UNISINOS

Programa Interdisciplinar de Pós-Graduação em

Computação Aplicada

Mestrado Acadêmico

Júlio Cezar Santos Pires

BSPMon - Um Sistema de Monitoramento Preditivo de Recursos em Cloud Computing para Aplicações Bulk Synchronous Parallel

São Leopoldo, 2014

UNIVERSIDADE DO VALE DO RIO DOS SINOS — UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA
NÍVEL MESTRADO

JÚLIO CEZAR SANTOS PIRES

BSPMON - UM SISTEMA DE MONITORAMENTO PREDITIVO DE RECURSOS EM CLOUD COMPUTING PARA APLICAÇÕES BULK SYNCHRONOUS PARALLEL

SÃO LEOPOLDO
2014

Júlio Cezar Santos Pires

BSPMON - UM SISTEMA DE MONITORAMENTO PREDITIVO DE RECURSOS EM CLOUD COMPUTING PARA APLICAÇÕES BULK SYNCHRONOUS PARALLEL

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos — UNISINOS

Orientador:
Prof. Dr. Rodrigo da Rosa Righi

Co-orientador:
Prof. Dr. Cristiano André Costa

São Leopoldo
2014

P667b Pires, Júlio Cezar Santos

BSPMon - um sistema de monitoramento preditivo de recursos em cloud computing para aplicações Bulk Synchronous Parallel / Júlio Cezar Santos Pires -- 2014.

70 f. : il. color. ; 30cm.

Dissertação (mestrado em Computação Aplicada) -- Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, RS, 2014.

Orientador: Prof. Dr. Rodrigo da Rosa Righi; Coorientador: Prof. Dr. Cristiano André Costa.

1. Sistemas operacionais distribuídos (Computadores). 2. Computação em nuvem. 3. Monitoramento - Predição. 4. Padrões de Uso. 5. BSP. II. Título. II. Righi, Rodrigo da Rosa. III. Costa, Cristiano André.

CDU 004.75:004.451

Catalogação na Publicação: Bibliotecária Eliete Mari Doncato Brasil - CRB 10/1184

Júlio Cezar Santos Pires

BSPMon - Um Sistema de Monitoramento Preditivo de Recursos em Cloud Computing para Aplicações Bulk Synchronous Parallel

Dissertação apresentada à Universidade do Vale do Rio dos Sinos – Unisinos, como requisito parcial para obtenção do título de Mestre em Computação Aplicada.

Aprovado em 29/04/2014

BANCA EXAMINADORA

Lucas Mello Schnorr – UFRGS

Jorge Luis Victória Barbosa – UNISINOS

Rodrigo da Rosa Righi - UNISINOS

Cristiano André da Costa - UNISINOS

Prof. Dr. Rodrigo da Rosa Righi (Orientador)

Visto e permitida a impressão
São Leopoldo,

Prof. Dr. Cristiano André da Costa
Coordenador PPG em Computação Aplicada

The scientific man does not aim at an immediate result.
He does not expect that his advanced ideas will be readily taken up.
His work is like that of the planter — for the future.
His duty is to lay the foundation for those who are to come,
and point the way. He lives and labors and hopes.
– NIKOLA TESLA

AGRADECIMENTOS

Em primeiro lugar, gostaria de expressar minha profunda gratidão ao meu orientador Rodrigo Righi e ao co-orientador Cristiano Costa. Foi uma honra e um enorme privilégio ter trabalhado com ambos. Obrigado pela orientação, apoio, exemplo, motivação, paciência e conhecimentos compartilhados, que foram fundamentais para o meu crescimento pessoal e científico durante esta etapa da minha vida.

Ao Dr. Joachim Gehweiler por sempre estar aberto a questionamentos científicos e técnicos a respeito do seu trabalho de doutorado, que foi utilizado neste projeto.

Ao Programa Interdisciplinar de Pós-Graduação em Computação Aplicada da UNISINOS, PIPCA, pela oportunidade de realização deste trabalho. Aos demais professores do corpo docente pelos conhecimentos compartilhados.

RESUMO

Com os constantes avanços tecnológicos, surgem novas tendências para prover uma base de serviços para a nova era da tecnologia da informação. Com isso, surgem novos paradigmas para sistemas distribuídos, como, por exemplo, a Computação em Nuvem (ou *Cloud Computing*), que possui como ideia base a disponibilização de recursos computacionais sob demanda por meio da *Internet*, permitindo, assim, a sua utilização em qualquer lugar e pelos mais diversos tipos de aplicações. Entre as principais características da computação em nuvem, tem-se a elasticidade, provisionamento de serviço e cobrança baseada na utilização efetiva dos recursos. Visando tornar estas características, na prática, possíveis, torna-se indispensável que a infraestrutura disponha de um sistema de monitoramento. Neste contexto, este trabalho apresenta o BSPMon, um sistema de monitoramento de recursos preditivo para aplicações paralelas em *Cloud Computing*. Com o objetivo de ter um controle fino sobre os recursos computacionais, o BSPMon coletará métricas de desempenho nos três níveis da infraestrutura: máquina física, máquina virtual e aplicação, efetuando, desta forma, um monitoramento hierárquico multinível dos recursos. De posse destas métricas de desempenho, o BSPMon efetuará predições sobre as demandas, visando melhores resultados para a tomada de decisão em situações de migração, previsão, controle sobre o SLA, provisionamento e consolidação dos recursos. O sistema proposto atuará no nível de *middleware*, de forma transparente para a aplicação. A partir das avaliações obtidas na predição, os resultados apontam baixa intrusividade na infraestrutura, eficiência energética e predições com taxa de acerto superior a 90%.

Palavras-chave: Computação em Nuvem. Monitoramento. Predição. Padrões de Uso. BSP.

ABSTRACT

Due to constant technological advances, there are new trends to provide a service base for the new era of information technology. Thus, there are new paradigms for distributed systems, for example, Cloud Computing, which has as basic idea of the provision of computational resources on demand via the Internet, thus allowing their use anywhere and for many different types of applications. Among the main features of cloud computing, there is elasticity, service provisioning and billing based on the effective use of resources. In order to make these features in practice possible, it is essential that the infrastructure to have a monitoring system. In this context, this work presents the BSPMon, a monitoring system of predictive features for parallel applications in Cloud Computing. In order to have fine control over computing resources, the BSPMon collect performance metrics in three levels of infrastructure: physical machine, virtual machine and application, making thus a multilevel hierarchical resource monitoring. With such performance metrics, the BSPMon shall make predictions about the demands, to obtain better results for decision making in migration scenarios, prediction, control over the SLA, provisioning and consolidation of resources. The proposed system will operate in the middleware level, transparently to the application. From the evaluations obtained in the prediction, the results indicate low intrusiveness in infrastructure, energy efficiency and predictions accuracy rate above 90 %.

Keywords: Cloud Computing. Monitoring. Forecast. Usage Patterns. BSP.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
BSP	Bulk Synchronous Parallel
CPU	Central Processing Unit
DTW	Dynamic Iime Warping
E/S	Entrada e Saída
EC2	Elastic Compute Cloud
HPC	High Performance Computing
IAAS	Infrastructure as a Service
IP	Internet Protocol
JVM	Java Virtual Machine
JMS	Java Message Service
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ME	Mean Error
MPI	Message Passing Interface
MSE	Mean Squared Error
OCA	OpenNebula Cloud API
PAAS	Platform as a Service
QOS	Quality of Service
RAM	Random Access Memory
RPC	Remote Procedure Call
ROI	Return on Investment
SAAS	Software as a Service
SLA	Service Level Agreement
SLO	Service Level Objectives
SOM	Self-Organizing Map
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VM	Virtual Machine
XML	Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO

Cloud Computing, ou Computação em Nuvem, é um modelo para permitir o acesso sob demanda para um conjunto de recursos de computação configurável. Redes, servidores, sistemas de armazenamento, aplicativos e serviços podem ser rapidamente fornecidos e liberados com o mínimo esforço (em teoria) na interação entre usuários e provedores de serviço (VáZQUEZ et al., 2011).

Para viabilizar essa dinâmica e oferecer controle para usuários e administradores da Nuvem, a utilização de mecanismos de monitoramento de recursos é imprescindível. Elasticidade, dinamicidade da infraestrutura e das aplicações, tolerância a falhas, balanceamento de carga e acordos de nível de serviço (SLA) são características relacionadas à Nuvem e que necessitam de um sistema de monitoramento para controlá-las e efetivá-las (DE CHAVES; URIARTE; WESTPHALL, 2011).

Por meio da utilização de sistemas de monitoramento, torna-se possível a obtenção de diversas informações dos componentes da infraestrutura, aplicação ou rede. Os dados coletados por estes sistemas podem ser utilizados para detecção de anomalias, análise e predição de desempenho, além de servir como objeto de entrada para a tomada de decisão no escalonamento de recursos ou para avaliação do ambiente por um desenvolvedor visando entender o comportamento de uma determinada aplicação.

Atualmente, muitos dos sistemas computacionais desempenham um papel vital na maioria das organizações, e uma possível interrupção de um determinado serviço ou recurso pode acarretar uma grande perda financeira. Desta forma, a utilização de um sistema de monitoramento é de suma importância em ambientes dinâmicos, onde existe a necessidade de garantir que os serviços estão operacionais em tempo integral (24 horas por dia e sete dias por semana). A fim de atingir a disponibilidade e operacionalidade dos serviços, sistemas de monitoramento tornam-se peças-chave dentro de uma infraestrutura de computação em Nuvem, possibilitando um controle fino sob a utilização dos recursos existentes, bem como sua respectiva disponibilidade e detecção de violações de SLA.

Para fornecer a qualidade desejada no serviço, torna-se necessário que, além do acordo firmado pelo contrato SLA entre cliente e provedor, que os respectivos serviços sejam continuamente monitorados durante o tempo de execução, de modo a determinar a necessidade de adaptar a aplicação, provisionar recursos ou renegociar o contrato acordado previamente.

Neste contexto, um sistema de monitoramento, para se tornar efetivo, precisa tratar as etapas de coleta e análise de padrões das taxas de utilização dos recursos disponíveis dentro da infraestrutura da nuvem. A verificação destas etapas é pertinente para esse ambiente, uma vez que pode antecipar decisões de gerenciamento, melhorar o aproveitamento dos recursos disponíveis, reduzir o consumo energético e minimizar custos financeiros. O sistema de monitoramento proposto efetuará uma constante verificação sobre os recursos com o objetivo de atingir três marcos importantes: predição de desempenho, verificar possíveis falhas no SLA e,

também, estimar quando e onde é possível diminuir o consumo de energia.

Porém, mesmo com a coleta de determinadas informações de desempenho e disponibilidade dos recursos físicos e virtuais, um dado isolado por si só não é capaz de fazer o sistema efetuar a melhor tomada de decisão no momento do provisionamento e/ou consolidação dos recursos. Logo, torna-se necessária a combinação de uma ou mais métricas de desempenho de diferentes níveis da infraestrutura da Nuvem para um melhor controle sobre os recursos computacionais.

Desta forma, o sistema de monitoramento de recursos estará alinhado com as ações que podem ser tomadas por meio da combinação de dados de monitoramento de uma ou mais fontes. De posse das métricas de monitoramento, pode-se ter meios para a tomada de decisão e, assim, efetuar a migração de processos e máquinas virtuais, consolidar recursos e, consequentemente, reduzir custos por meio da economia de energia e redução nas violações de SLA. Assim, será utilizado neste trabalho o monitoramento em três níveis (máquina física, máquina virtual e aplicação), visto que esse, tende a ser mais eficiente devido à combinação dos três níveis da hierarquia dos recursos computacionais presente na infraestrutura da Nuvem.

De posse de um sistema de monitoramento eficiente, torna-se possível prover a elasticidade sobre os recursos, uma característica-chave da Nuvem. Uma das empresas mais tradicionais na área, a Amazon, disponibiliza essa propriedade com a cooperação entre dois de seus sistemas: *Amazon CloudWatch* e o *Amazon Elastic Compute Cloud* (EC2). O primeiro atua como um sistema de monitoramento, no qual são definidos alarmes e métricas para coleta de dados. O segundo permite definir regras que, quando acionadas, podem redimensionar as instâncias de máquinas virtuais em execução. Ambos possuem uma interface em linha de comando na qual é possível fazer as requisições. Cada novo conjunto de serviços a ser executado necessita da criação de novas regras específicas para atender a escalabilidade. Além disso, recursos com diferentes características podem precisar da reformulação das regras da mesma maneira.

Uma das principais vantagens de um sistema de monitoramento com a utilização de padrões de uso está relacionada à economia de energia elétrica, e esta pode ser dada pela consolidação de recursos em situações em que há subutilização destes na infraestrutura e previsão da demanda computacional a fim de estar alinhado com o contrato SLA, prevenindo multas em caso de falhas.

Um padrão de uso pode ser composto pela análise histórica de uma ou mais métricas, tais como a variação nas taxas de utilização de CPU (*Central Processing Unit*), rede, disco e memória RAM (*Random Access Memory*). Assim, é possível estabelecer curvas de comportamento e tomar ações proativas que visam trazer, além de benefícios no âmbito de energia, vantagens para usuários e administradores da Nuvem. O primeiro executa a sua aplicação com um conjunto de recursos ideal ou próximo disso para garantir o SLA. O segundo consegue planejar o uso dos recursos de sua Nuvem e, assim, pré-alocá-los a requisições que entram em vigor no futuro. Contrariamente à ideia largamente difundida, uma Nuvem tem recursos limitados e a administração deles perante as requisições de usuários é pertinente para a adoção do próprio sistema de Nuvem. Em adição, uma vez que recursos podem ser desalocados, além da diminui-

ção da energia com o desligamento das máquinas, é possível adaptar o sistema de resfriamento para consumir menos, uma vez que os alvos foram reduzidos.

A previsão de séries temporais é um dos desafios da mineração de dados. A previsão dos valores futuros é provida em função dos valores passados numa determinada série. O uso de séries temporais possui uma ampla aplicabilidade e essas podem fornecer bons modelos preditivos. As previsões presentes dos modelos de séries temporais são baseadas apenas no comportamento das variáveis nas quais se pretende efetuar a predição. Segundo (PINDYCK; RUBINFELD, 2004), um modelo de série temporal reflete o padrão de movimentos passados de uma variável e usa essa informação para prever seus movimentos futuros.

A predição de valores futuros pode ser decisiva para o sucesso ou fracasso em determinadas situações. Segundo (RUTA; GABRYS; LEMKE, 2011), a previsão de valores futuros em séries temporais é vital para a obtenção de vantagem competitiva. De acordo com (CHATFIELD, 2002), boas predições são vitais em muitas áreas, tais como: científica, industrial, comercial e atividades econômicas. Porém, para que seja possível efetuar a projeção dos valores futuros da série, é necessário que exista uma base histórica de dados das variáveis a serem previstas.

Para efetuar a predição de valores futuros nas séries temporais, torna-se necessário o uso de técnicas, algoritmos e métodos que visam auxiliar esse processo, tais como: Médias móveis (MA, do Inglês *Moving Average*), modelos autorregressivos, suavização exponencial e modelos ARIMA (*Autoregressive Integrated Moving Average*).

1.1 Objetivos

O objetivo do sistema de monitoramento proposto pode ser declarado como segue:

Dada uma aplicação de alto desempenho (High Performance Computing - HPC) desenvolvida sob o modelo BSP, um SLA e uma infraestrutura de nuvem, o sistema de monitoramento efetuará a coleta, combinação dos dados e predição de desempenho, tendo em vista a descoberta de etapas em que se deve provisionar e/ou consolidar recursos, visando o cumprimento do SLA e o melhor aproveitamento dos recursos disponíveis na infraestrutura da Nuvem.

Visando um controle mais fino sobre os recursos computacionais na infraestrutura da Nuvem, este trabalho aborda maneiras de oferecer predição de desempenho sem onerar o usuário com dados específicos de recursos ou de sua aplicação. Nesse sentido, tem-se o monitoramento multinível, ou seja, atuando nos três níveis da Nuvem e, também, nos padrões de utilização dos recursos como objeto de pesquisa para se atingir esse objetivo.

Com a utilização do sistema de monitoramento, será possível detectar oportunidades de ganho, seja esta em relação ao provisionamento ou relacionado à receita, neste caso, economizando energia com a consolidação de recursos. Além disso, o monitor detectará falhas de SLA, conforme as coletas e análises periódicas (Recurso x SLA), visando à verificação e ao cumprimento do contrato firmado entre cliente e prestador de serviço.

Desta forma, com o monitoramento nos três níveis, será possível identificar processos muito lentos ou situações muito discrepantes com a realidade. Neste caso, o sistema de monitoramento por meio de informações de desempenho das máquinas físicas, máquinas virtuais e aplicação, enviará solicitações para o administrador da Nuvem ou um determinado gerenciador / escalonador organizar o recurso afetado de maneira que este tenha um maior ganho possível dentro da infraestrutura.

Além do controle sobre a disponibilidade dos recursos, ações baseadas em padrões podem ser úteis nas seguintes situações: (i) predição de desempenho, (ii) tratamento da energia consumida pela Nuvem; (iii) migração de máquinas virtuais; (iv) migração de processos; (v) auxílio na administração dos recursos para o provedor da nuvem; (vi) possibilidade de estimar o tempo para executar um serviço; e (vii) gerenciamento do SLA entre consumidores e o provedor de serviço.

1.2 Organização do texto

Esse documento está organizado em 7 capítulos. O capítulo 1 apresentou a introdução sobre o tema objeto da pesquisa, além dos principais objetivos. O restante do texto está organizado seguindo a seguinte estrutura:

- **Capítulo 2 - Referencial Teórico:** descreve a fundamentação teórica, com conceitos e características de computação em nuvem, e seus modelos de serviço e implantação, padrões de utilização de recursos em *Cloud Computing*, monitoramento de recursos computacionais e aplicação. Por fim, é apresentado o conceito de *Service Level Agreement* (SLA);

- **Capítulo 3 - Trabalhos Relacionados:** apresenta os trabalhos relacionados a esta pesquisa, elencando as principais ferramentas e *middlewares* que possuem sistemas de monitoramento. Por fim, é feito um comparativo e são apresentadas as lacunas dos sistemas atuais;

- **Capítulo 4 - Modelo de Monitoramento Preditivo:** Neste capítulo, é apresentado o modelo do BSPMon, um sistema de monitoramento preditivo de recursos em computação em nuvem;

- **Capítulo 5 - Implementação do Modelo:** Neste capítulo, é descrito o protótipo do sistema de monitoramento preditivo de recursos baseado no modelo do capítulo 4;

- **Capítulo 6 - Avaliação:** Neste capítulo, são apresentados os resultados obtidos com do sistema de monitoramento proposto;

- **Capítulo 7 - Conclusão:** Finalizando este documento, neste capítulo, são revistas as principais contribuições e considerações finais da pesquisa, além de uma lista de possíveis trabalhos futuros, extensões e evoluções no modelo proposto.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta uma revisão da literatura sobre os conceitos fundamentais relacionados ao desenvolvimento deste trabalho, abordando, em um primeiro momento, a Computação em Nuvem, monitoramento de recursos, séries temporais, BSP e SLA, que serão utilizados nos capítulos posteriores deste trabalho. O entendimento dos tópicos que compõem este capítulo se faz necessário para propor o modelo e, também, para a compreensão de algumas definições que aparecem no decorrer do texto.

