Plataforma Para a Construção de *Grids* de Agentes Para o Gerenciamento de Sistemas

Marcos Dias de Assunção, Fernando Luiz Koch e Carlos Becker Westphall

***Abstract*—Este trabalho apresenta aspectos de implementação de uma arquitetura de *grids* de agentes aplicável ao gerenciamento de redes e sistemas e alguns experimentos de distribuição das atividades de análise usando esta arquitetura. O gerenciamento centralizado de redes de computadores e de sistemas pode nos levar à situações em que grandes volumes de dados precisam ser manipulados e analisados por uma única estação de gerenciamento na rede. Além do alto custo, tal abordagem tem se apresentado pouco extensível à medida que cresce o número de dispositivos no ambiente gerenciado. A arquitetura proposta e que vem sendo estudada tem como objetivo básico aplicar princípios da computação em *grid*, usando agentes de software como infra-estrutura para construção do *grid*, com o objetivo de otimizar alguns problemas inerentes ao gerenciamento centralizado como a manipulação e análise de grande volume de informação.**

***Index Terms*-- *grids* de agentes, computação em *grid*, gerência de redes, gerência de sistemas.**

I. Introdução

Alguns problemas computacionais requerem, para serem resolvidos, recursos de processamento, armazenamento e comunicação que não podem ser reunidos em um ambiente computacional sob uma mesma administração por questões econômicas ou técnicas. Estes problemas estão freqüentemente presentes em áreas científicas e comerciais e são comumente chamados de Aplicações de Grande Desafio [1]. Na gerência de grandes redes encontramos uma situação semelhante onde existe um enorme número de elementos e sistemas a serem gerenciados. A análise das informações de gerenciamento, em ambientes como este, pode culminar em uma situação onde não existe, em um único servidor, uma capacidade computacional para realizar um gerenciamento eficiente.

A solução destes problemas necessita da utilização de hardware de alto desempenho disponibilizado por supercomputadores ou clusters de computadores. Com a evolução de uma série de tecnologias, uma solução alternativa tem sido a utilização de recursos distribuídos. Esta agregação e compartilhamento de recursos conectados em rede tem sido chamada de computação em *grid*, ou *grid computing* [2].

As tecnologias *grid* têm sido vistas como uma forma de possibilitar uma integração de recursos e sistemas de uma forma mais flexível e rápida formando o que se pode chamar de organizações virtuais dinâmicas [3]. A caracterização das tecnologias *grid* como forma de possibilitar o compartilhamento coordenado e controlado de recursos evidencia a necessidade de utilização de uma arquitetura baseada em serviços [4].

Neste cenário, onde existem inúmeros recursos e serviços envolvidos, o uso de computação baseada em agentes é recomendável [5], pois os agentes podem ser usados como uma forma de implementação deste paradigma de orientação por serviços. Neste caso, os agentes podem definir as suas necessidades de recursos e serviços, procurar por outros agentes que oferecem os serviços e recursos requeridos e negociar com estes a utilização dos mesmos.

O propósito de se utilizar *grids* de agentes na gerência de redes de computadores é usar uma arquitetura de *grid*, baseada em princípios de tecnologias de agentes de software, para distribuir e otimizar as tarefas de gerenciamento, como coleta e análise de informações.

A proposta de uma abordagem de gerenciamento baseada em *grids* de agentes foi apresentada em trabalhos anteriores [6],[7],[8]. A implementação e experimentação de toda a plataforma de gerenciamento representam um trabalho complexo e por isso temos trabalhado na implementação de alguns elementos desta arquitetura. Neste trabalho são apresentados detalhes de implementação e funcionamento da arquitetura de *grids* de agentes que vem sendo desenvolvida.

O restante deste artigo está organizado da seguinte maneira: na segunda seção é apresentada a arquitetura usada no *grid* de agentes de gerência; a aplicação da arquitetura é comentada na seção 3; na seção 4 são mostrados detalhes da implementação e alguns resultados alcançados; e por fim, na seção 5 estão as conclusões e trabalhos futuros a serem realizados.

