



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**UMA ARQUITETURA PARA GERENCIAR
AMBIENTES VIRTUALIZADOS BASEADA NOS
CONCEITOS DAS ÁREAS FUNCIONAIS DE GERÊNCIA**

ANDRIELE BUSATTO DO CARMO

Dissertação apresentada como
requisito parcial à obtenção do grau
de Mestre em Ciência da
Computação.

Orientador: Prof. Dr. César Augusto Fonticelha De Rose

Porto Alegre
2009

Dados Internacionais de Catalogação na Publicação (CIP)

C287a Carmo, Andriele Busatto do
Uma arquitetura para gerenciar ambientes virtualizados baseada nos conceitos das áreas funcionais de gerência / Andriele Busatto do Carmo. – Porto Alegre, 2009.
60 f.

Diss. (Mestrado) – Fac. de Informática, PUCRS.
Orientador: César Augusto FonticIELha De Rose

1. Informática. 2. Redes de Computadores – Gerência.
3. Máquinas Virtuais. 4. Arquitetura de Redes. I. De Rose, César Augusto FonticIELha. II. Título.

CDD 004.65

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**Uma Arquitetura para Gerenciar Ambientes Virtualizados Baseada nos Conceitos das Áreas Funcionais da Gerência**", apresentada por Andriele Busatto do Carmo, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Processamento Paralelo e Distribuído, aprovada em 27/01/09 pela Comissão Examinadora:

Prof. Dr. César Augusto FonticIELha De Rose -
Orientador

PPGCC/PUCRS

Prof. Dr. Avelino Francisco Zorzo -

PPGCC/PUCRS

Prof. Dr. Luciano Paschoal Gasparry -

UFRGS

Homologada em 27/04/10, conforme Ata No. 007 pela Comissão Coordenadora.

Prof. Dr. Fernando Gehm Moraes
Coordenador.

“O degrau de uma escada não serve simplesmente para que alguém permaneça em cima dele, destina-se a sustentar o pé de um homem pelo tempo suficiente para que ele coloque o outro um pouco mais alto.”

Thomas Henry Huxley

*Aos meus tão amados pais Jair (in memoriam)
e Ivete.*

Agradecimentos

À Deus, que através de minha fé permite que eu siga em busca dos meus sonhos. Pela vida que tenho, pelas pessoas que fazem parte da minha vida. Por tudo que já conquistei e pelo que ainda vou conquistar.

À minha família, principalmente meu pai e minha mãe. Por todo amor, carinho, incentivo e todo tempo que dedicaram à mim desde os primeiros anos de vida até os dias de hoje. À minha irmã, que mesmo longe sempre esteve perto no pensamento e no coração. Pelo apoio e por estar sempre ao meu lado eu agradeço. Amo vocês de todo coração!

Ao meu orientador César De Rose, pela oportunidade, disponibilidade, conselhos, compreensão e por ser um exemplo de orientador e professor. Por todo aprendizado que me foi proporcionado eu agradeço.

Ao meu namorado Ricardo, que a cada dia de convivência me proporciona momentos de felicidade plena. Por todo amor que me faz sentir e por tudo que representa na minha vida. Agradeço por ser minha razão de viver e por me fazer tão feliz.

Aos meus amigos de fé e de tantos anos: Juciane, Caroline, Márcia, Roberto, Rafael, Gabriele e Mateus. Pela amizade, pelos bons momentos e pelo carinho com o qual sempre fui recebida a cada volta à minha tão amada cidade natal. Pelos momentos felizes e por todo apoio que sempre recebi.

Aos amigos que fiz no Centro de Pesquisa em Alto Desempenho da HP: Rodrigo, Everton, Mauro, Guilherme, Élder, Roberto, Yves, Fedrizzi e Thiago, pelos momentos de estudo e de riso. Agradeço de coração esses dois anos de agradável convivência.

Aos meus novos amigos: Mateus, Thiago, Márcio, Alexandra e Alexandre. Agradeço pela amizade construída, pelo apoio e pela convivência nesses dois anos de mestrado. Pela amizade que vou levar pra vida.

Finalmente, agradeço à todas as pessoas que não mencionei, mas que foram importantes para meu desenvolvimento profissional e pessoal durante essa etapa de minha vida.

Resumo

A virtualização de ambientes computacionais tem cada vez mais atraído a atenção dos meios acadêmico e corporativo. Em virtude de sua utilização estar cada vez mais presente nesses meios, formas de gerenciar a complexidade que ambientes virtualizados apresentam tornam-se de grande importância, pois permitem controlar eventos no ambiente. Embora atividades de gerência sejam importantes, não há um padrão que auxilie neste processo de controle e monitoração de ambientes que se utilizam de virtualização. Uma forma de preencher esta lacuna é buscar, em outras áreas da computação, conceitos que dêem base e que auxiliem no processo de concepção de atividades gerenciais. Deste modo, é possível utilizar conceitos aplicados em diferentes áreas da computação para gerenciar ambientes virtualizados, conhecidos como áreas funcionais de gerência. Sendo assim, o presente trabalho consiste na concepção de uma arquitetura para gerenciar ambientes virtualizados, tendo como base as recomendações das áreas funcionais de gerência adaptadas para a realidade de virtualização.

Palavras-chave: Gerência de sistemas, monitoração, virtualização, áreas funcionais de gerência, FCAPS.

Abstract

Virtualization technologies are increasingly gaining the academic and corporate attention. Considering the fact that the use of virtualization is growing each day, it's very important to develop a mechanism to manage the complexity brought by virtual environments, because these will allow for control of the issues present in such environments. Although management activities are important, there is no standard to assist in the process of controlling and monitoring in virtual environments. One way to solve this problem is to find recommendations in other areas of computing, such as the functional areas of management. Therefore, this work consists in an architecture conception to manage virtual environments by means of an adaptation of the management model proposed by the functional areas of management.

Keywords: Systems management, monitoring, virtualization, functional areas of management, FCAPS.

Lista de Figuras

Figura 1	Virtualização a nível de <i>hardware</i>	18
Figura 2	Virtualização a nível de sistema operacional	18
Figura 3	Arquitetura para gerência de ambientes virtualizados	36
Figura 4	Ambiente para a arquitetura de gerência de ambientes virtualizados	40
Figura 5	Algoritmo correspondente ao serviço: Verificar o <i>status</i> das máquinas virtuais	41
Figura 6	Algoritmo correspondente ao serviço: Controlar as aplicações	42
Figura 7	Algoritmo correspondente ao serviço: Conhecer as configurações de rede	43
Figura 8	Algoritmo correspondente aos serviços: Manter <i>logs</i> & Contabilizar uso de recursos	44
Figura 9	<i>Framework</i> para Emulação de Sistemas Distribuídos	49
Figura 10	<i>Framework</i> para Emulação de Sistemas Distribuídos com módulo de gerência	50
Figura 11	Utilização de memória de uma máquina virtual antes e após o disparo do alarme	53
Figura 12	Número de máquinas virtuais no ambiente antes e após o disparo do alarme	53

Lista de Tabelas

Tabela 1	Características apresentadas pelas ferramentas de gerência estudadas . . .	34
----------	--	----

Lista de Siglas

CIM	<i>Common Information Model</i>
CPU	<i>Central Processing Unit</i>
DMTF	<i>Distributed Management Task Force</i>
FCAPS	<i>Fault, Configuration, Accounting, Performance, Security</i>
HMC	<i>IBM Hardware Management Console</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IP	<i>Internet Protocol</i>
ISO	<i>International Organization for Standardization</i>
JVM	<i>Java Virtual Machine</i>
ODBC	<i>Open Data Base Connectivity</i>
OSI	<i>Open Systems Interconnection</i>
POP3	<i>Post Office Protocol</i>
RAM	<i>Random Access Memory</i>
RSS	<i>Really Simple Syndication</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SNMP	<i>Simple Network Management Protocol</i>
USB	<i>Universal Serial Bus</i>
VMM	<i>Virtual Machine Monitor</i>
WBEM	<i>Web-Based Enterprise Management</i>
XML	<i>Extensible Markup Language</i>

Sumário

1	Introdução	14
2	Ambientes Virtualizados	16
2.1	Conceitos	16
2.1.1	Virtualização a Nível de <i>Hardware</i>	17
2.1.2	Virtualização a Nível de Sistema Operacional	18
2.1.3	Linguagem de Alto Nível de Máquinas Virtuais	19
2.2	Características de Máquinas Virtuais	19
2.3	Aplicações de Máquinas Virtuais	20
3	Gerência de Ambientes Computacionais	22
3.1	Gerência: Visão Geral	22
3.2	Gerência de Ambientes Virtualizados	23
3.3	Áreas Funcionais de Gerência	25
3.3.1	Gerência de Falhas	26
3.3.2	Gerência de Configuração	27
3.3.3	Gerência de Contabilidade	27
3.3.4	Gerência de Desempenho	28
3.3.5	Considerações sobre o Capítulo	28
4	Estudo sobre Ferramentas de Gerência para Ambientes Virtualizados	29
4.1	IBM Director	29
4.2	BixData	30
4.3	Enomalism	31
4.4	Citrix XenServer Express Edition	32
4.5	Cassatt	32
4.6	Considerações acerca das ferramentas estudadas	33
5	Arquitetura Proposta	35
5.1	Descrição da Arquitetura	35
5.2	Serviços	41
6	Implementação da Arquitetura e Testes Realizados	46
6.1	Tecnologia de Virtualização Xen	46
6.2	Arquitetura de Gerência WBEM	47
6.3	Ambiente de Testes	48
6.3.1	Estudo de caso: <i>Framework</i> para Emulação de Sistemas Distribuídos	48
6.3.2	Integração da arquitetura de gerência com os módulos do <i>framework</i>	50
6.3.3	Testes	52

6.3.4	Considerações a partir dos testes realizados	53
7	Considerações Finais	55
	Referências	57

1 Introdução

Nos últimos anos, a virtualização de ambientes computacionais vem conquistando espaço junto aos cenários acadêmico e corporativo. Embora o conceito de máquinas virtuais tenha surgido na década de 60 [1], o uso da virtualização em ambientes computacionais ganha destaque devido aos benefícios que pode proporcionar. Dentre os principais, pode-se observar o melhor aproveitamento do *hardware* e a possibilidade de integrar diferentes plataformas de sistemas operacionais [2].

Além disso, tecnologias de virtualização auxiliam na redução de custos, diminuindo a quantidade de *hardware* utilizado, proporcionando um ambiente mais enxuto e com os mesmos serviços fornecidos por ambientes que não utilizam-se desta tecnologia. Se antes havia a necessidade de possuir um parque computacional maior para ter servidores que trabalhem com diferentes propósitos, através das tecnologias de virtualização é possível que se tenha a consolidação destes servidores, onde estes, mesmo com diferentes objetivos podem executar suas atividades estando alocados na mesma estrutura física.

Frente a essas inovações tecnológicas, surge a necessidade de gerenciar os ambientes computacionais onde essas estão inseridas, proporcionando às pessoas envolvidas maior controle da utilização dos recursos tanto físicos quanto virtuais, além das atividades que são realizadas. Embora estas necessidades sejam claras, não há um padrão de gerência específico para ambientes virtualizados, levando os envolvidos neste processo ao interesse por padrões provenientes de outras áreas da computação.

Como forma de nortear as atividades gerenciais, são utilizadas as recomendações das áreas funcionais de gerência, as quais têm origem na área de redes de computadores, e são parte que compõe o modelo de referência OSI (do inglês, *Open Systems Interconnection*) [3]. As áreas funcionais de gerência, também conhecidas como FCAPS (do inglês, *Fault, Configuration, Accounting, Performance, Security*), apresentam recomendações para as gerências de falhas, configuração, contabilidade, desempenho e segurança.

Para cada uma dessas áreas as recomendações permitem a padronização de atividades gerenciais dentro do contexto em que estão inseridas. Para cada área funcional há um propósito a ser seguido pela unidade que utiliza-se de suas recomendações, fazendo com que as atividades de gerência sejam bem definidas e, com isso, possam dar maior sustentabilidade para monitorar e controlar os eventos do ambiente.

Para realizar a gerência (que compreende atividades de monitoração e controle [4]) de ambientes virtualizados, as recomendações das áreas funcionais de gerência, adaptadas para tal cenário, podem ser utilizadas para auxiliar na identificação dos módulos que deverão fazer parte

da arquitetura, definindo os serviços que irá prover, além de permitir que se tenha conhecimento das gerências que estão sendo realizadas no ambiente.

Embora existam ferramentas disponíveis para gerenciar ambientes físicos e virtuais, estas ferramentas apresentam limitações e, por isso, não atendem aos requisitos básicos identificados como necessários para realizar atividades de gerência.

Para os testes do presente trabalho será utilizado o ambiente em que o mesmo encontra-se inserido. Este ambiente consiste em um *framework* baseado em virtualização que realiza emulação de sistemas distribuídos, onde as máquinas virtuais estão hospedadas em máquinas reais, organizadas em um *cluster* de computadores [5].

Frente a este cenário, o presente trabalho consiste no desenvolvimento de uma arquitetura de gerência de ambientes virtualizados que segue as recomendações das áreas funcionais de gerência, mais especificamente das gerências de falhas, configuração, contabilidade e desempenho.

Além disso, a arquitetura proposta, através dos módulos que possui poderá ser utilizada em diferentes contextos que utilizam-se da tecnologia de virtualização, visto que o ambiente pode ser monitorado ou controlado independente de ser um *Data Center* ou um ambiente cujo objetivo seja emular ambientes distribuídos baseados em virtualização.

Desta forma, o objetivo do trabalho é prover uma arquitetura que siga uma padronização no âmbito gerencial e com isso possa servir como um modelo de atividades gerenciais a serem realizadas, mantendo ambientes virtualizados controlados e monitorados de acordo com as áreas funcionais estudadas.

Este trabalho está organizado da seguinte forma: o Capítulo 2 apresenta conceitos de virtualização, além da utilização, características e aplicações de máquinas virtuais. O Capítulo 3 apresenta uma visão geral sobre atividades gerenciais, a gerência como atividade direcionada para ambientes que utilizam-se da tecnologia de virtualização, e ainda aborda as recomendações das áreas funcionais de gerência do modelo de referência OSI. O Capítulo 4 apresenta as ferramentas de gerência de ambientes virtualizados existentes. O Capítulo 5 apresenta a arquitetura de trabalho proposta e o Capítulo 6 a descrição do ambiente de testes da arquitetura. Por fim, o Capítulo 7 apresenta as considerações finais e trabalhos futuros.

2 Ambientes Virtualizados

Embora o conceito de máquinas virtuais tenha surgido na década de 60 [1], atualmente é um assunto em evidência e que gera investimentos de grandes empresas como é o caso da HP [6], da IBM [7] e da AMD [8]. Este interesse é decorrente de características que ambientes virtualizados podem proporcionar, como a possibilidade de integrar diferentes sistemas operacionais, o melhor aproveitamento do *hardware* [2], e ainda maior flexibilidade de um sistema, pois é possível executá-lo em diferentes plataformas. Além disso, permite um melhor balanceamento de carga, disponibilidade e segurança, através de mecanismos como migração de máquinas virtuais entre diferentes máquinas físicas [2].

É objetivo deste capítulo apresentar alguns conceitos acerca de máquinas virtuais, suas características e aplicações.

2.1 Conceitos

Virtualização de ambientes computacionais consiste no processo de criação de um grupo ou subgrupo lógico de recursos, de modo que estes sejam acessados de forma a parecer o máximo possível com recursos físicos de um ambiente computacional convencional [9]. Neste contexto, as máquinas físicas são também chamadas máquinas hospedeiras (*host*), e as máquinas virtuais são chamadas de máquinas convidadas (*guest*).

Uma forma de visualizar as diferentes abstrações de máquinas virtuais é imaginá-las como se fossem fatias na pilha de *hardware/software*. Um sistema computacional é composto de camadas, iniciando com o *hardware* e incluindo camadas de sistema operacional e programas aplicativos que executam sobre o sistema operacional. O *software* de virtualização realiza a inserção de camadas em diferentes lugares do sistema, podendo-se destacar três exemplos destas camadas de virtualização, que incluem virtualização a nível de *hardware*, a nível de sistema operacional e ainda, a nível de linguagens de alto nível de máquinas virtuais [9].

Segundo Mendel Rosenblum [9] e Susanta Nanda et al. [10], a descrição para as virtualizações em vários níveis é dada como segue nas subseções do presente capítulo.

