

# Monitoração de Disponibilidade e Desempenho de Servidores Críticos usando uma Abordagem Descentralizada

Juliano Valentini , Luciano Paschoal Gaspary

<sup>1</sup>Programa Interdisciplinar de Pós-Graduação em Computação Aplicada – PIPCA  
Universidade do Vale do Rio dos Sinos – UNISINOS  
Av. Unisinos, 500 – CEP 93.022-000 – São Leopoldo, RS

{julianov, paschoal}@exatas.unisinos.br

**Abstract.** *Due to the growth of size and number of services provided by current computer networks, the centralized management paradigm tends to lead to a lot of management traffic and to the overload of the central management station (when a large number of devices is supposed to be monitored). Some approaches have been proposed by the scientific community to solve this problem. This paper describes the usage of Script MIB, proposed by the IETF, to monitor in a decentralized fashion the availability and performance of critical servers.*

**Resumo.** *Devido ao aumento do tamanho e da quantidade de serviços oferecidos pelas redes de computadores atuais, o paradigma centralizado de gerenciamento tende a provocar um aumento do tráfego de gerenciamento e a sobrecarregar a estação central (quando um grande número de dispositivos deve ser monitorado). Para amenizar essa problemática, algumas abordagens de gerenciamento distribuídas têm sido propostas pela comunidade científica. Este artigo descreve a utilização da MIB Script, proposta pelo IETF, para a monitoração descentralizada de disponibilidade e desempenho de servidores críticos.*

## 1. Introdução

As redes de computadores estão passando por um grande crescimento no tamanho e no número de serviços que oferecem. Normalmente são gerenciadas por um sistema central, que é responsável por controlar um grande número de elementos. Com o acréscimo do tamanho da rede, o gerente central acaba ficando sobrecarregado, a ponto de não ser mais capaz de monitorar todos os elementos da mesma. O problema da escalabilidade torna-se pior quando ocorre um problema e, além de continuar com a monitoração, o gerente precisa executar procedimentos de recuperação, sobrecarregando-o ainda mais. Outra limitação de um sistema centralizado está relacionada com o tráfego de gerenciamento, que muitas vezes é enviado de uma ponta a outra da rede, consumindo banda (em geral escassa) dos canais de longa distância.

Este trabalho relata um experimento utilizando uma solução alternativa ao gerenciamento centralizado. Se os problemas do gerenciamento centralizado são a sobrecarga do gerente central e o tráfego de gerenciamento, o que se busca é distribuir as tarefas de gerenciamento entre várias entidades da rede. Nesse contexto, o presente trabalho descreve a utilização da MIB Script [3], proposta pelo IETF, para a monitoração descentralizada de disponibilidade e desempenho de servidores críticos.

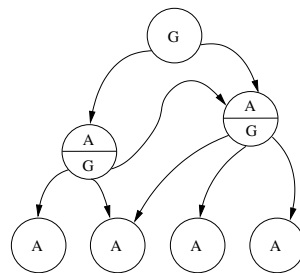
O artigo está organizado da seguinte forma: a seção 2 descreve alguns conceitos e classificações de gerenciamento distribuído. A seção 3 revisa a MIB Script, incluindo sua

estrutura e forma de utilização. Na seção 4 descreve-se o experimento realizado. O artigo encerra na seção 5 com as considerações finais.

## 2. Gerenciamento Distribuído

Uma solução de gerenciamento distribuído de redes, em geral, é composta por gerentes, agentes e entidades que desempenham ambos os papéis, chamadas de gerentes intermediários. Os gerentes são responsáveis pela delegação de tarefas de gerenciamento da rede. Os gerentes intermediários recebem essas tarefas e são responsáveis pela execução das mesmas. Os agentes monitoram variáveis e contabilizam informações sobre as estações da rede, que são armazenadas em uma MIB (*Management Information Base*). Os gerentes podem consultar informações ou receber alertas dos gerentes intermediários sobre as tarefas que foram delegadas e estão sendo executadas.