II. A Arquitetura de *Grids* de Agentes

Durante o desenvolvimento do trabalho, uma primeira proposta de arquitetura para o *grid* de agentes foi sugerida em [6]. Esta arquitetura está baseada no fluxo tradicional do gerenciamento de redes de computadores apresentado por Koch & Westphall [9]. Levando em consideração este fluxo

Marcos Dias de Assunção e Carlos Becker Westphall, Universidade Federal de Santa Catarina, Laboratório de Redes e Gerência, Campus Universitário Trindade, Florianópolis-SC, Brasil.
(Email: assuncao@lrg.ufsc.br, westphal@lrg.ufsc.br).

Fernando Luiz Koch, Intelligent Systems Group, Institute of Information and Computing Sciences, Utrecht University, The Netherlands.
(Email: fkoch@acm.org).

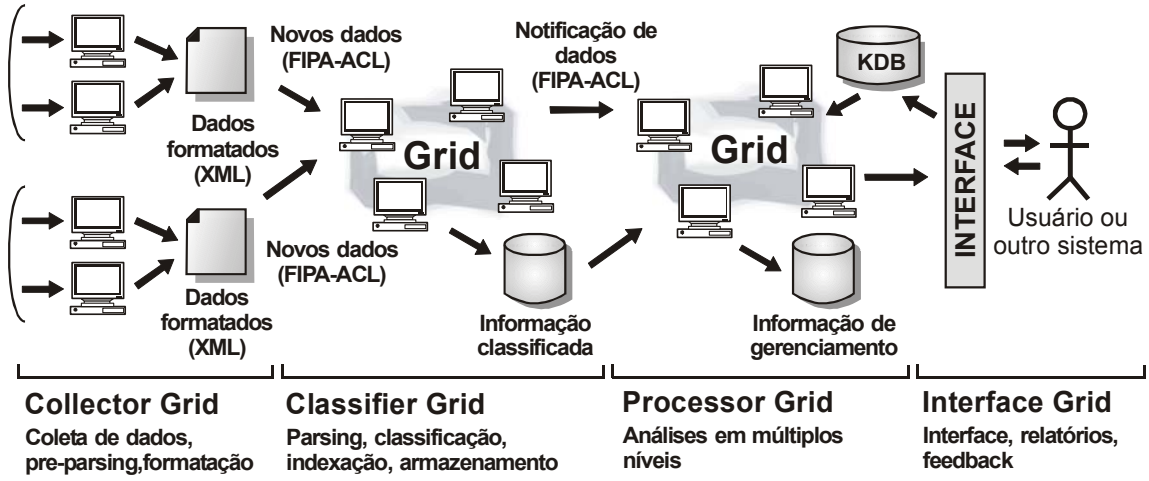

Figura 1. Visão geral da arquitetura de *grid.*

tradicional de gerenciamento a arquitetura foi dividida em componentes que descrevem cada uma das etapas do gerenciamento: a coleta de dados dos dispositivos gerenciados da rede; o tratamento e armazenamento destes dados coletados; a análise destas informações e; a apresentação ao gerente humano ou a outro sistema de gerência através de interfaces. Uma visão abstrata do sistema de gerência idealizado pode ser visto na **Figura 1**, e os componentes desta arquitetura são descritos a seguir:

***Grid* Coletor**: os agentes neste *grid* inicial fazem a interface com os dispositivos de rede através de algum protocolo de gerenciamento ou outra interface, extraindo os dados e enviando-os para o próximo passo; a informação extraída dos elementos de rede é formatada em uma representação XML garantindo a uniformidade para os demais elementos do *grid*;

***Grid* Classificador**: neste passo os dados recebidos são classificados e armazenados, eles são organizados e agrupados de uma forma que o desempenho na recuperação destes dados possa ser maximizado; esta é uma tarefa complexa onde os dados relevantes precisam ser extraídos, classificados e indexados;

***Grid* de Análise ou Processamento**: neste ponto os dados classificados são processados usando as regras de gerenciamento armazenadas na base de conhecimento do sistema; em um grande sistema existirão grandes volumes de dados para serem processados usando centenas ou milhares de regras de análise;

Elemento de Interface: nesta fase final, os dados de gerenciamento são apresentados para o mundo externo através de relatórios, alertas ou modelos abstratos de dados que possibilitam a integração com outros sistemas; idealmente, nós podemos explorar formatos ubíquos que possuem aceitação popular, desde alertas de e-mail até interfaces em XML.

