**Daniel Oliveira Nascimento**

# *Gerenciamento de Ambientes Computacionais Heterogêneos Via Web*

Dissertação apresentada ao Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais, como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador:

Sérgio Vale Aguiar Campos

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Belo Horizonte

30 de janeiro de 2008

# *Agradecimentos*

Agradeço ao meu orientador Sérgio Campos por me animar quando eu achava que não ia conseguir, pelo auxilio e dedicação como aluno da graduação durante a inciação científica e no curso do mestrado.

Agradeço a empresa International Syst por tornar possível esse trabalho, oferencendo todo o suporte necessário para a conclusão do mesmo. Agradeço a todos os funcionários dessa empresa que contribuiram de alguma forma com o desenvolvimento do sistema.

Gostaria de agradecer especialmente ao Hélio Marques por toda paciência durante o tempo que trabalhamos juntos no planejamento e desenvolvimento da solução. Agradeço também aos meus amigos Ítalo e Bárbara pelo apoio durante esse período.

Gostaria de agradecer aos meus pais por sempre acreditarem em mim, por sempre estarem presentes e por me apoiarem nos momentos mais difíceis. Obrigado por toda a ajuda que vocês me deram durante toda minha vida. Obrigado também à todos os meus familiares, pelo apoio.

Obrigado a todos.

# *Resumo*

Os parques computacionais das organizações públicas estaduais e federais existentes no Brasil são singulares pois geralmente estão espalhadas por grandes extensões geográficas, não possuem pessoal especializado para prestar o suporte localmente, lidam com uma diversidade de sistemas operacionais e em alguns casos são interconectadas por tecnologias diferentes. Quando essas máquinas estão interconectadas pela Internet utilizando provedores diferentes, estes impõem políticas de segurança distintos que dificultam a interação de uma ferramenta de gerência com o conjunto de dispositivos a serem monitorados.

Neste trabalho foi desenvolvido uma ferramenta de gerência para os grandes parques computacionais do setor público, levando em consideração diversas características e peculiaridades desse ambiente.

Atualmente as ferramentas existentes para gerência de redes desse porte possuem limitações que dificultam a sua implantação e utilização nesse tipo de ambiente. Algumas dessas ferramentas que possuem as funcionalidades necessárias são caras e exigem pessoal treinado para manipular e personalizar o sistema visando atender as necessidades deste tipo de cliente.

Em relação às ferramentas disponíveis, este trabalho apresenta um conjunto de contribuições que possibilitam que os objetivos do sistema sejam atendidos. São elas: possuir o código aberto, permitir que com um mínimo de configuração inicial das unidades o sistema possa funcionar sem intervenção local dos usuários das máquinas no ambiente monitorado e a capacidade de atender a milhares de máquinas simultaneamente.

O sistema desenvolvido está em operação em aproximadamente 100 escolas de Minas Gerais participantes do programa *Escolas em Rede* da secretaria de educação de Minas Gerais. Quando esse programa estiver completo serão quatro mil escolas monitoradas pelo sistema. Neste ambiente a ferramenta se comportou como esperado possibilitando a gerência centralizada dessas escolas.

# *Abstract*

The public organizations' computational environment existing in Brazil are singular because they are usually spread over large geographic areas, do not have specialized staff to provide local support, deal with a diversity of operating systems, which some times are interconnected by different technologies.

When these machines are interconnected by the Internet using different providers, they require different security policies that make a tool interaction with devices managed more difficult.

In this work we propose a tool for assist in network management for big enviroments from governments, taking into account several of the features and peculiarities of this environment.

Currently the tools available for management of such networks have limitations that hamper its deployment and use in that kind of environment. Some tools that could provide the required funcionalities are expensive and require qualified staff to manipulate and customize the system to meet the needs of this type of customer.

Comparing with other tools available this work presents a set of contributions that allow the goals of the system are met. It has open code, allow a minimum of initial configuration of units so the system can operate without intervention of the local users of the machines in the environment managed and the ability to answer thousands of machines simultaneously.

The system developed is in operation in about one hundred Minas Gerais schools participants of the program *Escolas em Rede* from Secretary of Education of Minas Gerais. Near its completion we will be monitoring four thounsand schools. In this environment the tool has behaved as expected offering the possibility of centralized management of these schools.

# *Sumário*

# *Lista de Figuras*

# *Lista de Tabelas*

# *Lista de abreviaturas e siglas*

| | | |
|---|---|---|
| SEEMG | Secretaria de Educação de Minas Gerais, | p. 3 |
| MIB | Management Information Base, | p. 10 |
| SNMP | Simple Network Management Protocol, | p. 11 |
| TCP | Transmission Control Protocol, | p. 12 |
| UDP | User Datagram Protocol, | p. 12 |
| SMIv1 | Structure of Management Information Version 1, | p. 13 |
| SMIv2 | Structure of Management Information Version 2, | p. 13 |
| OID | Object Identifier, | p. 13 |
| ASN.1 | Abstract Syntax Notation One, | p. 13 |
| WBEM | Web Based Enterprise Management, | p. 14 |
| DMTF | Distributed Management Task Force, | p. 15 |
| CIM | Common Information Model, | p. 15 |
| HTTP | HyperText Transfer Protocol, | p. 15 |
| XML | Extensible Markup Language, | p. 15 |
| CIMOM | CIM Object Manager, | p. 15 |
| MOF | Managed Object Format, | p. 16 |
| DTD | ocument Type Definition, | p. 16 |
| WQL | WBEM Query Language, | p. 17 |
| ANSI | American National Standards Institute, | p. 17 |
| SQL | Structured Query Language, | p. 17 |
| WMI | Windows Management Instrumentation, | p. 18 |
| DCOM | Distributed Component Object Model, | p. 18 |
| NRPE | Nagios Remote Plugin Executor, | p. 19 |
| NSCA | Nagios Service Check Acceptor, | p. 20 |
| ADSL | Asymmetric Digital Subscriber Line, | p. 36 |
| MEEV | Máquina de Estados de Envio de Variáveis, | p. 38 |
| MERV | Máquina de Estados de Recepção de Variáveis, | p. 38 |

# 1 Introdução

Os ambientes computacionais têm se tornado cada vez maiores, mais velozes e complexos [21]. A especialização das aplicações cria a necessidade de se manter ambientes computacionais heterogêneos que elevam o grau de complexidade do ambiente. Com isso grandes parques computacionais que requerem uma equipe de profissionais para instalar, configurar, otimizar e mantê-los [23] estão ficando cada vez mais freqüentes. Porém nem sempre a existência dessa equipe é possível ou viável, fazendo com que os responsáveis por estes ambientes fiquem sobrecarregados e como conseqüência mais suscetíveis a erros.

Em nosso país onde os recursos governamentais são escassos e limitados, o desafio de manter estas grandes estruturas, com centenas e às vezes milhares de máquinas, em diversas divisões públicas torna esse problema crítico, pois o pessoal capacitado para realizar esse tipo de tarefa fica concentrado em grandes centros atendendo as diversas repartições menores localizadas no interior do país. Nestas repartições, por sua vez, a tarefa de gerenciar as redes locais é, muitas vezes, delegada a uma pessoa não técnica. Nesses casos as ferramentas atuais de monitoramento e gerenciamento de redes não estão compatíveis com o nível de conhecimento destas pessoas, provendo informações de muito baixo nível e portanto de pouco serventia para esses usuários leigos.

Essas redes, geralmente, possuem componentes distribuídos por grandes extensões geográficas que são conectados por meios diversos. Neste contexto são inseridos diferentes provedores de acesso com políticas de segurança distintos que dificultam a interação de uma ferramenta de gerência com o conjunto de dispositivos a serem monitorados.

Atualmente as ferramentas existentes para gerência de redes desse porte possuem limitações que dificultam a sua implantação e utilização nesse tipo de ambiente. Algumas são caras e exigem pessoal treinado para manipular e personalizar o sistema para atender as necessidades deste tipo de cliente.

Outras possuem código aberto, podendo até mesmo serem gratuitos, porém foram desenvolvidas visando monitorar e gerenciar dispositivos que se encontram numa mesma rede, ou no mínimo são acessíveis diretamente pela unidade central de gerência. Nessas ferramentas uma modificação em sua lógica para gerenciar máquinas inacessíveis, devido aos problemas de conexão expostos anteriormente, implicaria em grandes alterações em suas implementações ou na perda de alguma funcionalidade desejável.

Neste trabalho é proposto o desenvolvimento de uma ferramenta capaz de realizar a gerência dos grandes parques computacionais do setor público, levando em consideração as diversas características e peculiaridades desse ambiente.

## 1.1 Sistemas de Gerência

Um sistema de monitoramento pode ser responsável por colher métricas de utilização da rede e dos recursos das máquinas, fazer o inventário do parque computacional identificando dispositivos e aplicações instalados e gerar alertas quando alguma condição crítica no ambiente é atingida.

Um sistema de gerência fornece meios para interação com o ambiente que permitem modificar o comportamento das aplicações ou dispositivos nele contido. Num ambiente bem monitorado este sistema pode interagir com os dados monitorados para realizar tarefas automaticamente ou oferecer informações relevantes visando melhorar a atuação do administrador do ambiente.

No capítulo 2 este tema é aprofundado e são apresentadas várias definições e conceitos utilizados durante o desenvolvimento do trabalho.

## 1.2 Projeto Escolas em Rede

O governo do estado de Minas Gerais está implantando o programa Escola em Rede no estado. Este programa prevê a compra de computadores que serão instaladas em todas as 4 mil escolas do ensino público estadual. Visando uma maior flexibilidade e robustez do ambiente criado e, além disso, uma significativa economia de recursos do estado, será dado preferência à utilização de softwares livres nas máquinas para a realização de todas as tarefas que puderem ser executadas por estas ferramentas.

O objetivo do programa é proporcionar o contato dos alunos com os recursos tecnológicos até então não disponíveis para eles nas escolas estaduais. Isso proporcionará o aprendizado de novas ferramentas para à elaboração de trabalhos escolares e consequentemente preparando estes alunos para um mercado de trabalho mais competitivo. Outra consequência positiva do programa é dar acesso aos alunos à grande fonte de informação que é a Internet com o objetivo de estimular a capacidade crítica deles, enriquecer pesquisas escolares e aumentar a compreensão de assuntos contemporâneos.

A Secretaria de Educação de Minas Gerais (SEEMG) firmou convênios com a iniciativa privada para viabilizar o programa. Em um deles um fabricante de processadores cederá ao governo esses dispositivos para a instalação nas máquinas adquiridas para o programa. Outros convênios prevêem a capacitação de professores em informática, liberação de softwares educacionais e facilidades para o professor comprar computadores a preços acessíveis. Está previsto ainda a implementação do programa Aluno Técnico em 50 escolas de ensino médio. Os estudantes serão treinados para montar e fazer manutenção de computadores, abrindo a eles oportunidade de ingressar no mercado de trabalho.

Com essas medidas alunos e professores passarão a ter à sua disponibilidade intrumentos de ensino antes inacessíveis, melhorando assim a qualidade do ensino público estadual.

## 1.3 Ambientação

O projeto escolas em rede da Secretaria de Educação de Minas Gerais (SEEMG), foi o grande motivador desse trabalho. As soluções escolhidas na implementação de todo o sistema visam atender às necessidades das secretarias na gerência de recursos e apoio às escolas.

A SEEMG está montando uma estrutura central em Belo Horizonte de apoio às escolas espalhadas por toda Minas Gerais. Essa unidade deverá monitorar o uso dos recursos computacionais, verificando se a estrutura instalada está sendo utilizada de forma adequada, e detectando causas de problemas para facilitar o suporte às essas escolas e direcionar futuros investimentos em infra-estrutura.

As escolas estaduais de Minas, frequentemente não dispõem de funcionários técnicos capazes para cumprir o papel de administrar os laboratórios que estão sendo instalados. Como esta tarefa fica centralizada cabe às escolas solicitar este suporte quando

necessário. Sendo esta estrutura muito grande e distribuída geograficamente por todo o estado, surgiu a necessidade de uma ferramenta que auxilie nesta tarefa e agilize a soluções dos problemas.

No capítulo 7 será apresentado uma análise do comportamento do sistema neste ambiente da secretaria da educação mostrando as dificuldades e as soluções encontradas para a implantação do sistema.

## 1.4 Descrição do Sistema

Como será utilizada principalmente por usuários não especialistas, essa ferramenta deve necessitar do mínimo de configuração e manutenção possíveis. Por este motivo as unidades que serão executadas nas estações de trabalho e nos servidores responsáveis por coletar as informações nessas máquinas foram desenvolvidas de forma que, após instaladas, necessitem apenas das informações de registro e da localização do servidor na tela de administração do servidor. Com isso passam a funcionar automaticamente sem intervenção de um administrador local e todas as configurações mais importantes possam ser feitas remotamente por meio do próprio sistema.

No nível superior às estações é executado a unidade intermediária, ou servidor de monitoramento, que é ao qual os agentes se conectam para enviar os dados coletados e requisitar configurações. O sistema desenvolvido permite a criação de uma hierarquia de unidades intermediárias. Isso foi feito para que futuramente caso o sistema comece a saturar, possam ser adicionados outros níveis na hierarquia melhorando a escalabilidade. Outra característica das unidades intermediárias é que elas permitem a adição de todos os componentes que estão nas centrais, exceto a capacidade de ativar os demais componentes, para que seja possível posteriormente a instalação de outras centrais de monitoramento.

Acima desse nível existe a unidade central que é onde os administradores de todo o sistema da secretaria acompanham as atividades nas escolas e prestam suporte às mesmas. Com o auxílio da interface web desenvolvida eles interagem com a unidade visualizando as variáveis coletadas, cadastrando e ativando as outras unidades ou configurando os agentes.

Pensando no modo de interação entre o administrador da central e o responsável da escola, a localização das escolas na interface de gerência é orientada pela posição geográfica da escola.

Os capítulos 4, 5 e 6 detalham os componentes, as tecnologias e as soluções empregadas no desenvolvimento do sistema.

## 1.5 Objetivos

Esse trabalho propõe o desenvolvimento de uma ferramenta de monitoramento remoto que, em conjunto com outras ferramentas, permita a criação de um sistema de gerenciamento remoto de uma rede de computadores grande e heterogênea.

O sistema deve permitir que um administrador identifique falhas nas máquinas e possa atuar nessas falhas remotamente facilitando a tarefa de administração com a oferta de mais informações sobre o ambiente e diminuindo o tempo de manutenção do sistema. Para estes usuários a interface deve ser rica em detalhes que auxiliem a execução das tarefas de gerência.

Para permitir que as ações de gerência sejam executadas em qualquer dispositivo monitorado a solução deve ser capaz de enviar e receber informações a esses equipamentos, mesmo com os limites impostos pelos provedores de acesso. As informações enviadas para os dispositivos devem permitir que sejam adicionadas novas variáveis para coleta, assim como permitir a alteração das configurações da variáveis já cadastradas. Como não é possível garantir a qualidade das conexões nem sua persistência o sistema deve ser capaz de armazenar as ações e valores coletados para serem transferidos no instante que as mesmas estiverem estabelecidas.

Visando a detecção prematura dos problemas que ocorrem na rede deseja-se que o sistema envie alarmes avisando sobre a ocorrência de problemas o mais rápido possível.

Com o objetivo de atender a milhares de estações enviando informações simultaneamente é desejável que a solução possua alternativas para filtrar a quantidade de informação trafegada. Assim como suportar a existência de pontos de controle e monitoração próximos às unidades monitoradas.

Os componentes que recebem informações de outras entidades devem ser capazes de identificar se elas pertecem à hierarquia de coleta e têm permissão para enviar dados para as instâncias superiores. Além disso toda comunicação entre as unidades do sistema deve ser segura.

Ao final deste trabalho também será apresentado um estudo de caso, sobre o com-

portamento da aplicação, em algumas escolas estaduais participantes do projeto escolas em rede do governo do estado de Minas Gerais. Como esse projeto é o motivador deste trabalho, e portanto, será importante observar o comportamento da ferremanta nesse ambiente e observar como as soluções escolhidas se comportam em situações reais.

## 1.6 Contribuições

A principal contribuição desse trabalho é o desenvolvimento de um sistema de gerência de redes para ser utilizado em grandes redes de computadores, dispersa em uma grande extensão territorial e com poucos administradores capacitados disponíveis para atender toda essa rede.

Em relação as outras ferramentas disponíveis este trabalho apresenta um conjunto de contribuições que possibilitam que os objetivos do sistema sejam atendidos. Entre elas está o fato da ferramenta possuir o código aberto. Isso junto com a maneira que o sistema foi desenvolvido proporciona uma flexibilidade importante que permite que máquinas com diferentes sistemas operacionais, e as propriedades diversas dessas máquinas, sejam monitoradas de forma centralizada por um mesmo conjunto de ferramentas.

Outra contribuição do sistema é permitir que com um mínimo de configuração inicial das unidades o sistema possa funcionar sem intervenção local dos usuários das máquinas no ambiente monitorado. Alterações nas configurações das unidades, intervenções nas máquinas e atualizações da unidades podem ser realizadas remotamente. Para isso basta que o administrador utilize a interface do sistema disponibilizada na Internet. Essas características do sistema permitem que poucos administradores experientes gerenciem de maneira eficaz máquinas diversas espalhadas em várias localidades.

Vale ressaltar como uma contribuição também a capacidade de atender a milhares de máquinas simultaneamente. Completando assim as principais características que tornam o sistema único e o capacitam para atender a demanda das grandes redes públicas brasileiras.

# 2 *Gerência de Sistemas*

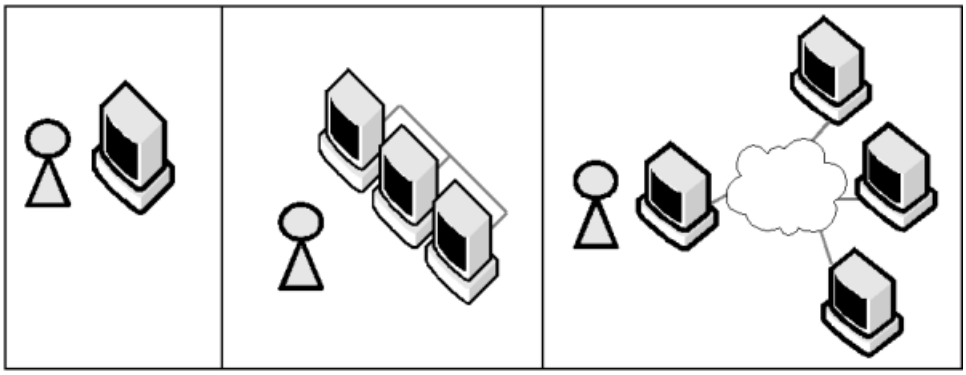

Figura 2.1: Evolução de um ambiente monitorado.