2.1 Cloud Computing

Computação em Nuvem (ou *Cloud Computing*) possui como ideia base a disponibilização de recursos computacionais sobre a *Internet*, permitindo sua utilização em qualquer lugar e por meio dos mais diversos tipos de aplicações. Este modelo de computação possui como premissa a capacidade de acessar um grande poder computacional, criando a ideia da disponibilidade de recursos ilimitados, permitindo a elasticidade destes recursos sob demanda, eliminando, desta forma, a necessidade de uma alocação prévia quanto a sua utilização.

Segundo (MELL; GRANCE, 2011), a computação em nuvem é um modelo para permitir o acesso à rede sob demanda a um conjunto compartilhado de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicativos e serviços) que podem ser, rapidamente, provisionados e liberados com o esforço de gerenciamento mínimo ou interação com o provedor de serviços.

2.1.1 Virtualização

Com a aquisição de recursos físicos por parte de empresas para solucionar os seus problemas de tecnologia da informação, acabam por implicar custos elevados, seja no investimento feito para a obtenção destes recursos, como também na manutenção e preservação da infraestrutura atual. Desta forma, surge a necessidade de buscar recursos virtuais que substituam os equipamentos físicos com a virtualização.

A Virtualização de recursos está diretamente ligada à Computação em Nuvem, sendo este um dos motivos das empresas adotarem esta tecnologia, possibilitando, desta forma, a facilidade no provisionamento e consolidação de recursos e, assim, a redução nos custos na aquisição de novo *hardware* e consumo de energia. Na virtualização, uma máquina virtual (*Virtual Machine* - VM) pode ser descrita como uma simulação de uma máquina física, porém que não existe fisicamente e sim implementada por *software* para execução numa máquina real.

Entre os tipos de virtualização, temos a virtualização baseada em *software*, conhecida como *Virtual Machine Monitor* (VMM) ou *Hypervisor*. O *Hypervisor* é uma camada de *software* entre o sistema operacional e o *hardware* responsável pela virtualização, gerenciamento e controle

sobre os recursos físicos compartilhados pelas máquinas virtuais, como, por exemplo, memória, processador e disco. Atualmente, existem diversas opções de Hypervisores, tal como o Xen(BARHAM et al., 2003), KVM(KIVITY et al., 2007), Hyper-V, VirtualBox e VMware.

Sobre um Hypervisor, encontram-se os *middlewares* de *Cloud Computing*, tal como o Eucalyptus (NURMI et al., 2009a), Amazon EC2, OpenStack e OpenNebula, que são responsáveis por efetuar o gerenciamento e o aprovisionamento de máquinas virtuais dentro de uma infraestrutura da Nuvem.

Com a *Cloud Computing*, é possível ter, de forma rápida e transparente, a escalabilidade nos recursos computacionais, ou seja, em qualquer momento, podem ser criadas novas máquinas virtuais, baseadas em imagens de VM's já existentes, como também pode-se disponibilizar mais recursos para as máquinas virtuais que já estejam implantadas na infraestrutura de Nuvem. Em relação à migração, tem-se a possibilidade de migrar uma máquina virtual de um servidor físico para outro em qualquer momento. Assim, permitindo, também, a migração da máquina virtual com o sistema em funcionamento, técnica chamada *live migration*, reduzindo, desta forma a indisponibilidade do serviço.

Desta forma, com a Computação em Nuvem e o uso da virtualização de recursos, torna-se possível a economia de energia, além do aumento de desempenho com o acréscimo de recursos. Segundo (MELL; GRANCE, 2009), no mundo, existiam (em 2009):

- 11,8 milhões de servidores físicos em *datacenters*;
- com uma média de utilização de 15% de sua capacidade;
- O consumo médio de energia de um servidor quadruplicou entre 2001 e 2006;
- Tecnologias verdes podem reduzir os custos de energia em 50%;
- TI produz 2% das emissões de dióxido de carbono global.

2.1.2 Características

A Computação em Nuvem é composta pelas seguintes cinco características essenciais: *On-demand Self-Service, Broad Network Access, Resource Pooling, Rapid Elasticity, Measure Service*. Segundo (MELL; GRANCE, 2011), estas características podem ser descritas da seguinte forma:

- *On-Demand Self-Service* - Um consumidor pode, unilateralmente, requisitar o provisionamento de recursos automaticamente, tais como tempo de servidor e armazenamento de rede conforme a necessidade, sem qualquer interação humana com cada prestador de serviço;

- *Broad Network Access* - Os recursos e serviços estão disponíveis através da rede e acessíveis por meio de mecanismos padrões que promovem a utilização de plataformas heterogêneas (por exemplo, telefones celulares, *tablets*, *notebooks* e estações de trabalho);

- *Resource Pooling* - Os recursos computacionais do provedor são agrupados para atender múltiplos consumidores por meio de um modelo chamado *multi-tenant*, com diferentes recursos físicos e virtuais atribuídos dinamicamente e redistribuídos de acordo com a demanda do consumidor. Há um senso de independência de localização em que o cliente, geralmente, não tem controle ou conhecimento em relação à localização exata dos recursos disponibilizados, mas pode ser capaz de especificar o local em um nível maior de abstração (por exemplo, país, estado ou *datacenter*). Exemplos de recursos incluem o armazenamento, processamento, memória e largura de banda de rede;

- *Rapid Elasticity* - Os recursos podem ser elasticamente provisionados e liberados, em alguns casos, automaticamente, para escalar rapidamente, conforme a demanda. Para o consumidor, os recursos disponíveis para provisionamento frequentemente parecem ser ilimitados, e podem ser adquiridos em qualquer quantidade e a qualquer momento;

- *Measured Service* - Os serviços da nuvem são controlados e monitorados automaticamente, aproveitando uma capacidade de medição em algum nível de abstração apropriado para o tipo de serviço (por exemplo, processamento, armazenamento, largura de banda e contas de usuários ativos). A taxa de utilização de recursos pode ser monitorada, controlada e reportada, oferecendo transparência tanto para o provedor, quanto para o consumidor do serviço utilizado.

2.1.3 Modelos de Implantação

Para a implantação de uma determinada Nuvem computacional, torna-se necessário verificar previamente as necessidades da aplicação ou do serviço que será disponibilizado para o usuário, além do tipo de contrato SLA firmado entre cliente e prestador de serviço. Entre os modelos de implantação conhecidos, estão o de Nuvem Pública, Privada, Comunitária e Híbrida.

- **Nuvem Pública** - Neste modelo, a Nuvem é disponibilizada para o público em geral ou para grandes grupos industriais. A Nuvem é implementada por um prestador de serviço, que deve ser capaz de garantir o desempenho e a segurança da mesma;

- **Nuvem Privada** - Neste modelo, a infraestrutura é utilizada por apenas uma organização. Porém, pode ser administrada tanto pela própria organização, localmente ou por terceiros;

- **Nuvem Comunitária** - O modelo de Nuvem comunitária é caracterizado pelo fato da infraestrutura ser compartilhada por várias organizações para o suporte de uma comunidade específica que partilhe as mesmas preocupações como missão, requisitos de segurança,

política e considerações de conformidade. Pode ser gerenciado pelas organizações ou por terceiros e podem existir localmente ou remotamente (MELL; GRANCE, 2011);

- **Nuvem Híbrida** - No modelo de nuvem híbrida, a infraestrutura é composta por dois ou mais modelos de implementação, sendo que cada nuvem permanece como uma entidade única, mas que estão unidas pelo uso de tecnologia proprietária ou padronizada, garantindo a portabilidade de dados e aplicações (MELL; GRANCE, 2011).

2.1.4 Camadas de Serviço

Dentro dos modelos de implantação, têm-se diferentes níveis de camadas de serviço, entre elas, podem ser citadas: SaaS (Software como serviço), PaaS (Plataforma como serviço) e o IaaS (Infraestrutura como serviço).

- A camada SaaS (*Software as a Service* / Software como Serviço) possui como principal característica, fornecer aplicações para usuários, que possam ser acessíveis em qualquer momento e em qualquer lugar por meio da *Internet*. Entre os produtos que se encaixam nesta categoria, estão aplicações e serviços dispostos na Web 2.0. Pode-se citar como exemplos de aplicações o Google Docs, Google Mail (GMail), entre outros;

- A camada PaaS (*Platform as a Service* / Plataforma como Serviço) possui como uma das principais características dispor uma plataforma para o desenvolvedor criar e implementar aplicações na nuvem sem a necessidade da preocupação em relação ao *hardware* que será executado. Entre os produtos que se enquadram nesta categoria, estão API's (*Application Programming Interface*) e *frameworks* para programação. Pode-se citar como exemplo o Google App Engine (GAE), Microsoft Azure, entre outros;

- A camada IaaS (*Infrastructure as a Service* / Infraestrutura como Serviço) possui como uma das principais características, o fornecimento de uma infraestrutura de hardware e serviços sob demanda para o consumidor final. Entre os principais recursos computacionais que se enquadram nesta categoria, estão a CPU, unidade de armazenamento, memória RAM, largura de banda, entre outros. Como infraestrutura, pode-se citar o Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3), Eucalyptus, OpenStack, OpenNebula, entre outros.

Em um nível mais alto de abstração, existem os *middlewares* para a federação de Nuvens. Nesta abordagem, uma única entidade serve como um portal para diferentes soluções IaaS independentes, resolvendo, assim, a interoperabilidade e aumentando a quantidade de recursos. Segundo (ELMROTH et al., 2009), a federação de Nuvens é uma alternativa rentável para o dimensionamento sobre a quantidade de servidores a fim de lidar com determinados picos de carga, como, por exemplo, utilizando recursos extras de outros provedores durante momentos de

maior carga. Da mesma forma, recursos subutilizados podem ser disponibilizados para outros locais durante períodos de menor carga como uma fonte de renda extra.

2.2 Monitoramento

A etapa de monitoramento refere-se à coleta e análise regular de informações de desempenho sobre os estados de um determinado objeto. Com o uso de ferramentas e sistemas de monitoramento, torna-se possível a obtenção de informações dos estados dos recursos presentes na infraestrutura, sendo eles: máquinas físicas, máquinas virtuais, aplicações e seus respectivos processos.

2.2.1 Monitoramento em Cloud Computing

De acordo com (ACETO et al., 2012), o monitoramento de Computação em Nuvem é uma tarefa de suma importância para os provedores e consumidores de serviços em *Cloud*. Por um lado, é uma ferramenta fundamental para controlar e gerenciar infraestrutura de *hardware* e *software*, e, por outro lado, fornece informações e Indicadores-Chave de Desempenho (KPI) para ambas as plataformas e aplicações.

Em *Cloud Computing*, o monitoramento de recursos é imprescindível para manter a elasticidade necessária e cumprir os acordos de níveis de serviço (SLA - *Service-Level Agreement*), são requisitos essenciais para uma correta disponibilização e utilização dos recursos, visto que, dependendo do cenário ou situação, seja necessário provisionar e/ou consolidar os recursos da Nuvem, aumentando ou diminuindo conforme a demanda.

O monitoramento contínuo dos recursos presentes na infraestrutura de Nuvem e seus respectivos SLAs fornecem aos prestadores e consumidores de serviço informações da qualidade do serviço (QoS) oferecido visando o uso eficiente da infraestrutura. Desta forma, tornando possível identificar gargalos, falhas no contrato SLA e desperdício de recursos computacionais, melhorando, assim, as decisões a serem tomadas sobre o provisionamento e a consolidação de recursos.

A necessidade em gerenciar ou monitorar os recursos dispostos na nuvem está diretamente associada no fato de garantir a Qualidade de Serviço (*Quality of Service* - QoS), acordos de SLA (*Service Level Agreement*), alta disponibilidade e segurança. De posse de mecanismos adequados para medição, torna-se possível o uso de diferentes estratégias para a alocação de recursos.

Segundo (ACETO et al., 2012), Computação em Nuvem envolve muitas atividades para as quais o monitoramento é uma atividade essencial, tais como:

- **Planejamento de Recursos**: Um dos maiores desafios para prestadores e consumidores de serviços na Nuvem. De posse de métricas de desempenho, torna-se possível efetuar um planejamento da capacidade dos recursos;

- **Gerenciamento de Recursos**: Visando gerenciar ambientes complexos como a Nuvem, é essencial sistemas de monitoramento que capturem precisamente os estados dos recursos presentes na infraestrutura;

- **Gerenciamento de Datacenter**: Serviços de Nuvem são fornecidos *datacenters*, cujo gerenciamento destes é uma atividade importante. De acordo com (WANG et al., 2011), monitoramento e análise de dados são dois elementos fundamentais para o gerenciamento de *datacenters*. A tarefa de monitoramento registra as métricas de hardware e software desejadas, enquanto que a análise avalia estas métricas para identificar sistemas ou aplicações para solução de problemas, provisionamento de recursos ou outros problemas de gerenciamento;

- **Gerenciamento de SLA**: Com a flexibilidade em termos de gestão de recursos fornecidos pela Computação em Nuvem, tornam-se necessários contratos os quais ficam acordados os respectivos níveis de atendimento do serviço prestado;

- **Gerenciamento de Desempenho**: Devido ao compartilhamento dos recursos, alguns nós podem possuir um desempenho inferior a outros existentes na infraestrutura da Nuvem;

- **Faturamento**: Permite ao consumidor verificar efetivamente o uso dos serviços prestados pelo provedor, além de pagar proporcionalmente os recursos que, de fato, foram utilizados;

- **Solução de Problemas**: Devido à complexidade das infraestruturas de Nuvem, o monitoramento torna-se necessário visando localizar a origem e as causas do problema.

2.2.2 Recurso

O monitoramento no nível de recursos tem como objetivo obter informações das métricas das entidades gerenciadas presentes na infraestrutura. Entre os principais níveis dos recursos de arquitetura da Nuvem, tem-se:

- Infraestrutura: *Infrastructure as a Service* (IaaS), fornecendo infraestrutura de serviços numa plataforma de virtualização. Neste nível, é considerado todos os recursos presente dentro de uma infraestrutura de *Cloud Computing*.

- Plataforma: *Platform as a Service* (PaaS), consiste na entrega de uma plataforma e/ ou pilha de serviços consumidos pela infraestrutura da *Cloud Computing*. Neste nível, consideram-se os recursos como as aplicações/serviços disponíveis dentro da Nuvem.

- Aplicação: *Software as a Service* (SaaS), fornecendo *software* como serviço através da *Internet*, onde o fornecedor do serviço é responsável por manter toda infraestrutura para seu respectivo uso. Neste nível, é mensurado o quanto a aplicação consome dentro da

Nuvem, tal como métricas de desempenho, como por exemplo, E/S, consumo de CPU e memória RAM.

Entre as métricas monitoradas, tem-se a CPU, memória RAM, disco de armazenamento e a largura de banda. Dentro deste cenário, tem-se a representação do poder de processamento da CPU dado por meio do número de instruções executadas por segundo e seu respectivo percentual de uso. Disco de armazenamento com dados de leitura/escrita e sua taxa de utilização, além da memória RAM, com informações sobre sua taxa de utilização. Por fim, tem-se a largura de banda, com informações sobre o número de *bytes* enviados e recebidos por segundo.

De posse das métricas de monitoramento coletadas, estas poderão ser utilizadas por administradores no âmbito de resolução de problemas e planejamento da infraestrutura, como também para o desenvolvedor, visando buscar a avaliação comportamental e depuração de suas respectivas aplicações. O monitoramento de uma determinada aplicação torna-se necessário devido à crescente carga de trabalho a qual são submetidas às aplicações. Em ambientes distribuídos, como *Cloud Computing*, a aplicação é distribuída para computação no nível de processo aos demais recursos computacionais (tal como máquina física e/ou virtual) presente na infraestrutura. Desta forma, o monitoramento no nível de aplicação pode ser utilizado para efetuar um melhor controle sobre desempenho conforme aos recursos disponíveis.

Cloud Computing elimina a grande carga econômica de instalação de recursos e custos operacionais, tornando-se uma escolha preferível na computação de alto desempenho (*High Performance Computing* - HPC), provendo facilidades para os cientistas que possuem a necessidade de validar suas ideias por meio da utilização de recursos em grande escala. (ZHAO; LI, 2012)

Com o monitoramento no nível de processo, torna-se possível um melhor controle sobre a utilização dos recursos, inclusive em situações nas quais um processo possui uma alta taxa de comunicação com outro distante geograficamente, podendo, neste caso, efetuar a migração deste para uma máquina mais próxima, diminuindo assim a latência na comunicação entre eles.

Também pode ser interessante o monitoramento no nível de aplicação em situações nas quais um processo específico pode estar sobrecarregando a máquina hospedeira, afetando os demais recursos, inclusive diminuindo o tempo de processamento em casos nos quais outros processos espalhados pela Nuvem estejam esperando uma resposta para dar continuidade à computação.

De posse de informações de monitoramento de uma determinada aplicação, torna-se possível verificar quais processos exigem mais recursos de comunicação e computação. Em situações de comunicação, pode-se migrar para máquinas com menor distância / latência para máquinas com maior vazão. Em casos de computação, pode-se migrar os processos para máquinas com maior poder de processamento e recursos disponíveis.

2.2.3 Padrões

Em um ambiente dinâmico, como em Computação em Nuvem, torna-se necessária a análise histórica das métricas coletadas, visando à identificação de padrões de uso que possam auxiliar

no gerenciamento dos recursos e detecção de anomalias na infraestrutura.

Esta etapa de análise histórica visa à busca de padrões que possam ocorrer durante o decorrer de uma janela de tempo. Uma das técnicas existentes envolve o uso de séries temporais. Uma série temporal é uma sequência de números reais, em que cada número representa um valor de dados em um ponto no tempo(WU; AGRAWAL; EL ABBADI, 2000). Para a análise sobre a sequência dos pontos, podem ser utilizadas técnicas de *pattern matching*, tais como DFT (*Discrete Fourier Transform*) e DTW (*Dynamic Time Warping*). Como medida de similaridade entre as amostras, ambas as abordagens assumem distância euclidiana(WU; AGRAWAL; EL ABBADI, 2000).

