A Policy-Based Hierarchical Approach for Management of Grids and Networks

Tiago Fioreze[1], Ricardo Neisse[1], Lisandro Granville[2], Maria Janilce Almeida[2], Aiko Pras[1]**

[1] University of Twente, The Netherlands
[2] Federal University of Rio Grande do Sul (UFRGS), Brazil

** Supported by CNPq scholarship - Brazil

Abstract

Grids are distributed infrastructures that have been used as an important and powerful resource for distributed computing. Since the nodes of a grid can potentially be located in different administrative domains, the underlying network infrastructure that supports grid communications has to be properly configured to provide efficient node-to-node message exchange. Currently, the management of grids and networks is performed by different and independent tools: there is no tool integration to provide, for example, an automatic network configuration. This forces grids administrators to request, in a non-automated fashion, the configuration of network devices to network administrators. In this paper we present a policy translation mechanism that generates network policies from grid requirements expressed in grid policies. The paper also presents a prototype tool that allows grid administrators to define grid policies, and network administrators to define corresponding translation rules. The proposed translation mechanism and its implementation in the associated tool show that the management of grids and networks can be executed in an integrated way.

1-4244-0143-7/06/$20.00 ©2006 IEEE

Introduction

- Grids are widely used for sharing distributed computing resources often located in different administrative domains
- The underlying network infrastructure has to provide communication services (e.g., managing QoS) among the grids entities
- The current grid toolkits do not interact either QoS provisioning architectures or network management systems
- We propose a policy translation architecture where network management policies are automatically created from grid management policies

1. Introduction

Grids are distributed infrastructures that allow the use of computing resources potentially distributed across different administrative domains. Processing power, memory, and storage are examples of resources shared by typical grid applications. Grid applications often present intensive message exchange that requires a reasonable amount of bandwidth and multicast communication support. In this context, besides grids themselves, underlying network infrastructures need to be properly managed in order to allow a feasible communication among grid entities.

Since grid resources are usually distributed across administrative domains, and each domain has different management policies, the task of configuring each domain of a grid demands coordinated management. In this scenario, two administrative figures are involved: the grid administrator and the network administrator. The grid administrator is responsible for managing grid resources, while the network administrator maintains the communication support to allow grid users to access grid resources.

The configuration of the underlying network involves, among other things, the management of QoS provisioning architectures such as DiffServ [1] and IntServ [2]. Nowadays, grid toolkits [3] [4] do not interact either QoS provisioning architectures or network management systems. That forces grid and network administrators to manually interact with each other in order to perform the required configuration. Thus, although grid toolkits provide support for grid management, the support for an integrated management of grids and networks is still missing.

In this paper we present an architecture and associated Web-based prototype that enables network administrators to define translation rules used to map grid policies (defined by grid administrators) to network policies. The main contribution of this work is that the proposed approach allows network policies to be automatically generated from grid policies, and to be deployed by a policy system [5] instead of forcing a manual interaction between grid and network administrators.

1-4244-0143-7/06/$20.00 ©2006 IEEE

Related work

- General grid toolkits provide services to manage and install a grid
- Toolkit examples:
  - AccessGrid
  - Globus
- Current toolkits provide grid management but do not support integrated management of grids and networks

2. Related work

This section presents the most common grids toolkits used nowadays and also policy-based approaches that provide grid management.

2.1 Grid toolkits

The management of grid resources is not trivial, since the grid resources may be located in different administrative domains. Typical grid management tasks that need to be coordinated in the grid environment are, for example, user authentication and resource scheduling. In order to simplify grid management tasks, software libraries, which are called *toolkits* in the grid world, have been developed. These toolkits provide basic services and try to reduce the initial work needed to install and manage a grid environment. AccessGrid [4] and Globus [3] are widely known examples of grid toolkits.

A commonly required network configuration in a conference grid, implemented for instance with AccessGrid, is to allocate network bandwidth for multicast audio and video flows. This configuration must be executed in all administrative domains that are part of the grid, to guarantee feasible audio and video transmission. The current version of AccessGrid, however, considers that the underlying network resources will be available prior and during a videoconferencing session thanks to network operator intervention, which is not always true.