III. Aplicação da Arquitetura

Um cenário onde os *grids* de agentes podem ser aplicados consiste de uma situação onde uma empresa qualquer desenvolve, instala e gerencia sistemas de telecomunicações para empresas situadas em diferentes sites. Os dados de desempenho coletados dos sistemas operacionais e dos sistemas fornecidos pela empresa são enviados a um centro de gerência através de protocolos como HTTP ou SMTP. O centro de gerenciamento é responsável por analisar estas informações.

Em uma situação de gerenciamento onde um enorme volume de dados é coletado da rede gerenciada e onde é necessário um grande poder de processamento para a realização das análises de dados é interessante poder utilizar os próprios recursos dos sites gerenciados para realizar a análise das informações.

Nesta situação, os recursos da rede gerenciada funcionam como colaboradores do próprio *grid* nas atividades de gerenciamento. Esta abordagem é interessante porque os próprios recursos da rede são usados para que a mesma seja gerenciada de maneira mais eficiente e onde um grande conjunto de dados pode ser analisado.

Esta visão é apresentada de maneira abstrata na **Figura 2**: os dados são coletados da rede gerenciada pelos agentes coletores de dados (a); os dados a serem analisados são repartidos e dão origem às tarefas de análise (b); as tarefas são atribuídas aos recursos da rede gerenciada para serem processados (c); o resultado deste processamento dá origem aos relatórios de gerenciamento (d) que são apresentados a uma interface para o gerente humano (e).

IV. Implementação e Resultados Experimentais

Com o objetivo de validar os conceitos de *grids* de agentes na gerência de redes e de sistemas, um protótipo e exemplos demonstrando o funcionamento de alguns componentes do sistema de gerenciamento foram desenvolvidos. Sendo assim, foi escolhida uma plataforma de agentes, a qual foi adaptada às necessidades de desenvolvimento deste trabalho e sobre ela

foram implementados os serviços e funcionalidades adicionais.

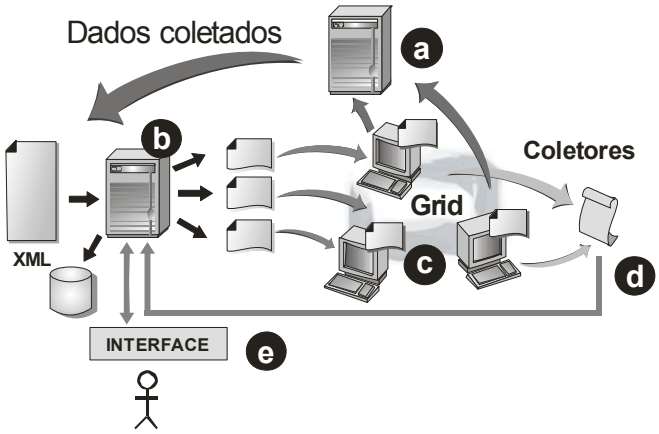

Figura 2. Grid formado pela própria rede gerenciada.

A. A Plataforma Utilizada

Para o desenvolvimento da arquitetura de *grid* proposta foi escolhido o *framework* para construção de agentes [10]. O *framework* funciona com o conceito de *container* de agentes. Um *container* pode ser visto como uma aplicação que possui vários agentes em execução. Traduzindo isto para Java, podemos dizer que os agentes são *Threads* da aplicação *container*.

Os agentes no *framework* são compostos basicamente por:

- Base de conhecimento onde são armazenados os fatos e regras que constituem o conhecimento do agente.
- Uma máquina de inferência que é usada para inferir sobre as regras armazenadas na base de conhecimento.
- Os objetivos que o agente deve perseguir durante o seu ciclo de vida.