Com a análise de séries temporais, é possível encontrar padrões que possam determinar situações que se repetem ao longo do tempo. Um padrão de utilização pode ser caracterizado por uma ação cíclica por parte de um recurso (seja este aplicação, máquina física ou virtual), no qual pode interferir no uso futuro dos recursos. A verificação dele passa por um estudo de similaridades. Elas estão baseadas no fato de que o estado de utilização atual do recurso possui uma grande probabilidade de já ter ocorrido no passado. A análise pode envolver predições que envolvem um grande volume de dados, tornando-a, muitas vezes, custosa.

Aplicações de alto desempenho (HPC), por exemplo, do tipo BSP (*Bulk Synchronous Parallel*), são aplicações que possuem N processos com uma série de superetapas (computação, comunicação e sincronização). Um dos benefícios da análise de padrões neste caso, é que através da descoberta dos padrões de comunicação e computação, têm-se meios para decidir qual será o processo candidato para migração.

Desta forma, visando tratar de forma eficiente este problema e, assim, não tornar o sistema de monitorando um possível gargalo, consumindo recursos preciosos da nuvem, torna-se necessária a utilização de técnicas de mineração de dados e sistemas distribuídos para análise de grande massa de dados. Entre eles, pode-se citar o Apache Hadoop, uma plataforma escalável que permite verificar e processar grandes volumes de dados distribuídos entre diferentes nós.

Tarefas de mineração de dados temporais podem ser agrupadas da seguinte forma: (i) a previsão, (ii) classificação, (iii) agrupamento, (iv) busca e recuperação e (v) a descoberta de padrões. Das cinco categorias listadas acima, as quatro primeiras foram investigadas extensivamente na análise tradicional de séries temporais e no reconhecimento de padrões. A descoberta de padrões inclui a identificação de um padrão ou padrões frequentes, sendo este último um padrão periódico que possui regularidades em séries temporais (SRIDEVI; RAJARAM; SWADHIKAR, 2010).

2.3 Service Level Agreement

Um acordo de nível de serviço (*Service Level Agreement* - SLA) é um contrato entre um cliente e um determinado fornecedor de serviço. Neste contrato, são definidos quais serviços serão prestados e seus respectivos níveis de atendimento pelo fornecedor.

Os SLA's, de uma maneira geral, são acordos contratuais entre o prestador do serviço e o usuário, descrevendo o serviço, as taxas, as multas, os bônus, os termos e as condições do contrato. Métricas de desempenho específicas que são usadas para verificar as condições de prestação de serviço e são chamadas de objetivos de nível de serviço (*Service Level Objective* - SLO). (YAZIR et al., 2012)

A negociação e efetivação do SLA envolvem a seleção de diversos prestadores de serviço com base em suas respectivas ofertas de qualidade, de modo a estabelecer em contrato estas características para o serviço a ser prestado. Cada SLA possui métricas que devem ser cumpridas pelo prestador de serviço. Definir quais serão estas métricas torna-se uma questão importante para um melhor controle e correto funcionamento do serviço. Para cada métrica selecionada, deve-se definir quais são os intervalos para coleta pelo serviço de monitoramento. Além disso, deve ser possível determinar se a métrica foi corretamente coletada junto aos recursos presentes na infraestrutura.

Em Computação em Nuvem, devido ao compartilhamento de recursos entre diversos clientes, torna-se necessária a utilização de sistemas que possam efetuar a medição e avaliação do contrato e os níveis de serviço disponibilizados e atendidos ao cliente. Esta avaliação é feita com o uso de sistemas de monitoramento, que efetuam constantemente verificações dos níveis de atendimento atuais dos recursos sob a infraestrutura de Nuvem contratada.

De acordo com (WUHIB; STADLER; CLEMM, 2006), uma das ferramentas-chave para o monitoramento do nível de serviço é o TCA (*Threshold Crossing Alerts*), responsável por validar e garantir que os níveis de serviço são, de fato, respeitados, notificando o provedor de serviços quando um determinado parâmetro ultrapassar um determinado valor limite e, por consequência, enviando para áreas em que uma ação preventiva pode ser tomada. Para a definição, requisitos e modelagem de um SLA, existem diferentes linguagens e formalismos, tais como: SLAng (LAMANNA; SKENE; EMMERICH, 2003), WSLA(KELLER; LUDWIG, 2003), RBSLA (PASCHKE, 2005), entre outros.

2.4 Séries Temporais

Uma série temporal pode ser definida como um conjunto de observações quantitativas dispostas em ordem cronológica. Em geral, assumindo que o tempo é uma variável discreta (KIRCHGÄSSNER; WOLTERS; HASSLER, 2012), ou seja, possui características mensuráveis que podem assumir um número finito ou infinito de valores inteiros contáveis. Conforme (WEI, 1994), a natureza intrínseca das séries temporais está relacionada as observações serem dependentes ou correlacionadas, respeitando a ordem cronológica dos dados.

As séries temporais podem ser classificadas em dois tipos: Contínuas e Discretas. As Séries Temporais contínuas são registradas continuamente ao longo do tempo, enquanto as Séries temporais discretas são obtidas em intervalos de tempo específicos (WEI, 1994). Os resultados das observações das variáveis durante um determinado período de tempo são feitas em intervalos

regulares, como, por exemplo: diariamente, mensalmente, anualmente, entre outros. De posse destas observações, torna-se possível o uso de técnicas para análise de previsão e tendências de séries temporais.

A predição utilizando séries temporais envolve uma grande variedade de técnicas e métodos. Um horizonte de previsão pode ser curto, médio ou longo. Um fator que deve ser considerado ao se efetuar previsões utilizando séries temporais: as variáveis cujos valores estão dispostos na série podem ser afetadas por diversos fatores, classificadas como: tendência, ciclo, sazonalidade e ruído aleatório ou erro.

- *Tendência*: reflete o sentido de deslocamento da série ao longo do tempo;
- *Ciclo*: movimento com variações de tendência que tende a ser periódico;
- *Sazonalidade*: movimento com variações regulares na série;
- *Ruído aleatório ou erro*: compreende a variabilidade dos dados e não pode ser modelado, captando os efeitos que não podem ser descritos nos outros métodos.

A análise de tendência utiliza os dados históricos para efetuar previsões futuras. Porém, em algumas situações, pode não ser possível explicar as oscilações de uma determinada variável ao longo de uma janela de tempo, pois seus movimentos podem estar associados, em determinados casos, a outras variáveis que afetam a variável analisada.

Os métodos para previsão de séries temporais podem ser classificados como qualitativos e quantitativos. A abordagem qualitativa é utilizada quando os dados históricos não estão disponíveis, desta forma, envolve a ajuda e o julgamento de especialistas para o desenvolvimento das previsões das respectivas séries. Por outro lado, os métodos quantitativos são, geralmente, utilizados quando as informações históricas das variáveis a serem previstas estão disponíveis.

2.4.1 Metodos de Prediçao

Existem dois tipos de métodos em séries temporais: Univariados e Multivariados. No método univariado, a predição é baseada em apenas uma única série de dados. Já no método multivariado, a predição é composta por mais de uma série, porém, sem qualquer causalidade entre as séries.

2.4.1.1 Naive

Naive, também conhecido como modelo ingênuo, é um método de previsão de séries temporais que utiliza o último valor da série como sendo o valor futuro.

2.4.1.2 Médias Móveis

Médias móveis (do Inglês *Moving Average* - MA) é um método de previsão de séries temporais que possui como principal objetivo fornecer um valor médio das amostras da série dentro de um determinado período. Uma de suas vantagens é a de ser mais simples que os demais e de fácil uso. É utilizado para previsões por meio de dados históricos. Neste modelo, quanto mais dados forem incluídos, maior será a suavização dos pontos previstos.

- **Média Móvel Simples**: Também conhecida como média aritmética, é feita por meio do somatório de um determinado conjunto de valores, dividindo-os pela quantidade de elementos do conjunto em um período de tempo. Neste modelo, novas amostras são incluídas na janela de previsão, enquanto as antigas são removidas na mesma quantidade. A fórmula 2.1 representa uma equação de uma média móvel simples. As variáveis V representam as amostras, enquanto a variável N representa a janela de tempo.

$$MMS = \frac{V1 + V2 + V3... + VN}{N} \tag{2.1}$$

- **Média Móvel Ponderada**: É uma extensão da média móvel simples. Possui como principal característica poder dar pesos maiores para valores mais recentes. Neste modelo, os valores antigos não são descartados diretamente como no modelo de média móvel simples. Estes valores, mesmo fora da janela de tempo, continuam mantendo uma participação, porém vão diminuindo com o tempo.

2.4.1.3 Holt-Winters

O método de Holt-Winters é uma técnica popular de suavização exponencial utilizada para fornecer previsões de curto prazo a partir de determinada série temporal(MAKRIDAKIS; WHEELWRIGHT; HYNDMAN, 2008). Este método possui dois modelos, o aditivo e o multiplicativo. A seleção do modelo ideal dependerá da série temporal considerada.

- **Holt-Winters Aditivo**: No modelo aditivo, as observações são consideradas atribuindo a soma da tendência, do efeito sazonal e do ruído aleatório;

- **Holt-Winters Multiplicativo**: É um modelo de suavização exponencial para séries temporais com tendência linear. Um padrão multiplicativo é um modelo no qual a variação sazonal pode ser explicada por meio de uma constante sazonal.

2.4.1.4 ARIMA

O modelo ARIMA (*Autoregressive Integrated Moving Average*) é um modelo autorregressivo integrado de médias móveis para previsão de séries temporais, também conhecido pelo

nome de modelo Box-Jenkins. É um dos métodos quantitativos para previsão de séries temporais mais difundidos, descrita pelos autores George Box e Gwilym Jenkins.

Este modelo baseia-se na ideia de que séries temporais não-estacionárias, ou seja, não crescem ao longo do tempo, podendo ser modeladas a partir de diferenciações, além da inclusão de componentes autoregressivos integradores e médias móveis.

2.4.1.5 Avaliação de Previsões

O uso de um determinado modelo nem sempre produz boas previsões, em algumas situações, os valores tendem a apresentar resultados estimados muito distantes dos reais observados na série temporal. Desta forma, para efetuar a seleção de um modelo de previsão de séries temporais adequado, é necessário mensurar os erros de previsão.

- **Erro Médio** (ME - Mean Error): Consiste na média aritmética de todos erros de previsão;
- **Erro Médio Absoluto** (MAE - Mean Absolute Error): É a média de erros em um conjunto de previsões, verificando os valores absolutos das diferenças entre a previsão e a observação;
- **Erro Quadrático Médio** (MSE - Mean Squared Error): É a média aritmética da soma dos quadrados dos erros de previsão;
- **Erro Percentual Absoluto Médio** (MAPE - Mean Absolute Percentage Error): Consiste na média de erros percentuais de previsão.

2.5 Bulk Synchronous Parallel

O BSP (*Bulk Synchronous Parallel*)(VALIANT, 1990) é um modelo de programação para computação paralela. Este modelo é composto uma sequência de superetapas globais. Cada superetapa é composta com as seguintes fases: computação, comunicação e barreira de sincronização.

- **Computação**: Nesta fase, também conhecida como computação local, cada processador efetua seus cálculos concorrentemente e independentemente com os demais, cada qual utilizando apenas os valores de suas memórias locais;
- **Comunicação**: Nesta fase, após a computação local, os processos trocam informações por meio da comunicação global entre os processadores participantes da computação;
- **Barreira de Sincronização**: Cada processo que terminar sua fase de comunicação fica esperando numa barreira de sincronização por todos os demais processos que ainda não finalizaram a fase de comunicação. Uma próxima superetapa somente iniciará quando

todos os processos estiverem completados suas respectivas comunicações. Esta fase é responsável por finalizar uma superetapa.

Cada uma das fases presentes dentro de uma superetapa possui um custo, sendo elas: custo de computação, comunicação e da barreira de sincronização. O custo individual de uma superetapa é baseado no somatório dos custos individuais de cada fase, representada pela equação 2.2.

$$CustoTotal = \sum CustoSuperEtapa \tag{2.2}$$

Já o custo de uma superetapa individual é representado pela soma do custo de computação, custo de comunicação e custo da barreira de sincronização, demonstrado pela equação 2.3.

$$CustoSuperetapa = CustoComp + CustoComm + CustoBarreira \tag{2.3}$$

2.6 Considerações sobre o Capítulo

Neste capítulo, foi apresentada a contextualização dos principais tópicos para uma melhor compreensão da pesquisa e conclusões deste trabalho. Além de *Cloud Computing*, suas principais características, monitoramento de recursos, acordos de SLA, métodos de previsão baseados em séries temporais, e, por fim, é apresentado o modelo BSP.

Atualmente, o monitoramento de recursos em infraestrutura de *Cloud Computing* é feito através de soluções próprias dos *middlewares* IaaS, além de ferramentas de uso geral, ou seja, que também são utilizadas em *Clusters* e/ou *Grids*. No contexto de prover uma melhor gestão sobre os recursos disponíveis, o desenvolvimento de soluções que visam efetuar o monitoramento em sistemas de *Cloud Computing*, trabalhando em conjunto com o SLA, torna-se um aspecto importante para uma melhor gestão da infraestrutura.

No sentido de efetuar um melhor controle sobre os recursos dispostos na infraestrutura da Nuvem, tem-se a utilização de técnicas para predição de desempenho, como por exemplo, previsão de séries temporais, que é um dos desafios dentro da área de mineração de dados, onde os valores futuros são previstos em função dos valores passados. A importância do uso de séries temporais se dá pela relação de dependência entre as observações ordenadas no tempo.

3 TRABALHOS RELACIONADOS

Este capítulo tem como objetivo apresentar alguns dos principais projetos relacionados a *Cloud Computing* e ao objeto de pesquisa deste trabalho, o monitoramento de recursos preditivo para aplicações paralelas em *Cloud Computing*. Os trabalhos relacionados foram selecionados a partir dos tópicos de *middlewares* de Computação em Nuvem que possuem soluções próprias e sistemas de monitoramento de uso geral. Este levantamento prioriza a descrição das características de monitoramento dos sistemas selecionados e a apresentação das lacunas existentes. Por fim, é realizado um comparativo entre as principais características que cada trabalho.

3.1 Cloud Computing

3.1.1 PCMONS

O PCMONS (Private Cloud MONitoring System) é um sistema de monitoramento *Open-Source* para nuvens privadas, que abstrai a heterogeneidade da nuvem por meio da camada de integração, permitindo, dessa forma, um monitoramento uniforme de diferentes plataformas e tecnologias de virtualização (DE CHAVES; URIARTE; WESTPHALL, 2011). O PCMONS é composto pelos seguintes módulos:

- **Node Information Gatherer**: Este módulo é responsável por coletar informações de desempenho locais da máquina na qual se encontra instalado. A versão atual captura informações relacionadas às VMs locais e as envia para agregação de dados;
- **Cluster Data Integrator**: Neste módulo, existe um agente que coleta e prepara os dados capturados para a camada seguinte;
- **Monitoring Data Integrator**: Este módulo é responsável por coletar e persistir os dados de monitoramento em base de dados;
- **VM Monitor**: Este módulo é responsável por injetar *scripts* nas VMs para a coleta de informações de desempenho, como carga de CPU e memória RAM;
- **Configuration Generator**: Recupera as informações de desempenho da base de dados para a configuração e visualização;
- **Monitoring Tool Server**: Responsável por receber informações de monitoramento de diferentes recursos;
- **User Interface**: Utiliza a interface do Nagios para apresentação dos dados.

O PCMONS foi desenvolvimento sob o IaaS Eucalyptus e utiliza os recursos do Nagios para efetuar o monitoramento dos recursos da infraestrutura da Nuvem.

3.1.2 DARGOS

O DARGOS é uma arquitetura distribuída de monitoramento na Nuvem utilizando um modelo híbrido *Pull/Push* para disseminar informações de monitoramento dos recursos (POVEDANO-MOLINA et al., 2013). Em adição, foi desenvolvido para prover de forma flexível e extensível a adição de novas métricas no sistema. A arquitetura principal do DARGOS é composta com os seguintes componentes:

- **Node Supervisor Agent (NSA)**: O NSA se inscreve para receber informações de desempenho dos componentes NMA distribuídos na infraestrutura;

- **Node Monitoring Agent (NMA)**: Este componente é responsável por coletar estatísticas de desempenho dos recursos (CPU, memória, entre outros) em uma determinada máquina e enviar para o componente NSA.

Conforme (POVEDANO-MOLINA et al., 2013), o sistema de monitoramento DARGOS possui como principal característica ser adaptativo, escalável e com baixa intrusividade.

3.1.3 AWS - Amazon Web Services

A Amazon, uma empresa multinacional de comércio eletrônico, possui como plataforma de Computação em Nuvem, o Amazon Web Services (AWS). Na parte central desta plataforma, encontra-se o Amazon EC2 (*Elastic Compute Cloud*). Neste ambiente, estão disponíveis imagens pré-configuradas para as máquinas virtuais, chamadas *Amazon Machine Image* (AMI). Para utilização de um serviço sob esta plataforma, pode-se usar estas imagens ou também criar uma nova Amazon AMI com o ambiente necessário para a execução de uma determinada aplicação e/ou serviço.