2.1.1 Virtualização a Nível de *Hardware*

No nível de *hardware* a camada de virtualização encontra-se sobre o *hardware* que exporta a abstração de máquina virtual. Neste caso, é explorada a similaridade entre as arquiteturas de plataformas *guest* e *host* para reduzir o custo de interpretação das instruções. Sendo assim, algumas instruções são repassadas diretamente ao *hardware* reduzindo o custo de interpretações e, com isso, o modelo acaba ganhando desempenho.

Para o computador há a possibilidade de operar por meio de dois modos: o modo supervisionado e o modo usuário. Neste sentido, no modo supervisionado qualquer instrução, seja ela privilegiada ou não pode ser executada. É neste modo que o VMM (do inglês, *Virtual Machine Monitor*) opera. Já no modo usuário somente as instruções não privilegiadas podem ser executadas, e neste modo as máquinas virtuais operam. Cabe ressaltar que uma arquitetura apresenta duas categorias para suas instruções: privilegiadas e não privilegiadas. As instruções privilegiadas alteram o estado do sistema, um pedido de I/O e um pedido de escrita na memória são alguns dos exemplos deste tipo de instrução. As instruções não privilegiadas não alteram o estado do sistema, neste caso a leitura de dados na memória, por exemplo, caracteriza-se uma instrução não privilegiada.

Desta forma, as instruções não privilegiadas poderão ser executadas diretamente no *hardware*, ao passo que as privilegiadas, quando uma máquina virtual necessitar executá-las, terão que gerar uma exceção para que o VMM capture-a e então emule a mesma.

Para tanto, para que uma tecnologia de virtualização funcione satisfatoriamente, a máquina virtual deve ser capaz de interromper a execução de uma instrução privilegiada, através de uma exceção, e passá-la ao VMM para que este cuide de sua execução. Isso porque no ambiente do VMM, múltiplas máquinas virtuais podem ter seu próprio sistema operacional rodando para executar instruções privilegiadas e ter acesso à CPU (do inglês, *Central Processing Unit*), por exemplo. Quando um pedido de execução de uma instrução privilegiada ocorre durante a execução de uma também instrução privilegiada, ao invés de gerar uma exceção e interromper a execução, a instrução é enviada ao VMM. Isso permite ao VMM ter o controle completo da máquina e manter cada máquina virtual isolada das demais. O VMM então, ou executa a instrução no processador, ou emula os resultados e retorna eles à máquina virtual. Contudo, a maioria das plataformas populares x86 não possuem virtualização completa, ou seja, uma máquina com instruções privilegiadas falha silenciosamente ao invés de provocar uma interrupção causada por uma exceção quando executada com privilégios insuficientes. Assim, a técnica de virtualização deve possuir formas de contornar situações como esta e passar o controle ao VMM quando a execução de uma instrução falhar. A maioria dos emuladores usam técnicas como "*code scanning*" e "*dynamic instruction rewriting*" para contornar tais situações.

A Figura 1 [10], ilustra a virtualização no nível da camada de abstração do *hardware*, onde a camada de virtualização encontra-se sobre este e as máquinas virtuais com seus sistemas

sobre essa camada. Alguns exemplos de virtualização a nível de *hardware* são VMWare [11], Xen [12], Virtual PC da Microsoft [13], dentre outros.

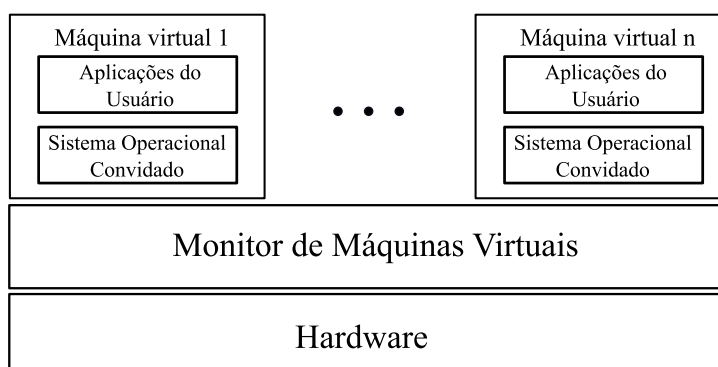


Figura 1: Virtualização a nível de *hardware*

2.1.2 Virtualização a Nível de Sistema Operacional

Virtualização a nível de sistema operacional tende a possuir propriedades como alto grau de isolamento (tanto das máquinas virtuais quanto das máquinas físicas subjacentes), suporte a diferentes sistemas operacionais e aplicações, baixo risco e fácil manutenção [10]. Nesse nível as máquinas virtuais compartilham o *hardware* bem como o sistema operacional na máquina física, utilizando a camada de virtualização no topo do sistema operacional para apresentar múltiplas máquinas independentes e isoladas para o usuário, como pode ser visto na Figura 2 [10].

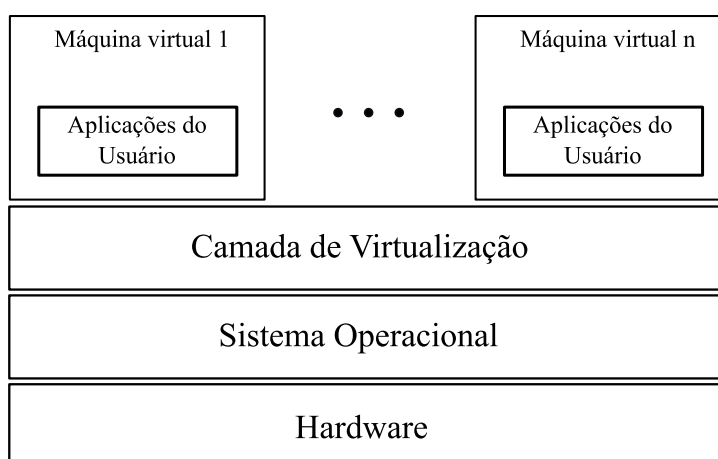


Figura 2: Virtualização a nível de sistema operacional

Um ambiente operacional é constituído por sistema operacional, bibliotecas de usuário, aplicações, sistemas específicos de estruturas de dados, além de outras configurações do ambiente. Se toda essa estrutura fosse conservada, o trabalho de uma aplicação em notificar qualquer diferença deste ambiente para um ambiente real se tornaria mais difícil. Essa é a idéia chave por trás

da técnica de virtualização a nível de sistema operacional, onde a camada de virtualização acima do sistema operacional possui uma partição visível por máquina virtual, funcionando como uma réplica do ambiente operacional da máquina física. Com um particionamento cuidadoso e uma técnica de transmissão, cada máquina virtual pode ser capaz de exportar um ambiente operacional completo e isolado das demais máquinas virtuais e também da máquina física. Alguns exemplos de virtualização a nível de sistema operacional são FreeBSD *Jails* [14], OpenVZ [15], dentre outros.

2.1.3 Linguagem de Alto Nível de Máquinas Virtuais

Neste nível, a camada de virtualização funciona como um programa aplicativo que executa sobre um sistema operacional. Sendo assim, a camada de virtualização encontra-se entre a execução da aplicação e o sistema operacional.

Neste modelo de virtualização a aplicação pode ser executada em qualquer sistema operacional, desde que este sistema tenha suporte para o virtualizador. Muito embora tais aplicações sejam extremamente portáteis, o desempenho acaba sendo um ponto negativo, uma vez que para cada instrução a ser executada na máquina virtual há uma instrução correspondente que será executada pelo virtualizador.

Na JVM (do inglês, *Java Virtual Machine*), por exemplo, a idéia é ter capacidade para criar uma máquina virtual no nível de aplicação, e que possa se comportar como uma máquina para um conjunto de aplicações, como qualquer outra máquina. Como uma máquina normal ela tem que ser capaz de fornecer um ambiente operacional para suas aplicações.

Como exemplos podem ser observados o Smalltalk [16] e o Java [17].

2.2 Características de Máquinas Virtuais

Segundo Mendel Rosenblum [9], as características apresentadas por máquinas virtuais são: compatibilidade de *software*, capacidade de isolamento, encapsulamento e desempenho. Tais características são apresentadas como segue:

Compatibilidade de *Software*. A máquina virtual fornece uma abstração de compatibilidade, onde todo *software* que for escrito para ela poderá ser executado. Por exemplo, uma máquina virtual a nível de *hardware* irá executar todos os *softwares*, sistemas operacionais e aplicações escritos para o seu *hardware*. Da mesma forma, uma máquina virtual a nível de sistema operacional executará aplicações para aquele sistema operacional em particular independente da arquitetura de *hardware*, e máquinas virtuais de alto nível executarão programas escritos em linguagem de alto nível independente do sistema operacional.

Capacidade de Isolamento. A abstração de máquina virtual isola o *software* que executa na máquina virtual de outras máquinas virtuais e máquinas reais. Este isolamento permite que mesmo na existência de *bugs* ou ataques, outras partes do sistema não sejam afetadas. Além do isolamento dos dados, a camada de virtualização pode fornecer isolamento do desempenho, onde os recursos consumidos pela máquina virtual não prejudicam o desempenho de outras máquinas virtuais.

Encapsulamento. A camada de *software* que exporta a abstração de máquina virtual é um exemplo do que é conhecido como nível de indireção. Esta camada pode ser usada para manipular e controlar a execução do *software* na máquina virtual. Além disso, ela pode também utilizar esta indireção para melhorar o *software* e para fornecer um ambiente de execução melhor. Por exemplo, máquinas virtuais para linguagens de alto nível tipicamente permitem verificações do sistema em tempo de execução, o que pode reduzir erros de programação. Isto inclui *type-safe* [18], *memory-safe* [18] e *garbage-collection* [18, 19]. Devido às características apresentadas, esta camada é a que oferece melhor ambiente de execução para programação.

Desempenho. As técnicas utilizadas pelos VMMs para mapear as máquinas virtuais com segurança diretamente no *hardware* da máquina real resultam em baixo desempenho para elas. O trabalho do VMM é realizar ajustes para que o *hardware* da máquina virtual seja mapeado diretamente nos recursos de *hardware* reais. É devido a estas duas definições de interfaces (virtual e real), que uma máquina virtual em execução pode apresentar desempenho no *hardware* real. Por exemplo, a CPU real é emulada pelo VMM programando a máquina virtual para executar uma CPU real. Da mesma forma, dispositivos de I/O como discos, por exemplo, são emulados usando discos reais.

O uso de virtualização de *hardware* resulta em um *overhead* ao VMM mensurado em uma pequena porcentagem. O *software* executando em uma máquina virtual gasta maior parte do tempo de execução usando os recursos de *hardware*; o desempenho é o mesmo do *hardware* abaixo dele. O *overhead* do VMM ocorre quando é necessário obter o controle, com o intuito de manter um isolamento seguro ou mesmo a ilusão de transparência. Estes acontecimentos ocorrem relativamente com pouca frequência na maioria dos *workloads*.

Em virtude das características apresentadas por máquinas virtuais, é possível identificar suas aplicações. Sendo assim, a Subseção 2.3 destina-se à descrição das mesmas.

2.3 Aplicações de Máquinas Virtuais

Segundo Smith e Nair [20], verifica-se que cenários de aplicações de máquinas virtuais incluem:

- **Consolidação de servidores:** Consiste na possibilidade de hospedar múltiplos servidores em um único *host* físico, apresentando desempenho e isolamento de falhas.

- Independência de *hardware*: Característica que permite a execução de aplicações e sistemas operacionais em *hardwares* diferentes.
- Configuração de múltiplos sistemas operacionais: Funcionalidade que possibilita a execução de múltiplos sistemas operacionais simultaneamente, tanto para desenvolvimento quanto para testes.
- Desenvolvimento do *kernel*: Consiste na possibilidade de realizar modificações no *kernel* do sistema operacional em uma máquina virtual, sem que haja necessidade de existir uma máquina separada para testes.
- Computação em *Cluster*: Consiste na gerência da granularidade de uma máquina virtual, o que proporciona maior flexibilidade do que gerenciar *hosts* físicos de forma separada. Além disso, possibilita melhor controle e isolamento se comparada a soluções de imagem de um único sistema, pois utiliza-se da *live migration*¹ para realizar o balanceamento de carga.
- Suporte de *hardware* para sistemas operacionais customizados: Característica que consiste na possibilidade de desenvolver novos sistemas operacionais, utilizando-se do suporte ao *hardware* variado de sistemas operacionais como o Linux.

Através dessas descrições, foi possível verificar as principais características apresentadas por ambientes virtualizados assim como suas aplicações. Além disso, é possível reconhecer os benefícios que ambientes como este proporcionam, instigando o interesse de empresas de grande porte.

Valendo-se da complexidade, da administração minuciosa e dos riscos que tecnologias de virtualização inferem a ambientes computacionais [22], é que identifica-se a necessidade de buscar a padronização a nível gerencial de uma arquitetura que provê monitoração e controle do ambiente.

¹Característica que permite a migração de máquinas virtuais entre *hosts* físicos sem que haja necessidade de parar a execução das mesmas [21].

3 Gerência de Ambientes Computacionais

Técnicas gerenciais eficientes são sempre bem-vindas quando há necessidade de manter o controle e a organização de um ambiente, especialmente quando este apresenta inúmeros componentes interligados e que exercem funções essenciais ao funcionamento satisfatório do ambiente como um todo. Por este motivo, o presente capítulo apresenta na Seção 3.1 uma visão geral sobre gerência, a gerência direcionada para ambientes virtualizados na Seção 3.2, na Seção 3.3 as áreas funcionais de gerência, principal influência para o presente trabalho e, por fim, na Seção 3.3.5 as considerações do presente capítulo.

3.1 Gerência: Visão Geral

Percebe-se atualmente nas organizações em geral, a necessidade de melhorar a qualidade de seus serviços e com isso ampliar suas capacidades e conquistar novos espaços. A busca pelo aperfeiçoamento de suas atividades e pela continuidade da qualidade nos serviços oferecidos, faz com que estas organizações desenvolvam capacidades de controle, monitoração e organização de todo o cenário em que se encontram, dando origem à chamada gerência.

A gerência de um modo geral trabalha ao lado de uma forte aliada, a informação. A necessidade de ter as informações sobre cenário a sofrer a gerência é de fundamental importância, pois permite que se tenha uma visão global de tudo que se passa no ambiente e de tudo que faz parte dele [23].

Muito mais do que possuir as informações é preciso saber manipulá-las e utilizá-las de maneira que as mesmas possam fazer a diferença dentro do ambiente em que se encontram. Todas as informações referentes ao universo do qual fazem parte são importantes para se julgue necessário ou não realizar o controle de determinados setores ou recursos [24].

Saber utilizar e manipular as informações que se possui torna-se a chave de todo o processo gerencial que se deseja instalar na organização. Saber o que fazer com o que se apresenta diante do cenário gerenciável é fator determinante para que o sucesso seja alcançado e, desta forma, o exercício da gerência torne-se o principal aliado de todo o processo organizacional [25].

Além disso, uma visão geral de tudo que está envolvido no ambiente auxilia na tomada de decisões de forma natural, sem a ocorrência de preocupações no que tange à possibilidade de que uma possível alteração na organização resulte em algum dano para a mesma. É preciso conhecer todo o ambiente para que as decisões sejam as mais acertadas possíveis.

Ao principal responsável pelas decisões tomadas designa-se o papel de gerente. Será ele o encarregado de realizar as mudanças necessárias com o intuito de controlar adversidades existentes dentro do cenário configurado. Para tal, explica-se a real importância de se conhecer tudo que possa estar relacionado direta ou indiretamente à organização.

O gerente então responsabiliza-se em atingir metas, como as que seguem [26]:

- Prover maior disponibilidade dos recursos existentes, realizando constante monitoração dos mesmos dentro do cenário em que se encontram;
- Reduzir os custos operacionais, pois como as tecnologias e ambientes apresentam crescente e acelerada evolução, necessita-se de recursos que possam amenizar este fator;
- Aumentar a flexibilidade de operação e integração, pois com a constante evolução das tecnologias é possível através da gerência realizar a integração dos recursos existentes;
- Aumentar a eficiência, fazendo com esta característica seja atingida no momento em que algumas das metas citadas anteriormente encontrem-se bem definidas dentro do ambiente.

Transportando os conceitos mencionados anteriormente para a área da informática, pode-se perfeitamente perceber que é possível colocá-los em prática e com isso manter um ambiente organizado e, desta forma, os recursos bem gerenciados.

Para tal, conta-se atualmente com a ajuda de ferramentas que permitem realizar a gerência dos recursos envolvidos no ambiente computacional, disponibilizando-os de acordo com as necessidades apresentadas e fazendo com que o uso dos mesmos permaneça dentro de um padrão aceitável melhorando a organização do ambiente.

