A Viabilidade do Gerenciamento de Desempenho Pró-ativo Baseado em uma Arquitetura de Gerenciamento que usa Tecnologia Ativa

Edmundo Lopes Cecilio, Ana Paula Magalhães Dumont, Flávia Coimbra Delicato, Luiz Fernando Rust da Costa Carmo e Luci Pirmez

Núcleo de Computação Eletrônica - Universidade Federal do Rio de Janeiro
Tel: 021 2598-3159 - Caixa Postal 2324, Rio de Janeiro, RJ, Brasil

dumont@unisys.com.br, cecilio@ime.eb.br, flavia@eng.uerj.br, luci, rust@nce.ufrj.br

Resumo

A garantia de níveis qualidade de serviço específicos depende fundamentalmente da presença de um mecanismo de gerenciamento preciso, eficiente e rápido, tanto dos recursos da rede quanto das aplicações e dos serviços de comunicação. Esse mecanismo de gerenciamento deve ser de fácil atualização e flexível o suficiente para a implantação de novas funcionalidades com a devida rapidez. O gerenciamento de desempenho, em particular, é de vital importância, pois é quem deve monitorar a situação da rede no que diz respeito ao desempenho dos serviços sendo prestados e indicar quando e porque os níveis desejados de QoS não estão sendo alcançados. A tecnologia ativa – redes ativas e agentes móveis – vem sendo considerada em recentes pesquisas para o provimento da flexibilidade e distribuição necessárias a próxima geração de futuros sistemas de gerenciamento. Este artigo apresenta uma proposta de arquitetura de gerenciamento de desempenho que, além de usar tecnologia ativa, é pró-ativa. O trabalho proposto é baseado na arquitetura de gerenciamento distribuída ativa AGDA, que está sendo desenvolvida no NCE/UFRJ. O protótipo implementado tinha como objetivo investigar a viabilidade de se implementar um sistema de gerenciamento de desempenho pró-ativo usando um paradigma de mobilidade de código baseado em Java, oμCode.

Abstract

Only a precise, efficient and rapid management mechanism will be able to guarantee the specific quality of service levels overseeing a network's resources and applications as well as its communications services. This mechanism must have the capacity to be easily updated and yet be flexible enough to promptly assimilate the introduction of new functionalities. Performance Management is of vital importance as it is responsible for both monitoring network status as regards services being offered and indicating when and why desired QoS levels have not been achieved. Recent research has shown that Active Technology - active networks and mobile agents - offers the flexibility and distribution assets required for management systems of the future. This paper proposes a new architecture of Performance Management that, in addition to utilizing active technology, is also pro-active. This proposal is based on an Active, Distributed Management Architecture (AGDA) designed by the authors and developed at the Núcleo de Computação Eletrônica

(NCE) of the Federal University of Rio de Janeiro (UFRJ). The goal of the prototype was to investigate the viability of implementing a pro-active performance system running on a Java-based mobile agent paradigm μCode.

Palavras-chaves: gerenciamento de desempenho pró-ativo, gerenciamento distribuído, redes ativas, agentes móveis, QoS.

1. Introdução

A Internet aumenta a sua abrangência significativamente e deverá, em breve, assumir o papel de principal infra-estrutura de comunicação. Entretanto, a tecnologia utilizada na implementação da Internet – hardware e protocolos – terá que evoluir, uma vez que essa rede deverá oferecer suporte a serviços complexos para diferentes tipos de tráfego com diferentes requisitos de qualidade de serviço (QoS).

O desempenho da Internet em sua configuração atual é, em geral, aquém do desejado, em função do serviço prestado ser apenas de melhor esforço. Além de haver redundâncias em algumas camadas de protocolo, não há priorizações no roteamento e encaminhamento de pacotes e nem há mecanismos de controle de admissão de fluxos, entre outros fatores.

Além do desempenho insatisfatório, a Internet, bem como as redes de comunicação em geral, caracterizam-se pela dificuldade na integração de novos padrões, novas tecnologias e na instalação de novos serviços. O ciclo de vida para essas integrações é muito longo, podendo se estender por uma década. O uso de tecnologias como agentes móveis [1] e redes ativas [4] tem sido pesquisado como um meio de se reduzir significativamente a duração desse ciclo de vida.

A garantia de níveis de serviço específicos depende fundamentalmente da presença de um mecanismo de gerenciamento preciso, eficiente e rápido. Esse mecanismo deve ser de fácil atualização e flexível o suficiente para a implantação de novas funcionalidades com a devida rapidez. O gerenciamento de desempenho, em particular, é de vital importância, pois é quem deve monitorar a situação da rede no que diz respeito ao desempenho dos serviços sendo prestados e indicar quando e porque os níveis desejados de QoS não estão sendo alcançados.

As abordagens tradicionais de gerenciamento, como SNMP e CMIP, geram tráfego excessivo e normalmente apresentam retardos elevados para a acusação de uma situação anômala ou para o desencadeamento de ações corretivas. Estas, por sua vez, costumam ser desencadeadas somente após a ocorrência de uma falha, ou seja, reativas. É desejável que o gerenciamento seja distribuído, visando a redução do tráfego e dos retardos e, preferencialmente, pró-ativo em vez de meramente reativo.

Este artigo apresenta uma arquitetura de gerenciamento de desempenho pró-ativo em desenvolvimento no Núcleo de Computação Eletrônica (NCE) da UFRJ, que é baseada na arquitetura AGDA [20], que provê uma infra-estrutura de gerenciamento distribuído baseado em tecnologia ativa. O ambiente do projeto ServiMídia [5], que implementa um sistema multimídia distribuído, é utilizado como plataforma para o desenvolvimento e para os testes. São também apresentados os resultados obtidos em testes realizados por um protótipo, que teve como principal objetivo investigar a viabilidade do uso de uma infra-estrutura de mobilidade baseada na linguagem Java para a execução de gerenciamento pró-ativo.

A seção 2 apresenta o conceito de tecnologia ativa, bem como a sua relevância para o gerenciamento distribuído. A seção 3 descreve a AGDA, detalhando os seus elementos, os relacionamentos entre os mesmos e o seu funcionamento. A seção 4 apresenta a proposta de arquitetura de gerenciamento de desempenho pró-ativa que é o foco deste artigo. A Seção 5 descreve o protótipo que foi implementado, bem como o ambiente de testes utilizado, apresentando também os resultados obtidos. A seção 6 apresenta um resumo dos principais trabalhos relacionados. Por fim, algumas conclusões estão apresentadas na seção 7.