A figura 1 ilustra que as definições acima (gerente, gerente intermediário e agente) implicam uma estrutura hierárquica do sistema de gerenciamento. No topo da hierarquia está o gerente. Embaixo encontram-se os agentes. Os gerentes intermediários ficam entre o gerente e os agentes. Na próxima seção serão apresentadas as classes de sistemas de gerenciamento, onde ficará evidenciado que nem sempre é necessário instanciar todas essas entidades.



**Figura 1: Ilustração de um sistema de gerenciamento. Círculo com a letra G representa um gerente e círculo com a letra A, um agente. Os círculos divididos são os gerentes intermediários.**

### 2.1. Classes de Sistemas de Gerenciamento de Rede

Os sistemas de gerenciamento de rede podem ser classificados de acordo com o número de gerentes, gerentes intermediários e agentes que possuem. A seguir, são apresentadas quatro classes de sistemas de gerenciamento de rede, extraídas de [5]:

- $m = 1$ : gerenciamento centralizado
- $1 < m \ll n$ : gerenciamento fracamente distribuído
- $1 \ll m < n$ : gerenciamento fortemente distribuído
- $m \approx n$ : gerenciamento cooperativo

No esquema acima,  $m$  representa o número total de gerentes e gerentes intermediários e  $n$  o número total de elementos no sistema de gerenciamento, ou seja, a soma de  $m$  e o número de agentes.

Ao sair do gerenciamento centralizado em direção ao cooperativo, aumenta-se o grau de complexidade do sistema de gerenciamento. Para seja possível a descentralização é necessário um mecanismo que permita a distribuição das tarefas. Atualmente, a comunidade tem investigado muitas dessas tecnologias [1, 2, 8]. Nas seções

seguintes, será abordado uma dessas tecnologias – a MIB Script – utilizada como peça-chave da nossa arquitetura de gerenciamento por ser padronizada pelo IETF e compatível com SNMP (*Simple Network Management Protocol*).

### 3. MIB Script

A MIB Script [3] é um conjunto de objetos de gerenciamento que permite a delegação de *scripts* de gerenciamento para gerentes distribuídos fornecendo os seguintes atributos:

- transferência de *scripts* de gerenciamento para locais distribuídos;
- inicialização, suspensão, reinicialização e finalização dos *scripts* nesses locais;
- passagem de parâmetros para os *scripts* de gerenciamento;
- monitoração e controle dos *scripts* de gerenciamento que estão executando;
- transferência para o gerente dos resultados produzidos pelos *scripts* de gerenciamento que foram executados.

Os *scripts* a serem executados podem ser escritos em qualquer linguagem de programação, compilada ou interpretada, desde que esta seja suportada pela MIB Script. É possível estender a MIB Script de forma a suportar uma determinada linguagem que não é suportada nativamente.

#### 3.1. Uso da MIB Script

Observando a figura 2, a partir da estação de gerenciamento (1), o gerente de rede define *scripts* de monitoração e de ação. Esses *scripts* são transmitidos a um servidor Web (2) que funciona como um repositório de *scripts* (3). O gerente deverá, ainda, instalar os *scripts* que deseja executar na MIB Script via requisições SNMP (4). Nessas requisições são enviadas ao agente SNMP informações sobre os *scripts* como: nome, parâmetros, URL do *script*, linguagem em que foi escrito, descrição, entre outros. A MIB Script faz *download* do *script* (5, 6). Instalado o *script*, o gerente poderá, através de uma requisição SNMP (4), disparar e controlar a execução do mesmo.