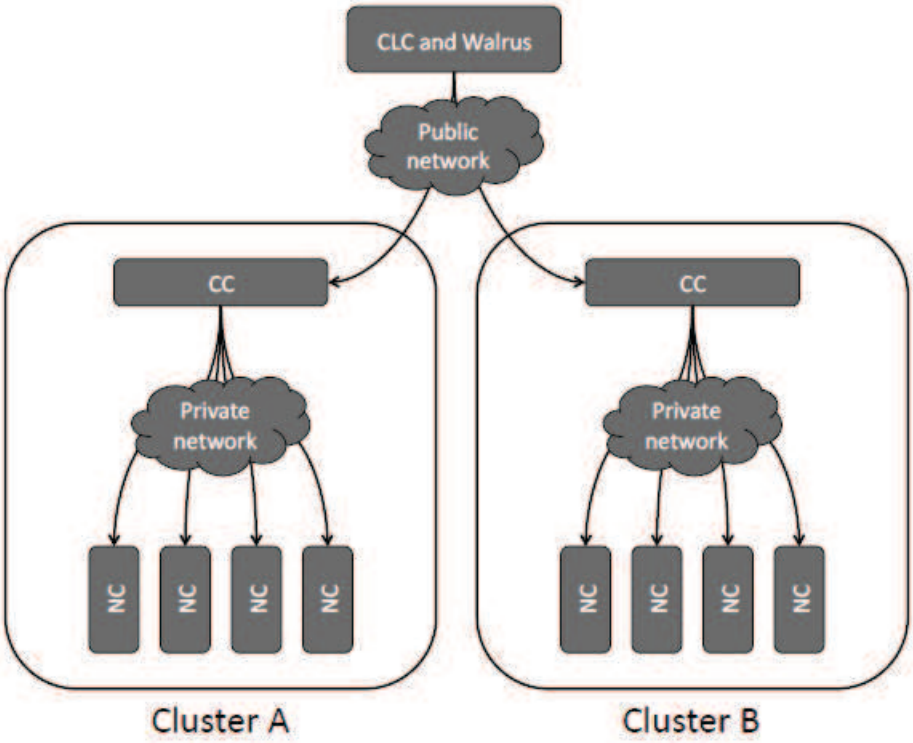
Entre as funcionalidades existentes de liberação de instâncias, existem as seguintes: Instanciação *On-Demand*, em que os recursos são adquiridos e instanciados conforme a demanda computacional e Instância Spot, que permite que sejam negociados os recursos não utilizados pelo Amazon EC2.

Os recursos disponíveis no Amazon EC2, são os seguintes: *Amazon Elastic Block Store* (Amazon EBS) - um serviço de armazenamento, o Amazon CloudWatch - um sistema de monitoramento de recursos (AMAZON, 2014) e aplicativos AWS (*Amazon Web Services*). Este sistema fornece informações e métricas de desempenho dos recursos, possui a capacidade de correlacionar comportamentos e descobrir padrões, além de ser possível efetuar a definição de alarmes. Por fim, o Amazon Auto Scaling, que permite o provisionamento e consolidação dos recursos por meio do monitoramento feito com o *Amazon CloudWatch*.

3.1.4 Eucalyptus

O Eucalyptus é uma plataforma *Open-Source* de Computação em Nuvem, que implementa o modelo comumente referido como Infraestrutura como Serviço (IaaS). (NURMI et al., 2009b). A Figura 1 demonstra a visão geral da arquitetura do Eucalyptus, com os componentes: *Node Controller* (NC), *Cluster Controller* (CC), *Storage Controller* (Walrus) e Cloud Controller (CLC).

Figura 1: Visão geral do Eucalyptus



Fonte: (NURMI et al., 2009a)

Conforme (NURMI et al., 2009b), o Eucalyptus é separado hierarquicamente pelos seguintes componentes:

- *Node Controller* - Responsável por controlar a execução, inspeção e o encerramento das instâncias de máquinas virtuais na máquina física na qual é executada;
- *Cluster Controller* - Responsável por reunir informações e agendamento da execução da máquina virtual em *Node Controller's* específicos, bem como a gerência no fluxo de requisições;
- *Storage Controller* (Walrus) - Responsável pelo serviço de armazenamento. Este serviço implementa a interface do Amazon S3 (*Simple Storage Service*), proporcionando um mecanismo para armazenar e acessar imagens de máquinas virtuais e dados do usuário;

- *Cloud Controller* - É o ponto de entrada da nuvem para usuários e administradores. É responsável por consultar gerentes do nó para obter informações sobre recursos, efetuar decisões de agendamento, efetuando pedidos aos *Cluster Controller's*.

Em relação ao monitoramento, o Eucalyptus, apesar de ser *middleware* bastante utilizado e reconhecido, não possui um módulo próprio para o monitoramento de recursos, seja ele máquina física, virtual ou aplicativo (processo). Porém, como forma de atender parte desta lacuna de monitoramento, o Eucalyptus disponibiliza a integração via *Scripts* com o Nagios e o Ganglia para monitorar seus recursos. O Nagios é responsável pela monitoração do *status* dos componentes e das máquinas virtuais e o Ganglia, pelo monitoramento de recursos sobre a infraestrutura de Nuvem projetada.

3.1.5 OpenNebula

O OpenNebula é um *middleware* IaaS *Open-Source* para Computação em Nuvem, que efetua o gerenciamento distribuído e dinâmico de infraestruturas virtuais, permitindo a instanciação e realocação de máquinas virtuais em um conjunto de máquinas físicas.

Entre os subsistemas disponíveis, tem-se o módulo de monitoramento. Este módulo possui diversos sensores, cada qual responsável por monitorar diferentes métricas de desempenho das máquinas físicas e virtuais presentes na infraestrutura da Nuvem, tais como CPU, memória RAM, armazenamento e rede (TORALDO, 2012). Além das informações de máquinas físicas, este módulo possui também sensores para o monitoramento de *hipervisores*, como, por exemplo, o KVM, XEN e VMWare.

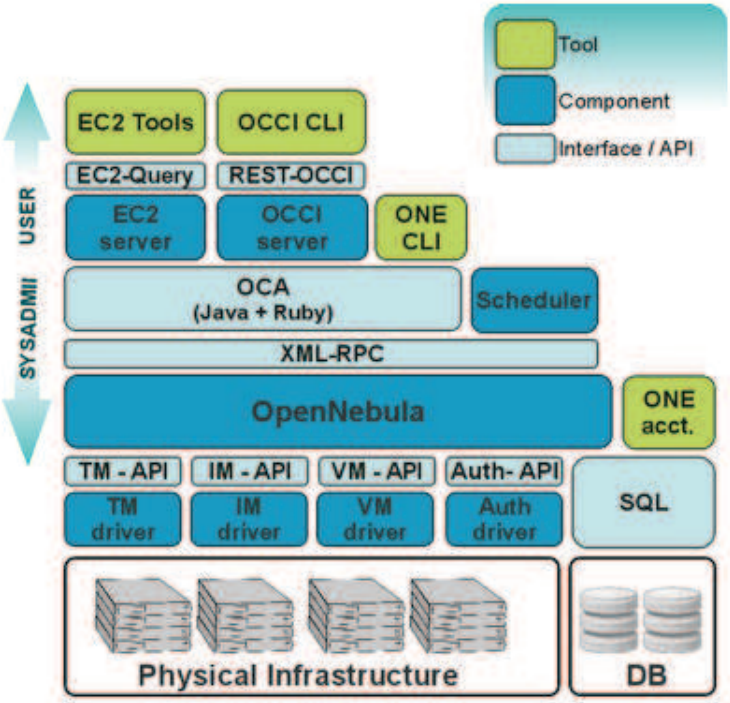
O OpenNebula também dispõe de diversas formas de integração e manipulação dos recursos presentes em sua infraestrutura, entre elas, destaca-se o XML-RPC, um protocolo para chamada de procedimento remoto (RPC – *Remote Procedure Call*), que utiliza a linguagem de marcação XML (*eXtensible Markup Language*) para codificação das chamadas de procedimento e o HTTP (*Hypertext Transfer Protocol*) como protocolo de transporte.

As configurações do intervalo de tempo do monitoramento de uma máquina física ou virtual podem ser definidas no *daemon* do OpenNebula. Atualmente, o tempo padrão deste período é definido como 10 minutos para máquinas físicas e virtuais.

Além disso, o OpenNebula provê um módulo de contabilidade da infraestrutura, chamado de OpenNebula Watch, um subsistema que fornece informações estatísticas relacionadas à utilização dos recursos da Nuvem. Estas informações coletadas a partir dos recursos gerenciados são armazenadas em intervalos de tempo predefinidos. A Figura 2 apresenta as interfaces disponibilizadas pelo OpenNebula.

O monitoramento de recursos presente no OpenNebula pode ser visualizado por meio da ferramenta de administração gráfica (OpenNebula Sunstone), visando ao gerenciamento da infraestrutura da Nuvem. Este sistema apresenta informações sobre consumo de recursos, como, por exemplo, CPU e memória. Porém, não efetua o monitoramento da aplicação/processo, do

Figura 2: Interfaces do OpenNebula



Fonte: (OPENNEBULA, 2014)

consumo real de que cada máquina virtual, além de não trabalhar com a descoberta de padrões de uso.

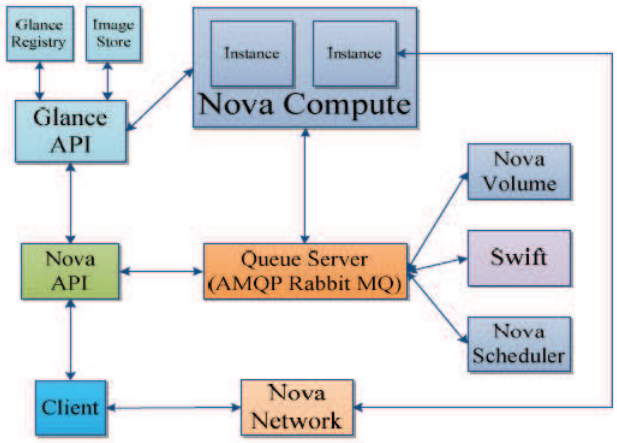
3.1.6 OpenStack

O OpenStack é uma plataforma de *Cloud Computing* do tipo *Infrastructure as a Service* (IaaS) para a construção de nuvens públicas e privadas. Este projeto nasceu da parceria entre a NASA (*National Aeronautics and Space Administration*) e a Rackspace (SEFRAOUI; AISSAOUI; ELEULDJ, 2012).

É composto pelos seguintes projetos: OpenStack Compute, responsável por provisionar e gerenciar grandes redes de máquinas virtuais, criando, desta forma, uma redundante e escalável plataforma de *Cloud Computing*. E o OpenStack Object Store, que fornece redundância e armazenamento escalável utilizando *Clusters*. A Figura 3 demonstra a visão geral da arquitetura do OpenStack e seus respectivos componentes.

Em relação ao monitoramento presente no *middleware* OpenStack, esse possui o módulo chamado Swift-recon, que retorna informações básicas do sistema operacional que está em execução na Nuvem. Outro módulo, chamado de Ceilometer (CASTILLO; MALLICHAN; AL-HAZMI, 2013), possui como proposta a coleta eficiente de dados de medição, em termos de custos na utilização de CPU e rede, permitindo, desta forma, a notificação das informações e, também, prover um sistema para a coleta de informações relacionadas ao uso dos recursos e,

Figura 3: Visão geral do OpenStack



Fonte: (OPENSTACK, 2014)

assim, poder efetuar a devida cobrança.

3.2 Monitoramento

3.2.1 Ganglia

O Ganglia é um projeto *Open-Source* que surgiu na Universidade da Califórnia. É uma ferramenta de monitoramento distribuído para computação de alto desempenho, utilizado na área de sistemas baseados em *Grid Computing* ou *Clusters* (MASSIE; CHUN; CULLER, 2004). Ele adota uma estrutura hierárquica; comunicação de estrutura de árvore entre os seus componentes, a fim de acomodar informações de grandes coleções de Grades ou *Clusters*. As informações coletadas pelo monitor do Ganglia incluem métricas de *hardware* e sistema, tais como tipo do processador, carga de CPU, uso de memória, uso de disco, utilização de rede, entre outras informações estáticas / dinâmicas do sistema (YANG; CHEN; CHEN, 2007).

Entre seus componentes, tem-se o Gmond (*Ganglia Monitoring Daemon*, um *daemon* multitarefa que executa o monitoramento sob cada nó presente na infraestrutura a ser gerenciada, coletando as informações locais e as transmitindo a todos os computadores na infraestrutura monitorada. E o Gmetad (*Ganglia Meta Daemon*, um *daemon* responsável pela coleta de informações a partir de dados de múltiplas fontes de Gmond e/ou Gmetad.

Existe, também, o Gmetric, que possibilita a inserção de métricas próprias para o sistema de monitoramento, além das já definidas por padrão no Gmond.

3.2.2 Nagios

O Nagios é um sistema de monitoramento *Open-Source*. É flexível e configurável. As principais tarefas do Nagios são para monitorar o *status* dos dispositivos de rede e seus serviços, além de notificar os administradores de sistema quando problemas forem encontrados. O núcleo do Nagios é um mecanismo agendador que efetua regularmente a verificação da rede especificada, bem como seus serviços (ISSARIYAPAT et al., 2012).

Além de efetuar o monitoramento dos recursos da infraestrutura de rede, também pode definir ações que possam ser executadas em situações predefinidas ou para a resolução proativa de problemas.

Entre as métricas que o Nagios pode monitorar, estão o uso de CPU, memória, disco, entre outras informações, que podem ser coletadas por meio de utilitários do próprio sistema operacional. É possível, também, desenvolver métricas customizadas para utilização do mecanismo de monitoramento do Nagios.

Para o monitoramento e verificações de padrões de uso, o Nagios possui o *plugin* checklogs, responsável por efetuar a verificação dos arquivos de *logs* do sistema em busca de algum determinado padrão, trabalhando em conjunto com expressões regulares e, assim, tornando possível determinar ações que possam ser executadas quando um padrão for encontrado nos arquivos de *log*.

3.3 Comparativo

Dentre os sistemas apresentados, foi verificado que nenhum possui monitoramento hierárquico e também monitoramento no nível de aplicação (processo). Alguns (quando possuem) monitoram apenas métricas básicas, como, por exemplo, consumo de memória e CPU. Dentro do ambiente de *Cloud Computing*, torna-se necessário um monitoramento que, além de efetuar a coleta de informação de desempenho de um recurso especifico, também capture e analise toda a árvore de recursos computacionais envolvidos e não apenas o recurso fonte. A Tabela 1 efetua uma análise comparativa sobre as funcionalidades existentes.

Os trabalhos relacionados foram resumidos e comparados, segundo suas principais particularidades. Para efeito de comparação, foram relacionadas às seguintes características:

- **Monitoramento de Máquina Física**: O sistema deve efetuar o monitoramento dos recursos sob uma máquina física, tal como CPU, memória RAM, disco e comunicação (rede);

- **Monitoramento de Máquina Virtual**: O sistema deve efetuar o monitoramento dos recursos que executam sob uma determinada máquina virtual;

- **Monitoramento de Processo**: O sistema deve ser capaz de monitorar dados de desempenho dos processos de uma determinada aplicação;

Tabela 1: Comparação das características dos trabalhos relacionados

Item de Comparação	Máquina Física	Máquina Virtual	Processo	Multinível	Preditivo
Ganglia	X	X			
Nagios	X	X	X		
PCMONS	X	X			
DARGOS	X	X	X		
AWS	X	X			
Eucalyptus	X	X			
OpenNebula	X	X			
OpenStack	X	X			

Fonte: Elaborado pelo autor

- **Monitoramento Multinível**: O sistema deve efetuar o monitoramento hierárquico nos três níveis: Máquina Física, Máquina Virtual e Aplicação (processo);
- **Monitoramento Preditivo**: O sistema deve prever anomalias e situações de desempenho conforme dados históricos.

3.4 Considerações sobre o Capítulo

Todos os trabalhos estudados possuem mecanismos de monitoramento, seja eles próprios como solução fim ou com a possibilidade de integração com outros sistemas. No contexto de *middleware* IaaS para *Cloud Computing*, foi possível verificar que o Eucalyptus não possui um sistema de monitoramento próprio, porém, disponibiliza meios de integração para outras ferramentas. Já o OpenNebula possui um sistema de monitoramento nativo, que efetua a coleta de métricas de máquinas físicas e virtuais. Enquanto o AWS e o OpenStack possuem soluções próprias, CloudWatch e Ceilometer, respectivamente, ambos atuando no nivel de máquina física e virtual.

No contexto de sistemas de monitoramento para *Cloud Computing* tem-se o PCMONS e o DARGOS. O PCMONS, utiliza-se das informações coletadas através do Nagios e as disponibiliza para o usuário através do seu arcabouço. Enquanto o DARGOS, possui como principal característica, dispor de uma arquitetura distribuída de monitoramento na Nuvem através de um modelo hibrido usando agentes. Quanto aos sistemas de monitoramento de uso geral, tem-se o Nagios e o Ganglia. Ambos sistemas são utilizados em diversas plataformas devido a suas flexibilidades e customizações.

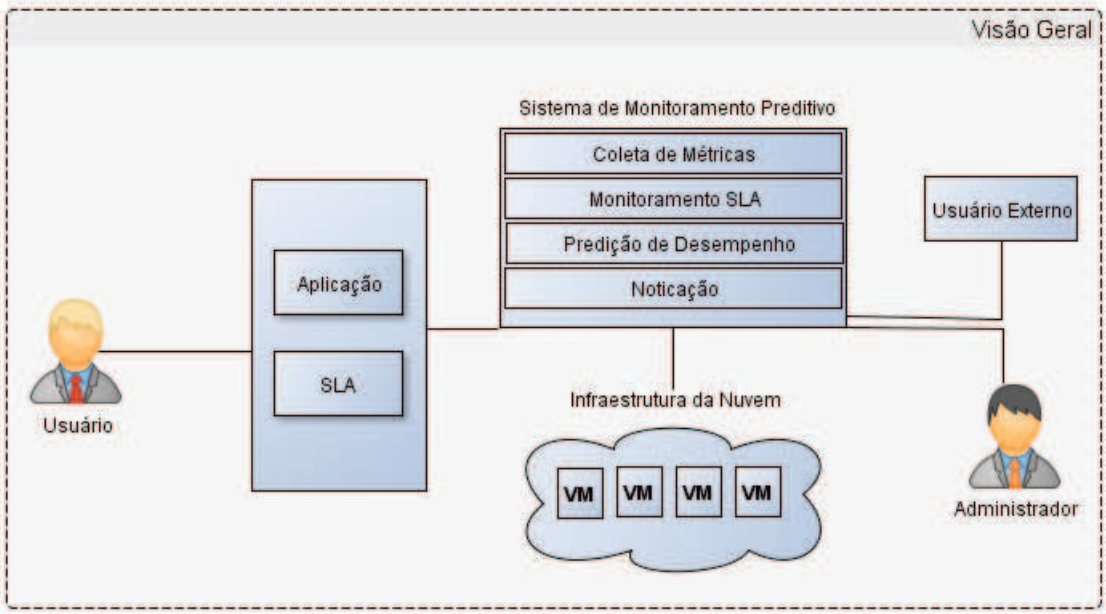
Em suma, este capitulo apresentou seis trabalhos relacionados ao monitoramento de recursos em *Cloud Computing*. Foi possível observar que o monitoramento de máquinas físicas e virtuais é um padrão dentre eles. O estudo dos trabalhos relacionados possibilitou a identificação de lacunas no contexto de monitoramento multinível e preditivo de recursos.