3.2 Gerência de Ambientes Virtualizados

Analogamente às diversas áreas da computação, diferentes cenários que utilizam-se da tecnologia de virtualização necessitam de atividades gerenciais eficazes e que proporcionem maior disponibilidade dos serviços oferecidos, bem como do ambiente em geral. Para isso, é necessário que técnicas de monitoração e controle sejam utilizadas e tentem garantir o sucesso do funcionamento dos sistemas que compõem o ambiente.

Para a realização de atividades de gerência é preciso que o controle e a monitoração sejam atividades bem definidas e aplicadas. A monitoração diz respeito ao conhecimento que se tem do funcionamento dos recursos e dos serviços, buscando informações referentes à execuções, capacidades, disponibilidade ou mesmo problemas apresentados por eles. Com tais informações é possível realizar o controle de todas as ações realizadas durante este processo, podendo-se tomar decisões em decorrência de alguns acontecimentos.

Em virtude da quantidade de recursos a gerenciar em ambientes virtualizados, cada vez mais são desenvolvidos estudos e ferramentas para trabalhar com atividades desta natureza. Alguns

deles tratam recursos em específico, como por exemplo gerência de memória [27–29], e outros trabalham para ter uma visão global do ambiente, como algumas ferramentas que serão apresentadas no Capítulo 4. Mas o fato é que todos eles, por motivos específicos, identificam a importância de gerenciar ambientes que utilizam-se dos benefícios que as tecnologias de virtualização proporcionam.

Quando o objetivo da gerência é ter uma visão global do ambiente, considerando aspectos físicos e virtuais do mesmo, é necessário que a ferramenta utilizada preocupe-se minuciosamente com cada componente do cenário computacional. É preciso que informações de uso dos recursos sejam mantidas, que acontecimentos no ambiente sejam salvos em um arquivo de histórico (*log*) e que sejam utilizados para gerar eventos. Tais eventos podem ser utilizados para informar aos responsáveis pelo funcionamento do ambiente sobre a ocorrência de tudo que está relacionado a ele.

Para isso, a gerência está presente em diferentes níveis de abstração do ambiente computacional. Ou seja, vai desde o protocolo utilizado para acessar os recursos até o nível de decisão a ser tomada em função da monitoração realizada. Sendo assim, em um nível mais baixo encontram-se os padrões (como exemplo pode-se observar o SNMP (do inglês, *Simple Network Management Protocol*) [3] e o WBEM (do inglês, *Web-Based Enterprise Management*) [30], além de outros) que provêm meios de buscar diretamente nos recursos as informações solicitadas por níveis acima na organização, realizando uma gerência de mais baixo nível quando solicitadas ações em função do que foi observado. Em um nível intermediário encontra-se o mecanismo que faz o pedido de monitoração e controle da camada acima para o padrão de gerência utilizado, e que processa as informações por ele recebidas. As informações processadas são enfim enviadas para a camada acima, que toma as decisões necessárias em função do que foi monitorado. Em um sistema de gerência simples, podem ser percebidos os papéis e níveis aqui mencionados, mas vale ressaltar que essa organização pode ser percebida de diferentes formas dependendo do ponto de vista de quem está trabalhando no contexto da gerência.

Ferramentas utilizadas para gerenciar ambientes virtualizados apresentam em sua estrutura a combinação de todos os níveis mencionados anteriormente, pois todos eles são necessários quando o serviço de gerência vai além da coleta de informações do ambiente, ou seja, quando em um nível mais alto, há interesse pela tomada de decisões e medidas quando identificado algum comportamento inesperado ou indesejado do ambiente.

Tais ferramentas procuram atender ambientes virtualizados específicos, limitando ou a tecnologia de virtualização utilizada ou mesmo algum outro aspecto do cenário computacional. Mas o fato é que a grande maioria não busca a padronização dos serviços que provêm, justamente por não existir nada especificamente para esse contexto tecnológico.

Sendo assim, se faz necessário buscar em áreas da computação aspectos gerenciais para guiar as atividades a serem realizadas, permitindo que serviços e atividades sejam utilizados em diferentes contextos cuja virtualização esteja presente. Embora existam esses padrões é pre-

ciso estudá-los e adaptá-los para a realidade na qual serão inseridos, estruturando as atividades gerenciais em ambientes virtualizados.

3.3 Áreas Funcionais de Gerência

Com o intuito de padronizar o modo de interconexão de rede de diversos fabricantes de computadores é que a ISO (i. e. *International Organization for Standardization*) definiu um modelo de arquitetura para sistemas abertos, o Modelo de Referência OSI, visando permitir a comunicação entre máquinas heterogêneas e definir diretivas que possibilitem a construção de redes de computadores independente da tecnologia utilizada para a implementação [31].

Diante disso, o modelo OSI define uma arquitetura de sete camadas para prover interconectividade entre diferentes máquinas de um sistema computacional. Seguindo esta estrutura, com exceção da camada de mais alto nível, cada camada utiliza-se dos serviços prestados pelo nível imediatamente inferior, prestando serviço para o nível imediatamente superior [31, 32].

Desta forma, o Modelo de Referência OSI tem como objetivo maior prover uma base comum que permita o desenvolvimento de padrões para a interconexão de sistemas computacionais, adotando padrões para a troca de informações. Para tanto, o modelo OSI necessita manter informações dos recursos envolvidos no ambiente computacional gerenciadas, bem como ter claro como gerenciá-los. Sendo assim, são definidos alguns modelos gerenciais, são eles [33]:

- **Modelo Informacional:** Define os objetos de gerência, as relações e as operações sobre esses objetos.
- **Modelo Funcional:** Descreve funcionalidades de gerência através de cinco áreas, sendo elas gerência de falhas, gerência de configuração, gerência de contabilidade, gerência de desempenho e gerência de segurança, também conhecidas como FCAPS. Estas áreas apresentam recomendações que auxiliam no processo de entender “o que” deve ser gerenciado sem discutir detalhes de “como” essa gerência será implementada [34].
- **Modelo Organizacional:** Estabelece uma hierarquia entre os sistemas de gerência em um domínio, dividindo o ambiente a ser gerenciado em diversos domínios.

Embora sejam definidos os três modelos mencionados anteriormente, para o presente trabalho a descrição de apenas um deles torna-se indispensável, do Modelo Funcional, que é responsável pela definição das áreas funcionais de gerência dentro do ambiente computacional. Embora sejam cinco as áreas funcionais, optou-se por trabalhar com quatro delas, que são gerência de falhas, de configuração, de contabilidade e de desempenho. O fato de não ser trabalhada a gerência de segurança está associado à complexidade inferida por esta área, podendo ser estudada em um trabalho a parte. A descrição destas áreas funcionais está adaptada ao cenário

em que o presente trabalho se encontra, ou seja, as recomendações encontram-se adequadas a ambientes virtualizados e serão descritas nas subseções que seguem.

3.3.1 Gerência de Falhas

Para manter as operações de um sistema computacional complexo funcionando de forma satisfatória, alguns cuidados devem ser tomados em relação ao sistema como um todo, assim como com cada componente do mesmo para que trabalhem em ordem [3]. Quando uma falha ocorre, algumas observações de nível gerencial tornam-se relevantes, como por exemplo:

- Saber identificar exatamente onde a falha ocorreu.
- Avisar ao usuário e/ou ao gerente de infra-estrutura que um ou mais componentes apresentam problemas, dando a eles (usuário/gerente de infra-estrutura) a autonomia para decidir qual é o melhor procedimento a ser tomado.
- Se possível, tomar medidas de modo que o ambiente possa continuar seu funcionamento, estando de acordo com especificações prévias do usuário/gerente de infra-estrutura.
- Reparar ou substituir os componentes falhos para restaurar o ambiente em seu estado inicial.

Para a definição de gerência de falhas é fundamental entender o conceito de falhas. Falhas devem ser distinguidas de erros, pois uma falha é uma condição anormal que requer atenção na forma de gerência (ou ação) para ser reparada, ao passo que um erro é um evento único. Falhas normalmente são provenientes de defeitos ou erros excessivos [3]. Por exemplo, se uma linha de comunicação é fisicamente cortada, nenhum sinal será transmitido. Certos erros (por exemplo, um único *bit* errado na linha de comunicação) podem ocorrer ocasionalmente e normalmente não são considerados falhas. É possível compensar alguns erros através do uso de mecanismos de controle de erros de vários protocolos [3].

Sendo assim, usuários/gerentes de infra-estrutura esperam e contam com a resolução dos problemas. A maioria deles espera ser notificada tão logo a falha ocorra para que os problemas sejam resolvidos. Para que este nível de resolução de falhas seja alcançado, é necessário rapidez na detecção e diagnóstico da falha como função da gerência [3].

Além disso, os usuários/gerentes de infra-estrutura esperam ser informados do *status* dos componentes do ambiente, bem como serem notificados através de alertas sobre falhas que venham a ocorrer mantendo *logs* de execução e estatísticas para que sirvam de auxílio na análise do funcionamento do ambiente como um todo.

3.3.2 Gerência de Configuração

Ambientes computacionais são compostos de componentes individuais e subsistemas lógicos que podem ser configurados para trabalhar com diferentes aplicações [3]. No caso de ambientes virtualizados, uma máquina virtual pode ser configurada para trabalhar como um servidor de banco de dados *web* ou como um servidor de arquivos, por exemplo.

A gerência de configuração apresenta informações referentes à manutenção, adição e atualização das relações entre os componentes e o *status* destes componentes durante a execução de uma aplicação. Em ambientes virtualizados, leva-se em consideração também informações das aplicações que executam nas máquinas virtuais que compõem o ambiente.

Além disso, a gerência de configuração caracteriza-se por manter informações relacionadas à ligação existente entre os componentes do ambiente, sejam eles físicos ou lógicos. Desta forma, qualquer atualização nas relações existentes entre estes componentes serão do conhecimento do usuário e/ou do gerente de infra-estrutura. Estas informações podem ser disponibilizadas na forma de mapas de configuração, ou mesmo de relatórios gerados através de uma rotina básica ou em resposta a um pedido deste relatório.

3.3.3 Gerência de Contabilidade

A gerência de contabilidade, em sua essência, consiste no conhecimento do uso real dos recursos do ambiente computacional para que possam ser estimados valores a serem cobrados pelos serviços e infra-estrutura fornecidos aos usuários. No entanto, em alguns casos, onde serviços e infra-estrutura são oferecidos, não há necessidade de cobrar pelo seu uso, pois podem ser parte de uma organização onde os serviços a beneficiam. Neste caso então, a gerência de contabilidade acaba auxiliando no conhecimento da utilização dos recursos para que os usuários e ainda os gerentes de infra-estrutura saibam da real utilização dos componentes que fazem parte do ambiente.

Sendo assim, é importante que os usuários e gerentes de infra-estrutura saibam especificar os tipos de informações de contabilidade a serem guardados, o intervalo desejado para o envio de informações a serem gravadas, e caso seja necessário, os algoritmos a serem usados no cálculo da cobrança.

3.3.4 Gerência de Desempenho

Ambientes virtualizados são compostos de diversos componentes, os quais devem se intercomunicar e compartilhar dados e recursos. Em alguns casos, é muito difícil que uma aplicação esteja dentro de certos limites de desempenho esperados.

Sendo assim, a gerência de desempenho concilia duas grandes categorias funcionais - monitoração e controle. Monitoração é a função que rastreia as atividades realizadas no ambiente. A função de controle proporciona à gerência de desempenho a capacidade de realizar ajustes, com o objetivo de melhorar o desempenho da infra-estrutura lógica e física. Alguns tópicos de desempenho importantes para o conhecimento do usuário e do gerente de infra-estrutura são:

- Identificar qual é o nível da capacidade de utilização dos recursos.
- Verificar se há sobrecarga de utilização dos componentes do ambiente.
- Verificar se as configurações do ambiente estão refletindo no que era esperado pelo usuário e o gerente de infra-estrutura.

Gerentes de infra-estrutura e usuários necessitam de estatísticas de desempenho que os ajudem a planejar, gerenciar e manter o ambiente virtualizado. Estatísticas de desempenho podem ser usadas para reconhecer os gargalos potenciais antes mesmo de causarem problemas aos usuários. Ações de correções apropriadas podem então ser tomadas.

3.3.5 Considerações sobre o Capítulo

É com base nas recomendações das áreas funcionais de gerência descritas neste capítulo que a arquitetura proposta está calcada. Além da arquitetura, os serviços também sustentam tais recomendações adaptando-as para o cenário de virtualização. Isso se deve à flexibilidade fornecida pelas áreas funcionais de gerência, o que permite que sejam utilizadas em diferentes ambientes computacionais.

4 Estudo sobre Ferramentas de Gerência para Ambientes Virtualizados

Neste capítulo apresenta-se um estudo sobre as ferramentas existentes para gerenciar ambientes virtualizados. Tais ferramentas estão relacionadas a arquitetura proposta por apresentarem o mesmo objetivo, que é o de gerenciar ambientes que utilizam tecnologias de virtualização. Ao final deste capítulo serão apresentadas considerações acerca das ferramentas estudadas.

4.1 IBM Director

O IBM Director [35] consiste em um conjunto de ferramentas integradas que permitem realizar a gerência de ambientes físicos e virtuais. Cada uma das ferramentas integradas ao IBM Director é também identificada como uma extensão desta ferramenta, como é o caso do IBM Virtualization Manager, responsável pela monitoração e controle das atividades realizadas em ambientes virtuais e que, por este motivo, está fortemente relacionada ao trabalho proposto.

A proposta do IBM Director é gerenciar um ambiente que possua um grande número de servidores, computadores *desktop*, *workstations*, *notebooks*, além de diferentes dispositivos ligados ao ambiente [35]. Para gerenciar o ambiente que possui suporte à virtualização, o módulo IBM Virtualization Manager realiza atividades de monitoração e controle, permitindo desta forma a visualização do estado dos recursos, bem como o relacionamento entre os recursos físicos e virtuais. Dentre as funcionalidades disponíveis, o IBM Virtualization Manager permite criar, modificar e migrar servidores virtuais. Além disso, permite trabalhar com ambientes virtuais que são gerenciados pelo IBM Hardware Management Console (HMC), Microsoft Virtual Server [36], VMWare [11] e Xen [12].

Para que as atividades de gerência sejam realizadas, são oferecidos ao usuário alguns serviços, dentre eles pode-se identificar o mapa de configurações, que permite a visualização da relação existente entre os recursos físicos e virtuais, proporcionando o melhor entendimento do ambiente. É possível ainda, identificar no mapa de configurações pontos do ambiente que apresentam problemas, pois sempre que um componente apresentar uma falha, esta ficará visível através de uma marcação na visualização.

Além do mapa de configurações, há a possibilidade de identificar situações de alarme, onde há a ocorrência de falhas. Para isso, há um conjunto de opções disponíveis ao usuário para que ele possa escolher as situações de falha das quais deseja ser notificado. Além da escolha por

opções que melhor atendem sua necessidade, o usuário/gerente de infra-estrutura pode decidir a melhor maneira de receber estas notificações, se por *e-mail*, celular ou ainda notificação em tela.

4.2 BixData

BixData [37] é uma ferramenta que proporciona à ambientes de servidores e *clusters* a realização de atividades de gerência e monitoração, além de possibilitar a geração de relatórios a partir da gerência e monitoração realizadas [38]. BixData foi projetada especificamente para trabalhar em ambientes que apresentam diferentes sistemas operacionais, pois consegue gerenciar *clusters* que possuem de duas a milhares de máquinas executando qualquer combinação dos sistemas operacionais Linux e Windows.

Dentre os serviços oferecidos pela ferramenta BixData, pode-se identificar:

- Monitoração de aplicações.
- Monitoração de desempenho.
- Monitoração de rede.
- Monitoração de *hardware*.
- Armazenamento operacional de dados e geração de relatórios.
- Gerência de sistemas.

Além disso, é possível através da BixData gerenciar ambientes virtualizados que trabalham com as tecnologias de virtualização VMware [11] e Xen [12]. É possível gerenciar as máquinas *host* (hospedeiras) e as máquinas *guest* (convidadas) que utilizam estas tecnologias de virtualização.

Para isso, a ferramenta BixData possui três componentes integrados que juntos realizam as atividades de gerência, são eles:

- BixAgent: Pode ser instalado nas máquinas em que há requisições de dados e também de gerência remota. Possui suporte nativo aos sistemas operacionais Linux e Windows.
- BixServer: Tem a capacidade de monitorar serviços e dispositivos de rede, além de funcionar como um meio de armazenamento operacional de dados. Permite ainda, monitorar serviços HTTP, SMTP, POP3, SNMP e outros sem requisição do BixAgent. O armazenamento operacional de dados suporta vários servidores SQL para armazenamento, tais como MySQL, PostgreSQL, MS SQL Server e ODBC. Além disso, BixServer pode gerenciar e coletar dados dos BixAgents executando em qualquer plataforma.