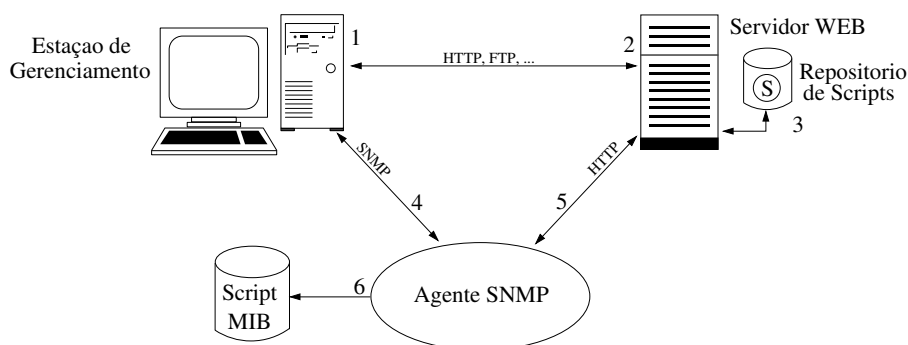


Figura 2: Uso da MIB Script

#### 3.2. Cenários de Uso

Nesta seção serão descritos alguns cenários de gerenciamento distribuído que poderiam ser implantados com base na MIB Script. Um dos objetivos da MIB Script é servir como suporte a gerentes intermediários em um sistema de gerenciamento distribuído. A idéia é implementar funcionalidades de gerentes intermediários através de *scripts* e executá-los dinamicamente em segmentos de rede remotos.

1. *Scripts para monitoração*: um gerente intermediário poderia monitorar um conjunto de variáveis de uma MIB em vários elementos da rede com o objetivo de detectar comportamentos irregulares ou para fins de contabilizar informações. O gerente intermediário poderia ser consultado pelo gerente ou reportar os resultados a ele enviando notificações.
2. *Scripts para teste de serviços*: um cenário empregando um gerente intermediário mais complexo é o que realiza teste de serviços. Esses testes incluem verificação da disponibilidade do serviço e os parâmetros estáticos e dinâmicos passados ao serviço. Uma maneira para implementar o teste serviços é simular o comportamento de usuários. Isso pode ser útil, por exemplo, quando a acessibilidade de certos documentos de um servidor web é essencial. Um *script* pode ser instalado em muitos locais remotos para verificar a disponibilidade e tempo de resposta regularmente e notificar o gerente em caso de falha.
3. *Scripts para gerenciamento e controle de serviços*: um serviço de rede pode ser implementado para executar processos de forma distribuída em várias instâncias da rede. Esses processos podem ser controlados por um gerente intermediário local executando no mesmo segmento de rede que o processo. Esses gerentes intermediários podem iniciar todos os processos pertencentes ao serviço, receber mensagens de erro desses processos, monitorar suas operações, reiniciar ou abortar o processo, se necessário, e notificar o gerente em caso de problemas sérios.

#### 4. Ambiente de Monitoração Configurado

O ambiente configurado tem como objetivo realizar a monitoração de servidores críticos de uma rede corporativa. A idéia é distribuir gerentes intermediários em locais estratégicos de forma que o gerente central possa delegar tarefas a eles relacionadas à análise de disponibilidade e desempenho desses servidores. Assim, esse gerente central passa a receber somente informações que são relevantes (por exemplo, notificação de que um serviço está indisponível).

Para que o gerente intermediário pudesse receber *scrips* do gerente central e executá-los foi preciso instalar dois *softwares* principais nas estações que desempenham esse papel. O primeiro foi o NET-SNMP [4], agente SNMP responsável por processar requisições e enviar respostas. O segundo foi o *Jasmin* [6], uma implementação da MIB Script. Por outro lado, para obter as informações dos servidores críticos foi utilizado um agente SNMP que implementa a MIB Host Resources (em cada servidor), definida em [7].

A MIB Host Resources define um conjunto de objetos para gerenciamento de computadores. As informações armazenadas por essa MIB são independentes de sistema operacional, serviços de rede ou qualquer programa aplicativo. Os dados estão organizados em tabelas que contêm informações gerais sobre o sistema, armazenamento, dispositivos instalados, programas e outras informações relevantes ao administrador de sistema. Dispondo das informações da MIB Host Resources, acessadas remotamente via requisições SNMP, o gerente pode tomar decisões em relação ao *host*.