4 MODELO DE MONITORAMENTO PREDITIVO

Os capítulos anteriores servem como base para as decisões de projetos definidas no sistema proposto. Neste capítulo, é apresentada a proposta de um modelo de sistema de monitoramento preditivo para Computação em Nuvem para aplicações *Bulk Synchronous Parallel* (BSP). Esse sistema servirá como ferramenta de apoio para os administradores da nuvem nas seguintes etapas: (i) gerenciamento; (ii) monitoramento; (iii) diagnóstico; (iv) predição de desempenho; e (v) planejamento dos recursos. Este modelo de monitoramento preditivo, possui como característica-chave, a coleta e a predição de métricas de desempenho presentes nos três níveis da infraestrutura da Nuvem: aplicação, máquina virtual e máquina física.

O sistema de monitoramento preditivo foi elaborado visando tornar-se uma ferramenta auxiliar para o supervisão e predição de recursos da hierarquia da Nuvem, monitorando métricas das aplicações paralelas desenvolvidas sob o modelo BSP, das máquinas hospedeiras e a das respectivas máquinas virtuais instanciadas na infraestrutura. A Figura 4 ilustra a arquitetura do modelo e seu respectivo fluxo de trabalho.

Figura 4: Visão geral do Sistema



Fonte: Elaborado pelo autor

O usuário de posse de uma aplicação desenvolvida sobre o modelo BSP e um SLA acordado com o provedor de serviço, submete a aplicação para execução dentro da infraestrutura da Nuvem. Uma vez que a aplicação esteja sendo executada, o sistema de monitoramento efetuará a coleta de métricas, avaliação continua do SLA, predição de desempenho e notificação em situações onde seja necessário o provisionamento de recursos, visando manter o acordo de nível de serviço.

4.1 Requisitos do Modelo

Uma das motivações para o desenvolvimento do modelo de sistemas de monitoramento preditivo está na capacidade em ter um controle mais fino sobre a taxa de utilização dos recursos existentes na infraestrutura, além da previsão de desempenho. De posse destas informações, torna-se possível o planejamento e organização dos recursos da infraestrutura. Em adição, o sistema de monitoramento foi elaborado com as seguintes características-chave:

- *Transparência no nível de aplicação*: A atuação do monitor ocorrerá sem a intervenção do usuário final, ou seja, o monitor executará de modo contínuo dentro da infraestrutura da Nuvem. O monitoramento da aplicação será concebido no nível de *middleware*;
- *Suporte a aplicações do tipo BSP*: O sistema efetuará o monitoramento de aplicações do tipo BSP, um dos modelos mais utilizados para programação paralela, em ambientes de *Cloud Computing*;
- *Monitoramento Multinível*: O sistema de monitoramento coletará informações de desempenho nos três níveis da hierarquia da infraestrutura: máquina física, máquina virtual e aplicação;
- *Predição de Desempenho*: O sistema de monitoramento efetuará a coleta, o registro e a predição de desempenho das métricas relacionadas aos três níveis da hierarquia de recursos;
- *Suporte a SLA*: O sistema de monitoramento efetuará constantemente a análise e validação das métricas coletadas junto ao SLA da aplicação;
- *Independente de Plataforma*: O sistema de monitoramento foi concebido para executar de forma independente de plataforma ou *middleware* de *Cloud Computing*;
- *Integrável e Expansível*: A arquitetura desenvolvida suportará interfaces para integração com outros sistemas, como, por exemplo, escalonadores e/ougerenciadores de recursos da infraestrutura da Nuvem.

4.2 Decisões de Projeto

O sistema de monitoramento proposto oferecerá um ambiente de coleta de métricas de desempenho dos recursos em três níveis (máquina física, máquina virtual e aplicação). De uma forma geral, as seguintes premissas são consideradas importantes em relação ao modelo proposto:

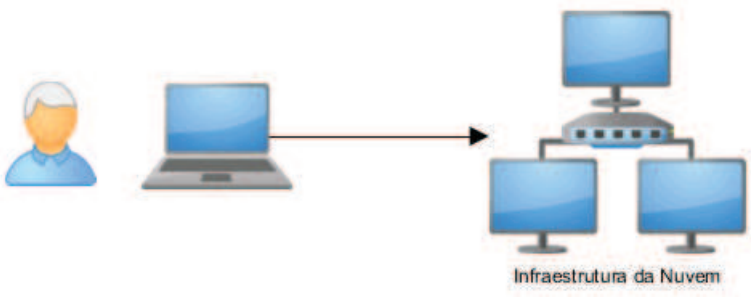
- Utilização de BSP para as aplicações, visto que é um dos modelos mais utilizados hoje em dia para programação paralela e também por ser modelado em fases que possuem uma função de custo.

- Desenvolver um sistema que efetue o monitoramento de aplicações do tipo BSP em ambientes de *Cloud Computing*;
- Desenvolver um sistema de monitoramento que coletará informações de desempenho nos três níveis: máquina física, máquina virtual e aplicação;
- Desenvolver um sistema de monitoramento independente de *middleware* de *Cloud Computing*;
- Efetuar o monitoramento da aplicação BSP no nível de *middleware*;
- O monitor ficará na máquina gerenciadora da Nuvem, ficando responsável pela coleta, pelo registro e pelo processamento das métricas das máquinas físicas, máquinas virtuais e aplicação;
- A atuação do monitoramento ocorrerá sem a intervenção do usuário final, ou seja, o sistema de monitoramento estará presente em todos os níveis da infraestrutura da Nuvem, sem que o usuário ou administrador necessite ativa-lo.

4.3 Modelagem

Dentro de uma infraestrutura de Computação em Nuvem, tem-se a diversidade de recursos, tal como máquinas físicas e virtuais. O sistema de monitoramento será responsável por analisar e registrar as informações de desempenho dos recursos computacionais e verificar se estes continuam válidos, conforme contrato definido pelo SLA. A Figura 5 apresenta uma visão geral sobre o procedimento de envio da aplicação pelo usuário ao sistema proposto.

Figura 5: Visão geral do envio da aplicação

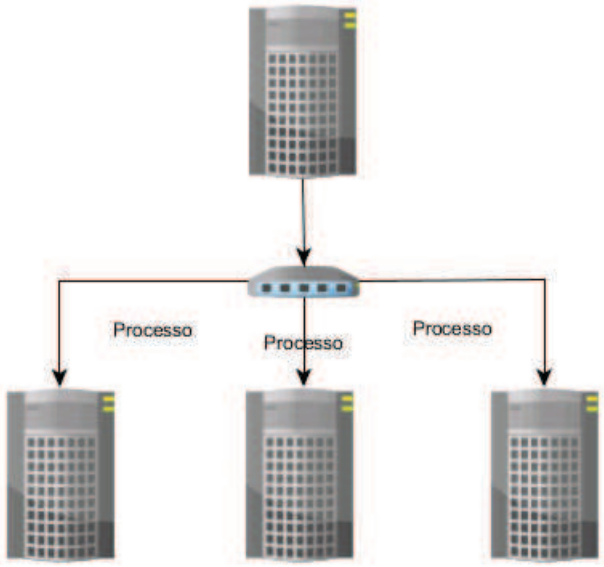


Fonte: Elaborado pelo autor

Após a submissão da aplicação para o *middleware* BSP dentro de uma infraestrutura de *Cloud Computing*, o sistema de monitoramento ficará responsável por efetuar o controle sobre

a taxa de disponibilidade dos recursos presentes na infraestrutura de nuvem e a constante fiscalização do mantimento e/ou violação dos contratos SLA. Este monitoramento será executado em três níveis (ou multinível): máquina física, máquina virtual e aplicação. A Figura 6 apresenta o fluxo da distribuição dos processos entre as máquinas físicas e, posteriormente, maquinas virtuais.

Figura 6: Visão geral sobre a distribuição dos processos BSP entre máquinas



Fonte: Elaborado pelo autor

No monitoramento multinível, cada máquina física será monitorada em relação aos seus recursos físicos e virtuais utilizados sobre o nó e pela aplicação, efetuando, assim também, o monitoramento de processos. Desta forma, tem-se uma hierarquia de monitoramento, atingindo do recurso físico até o nível de instruções, no caso, o monitoramento de uma aplicação e seus processos distribuídos na infraestrutura da Nuvem. Entre as políticas do modelo de sistema de monitoramento, tem-se:

- **Monitoramento Periódico**: O sistema de monitoramento será executado conforme intervalo de tempo previamente configurado, como, por exemplo, numa determinada frequência, coletando estas métricas e efetuando a análise histórica sobre as informações de desempenho coletadas;

- **Monitoramento Contínuo**: Neste modo de monitoramento, o sistema efetuará, continuamente as etapas de coletas, predição, validação do SLA, notificação e, por fim, retornando à fase inicial. Desta forma, a cada término de uma janela de execução, inicia-se uma próxima sem um intervalo previamente configurado entre as execuções;

- **Monitoramento Periódico-Adaptativo**: O sistema de monitoramento efetuará a adaptação elástica no tempo de monitoramento, ou seja, por meio de uma verificação periódica

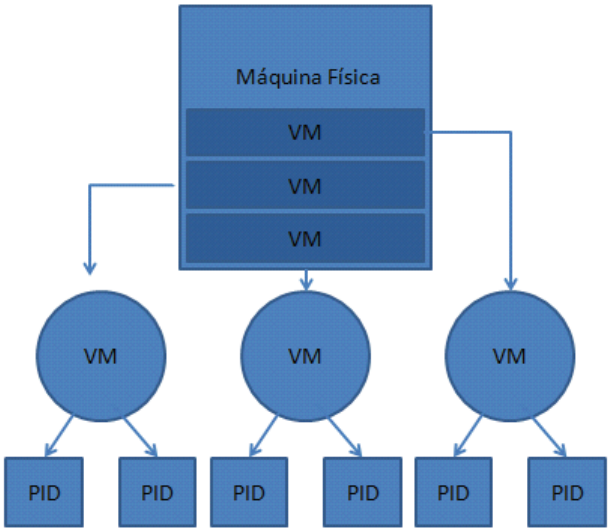
de tempo em tempo, o intervalo de monitoramento poderá aumentar ou diminuir conforme a criticidade do evento raiz. Em situações nas quais o período definido continue sempre estável, o monitoramento periódico-adaptativo aumentará o intervalo até a próxima coleta. Porém, em caso de falha ou violação do SLA, este tempo será diminuído pela metade, diminuindo progressivamente até chegar ao tempo inicial do monitoramento.

Após a execução do monitoramento sob os recursos, em situações em que o monitor encontre situações críticas de desempenho, falhas ou violações do SLA, esse enviará notificações, por exemplo, a um administrador da Nuvem ou a um determinado balanceador de carga para que este efetue a correção do problema da maneira menos custosa.

4.3.1 Métricas Monitoradas

As métricas de desempenho monitoradas compreendem os três níveis da Nuvem: Máquina física, máquina virtual e aplicação. Estas informações coletadas são agrupadas em métricas estáticas e dinâmicas. As estáticas são informações fixas, como arquitetura do processador ou número de processadores. Já as dinâmicas são informações que variam constantemente e, dessa forma são inspecionadas suas respectivas taxas de utilização a cada intervalo de monitoramento. A Figura 7 apresenta a hierarquia no monitoramento entre as entidades.

Figura 7: Níveis de Monitoramento: Máquina física, virtual e processo



Fonte: Elaborado pelo autor

Neste contexto, dentro de uma infraestrutura de *Cloud Computing*, uma máquina física é composta por uma ou mais máquinas virtuais, e dentro de cada máquina virtual um ou mais processos de uma aplicação.

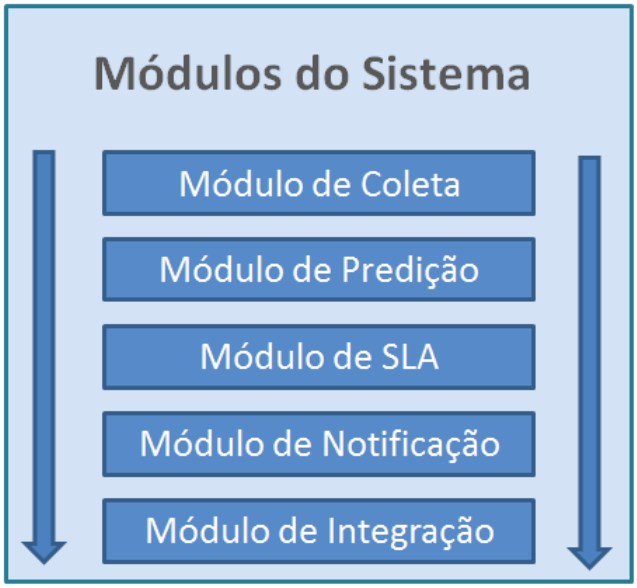
- **Máquina Física**: Uso de disco, uso de memória RAM, uso de CPU, tamanho do disco de armazenamento, máximo de memória, máximo de CPU, quantidade de memória RAM livre, espaço disponível em disco, percentual de CPU livre e a quantidade de máquinas virtuais em execução na máquina física;
- **Máquina Virtual**: Uso de memória RAM, uso de CPU, quantidade de envio e recebimento de dados pela rede;
- **Aplicação**: Custo de computação, custo de comunicação e custo de sincronização.

4.4 Módulos do Sistema

No nível da aplicação, o sistema de monitoramento proposto se comunicará com um *middleware* BSP visando à coleta das métricas de desempenho da aplicação paralela. No nível da infraestrutura IaaS da Nuvem, serão coletadas informações de desempenho dos recursos das máquinas físicas e virtuais. Esta coleta de métricas de desempenho deverá ter baixa intrusividade sobre o ambiente.

O sistema de monitoramento estará presente na máquina gerenciadora da infraestrutura da Nuvem, efetuando, constantemente, a coleta das métricas de desempenho das máquinas físicas e virtuais. Na camada da aplicação, estarão presentes sensores para a coleta das métricas do *middleware* BSP e as enviando-as para agregação para o monitor principal. O sistema de monitoramento preditivo BSPMon será composto pelos módulos de envio, seleção, distribuição, coleta, SLA e predição de desempenho. A Figura 8 apresenta uma visão geral dos módulos.

Figura 8: Visão Geral dos Módulos



Fonte: Elaborado pelo autor

- **Módulo de Coleta**: Responsável por efetuar a coleta das informações / métricas de desempenho das máquinas físicas e virtuais em toda infraestrutura da Nuvem;

- **Módulo de Predição**: Responsável por analisar e prever um padrão de desempenho conforme métricas coletadas. Para tal, as informações são armazenadas visando manter um histórico e assim possibilitar a construção de séries temporais. Neste contexto, após a formulação da série, é utilizado técnica de predição de séries temporais.

- **Módulo de SLA**: Responsável por validar se as métricas de desempenho coletadas estão de acordo com o contrato de SLA e também por efetuar o envio de notificações de eventos relacionados com as métricas processadas. Estes eventos podem ser enviados para um administrador da infraestrutura de Nuvem ou para um gerenciador/escalonador externo, solicitando a redução ou aumento de recursos conforme necessidade;

- **Módulo de Notificação**: Responsável por notificar falhas e/ou oportunidades de ganho sob os recursos presentes na infraestrutura;

- **Módulo de Integração**: Responsável por facilitar a integração de sistemas externos ao sistema de monitoramento, possibilitando, desta forma, o uso das métricas coletadas, bem como informações como o estado da infraestrutura, falhas e os respectivos SLA's.

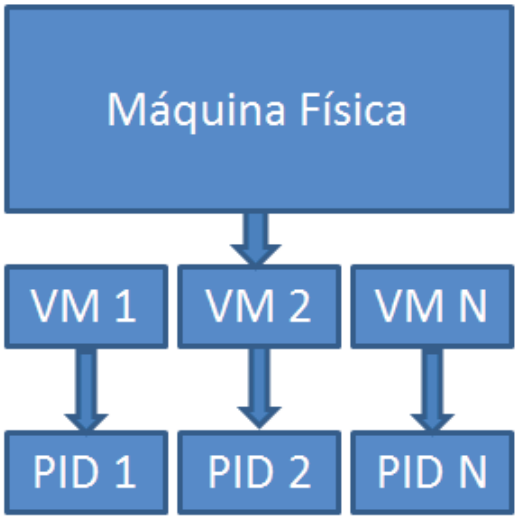
4.4.1 Módulo de Coleta

O monitoramento coletará as informações de desempenho das máquinas físicas, máquinas virtuais e aplicações (processos). De posse destas informações, será possível efetuar um controle mais fino sobre os recursos disponíveis na infraestrutura da Nuvem e, assim, notificar um administrador da Nuvem e/ou escalonador para que este reorganize os recursos da infraestrutura em situações de falhas no SLA ou oportunidade de ganho, no sentido de reduzir o número de recursos ativos ou subutilizados. A Figura 9 apresenta, de uma forma geral, os três níveis dos recursos a serem monitorados na infraestrutura.

A cada novo valor coletado, este será persistido em base de dados para posterior processamento e análise. Este módulo coleta dois grupos de informações (estáticas e dinâmicas). No primeiro grupo, são coletadas informações estáticas especificadas sobre os recursos, como, por exemplo, sistema operacional e arquitetura da máquina. No segundo grupo, são coletadas informações dinâmicas de desempenho dos recursos, como, por exemplo, uso de memória, CPU e armazenamento.