2. Tecnologia ativa

A partir de 1995, após as primeiras publicações oriundas de pesquisa financiada pela DARPA, a distribuição do controle e da programabilidade dos nós de comutação de uma rede passaram a ser investigadas seriamente. O objetivo dessas investigações era prover uma maior flexibilidade, tanto no gerenciamento quanto na disponibilização rápida de novos serviços em redes de comunicação [6], sem que os longos processos de padronização de protocolos e de formato de pacote tiverem que ser esperados. Atualmente, existem dois paradigmas sendo considerados: redes ativas e agentes móveis. Ambos oferecem o suporte a modelos que utilizam recursos computacionais no interior e/ou nas bordas da rede para o carregamento e a execução de programas "sob demanda". Embora os conceitos dessas tecnologias tenham sido originados em diferentes comunidades de pesquisas, visando resolver diferentes problemas, eles começam a se superpor em termos de foco e aplicabilidade, fato que está popularizando o termo *tecnologia ativa* para as referências a um ou ambos os paradigmas [7].

2.1. Redes ativas

Uma rede é dita ativa [4], [6] quando seus nós de comutação – os nós ativos – em vez de realizarem apenas a entrega de dados, têm capacidade de realizar processamento genérico, normalmente em prol da entrega de dados, ou seja, nas camadas de rede e/ou de enlace. Dessa forma, os fluxos de dados podem transportar programas, além dos dados, ou sinalizações que desencadeiem o carregamento desses programas quando necessário. Cada nó ativo é composto por um hardware computacional, normalmente baseado em um processador de uso geral, por um sistema operacional, por um ou mais ambientes de execução e por aplicações ativas. Um computador pessoal com sistema operacional Linux e máquina virtual Java (o ambiente de execução) pode ser considerado um nó ativo. As aplicações ativas seriam *bytecodes* Java executadas nesse nó ativo.

2.2. Agentes móveis

Um agente de software é uma entidade computacional que age em prol de um usuário e que executa uma tarefa específica para a qual tenha sido designado. Para isso ele atua de forma contínua e autônoma, tanto reativa quanto pró-ativa, tem capacidade de aprender e cooperar com outros agentes e dispensa constante intervenção, em um ambiente que, possivelmente, é habitado por outros agentes e processos [3] e [8].

Agentes móveis [3], [8] e [9] são caracterizados pela sua habilidade de moverem-se entre diferentes localidades via redes de comunicação. Eles viabilizam a transformação das redes atuais em plataformas programáveis remotamente. As principais vantagens no uso de agentes

móveis em comparação com a abordagem cliente-servidor ou com outros tipos de agentes podem assim ser resumidas: eficiência, economia de espaço, redução do tráfego na rede, interação assíncrona e em tempo real, robustez, operação em ambientes heterogêneos, extensibilidade em tempo real e fácil atualização [3].

Para que seja possível o uso de agentes móveis é necessária a presença de uma infra-estrutura de mobilidade. Esta vai prover as facilidades que oferecerão suporte aos modelos de ciclo de vida, computacional, de segurança, de comunicação e de navegação dos agentes [3].

2.3. Redes ativas versus agentes móveis

A diferença fundamental entre as duas tecnologias é que as redes ativas usam o conceito de processamento nas camada de rede e/ou de enlace, ou seja, voltado para o roteamento e/ou encaminhamento de pacotes, enquanto agentes móveis executam como aplicações. Sistemas baseados em agentes móveis são projetados para a construção de um ambiente de computação distribuído e interligado por um sistema de comunicação. O propósito das redes ativas é flexibilizar e tornar mais eficiente o próprio sistema de comunicação, em termos de encapsulamento de protocolos, configuração, instalação e manutenção de serviços de comunicação.

Agentes móveis podem migrar baseados em decisões autônomas, além de poderem também gerar processos filhos ou threads para tratarem de problemas específicos. Essas funcionalidades não estão previstas nas redes ativas. Adicionalmente, agentes podem trocar mensagens entre si, algo também não previsto nas redes ativas [7].

As infra-estruturas já implementadas para o suporte a redes ativas e agentes móveis diferenciam-se ainda quanto à disponibilidade de funcionalidades de segurança e proteção do nó ativo. Essas funcionalidades estão presentes apenas nas infra-estruturas de redes ativas.

No gerenciamento de redes, pode-se considerar que as funcionalidades a serem executadas estão, principalmente, no nível das aplicações, uma vez que o gerenciamento não tem como objetivo principal tratar diretamente do roteamento e/ou do encaminhamento de pacotes. Mas também há a necessidade de sejam realizados certos processamentos no nível de rede. Sendo assim, no que diz respeito a infra-estruturas para gerenciamento, convém empregar-se o termo tecnologia ativa, que engloba tanto os agentes móveis quanto as redes ativas.

2.4. Tecnologia ativa no gerenciamento distribuído

Os elementos da rede que devem ser gerenciados podem ser providos apenas de funcionalidades básicas no que se refere à obtenção de informações de gerenciamento. Capacidades genéricas e específicas podem ser providas via agentes enviados a esses elementos. Adicionalmente, o comportamento desses agentes pode ser modificado a qualquer momento. Assim, uma grande flexibilidade das aplicações de gerenciamento é obtida, uma vez que os elementos da rede podem ser configurados para atividades de gerenciamento de acordo com as necessidades do momento, sem as restrições do longo processo de padronização já citado. As atualizações de regras para a tomada de decisões, de limites para desencadeamento de ações ou de um protocolo de gerenciamento, por exemplo, poderiam ser feitas através de simples atualizações dos códigos dos agentes em questão, ou do envio de novos agentes.

Além da facilidade de atualização de funcionalidades, os agentes podem ser providos de capacidades avançadas no que diz respeito ao processamento das informações de gerenciamen-

to. Os tempos de detecção de problemas e de restabelecimento do funcionamento normal do elemento gerenciado, em caso de falhas, podem então ser bastante reduzidos, uma vez que o agente está sendo executado localmente. Pela mesma razão, o consumo de banda passante para fins de gerenciamento pode ser significativamente reduzido. Apenas uma quantidade mínima de informações, como relatórios de ocorrências, estatísticas resumidas e alarmes precisa ser enviada para as estações de gerenciamento.

