

Gerenciamento Integrado de Grids e Redes Baseado em Políticas

Ricardo Neisse , Lisandro Zambenedetti Granville ,
Maria Janilce Bosquioli Almeida , Liane Margarida Rockenbach Tarouco

Instituto de Informática – Universidade Federal do Rio Grande do Sul
Caixa Postal 15064 – 90501-970 Porto Alegre, RS

{neisse, granville, janilce, liane}@inf.ufrgs.br

Abstract. *This paper presents an hierarchical policy-based architecture whose goal is to integrate grid toolkits and network management tools. In the architecture proposed grid policies are translated to network policies following mapping rules defined by network administrators. The paper also describes a prototype implemented based on the architecture.*

Resumo. *Este artigo apresenta uma arquitetura hierárquica de gerenciamento baseado em políticas que tem como objetivo integrar os toolkits de grid e as ferramentas de gerenciamento de redes. Na arquitetura proposta as políticas de grid são traduzidas para políticas da rede seguindo regras de mapeamento definidas pelos administradores da rede. Este artigo também descreve um protótipo implementado a partir da arquitetura.*

1. Introdução

Devido aos requisitos de rede das aplicações e alta distribuição dos recursos a implementação e o gerenciamento de uma infra-estrutura de *grid* não é uma tarefa trivial. Por esses motivos, foram desenvolvidas bibliotecas de software, chamadas de *toolkits*, que fornecem as funcionalidades e serviços básicos de gerenciamento para manutenção de *grids* [glo, 2003] [Steen et al., 1999]. Os *toolkits* possuem facilidades para que o administrador do *grid* gerencie os recursos e mantenha os serviços críticos operando de maneira adequada.

Em um cenário típico de gerenciamento de *grids* o administrador do *grid* coordena a operação do *grid* usando o suporte fornecido pelos *toolkits* e interage manualmente com os administradores da rede em cada domínio administrativo para garantir que as configurações de rede necessárias para operação do *grid* foram executadas. Portanto, apesar de existirem ferramentas para o gerenciamento das infra-estruturas de *grid*, a maioria delas considera que a rede já está corretamente configurada para a operação do *grid*, o que nem sempre é verdadeiro.

Na tentativa de resolver os problemas de integração entre gerenciamento de *grid* e gerenciamento de rede este trabalho propõe uma arquitetura de gerenciamento baseado em políticas, que tem por objetivo integrar o gerenciamento dos recursos do *grid* com o gerenciamento da rede em cada domínio administrativo de rede através dos quais o *grid* pode estar disperso. O restante deste artigo está organizado como segue. A Seção 2 apresenta a arquitetura de gerenciamento hierárquica baseada em políticas proposta, a Seção 3 mostra o protótipo desenvolvido a partir dessa arquitetura e na Seção 4 este artigo é finalizado com conclusões e trabalhos futuros.

2. Mapeamento de Políticas de Grid para Políticas da Rede

Na arquitetura proposta neste trabalho as políticas de *grid* são mapeadas para políticas de rede através de um mecanismo de mapeamento baseado em regras. Embora o administrador da rede não defina na arquitetura as políticas de rede, ele ou ela é responsável por definir as regras de mapeamento que geram as políticas de rede necessárias pelo *grid*. Após as políticas de rede serem geradas elas são então traduzidas para ações de configuração nos dispositivos da rede.

Embora o objetivo deste trabalho não seja definir uma nova linguagem para políticas de gerenciamento de *grid*, é descrito um conjunto de elementos necessário para tais políticas, através de uma linguagem hipotética que é baseada em trabalhos previamente desenvolvidos [Sundaram et al., 2000] [Sundaram and Chapman, 2002]. Novos elementos para definição de políticas de *grid* são necessários pois as soluções encontradas não permitiam a definição integrada de requisitos de QoS de rede nem expressar políticas considerando *proxies*, conceito usado nos *toolkits* de *grids*.

Assim, na abordagem apresentada neste trabalho, uma política de *grid* (Listagem 1, primeira coluna) é composta por uma declaração condicional (*if*) contendo elementos relacionados aos usuários do *grid* (*user*), *proxies* (*proxy*) recursos (*resource*) e restrições temporais (*startTime* e *endTime*). Além das condições a política contém também ações, específicas relacionadas com o QoS necessário na rede (*networkQoS*).

```
if (user == "neisse" and
    startTime >= "11/25/2003 00:00:00" and
    endTime <= "11/25/2003 23:59:59") {
  if (resource == "LabTec Cluster") {
    allowAccess = true;
    login = griduser;
    maxProcessing = 50%;
    networkQoS = remoteProcessControl;
  }
  if (proxy == "LabTec Cluster" and
      resource == "UFRGS Data Server") {
    allowAccess = true;
    maxAllowedStorage = 40GB;
    networkQoS = highThroughputDataIntensive;
  }
}

if (srcAddress == "143.54.47.242" and
    srcPort == "*" and
    dstAddress == "143.54.47.17" and
    dstPort == "80" and
    DSCP == "*" and proto == "TCP" and
    startTime >= "11/25/2003 00:00:00" and
    endTime <= "11/25/2003 23:59:59") {
  bandwidth = 10Mbps;
  DSCP = 1;
  priority = 4;
}
```