Alguns dos elementos que compõem um container são:

- Um diretório onde são armazenadas informações inerentes aos agentes do container, rotas para outros *containers*, informações sobre as linguagens de conteúdo de mensagens que são aceitas pelos outros *containers*, codificação de mensagem aceita e protocolos utilizados para transporte das mensagens.
- Uma base de conhecimento global que é compartilhada por todos os agentes do container.
- Um agente gerenciador que é o agente responsável por processar as mensagens destinadas ao container.

Outro módulo importante e utilizado no conceito de *grid* de agentes é o de comunicação, ou de rede. Quando um *container* é criado, é fornecido o seu *Communicator*. O *Communicator* define as primitivas de comunicação utilizadas pelo *container* para que este possa ser ligado a outros *containers* localizados em estações diferentes.

Em alguns casos é necessário que o agente execute código Java que não pode ser representado de forma lógica. Atividades como as funções de gerenciamento a serem executadas para coletar estas informações devem ser representadas de forma lógica.

A ligação entre o conhecimento dos agentes e entre estas atividades, como o uso de classes e métodos de bibliotecas, deve ser feita através da implementação de uma interface chamada *SkillInterface*.

Neste caso, quando alguma regra procurada na base de conhecimento fecha com uma regra que representam uma *Skill*, os termos serão passados para a *SNMPSkill* que executará a ação correspondente. No exemplo da **Figura 3** temos uma amostra de como este mapeamento funciona para o caso de uma *Skill* SNMP.

```
snmp(Cmd,A1,A2,A3) :-
   !br.ufsc.lrg.agentgrid.skill.SNMPSkill(Cmd,A1,A2,A3).
[...]
snmpcollect(Address,Community,OIDList,Result) :-
   snmp(connection,Connection,Address,Community),
   snmp(get,Connection,OIDList,Result).
```

Figura 3. Exemplo de mapeamento de regras.

B. Início de Um Nó Grid

O *grid* de agentes é formado por vários *containers* de agentes interligados usando protocolos como HTTP ou SMTP. Quando um nó *grid* é iniciado ele pode se ligar a um nó já existente ou ser a raiz para um determinado *grid*. Para realizar esta ligação o nó que está sendo iniciado envia uma mensagem FIPA para o nó ao qual ele está se ligando e este atualiza a sua tabela de rotas. O nó iniciado também atualiza a sua tabela de rotas, adicionando o nó ao qual ele se ligou como uma rota padrão. Esta idéia é ilustrada na **Figura 4** onde temos um exemplo de um nó que foi iniciado e se ligou a um nó já existente e que já possuía um nó ligado a ele: o nó C solicita que A registre sua presença (1). A confirma a mensagem de C (2) e ambos atualizam suas tabelas de rotas (3).

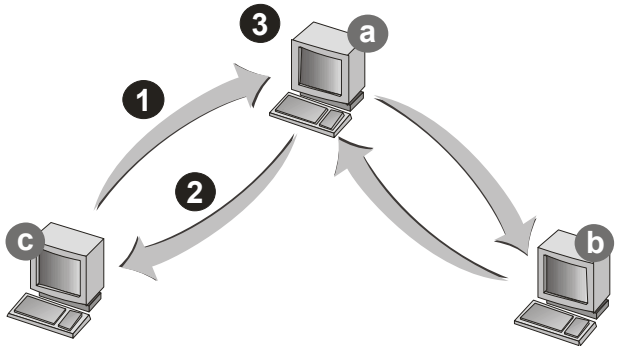

Figura 4. Nó *grid* sendo iniciado e se ligando a outro.

C. Arquivos XML de Definição

Os arquivos de definição carregados por um nó *grid* no momento em que ele é iniciado são documentos XML. Estes documentos contêm a lógica da aplicação para a qual o nó será utilizado. Por exemplo, se um nó *grid* desempenhará o papel de um coletor de dados, ele deve ser provido com conhecimento e habilidades necessárias para que ele possa realizar estas atividades. Se o nó for usado para realizar atividades de análise de informações, ele deverá ter o

conhecimento inicial necessário para realizar as atividades para as quais ele se destina. Um exemplo bastante simples de um arquivo de definição usado para criar um agente coletor é apresentado na **Figura 5**.

```
<?xml version="1.0"?>
<grid>
 <DEBUG action="echo" name="snmp-test"/>
 <agent name="collector-1">
  <kdb>
   equipment('127.0.0.1','public').
   netConfig :-
    equipment(Address,Community),
    snmpConnection(Address,Community,Connection),
    monitor('interf','config',Conn,TreeResult),!.
  </kdb>
  <schedule name="collectNet" time="60">
     netConfig().
  </schedule>
 </agent>
</grid>
```

Figura 5. Exemplo simples de arquivo de definição.