Na abordagem convencional de gerenciamento, como SNMP, certos aspectos na medida do desempenho de redes são difíceis de serem considerados, dada a influência causada pelo retardo gerado pela transmissão dos dados pela rede. Quando o polling é utilizado, a precisão de medidas temporais é questionável [3], pois o retardo da rede está presente no transporte de todas as informações que serão processadas na estação de gerenciamento. O envio de um agente móvel permite que análises locais sejam realizadas no elemento a ser gerenciado. Dessa forma, os retardos causados pela rede deixam de ser considerados [3], uma vez que o processamento é local. Granularidades mais finas para os valores dos parâmetros a serem observados e intervalos menores entre medições podem também ser utilizados.

Por outro lado, quando o número de elementos a serem gerenciados for pequeno, o consumo de banda e de tempo para a transferência dos agentes móveis certamente será significativo em relação aos seus equivalentes na utilização de uma abordagem convencional de gerenciamento. Algumas medições e uma comparação das abordagens de gerenciamento baseadas em SNMP e em redes ativas estão disponíveis em [10].

3. Arquitetura de Gerenciamento Distribuído usando Tecnologia Ativa (AGDA)

A AGDA [20] foi projetada com o objetivo de disponibilizar uma infra-estrutura que permita o gerenciamento distribuído de serviços, de nós ativos de redes de comunicação e estações de trabalho organizadas em domínios. Um domínio é composto por um ou mais segmentos de rede. Os instrumentos de gerenciamento são os agentes, que podem ser enviados ou deslocarem-se voluntariamente entre os elementos gerenciados da rede.

A AGDA, por utilizar-se de tecnologia ativa, apresenta as seguintes vantagens no gerenciamento em relação à abordagem centralizada do SNMP: (i) redução no tráfego de informações, (ii) redução dos retardos associados às ações de gerenciamento, e (iii) flexibilidade na atualização ou na introdução de novas funcionalidades. Os agentes de gerenciamento podem ser dotados de capacidades avançadas de processamento, podendo até mesmo desencadear ações corretivas, local e imediatamente. A Figura 1 apresenta uma visão geral da AGDA com os seus elementos.

A arquitetura é organizada, no mínimo, em três níveis hierárquicos. No primeiro nível (o mais alto) da hierarquia encontra-se o Gerente Mor. Este gerente é fixo em uma estação de trabalho e é responsável pelo gerenciamento de um conjunto de domínios. No segundo nível hierárquico encontram-se os Gerentes de Domínio (GD), sendo cada um responsável pelo gerenciamento de um único domínio. No terceiro nível hierárquico encontram-se os demais elementos gerenciadores, que são os Inspetores, os Especialistas e os Guardiões. Esses agentes são os responsáveis de fato pelo desencadeamento das ações referentes ao monitoramento, comparações, geração de alarmes, tentativas de correção e emissão de relatórios nas diferentes

áreas funcionais de gerenciamento - falha, desempenho, segurança, configuração e contabilização. Essas ações estão limitadas àquelas que demandem processamento simplificado, em função desses elementos estarem quase sempre instalados em equipamentos com reduzida disponibilidade de processamento. Além disso, os códigos desses elementos são implementados na forma de agentes, que não podem gerar tráfego excessivo para o seu próprio transporte. Os gerentes são responsáveis pela agregação de informações pelas emissões de relatórios em seus respectivos níveis. São eles que realizam os processamentos mais complexos que venham a ser necessários para o desencadeamento das ações de gerenciamento apropriadas e que estejam além da capacidade dos elementos gerenciadores do terceiro nível hierárquico. A distinção entre os processamentos simplificados de um Inspetor e complexos de um Gerente de Domínio será exemplificada na Seção 4.

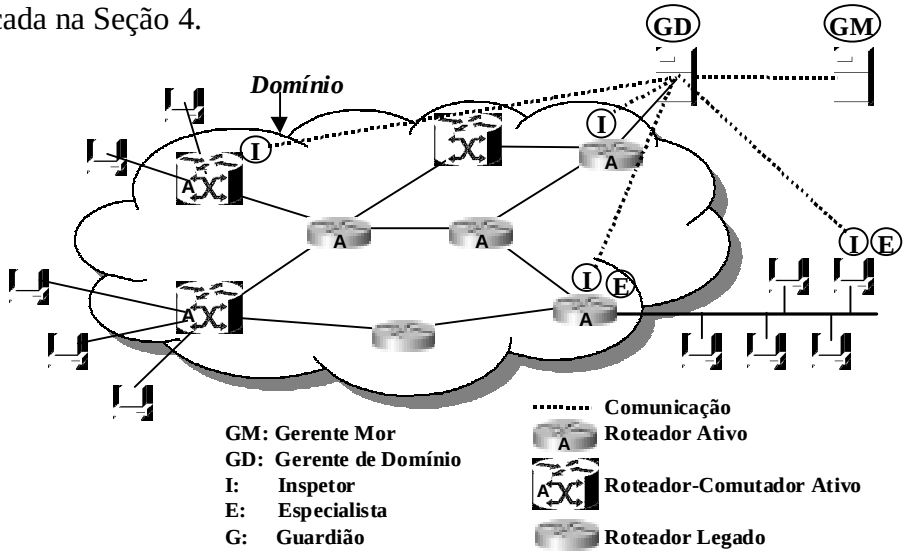

Figura 1 – Elementos da AGDA

A seguir serão descritas as funcionalidades de cada um dos elementos da arquitetura.

3.1. Gerente Mor

O Gerente Mor cria os Gerentes de Domínio e os envia para seus respectivos domínios, baseado na Base de dados de Topologia Geral (BTPG), a qual contém a topologia de todos os domínios subordinados a este Gerente Mor. Periodicamente, o Gerente Mor cria Guardiões e os envia para cumprir tarefas relativas ao monitoramento da integridade dos demais elementos da AGDA. Dependendo da quantidade de domínios gerenciados, pode haver mais de um Gerente Mor. Além da BTPG, o Gerente Mor utiliza-se de duas outras bases de dados: a Base Geral de Códigos (BGC) e a Base de Informações Gerenciais dos Domínios (BIGD). A BGC contém os códigos dos demais elementos da AGDA, enquanto a BIGD armazena informações consolidadas sobre o gerenciamento dos domínios recebidas de cada GD. Essas informações são utilizadas para a emissão de relatórios e também como subsídios para o planejamento de capacidade da rede.