Listagem 1. Exemplo de política de grid e de rede

A Listagem 1 (segunda coluna) apresenta um exemplo de uma política da rede gerada a partir das regras de mapeamento. Essa política declara condições onde para o tráfego gerado a partir do host 143.54.47.242 para o host 143.54.47.17, de qualquer porta origem (*), endereçado para porta HTTP (porta 80 sobre protocolo TCP), e com qualquer valor DSCP (*) reservará 10Mbps de banda, marcará os pacotes com o valor 1 no campo DS [Blake et al., 1998], e receberá prioridade 4. Essa política é válida somente durante o dia 25 de novembro de 2003. O problema que se tem aqui é como gerar esta política de rede dada uma política de *grid* contendo um QoS da rede.

O processo de mapeamento de políticas do *grid* para políticas da rede foi dividida em nove passos (Figura 1): primeiro o administrador do *grid* define e armazena em um repositório global as políticas do *grid* e as classes de serviço da rede necessárias através de um editor de políticas (1); então o administrador da rede de cada domínio administrativo define e armazena em um repositório local do domínio um conjunto de regras de mapeamento usando um editor de regras de mapeamento (2); quando a política do grid for aplicada o mecanismo de mapeamento recupera a política do repositório global de políticas (3), recupera o conjunto de regras de mapeamento do repositório local (4) e executa as regras de mapeamento usando informações recuperadas do *toolkit* como endereços e protocolos de rede dos recursos (5); o mecanismo de mapeamento armazena as políticas de rede criadas em um repositório local de políticas (6); por fim o mecanismo de mapeamento sinaliza um conjunto de PDPs no domínio

local para aplicar as políticas de rede criadas em um conjunto de PEPs (7) onde as políticas da rede são recuperadas do repositório local (8) e traduzidas para ações de configuração nos PEPs do domínio local.

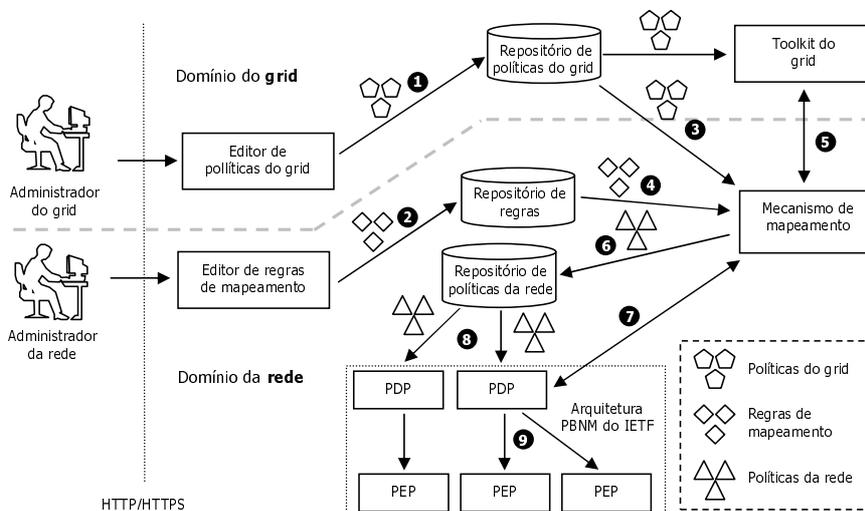


Figura 1: Policy Mapping Architecture

As regras de mapeamento são definidas pelos administradores da rede usando uma linguagem de condições-ações orientada a objetos, que são muito similares as políticas de grid e rede, com exceção que nesse caso elas controlam o processo de mapeamento. Desse modo, as regras de mapeamento pode ser compreendidas como meta-políticas que controlam o processo de mapeamento. Novas regras de mapeamento são definidas considerando um conjunto de objetos que endereçam as políticas de *grid* originais e as novas políticas de rede a serem criadas. Quatro objetos globais são implicitamente instanciados antes da avaliação de uma regra de mapeamento: *schedule*, *srcResource*, *dstResource*, e *requiredQoS*, identificando em conjunto um par de comunicação do *grid*.