D. A Análise dos Dados

Uma das vantagens a serem evidenciadas da utilização do *grid* de agentes no gerenciamento é a possibilidade de distribuir a carga de processamento das atividades de gerenciamento diminuindo o tempo gasto nas atividades de análise e utilizando recursos ociosos. Os critérios para distribuição das atividades de análise consistem na disponibilidade de conhecimento e de recursos computacionais para realizar as atividades.

Nos modelos implementados, os nós que são inseridos no *grid* possuem tarefas de análise pré-definidas. Quando eles não são capazes de analisar estas informações usando os recursos que eles têm disponíveis, eles procuram no *grid* por outros agentes e delegam as atividades a eles. Eles então iniciam as atividades de análise, requerendo os dados de que precisam.

Os testes de distribuição foram realizados em uma rede local usando um número pequeno de equipamentos. A descrição da configuração e capacidade dos equipamentos utilizados é descrita na **TABELA** . Embora simples os experimentos demonstram os princípios de distribuição de tarefas de análise apresentados pela arquitetura, usando técnicas de escalonamento simples, mas que apresentam a viabilidade do sistema.

Para a realização dos testes optou-se por usar um exemplo de tarefa de análise. Esta tarefa é designada aos agentes de análise no *grid*, de acordo com o critério usado para distribuição, para que estes realizem as análises correspondentes.

No primeiro cenário de testes montado, um agente de análise tem como função realizar um conjunto de tarefas de análise que chegam entre intervalos de tempo. Este agente foi configurado para que nunca tenha capacidade de realizar esta tarefa, e por sua vez, deve enviá-la para ser executada por um agente registrado no *grid* que tenha capacidade de realizá-la. Para isso, ele consulta o diretório do sistema em busca dos agentes aptos a realizarem aquela tarefa – vale lembrar que o agente deve ter o conhecimento para realizá-la – e envia esta tarefa para que ele a realize.

TABELA I
EQUIPAMENTOS USADOS PARA OS TESTES.

Equipamento	Descrição
Máquina A	Processador Pentium III 800 MHz 192MB de Memória RAM 20GB de Disco Rígido Sistema Operacional Windows XP
Máquina B	Processador Duron 1.2 GHz 256MB de Memória RAM 40GB de Disco Rígido Sistema Operacional Linux
Máquina C	Processador Pentium II 233 MHz 64MB de Memória RAM 4GB de Disco Rígido Sistema Operacional Windows 2000
Máquina D	Processador Pentium MMX 233 MHz 64MB de Memória RAM 1.6GB de Disco Rígido Sistema Operacional Windows 2000

Um teste foi efetuado usando apenas um agente de análise – além do agente que está delegando a tarefa e que não irá executá-la - registrado no *grid* de agentes (Máquina A) para realizar as tarefas de análise. O tempo entre chegada de tarefas foi configurado para 10 segundos. Enquanto que o intervalo de dados foi configurado como 2 horas. Neste caso, o agente irá verificar todas as coletas do uso de processador dos sistemas durante este período.

Durante a realização dos experimentos, foram medidos alguns valores referentes à utilização dos recursos onde os analisadores registrados no *grid* estavam em execução durante a elaboração dos testes. O resultado da média da utilização de recursos no Equipamento A, realizando a tarefa de análise descrita anteriormente e não realizando a tarefa, é apresentado na **Figura 6**.

Neste gráfico apresentado na **Figura 6** pode-se perceber algumas características da tarefa usada para estes testes, como o fato dela acarretar considerável uso do processador, elevando a sua utilização para 67%, além de utilizar um pouco da memória RAM do sistema.

