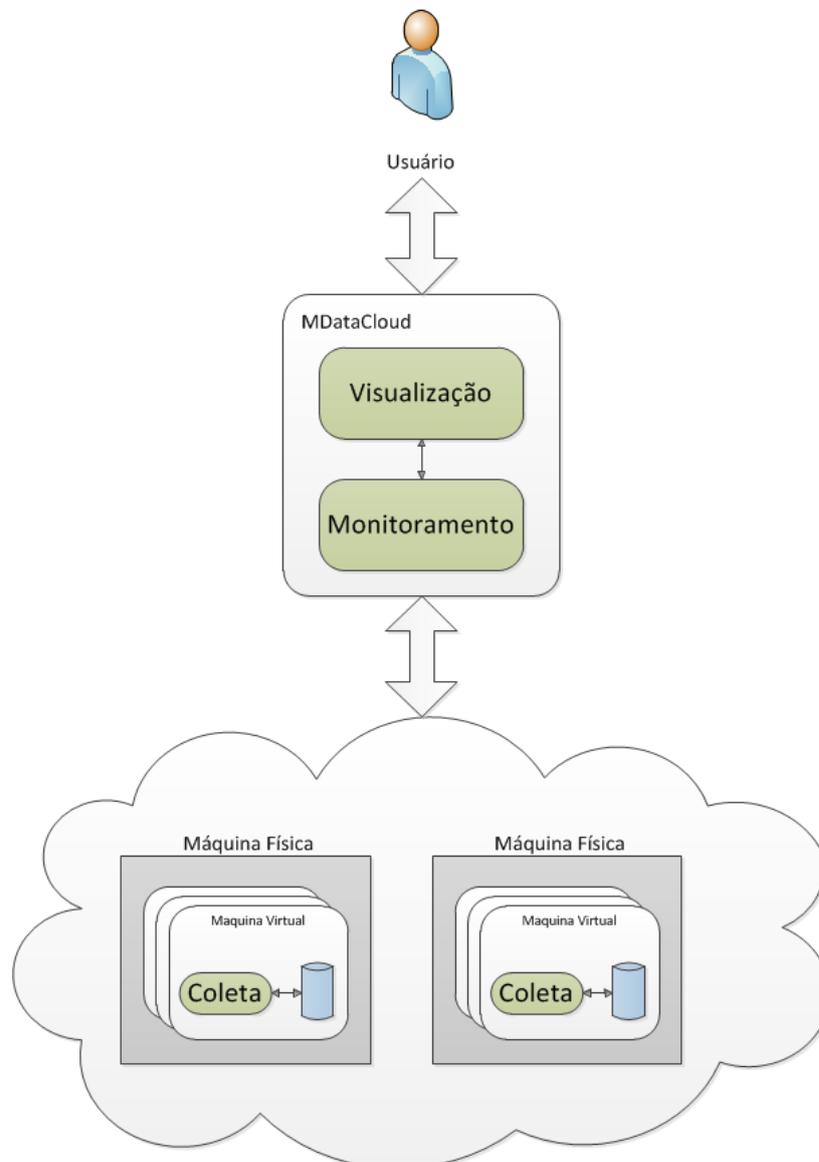


Figura 2 - Arquitetura do MDataCloud.

Para o desenvolvimento da ferramenta, definimos as seguintes entidades:

- **Parâmetros:** os itens do serviço de banco de dados a serem monitorados.
- **Regras:** entidades que definem se os valores coletados estão de acordo com os valores estabelecidos.
- **Ações:** medidas executadas para que o serviço de banco de dados se mantenha dentro dos valores estabelecidos.