3.2. Gerente de Domínio

O Gerente de Domínio é o responsável pela gerência de seu domínio, que inclui a emissão de relatórios e a tomada de decisões referente ao desencadeamento de ações, quando necessá-

rias, para a correção ou, pelo menos, a minimização dos efeitos de situações desfavoráveis que venham a ocorrer. O monitoramento é, de fato, realizado pelos Inspetores. As ações corretivas ou minimizadoras são desencadeadas por Especialistas. O GD recebe informações (previamente analisadas e filtradas) decorrentes do monitoramento por parte dos Inspetores e, quando necessário, envia Especialistas.

As informações referentes ao gerenciamento do domínio que o GD recebe dos Inspetores e Especialistas, após consolidação, são armazenadas em uma Base de Informações Locais do Domínio (BILD). Há também uma Base de Associação (BAS) que armazena associações entre tarefas (ações) de gerenciamento e Especialistas. A associação que ocorre na BAS pode ser criada manualmente ou então ser inferida com o auxílio de ferramentas de inteligência artificial.

O funcionamento do GD pode ser visualizado através do seu ciclo de vida:

1. Ativação, como processo, na estação determinada por um Gerente Mor. Periodicamente, envia uma mensagem de controle para o Gerente Mor sinalizando que está ativo.
2. Criação ou atualização de uma Base de Topologia do Domínio (BTPD), que armazena informações sobre a topologia de seu domínio.
3. Envio dos agentes Inspetores para os elementos a serem gerenciados.
4. Recebimento de informações e alarmes, quando for o caso, dos Inspetores.
5. Classificação, consolidação e armazenamento das informações recebidas na BILD.
6. Criação de Especialistas, se assim for determinado pela classificação, e envio dos mesmos para onde forem necessários.
7. Envio de informações de gerenciamento do domínio ao Gerente Mor quando necessário.
8. Emissão de relatórios sobre o gerenciamento do domínio quando necessário.

3.3. Inspetores

Os Inspetores são responsáveis pela coleta e análise dos dados referentes às funcionalidades que estão sendo gerenciadas no elemento em que atuam. A coleta é feita com o uso das facilidades disponíveis, desde o acesso direto às MIBs até o uso de agentes proxy. Agentes proxies têm por finalidade disponibilizar uma interface padrão com o Inspetor e, ao mesmo tempo, interagir com os elementos gerenciados via interfaces proprietárias. Os dados coletados sofrem análise local, restrita a um processamento simplificado, feito pelo próprio Inspetor, para que o elemento gerenciado em questão não seja sobrecarregado, mas também para que apenas um mínimo de informações sejam enviadas ao Gerente de Domínio. Esse processamento pode realizar desde uma filtragem das informações coletadas, não enviando informações repetidas, até uma fusão de informações relevantes diferentes em uma mesma mensagem para o GD. Além dessa operação assíncrona em relação ao GD, o Inspetor também pode operar de forma síncrona, atendendo a pedidos específicos daquele gerente.

O Inspetor é implementado com a organização que é apresentada, de forma esquemática,

na Figura 2. A sua implementação é configurável dinamicamente, de forma a permitir que as áreas funcionais de gerenciamento sejam tratadas independentemente, de acordo tanto com a natureza e capacidade de processamento disponível no elemento a ser gerenciado quanto com as intenções do administrador do domínio e disponibilidade de ferramentas de gerenciamento. Tal funcionalidade é viabilizada com o uso da tecnologia de desenvolvimento de software orientado a componentes [11]. As funções básicas tratam de tarefas referentes ao funcionamento do Inspetor em si e de tarefas comuns às áreas funcionais de gerenciamento (que são implementadas na forma de componentes), como por exemplo, comunicação com o Gerente de Domínio e interação com um Guardião. O bloco "comunicações" da figura refere-se às facilidades de comunicação de dados da infra-estrutura de mobilidade do elemento gerenciado que hospeda o Inspetor.

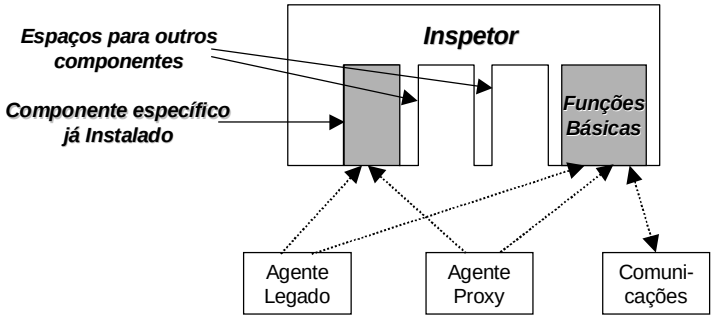

Figura 2 – Organização esquemática da implementação do Inspetor

3.4. Especialistas

As informações recebidas dos Inspetores podem levar o GD a concluir, após análise, que coletas de dados e novas análises, ou mesmo ações mais especializadas, sejam necessárias. Nesse caso, agentes móveis Especialistas são enviados ao elemento gerenciado em questão. Os Especialistas não são organizados de forma configurável dinamicamente como os Inspetores. Eles são programas autônomos, desenvolvidos apenas para exercer uma função bem definida. Exemplos de ações que podem ser realizadas por Especialistas são a alteração de rotas, a reconfiguração de utilização de memória, a comunicação com outras aplicações com a finalidade de desencadear adaptações, etc.

3.5. Guardiões

O Guardião tem como objetivo verificar se os demais elementos da AGDA estão funcionando corretamente e se os códigos dos mesmos estão íntegros, ou seja, se não foram alterados inadvertidamente ou falharam. Cada um dos demais elementos da AGDA implementa uma interface para interação com o Guardião, onde há o suporte para as funcionalidades como autenticação, autorização, controle de execução e verificação.