Uma pessoa responsável por uma máquina pode observar alguns parâmetros dessa máquina, como a utilização de memória ou do disco rígido, para avaliar a capacidade dessa máquina de realizar alguma tarefa ou solucionar algum problema. Nesse caso pode se realizar o monitoramento, ou observação dos parâmetros selecionados, e o controle, solução dos problemas, localmente.

Quando essa pessoa se torna responsável por várias máquinas ligadas a uma rede se torna mais eficaz realizar o monitoramento remotamente e estando elas no mesmo local ainda é possível realizar o controle diretamente sem auxílio de uma ferramenta apropriada. A partir do momento que essas máquinas são colocadas em locais separados se torna necessário que tanto o monitoramento quanto o controle sejam feitos remotamente e é nesse contexto que o sistema aqui apresentado deve atuar.

Mais formalmente, gerência de sistemas é a prática de *(a)* Monitorar e controlar uma rede existente de modo que os computadores permaneçam rodando e satisfaçam as expectativas dos usuários, *(b)* planejar extensões e modificações da rede para satisfazer as novas demandas, *(c)* incorporar novos elementos à rede, facilmente, sem interferir nas operações existentes [24].

Todos esses itens devem ser realizados ou facilitados por uma ferramenta de ge-

rência de redes. Normalmente essas ferramentas são encarregadas de monitorar e controlar a rede visando facilitar a tarefa dos administradores de planejar e expandir a rede.

A parte de monitoração da gerência de redes está preocupada com a observação e análise o estado e do comportamento dos sistemas e das subredes que compõe o conjunto a ser gerenciado. Enquanto que a parte de controle da gerência se ocupa com a modificação de parâmetros as atuações nesses sistemas [30].

Todas as cinco principais áreas funcionais da gerência de redes, definidas pela ISO [27] e amplamente utilizadas como guia para o desenvolvimento de um ambiente de gerência de redes, estão relacionadas tanto com as tarefas de monitoração quanto com o controle. Abaixo são apresentadas essas áreas.

**Gerência de falhas** - Gerência de falhas engloba detecção, isolamento e correção das falhas de um sistema. As falhas impedem o sistema de atender seus objetivos funcionais. A detecção de erros auxilia na identificação e correção das falhas. Algumas tarefas do gerênciamento de falhas são examinar relatórios e corrigir erros.

**Gerência de contabilização** - Contabilização habilita a cobrança pela utilização dos recursos e a identificação dos custos da utilização de determinados recursos. Para a gerência de contabilização são necessários gravar a utilização dos recursos e garantir a disponibilidade de um recursos para um usuário entre outras tarefas.

**Gerência de configuração** - A gerência de configuração deve identificar, controlar, coletar e prover informações que possibilitem a inicialização, configuração e a garantia do funcionamento contínuo dos serviços. Para uma ferramenta realizar a gerência de configuração de um sistema é necessário por exemplo que ela modifique parâmetros que controlam o funcionamento do sistema, coletar informações sobre o funcionamento de um recurso e obter alertas sobre a modificação do comportamento de um componente.

**Gerência de desempenho** - Permite avaliar o comportamento e a eficiência dos componentes do sistema. Para isso são necessários a geração de relatórios sobre a utilização dos componentes, prever o comportamento do ambiente em condições extremas e modificar o comportamento do sistema para atender uma demanda de desempenho.

**Gerência de segurança** - O propósito dessa função é suportar a aplicação de políticas de segurança para o sistema. Portanto um sistema deve informar se alguma política de segurança foi descumprida e criar e remover regras para utilização dos componentes.

## 2.1 Conceitos Básicos

Essa seção apresenta diversos conceitos e definições que serão utilizados ao longo da dissertação. As definições estão relacionadas com a gerência de redes e comunicação entre processos.

### 2.1.1 Informações Monitoradas

As informações disponíveis para um sistema de monitoração podem ser classificadas da seguinte forma [30]:

**Estática** - Essa informação caracteriza a configuração atual dos componentes, e esse tipo de informação varia muito pouco com o tempo. Exemplos: nome do processador e quantidade de memória.

**Dinâmica** - Esse tipo está relacionado com os eventos na rede, como mudanças de estado. Também podem estar associadas a utilização pontual de um recurso. São variáveis que modificam bastante com o tempo. Exemplos: Tráfego na rede e consumo de memória.

**Estatística** - As informações desse tipo geralmente são derivadas das informações dinâmicas, por algum cálculo matemático. Exemplo: Média de utilização da rede em um dia.

### 2.1.2 Modelo Básico de Gerência

Abaixo são descritos os cinco maiores componentes de um sistema de monitoração [30]:

**Aplicação de monitoração** - Esse componente inclui as funções de gerência que são visíveis para o usuário, como monitoração de desempenho, de falhas e contabilização.

**Gerente de coleta** - Esse módulo é responsável por coletar as informações disponibilizadas pelos outros componentes do sistema.

**Agente** - Esse módulo coleta e grava informações nos dispositivos monitorados e enviam as informações para o sistema monitor, geralmente o gerente de coleta.

**Objetos gerenciados** - Esse é a informação de gerência que representa um recurso e suas atividades. Nesse trabalho esses objetos também são chamados de variáveis monitoradas.

**Agente de monitoração** - Esse módulo gera análises estatísticas das informações coletadas.

Podemos identificar esses componentes na maioria das ferramentas de monitoramento, agrupadas ou isoladamente. E esse modelo é expansível para utilização em sistemas de gerência.

### 2.1.3 Consultas Periódicas e Comunicação de Eventos

As informações úteis para a monitoração da rede são coletadas e armazenadas por agentes e disponibilizadas por um ou mais sistemas de gerência. Duas técnicas podem ser utilizadas para buscar as informações nos dispositivos: Consultas Periódicas e Comunicação de Eventos [30].

**Consultas Periódicas** - Neste tipo de coleta o gerente entra em contato com os agentes para requisitar as informações armazenadas por ele. A requisição pode ser tanto por uma única variável ou uma lista. A requisão pode ser também tanto originada por uma pesquisa feita pelo usuário quanto por uma busca às informações armazenadas localmente no agente.

**Comunicação de Eventos** - Neste caso a iniciativa parte do agente enquanto o gerente fica à espera aguardando a chegada das informações. Um agente pode enviar informações periodicamente ou quando ocorrer um evento significante. A comunicação de eventos é útil para detectar eventos assim que eles ocorrem ou para monitorar valores que se alteram muito pouco.

### 2.1.4 Base de Informações de Gerência

Para realizar as funções planejadas, o programa de gerência de redes precisa acessar sua base de informações de gerência local ou MIB (*Management Information Base*) e as remotas. A MIB local em um agente contém informações necessárias para a gerência da rede, incluíndo informações que refletem a configuração e o comportamento dos dispositivo monitorado e parâmetros que podem ser usados para controlar esse dispositivo. A MIB local no gerente contém informações específicas para o gerente além do histórico ou resumo das informações dos agentes sob seu controle [30].

A maioria dos sistemas de gerência modernos mantêm uma MIB compacta que define um conjunto de objetos para todos os tipos de dispositivos que são responsáveis por gerenciar. Os administratores desses sistemas tipicamente transformam essas MIB's para um formato que o sistema possa usar. Uma vez carregada, os administradores podem se referir aos objetos usando apenas as identificações dos mesmos [26].

### 2.1.5 Protocolos Mestre/Escravo

O modelo mestre/escravo é um protocolo de comunicação onde, após estabelecida a comunicação, uma entidade passa a controlar a comunicação. Essa entidade, chamada de mestre, envia os comandos para outra entidade, escravo, que responde a essas requisições.

É possível haver um inversão de papéis nesse tipo de comunicação e a entidade que inicialmente era o escravo passar a enviar informações de controle. Para que isso ocorra em algum momento da comunicação o mestre deve informar ao escravo que ele abre mão do papel de mestre e se este aceita se tornar mestre.

Este mecanismo é utilizado principalmente em protocolos altamente distribuídos onde os componentes da rede não se conhecem e não podem ser localizar. Nesses casos um componente central espera o contato dos outros componentes e, quando este contato ocorre e as informações que o componente gostaria de enviar são transferidas, realizam essa inversão do controle para o componente central mandar os seus dados. Esse mecanismo é utilizado, por exemplo, em redes "*bluetooth*" [18].

# 3 *Trabalhos Relacionados*

Existem vários trabalhos feitos visando uma gerência de redes eficaz. Muitos desses trabalhos implementam mecanismos de monitoração e gerência próprios com o objetivo de otimizá-los para um ambiente específico.

Alguns trabalhos se baseam em padrões de monitoração, que muitas vezes são mais antigos e melhor suportados, para coleta de informações e implementam formas personalizadas de interação com o ambiente monitorado para realizar a gerência da rede.

Outras tecnologias oferecem padrões de gerência completos e que buscam ser genéricos o suficiente para serem utilizados nos mais diversos ambientes e interagir com os vários sistemas existentes.

Neste capítulo serão apresentados padrões e ferramentas que se propõem a implementar ou auxiliar a monitoração e gerência de redes de computadores.

## 3.1 Padrões de Gerência

A grande importância que os sistemas computacionais passaram a ter dentro das corporações criou a necessidade da elaboração de sistemas de gerência eficazes. Portanto, há algum tempo se discutem formas de estruturar esses sistemas e a definição de padrões que garantam a interoperabilidade entre os diversos produtos.

Aqui são apresentados dois desses padrões que ganharam grande apoio das entidades padronizadoras nos últimos tempos e são discutidos suas particularidades.

### 3.1.1 SNMP

Implicitamente a arquitetura SNMP *(Simple Network Management Protocol [19])* é uma coleção de estações de gerência e elementos de rede. As estações de gerência

executam as aplicações que monitoram e controlam o elementos da rede. Elementos de rede são dispositivos como estações de trabalho, roteadores, servidores, e outros, que possuem agentes de gerência responsáveis por realizar tarefas requeridas pelas estações de gerência [19].

O SNMP foi proposto como um protocolo simples e temporário que seria substituído no futuro por outros padrões. Porém este padrão ainda é muito utilizado hoje e possui algumas vantagens como [31]:

- Consome poucos recursos da rede;
- Fácil configuração e uso;
- É um padrão amplamente utilizado e está implementado em diversos dispositivos;

Um pacote SNMP transporta apenas sequências de octetos. O transporte desses dados pode ser feita por segmentos TCP (*Transmission Control Protocol*) ou UDP (*User Datagram Protocol*), porém as implementações padrão geralmente utilizam UDP. A justificativa para isso é que com o SNMP está monitorando uma rede que pode estar com problemas a exigência do estabelecimento de uma conexão, como é feito no TCP, pode sobrecarregar uma rede já saturada, mas a escolha do UDP faz com que os alarmes enviados pelos agentes possam ser perdidos e o gerênte não seja informado do problema.

Atualmente o protocolo se encontra na terceira versão. O mesmo foi evoluindo buscando atender as novas demandas dos ambientes corporativos. Devido ao seu foco na simplicidade e também por ter sido pensado como um protocolo temporário, o SNMP possui desvantagens que não puderam ser corrigidas nessas atualizações. São elas:

- A MIB *(Management Information Base)*, que é onde os objetos de gerenciados pelo SNMP são definidos, não define características sobre o comportamento desses objetos;
- Falhas de segurança existem em várias implementações do SNMP [1];
- O SNMP é um protocolo de comunicação atrelado à forma como as informações coletadas são armazenadas, isso impede a evolução do sistema porque, como citado anteriomente, a maneira como os objetos são definidos não suporta todas as características do mesmo.

**Estrutura da Informação de Gerência**

O primeiro passo para entender que tipo de informação um dispositivo pode prover é entender como essa informação é representada no contexto do SNMP. A SMIv1 *(Structure of Management Information Version 1)* faz exatamente isso: define precisamente como os objetos gerênciados são nomeados e especifica seus tipos [26]. O SMIv2 extende a árvore do SMIv1 adicionando o ramo snmpV2 na subárvore Internet, adicionando novos tipos para os objetos e algumas outras modficações.

A definição dos objetos gerenciados pode ser dividida em três atributos:

**Nome** - O nome, ou OID (*Object Identifier*), unicamente define um objeto gerenciado. Esses nomes normalmente em duas formas: numerico e legível. Em qualquer caso, os nomes são longos e inconvenientes. Nas aplicações SNMP, existem várias tentativas de melhorar a navegação pelo espaço de nomes convenientemente.

**Tipo e sintaxe** - Um tipo de um objeto gerenciado é definido usando um subconjunto da ASN.1 (*Abstract Syntax Notation One*). A ASN.1 é uma maneira de especificar como a informação é representada e transmitida entre gerentes e agentes, dentro do contexto do SNMP. Uma característica importante do ASN.1 é que a notação é independente de máquina.

**Codificação** - Uma única instância de um objeto é codificada em numa sequência de octetos usando o BER (*Basic Enconding Rules*). O BER define como os objetos são codificados e decodificados de maneira que eles possam ser transportados em uma rede.

***Object Identifier's***

Os objetos gerenciados são organizados numa hierarquia de árvore. Essa estrutura é a base do esquema de nomenclatura do SNMP. Um OID é composto por uma sequência de inteiros baseado nos vertices da árvore, separados por pontos(.). Porém, também existe a forma legível, essa forma nada mais é que uma série de nomes separados por pontos, representando os vértices da árvore. A figura 3.1 mostra alguns níveis dessa árvore.

Após a definição dos OID's é necessário a definição dos objetos. Toda definição de objeto segue o formato descrito no código de modelo 3.1.1.

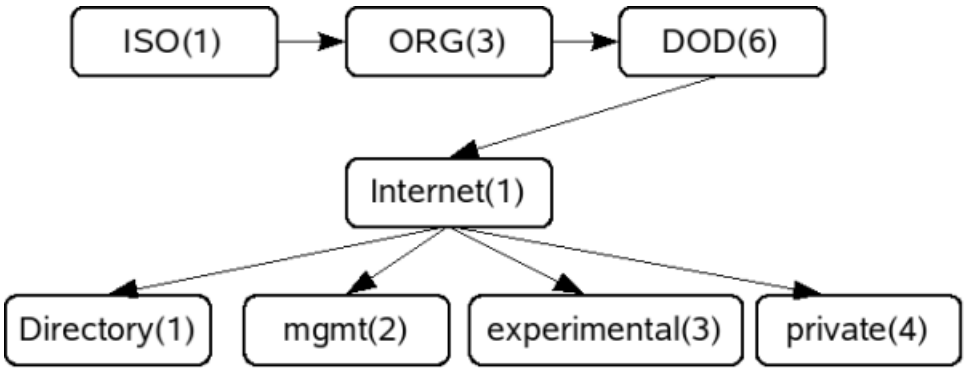


Figura 3.1: Hierarquia de objetos de gerência do SNMP.

**Código 3.1.1** Modelo de variável ASN.1 genérico

```
<name> OBJECT-TYPE
     SYNTAX <datatype>
     ACCESS <read-only, read-write, write-only, ou not-accessible>
     STATUS <mandatory, optional, ou obsolete>
     DESCRIPTION
         "Descrição do objeto gerenciado."
     ::= { <OID único que identifica o objeto> }
```

#### Operações suportadas pelo SNMP

As três operações básicas do SNMP são *snmpget*, *snmpset* e *snmpwalk*. O *snmpget* lê um valor de um dispositivo gerenciado, *snmpset* altera um valor em um dispositivo, e *snmpwalk* lê um conjunto de valores da MIB de um dispositivo [26].

Existe também a possibilidade de um agente enviar informações críticas para um gerente quando ocorrer alguma situação anormal no dispositivo monitorado. Esse mecanismo é chamado de *trap*. As *traps* que um agente pode gerar são definidas em sua MIB. Um agente pode gerar várias *traps* e é possível descobrir quais são as possíveis fazendo uma busca pelos termos *TRAP-TYPE (SMIv1)* e *NOTIFICATION-TYPE (SMIv2)*.

### 3.1.2 WBEM

O WBEM *(Web Based Enterprise Management)* [20] é um conjunto de tecnologias de gerenciamento e padrões de Internet desenvolvido para unificar o gerenciamento de ambientes computacionais distribuídos, facilitando a troca de dados entre tecnologias e plataformas diversas.

**Arquitetura**

O DMTF [4] (*Distributed Management Task Force*) é o grupo responsável por padronizar a arquitetura WBEM, a qual usa o CIM (*Common Information Model*) para modelar informações, XML (*Extensible Markup Language*) para gerenciar informações e o HTTP (*HyperText Transfer Protocol*) como protocolo de transporte [29].

Resumindo o WBEM pode ser definido como o conjunto de padrões que definem camadas de abstração que escondem a complexidade de acessar informações de gerência [25].