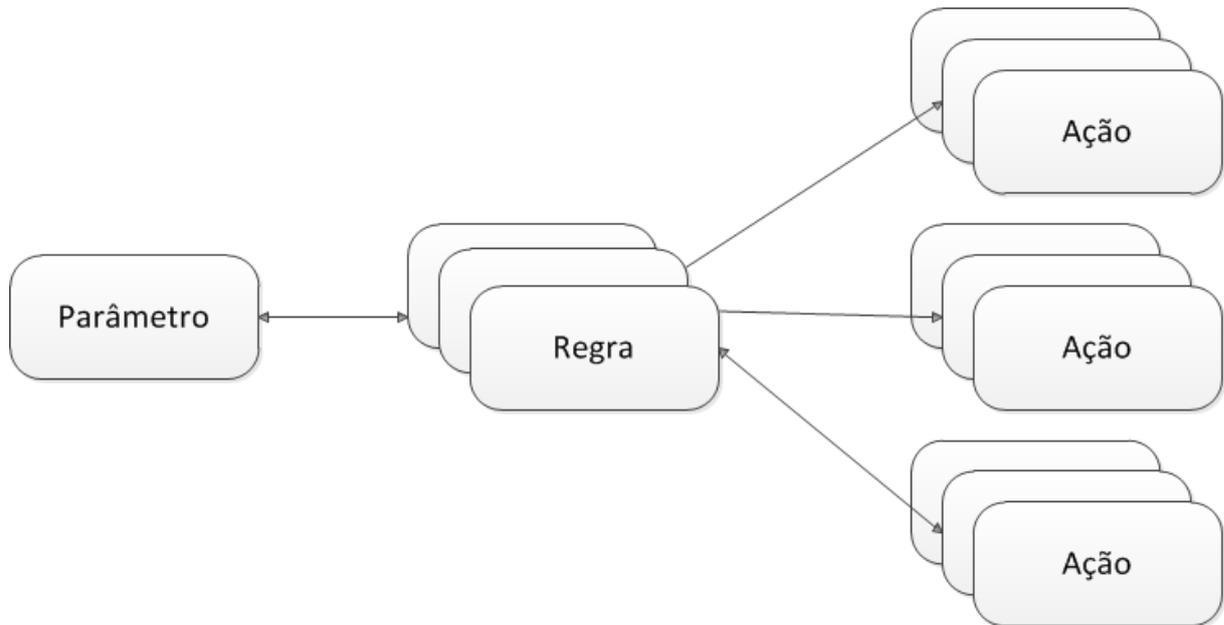


Figura 3 - Relacionamento Entre as Entidades.

Como mostra a Figura 3, um parâmetro está relacionado com 0 ou mais regras enquanto que uma regra está relacionada a apenas um parâmetro. A relação entre regras e ações é do tipo m para n: uma regra pode ter 0 ou mais ações associada a ela e uma ação também pode ter 0 ou mais regras associadas a ela.

Para cada recurso a ser monitorado, tem-se a uma entidade Parâmetro. Essa entidade contém um nome para identificação e um comando a ser utilizado pela ferramenta para a recuperação dessa informação. Se o usuário deseja recuperar o tempo de resposta, por exemplo, ele deve criar um novo parâmetro denominado “tempo de resposta” e indicar qual comando a ferramenta deve executar para que essa informação seja recuperada do serviço de banco de dados.

Para cada parâmetro, temos um conjunto de regras associadas que indicam valores de mínimo e de máximo esperados para a informação recuperada. Quando uma determinada regra é satisfeita, a ferramenta executa uma ação que está associada a essa regra. Como exemplificado na Figura 4, ao determinarmos que o parâmetro tempo de resposta deva ficar

entre 10ms e 60ms, garantir que este parâmetro fique nesse intervalo requer que se observe dois casos. O primeiro caso ocorre quando o valor estiver entre 0ms e 10ms e pode ser tratado criando-se uma Regra 1 com valor mínimo 0ms e valor máximo 10ms. O segundo caso ocorre quando os valores estiverem acima de 60ms, e que pode ser dividido em dois subcasos. O usuário pode entender que entre 60ms e 80ms a situação não é crítica e definir uma segunda regra que executa uma ação que faz pequenas alterações. Quando os valores estiverem acima de 80ms, pode-se definir uma regra 3 que faz alterações maiores na configuração do banco de dados.



Figura 4 – Como as Regras são Definidas.

As ações determinam quais medidas a ferramenta deve tomar caso alguma regra seja atendida. Seguindo o exemplo anterior, quando o parâmetro tempo de resposta atender a Regra 2, esta deve acionar uma ação associada que pode ser a criação de novos índices no banco de dados visando voltar o tempo de resposta ao intervalo especificado. A Regra 3, por sua vez, pode levar ao aumento do tamanho do *buffer* de memória. Note que para uma determinada regra poderá existir n ações.

## 4.2 Cenário de Utilização

Para ilustrar o funcionamento do MDataCloud, destacamos um cenário de sua utilização. Supondo que um parâmetro denominado “Tempo de Resposta” esteja sendo monitorado pela ferramenta como mostra a Figura 5. Para este parâmetro estabelecemos que seu valor não deve ser inferior a 10ms nem superior a 60ms e definimos as seguintes regras de monitoramento: a “Regra 1” determina que, se o valor obtido estiver entre 0ms e 10ms, a ferramenta executará a “Ação 1” que consiste em diminuir o tamanho da área do buffer de memória; a “Regra 2” determina que, quando o valor do parâmetro “Tempo de Resposta” estiver entre 60ms e 80ms, a ferramenta deve executar a “Ação 2”, que cria novos índices no serviço de banco de dados; podemos ainda querer executar uma ação para quando os valores ultrapassarem 80ms, definimos então a “Regra 3” que executa a “Ação 3” que aumenta o tamanho do buffer de memória utilizado.

