UMA FERRAMENTA DE GERENCIAMENTO DE REDES BASEADA EM AGENTES MÓVEIS

PEDRO BRACONNOT VELLOSO*, MARCELO GOLÇALVES RUBINSTEIN†, OTTO CARLOS M. B. DUARTE*

*Grupo de Teleinformática e Automação
COPPE/EE – Programa de Engenharia Elétrica
Universidade Federal do Rio de Janeiro

†Dept. de Eng. Eletrônica e Telecomunicações
Universidade do Estado do Rio de Janeiro

Emails: pvelloso@gta.ufrj.br, rubi@uerj.br, otto@gta.ufrj.br

Resumo— Nos últimos anos, diversas propostas baseadas no paradigma de agentes móveis têm sido apresentadas com o objetivo de aumentar a eficiência no gerenciamento de redes de grande escala. Este trabalho propõe e implementa uma ferramenta de gerenciamento de redes baseada em agentes móveis e uma plataforma simplificada de agentes móveis, que objetivam avaliar a eficiência e a flexibilidade providas por esta nova abordagem. A ferramenta permite ao usuário criar, editar e executar tarefas de gerenciamento e a plataforma simplificada possibilita o envio de mensagens, a migração e a execução dos agentes.

Abstract— *In the last few years, several proposals based on mobile agents paradigm have been presented in order to increase large scale network management efficiency. This paper presents the design and the implementation of a network management application based on mobile agents and a simple mobile agent infrastructure, that aim to evaluate the efficiency and the flexibility provided by this new approach. The application enables the user to easily create, edit, and perform management tasks. The infrastructure provides agent communication, migration and execution.*

Key Words— Mobile Agents, Network Management.

1 Introdução

As aplicações de gerenciamento convencionais utilizam o paradigma cliente/servidor. Um dos principais problemas deste modelo é sua característica centralizada, que implica uma sobrecarga de processamento na estação de gerenciamento e um enorme gasto da banda passante da rede para a troca de mensagens com os elementos gerenciados. Uma das propostas para solucionar este problema é a utilização de agentes móveis, que são programas capazes de se locomover através de uma rede de computadores, podendo ser executados em diferentes nós da rede. Este novo paradigma permite descentralizar a tarefa de gerenciamento, diminuir o tráfego da rede, flexibilizar a realização das tarefas e introduzir conceitos de inteligência e aprendizado.

Diversos estudos vem sendo realizados com o objetivo de avaliar o desempenho dos agentes móveis. Ismail *et al.* (Ismail e Hagimont, 1999) realizam uma comparação entre o desempenho dos agentes móveis e do modelo cliente/servidor mostrando que o uso de agentes móveis pode possibilitar um significativo aumento de desempenho. Bohoris *et al.* (Bohoris et al., 2000) apresenta um sistema de monitoração de desempenho de redes baseado em agentes móveis e compara o seu desempenho com CORBA e Java-RMI. Outros pesquisadores utilizam agentes inteligentes no gerenciamento de redes. Neuman *et al.* (da Silva et al., 2001) aplicam técnicas de inteligência artificial, baseadas em lógica difusa, em agentes móveis, a fim de realizar um gerenciamento pró-ativo distribuído, enquanto Oliveira *et al.* (Nascimento e Oliveira, 1999) utilizam redes neurais para gerenciar redes ATM de modo pró-ativo.

Este trabalho descreve uma ferramenta de gerenciamento de redes baseada em agentes móveis, visando reduzir o tempo gasto e o número de mensagens trocadas na realização desta tarefa e proporcionar uma maior flexibilização no gerenciamento. Além disso, devido ao alto custo computacional apresentado pela maioria das plataformas de agentes móveis, foi desenvolvida uma plataforma simplificada que implementa apenas as necessidades básicas da ferramenta proposta, de modo a reduzir o custo computacional e proporcionar uma melhor integração com a aplicação de gerenciamento. Essa aplicação permite ao usuário definir tarefas e analisar de forma amigável os dados obtidos. A plataforma de agentes móveis é o ambiente de mobilidade que permite a execução, a migração e a troca de mensagens entre os agentes.