- BixDesktop: Permite soluções em tempo-real para manipulação de dados. Pode executar em plataformas Linux e Windows.

BixData permite ainda que seja monitorado o desempenho das aplicações, e que estas informações sejam acompanhadas em tempo-real através de gráficos que realizam a representação dos dados. É possível também visualizar os dados de configuração dos componentes da rede através do mapa de configurações, verificando a relação existente entre eles. Outra característica presente é a possibilidade de armazenar os *logs* dos acontecimentos no ambiente em que a gerência está sendo realizada.

Observa-se também a possibilidade de gerenciar o ambiente através de atividades de caráter operacional, como por exemplo, iniciar, desligar, pausar e ainda desligar remotamente *guests* VMware [11] e Xen [12]. BixData apresenta-se em duas versões, a primeira delas *free*, com algumas restrições de uso e a segunda uma versão paga com todas as funcionalidades mencionadas sem restrições de utilização.

4.3 Enomalism

Enomalism [39, 40], assim como as demais, é uma ferramenta de gerência de ambientes virtualizados que foca na realização de atividades de cunho operacional, ou seja, atividades que estejam diretamente relacionadas ao funcionamento do ambiente. Atividades que envolvem a monitoração do ambiente não fazem parte das atividades principais da ferramenta.

Com isso, Enomalism permite criar, iniciar, pausar, reiniciar e destruir máquinas virtuais, além de permitir criar, editar e desfazer conexões de rede. Enomalism também permite atualizar pacotes de aplicações de *software* em várias máquinas virtuais através de um processo chamado *VMCasting*¹. Além disso, Enomalism permite migrar máquinas virtuais, configurar quem pode gerenciar e usar domínios específicos, dentre outras funcionalidades.

Neste sentido, a monitoração concentra-se no conhecimento do *status* dos recursos que fazem parte do ambiente. É através do *status* dos recursos que se tem o conhecimento de quais deles apresentaram falhas ou que seu funcionamento permanece em bom estado.

Outra característica presente nesta ferramenta está relacionada às aplicações que executam no ambiente computacional, pois estas são monitoradas em relação a seu *status* também, tornando-se possível ter o conhecimento do estado em que se encontram, se em funcionamento ou não.

A ferramenta Enomalism não monitora o ambiente através do uso de alarmes sempre que um evento diferente do que foi proposto ocorrer, somente através do *status* é possível identificar quando uma aplicação apresentou problema ou algo não esperado ocorreu.

¹Processo que permite atualização de pacotes para que sejam distribuídos via RSS *feed* [41] para as máquinas virtuais [42].

Desta forma, a arquitetura da ferramenta Enomalism trabalha principalmente questões relacionadas ao funcionamento do ambiente e de que forma algumas configurações podem ser realizadas em função de observações a cerca do ambiente.

4.4 Citrix XenServer Express Edition

XenServer Express Edition [43] é uma ferramenta *free* que compõe o conjunto de ferramentas da família XenServer, agora pertencente à empresa Citrix [44]. Esta ferramenta é utilizada para realizar a gerência de ambientes virtualizados que utilizam-se da tecnologia Xen [12] em ambientes de servidores ou que utilizam *cluster computing*.

A gerência proporcionada pela XenServer Express Edition está diretamente ligada a questões de monitoração e controle do ambiente virtualizado. As funções proporcionadas pela XenServer Express Edition são: criação, inicialização, parada, reinicialização, resumo, migração e destruição de máquinas virtuais, além de permitir reinicializar e desligar servidores físicos. Além disso, existem administradores de monitoração de desempenho, o que possibilita o acesso em tempo-real de informações de desempenho das máquinas virtuais e métricas de desempenho para processador, memória, disco e I/O da rede.

XenServer Express Edition permite ainda configurar prioridades e limites para uso de processador, memória, disco e I/O de rede. Sendo assim, é possível acompanhar o uso dos recursos do ambiente no momento em que aplicações estão sendo executadas.

Pode-se então inferir que a gerência proporcionada pela ferramenta XenServer Express Edition permite a monitoração dos recursos físicos e virtuais, embora não possua mecanismos de alarmes para que o usuário e o gerente de infra-estrutura sejam notificados de eventuais problemas. A gerência desta ferramenta encontra-se no campo de controle do funcionamento do ambiente e no cuidado com a infra-estrutura existente para execução das aplicações que fazem parte do cenário computacional.

4.5 Cassatt

Cassatt é um produto de *software* que possui quatro edições diferentes para realizar atividades em ambientes de servidores e *cluster computing*. Em uma destas edições há a possibilidade de gerenciar ambientes virtualizados [22] que fazem uso das tecnologias Xen [12] e VMWare [11].

Seu sistema de gerência pode ser percebido através de ações de gerência de falhas, a partir da monitoração de acontecimentos nos ambientes físicos e virtuais. Em casos de falha de

algum recurso, são adotadas ações que permitem o funcionamento do sistema, realizando as substituições necessárias dos recursos que apresentaram este comportamento.

Não é utilizada a política de alarme, mas todas as ações tomadas são previamente definidas pelo usuário, tendo a participação do mesmo na escolha pelo melhor caminho a ser seguido no momento em que uma falha ocorrer. O controle de uso dos recursos e a verificação do modo como estão sendo utilizados não faz parte das atribuições de gerência desta ferramenta. Seu foco encontra-se justamente no funcionamento do ambiente e em mantê-lo desta forma.

4.6 Considerações acerca das ferramentas estudadas

Antes de iniciar as considerações sobre as ferramentas estudadas, é importante salientar que a escolha por desenvolver uma arquitetura de gerência e não uma ferramenta, está relacionada ao fato de que o objetivo do trabalho é prover um padrão de atividades gerenciais, de forma que estas estejam apoiadas em recomendações já aplicadas em outras áreas da computação. Com isso, é possível que esta arquitetura padrão tenha flexibilidade para ser utilizada em diferentes ambientes, cuja ferramenta principal sejam as tecnologias de virtualização.

Levando em consideração o estudo sobre as características e funcionalidades das ferramentas estudadas, pôde-se identificar algumas limitações nas mesmas. Iniciando pela ferramenta da IBM, embora busque também nas FCAPS a padronização necessária para trabalhar com atividades de gerência [45] e, além disso, ofereça serviços relevantes que proporcionam o controle do ambiente, é uma ferramenta que não apresenta fácil adaptação aos diferentes cenários que utilizam virtualização. Caracteriza-se como uma ferramenta para o meio corporativo, não oferecendo muitas condições para que o meio acadêmico possa utilizá-la em diferentes contextos.

Outra ferramenta estudada é a BixData, cujas limitações identificadas são a falta de padronização para os serviços de gerência oferecidos, e ainda limitações referentes à adaptação de seus serviços a outros cenários de virtualização. Além disso, esta ferramenta restringe as tecnologias de virtualização a serem utilizadas, possuindo uma interface compatível apenas com o VMware [11] e o Xen [12].

A terceira ferramenta estudada foi a Enomalism, que se direciona para uma gerência mais operacional, ou seja, trabalha questões relacionadas ao funcionamento do ambiente e em como sua estrutura será montada, não havendo preocupações com o tratamento e o retorno de informações aos usuários e gerentes de infra-estrutura. Por este motivo, não apresenta as mesmas preocupações que a arquitetura proposta.

Outra ferramenta é a Citrix XenServer Express Edition, que além de atividades operacionais como a Enomalism, possui funcionalidades relativas a monitoração do ambiente, como verificar a taxa de uso de CPU, memória livre e informações dos demais recursos através de gráficos fornecidos em tempo de execução das aplicações. Embora apresente tais características, está limitada ao uso com a tecnologia de virtualização Xen [12].

Por fim, a quinta ferramenta estudada foi a Cassatt, cujo ponto fraco é caracterizado pela ausência de funcionalidades de notificação ao usuário, ou mesmo ao gerente de infra-estrutura a respeito de algum evento inesperado. Desta forma, não há possibilidade de, através de alarmes, informar aos responsáveis pelo funcionamento do sistema problemas ou mesmo comportamentos inesperados. Além disso, não utiliza-se de padrões para prover seus serviços.

Na Tabela 1, constam as principais características/funcionalidades identificadas como necessárias a uma ferramenta que realiza gerência de ambientes virtualizados. Estas características/funcionalidades são verificadas quanto à sua presença/ausência nas ferramentas estudadas. Cabe ressaltar que os traços indicam a ausência da característica/funcionalidade em questão, e o sinal “√” a presença.

Tabela 1: Características apresentadas pelas ferramentas de gerência estudadas

Desvantagens / Ferramenta	IBM Director	Enomalism	BixData	Citrix XenServer	Cassatt
Fácil adaptação a diferentes ambientes computacionais	-	-	-	-	-
Restrito a tecnologias de virtualização específicas	√	√	√	√	√
Segue padrões de gerência	√	-	-	-	-
Possui essencialmente atividades de controle	-	√	-	-	-

Sendo assim, é possível verificar que embora existam ferramentas que se proponham a gerenciar ambientes virtualizados, nenhuma delas atende a todos os requisitos identificados como necessários à uma arquitetura com essa finalidade. Tais requisitos serão descritos no Capítulo 5, e servirão como base para que a arquitetura estruture-se de forma simples e modular, uma vez que para cada área funcional de gerência, será pensada a estrutura que poderá melhor flexibilizar a adição de novos serviços.

5 Arquitetura Proposta

A crescente utilização de tecnologias de virtualização, tanto em ambientes acadêmicos, quanto em ambientes corporativos pode ser explicada pelos benefícios que tais tecnologias oferecem à ambientes que a utilizam. No entanto, tais benefícios trazem consigo um grau de complexidade maior se comparado a ambientes computacionais convencionais, o que exige atividades gerenciais bem definidas.

A necessidade de gerenciar ambientes virtualizados é bastante significativa, e embora existam ferramentas que trabalhem questões gerenciais para ambientes como estes, não há um padrão que norteie as atividades gerenciais necessárias ao bom funcionamento do mesmo.

Desta forma, o presente trabalho tenta solucionar essas questões através da concepção de uma arquitetura de gerência de ambientes virtualizados, baseando-se nas recomendações das áreas funcionais de gerência, com o intuito de nortear e dar base à estrutura de controle e monitoração criada. O presente capítulo está estruturado da seguinte forma: a Seção 5.1 apresenta a descrição da arquitetura de gerência proposta, detalhando cada componente da mesma. E, por fim, a Seção 5.2 os serviços que estão atrelados à arquitetura, juntamente com seus algoritmos representados através de diagramas de atividades.

5.1 Descrição da Arquitetura

A arquitetura apresenta uma estrutura modular, ou seja, possui módulos diferentes pensados para acomodar cada uma das áreas funcionais de gerência e atender à suas recomendações, como pode ser observado na Figura 3. Vale lembrar que os módulos que refletem os conceitos das FCAPS apresentam uma estrutura maior que leva o nome da área funcional correspondente. Os demais módulos responsabilizam-se pela comunicação com as partes envolvidas na arquitetura, além de poder realizar processamentos, como é o caso do *Refinamento de Dados*, e que será explicado a seguir. A busca por conceitos de outras áreas para dar base à arquitetura proposta, está fortemente ligada ao fato de que através da modularidade proporcionada é possível que sejam identificados serviços a serem oferecidos em cada um dos módulos, inferindo maior facilidade quando um novo serviço deve ser acrescentado à arquitetura.

Para o maior entendimento da Figura 3, serão descritos a seguir cada um dos módulos e papéis que compõe a arquitetura. Sendo assim, são eles:

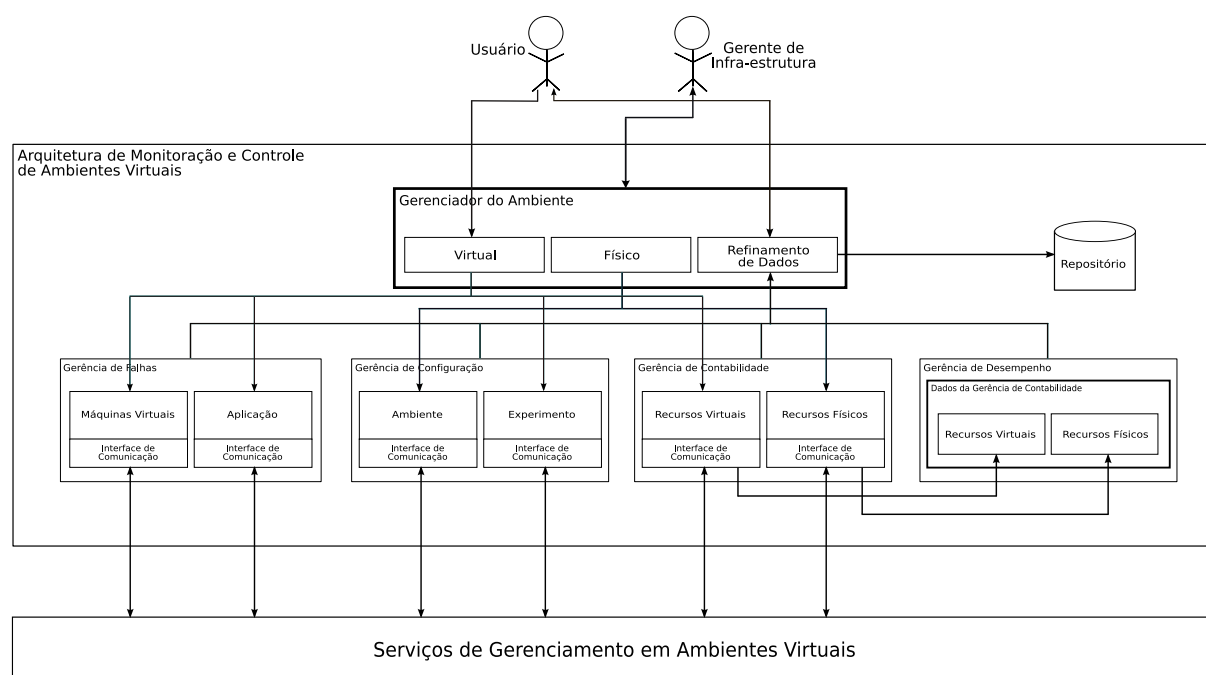


Figura 3: Arquitetura para gerência de ambientes virtualizados

- **Usuário:** Este papel é destinado à pessoa ou mesmo ao sistema que deseja utilizar o ambiente, mas cuja infra-estrutura virtual lhe é transparente. Ou seja, é possível realizar a configuração do ambiente e das aplicações, decidindo a quantidade de máquinas a serem utilizadas e ainda, questões relacionadas à configuração das aplicações sem que o ambiente virtualizado fique visível ao usuário. Este papel tem acesso aos demais módulos através da comunicação com o módulo *Virtual* da estrutura *Gerenciador do Ambiente*. Através desse módulo são solicitadas as informações necessárias ao usuário. Cabe ressaltar que embora o usuário tenha acesso ao módulo virtual, esta estrutura lhe é transparente, ou seja, mesmo acessando o módulo o usuário tem a impressão de estar configurando um ambiente físico convencional.
- **Gerente de Infra-estrutura:** Este papel destina-se ao responsável pela gerência do ambiente como um todo, tanto do físico quanto do virtual. O gerente de infra-estrutura tem o conhecimento de tudo que está relacionado aos dois ambientes, tendo autonomia para realizar as modificações necessárias tanto no ambiente físico quanto no virtual. Este papel tem acesso aos módulos *Virtual* e *Físico* da estrutura *Gerenciador do Ambiente*.
- **Gerenciador do Ambiente:** Esta estrutura é responsável por realizar a comunicação do usuário e do gerente de infra-estrutura com os demais módulos da arquitetura. Além disso, possui um módulo responsável por realizar o tratamento dos dados monitorados durante e ao término da execução das aplicações. Sua estrutura é composta pelos módulos *Virtual*, *Físico* e *Refinamento de Dados*, e serão descritas como segue:

Virtual: Este módulo fica responsável por realizar a comunicação do usuário e do gerente de infra-estrutura com os demais módulos da arquitetura, módulos estes que realizam o controle e a monitoração de questões relacionadas apenas ao ambiente virtual. Pode-se mencionar, dentre outros exemplos, a criação do ambiente virtual com a especificação das aplicações que farão parte dele.

Físico: Este módulo é responsável por realizar a comunicação apenas do gerente de infra-estrutura com os demais módulos da arquitetura, módulos estes que realizam o controle e a monitoração do comportamento do ambiente físico. Pode-se mencionar, dentre outros exemplos, algum problema apresentado por alguma das máquinas físicas do parque computacional.