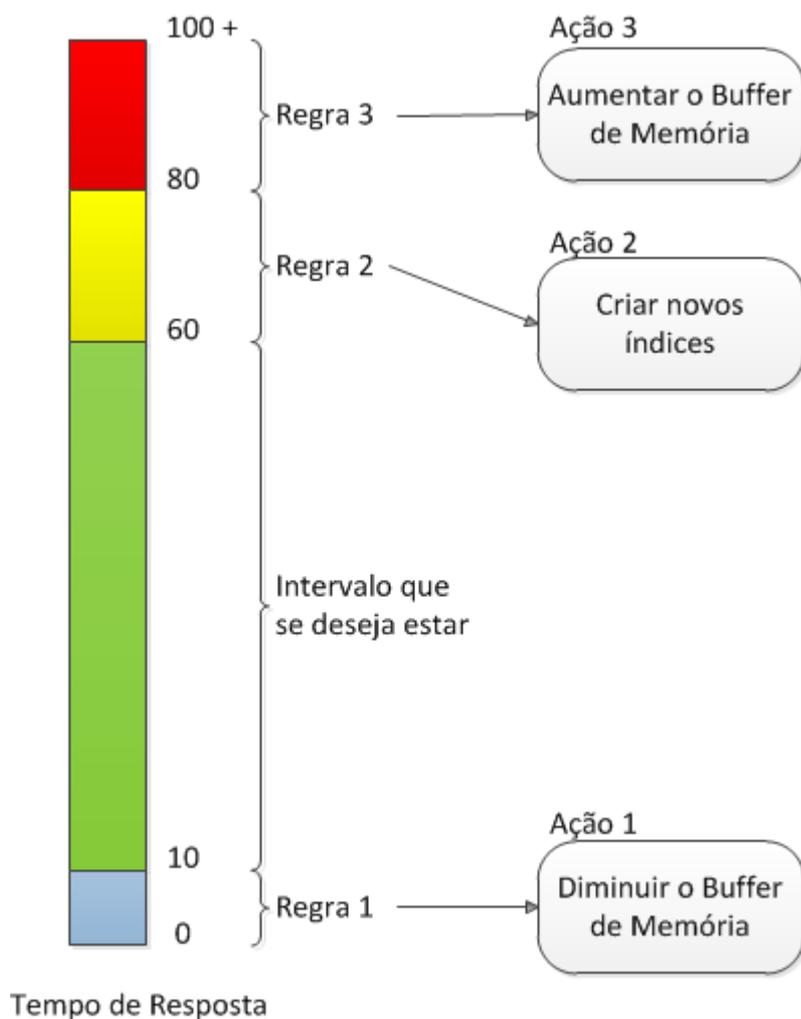


Figura 5 – Exemplo de Configuração.

A cada determinado período de tempo configurado na ferramenta, o serviço de coleta, configurado em cada nó que fornece o serviço de banco de dados, captura informações sobre o parâmetro “Tempo de Resposta” e os envia ao serviço de monitoramento. O serviço de monitoramento por sua vez, salva os dados recebidos e os analisa. Quando o valor médio de algum desses parâmetros atender uma de suas regras, o serviço de monitoramento acionará a ação associada a essa regra. Por exemplo, se para o parâmetro “Tempo de Resposta” o tempo médio ficar menor que 10ms, temos que a “Regra 1” foi atendida, então a ferramenta executará a “Ação 1” na máquina em que esses valores estão sendo recuperados. Da mesma forma, se o tempo de resposta ficar acima de 80ms, a ferramenta executará a “Ação 3” para tentar regularizar esse tempo, pois a “Regra 3” foi atendida.

### 4.3 Implementação

O MDataCloud foi implementado utilizando-se de WebServices na linguagem Java. A IDE utilizada no desenvolvimento foi o NetBeans 7.0 devido a sua facilidade de uso e pelo fato do aluno ter uma maior familiarização com a ferramenta. Como servidor de implantação dos serviços foi utilizado o Glassfish 3.1 devido este ser uma ferramenta livre e de manipulação simples. Esta seção ilustra como os principais serviços do MDataCloud foi implementado.