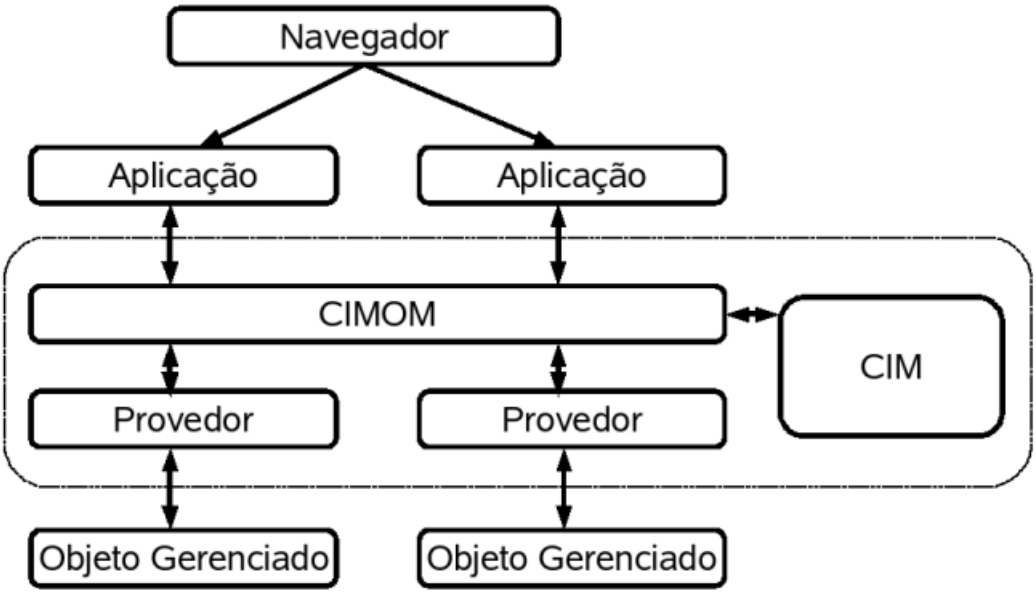


Figura 3.2: Arquitetura WBEM.

A figura 3.2 ilustra a arquitetura WBEM, que inclui:

- As aplicações acessam as informações enviando uma mensagem de requisição para o CIMOM em vez de acessar diretamente os provedores;
- O gerenciador de objetos CIMOM *(CIM Object Manager)* é responsável por rotear informações sobre eventos e objetos entre os componentes. Este componente responde às operações definidas pelo CIM como criar, apagar e alterar. Ele também verifica a sintaxe e a semântica das mensagens, e provê a segurança;
- E os provedores também chamados de agentes de instrumentação, obtêm as informações dos recursos e transferem para o CIMOM.

A seguir serão explicados com mais detalhes os padrões que compõe a definição do WBEM.

**Base de Informação de Gerência**

O CIM é um modelo de armazenamento para gerência de informação de sistemas, aplicações, redes e usuários [2]. O propósito do CIM é oferecer um formato comum para que estas informações sejam usadas por diferentes sistemas. O CIM é um modelo conceitual independente de implementação ou linguagem e oferece uma abordagem orientada a objetos para informações de gerência física e lógica.

A linguagem de modelagem proposta no padrão é orientada a objetos e permite uma definição bastante detalhada dos objetos monitorados. Essa modelagem possui os conceitos de herança e relacionamentos entre classes, ficando mais explícita a relação entre os objetos gerenciados.

A especificação do CIM é a definição de todos os termos, métodos e conceitos usados para produzir o modelo de dados usado pelo CIM. Essa especificação descreve como o modelo de dados é representado utilizando arquivos MOF *Managed Object Format* ou utilizando XML (xmlCIM). Um arquivo MOF é um arquivo texto que usa uma sintaxe específica para representar as definições CIM. A seguir essa representação será descrita com mais detalhes. A representação xmlCIM define elementos XML em uma DTD (*ocument Type Definition*), que pode ser usada para representar objetos no formato XML. O mecanismo de transporte sobre HTTP da representação XML permite que as implementações CIM interajam de uma maneira aberta e padronizada [25].

A representação dos objetos feita pelo CIM se utiliza de elementos básicos com diferentes relacionamentos entre si para modelar objetos que existem no mundo real. Esses elementos e relações estão descritos abaixo:

**Classes** - Uma classe define a natureza de um objeto no mundo real. Ela é um protótipo que define propriedades e os métodos comuns à um determinado tipo de objeto [2].

**Propriedades** - São valores que representam uma característica de uma classe. Uma propriedade é única e tem escopo válido dentro de uma classe. Um tipo especial de propriedades são as referências. Referências identificam as relações que as associações definem entre as classes.

**Métodos** - Um método é uma operação que pode ser chamada. A assinatura de um método inclui o seu nome, o tipo de retorno e opcionalmente parâmetros de entrada e retorno. Um método geralmente representa o comportamento de uma classe.

**Associação** - As associações são um tipo de classe que contêm uma ou mais referências. Associações representam relações entre duas ou mais classes.

**Indicadores** - São uma representação da ocorrência de um evento. Instâncias de uma indicação são transientes e não podem ser coletadas. Elas só são recebidas se inscrevendo para recebê-las antes delas ocorrerem.

**Qualificadores** - Proveêm informações adicionais sobre classes, associações, indicações, métodos, parâmetros para métodos, propriedades ou referências.

### Managed Object Format

As informações que gerência do CIM potencialmente poderia ser representada que diversas formas. O especificação do CIM define uma linguagem baseada na linguagem de definição de interface (IDL) chamada MOF (*Managed Object Format*) [2].

A sintaxe do MOF é uma maneira de fazer a definição dos objetos de forma textual. Arquivos com definições de classes nesse formato geralmente são usados para acrescentar novos classes aos esquemas CIM definidos em um CIMOM.

Um arquivo MOF geralmente é composto de uma série de definições de classes e instâncias. No código 3.1.2 é dado um exemplo.

### WBEM Query Language

Com o propósito de representar os objetos reais gerenciáveis, é importante que o CIM tem uma técnica poderosa para recuperar as informações que ele armazena [25]. A *WBEM Query Language* (WQL) é um subconjunto do ANSI SQL com algumas mudanças semânticas para realizar operações sobre o repositório CIM. Diferente do SQL o WQL é uma linguagem apenas de leitura, ela não permite modificar, alterar ou inserir informações nesse repositório. A tabela 3.1 mostra o mapeamento das entidade do SQL para o WQL [13].

Os clientes WBEM utilizam o WQL para pesquisar e filtrar informações. O CIMOM é encarregado de redirecionar as consultas que recebe dos agentes para o provedor que

**Código 3.1.2** Definição de uma classe CIM no formato MOF

```
class CIM_USBHub : CIM_USBDevice {

      [Description (
          "Indicates whether power to the Hub Ports is individually or "
          "gang-switched. If this value is FALSE, power is switched "
          "individually for each Port. If this value is TRUE, power is "
          "switched to all Ports on the Hub at once.")]
   boolean GangSwitched;

      [Description (
          "Number of downstream Ports on the Hub, including those "
          "embedded in the Hub's silicon. Individual USBPorts are "
          "associated with the Hub using the USBPortOnHub association.")]
   uint8 NumberOfPorts;
};
```

| SQL Conceito | WQL Representação |
|---|---|
| Tabela | Classe CIM |
| Linha | Instância CIM |
| Coluna | Propriedade CIM |

Tabela 3.1: Conceitos SQL mapeados no WQL

possui a informação solicitada. Com a linguagem podem ser solicitadas informações contidas em uma instância de uma classe ou de um conjunto de classes. O exemplo abaixo mostra um consulta WQL, utilizada para recuperar a propriedade *LoadPercentage* da classe *Linux_Processor*.

```
select LoadPercentage from Linux_Processor;
```

### WMI

O WMI [16] (*Windows Management Instrumentation*) é uma implementação da Microsoft para as tecnologias WBEM. Esta ferramenta oferece uma maneira padrão de acessar e coletar informações de sistemas windows sendo utilizada como base para diversas aplicações da microsoft para gerenciamento de redes windows.

O WMI segue exatamente o WBEM, pois não utiliza mensagens cimXML sobre o protocolo HTTP definido no padrão para transporte de informações. Em seu lugar

utiliza o DCOM (*Distributed Component Object Model*) uma tecnologia proprietária que dificulta a comunicação direta do WMI com outras ferramentas baseadas no WBEM.

## 3.2 Ferramentas

Existem várias ferramentas que permitem monitoramento remoto no Linux. Um grande número dessas ferramentas é baseada no protocolo SNMP. Uma ferramenta que se destaca nesse cenário é o Nagios [8], que utiliza uma arquitetura própria, mas que permite a interação com outros padrões.

O suporte à gerência remota é um pouco limitado nessas ferramentas e poucas delas oferecem uma solução integrada de monitoramento e atuação eficaz. O Tivoli é uma dessas ferramentas que oferece essa facilidade, porém é uma ferramenta de código fechado voltado para o uso em grandes corporações.

Uma característica da maioria das ferramentas é a vocação para monitorar redes locais, assumindo que as máquinas monitoradas estão sempre acessíveis a partir da central de monitoramento. Isso não é verdade no ambiente que está sendo tratado onde as unidades que devem ser monitoradas estão dispersas e protegidas por mecanismo que impedem o acesso direto a essas máquinas partindo de um ponto qualquer da Internet.

Abaixo são apresentadas essas ferramentas e algumas limitações das mesmas. Existem várias outras soluções disponíveis no mercado, baseadas nos padrões abertos como os descritos acima ou fechados, porém as ferramentas apresentadas estão entre as mais utilizadas e ilustram bem o contexto atual.

### 3.2.1 Nagios

O Nagios é uma ferramenta poderosa de gerência de redes que possui mecanismos próprios de monitoramento e gerência. Esta é fragmentada em vários componentes responsáveis por realizar tarefas determinadas no ambiente monitorado. Com isso é possível obter ferramentas simples e eficientes no que elas se propõem a fazer.

Esta ferramenta basicamente executa um conjunto de operações para coleta dos objetos pré-configurados localmente nos intervalos em que elas foram configuradas para isso. Os objetos são representados de uma forma particular em arquivos de configuração.

Essa funcionalidade sozinha não é suficiente para administrar uma rede de computadores, pois é inviável, e às vezes impossível, buscar informações de outras máquinas com a execução local de uma tarefa. Para isso o Nagios possui uma extensão chamada *Nagios Remote Plugin Executor* que se encarrega de buscar informações que estão disponíveis numa máquina remota, esse mecanismo será apresentado com mais detalhes no tópico específico.

Mas como existem máquinas que não são acessíveis a partir da central de monitoramento, foi criada uma outra extensão que envia periodicamente as informações configuradas num dispositivo remoto para a central o NSCA. Esse mecanismo também será detalhado em outro tópico.

### Objetos

Um objeto do Nagios descreve uma unidade especifica como um máquina, um serviço ou até mesmo um comando e também o grupo ao qual ele pertence [17]. A definição dos objetos também permite a definição de heranças, dessa forma é possível definir vários objetos com características semelhantes sem duplicar as definições.

As definições dos objetos do Nagios seguem o modelo 3.2.1. Onde o tipo do objeto pode ser: *timeperiod*, *command*, *contact*, *contactgroup*, *host*, *service*, *hostgroup*, *servicegroup*, *hostdependency*, *servicedependency*, *hostescalation*, *serviceescalation*, *hostextendedinfo* ou *serviceextendedinfo*. E os parâmetros possíveis estão associados a cada objeto.

**Código 3.2.1** Modelo de definição de um objeto do Nagios

```
define <tipo do objeto> {
   <parâmetro> <valor>
   <parâmetro> <valor>
   ...
}
```

Para que o processo de monitoração ocorra deve-se associar um objeto serviço (*service*) a um objeto que define uma máquina (*host*). Essa definição deve conter os intervalos de coleta, períodos em que deve ser coletada a variável, o comando associado e outros parâmetros.

### *Nagios Remote Plugin Executor*

O Nagios possui um mecanismo que permite a execução de plugins remotamente chamado *Nagios Remote Plugin Executor* (NRPE). Para a utilização desse recurso é necessário além da instalação dos plugins desejados na máquina remota que [17]:

- O processo *nrpe* deve ser excutado;
- O serviço inet esteja configurado com privilégios de administrador;
- A extensão *check_nrpe* deve estar intalado na central de monitoramento.

O processo *nrpe* possui um arquivo de configuração local que define quais comandos podem ser acessados na máquina em que está instalado. A extensão *check_nrpe*, por sua vez, é configurada para apenas realizar a busca pelos objetos disponíveis na máquina remota [22]. Note que o processo de configuração é separado e independente, não existe garantia de que os objetos que a extensão irá buscar nas estações remotas estão realmente disponíveis. As conexões realizadas para transferência de informações pode ser autenticadas e criptografadas.

### *Nagios Service Check Acceptor*

O Nagios oferece um mecanismo para que os dispositivos monitorados possam enviar informações que foram coletadas sem que o servidor tenha que buscar por elas através de consultas periódicas. Esse mecanismo é chamdo de *Nagios Service Check Acceptor* (NSCA). O NSCA permite que máquinas que não são alcançáveis pelo servidor sejam monitoradas e permite também a criação de uma hierarquia de coleta com vários servidores Nagios.

O funcionamento dessa extensão é semelhante ao do NRPE porém com o sentido invertido. Os objetos que serão enviados pelo modo passivo devem ser criados no servidor que irá receber os valores e devem possuir um parâmetro informando que o método de coleta desses objetos é passivo. No outro lado da conexão, onde os valores do objetos serão obtidos, os comandos responsáveis por realizar a coleta dos dados devem ser agendadas para serem executadas periodicamente e os valores devem ser enviados pelo comando *send_nsca*.

### 3.2.2 Tivoli

Esta ferramenta da IBM provê soluções de gerência inteligente de infraestruturas. O Tivoli, através da monitoração dos vários componentes de um sistema, provê visões integradas para gerência e otimização desses recursos. A ferramenta permite ainda a definição de políticas de alocação de recursos, segurança, armazenamento e gerência [5].

Alguns de seus componentes são baseados no padrão de gerência WBEM descrito anteriormente e compartilha com essa tecnologia os mecanismos de representação dos objetos e os mecanismos de comunicação. Em cima dessa base o tivoli construiu componentes capazes de realizar uma gerência de alto nível dos componentes da rede. Essas características permitem que o Tivoli ofereça soluções integradas e automatizadas para gerência de toda a infraestrura de uma grande empresa.

Os vários componentes do tivoli oferecem suporte a gerência a um conjunto de aplicações responsáveis por uma parte da infraestrutura da empresa como por exemplo os bancos de dados, os servidores de páginas, e outros. Essas ferramentas específicas se integram formando um sistema único de gerência da rede.

As desvantangens dessa solução é que, por ser uma solução complexa, possui um alto custo. Outra limitação é o código ser fechado o que limita a possibilidade de personalização da solução, pois exige que essa tarefa seja executada por pessoas com grande conhecimento da ferramenta.

### 3.2.3 OpenNMS

O OpenNMS consiste em um projeto de código aberto suportado por uma comunidade e também uma organização que oferece serviços, treinamento e suporte [9]. O programa possui uma estrutura centralizada de coleta de informações através de consultas enviadas aos agentes monitorados.

Esta ferramenta de monitoramento possui como diferencial o descobrimento automático de recursos em uma rede. Durante intervalos regulares a rede é varrida em busca dos serviços disponíveis. Caso um serviço desapareça da rede o intervalo de consultas é reduzido visando detectar o mais rapidamente o retorno do mesmo.

Esse mesmo mecanismo é utilizado para coleta de variáveis monitoradas pelo protocolo SNMP. É possível configurar comandos para serem executados quando alguma

dessas variáveis atinge um limite e alarmes podem ser enviados para uma determinada pessoa ou grupo de pessoas responsáveis pela manutenção do serviço.

Além dessas funcionalidades existe a opção de criar um base de dados de inventário com os componentes da rede e relatórios estatísticos das variáveis coletadas.

# 4 *Arquitetura*

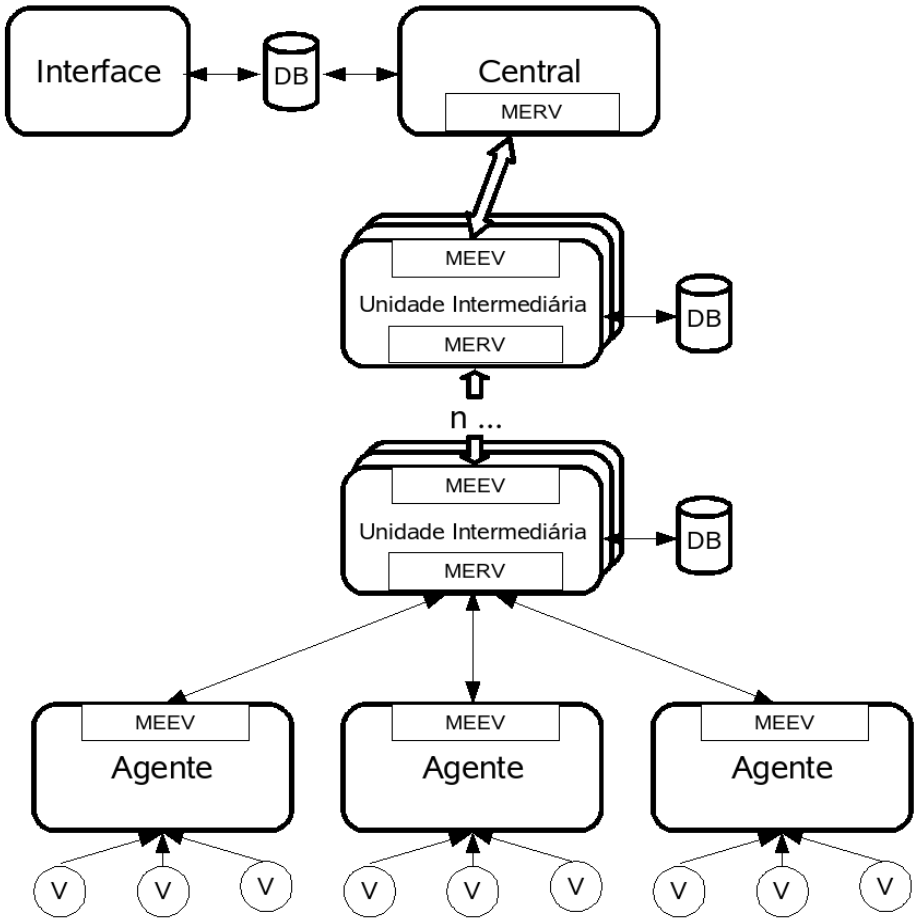


Figura 4.1: Arquitetura do Sistema de Monitoramento da SEEMG.

O sistema implementado possui uma unidade central de monitoramento responsável por armazenar as informações de todo o ambiente monitorado. Essa unidade estará disponível na Internet e deve possuir um endereço fixo conhecido por todos os componentes da rede a ser monitorada. Além dessa unidade existem unidades intermédiárias responsáveis por filtrar e repassar as informações coletadas pelas unidades de coleta ou agentes, como descrito na figura 4.1.