A toolkit that explicitly considers an integrated network management infrastructure is Globus. By means of its Globus Architecture for Reservation and Allocation (GARA) [5], interfaces for processor and network reservations are provided to Globus users. Globus also explicitly defines the concept of proxy, which is a grid resource that runs grid tasks on behalf of a human user using the same access privileges from that user.

Although grid toolkits already provide proper grid resource management, they do not interact either QoS provisioning architectures or network management systems, which is a requirement for proper communication among the entities participating in a grid environment.

1-4244-0143-7/06/$20.00 ©2006 IEEE

Related work

- Policy-based grid solutions have been proposed for grid management
- These solutions allow the specification of grid resources and users parameters
- However, QoS provisioning and network management cannot be specified in the current solutions

2.2 Policy-based grid management

In addition to the management support provided by grid toolkits, some policy-based solutions have also been proposed for grid management. Sundaram *et al.* [6] propose a grid policy approach where grid resource and user access can be controlled, but that approach supports neither the specification of network QoS parameters nor the concept of user proxies.

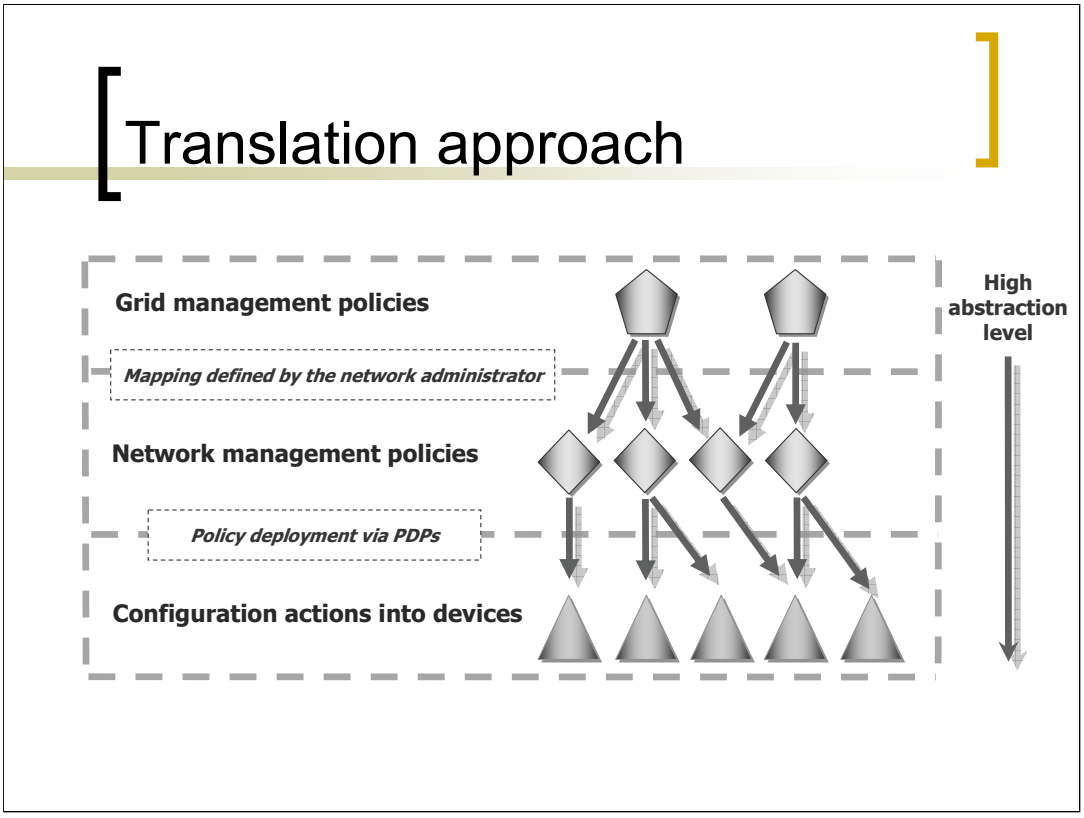
Sahu *et al.* [7] define a management service where global grid policies are combined with policies for each local domain. Local policies have higher priority than global ones. For example, if a global policy defines a 10GB disk allocation in a server, but the local administrator defines only 5GB, the local policy is chosen. That may lead to potential conflicts of interest between the grid and network administrator, as well as impacting the definition of grid and network policies.