O experimento realizado monitora as variáveis *hrSystemUptime* que acumula o tempo desde a última vez que o *host* foi inicializado, e *hrProcessorLoad* que mantém a porcentagem de carga média do processador durante o último minuto. Para realizar essa monitoração foi criado um *script* Java, ilustrado na figura 3, que gera um alarme caso detecte que o sistema foi reiniciado ou quando o processador ficou com uma carga superior a 90%.

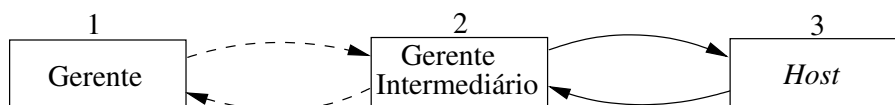
```

...
{
  ...
  SnmpPeer peer = new SnmpPeer("YODA",
                              InetAddress.getByName("200.132.73.112"),
                              "public" );
  SnmpConnection connection = new SnmpConnection(peer);
  Vector varbinds = new Vector();
  OID oid = new OID("1.3.6.1.2.1.25.1.1");
  varbinds.addElement(oid);
  oid = new OID("1.3.6.1.2.1.25.3.3.1.2");
  varbinds.addElement(oid);
  while (true) {
    Varbind[] result = connection.getNextRequest(varbinds);
    if (result == null) {
      out.println("No response");
    }
    else {
      alarm1.clear();
      alarm1 = getAlarm("reboot");
      newUptime = Double.parseDouble(result[0].value.content);
      if (newUptime < oldUptime) {
        alarm1.update("Host has been rebooted"); }
      else {
        alarm1.clear(); }
      oldUptime = newUptime;
      alarm2.clear();
      alarm2 = getAlarm("overload");
      ProcessorLoad = Integer.parseInt(result[1].value.content);
      if (ProcessorLoad > 90) {
        alarm2.update("Host has been overloaded"); }
      else {
        alarm2.clear(); }
    }
  }
}

```

**Figura 3: Fragmento do *script* Java executado pelo gerente intermediário**

Conforme pode ser observado na figura 4, a partir da estação de gerenciamento (1) o *script* de monitoração foi instalado e, em seguida, inicializado na MIB Script do gerente intermediário (2). Neste momento, o *host* (3) passou a ser monitorado continuamente pelo gerente intermediário.



**Figura 4: Ilustração do cenário do experimento. Configuração do gerente: CPU Intel Pentium 4 1,7GHz com 512MB de RAM e sistema operacional Linux Slackware 8.1; Configuração do gerente intermediário: CPU Intel Pentium 1 200MHz com 48MB de RAM e sistema operacional Linux Slackware 7.1; Configuração do *Host*: CPU Intel Pentium 4 1,7GHz com 512MB de RAM e sistema operacional Windows XP. As linhas contínuas representam tráfego constante e as linhas tracejadas representam tráfego eventual.**

Durante o experimento, foi medido o tamanho das requisições SNMP. Observou-se que para monitorar a variável *hrSystemUptime*, por exemplo, consome-se 168 bytes (82 bytes para GET e 86 bytes para RESPONSE). Fazendo uma projeção, em uma abordagem de gerenciamento centralizado, de um cenário onde o gerente central é responsável pela monitoração de 120 servidores e que, em cada servidor, devem ser monitoradas 20 variáveis do mesmo tamanho, chega-se a um total de 403.200 bytes de tráfego por ciclo de monitoração. Em uma abordagem de gerenciamento distribuído, introduzindo-se 6 gerentes intermediários, os quais, cada um, ficaria responsável por gerenciar 20 dos 120 servidores, obtém-se uma redução de aproximadamente 83,34% do tráfego entre os gerentes intermediários e os servidores (*hosts*), ou seja, carga dividida entre os 6 gerentes

intermediários. Já o tráfego entre o gerente central e os gerentes intermediários pode ser considerado desprezível, tendo em vista que o gerente central faz apenas duas requisições. Uma para instalar os *scripts* de monitoração e de ação na MIB Script e outra para disparar a execução dos mesmos. Quando ocorre algum problema na rede o gerente central apenas recebe um alerta.