O sistema provê um mecanismo de registro que permite o controle das cópias ins-

taladas e a identificação dos componentes monitorados. São utilizados pares de chaves de ativação e registro para identificar os componentes do sistema e garantir que eles estão autorizados a entrar na hierarquia de gerência.

Os agentes são capazes de identificar e quantificar os componentes físicos instalados no ambiente computacional, assim como, monitorar a utilização desses recursos, enviando essas informações para a unidade central numa frequência definida. A aquisição dessas informações pode ser feita acessando as MIBs de outros padrões de gerência como WBEM ou SNMP.

Os componentes do sistema podem ser configurados remotamente permitindo assim o controle das variáveis a serem monitoradas. Esse mesmo sistema permite que os componentes executem uma determinada tarefa no sistema monitorado, possibilitando uma atuação controlada por uma entidade remota, modificando este ambiente. Existe a possibilidade de se realizar transferências de arquivos entre a entidade de controle e os componentes do sistema de monitoramento.

Foram acrescentadas unidades intermédiarias ao sistema desenvolvido que podem oferecer formas de realizar a gerência nos níveis mais baixos da hierarquia, quando isso for possível ou necessário, visando a diminuição do tempo de respostas às falhas ou a melhoria da escalabilidade do sistema.

As máquinas monitoradas estarão organizadas na interface de acordo com sua localização geográfica, que não está necessariamente relacionada com a hierarquia de gerência. Porém todas as unidades intermediárias devem possuir uma forma de se conectar à Internet e conhecer o endereço da unidade superior.

As unidades central, intermediárias e os agentes foram implementados em C/C++, com a intenção de implicar ao ambiente gerenciado o menor impacto possível. O armazenamento das informações é feito em banco de dados MySQL [7] nas unidades central visando obter maior capacidade de gerência e garantia de integridade dos dados. Nos agentes foi escolhido o gerenciador de banco de dados SQLite [14] por poder ser integrado à aplicação e ser mais simples. E a interface está implementada em HTML, Php e Java objetivando um aplicação que funcionasse bem em diversas plataformas.

## 4.1 Unidades do Sistema

As unidades do sistema correspondem à elementos computacionais organizados em uma estrutura hierárquica geográfica. Para cada unidade podem ser definidas variáveis a serem coletadas, ou objetos monitorados, e a essas variáveis estão associados o método e o período de coleta.

Essas unidades são identificadas por uma chave de ativação única, que está associada às suas informações geográficas que são utilizadas para representar essa unidade em mapas. Esses mapas são utilizados para facilitar a localização das unidades monitoradas.

A hierarquia de monitoramento é composta por três unidades distintas, que são: unidade central, unidade intermediária e a unidade de coleta ou agentes.

### 4.1.1 Unidade Central

A unidade central, ou apenas central daqui para frente, é responsável por controlar todas as outras unidades e armazenar as informações enviadas por elas. Essas informações são inseridas em um banco de dados mysql e são manipuladas de forma a minimizar o espaço de armazenamento necessário. Mais a frente, onde é descrita a coleta de variáveis, esse processo será descrito com mais detalhes.

Esta unidade está ligada à uma interface de administração que exibe as informações coletadas por todas as unidades da hierarquia. Essa interface também é responsável por prover um meio do administrador interagir com a unidade central ditando quais ações devem ser tomadas para cada unidade.

Outra função dessa unidade é controlar o registro e o envio de informações das outras unidades. Para isso a unidade central recebe as informações cadastrais de uma unidade e gera a ativação da mesma. A central recebe da interface o conjunto de variáveis que uma unidade deve monitorar. Após gerar essas duas informações o sistema encarrega a central de enviá-las para as outras unidades.

Toda a comunicação entre a interface e a central ocorre através do banco de dados descrito em detalhes na subseção 6.1. Sempre que uma unidade se comunica com a central, a central deve checar se há informações novas para esta unidade no banco e enviá-las imediatamente na mesma conexão.

**Coleta das Variáveis**

Para qualquer variável coletada em um agente com uma chave de ativação, a coleta é uma n-upla: (Chave,variável,instante,valor).

O algoritmo de coalescência de blocos é seguido na inserção de um valor na base. Neste algoritmo, descrito no código 4.1.1, valores são inseridos no banco de dados apenas quando o valor recebido é diferente do valor armazenado. Quando estes valores são iguais o momento da chegada da última coleta é atualizado no banco, todo o período que uma variável se manteve com determinado valor é representado por duas entradas.

**Código 4.1.1** Algoritmo de coalecência de blocos

```
Para uma unidade k
Seja Vk o último valor armazenado para a Variável;
Seja Valor o valor coletado para a Variável;

Se ∄Vk, insere(Vk,Flag ← 0);
Se ∃Vk então
    Se Vk = Valor então
        Se Flag = 0 então
            insere(Vk,Flag ← 1);
        Se Flag = 1 então
            atualiza(instante);
    Se Vk ≠ Valor então insere(Vk,Flag ← 0);
```

Para as unidades hierarquicamente superiores ao agente, é aplicada uma função de sumarização onde os valores de uma variável, presente em várias unidades, são agregados nessas instâncias. A função de quantificação (ou contadora) está descrita no código 4.1.2.

Neste processo são criadas automaticamente novas variáveis que representam a agregação de valores que as unidades abaixo na hierarquia armazenam para uma variável.

A função de média foi derivada a partir do cálculo da média para valores em intervalos regulares:

$$x_1, x_2, x_3, ..., x_n \rightarrow \overline{x_n} = \frac{\sum_{i=1}^{n} x_i}{n}$$
$$\overline{x_{n+1}} = \frac{\sum_{i=1}^{n} x_i + x_{n+1}}{n+1}$$
$$\frac{\sum_{i=1}^{n} x_i + x_{n+1}}{n+1} = \frac{n\overline{x_n} + x_{n+1}}{n+1}$$

**Código 4.1.2** Algoritmo de quantificação

```
Para a unidade i
Para variável tipo texto
    Seja Qi = Q_Variavel_Valor associada a variável e o valor
Para outras variáveis
    Seja Qi = Q_Variavel
Se ∄Qi então insere(Qi)

Qi = Qi_anterior − 1
Qi = Qi + 1
```

A função de soma é bem parecida com a anterior com a diferença de ser aplicada apenas às variáveis numéricas e ser retirado da variável superior o valor anterior e adicionado o atual.

A função de união é aplicada apenas às variáveis do tipo lista, e esta função nada mais é do que uma função de união de listas. Calculada como no exemplo abaixo:

$$\{(2,verde);(1,azul)\} \cup \{(2,azul);(1,preto)\} = \{(3,azul);(2,verde);(1,preto)\}$$

**Mecanismo de Depuração**

Para a depuração do sistema foi utilizado um mecanismo para habilitar/desabilitar a exibição de informações de depuração com os processos em execução sem ter que reiniciá-los. Para isto foram usados o esquema de SIGNAL (SIGHUP) do Unix e uma técnica de reabertura dos arquivos de saída de erros e padrão. Ao SIGHUP, o servidor lê a configuração de depuração de um arquivo pré-definido obtendo os níveis de debug e o arquivo para saída das informações. Isto é bem útil na depuração de processos em máquinas remotas utilizando uma sessão telnet ou ssh e redirecionado a saída para o terminal virtual criado ou para um arquivo para análise posterior.

Opções usuais de depuração incluem a exibição das transições nas máquinas de estado da implementação dos protocolos, dump de pacotes recebidos e transmitidos, controle de timeouts, operações em listas e filas de ações e operações na base de dados SQL.

Considerando a estrutura hierárquica pai/filhos processos servidores (central e intermediário), os filhos novos herdam as opções de depuração, sejam passadas nos argumentos do disparo inicial do servidor quanto através do mecanismo do SIGHUP.

### 4.1.2 Unidade Intermediária

Uma unidade intermediária tem o papel de repassar as informações vindas dos agentes ou de outras unidades intermediárias para uma unidade superior na hierarquia. Essa unidade pode ser uma central ou outra intermediária. Na comunicação dessa unidade com uma superior ela é encarregada de requisitar todas as informações destinadas aos agentes sob sua responsabilidade.

As unidades que se conectam à unidade intermediária devem considerá-la como uma unidade central. Para isso essa unidade replica várias funções da central e isso permite que seja criada uma hierarquia de gerência. Além disso é possível anexar uma interface a essa unidade com poder para gerênciar toda estrutura abaixo dela.

Porém esse unidade têm a limitação de não prover a capacidade de registrar nenhuma unidade. Essa função é exclusiva da central. Uma função específica dessa unidade é agregar e repassar as informações dos agentes para à central, essa função é desempenhada por um mecanismo de filas de mensagens que será descrito mais a frente.

A mesma estrutura de banco de dados da central (subseção 6.1) está disponível na unidade intermediária que permite que ela exerça as funções da central. Todas as informações das unidades abaixo desta podem ficar armazenadas no seu banco de dados local além de serem enviadas para uma unidade superior.

Essa forma de armazenamento suporta a criação de uma esquema de gerência em que a qualidade e a quantidade de informações pode ir aumentando caso se queira melhorar a escalabilidade do sistema.

#### Mecanismo de Filas de envio de informações

Quando um agente ou uma unidade intermediária envia informações para uma unidade intermediária, transmite também, no início da conexão a sua chave de ativação. Essa chave identifica e valida a unidade que está iniciando a transmissão.

As informações recebidas pela unidade intermediária são armazenadas no seu banco de dados local para uma possível consulta futura. Mas além disso as informações recebidas devem ser repassadas para outras unidades. Como as conexões entre as unidades não estão ativas sempre, no momento em que uma unidade envia dados para uma unidade intermédiaria possivelmente ela não estará enviando dados para

uma unidade superior, por isso, todos esses dados recebidos são enfileirados na fila de transmissão de dados.

Os dados inseridos nessa fila são identificados pela chave de ativação da unidade que as enviou. Um bloco de informações consistente termina com um pacote tendo uma marcação para que o módulo intermediário possa enviá-lo. Assim toda a unidade que tenha dados na fila e possua pelo menos um pacote com a marcação de transmissão concluída possui informações que devem ser enviadas para outras unidades.

O mesmo processo ocorre quando a informação é originada na central ou numa unidade intermediária e deve seguir para um agente. Ao passar por uma unidade intermediária ela é enfileirada seguindo o mesmo processo descrito anteriormente.

### 4.1.3 Unidade de Coleta

As unidades de coleta, ou agentes, são reponsáveis por coletar as variáveis solicitadas e enviar para as unidades superiores. No momento indicado o agente executa o comando que retorna a variável e armazena esse valor num banco de dados sqlite. A implementação dessa unidade foi divida em dois programas independentes. Um responsável por controlar a coleta das variáveis e o outro cuida do envio dos dados.

Este banco (ilustrado na figura 6.4) é utilizado para armazenar as configurações e as informações de gerência do agente. Outra função do banco é para arquivar as variáveis coletadas enquanto elas não são enviadas. Isso permite que os agentes continuem coletando dados enquanto as escolas estão sem conexão com a Internet por exemplo. Além da chave de registro, informações básicas sobre a localização da máquina e alguns dados sobre o responsável pela manutenção da mesma, deve ser informado para o agente qual é o endereço da unidade intermediária em que ele deve conectar para o recebimento das configurações e o envio das coletas. Esse endereço geralmente é o do servidor local que possui uma unidade intermediária instalada.

A MIB de um agente contém informações sobre a maneira que deve ser realizada a coleta como por exemplo, o comando a ser executado, o período entre as coletas e o momento da próxima execução. Também são armazenados nessa MIB os limites aceitos para essa variável. Essa última informação visa suportar futuramente que os próprios agentes enviem alertas comunicando se esses limites forem ultrapassados.

No processo de coleta são verificadas as variáveis que devem ser coletados no momento. Após feita a coleta, o valor da próxima execução é atualizado e o momento

que o agente deve reiniciar o processo de coleta também. Existem alguns modelos de comandos configurados para execução de buscas por objetos WBEM e SNMP. Os objetos disponíveis pelo CIMOM, por exemplo, são obtidos com a execução de chamadas WQL e as disponíveis na MIB SNMP podem ser coletadas com a execução de um *snmpget*.

O agente desenvolvido é flexível o suficiente para ser executado em outros sistemas operacionais. Atualmente o mesmo código utilizado na implementação linux pode ser compilado utilizando o cygwin [3] e ser executado no windows. Além disso não existe nenhuma parte essencial para o funcionamento do agente que seja específica para um sistema operacional, portanto qualquer implementação da máquina de estados do protocolo de comunicação do cliente pode se comunicar e enviar informações para o sistema de gerência.

## 4.2 Interface de Administração do Sistema

A interface de administração do sistema é responsável por acionar os diversos mecanismos de controle e configuração do sistema como o cadastro de componentes, configuração das variáveis que devem ser coletadas por um agente, a forma de exibição das mesmas e o agendamento das ações que deverão ser executadas nesses agentes.

Esta interface interage com a unidade central exclusivamente manipulando o banco de dados comum nos dois componentes.

### 4.2.1 Cadastro de Unidades

O cadastro das unidades é feito associando uma chave de ativação à essa unidade. Essa chave é gerada por uma função externa ao sistema sendo executado pela interface por meio de uma chamada remota a essa função.

Após a obtenção dessa chave, o operador informa ao sistema as coordenadas geográficas do novo componente, e essas informações são inseridas no banco de dados, tornando-se assim acessíveis pela unidade central. A figura 4.2 mostra uma tela de cadastro de variáveis.

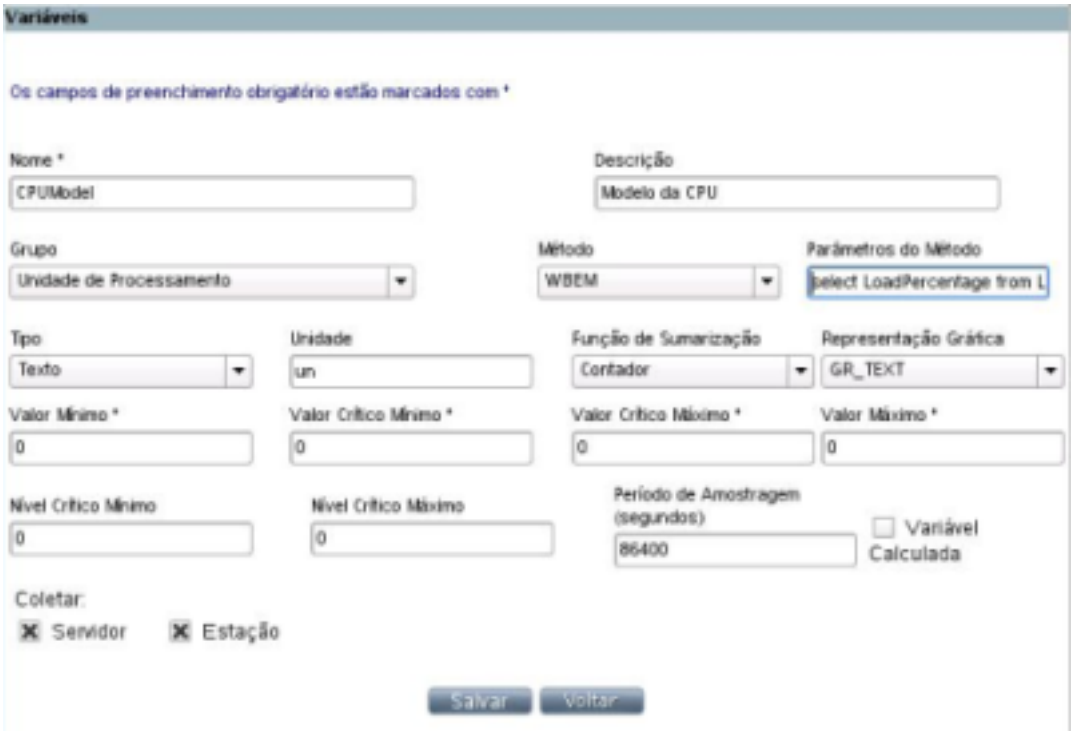


Figura 4.2: Tela de cadastro de variáveis.

### 4.2.2 Variáveis

Devem ser definidas todas as variáveis a serem coletadas, incluindo na sua identificação padrão: nome, tipo, forma de exibição gráfica, valores máximos e mínimos, função de sumarização, método de coleta e seus argumentos. Para cada unidade podem ser associadas variáveis com parâmetros específicos.

Uma variável representada na base têm consigo associado um código que será executado no agente, e o valor de retorno desse código é o valor da variável. Isso torna possível que a mesma variável seja coletada de forma diferente em ambientes distintos.

O tipo da variável pode ser numérico, textual, tuplas ou arquivos. Os três primeiros tipos sofrem um tratamento para serem exibidas nos níveis superiores da hierarquia geográfica. A essas variáveis são aplicadas as funções de sumarização.

A função de sumarização é utilizada para transportar e agregar o valor de uma unidade para as unidades geograficamente superiores. Funções típicas são: média, soma, contador, etc.

Cada coleta será uma $n$-upla contendo a identificação da unidade, o nome da variável, o instante de coleta e seu valor.

| Inventário de Hardware | Sistema Operacional | Serviço |
|---|---|---|
| Modelo da CPU | Carga do Processador | Sites mais Acessados |
| Tempo de Atividade | Estações Conectadas no Servidor | Bytes acessados por site |
| Tamanho da Partição | Usuários | Taxa de transmissão da Internet |
| Área livre na Partição | Total do swap | Tráfego por tipo de Protocolo |
| Tamanho da memória | Swap livre | Acessos à maquina no instante |
| Memória livre | Sistema Operacional | |
| Temperatura da CPU | | |
| Imagens da Câmera | | |

Tabela 4.1: Variáveis Coletadas

### Variáveis Coletadas

Atualmente o sistema em testes é capaz de coletar as variáveis presentes na tabela 4.1, mas o sistema permite que sejam adicionadas novas variáveis a qualquer momento através da configuração da nova variável na interface de administração.