Este trabalho está organizado da seguinte forma. Na Seção 2 são definidos o conceito de agentes móveis e as suas características. A Seção 3 descreve o modelo de gerenciamento e a arquitetura da aplicação. Detalhes referentes à implementação da ferramenta são apresentados na Seção 4. Por fim, na Seção 5 são apresentadas as conclusões deste trabalho e trabalhos futuros.

2 Os agentes móveis

O conceito de agentes móveis é baseado no paradigma da programação remota (*Remote Programming* - RP), que consiste basicamente em enviar um procedimento para ser executado em outro hospedeiro[1]. As principais vantagens (Lange e Oshima, 1999) oferecidas pelos agentes móveis são a flexibilidade, o processamento assíncrono e a capacidade de aprendizagem. Em relação à flexibilidade, os procedimentos não precisam ser preestabelecidos, permitindo que os agentes sejam criados independentemente da máquina na qual serão executados. A mobilidade de código proporciona um processamento assíncrono e distribuído. O código é independente do processo que o gerou e desta forma, a máquina que enviou um agente para realizar uma determinada tarefa em outro computador pode encerrar a conexão, ou até mesmo ser desligada, sem prejudicar a realização da tarefa. A capacidade de aprendizagem provê um alto grau de inteligência ao agente.

As aplicações de agentes móveis necessitam de ambientes que permitam a construção, o uso e a manutenção de agentes móveis. Estes ambientes, denominados plataformas de agentes móveis, provêem os serviços de migração, execução, comunicação, segurança e tolerância a falhas.

A plataforma Aglets (IBM, n.d.), implementada em Java, é uma das mais utilizadas por ser considerada uma plataforma de uso simples e por ser bem documentada. No entanto, o estudo realizado por Silva *et al.* (Silva et al., 2000) evidenciou alguns problemas de desempenho na plataforma Aglets, principalmente, devido ao protocolo ATP (*Agent Transfer Protocol*) que viabiliza a migração dos agentes.

Existem muitas outras plataformas baseadas em Java que oferecem suporte a sistemas de agentes móveis. Algumas delas são gratuitas e possuem o código disponível, como a Mole (Uni, 1995), enquanto que outras são pagas, como a Concordia (Mit, n.d.), e possuem apenas uma versão limitada disponível gratuitamente.

Um problema apresentado pela maioria das plataformas é o alto custo computacional, que muitas vezes prejudica o desempenho das aplicações. Por isso, desenvolveu-se neste trabalho uma infraestrutura simplificada de agentes móveis que implementa apenas as necessidades básicas da aplicação proposta, de modo a reduzir o custo computacional.

3 A ferramenta de gerenciamento proposta

O gerenciamento de redes convencional utiliza o protocolo SNMP (*Simple Network Management Protocol*) (Case et al., 1990). O grande problema do uso do SNMP deve-se ao fato de os sistemas de gerenciamento de redes mais complexos necessitarem de uma volumosa troca de informações entre o gerente e o agente e, como conseqüência, de uma grande quantidade de mensagens enviadas na rede. Uma das propostas para contornar este problema é a utilização do paradigma de agentes móveis.

Neste novo modelo, o gerente envia um agente móvel para executar a tarefa de gerenciamento localmente. Desta forma, o agente pode percorrer todos os nós da rede, armazenando as informações obtidas na execução da tarefa. Ao final da tarefa, o agente deve enviar os resultados para o gerente. A Figura 1 mostra um esquema de gerenciamento com agentes móveis, apresentado em (Rubinstein et al., 2000), e implementado neste trabalho.