4. Gerenciamento de desempenho pró-ativo usando tecnologia ativa

O gerenciamento de desempenho visa atingir dois objetivos conflitantes, a garantia de QoS para os usuários (aplicações) e a obtenção de um alto grau de multiplexação no uso dos recursos disponíveis [1]. Ele deve prover funções para medir, monitorar, avaliar e gerar informações sobre os níveis de desempenho alcançados. Essas informações devem ser utilizadas

principalmente com duas finalidades. A primeira é a geração de estatísticas periódicas que venham a ajudar no planejamento de capacidade da rede, enquanto que a segunda é indicar a ocorrência de desrespeitos aos níveis de QoS desejados. Nesse último caso, controles dos recursos da rede devem ser acionados com o objetivo de melhorar a QoS que está sendo prestada aos usuários [17].

O gerenciamento de desempenho não deve se limitar apenas à rede. Os sistemas de computação ligados a ela também devem ter seu desempenho gerenciado com a finalidade de se identificar degradações que sejam neles originadas. Da mesma forma, os serviços de comunicação, que podem envolver diversos elementos da rede, devem também ter o seu desempenho monitorado.

A quase totalidade das abordagens de gerenciamento é reativa. Recentemente tem sido pesquisada a abordagem pró-ativa [15], [16], que tem os seguintes objetivos: (i) identificar problemas na rede, nos serviços e nas aplicações antes que alguma degradação grave ou falha de QoS nos serviços que estão sendo prestados venha a ocorrer e, (ii) desencadear ações com o objetivo de eliminar ou reduzir as degradações ou falhas ou minimizar suas conseqüências. Algumas áreas funcionais de gerenciamento, como configuração e contabilização, são inerentemente reativas. Já as áreas de gerenciamento de desempenho, de falhas e de segurança podem operar de forma mais eficiente caso o gerenciamento seja realizado de forma pró-ativa.

No gerenciamento pró-ativo, análises das estatísticas coletadas devem identificar sintomas indicadores de que um ou mais problemas estão por acontecer. Podem ser utilizadas nessas análises simulações rápidas, extrapolações ou técnicas de inteligência artificial, com o objetivo de se precisar ou estimar o momento quando determinado problema ou evento vai ocorrer [18]. As operações que têm que ser realizadas nessas análises podem vir a ser muito dinâmicas e exigir grande processamento. Esse tem sido o fator limitante do emprego de técnicas pró-ativas no gerenciamento de redes. A distribuição do processamento de gerenciamento, bem como a flexibilidade oferecida pelo uso de tecnologia ativa, no entanto, podem viabilizar a disponibilização do processamento necessário em cada elemento gerenciado.

A utilização de tecnologias ativas, associada à capacidade de processamento cada vez mais poderosa do hardware computacional presente tanto em estações quanto em nós de comutação, pode tornar possível a exploração do monitoramento das tendências de comportamento dos parâmetros de desempenho. Essas tendências podem ser utilizadas para que, mediante extrapolações, seja feita a previsão de quando, de fato, caso as tendências se mantenham, limites de QoS seriam desrespeitados. O tempo ainda restante pode ser utilizado para que ações sejam desencadeadas com o objetivo de se tentar reverter essas tendências. Caso a reversão não seja possível, adaptações [14] podem ser desencadeadas nas aplicações geradoras e consumidoras de tráfego para que a qualidade de apresentação (QoP – Quality of Presentation) [14] seja mantida mesmo com a queda dos níveis de QoS disponíveis.

4.1. Implementação do Gerenciamento de desempenho pró-ativo

Os parâmetros descritos a seguir foram considerados relevantes, no trabalho ora em curso, para o gerenciamento de desempenho. Esses parâmetros são utilizados para que sejam definidos os níveis de QoS. O *tempo de resposta* é relevante para uma aplicação. Esse é o tempo

decorrido entre a introdução de uma solicitação e a disponibilização de uma resposta. Estações e nós de comutação devem ter a *capacidade de processamento*, a *quantidade de memória* e o *tamanho das filas de saída* gerenciados. Para um serviço de comunicação, são relevantes o *retardo*, a *variação do retardo*, a *taxa de erros*, a *taxa de perdas* e a *vazão*. Em um enlace de comunicação são importantes a *utilização*, a *taxa de erros* e a *taxa de perdas*.

A monitoração desses parâmetros é realizada pelo componente de software [11] de gerenciamento de desempenho pró-ativo instalado no Inspetor associado ao elemento (aplicação, serviço, nó de comutação, etc) a ser gerenciado. O componente de programa de Inspetor que trata do gerenciamento de desempenho será tratado, deste ponto em diante, como Inspetor de Desempenho (ID), muito embora, de fato, seja implementado na forma de um componente do verdadeiro Inspetor, de acordo com o que é previsto pela AGDA. Durante a monitoração são coletados dados das fontes disponíveis, como MIBs ou informações obtidas de sessões RTP-RTCP no caso de um serviço de comunicações, do sistema operacional da estação ou do nó de comutação, ou das próprias aplicações. A partir desses dados coletados são calculadas, no próprio Inspetor, as variações dos valores dos parâmetros que devem ser monitorados para que sejam detectadas as tendências dessas variações. Esse cálculo é simples, pois trata apenas da realização de operações aritméticas elementares, podendo portanto, ser realizado no próprio elemento gerenciado. O retardo absoluto de um serviço de comunicação pode ser utilizado como exemplo. Com duas observações consecutivas do valor do retardo obtém-se a variação do mesmo por unidade de tempo. Uma seqüência de variações fornece a tendência da variação, calculada como a média das variações.

O Inspetor de Desempenho tem em sua configuração a duração do intervalo a ser considerado no cálculo das variações e o número de variações que devem ser consideradas no cálculo de uma tendência, bem como do limite dessa tendência. Esse limite, quando atingido, leva ao envio, pelo Inspetor, de um alarme de tendência para o GD.

As informações que precisam ser armazenadas durante a monitoração devem ser mantidas em memória de forma a reduzir o tempo de acesso às mesmas. O tamanho dessas informações não deve ser grande, visando não consumir quantidade expressiva de memória. Apenas informações consolidadas devem ser mantidas armazenadas e somente pelo tempo que forem necessárias ou até que sejam transmitidas para o Gerente de Domínio.

O Inspetor de Desempenho pode também se comunicar com aplicações que desejarem receber os alarmes de tendência. A comunicação no Inspetor é então realizada através de duas interfaces: (i) entre o Inspetor de Desempenho e o Inspetor propriamente dito e (ii) entre o Inspetor de Desempenho e as aplicações, conforme pode ser observado na Figura 3. O Inspetor, por sua vez, utiliza-se da infra-estrutura de comunicação da AGDA, para comunicar-se com o GD. Essa comunicação produz o menor tráfego possível na rede, sendo desencadeada apenas quando necessária e consistindo apenas no envio de códigos e valores de parâmetros.