A sobrecarga originalmente concentrada na estação de gerenciamento é, agora, distribuída entre os gerentes intermediários. A carga de cada gerente intermediário está diretamente relacionada com o número de tarefas que lhe forem delegadas. Assim, é importante destacar a relevância da organização dos domínios de gerenciamento, a fim de balancear as tarefas a serem executadas entre os gerentes intermediários e evitar que estes sejam sobrecarregados.

## 5. Conclusões

O processo de gerenciamento de grandes de redes utilizando gerenciamento centralizado é uma tarefa cada vez mais complexa, tendo em vista o grande conjunto de serviços e o volume de tráfego que o gerenciamento demanda. Existe, também, a dificuldade de uma única estação dar conta de processar toda as requisições de monitoração e, ainda, executar as tarefas de reparo. Este artigo descreveu um experimento envolvendo o uso da MIB Script para implantar tarefas de gerenciamento descentralizadas.

A maior dificuldade encontrada neste trabalho foi a instalação e a configuração da MIB Script – o Jasmin. O projeto desta implementação foi concluído no ano de 2001. Portanto, é necessário configurar um ambiente utilizando os *softwares* e tecnologias daquela época, entre elas o JDK 1.1, por exemplo. Apesar da vasta documentação teórica sobre o Jasmin, a documentação da parte prática deixa a desejar, tornando-se difícil saber exatamente que versões de *software* são necessárias. A MIB Script necessita, ainda, do UCD-SNMP, que atualmente foi migrado para o NET-SNMP. Por tudo isso, conclui-se que o Jasmin precisa passar por uma grande atualização permitindo o seu uso junto com as novas tecnologias que surgiram desde então. Ao contrário disso, o seu uso acabará se tornando incompatível e inviável.

## Referências

- [1] GOLDSZMIDT, German S. **Distributed Management by Delegation**. 1996. Ph.D Thesis – Graduate School of Arts and Sciences, Columbia University, New York.
- [2] KAHANI, M. and BEADLE, H.W. P. Decentralized Approaches for Network Management. **Computer Communications Review**, v. 27, n. 3, p. 36-47, July 1997.
- [3] LEVI, D. and SCHÖNWÄLDER, Jürgen. **Definitions of Managed Objects for the Delegation of Management Scripts**, Nortel Networks, TU Braunschweig, RFC 3165, August 2001. Disponível em <<http://www.ietf.org>>.
- [4] NET-SNMP. **NET-SNMP Project Homepage**. Disponível em <<http://net-snmp.sourceforge.net>>.
- [5] SCHÖNWÄLDER, J. et al. Building Distributed Management Applications with the IETF Script MIB. **IEEE Journal on Selected Areas in Communications**, New York, v. 18, n. 5, p. 702-714, March 2000.
- [6] TU BRAUNSCHWEIG, NEC C&C RESEARCH LABORATORIES. **Jasmin – A Script MIB Implementation**. Disponível em <<http://www.ibr.cu.tu-bs.de/projects/jasmin>>.
- [7] WALDBUSSER, S., GRILLO, P. **Host Resources MIB**, Lucent Technologies Inc., WeSync.com, RFC 2790, March 2000. Disponível em <<http://www.ietf.org>>.
- [8] SIEGL, M., TRAUSMUTH, G. Hierarchical Network Management: A Concept and Its Prototype in SNMPv2. **Computer Networks and ISDN Systems**, v. 28, n. 4, p. 441-452, 1996.