Os agentes móveis são responsáveis por coletar as informações solicitadas pelo gerente em todos os nós da rede. Por questões de segurança, nas infraestruturas disponíveis, os agente móveis não possuem permissão para acessar todos os recursos disponibilizados por uma máquina. Assim, em cada elemento da rede é colocado um agente fixo, denominado agente tradutor, que desempenhará o papel de uma interface entre o agente móvel e o agente SNMP. Quando o gerente precisa de alguma informação ou fazer alguma atualização em uma variável da MIB (*Management Information Base*), basta definir um itinerário e enviar um agente móvel para realizar a tarefa. Em cada hospedeiro, o agente móvel passará o pedido para o agente tradutor, que fará o pedido para o agente SNMP, que por sua vez realizará a consulta ou a atualização na MIB e devolverá o resultado para o agente tradutor. O agente móvel receberá o resultado e migrará para o próximo nó, até percorrer todo o seu itinerário.

Por outro lado, ao invés de enviar apenas um agente para percorrer um determinado itinerário, a aplicação permite ao usuário enviar mais de um agente ao mesmo tempo para executar a mesma tarefa, a fim de minimizar o tempo de resposta. Neste caso, o itinerário é divido pelo número de agentes.

A ferramenta proposta neste trabalho provê um ambiente de gerenciamento baseado em agentes móveis que permite ao usuário criar, editar e executar tarefas de gerenciamento. Além disso, a ferramenta possibilita a definição de características básicas de gerenciamento, como a operação a ser executada, as variáveis a serem coletadas e o itinerário a ser percorrido, bem como características adicionais que definem como a tarefa deverá ser executada pelos agentes móveis. A ferramenta permite a utilização de múltiplos agentes na realização de uma mesma tarefa, a fim de maximizar a eficiência.

3.1 *A tarefa de gerenciamento*

O que define uma tarefa são as suas características, que podem ser classificadas em básicas e complementares. As características básicas de uma tarefa são aquelas que não podem ser definidas pelo programa, mas apenas pelo usuário e são essenciais para a sua execução. As características básicas são, na verdade, os atributos

[1] O conceito de programação remota surgiu como uma alternativa à chamada de procedimento remoto (*Remote Procedure Call* - RPC).

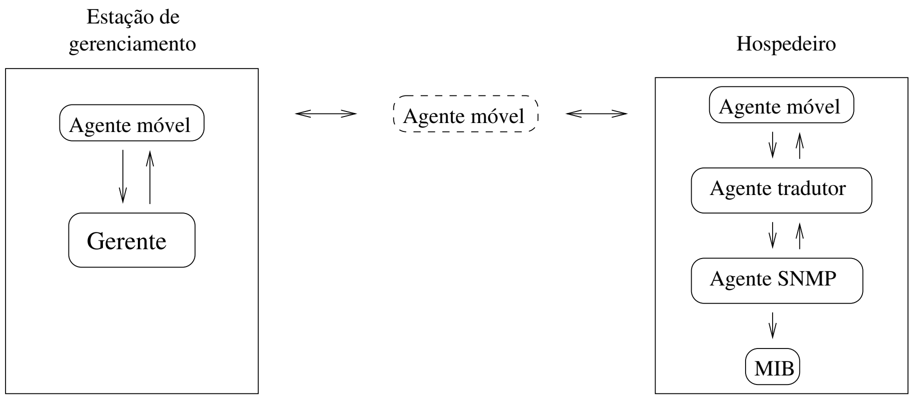

Figura 1: Modelo de gerenciamento baseado em agentes móveis.

da tarefa: o nome da tarefa, a operação a ser executada, podendo ser uma atualização ou uma consulta de uma variável da MIB, as variáveis de interesse para a tarefa e o itinerário, que contém a relação das máquinas que serão percorridas pelos agentes.