No GD, o alarme de tendência, que transporta consigo as variações que o desencadearam, é analisado para que, mediante extrapolação, seja calculado o instante de tempo quando, caso essa tendência se mantenha, o limite máximo do valor, no caso do retardo, será desrespeitado. O cálculo das extrapolações exige processamento mais complexo do que aquele realizado

para o cálculo de tendências, por isso é feito no Gerente de Domínio. A extrapolação pode envolver desde simples ajustes até técnicas de IA, bem como a consulta a diversas bases de dados como, por exemplo, o baseline da rede.

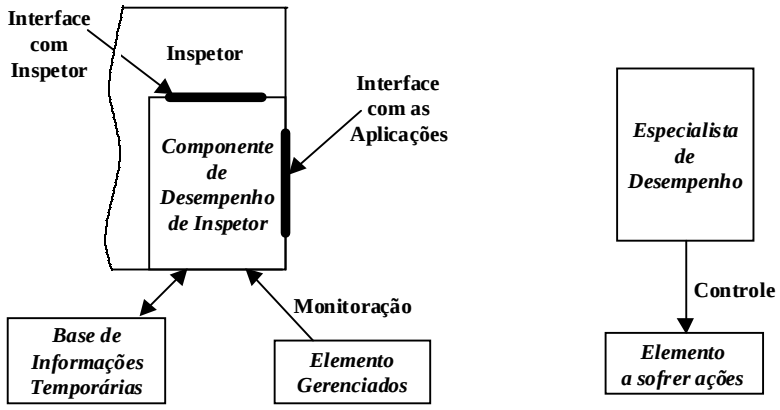

Figura 3 – Interfaces e relacionamentos do Inspetor e do Especialista de desempenho

O GD vai determinar então, após essa extrapolação, o intervalo de tempo disponível para que, se necessária, alguma ação seja desencadeada. Se esse for o caso, será obtido, mediante consulta à BAS, o Especialista de Desempenho que deve ser enviado, enquanto o próprio GD vai determinar para qual elemento gerenciado o mesmo deve ser enviado.

As seguintes ações podem ser realizadas, no trabalho que está sendo desenvolvido, de acordo com o tipo de elemento gerenciado: (i) nas aplicações, adaptações podem ser realizadas para que a QoP não seja comprometida com a queda do nível de QoS que está para acontecer; (ii) nas estações, buffers podem ser redimensionados e prioridades e políticas de escalonamento podem ser ateradas, (iii) nos nós de comutação rotas podem ser alteradas.

5. Ambiente de testes, protótipo implementado e análise de resultados

A implementação do protótipo da arquitetura proposta está sendo realizada com o uso da infra-estrutura de mobilidade µCode [12]. µCode é uma ferramenta de desenvolvimento de agentes móveis baseada em Java, projetada para prover mobilidade em aplicações distribuídas de forma extensível, flexível e pouco consumidora de processamento. µCode utiliza µServers como ambientes de execução, que devem estar instalados nos nós ativos. Os µServers executam os agentes, que são threads móveis escritas em Java, representando as aplicações ativas. A operação do µCode é baseada na criação, cópia e migração dessas threads. Apenas a mobilidade fraca [19] é disponibilizada, uma vez que as threads, ao migrarem, salvam seu estado em relação aos dados, mas não o seu estado de execução. É oferecido o suporte tanto ao envio quanto à recuperação de códigos, sejam fragmentos ou completos, com invocação síncrona ou assíncrona e com execução imediata ou posterior [19].

O ambiente de testes constitui-se de quatro computadores pessoais baseados no processador Pentium de 200 MHz com sistemas operacionais Red Hat 6.2 e Conectiva 6.0 e 64 Mb de memória cada, interligados por um único segmento de rede de tecnologia Ethernet de 100 Mbps.

5.1. Protótipo implementado

Foram implementadas as seguintes classes: (i) Gerente: representa um Gerente de Domínio, (ii) Inspetor, (iii) Especialista1 a EspecialistaN: representam os diferentes tipos de Especialistas disponíveis, (iv) Nó: representa os nós da rede. Ao ser invocada, essa classe recebe como parâmetros o endereço IP e a porta da máquina onde será instanciado. Cada nó está diretamente associado a um μServer que foi previamente carregado. A comunicação entre as classes no μCode é realizada através de RMI.

Dada a preocupação com o desempenho que é inerente à execução de aplicações programadas em Java, agravada pela presença de um ambiente intermediário em cada nó, o μServer, o protótipo que foi implementado e que está sendo discutido neste artigo tinha como finalidade verificar a viabilidade do emprego da infra-estrutura AGDA para o gerenciamento pró-ativo.

Foram obtidas vinte observações do intervalo de tempo decorrente desde a tomada de decisão, por um Inspetor em um elemento gerenciado (no caso, uma estação), de enviar um alarme de tendência até o final do carregamento de um Especialista nesse mesmo elemento (no caso do protótipo) gerenciado. Os processamentos incluídos nesse intervalo de tempo são os seguintes: (i) decisão, pelo Inspetor, de enviar um alarme, (ii) Inspetor: preparo e envio da mensagem que transporta o Alarme de Tendência, (iii) μServer do Inspetor: envio da mensagem, via RMI, para o GD, (iv) segmento de rede: transporte da mensagem, (v) μServer do GD: recepção da mensagem via RMI e entrega da mesma ao GD, (vi) GD: recepção, interpretação da mensagem e escolha, mediante consulta à BAS, do Especialista mais apropriado, (vii) GD: recuperação da BGC, preparo e envio do código do Especialista ao elemento gerenciado adequado, (viii) μServer do GD: envio do código ao elemento gerenciado, (ix) segmento de rede: transporte do Especialista, (x) μServer do elemento gerenciado: recepção e carregamento (instanciação) do código do Especilista que foi recebido, (xi) início da execução do Especialista.

O envio do código ao elemento gerenciado pode exigir até duas comunicações diferentes. A primeira comunicação não envia o código do Especialista e sim a URL de onde tal código deve ser recuperado, bem como uma identificação desse código. O μServer do elemento gerenciado, ao receber essa mensagem, verifica se o código em questão já está carregado em memória. Caso não esteja, uma segunda comunicação é desencadeada para que o código seja então recuperado da estação onde é executado o Inspetor.