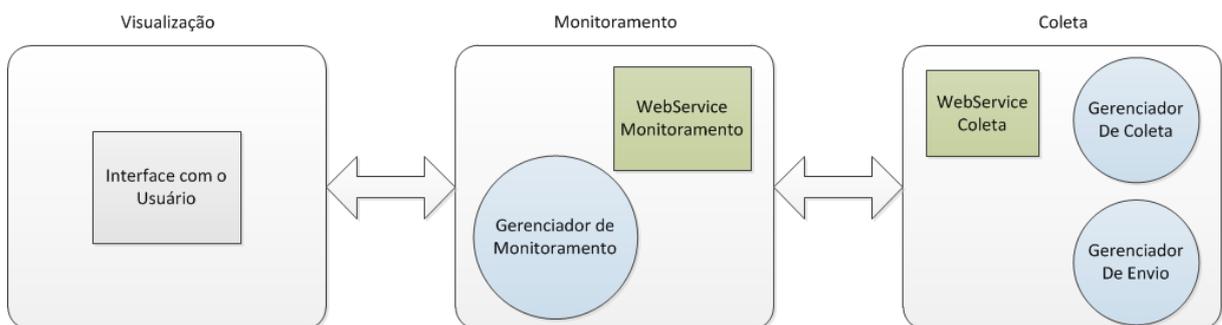


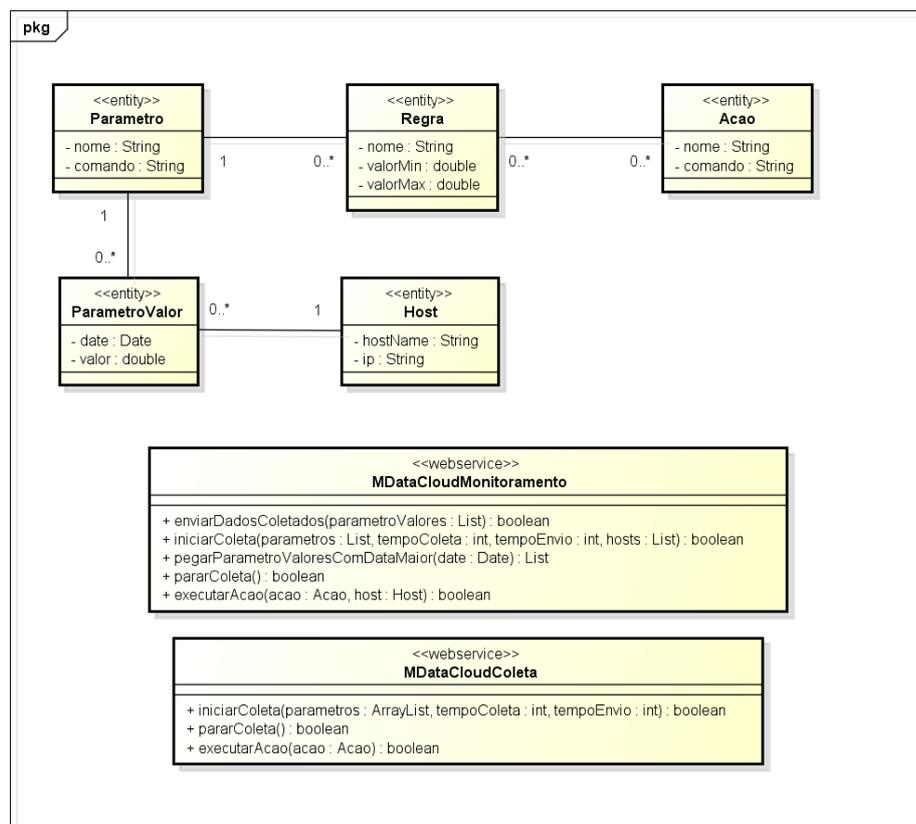
Figura 6 - Relacionamento Entre os Serviços.

O serviço de coleta é capaz de iniciar a coleta, parar a coleta e executar uma determinada ação no nó em que está configurado. Como podemos ver na Figura 6, foi implementada uma *thread* no serviço de coleta que é responsável por fazer a coleta das informações chamada Gerenciador de Coleta. Essa *thread* contém uma lista dos parâmetros que deve coletar a cada intervalo de tempo definido pelo usuário e armazenar os dados

coletados temporariamente em um banco de dados local. Uma segunda *thread*, denominada Gerenciador de Envio, é responsável por enviar esses dados coletados a cada intervalo de tempo também definido pelo usuário.

O serviço de monitoramento possui um *WebService* para se comunicar com o serviço de visualização, através desse *WebService* o serviço de visualização pode iniciar o monitoramento, recuperar os dados coletados, executar ações em determinados nós, dentre outras. Esse serviço possui uma *thread* chamada Gerenciador de Monitoramento que é responsável por monitorar os dados que estão sendo coletados. Toda vez que o monitoramento é iniciado, esse *thread* é iniciada para monitorar os valores e executar ações nos nós de forma automática.

O serviço de visualização é responsável por fazer a comunicação entre o usuário da ferramenta e o serviço de monitoramento, facilitando assim, o uso da ferramenta e permitindo ao usuário configurá-la de acordo com suas necessidades.



powered by astah®

Figura 7 - Diagrama de Classes.

O diagrama de classes representado na Figura 7 mostra como foram implementadas as entidade principais do MDataCloud e o relacionamento entre cada uma

delas. A classe parâmetro representa o recurso que queremos coletar do banco de dados, cada parâmetro possui um conjunto de regras que estão associadas a ele. Essas regras determinam intervalo de valores que acionam uma ação que está associada à Regra. Uma ação representa um determinado comando a ser executado no banco de dados a fim de trazer o valor de um determinado parâmetro para uma faixa considerada aceitável.