O sistema interage com dois protocolos de gerência diferentes para coletar essas variáveis o WBEM e o SNMP. Além desses métodos é possível a execução direta de comandos. Posteriormente serão acrescentadas mais algumas outras variáveis utilizando os mesmos dispositivos validando a flexibilidade da ferramenta. No caso específico da SEEMG essas variáveis foram escolhidas para monitorar quantas e quais máquinas adquiridas pelo governo estão em uso e para qual fim estão servindo.

### Exibição dos Valores Coletados

Navegando pela interface, quando se chega ao nível em que se encontra um agente sendo executado, as variáveis que esse agente coleta são exibidas pela interface de gerência. Existem duas formas de exibição de uma variável. Uma é a exibição direta do valor, que utiliza um formatação simples HTML com um código PHP/AJAX para isso. A segunda utiliza a ferramenta de geração de relatórios JasperReports [6] para renderizar alguns tipos de gráficos. Uma tela de exibição de variáveis está ilustrada pela figura 4.3.

As variáveis que são exibidas em níveis superiores da interface, que são variáveis obtidas por meio da agregação dos valores coletados pelos agentes, também são exibidas da mesma forma.

Na exibição direta da variável a última entrada inserida no banco de dados é recuperada e os valores do nome, valor associado e horário de coleta da variável são

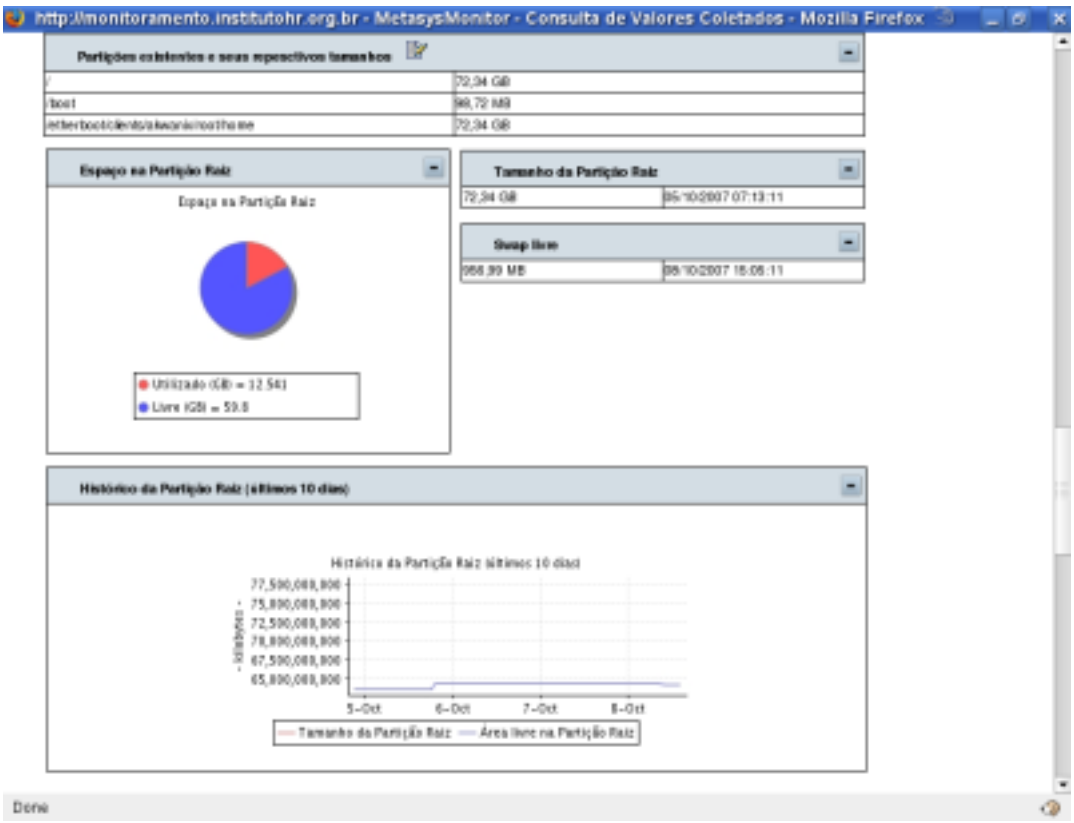


Figura 4.3: Tela de exibição de valores.

exibidos na tela. Para variáveis do tipo lista, a exibição é um pouco diferente, neste caso, para cada componente da lista é exibido o índice da lista seguindo dos valores que associados àquele indice.

Uma variável númerica também pode ser exibida graficamente. Os tipos de gráficos disponíveis no sistema serão divididos aqui em três grupos distintos, para uma explicação mais clara, de acordo com a quantidade de informação exibidas por eles. Os três tipos são:

**Gráficos simples** - Os gráficos de termômetro e velocímetro podem ser considerados simples pois exibem apenas o último valor coletado de uma variável númerica, quase como na exibição direta, porém acrescentando algumas informações sobre os limites aceitáveis. O gráfico de termômetro se diferencia do velocímetro por ter um limite crítico apenas em uma extremidade do conjunto de valores aceitáveis, enquanto o velocímetro tem nas duas.

**Gráficos de histórico** - Esse tipo de gráfico exibe um conjunto de valores da mesma variável distribuídos em um intervalo de tempo. Os valores da variáveis são apresentados no eixo vertical enquanto o instante da coleta aparece no eixo hori-


Figura 4.4: Conjunto de gráficos disponíveis.

zontal. Pertecem à esse grupo o gráfico de barras e de linha.

**Gráficos compostos** - Os gráficos compostos exibem informações combinadas de duas variáveis diferentes. O gráfico de torta, que pertence à esse grupo, exibe a informação de uma variável mais estável que define o valor máximo do gráfico combinado com outra que define a utilização atual. O gráfico de mútiplas linhas combina um conjunto de valores de duas variáveis num mesmo gráfico.

### 4.2.3 Ações

O sistema pode ser configurado para executar qualquer código nos dispositivos monitorados. As ações são configuradas na interface e inseridas no banco como uma nova variável. A partir desse momento essa ação passa a se comportar como uma variável a ser coletada com o diferencial de possuir um intervalo entre coletas igual a zero indicando para o agente que a variável deve ser coletada uma única vez.

Na base de dados uma ação é representada como uma variável pois compartilha com a mesma diversos atributos. E com a flexibilidade de uma variável ser definida por um código que será executado no agente se torna direta a associação de uma ação com uma variável.

Um exemplo de ação implementada no sistema é o bloqueio de um aplicativo em uma estação Linux. O sistema envia um comando para retirar as permissões de execução de um programa na estação como um método de coleta de uma variável. Essa ação é executada uma vez na estação produzindo o efeito desejado.

# 5 *Protocolo*

Como o sistema desenvolvido visa monitorar conjuntos de máquinas espalhados por grandes extensões territoriais, dificilmente seria possível oferecer acesso à todas as localidades utilizando o mesmo tipo de tecnologia. No caso específico da SEEMG foram contratados três provedores com dois tipos de tecnologias diferentes: rádio e ADSL (*Asymmetric Digital Subscriber Line*). A diversidade de conexões não permite supor nada sobre a qualidade das mesmas. Essa característica influenciou na escolha do TCP para camada de transporte, devido às suas características de garantir que os dados serão enviados de forma correta, na sequência apropriada e sem erros. Para a camada de rede foi escolhido o IP pela fato do sistema trafegar os dados pela Internet.

A diversidade de provedores, por sua vez, faz com que o sistema tenha que lidar com políticas de segurança diferentes adotada por cada um deles. Voltando ao exemplo da SEEMG algumas escolas monitoradas não possuem IPs válidos na Internet. Essas escolas se conectam a um servidor, localizado no provedor, que redireciona o tráfego de dados com destino às escolas de volta para elas.

Um outro provedor distribui IPs dinâmicos para seus clientes que podem mudar a qualquer momento, inclusive durante uma conexão de envio de dados coletados.

Essas características inviabilizam a utilização de mecanismos de consulta periódica para a coleta de objetos e impossibilitam que as ações de gerência sejam enviadas aos dipositivos diretamente. Portanto, para o envio das ações, foi escolhida a inversão mestre/escravo [18] descrita anteriormente na subseção 2.1.5 para que seja aproveitada a mesma conexão de recebimento de dados para o envio das ações que devem ser executadas pelos agentes.

## 5.1 Objetivos do Protocolo

As alternativas estudadas para implementar o transporte das informações possuiam limitações que levaram ao desenvolvimento de um protocolo próprio para o sistema.

A característica do SNMP e do Nagios que inviabilizam a utilização de uma dessas duas tecnologias foi a impossibilidade da central de gerência enviar informações para os agentes que não são podem ser diretamente acessados. Tanto no SNMP com o mecanismo de *traps* quanto no nagios com o NSCA é possível receber informações desses agentes mas não enviá-los.

Quanto ao WBEM, por ser um conjunto de várias tecnologias, possui o problema de ser complexa a sua adaptação. Portanto seria difícil e demorado implementar as modificações necessárias nas ferramentas existentes para atender as necessidades do ambiente alvo do sistema.

Como o HTTP, utilizado no WBEM, é um protocolo que estabelece uma conexão que permite o envio e recebimento de dados, seria possível implementar a funcionalidade de enviar ações para os agentes não acessíveis utilizando o recebimento de indicações para o envio das ações. Essa possibilidade não existe nas ferramentas com código aberto disponíveis o OpenWbem [11] e o Pegasus [10].

Outro problema dessa plataforma é a dificuldade de comunicação entre as implementações Windows e Linux. Pois a implementação da microsoft substitui o HTTP pelo DCOP impedindo a comunicação entre os dois ambientes.

Devido a tudo isso foi proposto um novo protocolo que suporta todas as funcionalidades desejadas para o sistema. Isso permitiu que alguns dos objetivos propostos, que necessitariam ser suportados por outras ferramentas como ativação e transferência de arquivos, fossem implementados em um mesmo sistema.

O protocolo foi desenhado com a finalidade de possibilitar a coleta de informações, possibilitar a configuração e atuação nas máquinas monitoradas, visando modificar o comportamento das mesmas. As funções do protocolo são:

- Ativação/Autenticação;
- Coleta;
- Configuração/Atuação;

- Transferência de arquivos.

Durante o projeto foram levantados alguns problemas que deveriam ser tratados pelo protocolo. São eles:

**Alcançabilidade da máquina** - Como as máquinas podem não ser endereçáveis pela Internet, apesar que elas devem estar obrigatoriamente conectadas, o protocolo utiliza uma inversão de cliente e servidor durante a conexão utilizando uma conexão iniciada pelo agente para que a unidade central possa enviar as mensagens de configuração e atuação.

**Latência/Consumo de banda** - O protocolo desenvolvido utiliza pacotes com o mínimo de informação possível para reduzir o tráfego de rede.

**Segurança** - O sistema deve se preocupar com unidades não desejadas enviando informações para central e que todas as unidades conectadas sejam previamente conhecidas para que sejam posicionadas corretamente na topologia geográfica. Para garantir isso foi criada uma etapa de ativação das unidades. Outra preocupação relacionada à segurança é a interceptação de pacotes com informações das escolas na rede, por isso todos os pacotes são transmitidos criptografados.

O protocolo é implementado por duas máquinas de estados: a Máquina de Estados de Envio de Variáveis (MEEV) e a Máquina de Estados de Recepção de Variáveis (MERV). A primeira inicia a conexão e envia as informações da unidade esperando, em seguida, pelas ações que podem ser transmitidas. A segunda é reponsável por esperar e receber as informações enviadas, em seguida deve enviar as ações que tem destinadas a algum agente relacionado com a unidade que está conectando. As duas máquinas de estados estão ilustradas no anexo B.

Nas seções seguintes o protocolo será dividido em funcionalidades que serão descritas com mais detalhes.

## 5.2 Iniciação do Protocolo

Toda conexão é sempre inciada pela MEEV. A MERV aguarda este contato e quando ele acontece envia o primeiro pacote de comunicação, o pacote de cumprimento, informando a identificação da unidade que irá receber os dados.

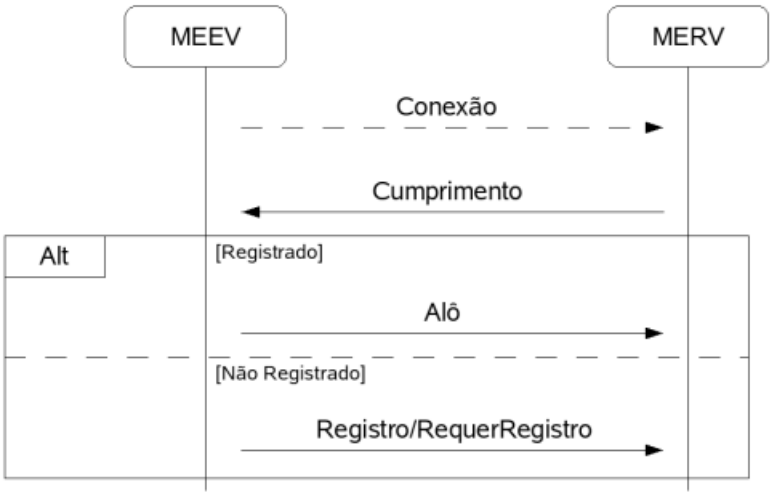


Figura 5.1: Iniciação do Protocolo.

Na sequência a MEEV informa qual o motivo do contato indicando como deseja conduzir a comunicação. Caso o pacote enviado seja de *Alô* significa que em seguida serão enviados dados coletados e requisições que estão na fila de envio da unidade. Esse pacote inicia o processo de coleta detalhado na seção 5.4 e na sequência da trasmissão pode ocorrer o processo de transferência de ações descrito na seção 5.5.

O pacote *Alô* contem a chave de ativação da unidade que permite a unidade que irá receber o pacote decidir se deve ou não continuar a conexão.

Em vez de um pacote *Alô* o MEEV pode enviar um pacote *Registro* ou *RequerRegistro* indicando que a unidade ainda não foi ativada e deseja receber uma chave de ativação. Assim se inicia o processo de ativação da unidade descrito na seção 5.3.

## 5.3 Ativação/Autenticação

Todas as unidades do sistema devem ser registradas e ativadas antes de iniciar o envio de dados. Isso é necessário para garantir que todas as unidades são conhecidas pela central e tem permissão para participar do sistema. O processo de ativação começa com a inserção das informações de registro nas unidades como dados do responsável, endereço da unidade, etc.

Após essa etapa a MEEV dessa unidade envia uma unidade superior na hierarquia um pacote de registro contendo essas informações. A unidade que recebe esse pacote, caso não seja uma central, enfileira o pacote como um variável coletada e novamente repassa esse pacote até encontrar uma central.

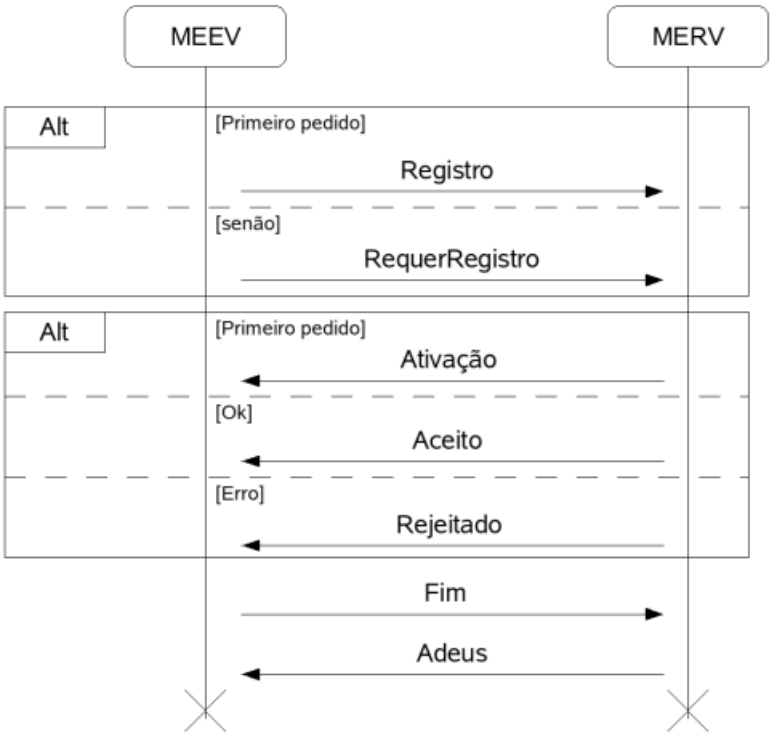


Figura 5.2: Processo de ativação das unidades.

A unidade que recebe esse pacote faz a checagem de consistência e integridade e sua MERV responde com um pacote de *Aceito* ou *Rejeitado* se a ativação para a chave de registro contida no pacote ainda não tiver sido gerada.

Ao receber um pacote diferente de *Ativação* a unidade enviará posteriormente, em outras conexões, pacotes de *RequererAtivação* no mesmo processo. Esse pacote difere do pacote de *Registro* por conter apenas a chave de registro diminuindo a quantidade de informação trafegada.

A central, ao receber um pacote de registro, insere as informações trazidas por ele em uma tabela de unidades candidatas a receber uma chave de ativação. Em um determinado momento o administrador na central cadastra a unidade e gera uma chave de ativação para a mesma. Neste instante a unidade é retirada da tabela de espera pela ativação. Após esse processo a central, quando receber um pacote *RequererAtivação* da unidade que acabou de ser cadastrada, irá responder com um pacote de ativação para a MEEV da unidade que enviou o pedido de ativação.

O pacote de ativação é enviado como uma ação para as unidades que transportam o pedido da ativação e somente na unidade que será ativada é que este pacote é visto como um pacote de ativação.