A combinação dos parâmetros de monitoramento será utilizada para determinar a quantidade de recursos que, de fato, uma aplicação está utilizando. A quantidade de recurso disponível em uma máquina física pode ser determinada pela soma dos recursos livres (além dela própria) em cada máquina virtual instanciada. Esta combinação torna-se necessária para um

Figura 9: Níveis de Monitoramento



Fonte: Elaborado pelo autor

melhor controle de quando for feita a consolidação e/ou provisionamento dos recursos. Como exemplo, pode-se citar os seguintes cenários:

- Cenário 1 - Consolidação de Recurso: A máquina física possui 1024 MB de memória, porém, 768 MB estão distribuídos por três máquinas virtuais instanciadas (Máquinas A, B e C). A aplicação está distribuída em todas as máquinas virtuais com diferentes quantidades de memória (128mb, 512mb, 128mb, respectivamente). Neste cenário, cada máquina está consumindo apenas 50 % da memória RAM. Desta forma, é possível notificar um escalonador externo que efetue a migração de processos das máquinas A e C (ambas com 64 MB de memória utilizada) para máquina B, que dispõe de 256 de memória RAM disponível. Logo após o término da migração, é possível efetuar a consolidação das máquinas virtuais A e C, liberando recursos da máquina física;

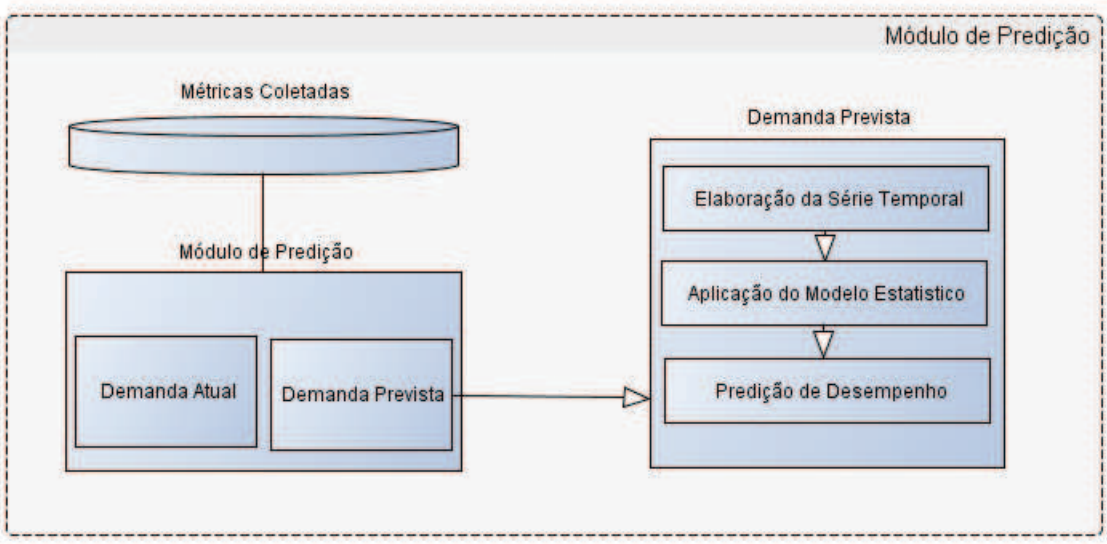
- Cenário 2 - Provisionamento de Recurso: A máquina física "A"possui 1024 MB de memória, porém, 90 % MB estão distribuídas por três máquinas virtuais instanciadas de forma igualitária (Máquinas A, B e C). Neste cenário, o consumo de memória cresce continuamente. Desta forma, visando a não comprometer o desempenho, a disponibilidade do serviço e o SLA, é enviada uma notificação ao administrador da Nuvem ou escalonador para esse efetue o provisionamento de mais recursos na infraestrutura, como, por exemplo, subindo uma máquina virtual em outra máquina física visando atender a demanda.

4.4.2 Módulo de Predição

O módulo de predição é responsável por efetuar a busca dos valores coletados das métricas e, então, efetuar a predição de desempenho baseada numa determinada janela de tempo. Após processados os dados monitorados e previstos um conjunto de valores para a janela de tempo, estes serão enviados para o módulo de SLA, para que sejam validados a partir da demanda atual e demanda prevista.

De posse das métricas coletadas, é, então elaborada a série temporal correspondente. Após a construção da série temporal, é, então, aplicado o modelo de predição através de um método estatístico de predição de séries temporais. A Figura 10 apresenta o fluxo de execução presente no módulo de predição. Ao término da execução do módulo, são retornados dois conjuntos de valores: a demanda atual, representada pelo consumo atual dos recursos, e a demanda futura, representada pelo consumo previsto pela série temporal.

Figura 10: Visão geral do Módulo de Predição



Fonte: Elaborado pelo autor

O fluxo do módulo de predição consiste nas seguintes etapas:

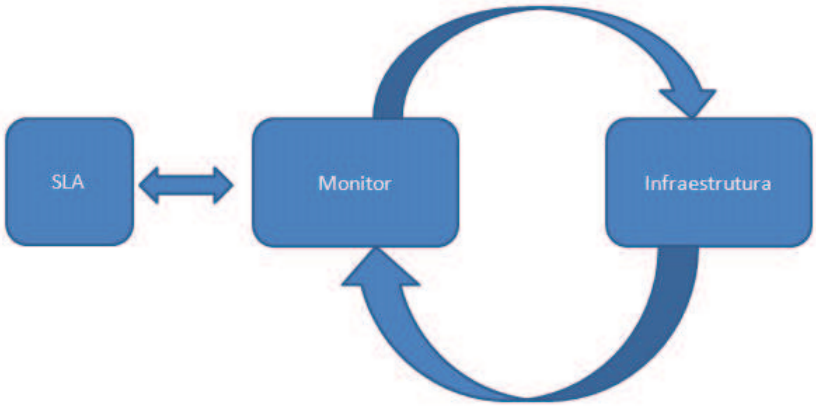
1. Recuperar na base de dados todas as informações de monitoramento coletadas até o momento;
2. Elaborar uma série temporal baseada nos valores recuperados e na frequência da coleta;
3. Processar as amostras a partir de um modelo estatístico;
4. Retornar um conjunto de valores, contendo o valor atual coletado e a demanda futura.

O conjunto de valores: demanda atual e demanda futura são repassados para o modelo de SLA para respectiva validação. Esta validação será feita com base no SLA submetido pelo cliente.

4.4.3 Módulo de SLA

O sistema de monitoramento efetuará, periodicamente (conforme política de monitoramento), a coleta de métricas de desempenho de cada recurso presente na infraestrutura da Nuvem. A cada coleta, as informações serão validadas junto ao SLA registrado da aplicação. Os parâmetros e seus limiares definidos no documento SLA serão comparados com os dados coletados. A Figura 11 apresenta uma visão geral do processo de coleta e SLA pela rotina de monitoramento.

Figura 11: Visão da comunicação entre módulo de monitoramento e SLA



Fonte: Elaborado pelo autor

O SLA pode ser utilizado para prever em contrato a demanda de mais recursos do que o assumido previamente. Desta forma, pode-se exigir que o SLA garanta que o serviço continue em funcionamento constantemente. A cada coleta das métricas de desempenho, estas informações serão validadas junto ao módulo de SLA. Nesta validação, podem ocorrer três cenários:

- Cenário 1: As métricas definidas no SLA estão muito acima da real necessidade da aplicação. Neste caso, solicita-se ao administrador da Nuvem ou provisionador externo que efetue a diminuição da utilização dos recursos pela aplicação do usuário. Desta forma, os recursos que estiverem ociosos em relação à aplicação e contrato SLA retornarão para o administrador da Nuvem e poderão gerar receita ao efetuar o empréstimo dele a outros usuários;

- Cenário 2: As métricas definidas pelo SLA estão bem ajustadas, ou seja, o consumo real da aplicação está de acordo com o contrato definido entre usuário e provedor da infraestrutura da Nuvem. Desta forma, tem-se um consumo de recursos ótimo em relação às necessidades da aplicação;

- Cenário 3: As métricas definidas pelo SLA foram subestimadas, ou seja, a aplicação demanda muito mais recursos do que acordado em contrato. Desta forma, torna-se neces-

sário o envio de uma requisição ao administrador da Nuvem ou provisionador externo que amplie a disponibilidade de recursos para aquela aplicação e envie notificação ao cliente final sob tal necessidade.

Sempre que ocorrer uma anomalia e/ou uma violação no SLA, o administrador e/ou escalonador será notificado desta ocorrência. Para que esse tome uma ação visando corrigir os problemas ocorridos devido à violação, podendo, neste caso, efetuar o provisionamento de novos recursos para a aplicação, visando o cumprimento do SLA contrato. O componente de SLA possui as seguintes características-chave:

- *Avaliação Continua do SLA*: As métricas serão constantemente avaliadas e validadas com os limiares definidos pelo SLA;

- *Potencial de Violação*: De posse das amostras das métricas coletadas pelo componente de monitoramento, estas informações serão analisadas pelo componente de predição. O potencial de violação refere-se a uma possível violação futura conforme resultado previsto recebido pelo componente de predição;

- *Potencial de Provisionamento*: O potencial de provisionamento refere-se a uma eventual necessidade de provisão de mais recursos para a infraestrutura;

- *Potencial de Consolidação*: O potencial de consolidação, conforme métricas coletas, refere-se a eventuais oportunidades de consolidação dos recursos da Nuvem.

4.4.4 Módulo de Notificação

O módulo de notificação é responsável por alertar o administrador da Nuvem e/ou escalonador sobre alguma anomalia ou oportunidade de ganho, seja estes em relação à redução dos recursos alocados devido à pouca demanda prevista ou devido à projeção futura indicar um aumento no uso dos recursos e a possibilidade de falha no SLA, desta forma, é enviada uma notificação prevendo eventuais multas ao prestador de serviço.

Para notificação ao administrador da infraestrutura, cada anomalia ou oportunidade de ganho identificada, será possível por meio de duas formas: recebimento da notificação por e-mail e/ou visualização das ocorrências.

Devido à quantidade de notificações, será utilizada a correlação de eventos, dada a possibilidade de inúmeros eventos em um curto espaço de tempo, de forma que seja possível identificar rapidamente caso esteja ocorrendo algum problema em algum recurso. A técnica utilizada será a de compressão, que compreende a detecção dentro de uma determinada janela de tempo, múltiplas ocorrências de um mesmo evento, resumindo e agregando os eventos correspondentes em um único.

Para a manipulação, envio e recebimento de e-mails, foi utilizada o JavaMail, uma API independente de plataforma, que provê uma interface padrão para acessar provedores de serviços para o gerenciamento de mensagens eletrônicas. Esta API é composta por um conjunto de classes abstratas que oferecem suporte para interação com servidores de e-mail.

4.4.5 Módulo de Integração

O módulo de integração serve como uma interface para um escalonador de recursos. Durante a conexão com esta interface, o escalonador poderá receber as seguintes informações: (i) Demanda atual e demanda futura; (ii) sla; (iii) notificações de falhas de SLA; (iv) alertas de oportunidades de provisionamento e/ou consolidação de recursos; Através da combinação de informações de desempenho, o módulo de integração poderá notificar duas situações:

- Possibilidade de falha no SLA:
    1. Efetuar um reajuste nos recursos. Visando o melhor uso dos recursos, pode-se ser efetuada a migração, consolidação e/ou provisionamento.
- Se recursos estiverem balanceados:
    1. Tentar uma atividade oportunista, como migração de processos/VMS, através da predição de desempenho.

4.5 Considerações sobre o Capítulo

Neste capítulo, foi apresentado o modelo de monitoramento preditivo de recursos para Computação em Nuvem. Este modelo é dividido em módulos, em que cada qual possui uma responsabilidade diferente durante a etapa de monitoramento. Os módulos presentes na arquitetura são: coleta, predição, SLA, notificação, visualização e integração.

No próximo capítulo, será descrita a implementação do protótipo a partir do modelo proposto neste trabalho, iniciando pela apresentação da implementação de cada módulo da arquitetura e finalizando com a descrição das decisões de implementação efetuadas para que o protótipo funcione de acordo com o modelo descrito neste capítulo.

5 IMPLEMENTAÇÃO DO MODELO

No capítulo anterior, foi apresentado o modelo proposto do sistema de monitoramento preditivo de recursos em Computação em Nuvem. Neste capítulo será descrita a implementação do modelo, relacionando as técnicas e tecnologias utilizadas para o desenvolvimento do protótipo. O sistema de monitoramento foi desenvolvido utilizando a linguagem de programação Java, a linguagem estatística R, o servidor de aplicação com suporte a plataforma JEE (*Java Platform, Enterprise Edition*) JBoss. Como infraestrutura de nuvem, foi utilizado o *middleware* IaaS de *Cloud Computing* OpenNebula. No nível de aplicação BSP, foi utilizado o middleware PUBWEB.

5.1 Coleta

O módulo de coleta é responsável por capturar informações de desempenho das métricas nos três níveis da infraestrutura e então, repassar para o módulo de predição. A cada coleta concluída, a informação é persistida em base de dados. Para a coleta das métricas de desempenho, foi desenvolvido neste módulo um conjunto de rotinas através da Sigar API (WVWARE, 2013).

As métricas de desempenho uma vez coletadas, são persistidas em base de dados. Para tal, foi utilizado o MongoDB, um banco de dados NoSQL orientado a documentos escrito em C++. De acordo com (BUNCH et al., 2010), o MongoDB foi desenvolvimento visando prover rapidez e escalabilidade no armazenamento de dados chave-valor.

No nível da aplicação, as informações são coletadas através da instrumentação efetuada no PUBWEB, um *middleware* Java para a computação massivamente paralela de aplicações BSP (*Bulk Synchronous Parallel*) (BONORDEN; GEHWEILER; HEIDE, 2006a). A Figura 12 apresenta a arquitetura. Desta forma, é possível observar e extrair métricas de desempenho das aplicações paralelas em execução na infraestrutura.

A Tabela 2 apresenta os métodos instrumentados no middleware PUBWEB para o nível de aplicação, visando a captura dos custos de uma aplicação desenvolvida sob o modelo BSP.

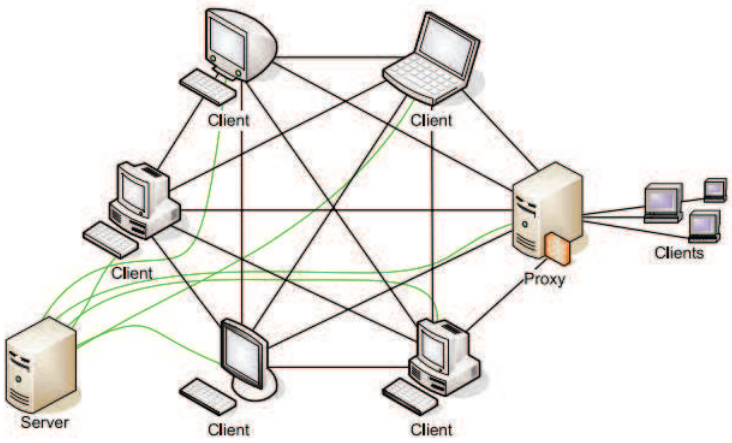
Tabela 2: Métodos instrumentados

Método	Função de Interceptação
bspMain(BSP bspProgram, Serializable args)	Computação
send (int pid, Serializable msg)	Comunicação
send (int pidLow, int pidHigh, Serializable msg)	Comunicação
send (int[] pids, Serializable msg)	Comunicação
sync	Sincronização
syncMig	Sincronização

Fonte: Elaborado pelo autor

Estes custos são relacionados ao tempo de execução do método analisado, ou seja, tempo

Figura 12: Arquitetura do PUBWEB



Fonte: (BONORDEN; GEHWEILER; HEIDE, 2006b)

de computação refere-se ao tempo gasto em computar o algoritmo, o tempo de comunicação é relacionado ao envio e confirmação de recebimento e o tempo de sincronização, é o tempo gasto para sincronização entre os processos BSPs.

5.2 Predição

O módulo de predição é responsável por prever o desempenho baseado nas métricas existentes na base de dados. Este módulo é invocado a cada novo período de coleta e então, atualiza as previsões com as novas amostras.

A cada período de predição, os dados são processados, visando gerar uma nova série temporal para análise estatística. Os dados então são enviados para o RServe (URBANEK, 2003), um servidor TCP/IP que permite a iteração com o R, uma linguagem estatística, sem a necessidade de inicializar uma nova sessão ou empacotamento da biblioteca. Cada conexão feita ao RServe possui um *workspace* separado, desta forma, não há problemas de compartilhamento de buffers entre as sessões. O código 1 demonstra o algoritmo que efetua a geração e predição de valores.

Result: Predição de desempenho num horizonte de N passos
while existir valores históricos para a métrica do
    recuperar valor atual;
    recuperar valores anteriores;
    gerar série temporal [valores anteriores + atual] ;
    prever N passos
end

Código 1: Algoritmo de Predição

De posse de uma base histórica para cada métrica monitorada, recupera-se o valor atual e

seus valores anteriores do recurso presente na Nuvem. Estes dados são recuperados através de uma consulta a base dados. A partir dessas métricas de desempenhos, gera-se uma série temporal através da linguagem estatística R. Esta série temporal gerada é então aplicada em um modelo de predição de séries temporais, neste caso, o modelo ARIMA. Após esta etapa, é então gerada a previsão de desempenho de num horizonte de N passos.

5.3 SLA - Service Level Agreement

O módulo de SLA (*Service Level Agreement*) é responsável por validar as informações de desempenho passadas a ele. Estas informações correspondem a demanda atual e a demanda futura. Para a validação, é feita a comparação das métricas com o SLA registrado para a aplicação. O SLA é submetido para análise num documento XML baseado no WSLA. Após as devidas validações de cada métrica e seus limiares, em caso de falhas, são enviados alertas para o módulo de notificação.

O SLA é enviado junto com a aplicação BSP, sendo este documento baseado no WSLA, um framework para monitoramento de SLA's em Web Services. Neste arquivo são descritas quais métricas serão monitoradas e os respectivos níveis de atendimento. O documento SLA foi definido com os seguintes parâmetros e métricas:

- *Metric*: Corresponde a um parâmetro individual a ser monitorado.
- *Frequency*: Define os intervalos de tempo de coletas para cada métrica definida no documento.
- *Threshold* Corresponde os limites máximos e mínimos que em caso forem excedidos deve ser alertado.

Os parâmetros definidos no documento de SLA são compostos de nome, tipo e unidade. Estes parâmetros podem ser separados por características. A Tabela 3 apresenta os campos disponíveis.

Tabela 3: Métricas SLA XML

Métrica	Threshold	Type
Availability	Valor ou Range	Absoluto ou Percentual
CPU	Valor ou Range	Absoluto ou Percentual
Memory	Valor ou Range	Absoluto ou Percentual
Disk	Valor ou Range	Absoluto ou Percentual
VirtualMachines	Valor ou Range	Absoluto ou Percentual

Fonte: Elaborado pelo autor

Estes atributos são necessários, visando a constante validação dos contratos de nível de serviço dos recursos dentro da infraestrutura da *Cloud Computing*. O modelo de SLA utilizado

é baseado em *threshold*, ou seja, através de limites, que podem ser um valor absoluto ou um marco percentual.

Um SLA Máquina Física é responsável por controlar as métricas dos recursos dos servidores presentes na infraestrutura. O SLA definido é representado pela figura 13.

Figura 13: SLA Máquina Física

```
<?xml version="1.0" encoding="UTF-8"?>
<sla name="SLA Maquina Fisica" type="Host">
    <metrics>
        <metric name="Availability">
            <threshold value="99" type="Percent" />
        </metric>
        <metric name="cpu">
            <threshold value="50" type="Usage Percent" />
        </metric>
        <metric name="memory">
            <threshold value="40" type="Usage Percent" />
        </metric>
        <metric name="disk">
            <threshold value="30" type="Usage Percent" />
        </metric>
        <metric name="virtualmachines">
            <metric name="provision">
                <threshold baseline="1" max="3" type="Value" />
            </metric>
        </metric>
        <violation>
            <actions>
                <action type="Notification" value="customer@mail.com" />
            </actions>
        </violation>
    </metrics>
</sla>
```

Fonte: Elaborado pelo autor

Um documento de SLA para Máquinas Virtuais é responsável por controlar as métricas das VMs instanciadas na infraestrutura. A Figura 14 apresentado o SLA definido para máquinas virtuais.