Os quatro objetos citados no parágrafo anterior são instanciados a partir da política de *grid*, e podem ser usados nas condições ou nas ações de uma regra de mapeamento. Além disso, para criar novas políticas da rede uma regra de mapeamento precisa instanciar objetos da classe *NetworkPolicy* e proceder com a manipulação de seus métodos para definir as condições e ações da política (*addCondition* e *addActions*). A Listagem 2 (coluna 1) apresenta um exemplo de uma regra de mapeamento que cria duas novas políticas da rede a partir de uma única política do *grid* e aplica essa políticas em um determinado conjunto de PEPs, usando uma linguagem de seleção de dispositivos [Ceccon et al., 2003].

```

if (srcResource.address/24 == 143.54.47.0/24 and
dstResource.address/24 != 143.54.47.0/24 and
dstResource.port == 80 and
dstResource.protocol == TCP) {
p1 = new NetworkPolicy();
p1.addCondition(startTime,">=",schedule.startTime);
p1.addCondition(endTime,"<=",schedule.endTime);
p1.addCondition(srcAddress,"==",srcResource.address);
p1.addCondition(dstAddress,"==",dstResource.address);
p1.addCondition(dstPort,"==",dstResource.port);
p1.addCondition(dstProtocol,"==","tcp");
p1.addAction(DSCP,2);

p2 = new NetworkPolicy();
p2.addCondition(startTime,">=",schedule.startTime);
p2.addCondition(endTime,"<=",schedule.endTime);
p2.addAction(DSCP,2);
p2.addAction(bandwidth,requiredQoS.requiredBandwidth);
}

if (srcResource.address/24 == 143.54.47.0/24 and
dstResource.address/24 != 143.54.47.0/24 and
dstResource.port == 80 and
dstResource.protocol == TCP) {
p1 = new NetworkPolicy();
...
inPEPs = select pep
.within[srcResource.address, 143.54.47.1]
.direction["in"] from
device.type["DiffServDevice"];
inPEPs[0].deployPolicy(p1);
p2 = new NetworkPolicy();
...
outPEPs = select pep
.within[srcResource.address, 143.54.47.1]
.direction["out"] from
device.type["DiffServDevice"];
outPEPs.deployPolicy(p2);
}

```

Listagem 2. Exemplo de política de grid e de rede

3. Implementação do Protótipo

A partir da arquitetura apresentada foi implementado um protótipo na linguagem PHP. Através do protótipo, o administrador do *grid* e o administrador da rede podem especificar através de editores as políticas de *grid* e as regras de mapeamento correspondentes. O protótipo é parte do ambiente de gerenciamento QAME (*QoS Aware Management Environment*), um sistema de gerenciamento de redes baseado na Web desenvolvido na UFRGS.

As regras de mapeamento foram implementadas como um sub-conjunto da linguagem PHP. Os administradores da rede podem criar as regras de mapeamento usando uma lista de comandos apresentados na interface do usuário, não é necessário um conhecimento prévio da linguagem PHP para criação das regras. O mecanismo de mapeamento foi implementado e testado em um domínio de rede com PDPs implementados em Java para configurar QoS em roteadores FreeBSD com AltQ.

4. Conclusão e Trabalhos Futuros

Foi apresentado neste artigo uma arquitetura de gerenciamento onde políticas de *grid* são mapeadas para políticas da rede através de um mecanismo de mapeamento baseado em regras. As políticas de *grid*, se comparadas com as políticas encontradas na literatura, permitem expressar condições mais adequadas relacionadas com as operações de reservas de recursos da rede. Além disso, as políticas de *grid* suportam o uso de proxies na avaliação dos parâmetros das regras. Portanto, as políticas de *grid* com maior poder de expressão colaboram para facilitar o gerenciamento dos *grids*.

Embora as regras de mapeamento sejam bastante flexíveis elas forçam os administradores da rede a aprender uma nova linguagem para definir mapeamentos mais adequados. Nesse ponto assistentes visuais poderiam facilitar a definição de regras de mapeamento. Como trabalhos futuros destacam-se a redução da complexidade na definição das regras de mapeamento, através da introdução de interfaces de usuário mais elaboradas, por exemplo, usando assistentes. Outro ponto a ser investigado é a avaliação do desempenho da arquitetura no que diz respeito ao impacto das transferências de políticas na rede e do desempenho do mecanismo de mapeamento, principalmente considerando o número de regras e níveis de aninhamentos.

Referências

- (2003). The globus project. <http://www.globus.org>.
- Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., et al. (1998). An architecture for differentiated services. Request for Comments: 2475. IETF.
- Ceccon, M. B., Granville, L. Z., Almeida, M. J. B., and Tarouco, L. M. R. (2003). Criação e visualização de domínios dinâmicos em ambientes de gerenciamento de redes. In *21o Simpósio Brasileiro de Redes de Computadores*, volume 2, pages 805–820.
- Steen, M., Homburg, P., and Tanenbaum, A. S. (1999). Globe: A wide-area distributed system. *IEEE Concurrency*, pages 70–78.
- Sundaram, B. and Chapman, B. M. (2002). Xml-based policy framework for usage policy management in grids. In *Grid'02 3rd International workshop on Grid Computing*.
- Sundaram, B., Nebergall, C., and Tuecke, S. (2000). Policy specification and restricted delegation in globus proxies. In *SuperComputing 2000*.