## 5.4 Coleta

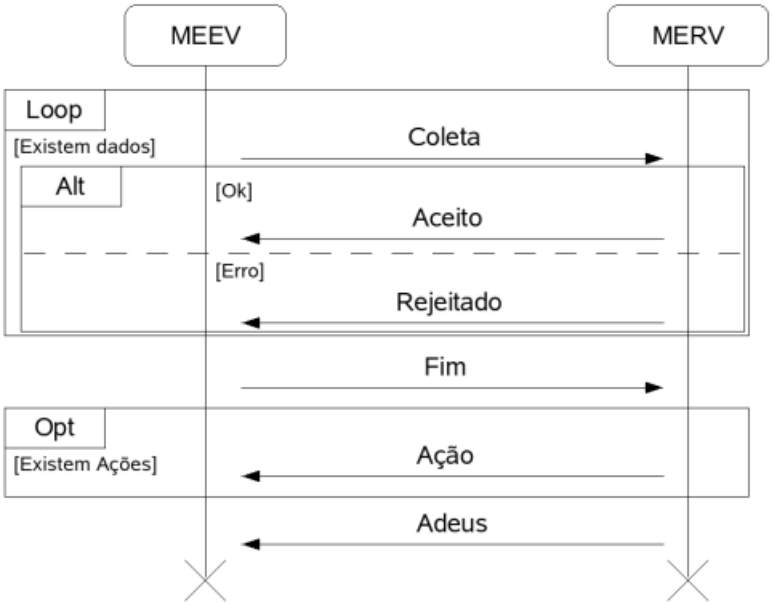


Figura 5.3: Processo de ativação das unidades.

A coleta ocorre somente após a configuração dos agentes. Com suas agendas preenchidas os agentes começam a coletar as variáveis solicitadas e armazenar num banco de dados. A medida que são coletadas o agente envia essas variáveis armazenadas para a unidade central.

Este processo se dá logo após a iniciação do protocolo. A MEEV envia um pacote com cada valor coletado, esperando da MERV a confirmação de recebimento com a vinda de um pacote *Aceito* ou *Rejeitado*. Atualmente uma unidade somente rejeita uma coleta caso seus valores não estejam consistentes. Uma variável coletada pode ser um arquivo disponível na máquina onde o agente está instalado o processo de transferência desse arquivo é descrito na seção 5.6.

Quando não há mais informações a serem enviadas a MEEV transmite um pacote de *Fim* indicando o término da coleta. Neste ponto a MERV pode responder com o envio de ações, iniciando o processo descrito na seção 5.5, ou o envio do pacote *Adeus* terminando a conexão.

Ao final do processo de ativação existe a possibilidade de nenhuma configuração de variável ser enviada junto com a ativação. Caso isso ocorra o agente ficará ativado e nunca coletará uma variável. Para que esse agente não fique incomunicável ele envia periodicamente pacotes de *Fim* para que seja possível enviar ações para esta unidade.

## 5.5 Configuração/Atuação

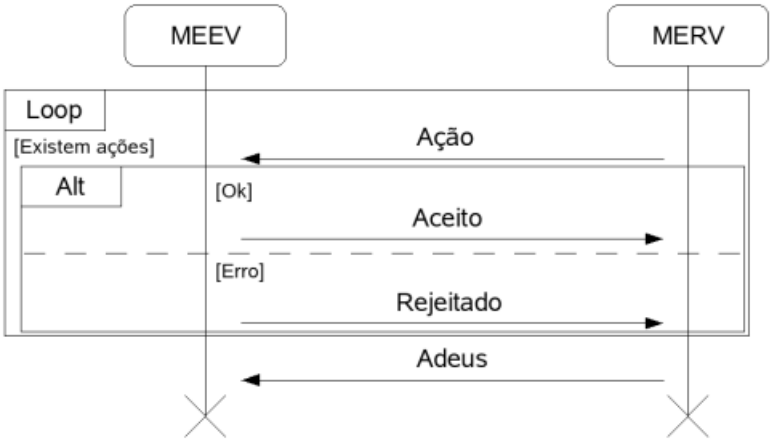


Figura 5.4: Processo de ativação das unidades.

Após o fim do processo de coleta a MERV assume o controle da comunicação e começa o envio de pacotes com ações para o MEEV. O processo de configuração e atuação são controlados pela unidade superior e o agente atua como cliente no processo, mas como é o agente que inicia a conexão, ele envia um pacote de fim de transmissão para indicar a inversão da conexão.

Esses pacotes podem ser *Configuração*, contendo informações para modificar ou adicionar uma nova variável num agente. Eles também pode ser pacotes de ativação que caminham pela hierarquia até chegarem à sua unidade de destino. No final do processo de envio de ações o protocolo é terminado com o envio do pacote *Adeus* pelo MERV.

Esse mecanismo é necessário para garantir que todas as unidades do sistema terão capacidade de receber as ações enviadas pela central, independente de seram diretamente acessíveis por ela ou não.

## 5.6 Transferência de Arquivos

A central tem o poder de requisitar um arquivo presente numa máquina monitorada. Para isso é necessário apenas que se saiba previamente o caminho desse arquivo e que seja criada uma variável para coletar esse arquivo.

O processo de transferência de arquivos ocorre durante o ciclo de coleta de variáveis. O início desse processo é marcado pelo envio do pacote *EnviaArquivo* pela MEEV.

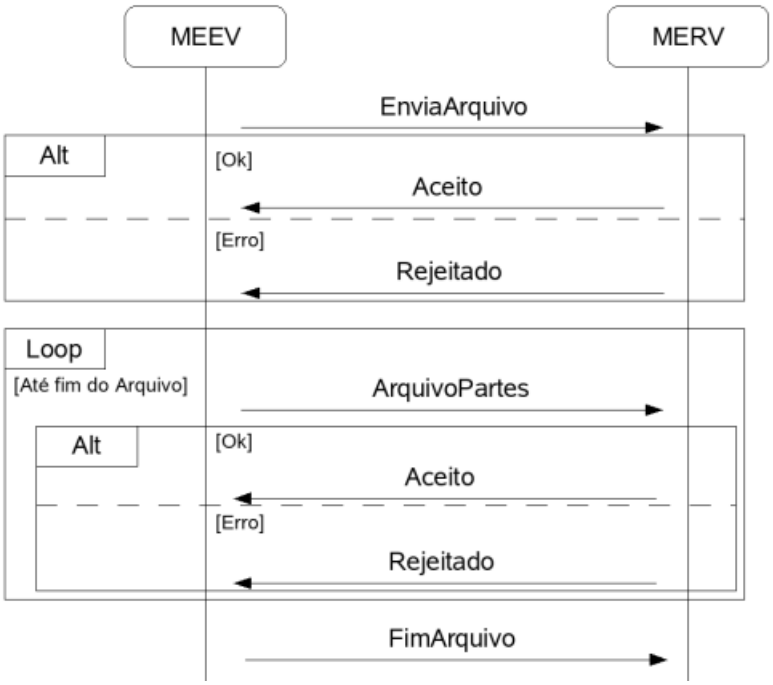


Figura 5.5: Processo de ativação das unidades.

Este pacote contém o nome do arquivo, suas permissões e seu tamanho. Se a MERV da outra unidade responder esse pacote com um *Aceito* o transferência começa. Os pedaços do arquivo são enviados nos pacotes *ArquivoPartes* que possuem também a identificação da unidade e uma checagem da integridade do bloco. Esses pacotes são respondidos com *Aceito* ou *Rejeitado* pela MERV. O processo é terminado com o envio do pacote *FimArquivo*.

# 6 *Estrutura de Armazenamento de Dados*

Neste capítulo será apresentado como cada unidade armazena suas informações de configuração e gerência e o histórico dos objetos coletados.

As unidades central e intermediarias possuem uma estrutura de banco de dados idêntica para armazenar as informações de gerência e os dados coletados que são colocados num banco de dados MySQL. As informações de configuração ficam a parte em um arquivo contendo o endereço para onde deve ser enviado os dados, quais portos devem ser utilizados nas conexões e as configurações do banco de dados.

Essa estrutura semelhante nessas unidades é necessária para que ambas suportem as mesmas funcionalidades facilitando a criação de uma hierarquia de gerência. Isso também auxilia o reaproveitamento de código entre a central e a unidade intermediária.

Os agentes, por sua vez, se utilizam de ferramentas mais simples para armazenar todas as informações necessárias. Todas elas estão gravadas num banco de dados SQLite, que consome menos recursos do dispositivo gerênciado, e simplifica a implementação da camada de persistência do programa.

Nas próximas seções serão explicadas as estruturas dos bancos de dados do agente, o banco da central e das unidades intermediárias e o formato do arquivo de configuração dessas unidades.

## 6.1 Banco de Dados da Unidade Central e Intermediária

Este é o principal banco de dados do sistema. Nele são criadas e armazenadas todas as configurações dos objetos que serão monitorados no ambiente e a partir dele são distribuídas para as outras unidades do sistema. Nesse banco estão disponíveis as

informações cadastrais de todos as unidades que compõem o sistema. Ele também é o destino de todos os objetos coletados no ambiente.

Outra função importante deste banco é servir como mecanismo de comunicação entre a interface homem-máquina e a unidade central. Portanto as informações devem estar estruturadas para possibilitar que a interface seja simples e compreensível para o usuário e ofereça um conjunto de informações para a unidade central que permita a tomada de decisões complexas pela mesma.

Como esse banco é manipulado por componentes fundamentais do sistema e armazena informações importantes, é necessário que ele possua também a capacidade de registrar falhas na sua operação para posterior análise de seu funcionamento.

Quando é instalado em uma unidade intermediária são acrescentadas duas tabelas que controlam as filas de retransmissão de dados que funcionam nessa unidade. Este mecanismo foi descrito na subseção 4.1.2.

Uma visão completa do banco com todas as suas tabelas e relacionamentos é mostrada na anexo A. Nas subseções seguintes as tabelas serão agrupadas de acordo com suas funções e descritas com mais detalhes.

### 6.1.1 Unidades

As tabelas presentes na figura 6.1 estão relacionadas ao cadastro e registro das unidades. As tabelas *NodeCandidate e ActivateAgenda* são utilizadas apenas no processo de registro e ativação. A interface utiliza esses dados para colocar essa unidade na tabela *NodeDef* que permitirá à nova unidade começar a enviar variáveis. No mesmo momento a interface retira a unidade da tabela *NodeCandidate* e acrescenta uma entrada na tabela *ActivateAgenda* que sinalizará que o protocolo deve enviar para a unidade a sua chave de ativação que foi gerada nesse processo.

Sempre que a unidade central recebe um pacote de **Registro** ou **RequererRegistro** ela verifica se existe uma entrada na tabela *ActivateAgenda* para enviar uma resposta a esse pacote. Se não houver nada nessa tabela e o pacote recebido for o de registro suas informações são armazenadas na tabela *NodeCandidate*. As informações inseridas no banco dizem respeito basicamente à localização da unidade, seu responsável e sua chave de registro.

A interface utiliza esses dados em combinação com a chave de ativação e outras

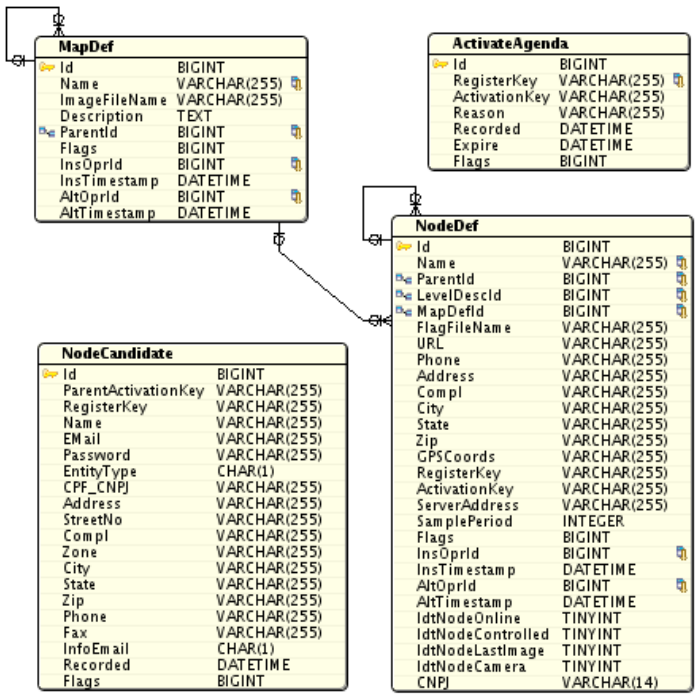


Figura 6.1: Tabelas relacionadas ao cadastro de unidades.

informações cadastrais inseridas pelo administrador nessa unidade, como os dados armazenados na tabela *MapDef* e para preencher a tabela *NodeDef*. A unidade central só irá receber dados de unidades que possuam uma entrada nessa tabela.

Quando essa entrada é criada, a unidade é removida da tabela *NodeCandidate* e uma nova é criada na *ActivateAgenda* para que a central informe a unidade sua nova chave de ativação.

### 6.1.2 Variáveis

Na figura 6.2 foram agrupadas as tabelas que definem os objetos que serão monitorados pelos agentes. A tabela *VariableDefs* guarda o esquema de como as variáveis devem ser coletadas e exibidas. A tabela *Method* contém modelos de mecanismos de coleta diferentes. Esses mecanismos podem ser buscar uma variável WBEM, SNMP, executar um método ou buscar um arquivo. Esses modelos são expandidos no momento que a central cria um pacote de configuração. Isso permite que a mesma variável possa ser coletada e exibida de formas diferentes para cada unidade.

Quando o administrador deseja observar um novo objeto em uma máquina monitorada ele se utiliza de um dos modelos disponíveis de variáveis e de métodos para criar uma nova entrada na tabela *NodeVariableDef* que referência as tabelas *VariableDef*,

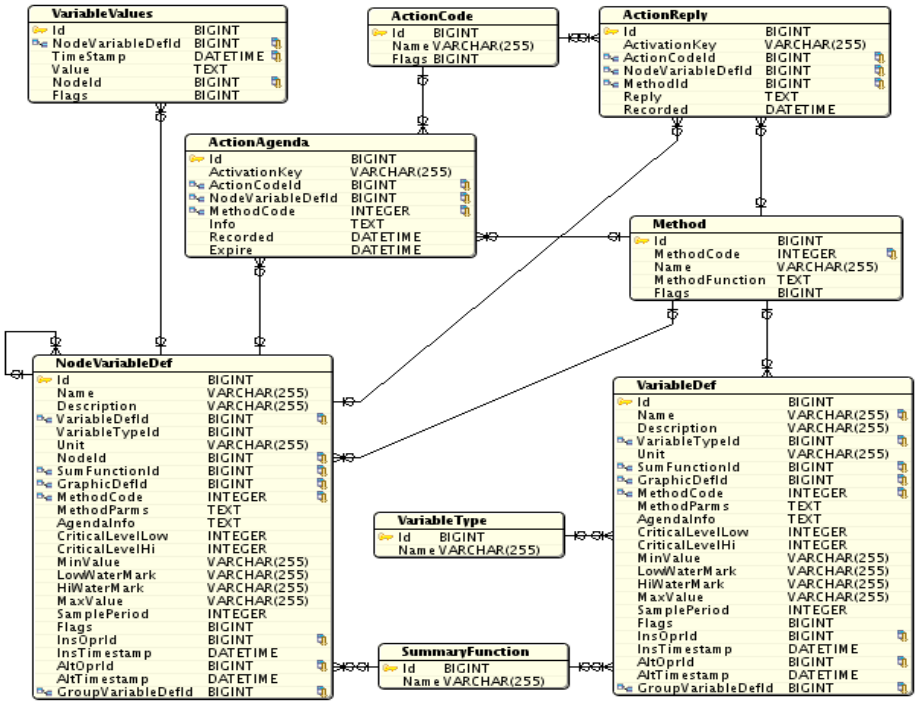


Figura 6.2: Tabelas relacionadas à coleta de variáveis.

*Method* e *NodeDef* para definir de maneira única um objeto.

Nesse processo é criada uma entrada na tabela *ActionAgenda* e a central se encarrega de transmitir essa nova variável ao agente referenciado pelo objeto.

### 6.1.3 Auditoria

Cada unidade tem associada a si um responsável. Essa pessoa deve ser informada sobre o comportamento da unidade e ser responsabilizada caso alguma falha ocorra na unidade. As tabelas da figura 6.3, permitem fazer uma auditoria no sistema, guardando informações dos responsáveis, dos operadores e como eles interagem com o sistema.

A tabela *Responsible* contém os dados pessoais de cada responsável por alguma unidade. A tabela *NodeResponsible* associa essa pessoa a uma unidade e registra as informações do operador que realizou as associações.

A tabela *Operator* contém informações cadastrais dos pessoas que têm permissão para controlar o sistema se utilizando da interface de administração. Na tabela *Config* estão algumas preferências dos operadores para interagir com o sistema. A tabela *AuditLog* guarda todas as ações executadas por um operador na central.

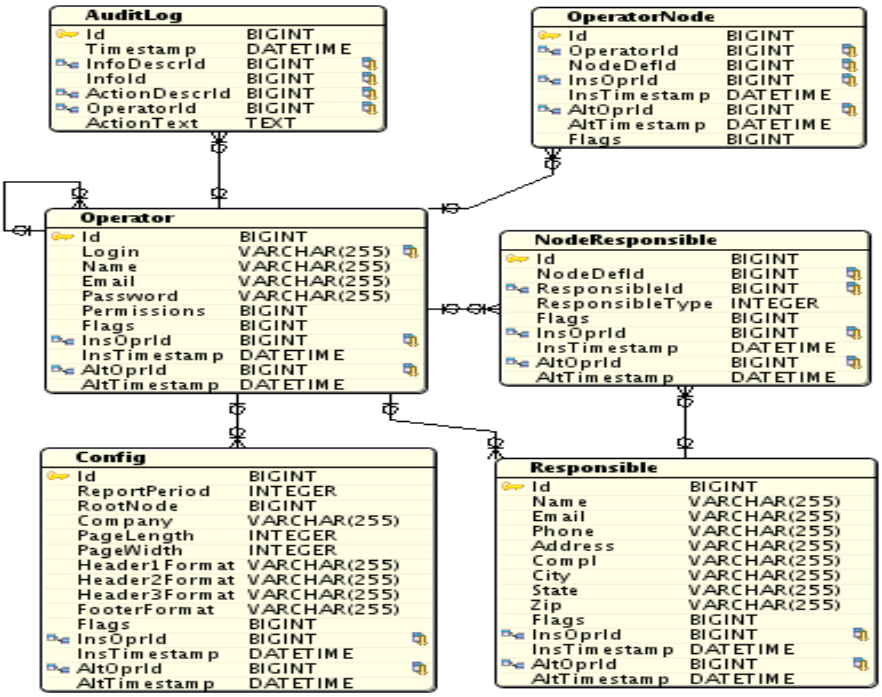


Figura 6.3: Tabelas relacionadas com a auditoria do sistema.