A classe Host representa cada nó que será monitorado pela ferramenta. E por fim a entidade parâmetro valor representa cada valor coletado para um determinado parâmetro em um determinado nó, ela armazena o nó que um determinado valor foi coletado e qual parâmetro está associado a este valor.

A classe MDataCloudMonitoramento é o WebService responsável pela comunicação com o serviço de monitoramento. No diagrama estão representados os métodos mais importantes desse WebService. O método + *enviarDadosColetados(parametroValores : List) : boolean* é usado pelo serviço de coleta quando este deseja enviar os dados que foram coletados para aquele nó e passa como parâmetro os valores que foram coletados. Para iniciar a coleta de dados e o monitoramento, o serviço de visualização chama o método + *iniciarColeta(parametros : List, tempoColeta : int, tempoEnvio : int, hosts : List) : boolean* passando os parâmetros que se deseja coletar, o tempo em segundos que o serviço de coleta deve coletar os dados, o tempo em segundos que o serviço de coleta deve enviar os dados para o serviço de monitoramento e quais nós se deseja monitorar. Também é possível que o serviço de visualização recupere os valores coletados a partir de uma determinada data através do método + *pegarParametroValoresComDataMaior(date : Date) : List*. Os métodos para parar a coleta e para executar uma ação manualmente são respectivamente + *pararColeta() : boolean* e + *executarAcao(acao : Acao, host : Host) : boolean*.

O serviço de coleta é representado na Figura 7 pela classe MDataCloudColeta. Neste serviço podemos iniciar a coleta onde o serviço está implantado utilizando o método + *iniciarColeta(parametros : ArrayList, tempoColeta : int, tempoEnvio : int) : boolean*, parar a coleta com o método + *pararColeta() : boolean* e executar uma ação no nó onde o serviço está implantado com o método + *executarAcao(acao : Acao) : boolean*.

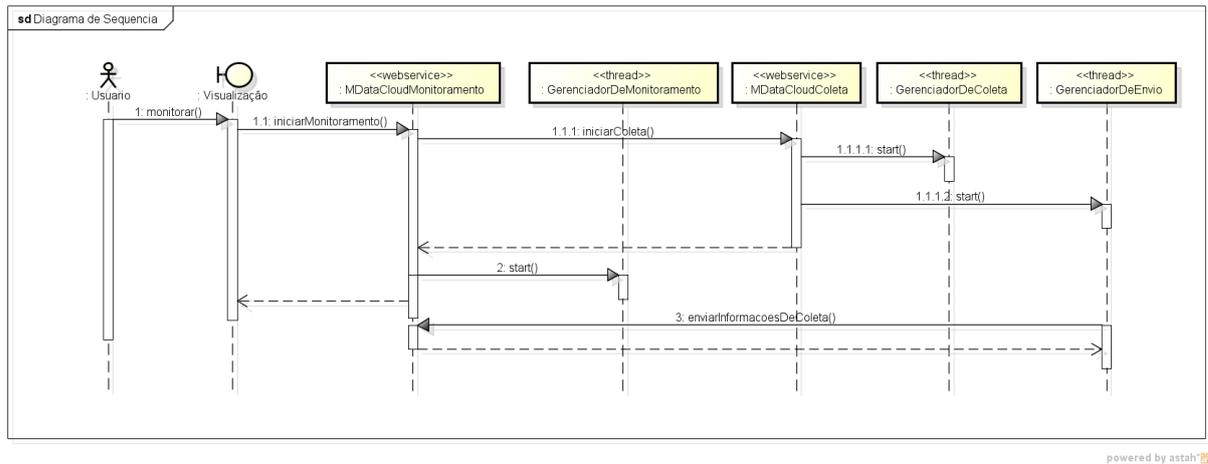


Figura 8 - Diagrama de Sequência.

O diagrama de sequência mostrado na Figura 8 representa o fluxo de mensagens trocadas pela ferramenta ao se iniciar o monitoramento. O usuário interage com o serviço de visualização escolhendo quais parâmetros e quais nós ele deseja monitorar e o serviço de visualização aciona o serviço de monitoramento passando-lhe as informações que o usuário escolheu.

Ao receber a mensagem de início de coleta, o serviço de monitoramento irá acionar o serviço de iniciar coleta em cada nó escolhido passado para cada um dos parâmetros que eles devem monitorar. Após iniciar os serviços nos nós, o serviço de monitoramento iniciará a *thread* responsável por ficar analisando os valores que foram coletados pelo serviço de coleta de cada nó. Ao perceber que uma determinada regra foi atendida, a *thread* responsável por analisar os valores recebidos irá executar uma determinada ação no nó em que a anormalidade foi detectada.