As características complementares apenas adicionam informações à tarefa e, caso não sejam definidas, o programa assume valores padrão para elas. Estas características são:

- falha: representa o que o agente deve fazer em caso de falha ao tentar migrar para um determinado nó. Ele pode seguir para o próximo nó ou retornar à origem, abortando a tarefa;
- entrega dos resultados: especifica a maneira como os resultados devem ser entregues ao gerente. Os resultados podem ser enviados pelo agente móvel em um nó especificado no itinerário ou o agente móvel pode retornar ao nó de origem carregando os resultados e entregá-los ao gerente;
- freqüência: determina a freqüência com que os resultados serão entregues. O valor padrão é um, ou seja, os resultados são entregues a cada nó visitado;
- medidas de tempo: define quais as medidas de tempo deverão ser realizadas. Pode ser medido o tempo total para a execução de uma tarefa e o tempo de execução de uma operação pelos agentes móveis;
- número de agentes: especifica o número de agentes que irão executar a tarefa. Quando mais de um agente é especificado para executar a tarefa, o itinerário é dividido de forma que a diferença do número de estações entre os novos itinerários não seja maior que um; isto quer dizer que um agente visitará o mesmo número ou, no máximo, uma estação a mais que outros agentes;
- número de execuções: determina o número de vezes que a tarefa será executada;
- intervalo: determina o intervalo de tempo, em segundos, entre duas execuções de uma tarefa, caso esta seja executada mais de uma vez.

3.2 A arquitetura da ferramenta

A arquitetura da ferramenta é dividida em camadas hierárquicas com quatro níveis: usuário, controle, agente e comunicação, como pode ser visto na Figura 2. O primeiro nível tem a função de prover a interface com o usuário. O nível de controle envolve tanto o controle das tarefas que estão sendo realizadas quanto o controle dos agentes móveis criados para executar as tarefas. O nível de agente representa os agentes de uma forma geral, incluindo os agentes móveis e os agentes tradutores (fixos). O último nível lida com toda a parte de comunicação da aplicação.

As camadas inferiores disponibilizam métodos para as camadas superiores. Desta maneira, cada camada interage apenas com as camadas adjacentes. A única exceção ocorre quando o servidor recebe mensagens dos agentes, pois ele entrega os resultados diretamente ao controlador de tarefas. Cada camada é composta por entidades que são responsáveis por prover os serviços prestados às camadas superiores. As entidades são:

- interface gráfica: propicia um ambiente no qual o usuário pode criar, editar ou executar uma tarefa de gerenciamento. O usuário também pode visualizar os resultados das operações realizadas pelos agentes;
- controlador de tarefas: tem a função de criar e executar as tarefas de gerenciamento. Ele também é responsável por escolher que tipo de agente será executado para cada tarefa;
- controlador de agentes: tem a função de executar os agentes e armazenar todas as informações necessárias para controlar todos os agentes em execução em uma mesma máquina;
- agentes móveis: são responsáveis por executar as operações de gerenciamento em cada nó da rede e entregar os resultados à estação de gerenciamento;
- agente tradutor: permite a comunicação do agente móvel com o agente SNMP. Desta forma, ele deve entregar os pedidos dos agentes móveis

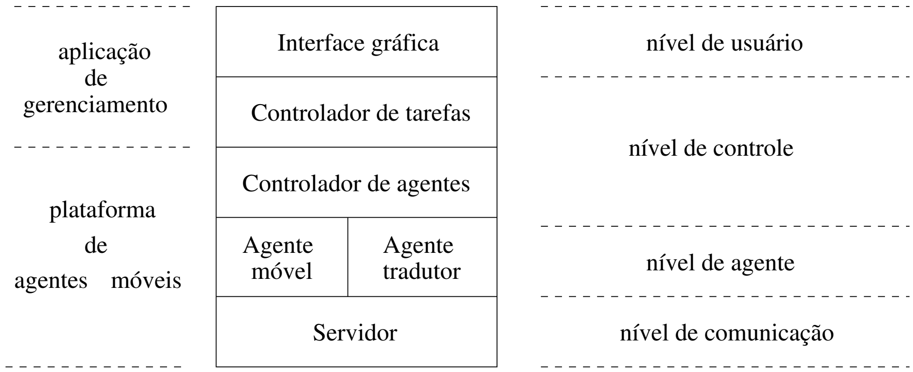

Figura 2: Modelo em camadas da ferramenta.