Refinamento de Dados: Este módulo é responsável pelo tratamento dos dados obtidos através da gerência dos ambientes físico e virtual. Este tratamento inclui a notificação de situações adversas através de alarmes, a construção de gráficos a partir de dados coletados para verificação de uso dos recursos e ainda, para avaliação do desempenho dos mesmos. É possível ainda, construir o mapa de configurações do ambiente, apresentando a relação existente entre os ambientes físico e virtual, além da ligação dos componentes de cada um destes ambientes. As informações são recebidas pelos módulos responsáveis pelas atividades de gerência. Além destas atividades, este módulo realiza a comunicação com o *Repositório* da arquitetura, gravando os dados coletados durante o processo de gerência, assim como os refinamentos realizados.

- **Gerência de Falhas:** Antes de mais nada, vale destacar que falha no contexto em que a arquitetura se encontra corresponde ao “desrespeito” à alguma configuração previamente realizada pelo usuário ou gerente de infra-estrutura, ou ainda, quando os mesmos solicitam a notificação de determinado comportamento, ou, por fim, quando um serviço e/ou componente tem seu funcionamento interrompido de modo inesperado. Esta estrutura é responsável pela monitoração dos ambientes físico e virtual, e está dividida em dois módulos:

Máquinas Virtuais: Este módulo monitora informações referentes às máquinas virtuais durante a execução das aplicações, verificando possíveis alterações em seu *status*. Sempre que houver modificações nestas configurações, é gerada uma notificação e enviada ao módulo *Refinamento de Dados* para que o usuário e o gerente de infra-estrutura sejam informados através de alarmes.

Aplicação: Neste módulo as informações monitoradas estão diretamente relacionadas às aplicações que estão sendo executadas nas máquinas virtuais. É verificado o funcionamento das aplicações, além de informar ao usuário e ao gerente de infra-estrutura, através do módulo *Refinamento de Dados*, situações em que estas aplicações apresentaram problemas. Como exemplo pode-se verificar a execução interrompida das aplicações sem que essa ação tenha sido previamente configurada.

- **Gerência de Configuração:** Esta estrutura é composta pelos módulos *Ambiente* e *Experimento*, que são responsáveis pela configuração do ambiente virtual e das aplicações. Outra atribuição desta estrutura é realizar a monitoração referente aos ambientes físico e virtual, e às aplicações através dos módulos que a compõe. Neste caso, a monitoração está relacionada ao funcionamento do ambiente, quando recursos físicos e/ou virtuais apresentam problemas ou ainda quando o usuário/gerente de infra-estrutura deseja ser notificado de algo em específico, por exemplo. Para isso, são utilizados alarmes. Este módulo pode também realizar a monitoração do uso dos recursos e gerar alarmes quando estes tiverem sua capacidade total de processamento excedida ou subutilizada (CPU e memória são exemplos de recursos). Embora exista a possibilidade de monitorar através do uso de *logs*, isso só é possível quando alarmes também são solicitados, utilizando as informações dos *logs* para tal, caso contrário os *logs* ficam à cargo da estrutura *Gerência de Contabilidade*. Vale ressaltar que esta estrutura apresenta um repositório local, que corresponde ao espaço do disco rígido reservado a cada máquina virtual. Este repositório é responsável pelo armazenamento de todas as informações referentes ao comportamento do ambiente físico e virtual no decorrer da execução das aplicações. Ao final, todas estas informações são encaminhadas ao repositório geral através da comunicação com o módulo *Refinamento de Dados*. Assim são gravados os *logs* de execução das aplicações, bem como o comportamento geral do ambiente durante este processo. A presente estrutura possui os seguintes módulos:

Ambiente: Módulo responsável pela configuração do ambiente virtual, além de monitorá-lo, enviando ao módulo *Refinamento de Dados*, através de alarmes, as informações referentes às configurações que não apresentaram o comportamento esperado.

Experimento: Módulo responsável pela configuração das aplicações que estão sendo executadas no ambiente virtual. Este módulo não realiza monitoração das aplicações que estão sendo executadas no ambiente, isso porque a estrutura *Gerência de Falhas* através do módulo *Aplicações* já o faz.

- **Gerência de Contabilidade:** Estrutura que realiza a contabilização do uso dos recursos físicos e virtuais através dos módulos *Recursos Virtuais* e *Recursos Físicos*. Esta estrutura também possui um repositório local, que corresponde ao espaço do disco rígido reservado a cada máquina virtual. Este repositório armazena todas as informações referentes ao uso dos recursos no decorrer da execução de uma aplicação. Ao final deste processo, as informações monitoradas são enviadas ao módulo *Refinamento de Dados*. A presente estrutura é composta por:

Recursos Virtuais: Este módulo também realiza a monitoração do uso dos recursos virtuais. A diferença entre este e o módulo *Ambiente* é que neste módulo não são gerados alarmes quando um recurso excede ou subutiliza sua capacidade total. Fica a seu critério apenas o armazenamento dos *logs* de uso dos recursos. Para tal, as informações monito-

radas e que são relativas ao uso dos recursos virtuais são armazenadas em seu repositório local.

Recursos Físicos: Este módulo também realiza a monitoração do uso dos recursos físicos. A diferença entre este e o módulo *Ambiente* é que neste módulo não são gerados alarmes quando um recurso apresenta determinado comportamento. Sendo assim, este módulo realiza o armazenamento das informações monitoradas, e que são referentes ao uso dos recursos físicos durante a execução das aplicações, em seu repositório local.

- **Gerência de Desempenho:** Esta estrutura é composta pelos módulos *Recursos Virtuais* e *Recursos Físicos*. Além disso, tem por finalidade coletar dados referentes ao uso dos recursos físicos e virtuais no decorrer da execução das aplicações. Estes dados ficam armazenados no repositório local da estrutura *Gerência de Contabilidade*, e embora sejam os mesmos coletados na estrutura *Gerência de Contabilidade*, são utilizados com outro propósito, neste caso para analisar o desempenho dos recursos no decorrer da execução das aplicações. Estas informações são encaminhadas até o módulo *Refinamento de Dados*, e serão devolvidas ao usuário e ao gerente de infra-estrutura na forma de gráficos.
- **Interface de Comunicação:** A interface de comunicação é responsável por realizar a comunicação entre os módulos da arquitetura e o protocolo de gerência utilizado. Este protocolo executará as configurações solicitadas e buscará as informações necessárias às atividades de monitoração. Essa interface poderá se comunicar com diferentes protocolos utilizados para gerenciar ambientes físicos e virtuais, como por exemplo, SNMP [3], WBEM [30], protocolos proprietários, dentre outros. Estes protocolos são representados na Figura 3 pelo módulo *Serviços de Gerenciamento em Ambientes Virtuais*.
- **Repositório:** Armazena todas as informações referentes aos ambientes físico e virtual para a execução das aplicações. Dentre as informações armazenadas estão: os dados “crus” coletados durante os processos de monitoração, todos os tratamentos realizados nos dados no módulo *Refinamento de Dados*, as configurações dos ambientes, dos recursos e dos alarmes, além dos relatórios e *logs* do ambiente.

Para o maior entendimento, a Figura 4 ilustra o ambiente em que a arquitetura de gerência pode ser utilizada. Como pode ser observado, a arquitetura encontra-se alocada em um ambiente de *cluster* de computadores, em que as máquinas comunicam-se através de uma rede. No *cluster Frontend* a arquitetura de gerência - Arquitetura de Monitoração e Controle de Ambientes Virtuais - comunica-se inicialmente com o usuário (que solicita as configurações e monitorações a serem realizadas) e, em seguida, com o protocolo de gerência (que, por sua vez, acessa os recursos e obtém as informações solicitadas pela arquitetura). O protocolo de gerência por acessar diretamente as informações que são extraídas do ambiente, possui diversos componentes alocados em diferentes lugares do mesmo. Desta forma, as partes de compõem o protocolo de gerência podem ser encontradas no *cluster frontend*, nas máquinas físicas do *cluster* e ainda

nas máquinas virtuais. Cada um destes componentes possui uma função dentro da hierarquia estabelecida pelo próprio protocolo.

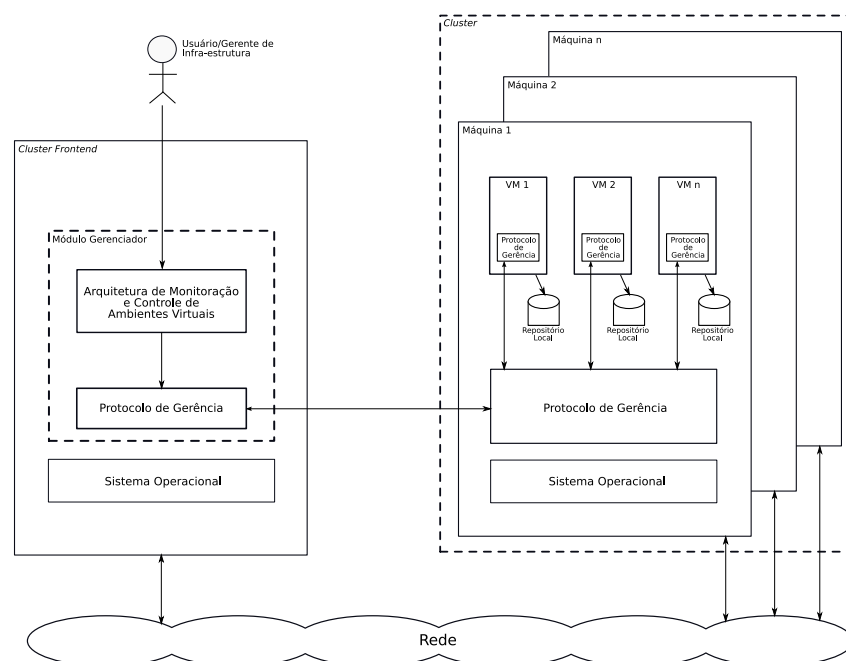


Figura 4: Ambiente para a arquitetura de gerência de ambientes virtualizados

Seguindo, é possível observar na Figura 4 alguns repositórios locais para cada uma das máquinas virtuais, que correspondem a seus discos locais, e serão utilizados como repositório de armazenamento das informações monitoradas sempre que essa monitoração for *offline*. Logo após o término da atividade de monitoração todos os dados armazenados serão encaminhados ao repositório geral da arquitetura que localiza-se no *cluster frontend*. Cabe ressaltar que o disco local das máquinas virtuais será utilizado como repositório de configuração, contabilidade e desempenho, fazendo com que as estruturas *Gerência de Configuração* e *Gerência de Contabilidade* façam a comunicação com o repositório geral da arquitetura através do módulo *Refinamento de Dados*. Tais repositórios são acessados apenas pelos módulos pertencentes às estruturas *Gerência de Configuração*, *Gerência de Contabilidade* e *Gerência de Desempenho*. Neste caso, as informações resultantes das monitorações realizadas serão enviadas ao repositório geral da arquitetura somente no final da execução das aplicações (monitoração *offline*), o que diminui o tráfego de informações entre as máquinas do *cluster*, prevenindo desta forma, um possível gargalo.

Além de estrutura de *cluster* de computadores, é possível que a arquitetura seja utilizada em ambientes servidores como *Data Centers*, por exemplo. Neste caso, o monitor é colocado no computador gerenciador e os demais componentes permanecem com o mesmo comportamento, representado através da Figura 4.

5.2 Serviços

Para a concepção da arquitetura foram levados em consideração alguns serviços básicos, os quais permitem realizar maior controle de um ambiente virtualizado.

Dentre os serviços propostos em conjunto com a arquitetura, tem-se:

1. Verificar o *status* das máquinas virtuais: Possibilidade de verificar o funcionamento das máquinas virtuais através de seu *status*. Este serviço é realizado pelo módulo *Máquinas Virtuais* da arquitetura. Sua relevância é identificada pelo fato de proporcionar o conhecimento de problemas decorrentes da sobrecarga de utilização dos recursos, como por exemplo, memória, processador, vazão do *link* de rede ou ainda, a parada de funcionamento de uma máquina virtual por motivos não identificados. Neste caso, tanto o usuário quanto o gerente de infra-estrutura poderão ser notificados através de alarmes e ainda, em um segundo momento, através de *logs* salvos durante o processo de monitoração. Para o melhor entendimento do processo, segue o algoritmo do serviço representado pela Figura 5. Neste algoritmo, os dados de entrada são: o tempo em que as monitorações deverão ser intercaladas, a lista de máquinas virtuais, o *status* que deve gerar o alarme e a operação à realizar em função do *status* capturado.

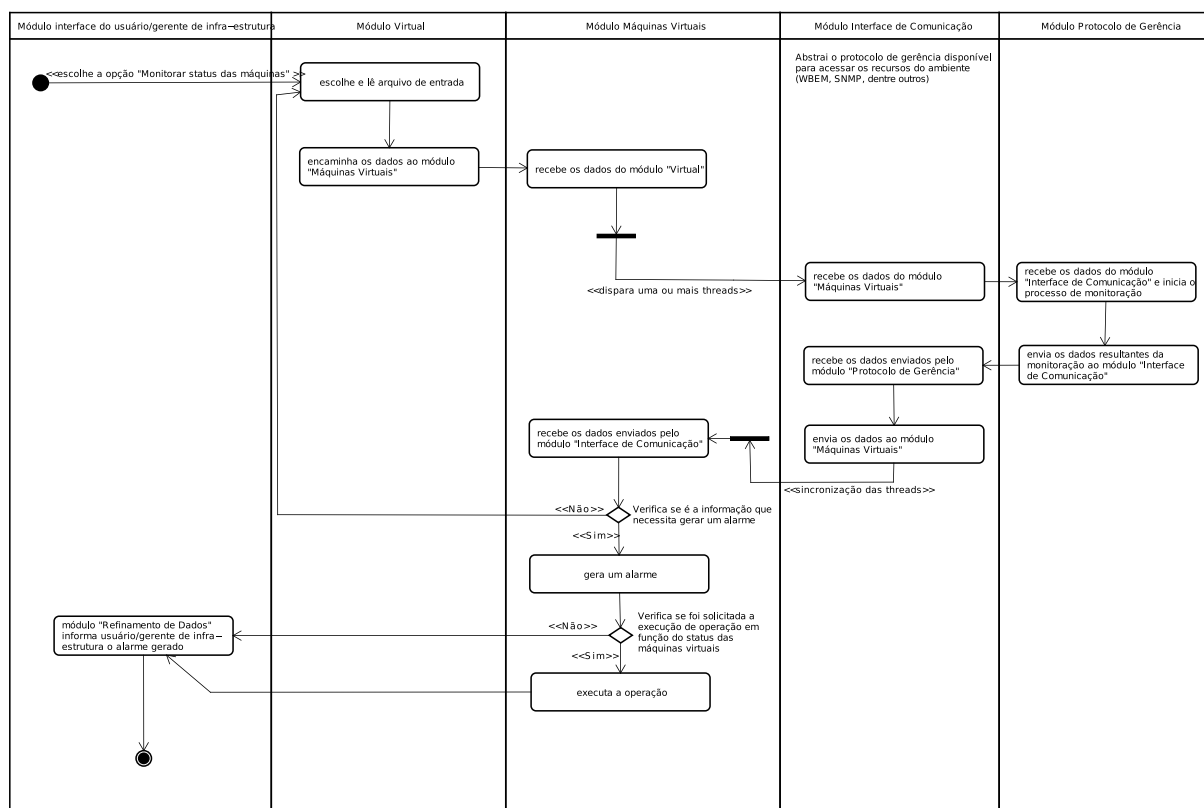


Figura 5: Algoritmo correspondente ao serviço: Verificar o *status* das máquinas virtuais

2. Controlar as aplicações: Permitir que uma aplicação apresente uma pausa, em função da necessidade do usuário e/ou do gerente de infra-estrutura, e que seu estado seja gravado no momento em que a pausa for realizada. Em um outro momento a aplicação poderá retornar sua execução a partir de gravação feita no ponto em que parou. O módulo da arquitetura que realiza esta atividade é chamado *Experimento*, descrito anteriormente. O algoritmo para o presente serviço pode ser visto através da Figura 6. Neste serviço, os dados de entrada são: aplicações que se deseja parar a execução e as máquinas físicas em que as máquinas virtuais se encontram.