Em um segundo experimento, todos os agentes foram registrados no *grid*, e configurados como aptos a realizarem as tarefas de análise, ou seja, possuíam o conhecimento ou regras para tal. O gráfico com a utilização média de recursos pode ser visto na **Figura 7**.

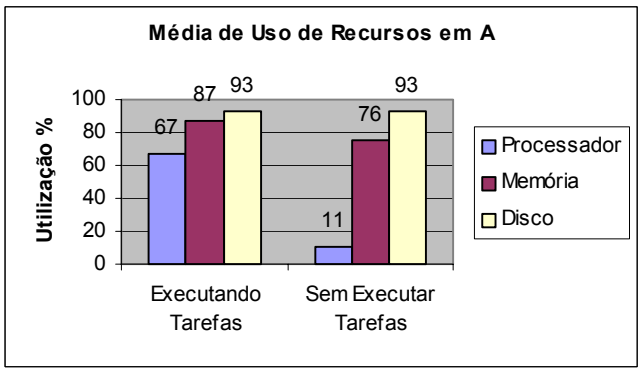

Figura 6. Apenas Máquina A realizando a tarefa.

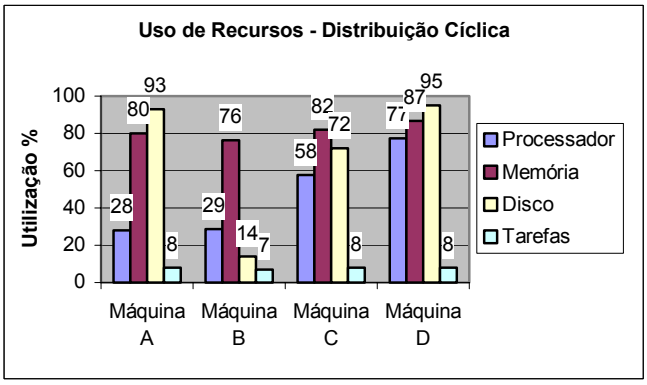

Figura 7. Distribuição cíclica de tarefas.

Neste gráfico pode ser percebida uma equalização no uso médio dos recursos. O uso do processador e de memória da Máquina A diminuiu, passando a 28% e 80% respectivamente.

Também foi testada uma distribuição aleatória entre os agentes aptos a realizarem a tarefa. Tais resultados podem ser vistos no gráfico apresentado na **Figura 8**.

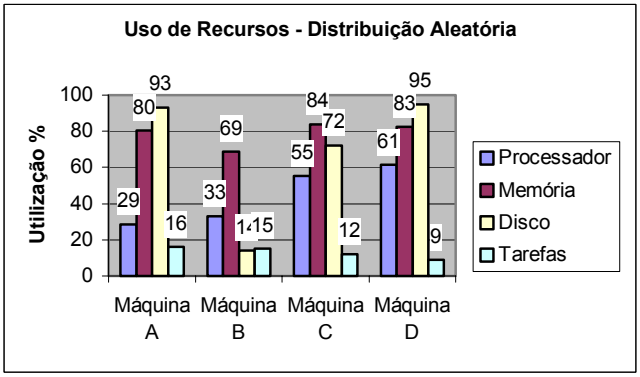

Figura 8. Distribuição aleatória de tarefas.

Tanto a abordagem de distribuição aleatória, quanto à distribuição cíclica apresentam uma equalização da utilização dos recursos nas atividades de análise de dados de gerência. Porém, ambas recaem no problema de usar estações não capazes ou com recursos escassos nas atividades de análise. Uma outra abordagem é a utilização dos medidores para o fornecimento de informações de desempenho e capacidade dos recursos computacionais de forma a auxiliar na atribuição das tarefas. Estas informações são usadas para guiar o agente responsável pelo escalonamento, na decisão, a respeito de qual agente está mais apto a realizar uma determinada atividade de análise.