para os agentes SNMP e, em seguida, retornar o resultado para o agente móvel. O agente móvel poderia se comunicar diretamente com o agente SNMP, mas foi utilizado um agente intermediário para não sobrecarregar o código do agente móvel e por questões de segurança;

- servidor: é o responsável por estabelecer a conexão entre as máquinas da rede. Neste trabalho, o servidor foi dividido em dois processos: um é responsável por enviar e receber as mensagens e o outro envia e recebe os agentes móveis.

4 A implementação da ferramenta

Este projeto foi implementado na linguagem de programação Java devido à portabilidade, à simplicidade e ao alto grau de reaproveitamento de código desta linguagem. Desta forma, esta será uma aplicação multiplataforma, podendo ser executada em qualquer sistema operacional. Além destas características, Java possui um mecanismo, chamado serialização, que permite armazenar o estado de um objeto na forma de *bytes* e depois recuperá-lo. O estado de um objeto contém, basicamente, informações sobre a classe e os valores de suas variáveis. Este objeto serializado pode ser armazenado tanto em um arquivo quanto em um *socket*. Isto possibilita que um programa, após ser interrompido, continue a execução a partir de onde foi interrompido pela última vez. Assim, este mecanismo viabiliza a implementação de agentes móveis.

Neste trabalho foi utilizado o *Java Development Kit* (JSDK 1.3.1) que possui um compilador e uma máquina virtual Java. Além disto, também foram utilizadas classes do pacote *AdventNetSnmpV1* (Adv, n.d.), que implementam a comunicação com os agentes SNMP.

4.1 A infra-estrutura simplificada de agentes móveis

A infra-estrutura de agentes móveis implementada contém dois servidores que viabilizam o envio de mensagens dos agentes para o gerente (servidor de mensagens), a execução e a mobilidade dos agentes (servidor de agentes).

A mobilidade é a principal característica dos agentes móveis e pode ser classificada de duas maneiras. A mobilidade "forte" (*strong mobility*), se caracteriza pela capacidade do agente de armazenar o contexto de execução, de maneira que permita ao agente começar sua execução pela instrução seguinte àquela na qual foi interrompida. Na mobilidade "fraca" (*weak mobility*) o agente não armazena o contexto de execução e por este motivo, deve ser executado do começo a cada vez que ele se mover. Como decidiu-se não utilizar nenhuma das plataformas de agentes móveis disponíveis na *Internet*, foi necessária a implementação de servidores que viabilizassem a mobilidade destes agentes e a troca de mensagens entre os agentes e a estação de gerenciamento. Neste trabalho, foi implementada a mobilidade "fraca".

O mecanismo de comunicação utilizado na implementação dos servidores foi o *Remote Method Invocation* - RMI. O RMI é a implementação em Java do RPC. Desta maneira, o RMI possibilita que um objeto armazenado em uma estação qualquer da rede possa invocar métodos em uma outra máquina virtual Java de uma outra estação da rede. Para implementar a mobilidade e a comunicação dos agentes utilizou-se o protocolo de transporte TCP (*Transfer Control Protocol*).

4.2 A interface gráfica

A interface permite ao usuário interagir com a aplicação, criando, editando, executando tarefas e visualizando os resultados das tarefas executadas. A Figura 3 mostra uma das telas da ferramenta, na qual o usuário deve escolher as características básicas da tarefa, como o nome da tarefa, a operação a ser executada, o que o agente deve fazer em caso de falha ao tentar migrar para algum nó e as variáveis que serão consultadas ou atualizadas.

Existem ainda outras telas que permitem ao usuário especificar as demais características de uma deter-

Figura 3: Tela criar tarefa - características.

minada tarefa, editar o itinerário e visualizar os resultados obtidos.