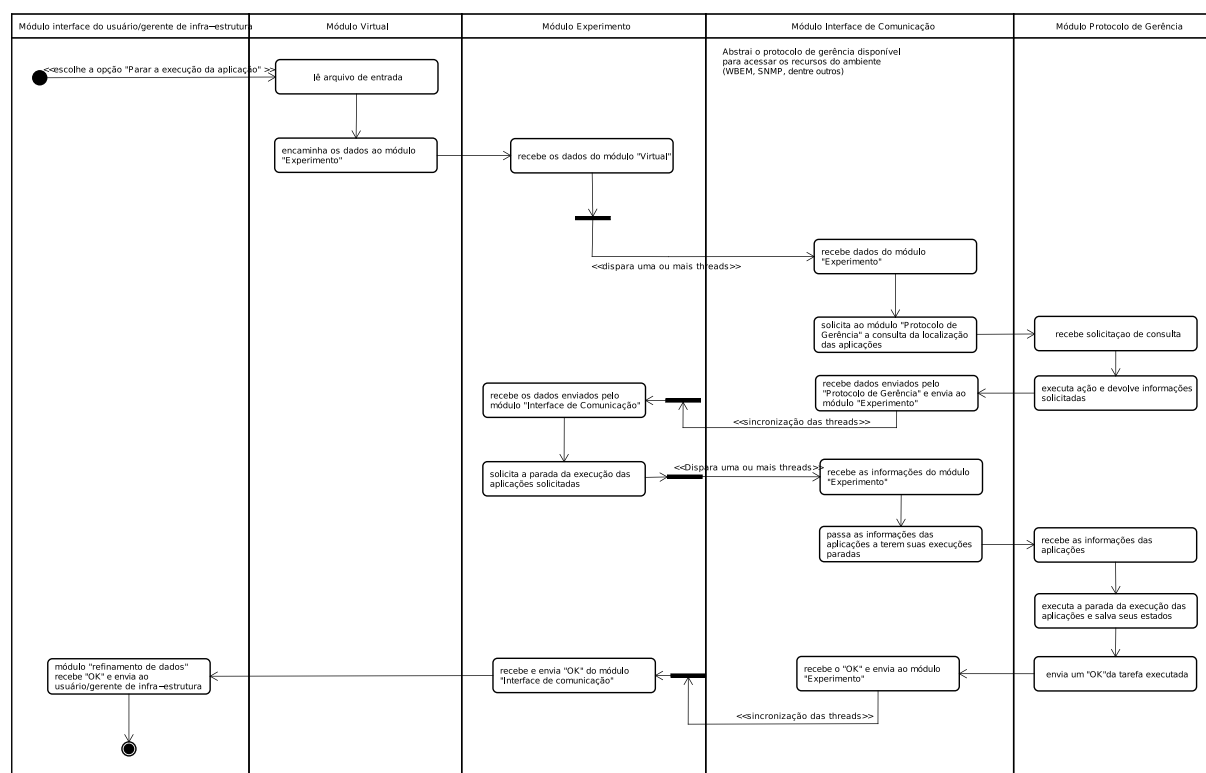


Figura 6: Algoritmo correspondente ao serviço: Controlar as aplicações

3. Conhecer as configurações de rede: Ter o conhecimento das configurações de rede, como por exemplo, o endereço IP (do inglês, *Internet Protocol*), vazão, latência, endereço físico e ainda o nome da máquina na rede. Neste caso, o usuário terá acesso a informações da rede ligadas ao ambiente virtual, e o gerente de infra-estrutura tanto do virtual quanto do físico. As informações de configurações de rede estáticas são importantes pelo conhecimento que se deve ter do ambiente, e as informações relativas a configurações dinâmicas têm relevância por servirem como apoio para o estudo de acontecimentos observados durante a execução das aplicações. Como exemplo pode-se mencionar o momento em que os valores para a vazão não foram os mesmos esperados no momento da configuração, ou ainda, a convergência das configurações para o que foi esperado. Este serviço pode ser fornecido através de mapas de configuração e alarmes. Como este serviço envolve monitoração e controle, os módulos da arquitetura envolvidos são *Dados da Gerência*

de Configuração (Ambiente) e ainda Ambiente. Segue, através da Figura 7, o algoritmo correspondente à este serviço, cujos dados de entrada são: máquinas à monitorar e informações desejadas.

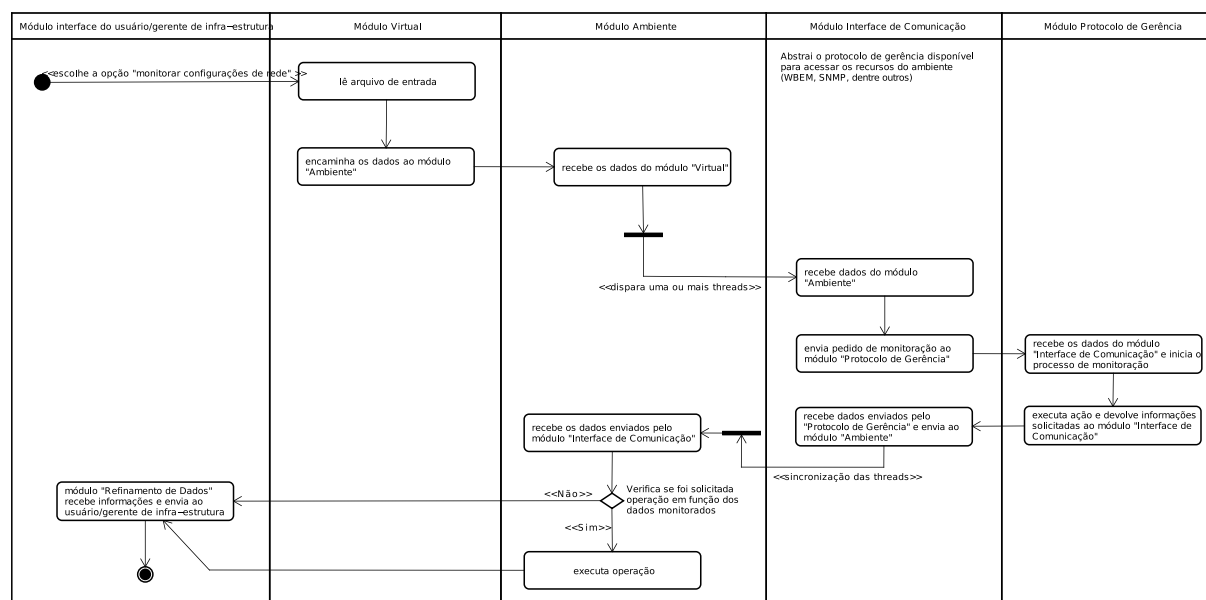


Figura 7: Algoritmo correspondente ao serviço: Conhecer as configurações de rede

4. Manter *logs*: Manter os *logs* de execução das aplicações, gerar relatórios com as informações referentes ao funcionamento de todo o sistema e das configurações e reconfigurações realizadas. A Figura 8 ilustra o algoritmo referente aos *logs* de uso dos recursos virtuais, e da mesma forma ilustra o serviço de número 5. Como dados de entrada pode-se observar: o recurso a ser monitorado e ainda o tempo para a monitoração da aplicação.
5. Contabilizar uso de recursos: Contabilizar o uso dos recursos físicos e virtuais durante a execução das aplicações, como por exemplo, memória, processador, disco, além de informações de rede, como latência e vazão. Estas informações são obtidas através dos módulos *Recursos Virtuais* e *Recursos Físicos* da estrutura *Gerência de Contabilidade*. Tais informações podem ser vistas pelo usuário e pelo gerente de infra-estrutura através de gráficos, de estatísticas, e ainda através de alarmes quando uma aplicação causar algum problema ao ambiente em função de possíveis sobrecargas no uso dos recursos.

Vale lembrar que, embora não tenha sido mencionado na descrição dos serviços, as formas utilizadas para que o usuário e o gerente de infra-estrutura tenham conhecimento de tudo que acontece nos ambientes físico e virtual, o módulo *Refinamento de Dados* disponibiliza as informações na forma de gráficos, de mapas de configuração, de alarmes, dentre outras.

Além da descrição dos serviços, é importante destacar algumas peculiaridades relativas ao funcionamento dos alarmes. É importante salientar que o usuário e o gerente de infra-estrutura têm a possibilidade de especificar que informações poderão gerar alarmes, e de que forma estes

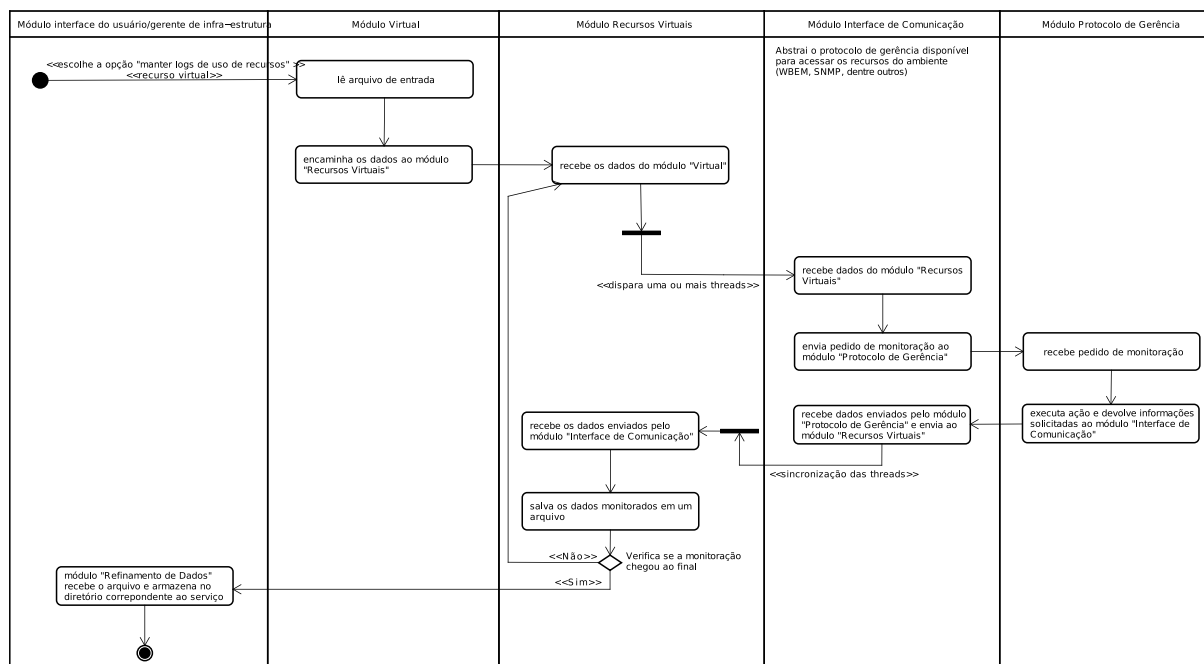


Figura 8: Algoritmo correspondente aos serviços: Manter *logs* & Contabilizar uso de recursos

alarmes poderão chegar até eles. Sendo assim, possibilidades de especificações para recebimento de alarmes são:

- Sempre que uma máquina virtual tiver seu funcionamento interrompido, sem que tenha sido previamente configurado.
- Sempre que uma configuração for alterada ao longo da execução das aplicações, sem que seja solicitado, tanto no ambiente físico quanto no virtual.
- Sempre que o uso dos recursos ultrapassar o limite estipulado.

As formas de recebimento destes alarmes também podem ser escolhidas, dentre elas pode-se observar: notificações em tela, por *e-mail* e ainda observados nos *logs* de execução mantidos. É possível ainda, que o usuário e o gerente de infra-estrutura optem pela continuação ou não da execução das aplicações em função dos alarmes gerados.

Um ponto interessante no que diz respeito aos alarmes é a forma como são trabalhados e disponibilizados ao usuário/gerente de infra-estrutura, pois podem ser realizadas combinações que representem o tipo de evento gerado e os recursos envolvidos neste processo. Podem ser utilizados padrões para caracterizar tipos de eventos, por exemplo, a sobrecarga ou subutilização de recursos, ou qualquer outro tipo de comportamento podem estar atrelados à valores que representam classes de eventos. Juntamente aos tipos de eventos, podem haver constantes para designar os recursos que podem gerar alarmes, como por exemplo CPU, memória, os ambientes físico e virtual como um todo, dentre outros. Tais constantes podem estar baseadas em padrões que usam potência de 2 (dois) para representar um tipo de evento, por exemplo: subutilização

de CPU (valor 1), sobreutilização de CPU (valor 2), subutilização de memória (valor 4), sobreutilização de memória (valor 8), e assim por diante. Da mesma forma que são previstos eventos para os recursos, pode-se utilizar os alarmes para algo mais amplo, como identificar o estado geral do ambiente, se o uso de recursos está muito alto, se as aplicações estão demorando muito em seus processamentos, além de outros. Desta forma, a camada que representa a arquitetura de gerência poderá receber múltiplos alarmes da camada inferior e enviar apenas a combinação destes alarmes (realizando a soma das potências) para a camada acima, unindo em apenas um evento todos os alarmes gerados.

Seguindo na estrutura da arquitetura, embora nenhum dos módulos que compõem a estrutura *Gerência de Desempenho* tenham sido mencionados na descrição dos serviços, é importante ressaltar que sua utilização se faz necessária quando se quer analisar o desempenho dos recursos observados no decorrer da execução das aplicações.

Através das descrições realizadas a cerca da arquitetura proposta, pode-se perceber que a utilização das recomendações das áreas funcionais de gerência na concepção desta arquitetura proporcionam além da modularidade, maior organização dos módulos que a compõe. Com isso, é possível que novos serviços sejam oferecidos e que sejam facilmente encaixados nos módulos correspondentes na arquitetura proposta.

6 Implementação da Arquitetura e Testes Realizados

Com o intuito de tornar válida a arquitetura proposta, cuja idéia é utilizar as recomendações das áreas funcionais de gerência para gerenciar ambientes virtualizados, foram implementados alguns módulos, aqueles cuja relevância mostrou-se maior no ambiente de testes.

Para isso, os módulos foram implementados na linguagem de programação Java, cuja característica de orientação a objetos facilitou a visualização dos módulos que compõem a arquitetura e com isso, auxiliou na generalização dos mesmos, para que a abstração de protocolos de gerência utilizados ficasse melhor evidenciada. A interface utilizada para realizar as configurações necessárias aos testes foi construída através de arquivos XML (do inglês, *Extensible Markup Language*), que deram estrutura às informações das quais a arquitetura se utiliza. Além disso, nos testes a tecnologia de virtualização utilizada foi Xen [12] (descrito na Seção 6.1) e o protocolo de gerência WBEM [46] (descrito na Seção 6.2). Ainda neste capítulo, é apresentada na Seção 6.3 os detalhes da implementação realizada.

6.1 Tecnologia de Virtualização Xen

Xen é uma tecnologia de virtualização originalmente desenvolvida pelo Grupo de Pesquisa de Sistemas no Laboratório de Computação da Universidade de Cambridge [47], fazendo parte de um projeto maior conhecido como XenServers [48]. Este projeto visa proporcionar um ambiente global para sistemas distribuídos, e o Xen por sua vez, desempenha um papel importante neste contexto no que se refere à utilização eficiente dos recursos computacionais, proporcionando a seus usuários a possibilidade de ter isolamento e proteção dos mesmos através da divisão eficiente da máquina física [47].

Levando-se em consideração o funcionamento do Xen, é sabido que o *hypervisor*, também chamado VMM, é invocado antes mesmo que qualquer sistema operacional seja carregado na máquina virtual. Esse componente se encarrega de gerenciar os recursos utilizados para comunicação, memória e processamento das máquinas virtuais [47].

No Xen, as máquinas virtuais são também chamadas de domínios, ou ainda Dom. Como o *hypervisor* não possui comunicação direta com as máquinas virtuais do ambiente, é necessário que um sistema inicial seja invocado pelo próprio *hypervisor*, conhecido como *Domínio0* ou simplesmente *Dom0*. A distinção entre um sistema instalado de modo convencional e um sistema instalado como *Domínio0* não é percebida, exceto o fato de que o *kernel* do *Dom0* deve

ser capaz de operar juntamente com o *hypervisor*. O *Dom0* é responsável pela comunicação com dispositivos de disco, rede, USB, dentre outros [47].

É de responsabilidade do *Dom0* controlar as demais máquinas virtuais do ambiente, também conhecidas como *Domínios U* (*DomU* (*Unprivileged Domains*) - *Domínios sem privilégios*). Além disso, responsabiliza-se por criar, iniciar e interromper os *DomU*. As máquinas virtuais sem privilégios podem se comunicar diretamente com o *hypervisor*, que controla os recursos de memória e processador [47]. Os demais recursos têm os *drivers* compartilhados pelo *Dom0*, fazendo com que todos os dispositivos compatíveis com ele sejam utilizados pelos *DomU*, mesmo que apresentem sistemas operacionais diferentes instalados [47].