Quando o início da coleta é acionado no serviço de coleta, esse serviço irá iniciar uma *thread* que gerencia a coleta e outra que gerencia o envio de dados ao serviço de monitoramento. A *thread* de coleta irá recuperar de tempos em tempos os valores para cada parâmetro recebido e irá salva-los em uma base de dados local. Por fim, a *thread* de envio irá de tempos em tempos enviar esses dados coletados ao serviço de monitoramento. Este serviço é responsável por analisá-los e executar ações de acordo com as regras e ações cadastradas pelo usuário.

## 5 AVALIAÇÃO

A avaliação desse trabalho foi realizada por meio de um estudo de caso, que consistiu em executar a ferramenta e verificar sua conformidade. Utilizando o serviço de visualização o usuário pode acessar facilmente a ferramenta. Neste serviço o usuário pode visualizar, editar, excluir e criar novas regras, ações e parâmetros. Também é possível iniciar ou parar a coleta, visualizar os dados que estão sendo coletados, executar uma ação de forma manual em algum nó e visualizar o histórico de ações que foram tomadas pela ferramenta.

A Figura 9 ilustra como adicionar um novo parâmetro na ferramenta. Ao abrir essa tela, o usuário deve definir um nome para o novo parâmetro e qual comando deve ser executado para que este parâmetro seja recuperado. Clicando no botão “Enviar” o parâmetro será adicionado na ferramenta. De forma semelhante temos para a ação. Ao cadastrar uma nova ação, o usuário deve inserir o nome e comando que esta ação deve executar no serviço de banco de dados ao ser executada nos campos nome e comando como mostra a figura abaixo. Clicando em enviar o usuário insere uma nova ação na ferramenta.



Figura 9 - Adicionar Parâmetro/Adicionar Ação.

A tela lista parâmetros permite visualizar todos os parâmetros configurados na ferramenta. Como mostra a Figura 10 é exibida a identificação, o nome e o comando de cada parâmetro cadastrado na ferramenta. A partir dessa lista é possível editar ou excluir cada parâmetro.



Figura 10 - Adicionar Parâmetro/Adicionar Ação.

Podemos na tela listar ações, visualizar as ações que estão cadastradas na ferramenta. A partir desta tela podemos alterar ou excluir uma determinada ação de forma semelhante como é feito para parâmetros.

Depois de inserirmos os parâmetros e as ações que a ferramenta executará podemos definir as regras de monitoramento. Como mostra a Figura 11, para definir uma nova regra, devemos informar o nome desta regra, o valor mínimo e o valor máximo que esta deve assumir, o parâmetro que está associado a ela, e selecionar quais ações ela deve executar quando atendida.



Figura 11 - Adicionar Regra.

Também podemos listar as regras que estão definidas na aplicação da mesma forma como é feito para parâmetros e ações. A tela lista regras mostrada na Figura 12 nos permite visualizar todas as regras definidas. A partir desta tela podemos editar ou excluir uma determinada regra existente.

**Listar Regras**

- [Início](#)
- [Iniciar Coleta](#)
- [Listar Parametro Valores](#)
- [Ações](#)
- [Hosts](#)
- [Regras](#)
- [Parametros](#)
- [Executar Ação](#)
- [Histórico de Ações](#)

[Adicionar](#) [Atualizar](#)

ID	Nome	Valor Mínimo	Valor Máximo	Parametro	
1	Regra 1	0.0	10.0	Tempo de Resposta	<a href="#">Alterar</a>   <a href="#">Excluir</a>
2	Regra 2	60.0	80.0	Tempo de Resposta	<a href="#">Alterar</a>   <a href="#">Excluir</a>
3	Regra 3	80.0	63000.0	Tempo de Resposta	<a href="#">Alterar</a>   <a href="#">Excluir</a>

Figura 12 - Listar Regras.

Para iniciar a coleta, devemos informar à ferramenta quais parâmetros serão monitorados e em quais nós iremos coletar esses parâmetros. Nesta tela também definimos o tempo em segundo que essa coleta será realizada, por exemplo, se no campo tempo de coleta estiver com o valor igual a 2, então a cada dois segundos o serviço de coleta irá coletar dados sobre os parâmetros. Além do tempo de coleta, definimos aqui o tempo de envio, que é o período de tempo em que o serviço de coleta enviará as informações coletadas para o serviço de monitoramento.

**Iniciar Coleta**

- [Início](#)
- [Iniciar Coleta](#)
- [Listar Parametro Valores](#)
- [Ações](#)
- [Hosts](#)
- [Regras](#)
- [Parametros](#)
- [Executar Ação](#)
- [Histórico de Ações](#)

Tempo de Coleta:	20
Tempo de Envio:	60
Parametros	<input checked="" type="checkbox"/> Tempo de Resposta <input checked="" type="checkbox"/> Vazao
Hosts:	<input checked="" type="checkbox"/> 192.168.0.11 <input checked="" type="checkbox"/> 192.168.0.13
<input type="button" value="Iniciar"/> <input type="button" value="Limpar"/>	

Figura 13 - Iniciar Coleta.