### 6.1.4 Configuração

Algumas informações das unidades central e intermediária precisam estar disponíveis antes do início da operação de ambas. Essas informações são colocadas num arquivo de configuração que é preenchido pela unidade quando esta é chamada no modo de configuração.

Esse arquivo armazena as informações no formato textual e é composto de seções e campos. As seções são representadas por linhas contendo um texto entre colchetes. Já os campos são definidos por um texto, que define o nome do campo, seguido do sinal de igual e o valor deste campo entre aspas. Todas as linhas que definem campos são terminadas por ponto-e-vírgula.

Atualmente este arquivo possui as seções *Server*, *Client* e *Database*, onde são definidas as configurações da MERV, MEEV (somente na unidade intermediária) e do acesso ao banco de dados respectivamente.

## 6.2 Banco de Dados da Unidade de Coleta

Visando a simplicidade e portabilidade do agente, todas as suas informações são mantidas no mesmo banco de dados. As informações de configuração dos agentes ficam armazenadas nas tabelas *Config* e *Server*. A tabela *Config* possui um formato semelhante à tabela *NodeCandidate* da central. Isso porque quase todos os dados presentes nessa tabela serão transportados para central no processo de ativação.

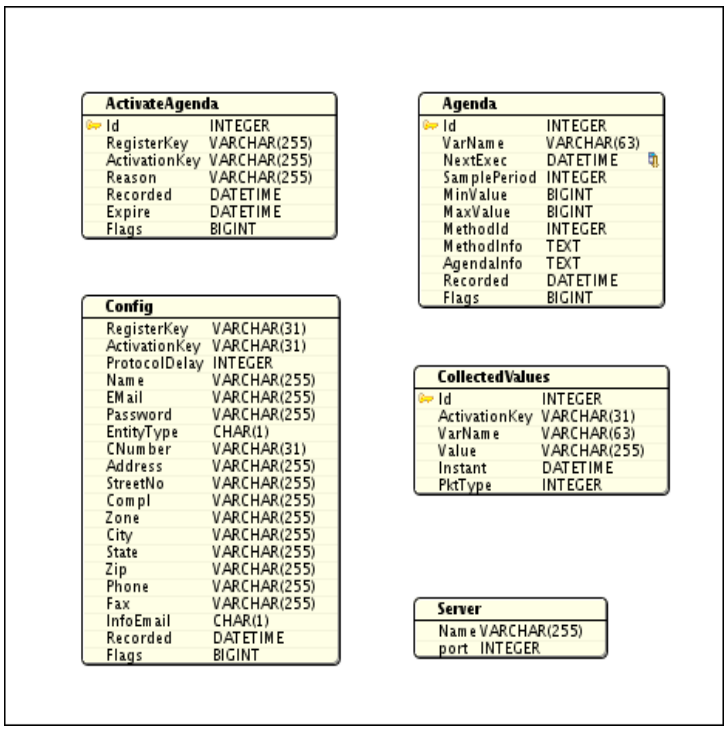


Figura 6.4: Banco de Dados dos agentes

Nesta tabela os campos *ProtocolDelay* e *Flags* são utilizados para o controle do funcionamento do agente. O *Flags* atualmente controla o processo de envio dos pacotes do processo de registro. *ProtocolDelay* controla o período de envio das coletas.

A tabela *Server* é usada para configurar o endereço e o porto da máquina a qual o agente deve se conectar para se ativar e enviar os dados coletados.

As informações semânticas sobre os objetos monitorados são guardadas na tabela *Agenda*. Nesta tabela estão qual o comando e como ele deve ser executado para coletar uma variável, os limites permitidos para a variável e o intervalo entre as coletas da mesma.

Após a coleta os valores obtidos ficam armazenados na tabela *CollectValue* até o momento em que são enviadas para alguma unidade superior.

# 7 *Estudo de Caso da Secretaria de Educação de Minas Gerais*

Visando modernizar o ensino no estado e oferecer recursos para as escolas prepararem os alunos para a utilização de computadores e da Internet, ferramentas essênciais para o mercado de trabalho comtemporâneo, o governo do estado de Minas Gerais criou o programa *Escola em Rede*. Quando estiver completo este programa irá informatizar todas as quatro mil escolas do estado.

A implantação de um programa dessa magnitude é cara e exige uma série de cuidados da SEEMG para garantir que os recursos adquiridos estão sendo utilizados corretamente. O sistema de gerência auxilia nessa tarefa provendo um mecanismo para monitoração das escolas. Dessa forma é possível determinar se as máquinas estão sendo utilizadas e como isso está acontecendo.

Atualmente uma parte das escolas já está com as máquinas funcionando e sendo monitoradas. Com o início da utilização delas aparece paralelamente a necessidade da prestação de suporte a essas escolas. Este suporte é oferecido por uma unidade da secretaria de educação, instalada na capital do estado, a todas as escolas de Minas Gerais. Neste serviço são utilizadas as informações monitoradas, as funções de gerência do sistema e o acesso remoto às máquinas quando isso é possível.

## 7.1 Descrição do Ambiente

Estão sendo enviadas para as escolas em média cinco máquinas, sendo uma delas um servidor, algumas máquinas para utilização na administração da escola e as restantes são empregadas na criação ou ampliação de um laboratório de informática para os alunos.

Na máquina servidora é instalado o sistema operacional Metasys Corporate, uma

distribuição linux desenvolvida para facilitar a gerência de uma rede. Essa máquina é responsável por autenticar os usuários da escola, rotear conexões com a Internet além de prover a iniciação remota para algumas máquinas do laboratório de informática que não possuem dísco rígido. Nesse servidor é instalado a unidade intermediária do sistema de gerência. Nas máquinas instaladas na administração da escola e nas demais que tiverem dísco rígido é instalado o sistema operacional Metasys Desktop.

Em algumas escolas essas máquinas serão integradas a uma estrutura computacional existente possivelmente dividindo espaço com máquinas com sistema operacional diferente. Em todas as máquinas da escola, incluindo o servidor e as máquinas que não possuem linux instalado, é instalada uma unidade de coleta.

Todas as escolas que recebem os computadores estão sendo conectadas à Internet por conexões banda larga de 128kbps. As conexões das escolas não são dedicadas ao monitoramento, portanto nos momentos de maior utilização da Internet nas escolas a latência das mensagens de coleta aumenta consideravelmente. A tecnologia da conexão e o provedor utilizado variam com a oferta dos serviços nas diferentes regiões do estado.

Um exemplo de provedor de acesso para as escolas é o Velox [15]. As escolas atendidas por este provedor possuem conexão ADSL e são endereçáveis, apesar do endereço oferecido às escolas ser trocado periodicamente. Isso permite que os administradores na central, utilizando a informação do endereço externo coletado no monitoramento, acessem o servidor da escola diretamente para realizar algum procedimento de manutenção.

Outro exemplo é o provedor RuralWeb [12] que fornece conexão por rádio às escolas, porém não fornece um endereço válido na Internet. Nas escolas conectadas por esse provedor a administração remota só é possível por meio do sistema de gerência. Existem outras empresas que provêem acesso para as escolas mas que possuem restrições de acesso semelhantes a esses dois exemplos.

### 7.1.1 Central de Monitoramento

A central de monitoramento é localizada em Belo Horizonte. Nela está situado um sistema redundante com duas máquinas sincronizadas que abrigam a central de monitoramento e interface de gerência. Neste mesmo local foi colocado uma central de suporte onde os operadores prestam atendimento às diversas escolas do estado.

Essa estrutura está protegida atrás de um *firewall* e conectada à Internet por duas conexões uma de 600 kbps e outra de 1 Mbps utilizadas simultaneamente com um esquema de balanceamento de carga para servir toda a estrutura montada neste local.

## 7.2 Analise de Desempenho

O sistema de gerência da SEEMG está operacional desde o dia 3 de outubro de 2007. Até o dia 14 do mesmo mês haviam 604 unidades cadastradas sendo 136 unidades intermediárias e 468 agentes. Nesse período foram armazenados mais de 31 mil valores de variáveis no banco de dados da central.

Foram escolhidas algumas medidas de desempenho para avaliar o comportamento do sistema num ambiente real. Essa análise irá servir para validar as escolhas feitas no projeto do sistema e prever se será necessário ampliar a infraestrutura para suportar a demanda quando o ambiente estiver completo com todas as unidades previstas.

Na subseção 7.2.1 será apresentado o impacto das unidades do sistema nos componentes do ambiente monitorado. Na subseção 7.2.2 será feita uma caracterização e uma análise do tráfego gerado pelo sistema.

### 7.2.1 Comportamento do Sistema de Monitoramento no Ambiente Monitorado

Uma característica importante de um sistema de gerência é a sua eficiência. Um sistema desse tipo não deve causar uma grande interferência no meio que está sendo monitorado sob pena de comprometer a análise dos dados de desempenho do sistema.

Por este motivo os agentes, que são distribuídos por todas as máquinas monitoradas, foram desenvolvidos visando consumir o mínimo de recursos possível. Todas as funcionalidades de coleta específica de uma variável foram delegadas às rotinas pré-configuradas pela própria variável. Isso, além de contribuir muito para a flexibilidade do sistema, permite que o programa "*MonitorAgent*", que é a parte do agente responsável pela aquisição dos dados, seja somente um escalonador de tarefas. Este programa mais o "*AgentProtocol*", responsável por tratar das conexões com as outras unidades, presentes em uma máquina monitorada representam o impacto do monitoramento na mesma.

O gasto do processamento dos agentes é mínimo. O processo de coleta tem o gasto

de processamento limitado pelas rotinas de coleta que podem ser modificadas e otimizadas à medida das restrições dos componentes monitorados. As rotinas utilizadas hoje constituem em chamadas a um servidor WBEM, a um SNMP e além da chamada direta à alguns comandos. Utilizando 20 variáveis configuradas o processo gastou 21 segundos para coletar esse valores. Esses dados foram medidos em um Sempron 3300+ com 512MB de memória utilizando a distribuição Linux openSUSE 10.2.

Como visto o processo de coleta consome uma quantidade de processamento pequena mas exige cuidados em relação aos intervalos entre as coletas. Atualmente o menor intervalo entre as coletas é de 30 minutos, o que é suficiente para o processo de coleta causar muito impacto no desempenho da máquina monitorada.

Quanto à utilização de memória, os agentes precisaram ser compilados estaticamente, para facilitar o processo de distribuição do sistema. Desta forma o mesmo executável pode ser instalado em sistemas Linux diferentes. Essa decisão foi tomada também pensando na estabilidade do sistema pois o executável não dependeria de nenhuma outra biblioteca.

Na mesma máquina utilizada na medida de processamento os dois programas que compõem o agente de monitoramento "*MonitorAgent*" e "*AgentProtocol*" consomem respectivamente 1031 KB e 1086 KB de memória ao serem inicializados. Estes dados foram obtidos observando o campo do total de memória gasto por meio do comando **size** do Linux. Este consumo de memória permite que o agente seja executado sem causar um grande impacto nas máquinas das escolas que têm em média 256 MB de memória principal.

As unidades intermediárias por terem um funcionamento bastante parecido com a central não serão analisadas com mais detalhes nessa seção. Estas unidades são executadas geralmente nos servidores que estão nas escolas, que possuem mais recursos computacionais que as estações, e recebem informações de um número pequeno de máquinas. Portanto essas unidades não é um gargalo no funcionamento do sistema e geram um impacto muito pequeno nessas máquinas.

A unidade central é encarregada principalmente de receber os dados enviados e determinar o local correto para inserí-los no banco de dados. A interface de gerência, outro componente do sistema muito interligado com a central, também tem seu desempenho limitado pelas inserções e consultas ao banco de dados. A análise do desempenho das ferramentas que estão na central será concentrada no desempenho do banco de dados.

A central de monitoramento da SEEMG está instalada numa máquina Pentium D de 3 GHz com 2 GB de memória principal rodando a distribuição Linux openSUSE 10.2. Está instalado também nessa máquina o banco de dados MySQL 5.0.45 que armazena as informações do sistema. Esta máquina é dedicada exclusivamente ao sistema de monitoramento.

No início da instalação do sistema nas escolas o consumo de processamento na central de monitoramento estava muito alto, mesmo com um número pequeno de máquinas enviando dados. Para 62 unidades enviando informações para a central a utilização da capacidade de processamento da máquina se mantinha em 95%.

Foram detectadas duas causas para este comportamento. A primeira foi o tamanho da *cache* utilizada pelo MySQL. Por padrão nesse banco de dados este tamanho é configurado para utilizar 2 MB da memória principal. Mudando esse parâmetro e analisando as informações geradas pela ferramenta *mysqlreport* foi observado que para esta quantidade de unidades uma cache de 7 MB garantiria uma taxa de acertos na *cache* de 99%. Modificando o tamanho da cache para 16 MB, portanto acima dos 7 MB, a utilização do processador da máquina caiu para uma média de 50%.

A segunda causa da baixo desempenho da ferramenta no início da operação era o agrupamento dos dados coletados nos níveis superiores da hierarquia. Devido à um erro de configuração em uma das variáveis ela estava sendo coletado em intervalos de 3 segundos. Isso gerava uma grande quantidade de consultas e inserções no banco de dados quando essa variável era agrupada nos seis níveis na hierarquia de exibição da interface. Desabilitando essa funcionalidade dessa variável, a utilização da capacidade de processamento da central se estabilizou numa média de 4%. Essa experiência foi importante para detectar que essa operação seria um problema se aplicada a variáveis muito dinâmicas quando o sistema estiver operando com um quantidade maior de unidades.

Com relação a utilização de disco na central para armazenamento das informações foi implementado um mecanismo para evitar que informações repetidas fossem gravadas no banco de dados. Este mecanismo é descrito no código 4.1.1. Como não existe ainda um mecanismo de ajuste dinâmico dos intervalos de coleta de uma variável elas estão sendo superamostradas pelo sistema visando a detecção dos eventos mais rapidamente.

Para analisar o comportamento desse mecanismo foram observados os valores de duas variáveis, uma dinâmica e outra estática, em uma máquina da Escola Estadual

Joseph Stalin Romano no dia 12/11/2007 às 21:30h. Essa escola começou a enviar informações no dia 03/11/2007. A variável estática que envia o nome do processador foi recebida pela central 34 vezes nesse período e foram gravados apenas 2 registros, que apesar de gerarem atualizações no banco de dados, possibilitaram uma economia de 94% de espaço.

A variável dinâmica que retorna a quantidade de memória livre disponível na máquina monitorada foi recebida pela central 836 vezes, porém gerou apenas 536 registros no banco de dados. Neste caso a economia foi de 36%.

### 7.2.2 Tráfego Gerado pelo Sistema de Monitoramento

Um outro ponto crítico do sistema é o tráfego gerado pela comunicação entre as unidades. No dia 21/11/07 durante o período entre 15 horas 7 minutos e 38 segundos e 15 horas 15 minutos e 2 segundos foram capturados 10000 pacotes do protocolo de comunicação do sistema de monitoramento na central. Este horário foi escolhido por possuir um grande número de escolas enviando informações.

Os pacotes tinham tamanho entre 40 e 670 bytes. Esses pacotes, gerados por 106 unidades distintas em 492 conexões diferentes, trafegaram 701368 bytes de dados pela rede. Isso significa que a taxa de transferência média gerada pelo sistema de monitoramento foi de 1579 bps. Na figura7.1 podemos observar picos de 6000 bps, sendo ainda um tráfego muito pequeno para a quantidade de máquinas conectadas.

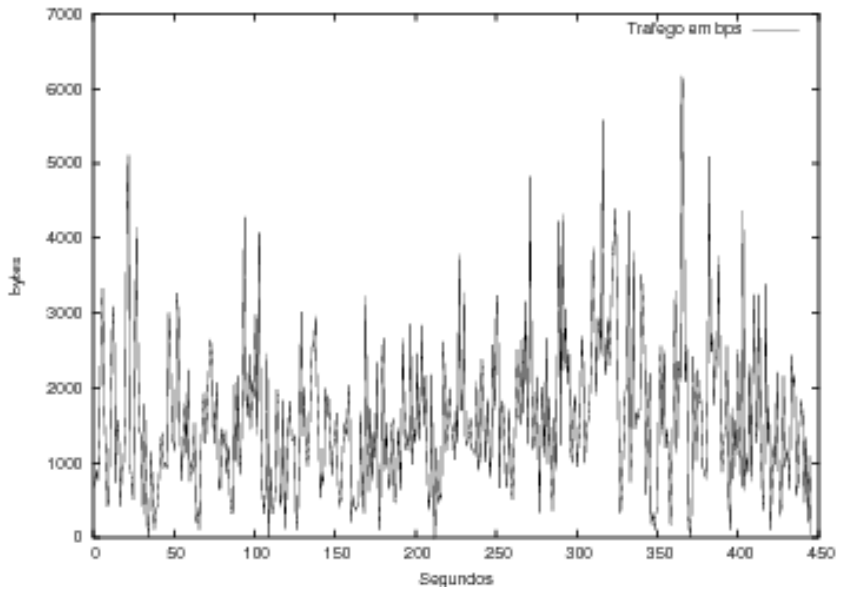


Figura 7.1: Tráfego gerado pelo monitoramento num período de 444 segundos.

# 8 Conclusões e Trabalhos Futuros

Gerenciar de maneira eficaz uma rede de computadores com centenas, ou às vezes milhares, de máquinas não é uma tarefa simples. Por este motivo surgiram várias ferramentas destinadas ao monitoramento e administração remota dessas redes, sendo algumas delas citadas nesse trabalho, que possuem o objetivo de resolver o problema em um cenário específico e quando utilizadas em outros ambientes possuem algumas limitações.

Os parques computacionais criados pelo setor público brasileiro possuem características próprias que diferenciam a tarefa de gerência remota nesse ambiente. O sistema proposto nesse trabalho conseguiu realizar essa tarefa em um ambiente grande e complexo como o do programa *Escola em Rede* da secretaria de educação de Minas Gerais.