Também foram realizados experimentos onde a distribuição foi feita baseando-se na capacidade e uso do processador dos recursos envolvidos (**Figura 9**). Poderiam ser usados outros fatores coletados pelo medidor de desempenho do nó *grid*, como memória ou uma taxa baseada em uma média de utilização.

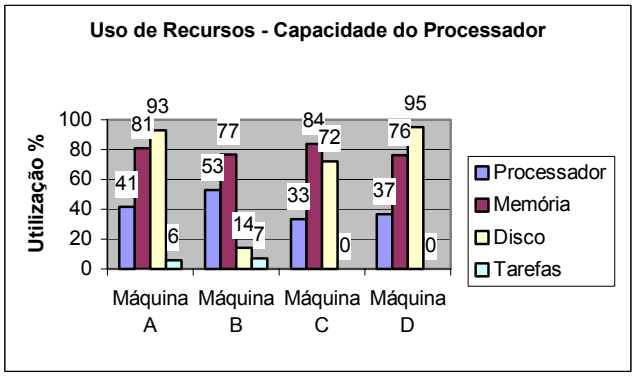

Figura 9. Distribuição baseada na capacidade e uso do processador.

V. Conclusões e Trabalhos Futuros

Este trabalho apresentou alguns resultados de implementação e experimentação de uma arquitetura de *grids* de agentes proposta para o gerenciamento de redes de computadores e de sistemas. A implementação do protótipo de sistema e sua experimentação foram apresentadas abordando apenas alguns pontos da arquitetura sugerida no trabalho.

Porém, é evidente que em um cenário de *grid,* muitos outros aspectos devem ser levados em conta, dentre eles os requisitos de segurança que não foram tratados neste trabalho. Outro fato que deve ser lembrado, é que se acredita que a utilização de uma arquitetura complexa pode ser justificável e aplicável em cenários onde o volume de informação e o ambiente gerenciado são muito grandes.

Além disso, a experimentação da arquitetura de *grid* foi realizada no âmbito de uma rede local, que era o ambiente que se tinha à disposição para o desenvolvimento do trabalho e realização dos testes. Embora tenha procurado utilizar protocolos ubíquos e ferramentas multi-plataformas para o desenvolvimento do sistema, este não foi testado em um cenário envolvendo múltiplos domínios administrativos.

Como trabalhos futuros podem ser destacados os seguintes:

- A aplicação de agentes móveis no *grid* de processamento, para prover mecanismos que possibilitem o balanceamento de carga.
- O estudo e aplicação de mecanismos de negociação entre agentes no *grid* de análise de informações e no *grid* de armazenamento de dados.
- O estudo, desenvolvimento e aplicação de agentes no *grid* com finalidades especiais como *brokers, traders* e *matchmakers*, para melhorar as capacidades de pesquisa, e diminuir o tráfego de mensagens que pode existir no ambiente de *grid* de agentes proposto.

VI. Referências

[1] National Coordination Office for Information Technology Research and Development. High Performance Computing And Communications: Foundation for America's Information Future. 1996. The Annual Supplement to the President's Budget. [Online]. Disponível em: <http://www.ccic.gov/pubs/blue96>.

[2] I. Foster, C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1999. 677p.
[3] I. Foster, C. Kesselman, S. Tuecke. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations". International Journal of Supercomputer Applications, v.15 n.3, 2001.
[4] I. Foster, C. Kesselman, J. Nick, S. Tuecke. "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration". Open Grid Service Infrastructure WG, Global Grid Forum, jun. 2002.
[5] M. Luck, P. McBurney, C. Preist. "Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing)", AgentLink, 2003, ISBN 0854 327886.
[6] M. D. Assunção, C. B. Westphall, F. L. Koch. Arquitetura de Grids de Agentes Aplicada à Gerência de Redes de Computadores e Telecomunicações. In: 21º Simpósio Brasileiro de Redes de Computadores, 2003, Natal-RN. p. 789-804.
[7] M. D. Assunção, C. B. Westphall, F. L. Koch. Grids of Agents for Computer and Telecommunication Network Management. In: ACM/IFIP/Usenix International Middleware Conference. 1st International Workshop on Middleware for Grid Computing, Rio de Janeiro, jun, 2003, p.186-193.
[8] M. D. Assunção, F. L. Koch, C. B. Westphall. "Grids of Agents for Computer and Telecommunication Network Management". Journal of Concurrency and Computation: Practice and Experience, John Wiley & Sons, Inc. (A ser publicado na edição de: março/abril de 2004).
[9] F. L. Koch, C. B. Westphall. "Decentralized Network Management using Distributed Artificial Intelligence". Journal of Network and Systems Management. Plenum Publishing Corporation. Meddletown, USA, v.9, n.4, dez. 2001, p.291-313.
[10] F. L. Koch, J-J. C. Meyer. Knowledge Based Autonomous Agents for Pervasive Computing using AgentLight. ACM/IFIP/USENIX International Middleware Conference, [Online] em IEEE Distributed Systems, Middleware 2003 Work-in-Progress, Rio de Janeiro, Brasil, 2003.