Podemos também visualizar os dados que foram coletados pela ferramenta. A figura abaixo mostra a tela em que visualizamos esses valores. Na tela mostrada na Figura 14 é possível selecionar a data de listagem. O usuário pode escolher por visualizar os valores coletados no ultimo minuto, hora, dia, semana ou mês. Dessa forma o usuário consegue visualizar um histórico de valores coletados.

**Listar Parametro Valores**

- [Inicio](#)
- [Iniciar Coleta](#)
- [Listar Parametro Valores](#)
- [Ações](#)
- [Hosts](#)
- [Regras](#)
- [Parametros](#)
- [Executar Ação](#)
- [Histórico de Ações](#)

Último Minuto

Parar Coleta

Nome Parametro	Data/Hora	Valor	Host
Tempo de Resposta	2011-05-22T16:51:55-03:00	25.0	192.168.0.11
Vazao	2011-05-22T16:51:55-03:00	40.0	192.168.0.11
Tempo de Resposta	2011-05-22T16:51:56-03:00	20.0	192.168.0.13
Vazao	2011-05-22T16:51:56-03:00	15.0	192.168.0.13

Figura 14 - Listar Parâmetros Valores.

Também é possível executar uma ação manualmente como mostra a Figura 15. Para tanto, é necessário apenas selecionar a ação que desejamos executar e o nó em que queremos executar essa ação.

**Executar Ação**

- [Inicio](#)
- [Iniciar Coleta](#)
- [Listar Parametro Valores](#)
- [Ações](#)
- [Hosts](#)
- [Regras](#)
- [Parametros](#)
- [Executar Ação](#)
- [Histórico de Ações](#)

Ação: Criar novos Indices

Host: 192.168.0.11

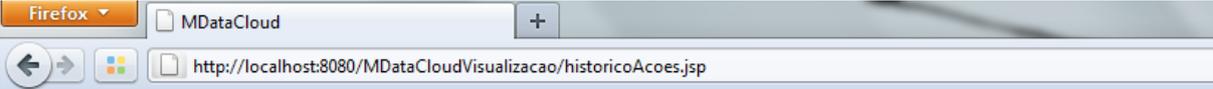
Figura 15 – Executar Ação de Forma Manual.

Depois de configurar a ferramenta, podemos verificar se esta irá se comportar da maneira que supomos. Para realizar testes na ferramenta configuramos duas máquinas virtuais com sistema operacional Ubuntu 10.10, SGBD MySql 5.5, e servidor Glassfish 3.1 para implantação do serviço de coleta. Para hospedar o serviço de monitoramento utilizamos uma máquina com servidor Glassfish 3.1 e Windows 7 Home Premiun.

Inserimos um parâmetro na ferramenta chamado “Tempo de Resposta” e criamos uma tabela no banco de dados de cada VM para coletar os valores para esse parâmetro. Criamos três regras para monitorarmos: a primeira regra analisa se os valores coletados para esse parâmetro estão no intervalo de 0 a 10. Caso seja atendida esta regra aciona uma ação que diminui o tamanho do buffer de memória. A segunda regra verifica se os valores coletados estão entre 60 e 80, e executa uma ação que cria novos índices no serviço de banco de dados. Por fim, a terceira regra verifica se os valores estão acima de 80 e chama uma ação que aumenta o tamanho do buffer de memória quando é atendida. Utilizamos o exemplo de avaliação conforme o explicado no item 4.2.

Iniciamos o serviço de coleta configurando o tempo de coleta de dados para 15 segundos, e o tempo de envio de dados ao serviço de monitoramento a cada 30 segundos. Depois de um tempo, alteramos o valor do parâmetro coletado no nó de ip 192.168.0.13 de forma que este atendesse a primeira regra. Como mostra a Figura 16, a ferramenta detectou que essa regra estava sendo atendida e executou a ação que diminui o tamanho do buffer de memória. Quando a ação foi executada voltamos o valor do parâmetro para o esperado.

Após verificarmos a execução da Ação 1, alteramos o valor coletado no nó de ip 192.168.0.11 para que a segunda regra fosse atendida para este nó, aguardamos um pouco e observou-se que a ferramenta executou a ação “Criar novos Índices” no nó correspondente de forma automática.



**Histórico de Ações**

- [Início](#)
- [Iniciar Coleta](#)
- [Listar Parametro Valores](#)
- [Ações](#)
- [Hosts](#)
- [Regras](#)
- [Parametros](#)
- [Executar Ação](#)
- [Histórico de Ações](#)

[Atualizar](#)

Ação	Host	Data	Tipo de Execução
Aumentar Buffer de Memoria	192.168.0.13	2011-05-22T17:25:19-03:00	AUTOMATICA
Criar novos Indices	192.168.0.11	2011-05-22T17:21:19-03:00	AUTOMATICA
Diminuir Buffer de Memoria	192.168.0.13	2011-05-22T17:16:19-03:00	AUTOMATICA
Diminuir Buffer de Memoria	192.168.0.13	2011-05-22T17:17:19-03:00	AUTOMATICA

Figura 16 – Ações Executadas.