5 Conclusões e Trabalhos Futuros

Este trabalho teve como objetivo desenvolver e implementar uma aplicação de gerenciamento utilizando a tecnologia de agentes móveis que permitisse ao usuário realizar, com facilidade, tarefas de gerenciamento.

O projeto foi desenvolvido em Java devido à simplicidade, à portabilidade e ao alto grau de reaproveitamento de código que a linguagem proporciona. Ele é dividido em duas partes principais: a aplicação de gerenciamento e a plataforma simplificada de agentes móveis.

A aplicação de gerenciamento implementada permite realizar medidas relacionadas ao tempo de execução das tarefas, o que é de grande utilidade para estudos de desempenho, tendo em vista que esta é uma nova área e, por isso, ainda está sendo muito investigada. A plataforma de agentes móveis implementada é uma infra-estrutura simplificada que possibilita a execução, a migração e a comunicação dos agentes. Além disso, com algumas modificações, a plataforma pode servir como base para outras aplicações.

A principal contribuição deste trabalho é o fato de ter sido desenvolvida uma aplicação que não somente contém toda uma infra-estrutura básica para realização de tarefas de gerenciamento utilizando agentes móveis como também provê muitas outras funcionalidades. Além disto, a aplicação é inteiramente dedicada, pois a parte referente à infra-estrutura dos agentes móveis foi implementada especificamente para esta aplicação.

Outro ponto importante a ser ressaltado é a modularidade da aplicação, que permite que novas classes e novos agentes sejam criados e adicionados ao programa principal. Assim, esta aplicação pode ser utilizada para novos estudos sobre o uso de agentes móveis no gerenciamento de redes. Um exemplo seria a criação de agentes inteligentes.

Como trabalhos futuros podem-se acrescentar novas funcionalidades à aplicação, como por exemplo a visualização dos resultados através de gráficos, o que possibilitaria monitorar variáveis ao longo do tempo de uma maneira muito mais eficiente. Outra possibilidade seria o desenvolvimento de agentes inteligentes que, de alguma forma, fossem capazes de reconhecer o comportamento da rede e notificar os eventos ou, até mesmo, tomar alguma decisão.

Referências

Adv (n.d.). *AdventNetSNMPv1*.

Bohoris, C., Liotta, A. e Pavlou, G. (2000). Software agent constrained mobility for network performance monitoring, *6th IFIP Conference on Intelligence in Networks (SmartNet 2000)*, Viena, Áustria, pp. 367–387.

Case, J., Fedor, M., Schoffstall, M. e Davin, J. (1990). Simple management network protocol, *Internet RFC 1157* .

da Silva, A. N., Ferraz, C. A. G., Ramalho, G. L. e de Souza, J. N. (2001). Proactive network management based on mobile agents and fuzzy logic, 19° *Simpósio Brasileiro de Telecomunicações (Telecom XXI)*, Fortaleza, Brasil.

IBM (n.d.). *Aglets*.

Ismail, L. e Hagimont, D. (1999). A performace evaluation of the mobile agent paradigm, *Int. Conf. on Object-Oriented Programming, Systems and Applications (OOPSLA'99)*, Denver, EUA, pp. 306–313.

Lange, D. B. e Oshima, M. (1999). Seven good reasons for mobile agents, *Communications of the ACM* **42**(3): 88–89.

Mit (n.d.). *Concordia*.

Nascimento, A. e Oliveira, M. F. M. (1999). Desenvolvendo agentes inteligentes para a gerência próativa de redes ATM, *Simpósio Brasileiro de Redes de Computadores*, pp. 681–696.

Rubinstein, M. G., Duarte, O. C. M. B. e Pujolle, G. (2000). Evaluating the network performance management based on mobile agents, *Lectures Notes in Computer Science* **1931**: 95–102.

Silva, L., Soares, G., Martins, P., Batista, V. e Santos, L. (2000). Comparing the performance of mobile agent systems: a study of benchmarking, *Computer Communications* **23**(8): 769–778.

Uni (1995). *The Home of Mole*.