Marcos Dias de Assuncão obteve o grau de bacharel em ciência da computação em 2001 na Universidade do Oeste de Santa Catarina, Brasil e o grau de mestre na Universidade Federal de Santa Catarina em 2004. É membro do Projeto Tagere (Tópicos Avançados em Gerência de Redes) do Laboratório de Redes e Gerência da Universidade Federal de Santa Catarina, trabalhando no desenvolvimento de aplicações multi-agentes para o gerenciamento de redes de computadores e sistemas.

Fernando Luiz Koch, obteve grau de Bacharel (1993) e Mestre (1997) em Ciência da Computação pela Universidade Federal de Santa Catarina, Brasil. Ele é candidato a PhD em Ciência da Computação no Grupo de Sistemas Inteligentes da Universidade de Utrecht, na Holanda, e atualmente atua como pesquisador adjunto no Grupo de Sistemas de Informação da Universidade de Melbourne, na Austrália. Ele também é um pesquisador associado ao Projeto Tagere (Tópicos Avançados em Gerência de Redes) do Laboratório de Redes e Gerência da Universidade Federal de Santa Catarina, trabalhando no desenvolvimento de aplicações multi-agentes para o gerenciamento de redes. Suas áreas de interesse são inteligência artificial, sistemas distribuídos e a implementação de agentes autônomos para aplicações práticas e sistemas embutidos.

Carlos Becker Westphall é Professor Titular do Departamento de Informática e de Estatística, líder do Grupo de Redes e Gerência e Supervisor do Laboratório de Redes e Gerência da UFSC desde 1993. Doutor em Informática, na especialidade de Gerência de Redes pela Université Paul Sabatier, em Toulouse na França em 1991. Mestre em Ciência da Computação em 1988 e Engenheiro Eletricista em 1985, ambos pela UFRGS. É membro da IFIP(WG 6.6) e do IEEE(CNON) onde atua desde 1994, respectivamente, na organização do International Syposium on Integrated Network Management e do Network Operations and Management Symposium que são considerados os melhores eventos internacionais na área de Gerência de Redes. Atuou como coordenador de equipe no projeto europeu MAX/ESPRIT-II com a participação da Alcatel-TITN, British Telecom, HP, CSELT, SIRTI e NKT. ´E coordenador de projetos fomentados pelo CNPq, atuando como pesquisador 1C e como consultor Ad Hoc do CNPq. Desde 1995, faz parte do Board of Editors e desde 2003 é Senior Technical Editor do Journal of Network and Systems Management da Kluwer Academic / Plenum Publishers. Desde 2000, atua como secretário do IEEE CNOM (Committee on Network Operation and Management). Desde 2003 é membro do core team of the TMF UP (TeleManagement Forum Universities Program). Membro em 2004-2005 do IEEE ComSoc Membership Programs Development Board. Membro do Editorial Board do Computer Networks Journal (The International Journal of Computer and Telecommunications Networking, Former title: Computer Networks and ISDN Systems) da Elsevier Computer Science, desde 20/01/2004. Membro do time de marketing para o lançamento de um trial de um ano (2004) para uma nova publicação eletrônica patrocinada pela IEEEM ComSoc na área de gerenciamento de redes e serviços.