Yang *et al.* [8] present a policy-based architecture organized in two layers. One of these layers follows the IETF concepts of Policy Enforcement Point (PEP) and Policy Decision Point (PDP) [9], while the other one uses the concept of programmable networks (active networks) in a policy middleware. The main drawback in this work is that it does not present how grid and network policies are defined in the proposed multi-layer architecture.

Sander *et al.* [10] propose a policy-based architecture for QoS configuration in different administrative domains that are members of a grid. The policies are defined in a low level language similar to the network policies defined by the IETF [11]. The Sander *et al.* approach defines an inter-domain signaling protocol that sequentially configures the grid domains that are members of an end-to-end communication path. Although the proposed architecture is based on policies, it does not provide any support that integrates such architecture with grid toolkits previously presented.

Considering these solutions, one can identify an integration problem in the current management of grids and networks. The problem consists in forcing the grid administrator to perform grid management using grid toolkits and to manually interact with network administrators in each domain to guarantee that the underlying network is properly configured for the grid operation. This leads to a situation where whenever a grid requirement that implies a new configuration in the network infrastructure is changed, a manual coordination between the grid and network administrators is needed. The support provided by grid toolkits to solve this situation is very limited and, in most cases, not even exist.

1-4244-0143-7/06/$20.00 ©2006 IEEE



3. Proposed translation approach

The system architecture proposed in our work defines a translation mechanism where network policies are generated following translation rules that specify how grid policies should be mapped. In such a translation process, the grid administrator defines grid management policies, and the network administrator defines translation rules. Then, an automated process generates the network policies applying the translation rules over grid policies. In this new scenario, the network administrator no longer defines static network policies: he or she is now in fact responsible for defining only the translation rules of the translation mechanism. The network policies generated by the translation mechanism are then translated to network configuration actions and executed by Policy Decision Points (PDPs) [9] of the regular IETF policy-based management architecture.

The next subsections present the definition and examples of grid and network policies, as well as the details of the mechanism used to implement the mapping process.

1-4244-0143-7/06/$20.00 ©2006 IEEE

Grid policies

- A hypothetical language is used to define grid policies
- The language defines a set of elements to express grid requirements:
  - Grid users and resources
  - Proxies and network QoS requirements