Figura 14: SLA Máquina Virtual

```
<sla name="SLA Maquina Virtual" type="VirtualMachine">
<metrics>
    <metric name="Availability">
        <threshold value="99" type="Percent"/>
    </metric>
    <metric name="cpu">
        <threshold value="50" type="Usage Percent"/>
    </metric>
    <metric name="memory">
        <threshold value="40" type="Usage Percent"/>
    </metric>
    <metric name="network">
        <threshold value="30" type="Received"/>
        <threshold value="30" type="Transmitted"/>
    </metric>
</metrics>
</sla>
```

Fonte: Elaborado pelo autor

O SLA de uma aplicação é composto por métricas de desempenho de um programa desenvolvimento sob o modelo BSP. Este SLA é composto pelas seguintes propriedades, representado pela figura 15.

Figura 15: SLA Aplicação

```
<sla name="SLA Aplicacao" type="Application">
<metrics>
    <metric name="computation">
        <threshold value="10" type="Cost"/>
    </metric>
    <metric name="communication">
        <threshold value="10" type="Cost"/>
    </metric>
    <metric name="sync">
        <threshold value="20" type="Cost"/>
    </metric>
</metrics>
</sla>
```

Fonte: Elaborado pelo autor

5.4 Notificação

O módulo de notificação é responsável por alertar o administrador da Nuvem e/ou escalonador que faça ajustes nos recursos, conforme identificado pelo módulo de SLA. A notificação é baseada nos thresholds das métricas e no respectivo SLA. O código 2 apresenta o algoritmo de notificação.

```
Result: Envia notificação baseado no SLA
while valores métricas SLA do
    recuperar demanda atual;
    recuperar demanda futura;
    if threshold atingido then
        envia notificação;
    end
end
```

Código 2: Algoritmo de Notificação

Neste contexto, recuperam-se os dados coletados e então, é efetuada uma analise das métricas do SLA através de dois tipos de demanda: atual e futura. A demanda atual é composta pelas métricas de desempenho correntes existentes dentro da Nuvem. Já a demanda futura, é composta pela previsão de consumo das métricas num horizonte N. A ativação de uma notificação é feita quando um *threshold* é atingido. De posse desta notificação, o administrador ou escalonador de recursos poderá efetuar os reajustes necessários visando o cumprimento dos acordos de SLA.

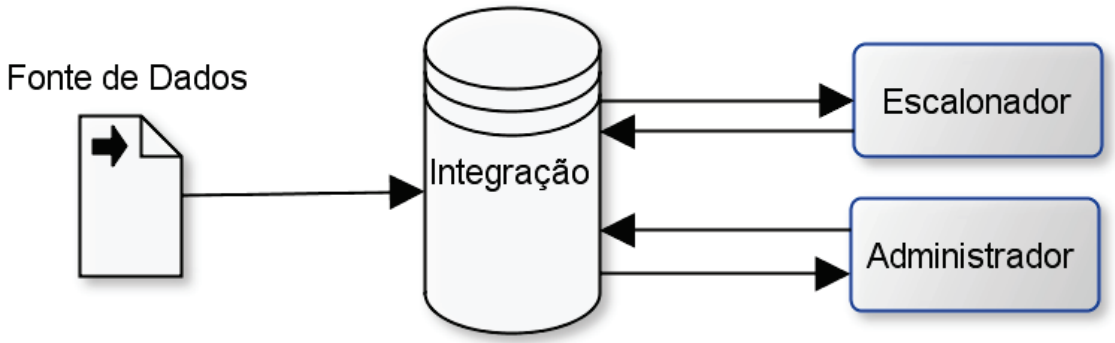
5.5 Integração

O módulo de integração é responsável por publicar métricas de desempenho e distribui-las para as aplicações externas, como por exemplo: escalonadores e gerenciadores de recursos. Uma das formas de integração é a utilização do JMS (*Java Message Service*) uma API Java para *middleware* orientado a mensagens, tornando possível que as aplicações se comuniquem através do serviço de mensageria. O modulo de integração foi desenvolvido sob o modelo *Publish/Subscriber*.

As mensagens enviadas são objetos que são transformados em XML para transmissão, para então serem processadas por um sistema externo. O *middleware* MOM (Message Oriented Middleware) utilizado para a camada de distribuições de notificações/eventos é o HornetQ, um sistema mensageria assíncrono de código aberto para a construção multi protocolo e alto desempenho (GIACOMELLI, 2012). A Figura 16 apresenta o modelo de integração e disponibilização dos dados.

As fontes de dados são compostas dos recursos presentes na infraestrutura e suas respecti-

Figura 16: Visão Integração dos Dados



Fonte: Elaborado pelo autor

vas métricas de desempenho coletadas. Na fase de integração, os dados fornecidos por estas fontes, são publicados e distribuídos a usuários ou serviços externos, como por exemplo, um escalonador de recursos, via JMS.

5.6 Considerações sobre o Capítulo

Neste capítulo, foram apresentadas as tecnologias utilizadas para o desenvolvimento deste trabalho. Como linguagem de programação, foi utilizado o Java e a linguagem estatística R. O middleware utilizado para suporte ao desenvolvimento de aplicativos BSP foi o PUBWEB, por além de outras coisas, possuir a capacidade de migração de processos entre máquinas. Por fim, como *middleware* de *Cloud Computing*, foi utilizado o OpenNebula.

Para cada módulo presente no modelo, foi implementado um serviço a parte, responsável por realizar suas devidas funcionalidades. Os módulos desenvolvidos foram: Módulo de coleta, responsável por capturar as informações das métricas de desempenho dos três níveis; Predição, responsável por prever os valores futuros das métricas persistidas na base de dados; SLA, responsável por efetuar a avaliação continua das informações de monitoramento coletadas juntos aos recursos; Notificação, responsável pelo envio de alertas para o gerenciador de recursos; E integração, responsável por publicar e distribuir via JMS as informações das métricas de desempenho coletadas.

No capítulo seguinte são apresentados os experimentos realizados com o protótipo e uma avaliação dos resultados coletados. Foram realizadas medições de intrusividade do sistema de monitoramento dentro da infraestrutura da Nuvem, avaliação do tempo de resposta do SLA, informações sobre a eficiência energética, ou seja, o quanto foi possível diminuir no consumo de energia em determinadas situações e aplicações.

6 AVALIAÇÃO

No capítulo anterior, foi descrito o protótipo desenvolvido a partir do modelo proposto neste trabalho. Este sistema de monitoramento foi desenvolvido visando à coleta, registro, predição e avaliação do SLA, buscando oportunidades de ganho de energia e melhoria no uso dos recursos computacionais dentro de uma infraestrutura de Nuvem. Neste capítulo, serão abordados os aspectos sobre a avaliação do modelo proposto, visando avaliar a intrusividade, o desempenho, eficiência energética, aspectos relacionados ao tempo de reação, validação do SLA, notificações e predições de desempenho.

6.1 Arquitetura dos Experimentos

Essa seção apresenta a arquitetura dos experimentos utilizado para a avaliação deste trabalho. Como gerenciador da Nuvem, foi utilizado o OpenNebula, um *middleware* IaaS *Open-Source* para Computação em Nuvem, que efetua o gerenciamento distribuído e dinâmico de infraestruturas virtuais, permitindo a instanciação e realocação de máquinas virtuais em um conjunto de máquinas físicas.

No *middleware* OpenNebula, quem efetua o gerenciamento dos nós *Slaves* é a máquina denominada *Front-End*, sendo esta responsável pela gestão de cada máquina física e virtual presente na infraestrutura da Nuvem. Desta forma, o sistema de monitoramento proposto, estará localizado dentro da máquina *Front-End*, visto que a mesma possui acesso a todas máquinas presentes na infraestrutura. Na Tabela 4, são apresentados os detalhes das versões de software, presentes na ambiente de avaliação.

Tabela 4: Ambiente de Software

Plataforma	Versão
Linux	CentOS 6.3
Kernel	2.6.32
Arquitetura	64 bits
Java	1.7.0_25
OpenNebula	4.3

Fonte: Elaborado pelo autor

Para avaliação, foi utilizado um cenário com três máquinas com capacidades heterogêneas. A principal é a *Front-End*, responsável por gerenciar a infraestrutura. As outras duas máquinas, cada qual, possui duas máquinas virtuais instanciadas. Na tabela 5, é apresenta as configurações de hardware das máquinas físicas.

Tabela 5: Ambiente de Hardware

Máquina	Processador	Memória
FrontEnd	I5 - 3.10GHz	8 GB
Nodo_A	I5 - 3.10GHz	4 GB
Nodo_B	I5 - 2.53GHz	4 GB

Fonte: Elaborado pelo autor

6.2 Aplicações

Nesta seção, são apresentadas as aplicações executadas nos experimentos, visando a avaliação do sistema de monitoramento. As aplicações foram desenvolvidas sob o modelo de programação paralela BSP, utilizando o middleware PUBWEB. A seguir, são apresentas as aplicações utilizadas nos experimentos.

- Fractal de Mandelbrot: O Fractal de Mandelbrot é definido como um conjunto matemático de pontos, cujos contornos formam um fractal bidimensional.
- Aplicação Sintética: Neste experimento, foi desenvolvida uma aplicação sintética, que simula um algoritmo BSP com grande demanda computacional. A ideia é explorar os aspectos relacionados à computação e troca de mensagens de aplicações paralelas entre recursos distintos, seja eles uma máquina física ou máquina virtual dentro da infraestrutura.

6.3 Intrusividade

De acordo (KAMOSHIDA; TAURA, 2008), a medição de intrusividade geralmente é feita através da coleta da quantidade de recursos que o agente utiliza durante o monitoramento. Isso acontece, devido ao fato do sistema de monitoramento compartilhar os mesmos recursos que a aplicação. De uma forma geral, sistemas de monitoramento, devem ser projetados visando não ser intrusivo para os recursos monitorados ou, se assim for, possuir baixa intrusividade. Na tabela 6, são apresentados os resultados dos experimentos relativos ao teste de intrusividade.

Tabela 6: Medição da Intrusividade

Item de Comparação	Fractal de Mandelbrot	Aplicação Sintética
Execução c/ Monitoramento Ativo	5:42	7:18
Execução c/ Monitoramento Inativo	5:40	7:15

Fonte: Elaborado pelo autor

Neste experimento, foi possível verificar que, em situações, onde a aplicação é executada com o monitoramento ativo, tem-se uma ligeira perda de desempenho, girando em torno de

0.58% para o Fractal de Mandelbrot. Já a intrusividade da aplicação sintética, a intrusividade detectada nos testes efetuados, fica em torno de 0.68%.

6.4 Avaliação do SLA

A avaliação do SLA tem como propósito, determinar o tempo de resposta/atuação do sistema de monitoramento em casos de falhas ou em situações onde existe a possibilidade de algum ganho dentro da infraestrutura, seja ele em relação à economia de energia ou realocando processos visando uma melhor distribuição, de forma que evite violações do SLA.

No experimento realizado, foi executada a aplicação do fractal de mandelbrot em duas máquinas físicas (Nodo A, Nodo B), cada qual com uma máquina virtual e cada uma delas com apenas um processo do middleware BSP para execução da aplicação. O fractal gerado é do tamanho de 2048x2048. Neste teste, o principal *threshold* avaliado será o consumo de memória. A tabela 7, apresenta o resultado da execução da aplicação em dois momentos distintos: No primeiro momento, de posse de um SLA configurado com *threshold* da memória da VM menor que 70%; No segundo momento, com o mesmo SLA ativo, porém, com as ações automáticas. Nestes dois cenários, a VM A possui 512mb, enquanto que a VM B, 2 GB de memória.

Tabela 7: Avaliação do Fractal de Mandelbrot

Item de Comparação	Threshold	Resultado
Fractal de Mandelbrot	VM Memory < 70%	Violação aos 2:33
Fractal de Mandelbrot (c/ ação automática)	VM Memory < 70%	Sem violação

Fonte: Elaborado pelo autor

Conforme apresentado nesta avaliação, no primeiro cenário, a aplicação executou normalmente, porem acabou violando um dos *thresholds* definido no SLA, ou seja, houve violação no acordo de nível de serviço definido. Já no segundo cenário, não ocorre violação, devido ao fato de ocorrer o disparo de ações automáticas, ou seja, a reorganização dos recursos ante cenário de falha, conforme os dados das predições, que já enviava alertas de potencial de violação a partir dos 2:15. A ação efetuada neste caso foi a migração, migrando o processo presente na VM A com pouca memória, para a VM B, que possui uma memória maior. Desta forma, não houve violação do SLA.

6.5 Avaliação Eficiência Energética

Nesta seção, é apresentada a avaliação do consumo de energia das aplicações testadas. O foco está em mensurar o quanto de consumo de energia é possível diminuir com o sistema de monitoramento proposto, se o mesmo, estiver trabalhando em conjunto de um escalonador de recursos.

Como ferramenta para o monitoramento do consumo de energia, foi utilizado o equipamento *Kill A Watt*. Neste tipo de ferramenta, são apresentadas as informações do consumo de energia dos equipamentos elétricos, como por exemplo: Amperes, Watts e kWh. Segundo a fabricante, a precisão do equipamento *Kill A Watt* é de 99,8%. A figura apresenta o equipamento Kill A Watt.

Figura 17: Medidor Kill A Watt

Fonte: (P3INTERNATIONAL, 2014)

Visando coletar métricas de desempenho e analisar o quanto de consumo elétrico é possível economizar. Inicialmente é preciso definir o consumo da máquina em modo de espera (ou *stand by*). Neste cenário, foi definido que o consumo em modo de espera, refere-se a casos onde a máquina física não possui máquinas virtuais instancias, ou seja, sem nenhum tipo de serviço em execução e pronta instanciar VM's visando a execução de algum serviço. Na tabela 8, são apresentados os consumos das máquinas físicas analisadas.

Tabela 8: Consumo em Standby

Máquina	Consumo em StandBy
Nodo A	70 Watts
Nodo B	70 Watts

Fonte: Elaborado pelo autor

No âmbito de testar a eficiência, foram elaborados dois cenários. No primeiro, a máquina Nodo A possui uma máquina virtual com memória de 1 GB. No segundo cenário, a máquina Nodo B possui 2 GB de memória disponível. A aplicação testada foi o fractal de mandelbrot, com tamanho definido em 2048x2048, cada VM executando apenas 1 processo. A tabela 9, apresenta os resultados desta execução.

Neste cenário, ambas as máquinas estavam com VM's fora do estado de espera, ou seja, instanciadas e prontas para receber um job. O consumo médio coletado do Nodo A durante a

Tabela 9: Consumo Standby x Consumo com VM Instanciada

Máquina	Consumo em StandBy	Consumo c/ VM Instanciada
Nodo A / VM A	70 Watts	105 Watts
Nodo B /VM B	70 Watts	108 Watts

Fonte: Elaborado pelo autor

execução da aplicação do fractal de mandelbrot foi de 105 Watts. Já o consumo médio no Nodo B, foi de 108 Watts. Em ambas as situações, as observações foram coletadas a cada 2 segundos, além de ambas as máquinas/processos estarem participando da mesma computação. No âmbito de testar o quanto que as predições podem auxiliar na economia de energia, foi executada a mesma aplicação, porem, com redução da memória da máquina virtual A (256 mb). Na tabela 10, é apresentada a relação consumo atual e consumo após efetuada uma ação automática.

Tabela 10: Consumo de energia: Resultados após migração

Máquina Física	Consumo	Consumo Pós-Migração
Nodo A	105 Watts	73 Watts
Nodo B	108 Watts	125 Watts

Fonte: Elaborado pelo autor

Neste experimento, a máquina virtual A teve sua memória RAM reduzida, visando gerar um cenário de violação de SLA. Com a ativação das ações automáticas, não ocorreu falha no acordo de nível de serviço, porém, houve uma migração de processo da VM A, presente no Nodo A, para a VM B presente no Nodo B. Neste contexto, foi possível verificar que após o processo migrado (VM A -> VM B), o consumo da máquina Nodo A, voltou a praticamente o estado do modo de espera. Porém, tendo dois processos para executar, o consumo da máquina Nodo B, aumentou para 125 Watts, ou seja, um acréscimo de 14% no consumo de energia desta máquina.

No experimento anterior, pode ser for avaliado, além do percentual de redução do consumo elétrico, como também, o quanto esse valor resultaria em economia de energia. Ao consultar portal da Aneel (Agência Nacional de Energia Elétrica), concessionária de distribuição RGE (Rio Grande Energia S/A), ano base 2014, tem-se a seguinte tarifa para 0,28478 R$/kWh. Se a aplicação, executar de forma continua durante 1 semana, tem-se o consumo acumulado de 21 kWh, resultando em 6 reais de custo para Máquina Nodo A, porem com uma economia em relação ao custo da indisponibilidade do serviço.

6.6 Avaliação da Predição

A avaliação da predição tem como objetivo, analisar situações onde o módulo SLA informou uma possibilidade de ganho e confrontar com o resultado real obtido, analisando se de fato

chegou-se ao resultado esperado e também, o percentual de acerto em casos de valor próximo do real. Foram analisadas 10 amostras das superetapas da aplicação BSP e cada novo valor no conjunto, foi elaborado uma série temporal e o modelo de previsão. A Tabela 11 apresenta uma amostra em cada superetapa, entre eles: o valor coletado, o valor previsto e a precisão da predição dos custos de computação do aplicativo BSP executado na Nuvem:

Tabela 11: Avaliação da Computação

Valor Coletado	Valor Previsto	Precisão
109	-	-
103	109	92,40%
102	105	96%
107	103	94%
101	104	95,94%
102	104	97,29%
104	103	98,64%
102	103	98,63%
101	103	97,26%
102	103	98,63%

Fonte: Elaborado pelo autor

A precisão média da previsão de um conjunto de 10 valores foi de 96,53%. De posse desta tabela, foi possível analisar que conforme aumenta o conjunto de valores históricos, mais precisa torna-se a predição de desempenho.