É importante ressaltar que a escolha pela tecnologia de virtualização Xen foi feita pelo fato de esta ser uma tecnologia *open source* e com uma versão *free* para que o meio acadêmico possa utilizá-la para fins de pesquisa.

6.2 Arquitetura de Gerência WBEM

A arquitetura de gerência WBEM foi elaborada pela DMTF (do inglês, *Distributed Management Task Force*) e padroniza as informações a respeito dos recursos gerenciados bem como do protocolo de monitoração e controle de ambientes distribuídos [46].

Para representar a informação, a especificação WBEM utiliza o CIM, (do inglês, *Common Information Model*), para codificar os dados CIM em informações XML e assim realizar o transporte dos dados, usa-se o CIM-XML e, por fim, o através do protocolo HTTP os dados são enviados [30].

Além disso, a arquitetura é composta por clientes WBEM, clientes de eventos WBEM, servidor WBEM e ainda provedores WBEM. Sendo assim, o cliente WBEM pode invocar operações de aplicações que estão sendo gerenciadas. O cliente de eventos WBEM recebe as informações de gerência através de alarmes. O servidor WBEM responsabiliza-se por receber as requisições dos clientes WBEM e realizar a interação com os provedores. E, por fim, os provedores WBEM realizam a interação diretamente com os recursos gerenciados.

Sendo assim, o padrão WBEM foi desenvolvido com o intuito de proporcionar a gerência de ambientes de sistemas distribuídos. Por este motivo foi escolhida para auxiliar no processo de validação da arquitetura.

6.3 Ambiente de Testes

6.3.1 Estudo de caso: *Framework* para Emulação de Sistemas Distribuídos

O ambiente utilizado para testar a arquitetura de gerência de ambientes virtualizados aqui proposta, foca no desenvolvimento de um *framework* de emulação automatizada para realizar experimentos com *grid computing* e sistemas distribuídos, utilizando a combinação de técnicas de paravirtualização e ferramentas de gerência de sistemas [5]. Este *framework* é desenvolvido pelo grupo de pesquisa do qual a arquitetura também faz parte, o que facilitou sua utilização para testes com a arquitetura.

O problema de trabalhar com emulação de sistemas está ligado ao fato de que alguns requisitos são necessários para tal, como por exemplo, a existência de uma pilha de *software* complexa, o controle dos recursos físicos, simulação de algumas estruturas e algumas vezes modificações no sistema operacional que dá suporte ao ambiente de testes [5].

Embora identificados tais problemas, a primeira versão do *framework* não apresentava estrutura para fornecer atividades de gerência ao ambiente emulador. Como pode ser visto na Figura 9, seus principais componentes eram:

- **Mapeador:** Sua função é decidir em quais *hosts* de um *cluster* as máquinas virtuais serão hospedadas, analisando questões referentes à disponibilidade dos recursos nos *hosts*. Para realizar o mapeamento, o usuário do *framework* deve, através de arquivos XML, descrever o ambiente real e ainda o ambiente virtual desejado. Dentre os objetivos do algoritmo de mapeamento, dois principais são: minimizar o número de *hosts* utilizados por usuário e também, minimizar o tráfego de dados pela rede do *cluster* através da organização das máquinas virtuais que se comunicam no mesmo *host*.
- **Deployer:** No processo de *deployment* as máquinas virtuais são criadas em cada *host* do *cluster*. Para realizar esta tarefa é utilizado um padrão de gerência (neste caso o protocolo de gerência WBEM) para invocar instruções do virtualizador utilizado (neste contexto, a tecnologia de virtualização Xen [12]).
- **Gerenciador de Rede:** Tem como função principal realizar a conexão das máquinas virtuais logo após o momento de suas criações. Com as conexões estabelecidas é possível inserir o sistema emulado a partir das características do sistema distribuído desejado. Após a descrição do ambiente virtual, são criadas sub-redes formadas pelas máquinas virtuais situadas em cada *host* do *cluster*, interligando estas sub-redes através do ajuste da latência e da vazão das conexões criadas. Inicia-se então a emulação do sistema distribuído. O protocolo SNMP [3] é utilizado para executar as funções do módulo *Gerenciador de Rede*.

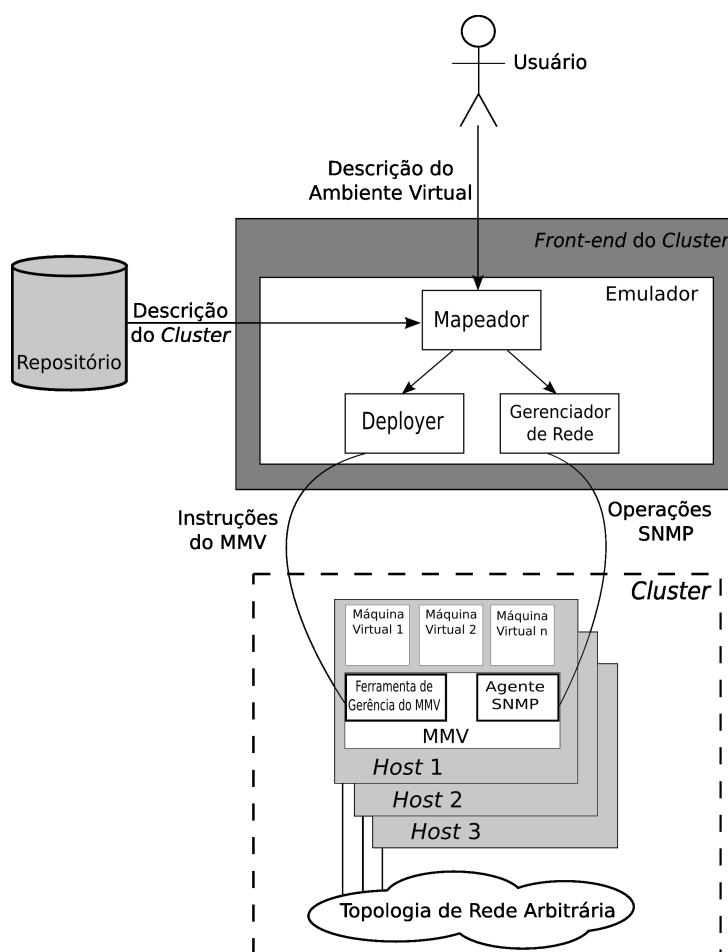


Figura 9: *Framework* para Emulação de Sistemas Distribuídos

Neste contexto identificou-se a necessidade de gerenciar o ambiente virtual com o intuito de monitorar as atividades nele realizadas e que foram montadas para emulação. A Figura 10 [5], ilustra o *framework* com o novo módulo de gerência, como pode ser visto: *Gerenciador de Experimento*.

- **Gerenciador de Experimento:** O gerenciador do experimento controla a execução do experimento de acordo com a descrição fornecida pelo usuário. Este módulo trabalha questões relacionadas ao modo como as aplicações serão configuradas, de que forma elas serão executadas, como terão suas execuções paradas e ainda, como os resultados retornarão aos usuários. Neste sentido, o Gerenciador de Experimento é composto por diferentes submódulos, com a mesma finalidade, gerenciar o ambiente virtual, mas com enfoques diferentes. Um dos submódulos é o monitor do ambiente, módulo este que corresponde à arquitetura de gerência de ambientes virtualizados aqui proposta, cuja função é processar as informações adquiridas através da monitoração, disponibilizando-as ao usuário através dos serviços mencionados no Capítulo 5, principalmente através de alarmes que identifiquem situações “adversas” no ambiente montado. O segundo submódulo diz respeito

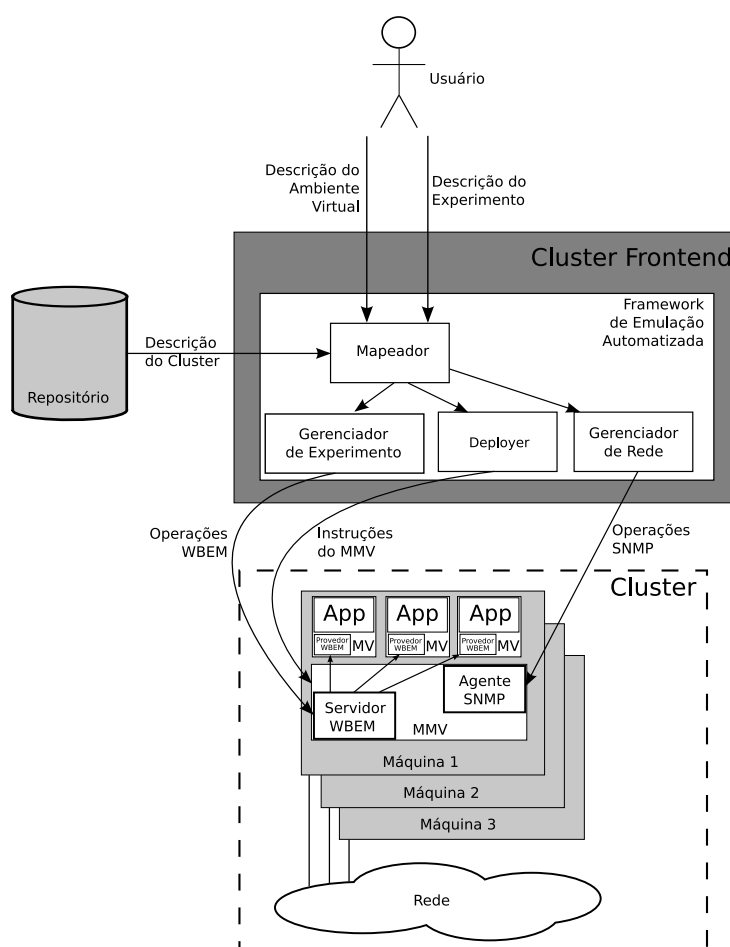


Figura 10: *Framework* para Emulação de Sistemas Distribuídos com módulo de gerência

à arquitetura de gerência baseada no protocolo de gerência WBEM [46], cuja função é realizar o acesso aos recursos, monitorando as informações solicitadas.

Analisando então o cenário aqui descrito, a proposta da arquitetura para gerenciar ambientes virtualizados tem como objetivo amenizar o impacto causado pela complexidade de trabalhar com emulação, que utiliza técnicas de virtualização, mais ainda por se tratar de emulação de sistemas distribuídos, haja visto as exigências inferidas.

Sendo assim, a Subseção 6.3.2 apresenta a integração da arquitetura proposta com o *framework* descrito nesta subseção, para finalidade de testes da mesma.

6.3.2 Integração da arquitetura de gerência com os módulos do *framework*

Visando amenizar os problemas que ambientes virtualizados podem enfrentar, em especial sistemas distribuídos que utilizam tecnologias de virtualização como base para executar suas funções, é que a arquitetura de gerência de ambientes virtualizados está integrada ao *framework* descrito na Subseção 6.3.1.

Após a concepção da arquitetura de gerência de ambientes virtualizados pensou-se nos módulos interessantes para que esta pudesse ser testada, e de certa forma útil para o trabalho que a utiliza. Desta forma, identificou-se os alarmes como sendo o objeto principal da validação, visto que estes são de grande valia para utilização no *framework* (descrito na Seção 6.3.1).

Neste cenário, os módulos da arquitetura implementados foram:

- Refinamento de Dados: Responsável por receber as informações dos alarmes e repassá-las ao usuário/gerente de infra-estrutura ou ainda, o sistema que esteja utilizando a arquitetura.
- Ambiente: Responsável por monitorar o uso de recursos pelas aplicações executadas no ambiente virtual. Nota-se que embora o módulo responsável pela contabilização de uso de recursos seja o módulo *Recursos Virtuais* da estrutura *Gerência de Contabilidade*, foi implementado no módulo ambiente devido ao fato de esta contabilidade ser utilizada juntamente com o *logs* de monitoração de uso dos recursos.
- Interface de comunicação: Responsável pela comunicação dos módulos da arquitetura com o protocolo de gerência, que por sua vez realiza o acesso aos recursos. O protocolo utilizado para validar foi o WBEM [46].

Através dos módulos implementados foi possível desenvolver os alarmes. Nesse sentido, uma forma bastante interessante de fornecer a funcionalidade de alarmes foi pensada com o intuito de simplificar a quantidade destes alarmes enviados aos usuários da arquitetura. Sendo assim, os alarmes foram separados por categorias: alarmes de sobreutilização (sobrecarga) e alarmes de subutilização (ociosidade). Seguindo ainda a estrutura pensada, os alarmes foram também separados por recursos: no caso, memória, CPU, disco, ambiente, e assim por diante. Para cada uma das combinações, por exemplo sobreutilização de memória, subutilização de CPU, foram atribuídos valores, valores estes seguindo a ordem de números potência de 2. Essas combinações são enviadas ao usuário da arquitetura juntamente com a máquina virtual que apresentou este problema. É importante salientar que o envio dos alarmes é controlado pelo número de máquinas que geram alarme, independente de terem gerado um ou mais, ou seja, sempre que uma máquina virtual gerar mais de um alarme, estes são agrupados e enviados para a camada acima apenas uma vez, deixando o trabalho da interpretação do que foi recebido para o usuário.

Com a implementação dos módulos anteriormente mencionados, foi possível salvar o *log* de uso dos recursos com o intuito de gerar gráficos que demonstrem esse uso. Com os gráficos é possível analisar o comportamento do experimento e/ou aplicação, pois pode-se acompanhar a evolução de uso desses recursos. Outra possibilidade interessante, está vinculada ao fato de poder desenhar o mapa das configurações realizadas e, com isso, ter conhecimento das máquinas virtuais no ambiente físico.

Cabe ressaltar que para que os alarmes fossem gerados foi necessária a configuração de um arquivo, descrito em XML, para que fossem informados as máquinas virtuais a serem monitoradas, os recursos destas máquinas virtuais a gerenciar, o tempo de monitoração e ainda os limites tanto para sobrecarga quanto para ociosidade de utilização dos recursos do ambiente.

Desta forma, pode-se observar o comportamento da arquitetura enquanto ferramenta para prover os serviços anteriormente mencionados. A análise deste comportamento será descrita na Subseção 6.3.4.

6.3.3 Testes

O teste realizado para reconfigurar dinamicamente o ambiente tem como objetivo demonstrar que os serviços de monitoração, neste caso específico, o de alarme, conseguem detectar a sobreutilização dos recursos pelas aplicações que estão sendo executadas no *cluster* de máquinas virtuais. Após a identificação do uso excessivo de memória pela aplicação, é enviado um alarme ao remapeador, que realiza um novo mapeamento para que o ambiente não apresente novamente este comportamento. Com isso, o ambiente físico utilizado para os testes é composto por quatro máquinas físicas (*hosts*), cada uma com um processador Pentium IV 2.8 GHz e 2.560MB de memória RAM. As máquinas estão conectadas através de um *switch* Fast Ethernet, e possuem instalado a plataforma de virtualização Xen [12]. Além disso, cada domínio 0 e as máquinas virtuais possuem instalados o servidor WBEM OpenPegasus [46] e o sistema operacional Debian GNU/Linux [49].

O ambiente virtual criado possui 32 máquinas virtuais com 256MB de memória cada, criadas nos quatro *hosts* disponíveis. No arquivo de descrição do experimento foi configurado o uso de no mínimo dois *hosts*, que a quantidade de memória possa ser aumentada ou diminuída e que as máquinas virtuais não podem utilizar mais que 80% da memória disponível para elas.

Para o remapeamento realizado pelo emulador são levadas em consideração a disponibilidade dos recursos no *cluster* e as condições impostas pelo usuário, que estão contidas no arquivo de descrição do experimento. Neste momento, o remapeamento é então realizado.

Para demonstrar o comportamento dos alarmes disparados pelo uso em excesso de memória, utilizou-se uma aplicação responsável por alocar 200MB de memória em cada máquina virtual. A arquitetura de gerência de ambientes virtualizados aqui apresentada, realizou a monitoração do uso dos recursos durante a execução da aplicação e no momento em que as máquinas virtuais excederam a quantidade de memória que lhes foi reservada. Cabe salientar a forma que a camada de monitoração aqui proposta trabalha a questão do recebimento dos alarmes, pois recebe múltiplos deles e, através do mecanismo de tratamento de alarmes envia apenas um à camada remapeadora. Através da combinação destes alarmes, conforme descrito no Capítulo 5, o remapeador consegue identificar os problemas encontrados e aí sim pode realizar um novo mapeamento para melhorar as condições de uso dos recursos.