5.2. Análise dos resultados

O intervalo de tempo considerado nas medições é relevante para a implementação do gerenciamento de desempenho pró-ativo, pois determina o limite de tempo mínimo a partir do qual ações podem, de fato, serem desencadeadas com o objetivo de reverter uma tendência de queda dos níveis de QoS.

Os resultados obtidos nas vinte medições que foram realizadas estão apresentados na Tabela 1.

O valor médio foi de 872 ms e o desvio padrão de 50,62 ms.

O processamento realizado pelo Gerente de Domínio, mesmo tendo sido feitos dois acessos ao sistema de arquivos (um para a escolha do Especialista e outro para a recuperação do código do mesmo) da plataforma na qual estava sendo executado, foi simplificado, pois não foram executadas extrapolações. Quando um processamento mais complexo for necessário, o intervalo de tempo em questão será maior. Esse gerente, no entanto, é executado em uma estação de trabalho com grande capacidade de processamento, fato que, dependendo da complexidade necessária à execução das tarefas mencionadas na etapa (v) limitará o aumento do intervalo de tempo analisado.

Tabela 1 – Resultados obtidos

Valores em milessegundos				
857	889	848	892	905
884	860	859	765	854
853	904	907	927	852
956	894	886	736	912

O carregamento do Especialista é responsável por parcela significativa desse intervalo de tempo. Chegou-se a essa conclusão quando foi utilizado um loop para enviar 20 vezes o mesmo especialista a um mesmo nó. O intervalo de tempo para a primeira execução de toda a experiência era sempre cerca de dez vezes superior àqueles para as demais execuções. Tal fato se deve ao não carregamento do Especialista em memória nas demais execuções, uma vez que μServer (via Class Loader da JVM) já percebia a presença do Especialista em memória. As medições referentes aos carregamentos subseqüentes não primeiro carregamento ao foram consideradas, uma vez que é pouco provável que seja necessário o envio do mesmo Especialista repetidas vezes em um pequeno intervalo de tempo. Visando a economia de tempo referente ao carregamento de novos objetos em memória, o código do Especialista, além de ser o mais compacto possível, deve incluir o menor número de novas instanciações de outras classes.

Os resultados obtidos permitem concluir que, mesmo com a presença do ambiente μCode e da máquina virtual Java, é viável a utilização de tecnologia ativa para a implementação de gerenciamento de desempenho pró-ativo, pois o tempo necessário para um Especialista estar em condições de desencadear uma ação é menor do que um segundo.

6. Trabalhos relacionados

A distribuição do processamento, transferindo processamentos do servidor de gerenciamento, como nas abordagens CMIP e SNMP, para próximo aos dados necessários ao gerenciamento, ou seja, para os próprios elementos a serem gerenciados têm sido pesquisada desde o surgimento de abordagens padronizadas de gerenciamento. O trabalho de Yemini [21] destacou o conceito pela primeira vez. A primeira geração de sistemas de gerenciamento distribuído utilizava basicamente o paradigma cliente-servidor. A segunda geração usa RPC ou a distribuição estáticas de objetos, como CORBA e Java RMI. para a distribuição das tarefas. As pesquisas atuais, já se configurando uma terceira geração, concentram-se no emprego de tecnologia ativa para a implementação da distribuição no gerenciamento [10].

A maioria dos trabalhos consultados usam tecnologia ativa, porém, não implementam o ge-

renciamento pró-ativo. O trabalho consultado que emprega uma abordagem pró-ativa não se utiliza de tecnologia ativa.

Smart Packets [13] utiliza o paradigma de redes ativas. Um protocolo específico é utilizado, o Active Network Protocol (ANEP), para a transferência do programas que não podem exceder 1 KB. A proteção ao nó ativo é forte, tendo sido implementada uma linguagem de alto nível, Sprocket, que não dispõe de ponteiros, não realiza o acesso a arquivos e nem faz gerenciamento de memória, que são ações potencialmente problemáticas. Um programa em Sprocket é compilado em outra linguagem, Spanner, com o objetivo de minimizar o tamanho do programa a ser transmitido na rede. Um ambiente de execução capaz de executar programas Spanner está presente em cada nó ativo. Um esquema de segurança realiza a autenticação dos programas a serem executados nos nós ativos. O paradigma de mobilidade não é utilizado.

A arquitetura Active Distributed Management (ADM) [7] é organizada em três camadas: gerenciamento de processos, básica de operação e ferramentas de gerenciamento. A primeira é o ambiente de execução, provendo funcionalidades de redes ativas e de mobilidade, disponibilizando funções para criação, remoção, interrupção, continuação, duplicação, movimentação, comunicação, serviços de diretório e de segurança. A camada de básica de operação disponibiliza padrões de navegação para os fragmentos de código ativo (da terceira camada) que de fato realizam tarefas de gerenciamento. Na terceira camada são implementadas as funções relativas ao gerenciamento propriamente dito, na forma das aplicações ativas. A linguagem utilizada é Java.

O projeto MIAMI [1] implementa o gerenciamento de desempenho com o uso de três agentes, dois estáticos e um móvel. O primeiro, estático, realiza a interface entre o gerenciamento de desempenho e o sistema de gerenciamento. Este agente, quando ordenado, cria um agente móvel e o envia a um elemento de rede para que realize funções de monitoramento e de consolidação localmente. Um terceiro agente, estático, serve como agente proxy, realizando a interface entre o agente móvel e o elemento a ser gerenciado. O agente móvel envia informações consolidadas ao primeiro agente estático periodicamente ou assincronamente quando limites configurados para parâmetros de desempenho forem ultrapassados. Todo o processamento referente ao gerenciamento é local. A linguagem utilizada também é Java.