Isso foi conseguido adotando várias medidas que em conjunto possibilitou alcançar esse objetivo. Uma dessas medidas foi concentrar o pessoal qualificado em uma localização, permitindo que desse ponto fossem observados o comportamento de todas as unidades da rede.

Outra medida foi utilizar o sistema de comunicação de eventos pelos agentes, que simplificou a tarefa de acrescentar novas unidades e facilitou a aquisição das informações. Isso em conjunto com a inversão mestre/escravo que possibilitou o envio de ações para unidades não endereçáveis na internet, que era uma característica de ambiente que dificultava a gerência.

Com essas duas medidas mencionadas acima o problema de falta de mão-de-obra especializada em alguns pontos do estado foi amenizado.

Entre os objetivos do sistema estavam a segurança nas comunicações e a capacidade das unidades identificarem quais delas pertecem ao sistema. O primeiro problema foi tratado criptografando as conexões e o segundo com as chaves de ativação que garante que uma unidade é conhecida pela central.

O sistema também deveria comunicar ao responsável por uma máquina as falhas que ocorressem nela. Isso foi feito analisando os dados que chegam na central e gerando um aviso quando algo anormal for detectado.

## 8.1 Trabalhos Futuros

Apesar do sistema ter cumprido seu objetivo existem alguns pontos que podem ser aperfeiçoados e funcionalidades que podem ser incorporadas para melhorar a eficiência do mesmo. Algumas dessas melhorias serão listadas aqui.

**Aperfeiçoamento do mecanismo de execução de ações** - Atualmente qualquer ação enviada pela central é executada em um agente. Seria mais seguro que essas ações fossem validadas de alguma forma com o objetivo de identificar se elas podem causar algum dano ao sistema.

**Agentes autônomos** - Toda a identificação das falhas no ambiente é feita na central. Os agentes possuem várias informações que os habilitam a detectar alguns problemas precocemente, gerando um alarme, e em alguns casos pederiam agir para recuperar a máquina monitorada de um estado de erro. Outra possibilidade com o aumento da independência dos agentes é a identificação de gargalos sazonais, utilizando a metodologia de conhecimento prévio [28] por exemplo, e postergar eventos prevendo congestionamentos ou contenção.

**Comunicação entre unidades intermediárias e central** - Quase toda troca de informações no sistema ocorrem entre a central e os agentes. Como as unidades intermediárias possuem toda a informação sobre as máquinas que mandaram informações por elas a central poderia delegar algumas ações para serem tomadas por essas unidades. Para isso seria necessário implementar um mecanismo de envio de políticas de gerência da central para a unidade intermediária.

**Avaliação estocásticas dos objetos definidos** - Este tipo de avaliação exige a análise de uma grande quantidade de dados para oferecer resultados úteis. Portanto só será possível com o sistema em funcionamento por um tempo. Uma análise cuidadosa dos dados obtidos no sistema pode auxiliar o aperfeiçoamento de alguns pontos do sistema como identificação de períodos adequados de amostragem, otimização de agentes de coleta específicos, otimização do armazenamento e na sumarização e outros.

**Aperfeiçoamento da iterface de administração** - Analisando a utilização da interface pelos operadores e identificando as operações mais comuns, será possível otimizar a exibição das variáveis destacando as mais relevantes para aquele operador. Identificando as tarefas de gerência mais comuns será possível propor novas formas de mapeamento e exibição de características e eventos.

# *Anexo A*

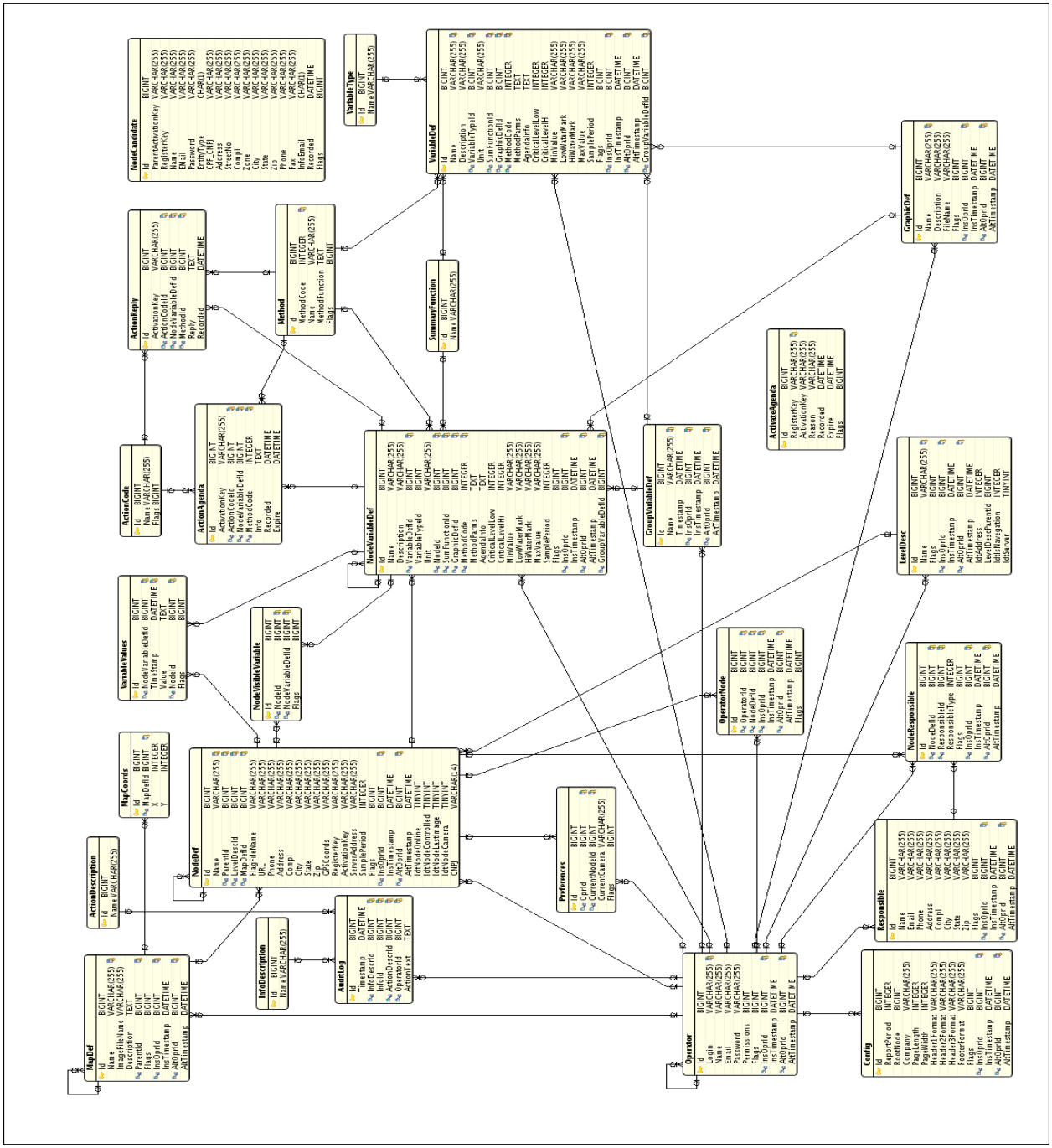

Figura 1: Estrutura do Banco de Dados da Unidade Central.

# *Anexo B*

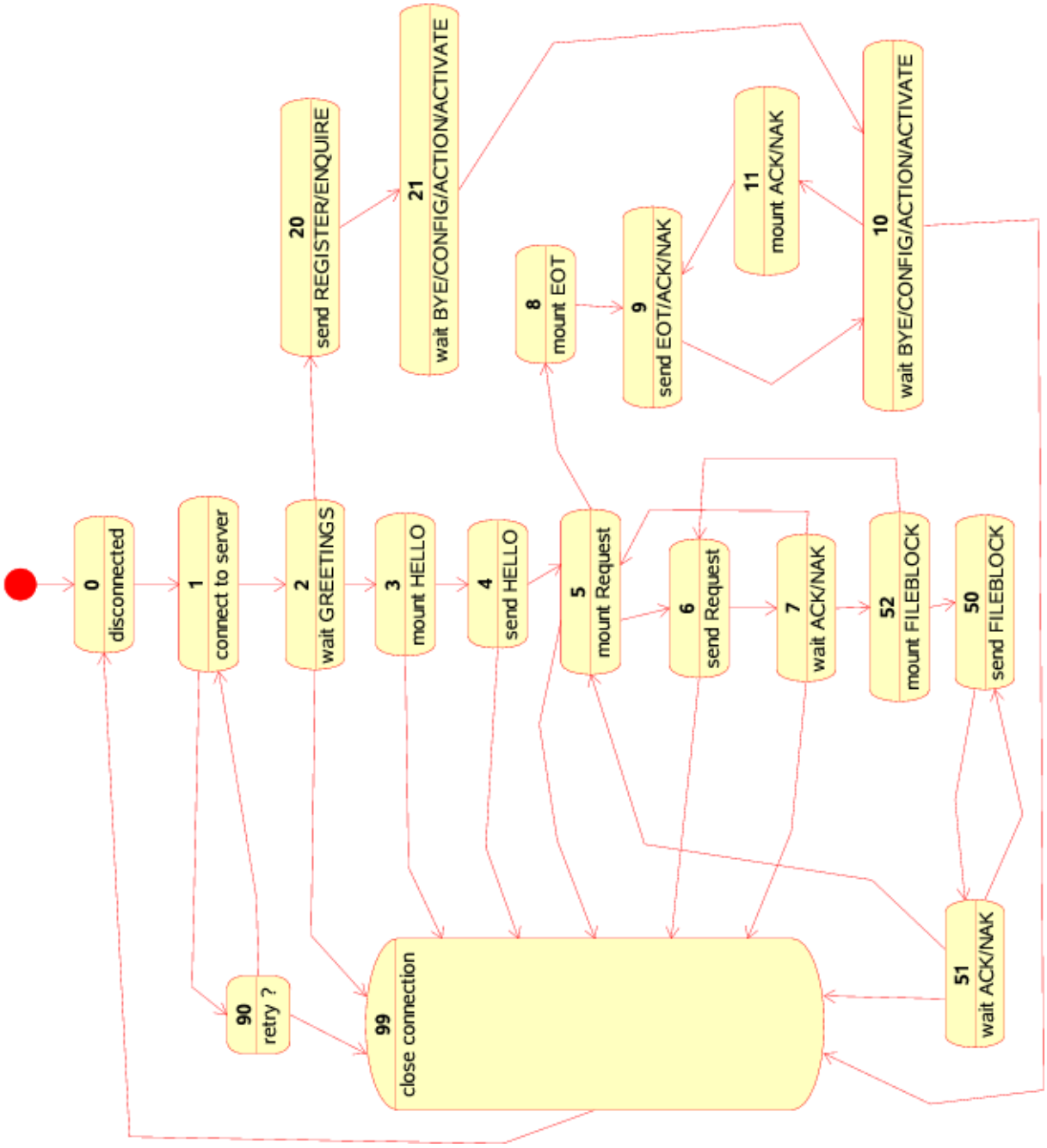


Figura 1: Máquina de Estados de Envio de Variáveis.

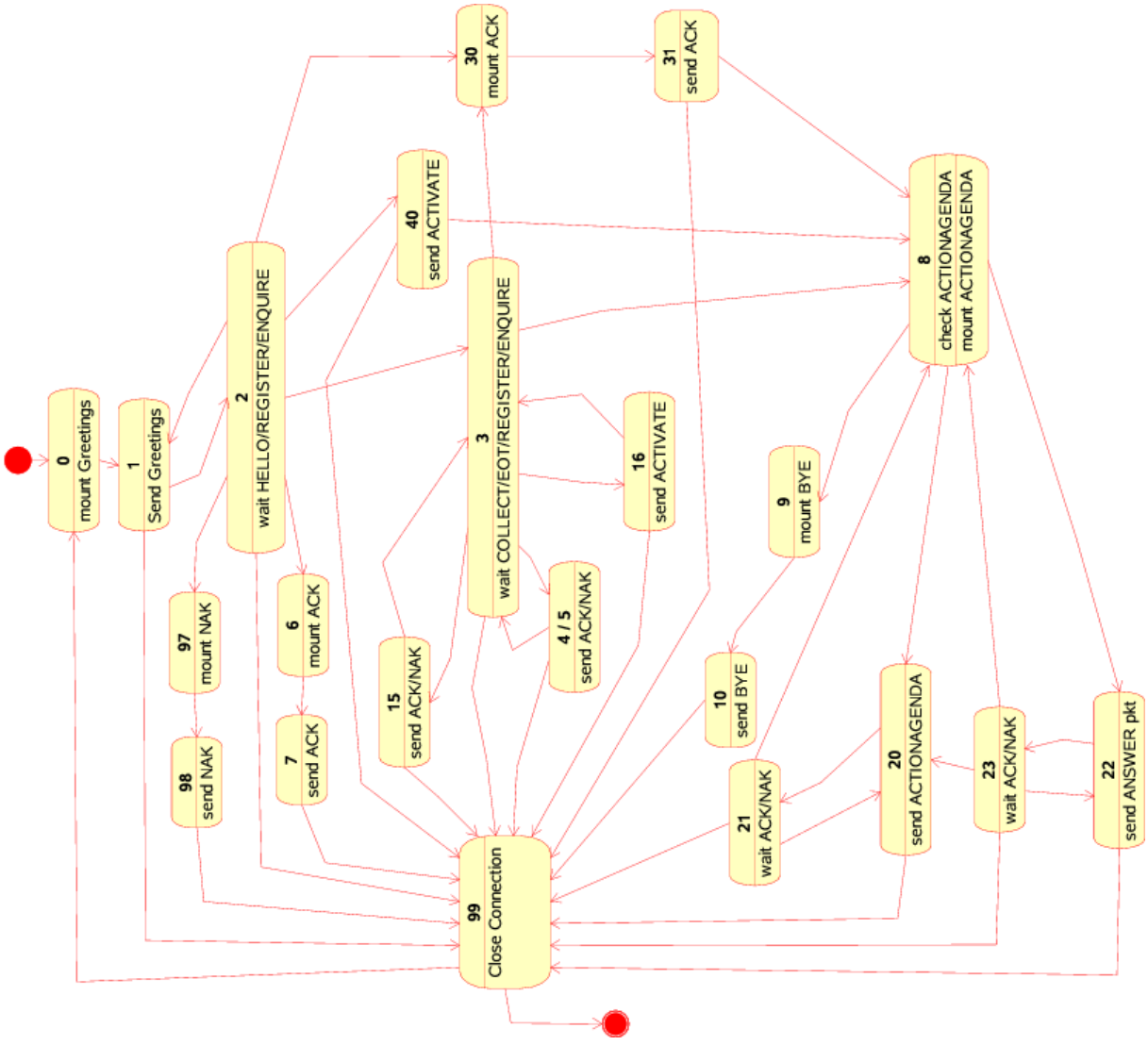


Figura 2: Máquina de Estados de Recepção de Variáveis.

# *Referências Bibliográficas*


[1] *CERT® Advisory CA-2002-03*.
http://www.cert.org/advisories/CA-2002-03.html.

[2] *Common Information Model (CIM) Standards*.
http://www.dmtf.org/standards/cim.

[3] *Cygwin*.
http://www.cygwin.com.

[4] *Distributed Management Task Force*.
http://www.dmtf.org.

[5] *IBM Tivoli*.
http://www.ibm.com/developerworks/tivoli.

[6] *JasperReports*.
http://www.jasperforge.org.

[7] *MySQL*.
http://www.mysql.com.

[8] *Nagios*.
http://www.nagios.org.

[9] *OpenNMS*.
http://www.opennms.org.

[10] *OpenPegasus*.
http://www.openpegasus.org.

[11] *OpenWBEM*.
http://www.openwbem.org.

[12] *RuralWeb*.
http://www.ruralwebtelecom.com.br.

[13] *Solaris WBEM Developer's Guide*.
http://docs.sun.com.

[14] *SQLite*.
http://www.sqlite.org.

[15] *Velox*.
http://www.velox.com.br.

[16] *Windows Management Instrumentation*. http://www.microsoft.com/whdc/system/pnppwr/wmi.

[17] Wolfgang Barth. *Nagios: System and Network Monitoring*. No Starch Press, 2006.

[18] P. Bhagwat. Bluetooth: technology for short-range wireless apps. *Internet Computing, IEEE*, 5:96–103, May/Jun 2001.

[19] J. D. Case, M. Fedor, M. L. Schoffstall, and C. Davin. RFC 1157: Simple network management protocol (SNMP), May 1990. See also STD0015. Obsoletes RFC1098. Status: STANDARD.

[20] Chris Hobbs. *A Practical Approach to WBEM/CIM Management*. CRC Press, Inc., Boca Raton, FL, USA, 2004.

[21] Paul Horn. autonomic computing: IBM's perspective on the state of information technology. Technical report, International Business Machines Corporation, Armonk, NY, USA, 2001.

[22] David Josephsen. *Building a Monitoring Infrastructure with Nagios*. Prentice Hall, 2007.

[23] Jeffrey O. Kephart and David M. Chess. Cover feature: The vision of autonomic computing. *j-COMPUTER*, 36(1):41–50, jan 2003.

[24] Lundy Lewis. *Managing computer networks: a case-based reasoning approach*. Artech House, 1995.

[25] Alain Lissoir. *Understanding WMI Scripting*. Digital Press, 2003.

[26] Douglas Mauro and Kevin Schmidt. *Essential SNMP*. O'Reilly, 2005.

[27] ISO/IEC 7498-4:1989: Information processing systems Open Systems Interconnection Basic Reference Model Part 4: Management framework. International organization for standardization, geneva, switzerland.

[28] S.L. Ricker and K. Rudie. A method for incorporating knowledge and communication into decentralized discrete-event systems.

[29] Mi-Jung Choi So-Jung Lee and Sun-Mi Yoo. Design of a wbem-based management system for ubiquitous computing services. *Management Developers Conference*, 2004.

[30] Willian Stallings. *SNMP, SNMPv2, and RMON: practical network management*. Addison Wesley Longman, 1996.

[31] Craig Tunstall and Gwyn Cole. *Developing WMI Solutions: A Guide to Windows Management Instrumentation*. Addison-Wesley Professional, 2002.