Para melhor demonstrar o comportamento das máquinas virtuais durante a execução da aplicação, foram gerados gráficos, como pode ser visto na Figura 11, que apresenta a utilização de memória por máquina virtual. É possível perceber que os três primeiros minutos foram utilizados pelo monitor do ambiente (a arquitetura proposta) para invocar o método que retorna a quantidade de memória utilizada. Após ter identificado a sobreutilização de memória o módulo remapeador é ativado. Levando em consideração a disponibilidade dos recursos do *cluster* e ainda as condições impostas pelo usuário, é realizado um novo mapeamento do ambiente. Nesta nova configuração o número de máquinas virtuais reduziu de 32 para 16 e a memória aumentou de 256MB para 512MB, como ilustrado na Figura 12.

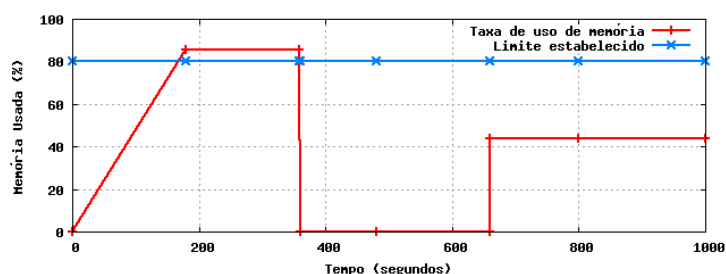


Figura 11: Utilização de memória de uma máquina virtual antes e após o disparo do alarme

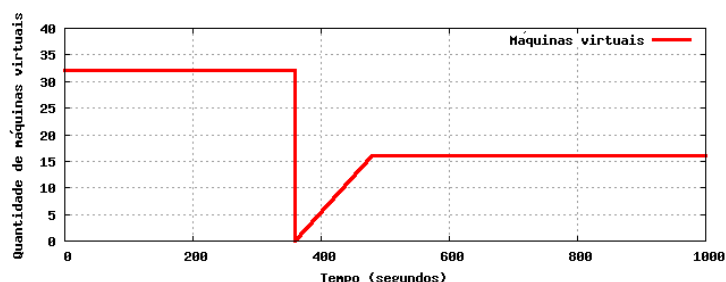


Figura 12: Número de máquinas virtuais no ambiente antes e após o disparo do alarme

Através da Figura 12, é possível perceber o uso de memória estabilizando em 44% decorridos 11 segundos do remapeamento. Após o novo mapeamento, as máquinas virtuais do ambiente são destruídas e logo em seguida recriadas. A aplicação que consome memória é então novamente executada.

6.3.4 Considerações a partir dos testes realizados

Concluído o teste de validação da arquitetura, pôde-se perceber que os serviços de alarmes apresentaram bons resultados, pois auxiliaram no processo de descoberta de situações adversas através do envio de notificações à camada mapeadora do *framework*. Este, por sua vez, realizou um novo mapeamento em função do que havia recebido.

O serviço de alarmes apresentou o comportamento esperado, e com isso proporcionou satisfatoriamente ao módulo que o utilizou, um meio de obter informações do ambiente de forma que auxiliem no processo de tomadas de decisão sempre que um comportamento inesperado é detectado.

7 Considerações Finais

Este trabalho apresentou uma arquitetura para gerenciar ambientes virtualizados e com isso amenizar os efeitos de problemas encontrados em ambientes deste tipo, como por exemplo, controlar os recursos do ambiente físico para que se mantenham funcionando, garantindo a disponibilidade de serviços que antes estavam distribuídos em diferentes máquinas físicas, além de outros desafios mencionados no Capítulo 2.

Para que a possibilidade de trabalhar na concepção de uma arquitetura de gerência de ambientes virtualizados fosse possível, foi necessário buscar conceitos e padrões que dessem norte à proposta, e que auxiliassem na identificação de alguns dos diversos serviços que podem ser oferecidos para o fim da gerência. Estes padrões não foram desenvolvidos especificamente para ambientes que utilizam-se dos benefícios das tecnologias de virtualização, mas foram adaptados para tal. Desta forma, as recomendações das áreas funcionais de gerência, também conhecidas como FCAPS, serviram como guia neste processo de desenvolvimento da idéia da arquitetura descrita no Capítulo 5.

Uma das idéias para a arquitetura foi deixá-la flexível em termos de tecnologias de virtualização e protocolos de gerência utilizados pra acessar as informações dos recursos do ambiente. Para isso, um módulo que trabalha como uma interface de comunicação entre os módulos da arquitetura e o protocolo foi pensada. Sendo assim, apenas essa interface deve ser alterada sempre que houver mudança no protocolo utilizado. Além disso, como a arquitetura proposta não prevê uma comunicação direta com o sistema de virtualização, este também fica desvinculado da arquitetura e com isso não há restrição quanto à tecnologia utilizada.

É sabido que a generalidade, neste caso da tecnologia de virtualização e protocolo de gerência, pode introduzir algumas limitações na arquitetura, como por exemplo impossibilitar o desenvolvimento de soluções específicas para uma única tecnologia, ou ainda não conseguir atacar um problema bem característico da mesma. Mas é importante destacar que a flexibilidade proporcionada, por não restringir o uso da arquitetura com qualquer tecnologia de virtualização e protocolos existentes, apresenta vantagens significativas e válidas para a proposta apresentada através deste trabalho. Dentre as vantagens pode-se destacar:

- Os módulos, independente do protocolo, poderão realizar suas funções sem apresentar problemas de compatibilidade com outros protocolos que poderão ser utilizados simultaneamente.
- Maior abrangência com relação às tecnologias e padrões utilizados.

- A arquitetura apresenta fácil adaptação ao cenário que será utilizada.

Além das observações mencionadas anteriormente e que auxiliaram na concepção de uma arquitetura flexível, é inegável a modularidade proporcionada pelas recomendações das áreas funcionais de gerência seguidas, pois para cada propósito gerencial, seja falhas, configuração, contabilidade ou desempenho, foi possível encaixar suas definições e funções propostas. É inegável também, a simplicidade que a arquitetura apresenta em função dos módulos que a compõe, pois o modo de disponibilização dos mesmos permite que novos serviços sejam propostos e acrescentados à arquitetura.

Como trabalhos futuros pode-se identificar a possibilidade de integrar à arquitetura uma estrutura responsável por tratar questões de segurança no ambiente, deixando-a completa de acordo com cinco áreas funcionais de gerência. Além disso, o mecanismo de alarmes pode apresentar características adicionais, combinando por exemplo alarmes com diferentes recursos, fornecendo deste modo, fácil adaptação da arquitetura aos diferentes cenários em que pode ser inserida.

Sendo assim, pôde-se perceber que as recomendações das áreas funcionais de gerência foram de grande valia, pois auxiliaram no processo de desenvolvimento de uma arquitetura simples, modular e de fácil adaptação ao cenário no qual está inserida, visto que na implementação dos módulos escolhidos foi possível identificar tais benefícios.

Referências

- [1] CREASY, R. J. The origin of the vm/370 time-sharing system. *IBM Journal of Research and Development*, IBM, Palo Alto, CA, USA, v. 25, n. 5, p. 483–490, sep 1981. ISSN 0018-8646.
- [2] BULLERS, J. W. I.; BURD, S.; SEAZZU, A. F. Virtual machines - an idea whose time has returned: application to network, security, and database courses. In: *SIGCSE '06: Proceedings of the 37th SIGCSE technical symposium on Computer science education*. New York, NY, USA: ACM, 2006. p. 102–106. ISBN 1-59593-259-3.
- [3] STALLINGS, W. *SNMP,SNMPV2,Snmpv3,and RMON 1 and 2*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998. ISBN 0201485346.
- [4] OLIVEIRA, M. et al. Introdução à gerência de redes atm. In: *XVI Simpósio Brasileiro de Redes de Computadores*. Rio de Janeiro, RJ, Brasil: SBC, 1998. p. 1–190.
- [5] CALHEIROS, R. N. et al. Applying virtualization and system management in a cluster to implement an automated emulation testbed for grid applications. In: *SBAC-PAD '08: Proceedings of the 2008 20th International Symposium on Computer Architecture and High Performance Computing*. Washington, DC, USA: IEEE Computer Society, 2008. p. 97–104. ISBN 978-0-7695-3423-7.
- [6] HP. *HP Virtual Server Environment*. Disponível em: <http://h71028.www7.hp.com/enterprise/cache/258348-0-0-0-121.html?jumpid=reg_R1002_USEN>. Acesso em: 31 jul. 2008.
- [7] IBM. *IBM Systems Virtualization*. Disponível em: <<http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/eicay/eicay.pdf>>. Acesso em: 30 mai. 2008.
- [8] AMD. *AMD64 Virtualization Codenamed "Pacifica" Technology Secure Virtual Machine Architecture Reference Manual*. Disponível em: <http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/33047.pdf>. Acesso em: 14 mai. 2008.
- [9] ROSENBLUM, M. The reincarnation of virtual machines. *Queue*, ACM, New York, NY, USA, v. 2, n. 5, p. 34–40, 2004. ISSN 1542-7730.
- [10] NANDA, S.; CHIUEH, T. cker. A survey of virtualization technologies. 2005. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.74.371;http://www.ecsl.cs.sunysb.edu/tr/TR179.pdf>>. Acesso em: 20 abr. 2008.
- [11] VIRTUALIZATION, Virtual Machine & Virtual Server Consolidation - VMware. Disponível em: <<http://www.vmware.com>>. Acesso em: 03 mai. 2008.

- [12] BARHAM, P. et al. Xen and the art of virtualization. In: *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 2003. p. 164–177. ISBN 1-58113-757-5.
- [13] MICROSOFT. *Virtual PC 2007 Technical Overview*. Disponível em: <<http://www.microsoft.com/windows/products/winfamily/virtualpc/default.mspx>>. Acesso em: 30 mai. 2008.
- [14] KAMP, P.-H.; WATSON, R. N. M. *Jails: Confining the omnipotent root*. 05 2000.
- [15] SWSOFT. *OpenVZ User's Guide*. 2005. Disponível em: <<http://download.openvz.org/doc/OpenVZ-Users-Guide.pdf>>. Acesso em: 30 mai. 2008.
- [16] SMALLTALK. Disponível em: <<http://www.smalltalk.org/main/>>. Acesso em: 03 mai. 2008.
- [17] ARNOLD, K.; GOSLING, J.; HOLMES, D. D. C. *The Java Programming Language*. Fourth. pub-AW:adr: Addison-Wesley, 2005. xxviii + 891 p. ISBN 0-321-34980-6.
- [18] FRAMPTON, D. et al. Demystifying magic: high-level low-level programming. In: *VEE '09: Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*. New York, NY, USA: ACM, 2009. p. 81–90. ISBN 1605583754.
- [19] TANENBAUM, A. S. *Sistemas Operacionais Modernos*. São Paulo, SP, Brasil: Prentice Hall Brasil, 2007. ISBN 8587918575.
- [20] SMITH, J. E.; NAIR, R. *Virtual Machines: Versatile Platforms for Systems and Processes*. San Francisco, CA, USA: Elsevier, 2005. ISBN 1558609105.
- [21] CLARK, C. et al. Live migration of virtual machines. In: *NSDI'05: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*. Berkeley, CA, USA: USENIX Association, 2005. p. 273–286.
- [22] CASSAT. *Managing Virtual Infrastructure*. Disponível em: <http://www.cassatt.com/sltm_virt_control.htm>. Acesso em: 25 jun. 2008.
- [23] SILVA, R. O. da. *Teorias da administração*. São Paulo: Pioneira, 2001.
- [24] ROBBINS, S. P. *Administração: mudanças e perspectivas*. São Paulo: Saraiva, 2003.
- [25] HAMPTON, D. R. *Administração contemporânea*. São Paulo: Makron Books, 1992.
- [26] FERNANDES, A. *Administração inteligente: novos caminhos para as organizações do século XXI*. São Paulo: Futura, 2002.
- [27] TOLENTINO, M. E.; TURNER, J.; CAMERON, K. W. Memory-miser: a performance-constrained runtime system for power-scalable clusters. In: *CF '07: Proceedings of the 4th international conference on Computing frontiers*. New York, NY, USA: ACM, 2007. p. 237–246. ISBN 1595936837.
- [28] WALDSPURGER, C. A. Memory resource management in vmware esx server. *SIGOPS Oper. Syst. Rev.*, ACM, New York, NY, USA, v. 36, n. SI, p. 181–194, 2002. ISSN 0163-5980.

- [29] BOBBA, J. et al. Performance pathologies in hardware transactional memory. In: *ISCA '07: Proceedings of the 34th annual international symposium on Computer architecture*. New York, NY, USA: ACM, 2007. p. 81–91. ISBN 978-1-59593-706-3.
- [30] HOBBS, C. *A Practical Approach to WBEM/CIM Management*. Boca Raton, FL, USA: CRC Press, Inc., 2004. ISBN 0849323061.
- [31] FOROUZAN, B. A. *Comunicação de Dados e Redes de Computadores*. Porto Alegre, RS, Brasil: Bookman, 2006. ISBN 8536306149.
- [32] STALLINGS, W. *Data and Computer Communications*. Upper Saddle River, NJ, USA: Prentice Hall, 2007. ISBN 0132433109.
- [33] TANENBAUM, A. S. *Computer Networks*. Upper Saddle River, NJ, USA: Prentice Hall, 2003. ISBN 0130661023.
- [34] CARVALHO, B. S. *Introdução ao Gerenciamento de Ambientes Computacionais*. Belo Horizonte, MG, Brasil: C/ Arte, 2003. ISBN 8587073796.
- [35] IBM. *IBM Systems: IBM Director Hardware and Software Support Guide*. 2008. Disponível em: <http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo_5.20/frn0_bk_compat_guide.pdf>. Acesso em: 30 mai. 2008.
- [36] CORPORATION, M. *Microsoft Virtual Server 2005 R2 Product Overview*. 12 2005. Disponível em: <https://portal.asap.com/en-US/Documents/Microsoft_Virtual_Server_2005R2_Datasheet.pdf>. Acesso em: 30 mai. 2008.
- [37] BIXDATA. *BixData User Guide*. 2008. Disponível em: <<http://www.bixdata.com/files/BixData%20-%20Use%20Guide.pdf>>. Acesso em: 10 jun. 2008.
- [38] JOOSTE, K. *Challenges in Cluster Management: A Technical Whitepaper*. Disponível em: <<http://www.bixdata.com/static/Whitepaper%20-%20BixData%20Challenges%20in%20Cluster%20Management.pdf>>. Acesso em: 10 jun. 2008.
- [39] ENOMALISM. *Enomalism Elastic Computing Platform: Building a Virtual Machine*. Disponível em: <http://www.enomalism.com/fileadmin/wink_tuts/0806_enomalism_buildvm.pdf>. Acesso em: 20 jun. 2008.
- [40] ENOMALISM. *Enomalism Elastic Computing Platform: Enomalism Overview*. Disponível em: <http://www.enomalism.com/fileadmin/wink_tuts/0806_enomalism_dashboard.pdf>. Acesso em: 20 jun. 2008.
- [41] COLD, S. J. Using really simple syndication (rss) to enhance student research. *SIGITE Newsl.*, ACM, New York, NY, USA, v. 3, n. 1, p. 6–9, 2006.
- [42] MATTHEWS, J. N. et al. *Running Xen: A Hands-On Guide to the Art of Virtualization*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2008. ISBN 0132349663, 9780132349666.
- [43] CITRIX. *Citrix XenServer Product Overview: Unlock the power of server virtualization*. Disponível em: <<http://www.xensource.com/Documents/XenServer41ProductOverview.pdf>>. Acesso em: 24 jun. 2008.

- [44] CITRIX. *Citrix*. Disponível em: <<http://www.citrix.com/lang/English/home.asp>>. Acesso em: 30 jun. 2008.
- [45] IBM. *Using Apache Derby databases, Part 1: Managed objects in a converged provider environment*. Disponível em: <<http://www.ibm.com/developerworks/opensource/library/os-ad-mgobj/>>. Acesso em: 30 mai. 2008.
- [46] ECKSTEIN, D. *WBEM Overview*. USA: Open Pegasus, 2003. Disponível em: <<http://www.openpegasus.org/uploads/40/3448/WBEMOverview.pdf>>. Acesso em: 30 mai. 2008.
- [47] CARMONA, T. *Virtualização*. São Paulo, SP, Brasil: Linux New Media do Brasil Editora Ltda, 2008. ISBN 8561024062.
- [48] HAND, S. et al. *Controlling the Xenoserver Open Plataform*. 2003.
- [49] FOSTER, I. et al. Virtual clusters for grid communities. In: *CCGRID '06: Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid*. Washington, DC, USA: IEEE Computer Society, 2006. p. 513–520. ISBN 0769525857.