Da mesma forma, alteramos os valores novamente no nó de ip 192.168.0.13 para que desta vez a regra 3 fosse atendida. Como era esperada, a ferramenta executou automaticamente a ação que aumenta o buffer de memória no nó em que os dados coletados atenderam a regra.

Assim foram executados testes que avaliaram se a ferramenta se comporta da forma esperada, executando ações de forma autônoma baseada nas configurações feitas pelo usuário. Verificamos também que a comunicação entre os serviços que a ferramenta oferece também ocorre de forma satisfatória.

## 6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o MDataCloud, uma ferramenta criada com objetivo de ser extensível, independente de plataforma e disponibilizada como um serviço. Por meio dessa ferramenta, os administradores de serviço de banco de dados em nuvem podem facilmente monitorar seus recursos e executar ações sobre estes serviços de forma automática, visando manter os níveis de acordo de serviço com os usuários da nuvem. A avaliação dessa ferramenta mostrou que a comunicação e a coordenação dos diversos serviços que a constitui foi implementada de forma satisfatória. Também concluímos que o MDataCloud consegue analisar os dados dos recursos que estão sendo monitorados e tomar decisões sobre eles de forma automática.

Como trabalhos futuros, pretendemos avaliar a ferramenta em um ambiente de nuvem real, utilizando um *benchmark* para gerar carga no sistema de banco de dados para verificar o comportamento da ferramenta. Pretendemos também adicionar novas funcionalidades a ferramenta, tais como permitir a visualização gráfica dos recursos que estão sendo monitorados e executar ações nos níveis de máquina virtual e nuvem.

## REFERÊNCIAS

[Amazon RDS 2011] Amazon. <http://aws.amazon.com/rds/>

[Brantner et al. 2008] Brantner, M., Florescu, D., Graf, D., Kossmann, D., and Kraska, T. (2008). Building a database on s3. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD '08, page 251, New York. ACM Press.

[Buyya et al. 2009] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. (2009). Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Gener. Comput. Syst.*, 25(6):599–616.

[Monteiro et al. 2011] Monteiro, J. M.; Macedo, J. A. F; Machado, J. C. Sintonia Automática de Banco de Dados em nuvem . In: Proceedings of the I Workshop on Autonomic Distributed Systems, WoSiDA 2011, collocated with SBRC 2011, Campo Grande, MS.

[Mell and Grance 2009] Mell, P. and Grance, T. (2009). Draft NIST Working Definition of Cloud Computing. National Institute of Standards and Technology. <http://csrc.nist.gov/groups/SNS/cloud-computing>.

[O'Brien et al. 2008] Liam O'Brien, Paul Brebner, and Jon Gray. Business transformation to soa: aspects of the migration and performance and qos issues. In SDSOA '08: Proceedings of the 2nd international workshop on Systems development in SOA environments, pages 35–40, New York, NY, USA, 2008. ACM.

[Sousa et al. 2009] Sousa, F. R. C., Moreira, L. O., e Machado, J. C. (2009). Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios. In: MOURA, R. S.; SOUZA, F. V.; OLIVEIRA, A. C. (Org.). Escola Regional de Computação (Ceará, Maranhão e Piauí, ERCEMAPI 2009, 1. ed. EDUFPI, Piauí.

[Sousa et al. 2010] Sousa, F. R. C. ; Moreira, L. O. ; Macedo, J. A. F. ; Machado, J. C. . Gerenciamento de Dados em Nuvem: Conceitos, Sistemas e Desafios. In: Pereira, A. C. M.; Pappa, G. L.; Winckler, M.; Gomes, R. L.. (Org.). Tópicos em Sistemas Colaborativos, Interativos, Multimídia, Web e Bancos de Dados. Belo Horizonte: SBC, 2010, v. 1, p. 101-130.

[Sousa 2010] Sousa, F. R. C. Software como um Serviço. Qualificação de Doutorado. Programa de Mestrado e Doutorado em Ciência da Computação. Universidade Federal do Ceará.

[Sousa et al. 2011] Sousa, F. R. C. ; Moreira, L. O. ; Macedo, J. A. F. ; Machado, J. C. Computação em Nuvem Autônoma: Oportunidades e Desafios. In: Proceedings of the I

Workshop on Autonomic Distributed Systems, WoSiDA 2011, collocated with SBRC 2011, Campo Grande, MS.

[Soror et al. 2010] Soror, A. A., Minhas, U. F., Abounaga, A., Salem, K., Kokosielis, P., and Kamath, S. (2010). Automatic virtual machine configuration for database workloads. *ACM Trans. Database Syst.*, 35(1):1–47.

[Xeround 2011] Xeround. <http://www.xeround.com/>