Em relação à avaliação dos custos de comunicação, a Tabela 12 apresenta o valor coletado, o valor previsto e a precisão da predição dos custos de computação do aplicativo BSP executado na Nuvem

Tabela 12: Avaliação da Comunicação

Valor Coletado	Valor Previsto	Precisão
1.17	-	-
1.28	1.17	91,40%
1.28	1.22	95,31%
1.25	1.24	99,2%
1.24	1.24	100%
1.31	1.24	94,65%
1.25	1.25	100%
1.27	1.25	98,42%
1.29	1.26	97,67%
1.26	1.26	100%

Fonte: Elaborado pelo autor

A precisão média da previsão de um conjunto de 10 valores de comunicação foi de 97,40%.

De posse desta tabela, reafirmando a análise anterior, foi possível observar que conforme o conjunto de valores históricos apesar de oscilações, tornou-se mais preciso conforme mais valores foram adicionados na série temporal.

O custo de sincronização, a Tabela 13 apresenta o valor coletado, o valor previsto e a precisão da predição dos custos de sincronização do aplicativo BSP executado na Nuvem.

Tabela 13: Avaliação da Sincronização

Valor Coletado	Valor Previsto	Precisão
1.72	-	-
1.78	1.72	96,62%
1.77	1.75	98,87%
1.80	1.76	97,77%
1.62	1.76	92,04%
1.76	1.73	98,29%
1.81	1.74	96,13%
1.79	1.75	97,76%
1.78	1.75	98,31%
1.82	1.76	96,70%

Fonte: Elaborado pelo autor

A precisão média da previsão de um conjunto de 10 valores coletados foi de 96,94%. De posse desta tabela, reafirmando a análise anterior, foi possível analisar que conforme o conjunto de valores históricos apesar de oscilações tornou-se mais preciso conforme mais valores foram adicionados na serie.

6.7 Considerações sobre o Capítulo

Neste capítulo, foi apresentada a avaliação do protótipo desenvolvido em relação ao modelo proposto. As aplicações utilizadas para avaliação foram: Fractal de Mandelbrot e uma aplicação sintética, que possui como única finalidade a execução de benchmark, cuja as operações foram: a execução de loops, cálculos matemáticos e trocas de mensagens, visando unicamente testar a predição e o consumo dos recursos computacionais.

Foi possível constatar, através dos experimentos, uma taxa de acerto igual ou superior a 90% na etapa de predição de desempenho dos recursos. Foi verificado também que o sistema de monitoramento possui baixa intrusividade.

Para comprovar que as predições de fato fazem sentido para um determinado contexto, foi desenvolvido um gerente, que recebe as notificações e aplica o reajuste nos recursos, conforme os valores futuros e a necessidade de manter o SLA. Nos experimentos, através de suas aplicações, foi possível constatar que as ações automáticas, quando acionadas pelo módulo de predição, reduziram o consumo de energia elétrica através da migração e consolidação de recursos.

7 CONSIDERAÇÕES FINAIS

Neste trabalho foram apresentados conceitos relativos à Computação em Nuvem, apresentando arquiteturas, camadas de serviço e modelos de implantação. Também foi apresentada a modelagem do protótipo, provas de conceito, bem como as tecnologias utilizadas para a avaliação do trabalho. Para tal, primeiramente efetuou-se uma revisão na literatura acerca de métodos de previsão utilizando dados coletados durante uma janela flutuante de tempo. Desta forma, foi utilizado as técnicas de series temporais para implementação do monitoramento preditivo.

Com este paradigma da computação, surgem novas necessidades e desafios, alguns já abordados em outros modelos de sistemas distribuídos, tais como Grades Computacionais. Entre as principais necessidades, está o correto gerenciamento dos recursos disponíveis dentro da infraestrutura de Nuvem, provendo a elasticidade necessária através de técnicas de provisionamento.

Porém, para que isso ocorra é necessário um constante monitoramento destes recursos computacionais, visando sempre comparar com os níveis de qualidade de serviço previstos e os acordos de SLA firmados entre o cliente final e o prestador de serviço. O sistema de monitoramento desenvolvido trabalhará em conjunto com o *middleware* OpenNebula, um gerenciador de infraestrutura de *Cloud Computing*.

Desta forma, o sistema de monitoramento tratado neste trabalho executará a coleta de métricas de desempenho em três níveis em aplicações de alto desempenho do tipo BSP: Nível de máquina física, máquina virtual e aplicação. Desta forma será possível um melhor controle sobre os recursos e assim, poder prover melhores informações para tomada de decisão de um administrador da infraestrutura ou escalonador externo, responsável pelo provisionamento e consolidação de recursos dentro da Nuvem. Além de um melhor controle sobre o SLA e utilização dos recursos através da utilização de padrões de uso para a predição de desempenho.

Um padrão de utilização é composto por uma ação cíclica, que possui tendência de ocorrer uma repetição durante um determinado período. Entre os benefícios de um sistema de monitoramento que utilize entre suas métricas o padrão de utilização está relacionado a um melhor controle e previsão sobre os recursos, podendo aumentar ou diminuir os recursos conforme a necessidade já previamente encontrada dentro de um padrão com base em informações históricas. Reduzindo desta forma o desperdício e assim diminuindo o consumo de energia e todas suas implicações.

7.1 Contribuições

As principais contribuições do sistema de monitoramento preditivo BSPMon concentram-se em monitoramento multinível e predição em aplicações paralelas do tipo BSP. Os itens listados apresentam as principais contribuições com a construção do sistema de monitoramento preditivo em *Cloud Computing*:

- Monitoramento multinível - Máquina física, virtual e aplicação;

- Sistema de monitoramento preditivo para aplicações de alto desempenho do tipo BSP;
- Análise / validação corrente e futura dos recursos sobre o SLA, analisando potenciais de violação, provisionamento e consolidação;
- Interface para integração com escalonadores, para que estes possam implementar as melhores políticas para a melhor organização e planejamento dos recursos presente na infraestrutura.

7.2 Trabalhos Futuros

Como ideias para trabalhos futuros:

- A combinação de mais de um método quantitativo para previsão de séries temporais. Alguns autores, tal como (GRANGER; RAMANATHAN, 1984), afirmam que a combinação de métodos para previsões de séries temporais pode superar as previsões individuais.
- Desenvolvimento de um módulo de tarifação integrado ao sistema de monitoramento, visando efetuar a geração dos custos a partir dos dados de utilização dos recursos, através de um conjunto de políticas de tarifação de custos, multas em falhas do SLA e crédito em casos de recursos subutilizados.
- Desenvolver um modulo de autogerenciamento, visando à construção de um sistema autonômico, desta forma, tornando-o capaz de efetuar auto-configuração, auto-proteção, auto-recuperação e auto-otimização dos recursos.
- Análise e predição de desempenho utilizando outros métodos estatísticos como Cadeias de Markov, Transformada Rápida de Fourier, além de técnicas de inteligência artificial (IA), tal como algoritmos genéticos, sistemas especialistas, redes neurais utilizando mapas SOM (Self-Organizing Maps).
- Desenvolvimento de um módulo de visualização.

REFERÊNCIAS

ACETO, G.; BOTTA, A.; DONATO, W. de; PESCAPÈ, A. Cloud Monitoring: definitions, issues and future directions. In: CLOUD NETWORKING (CLOUDNET), 2012 IEEE 1ST INTERNATIONAL CONFERENCE ON, 2012. **Anais...** [S.l.: s.n.], 2012. p. 63–67.

AMAZON. **Kill A Watt**. 2014.

BARHAM, P.; DRAGOVIC, B.; FRASER, K.; HAND, S.; HARRIS, T.; HO, A.; NEUGEBAUER, R.; PRATT, I.; WARFIELD, A. Xen and the art of virtualization. **ACM SIGOPS Operating Systems Review**, [S.l.], v. 37, n. 5, p. 164–177, 2003.

BONORDEN, O.; GEHWEILER, J.; HEIDE, F. M. auf der. Load balancing strategies in a web computing environment. In: **Parallel Processing and Applied Mathematics**. [S.l.]: Springer, 2006. p. 839–846.

BONORDEN, O.; GEHWEILER, J.; HEIDE, F. M. auf der. A web computing environment for parallel algorithms in java. In: **Parallel Processing and Applied Mathematics**. [S.l.]: Springer, 2006. p. 801–808.

BUNCH, C.; CHOHAN, N.; KRINTZ, C.; CHOHAN, J.; KUPFERMAN, J.; LAKHINA, P.; LI, Y.; NOMURA, Y. An evaluation of distributed datastores using the AppScale cloud platform. In: CLOUD COMPUTING (CLOUD), 2010 IEEE 3RD INTERNATIONAL CONFERENCE ON, 2010. **Anais...** [S.l.: s.n.], 2010. p. 305–312.

CASTILLO, J. A. L. d.; MALLICHAN, K.; AL-HAZMI, Y. OpenStack Federation in Experimentation Multi-cloud Testbeds. In: CLOUD COMPUTING TECHNOLOGY AND SCIENCE (CLOUDCOM), 2013 IEEE 5TH INTERNATIONAL CONFERENCE ON, 2013. **Anais...** [S.l.: s.n.], 2013. v. 2, p. 51–56.

CHATFIELD, C. **Time-series forecasting**. [S.l.]: CRC Press, 2002.

DE CHAVES, S.; URIARTE, R.; WESTPHALL, C. Toward an architecture for monitoring private clouds. **Communications Magazine, IEEE**, [S.l.], v. 49, n. 12, p. 130–137, december 2011.

ELMROTH, E.; MARQUEZ, F. G.; HENRIKSSON, D.; FERRERA, D. P. Accounting and billing for federated cloud infrastructures. In: GRID AND COOPERATIVE COMPUTING, 2009. GCC'09. EIGHTH INTERNATIONAL CONFERENCE ON, 2009. **Anais...** [S.l.: s.n.], 2009. p. 268–275.

GIACOMELLI, P. **HornetQ Messaging Developer? s Guide**. [S.l.]: Packt Publishing Ltd, 2012.

GRANGER, C. W.; RAMANATHAN, R. Improved methods of combining forecasts. **Journal of Forecasting**, [S.l.], v. 3, n. 2, p. 197–204, 1984.

ISSARIYAPAT, C.; PONGPAIBOOL, P.; MONGKOLLUKSAME, S.; MEESUBLAK, K. Using Nagios as a groundwork for developing a better network monitoring system. In: TECHNOLOGY MANAGEMENT FOR EMERGING TECHNOLOGIES (PICMET), 2012 PROCEEDINGS OF PICMET '12:, 2012. **Anais...** [S.l.: s.n.], 2012. p. 2771 –2777.

KAMOSHIDA, Y.; TAURA, K. Scalable data gathering for real-time monitoring systems on distributed computing. In: CLUSTER COMPUTING AND THE GRID, 2008. CCGRID'08. 8TH IEEE INTERNATIONAL SYMPOSIUM ON, 2008. **Anais...** [S.l.: s.n.], 2008. p. 425–432.

KELLER, A.; LUDWIG, H. The WSLA Framework: specifying and monitoring service level agreements for web services. **J. Netw. Syst. Manage.**, New York, NY, USA, v. 11, n. 1, p. 57–81, Mar. 2003.

KIRCHGÄSSNER, G.; WOLTERS, J.; HASSLER, U. **Introduction to modern time series analysis**. [S.l.]: Springer Verlag, 2012.

KIVITY, A.; KAMAY, Y.; LAOR, D.; LUBLIN, U.; LIGUORI, A. kvm: the linux virtual machine monitor. In: LINUX SYMPOSIUM, 2007. **Proceedings...** [S.l.: s.n.], 2007. v. 1, p. 225–230.

LAMANNA, D. D.; SKENE, J.; EMMERICH, W. SLAng: a language for defining service level agreements. In: THE NINTH IEEE WORKSHOP ON FUTURE TRENDS OF DISTRIBUTED COMPUTING SYSTEMS, 2003, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2003. p. 100–. (FTDCS '03).

MAKRIDAKIS, S.; WHEELWRIGHT, S.; HYNDMAN, R. **FORECASTING METHODS AND APPLICATIONS, 3RD ED**. [S.l.]: Wiley India Pvt. Limited, 2008.

MASSIE, M. L.; CHUN, B. N.; CULLER, D. E. The ganglia distributed monitoring system: design, implementation, and experience. **Parallel Computing**, [S.l.], v. 30, n. 7, p. 817–840, 2004.

MELL, P.; GRANCE, T. Effectively and securely using the cloud computing paradigm. **NIST, Information Technology Lab**, [S.l.], 2009.

MELL, P.; GRANCE, T. The NIST definition of cloud computing. **NIST special publication**, [S.l.], v. 800, p. 145, 2011.

NURMI, D.; WOLSKI, R.; GRZEGORCZYK, C.; OBERTELLI, G.; SOMAN, S.; YOUSEFF, L.; ZAGORODNOV, D. The eucalyptus open-source cloud-computing system. In: CLUSTER COMPUTING AND THE GRID, 2009. CCGRID'09. 9TH IEEE/ACM INTERNATIONAL SYMPOSIUM ON, 2009. **Anais...** [S.l.: s.n.], 2009. p. 124–131.

NURMI, D.; WOLSKI, R.; GRZEGORCZYK, C.; OBERTELLI, G.; SOMAN, S.; YOUSEFF, L.; ZAGORODNOV, D. The Eucalyptus Open-Source Cloud-Computing System. In: CLUSTER COMPUTING AND THE GRID, 2009. CCGRID '09. 9TH IEEE/ACM INTERNATIONAL SYMPOSIUM ON, 2009. **Anais...** [S.l.: s.n.], 2009. p. 124 –131.

OPENNEBULA. **OpenNebula**. 2014.

OPENSTACK. **OpenStack**. 2014.

P3INTERNATIONAL. **Kill A Watt**. 2014.

PASCHKE, A. RBSLA A declarative Rule-based Service Level Agreement Language based on RuleML. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE FOR MODELLING, CONTROL AND AUTOMATION AND

INTERNATIONAL CONFERENCE ON INTELLIGENT AGENTS, WEB TECHNOLOGIES AND INTERNET COMMERCE VOL-2 (CIMCA-IAWTIC'06) - VOLUME 02, 2005, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2005. p. 308–314. (CIMCA '05).

PINDYCK, R.; RUBINFELD, D. **Econometria**: modelos & previsões. [S.l.]: Elsevier, 2004.

POVEDANO-MOLINA, J.; LOPEZ-VEGA, J. M.; LOPEZ-SOLER, J. M.; CORRADI, A.; FOSCHINI, L. DARGOS: a highly adaptable and scalable monitoring architecture for multi-tenant clouds. **Future Generation Computer Systems**, [S.l.], 2013.

RUTA, D.; GABRYS, B.; LEMKE, C. A generic multilevel architecture for time series prediction. **Knowledge and Data Engineering, IEEE Transactions on**, [S.l.], v. 23, n. 3, p. 350–359, 2011.

SEFRAOUI, O.; AISSAOUI, M.; ELEULDJ, M. OpenStack: toward an open-source solution for cloud computing. **International Journal of Computer Applications**, [S.l.], v. 55, n. 3, p. 38–42, 2012.

SRIDEVI, S.; RAJARAM, S.; SWADHIKAR, C. An intelligent prediction system for time series data using periodic pattern mining in temporal databases. In: FIRST INTERNATIONAL CONFERENCE ON INTELLIGENT INTERACTIVE TECHNOLOGIES AND MULTIMEDIA, 2010, New York, NY, USA. **Proceedings...** ACM, 2010. p. 163–171. (IITM '10).

TORALDO, G. **OpenNebula 3 Cloud Computing**. [S.l.]: Packt Publishing Ltd, 2012.

URBANEK, S. Rserve–A Fast Way to Provide R Functionality to Applications. In: PROC. OF THE 3RD INTERNATIONAL WORKSHOP ON DISTRIBUTED STATISTICAL COMPUTING (DSC 2003), ISSN 1609-395X, EDS.: KURT HORNIK, FRIEDRICH LEISCH & ACHIM ZEILEIS, 2003 (HTTP://ROSUDA. ORG/RSERVE, 2003. **Anais...** [S.l.: s.n.], 2003.

VALIANT, L. G. A bridging model for parallel computation. **Commun. ACM**, New York, NY, USA, v. 33, n. 8, p. 103–111, Aug. 1990.

VáZQUEZ, C.; HUEDO, E.; MONTERO, R. S.; LLORENTE, I. M. On the use of clouds for grid resource provisioning. **Future Gener. Comput. Syst.**, Amsterdam, The Netherlands, The Netherlands, v. 27, p. 600–605, May 2011.

WANG, C.; SCHWAN, K.; TALWAR, V.; EISENHAUER, G.; HU, L.; WOLF, M. A Flexible Architecture Integrating Monitoring and Analytics for Managing Large-scale Data Centers. In: ACM INTERNATIONAL CONFERENCE ON AUTONOMIC COMPUTING, 8., 2011, New York, NY, USA. **Proceedings...** ACM, 2011. p. 141–150. (ICAC '11).

WEI, W. W.-S. **Time series analysis**. [S.l.]: Addison-Wesley publ, 1994.

WU, Y.-L.; AGRAWAL, D.; EL ABBADI, A. A comparison of DFT and DWT based similarity search in time-series databases. In: INFORMATION AND KNOWLEDGE MANAGEMENT, 2000, New York, NY, USA. **Proceedings...** ACM, 2000. p. 488–495. (CIKM '00).

WUHIB, F.; STADLER, R.; CLEMM, A. Decentralized service-level monitoring using network threshold crossing alerts. **Communications Magazine, IEEE**, [S.l.], v. 44, n. 10, p. 70–76, 2006.

WVWARE. **Hyperic SIGAR API**. 2013.

YANG, C.-T.; CHEN, T.-T.; CHEN, S.-Y. Implementation of Monitoring and Information Service Using Ganglia and NWS for Grid Resource Brokers. In: ASIA-PACIFIC SERVICE COMPUTING CONFERENCE, THE 2ND IEEE, 2007. **Anais...** [S.l.: s.n.], 2007. p. 356 –363.

YAZIR, Y.; AKBULUT, Y.; FARAHBOD, R.; GUITOUNI, A.; NEVILLE, S.; GANTI, S.; COADY, Y. Autonomous Resource Consolidation Management in Clouds Using IMPROMPTU Extensions. In: CLOUD COMPUTING (CLOUD), 2012 IEEE 5TH INTERNATIONAL CONFERENCE ON, 2012. **Anais...** [S.l.: s.n.], 2012. p. 614 –621.

ZHAO, H.; LI, X. Designing Flexible Resource Rental Models for Implementing HPC-as-a-Service in Cloud. In: PARALLEL AND DISTRIBUTED PROCESSING SYMPOSIUM WORKSHOPS PHD FORUM (IPDPSW), 2012 IEEE 26TH INTERNATIONAL, 2012. **Anais...** [S.l.: s.n.], 2012. p. 2550 –2553.