Francheschi [16] investigou o gerenciamento pró-ativo utilizando técnicas de inteligência artificial para a geração de alarmes. Um módulo (equivalente a um agente de gerenciamento SNMP) pró-ativo é carregado no elemento gerenciado contendo um serviço de verificação que se utiliza de IA. Este serviço obtém valores dos parâmetros gerenciados via RMON, compara-os com dados do baseline do elemento gerenciado e, após verificá-los, se for o caso, emite alarmes de tendência

7. Conclusões

A demanda por níveis diferenciados de QoS, bem como a constante necessidade de evolução nas redes de comunicação torna necessária a existência de um sistema de gerenciamento de desempenho flexível e pró-ativo. Visando atender a esses requisitos, este artigo apresentou

uma arquitetura que emprega a tecnologia ativa e realiza a monitoração das variações dos valores dos parâmetros referentes aos níveis de QoS negociados. Os agentes móveis viabilizam a necessária flexibilidade para a configuração, atualização e alteração do gerenciamento de desempenho. Já a monitoração das variações permite a detecção da tendência de comportamento dos níveis de QoS, visando permitir que, no caso dessa tendência se mostrar desfavorável, ações sejam desencadeadas para a reversão da mesma ou, pelo menos, a minimização das suas conseqüências, caso a tendência se mantenha.

Cabe destacar que o trabalho integra duas áreas de pesquisa recentes e ainda pouco relacionadas: o gerenciamento de redes baseado em tecnologia ativa [16] e o gerenciamento de desempenho pró-ativo [7].

Os resultados obtidos indicam que o tempo referente ao envio de um alarme de tendência ao Gerente de Domínio, ao processamento nesse gerente para escolha, preparo e envio de um Especialista, bem como o seu carregamento via RMI e μCode não requer nem um segundo, tempo que pode viabilizar a implementação do gerenciamento de desempenho pró-ativo tendo como base uma arquitetura de gerenciamento distribuída e usando tecnologia ativa, a AGDA.

Pode-se citar como limitações da arquitetura que está sendo proposta a falta de mecanismos de segurança e de proteção aos nós ativos e demais elementos a serem gereciados, como, por exemplo, as aplicações e serviços de comunicação.

Os trabalhos futuros de curto prazo incluem novas execuções do procedimento descrito neste artigo em um sistema de comunicação com vários segmentos de rede sujeitos a diversos tipos de tráfegos. Outras aplicações serão executadas de forma concorrente, tanto na estação do GD quanto no nó onde será executado o Especialista. Será também realizada a implementação dos processamentos do Gerente de Domínio e do Especialista referentes ao gerenciamento de desempenho, sempre na abordagem pró-ativa. O gerenciamento de serviços de comunicação será também implementado. Em médio prazo, será feita a integração da arquitetura descrita neste artigo com o projeto ServiMídia [14], tendo como objetivo, implementar o gerenciamento de desempenho pró-ativo daquele sistema multimídia distribuído. Em longo prazo serão criados mecanismos de segurança e de proteção.

8. Referências

[1] Bohoris, C., Pavlou, G., Cruickshank, H., *Using Mobile Agentes fos Network Performance Management*, IEEE/IFIP Network Operations and Management Symposium (NOMS´00), Honolulu, Hawaii, 2000.

[2] Rubinstein, M.G., Duarte, O.C.M. e Pujolle, G., *Evaluating the Network Performance Management Based on Mobile Agents*, .

[3] Bieszczad, A., Pagurek, B. e White, T., *Mobile Agents for Network Management*, IEEE Communication Surveys, Fourth Quarter, 1998.

[4] Psounis, K., *Active Networks: Applications, Security, Safety and Architectures*, IEEE Communications Surveys, Abril de 1999.

[5] Cunha, E.C., *Uma Estratégia de Criação e Apresentação de Documentos Multimídia Adaptativos em Rede*, Dissertação de Mestrado, NCE/IM/UFRJ, 2000.

[6] Tennenhouse, D.L. et al, *A Survey of Active Network Research*, IEEE Communications Magazine, janeiro de 1997.

[7] Kawamura, R. e Stadler, R., *Active Distributed Management for IP Networks*, IEEE Communications Magazine, Abril de 2000.

[8] Bradshaw, J. et al, *Software Agents*, AAAI Press / The MIT Press, Menlo Park, California, 1997.

[9] Hu, C., Chen, W., *A Mobile Agent-based Active network Architecture*, International Conference on Parallel and Distributed Systemas, ICPADS´00, IEEE, 2000.

[10] Rubinstein, M.G., Duarte, O.C.M. e Pujolle, G., *Evaluating the Network Performance Management Based on Mobile Agents*, 2000.

[11] Brown, A. e Wallnau, K.C., *The Current State of CBSE*, IEEE Software Magazine, outubro de 1998.

[12] Picco, G.P., *μCode: A Lightweight and Flexible Mobile Code Toolkit*, Proceedings of the 2nd International Workshop on Mobile Agents 98 (MA'98), Stuttgart (Germany), K. September 1998, Springer, Lecture Notes on Computer Science vol. 1477, pp. 160-171, 1998.

[13] Schwartz, B., Jackson, A.W., Strayer, T., Zhou, W., Rockwell, D. e Partridge, C., *Smart Packets: Applying Active Networks to Network Management*, ACM Transactions on Computer Systems, Fevereiro de 2000, páginas 67-68.

[14] Gomes, R.L., *Autoria e Apresentação de Documentos Multimídia Adaptativos em Redes*, Dissertação de Mestrado, NCE/IM/UFRJ, 2001.

[15] Data Communications, *Proactive LAN Management*, Data Communications Magazine, março de 1993.

[16] Franceschi, A.S.M., Rocha, M.A., Weber, H.L. e Westphall, C.B., *Proactive Network Management Using Remote Monitoring and Artificial Intelligence Techniques*, Proceedings of the 2nd IEEE Symposium and Communications (ISCC´97), junho de 1997.

[17] BRISA (Sociedade Brasileira para Interconexão de Sistemas Abertos), *Arquiteturas de* Redes de Computadores OSI e TCP/IP, Makron Books, 1994.

[18] Keshav, S. e Sharma, R., *Achieving Quality of Service through Network Performance Management*, Proceedings of Network and Operating System Support for Digital Audio and Video, Cambridge, 1998.

[19] Fuggeta, A., Picco, G.P. e Risso, F., *Understanding Code Mobility*, IEEE Transaction on Software Engineering, 24(5):146-155, Abril de 1998.

[20] Dumont, A. P. M, *AGDA: Uma Arquitetura de Gerenciamento Ativo Distribuído*, Dissertação de Mestrado, NCE/IM/UFRJ, março de 2002.

[21] Goldszmidt, G. e Yemini, Y. *Distributed Management by Delegation*, Proceedings of the 15th International Conference on Distributed Computing Systems, junho de 1995