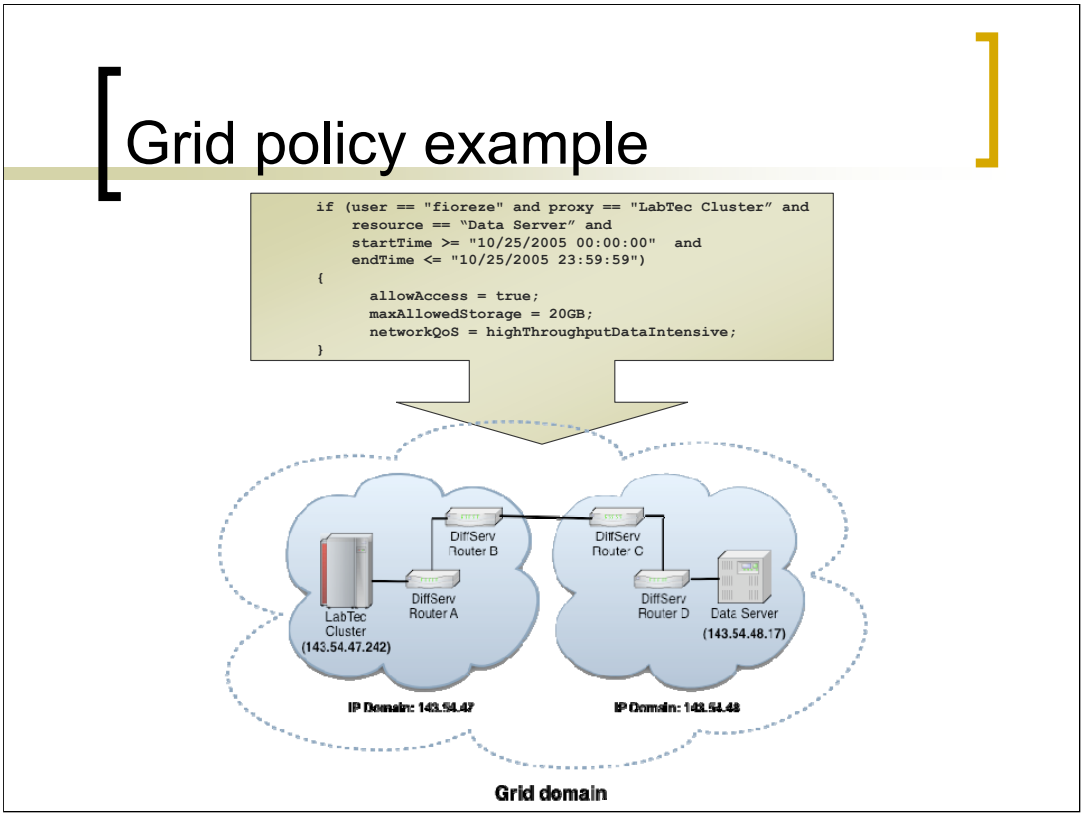
3.1 Grid Policies

None of the grid policy languages previously presented expresses grid requirements concerning users, resources, proxies, and network QoS all at the same time. Some of the languages provide polices to control proxy instantiation and lifetime [12], for example, but they do not consider bandwidth allocation.

Due to these restrictions, in our work grid polices are defined through a hypothetical language, which is based on a previous work about grid policies as defined in [6]. The important point about such hypothetical grid policy language is that it enables the definition of all grid requirements mentioned before. Policies are defined based on the condition-action model from the IETF, where a policy rule is composed by condition and action statements.

A policy condition consists of a list of variables and associated values that must evolve to true in order to turn the policy valid. An action is a list of variable attributions that are triggered when the policy condition just turns to be valid. More formally, a grid policy is composed of a conditional statement (`if`) containing conditional elements related to grid users (`user`), proxies (`proxy`), resources (`resource`), and time constraints (`startTime` and `endTime`), and also a list of actions that range from a simple action such as access permission (`allowAccess`) to an action that provides QoS requirements (`networkQoS`). The next pages present some examples of the policy language condition-action elements.

1-4244-0143-7/06/$20.00 ©2006 IEEE



This grid policy example shows that the user `fioreze` is able to access, during a specific period of time, a grid cluster (`LabTec Cluster`) located in the domain `143.54.47` and a storage server (`Data Server`) placed in the domain `143.54.48` via the grid cluster. In this policy the user does not have direct access to the storage server, but he is able to indirectly store information generated by the cluster into the storage server. Moreover, the grid policy contains the QoS requirement `highThroughputDataIntensive`, which means the network devices between the `LabTec Cluster` and the `Data Server` have to be configured to support a high throughput while transferring grid data. In addition, there is no specific network device configuration to be executed in the path linking the user's host and the storage server, since no policy directly binding the user to the storage is defined in the example.

1-4244-0143-7/06/$20.00 ©2006 IEEE

Network policies

- Network policies are automatically generated by translation rules
- Generated network policies use the same language used to defined grid policies

3.2 Network Policies

As mentioned before, network policies are generated by the translation mechanism. We define here a translation model where the network administrator is able to define translation rules to generate network policies given grid policies and QoS requirements. That releases the network administrator of having to manually create network policies accordingly to grid polices created by the grid administrator.

Final network policies generated from such translation mechanism are defined through the same hypothetical grid language used in the definition of grid policies.

1-4244-0143-7/06/$20.00 ©2006 IEEE

Network policy example

```
if (srcAddress == "143.54.47.242" and srcPort == "*"
    and
    dstAddress == "143.54.48.17" 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;
}
```

The example above shows the network policy resulting from the grid policy presented in the slide 7. The condition part of this policy states that during the entirety of November 25th, the traffic generated by host 143.54.47.242 sent to host 143.54.48.17, with any source port (*), addressed to the HTTP port (port 80), using TCP as the transport protocol and with any value as DSCP (*) will have, as actions, the allocation of 10Mbps of bandwidth and the value 1 marked in the DS field. It is interesting to notice that this network policy would be applied in each DiffServ router (A,B,C and D) between `LabTec Cluster` and `Data Server` showed in the slide 7.

In the next subsection, we present an example of a translation rule that would generate this network policy.

1-4244-0143-7/06/$20.00 ©2006 IEEE

Translation rule example

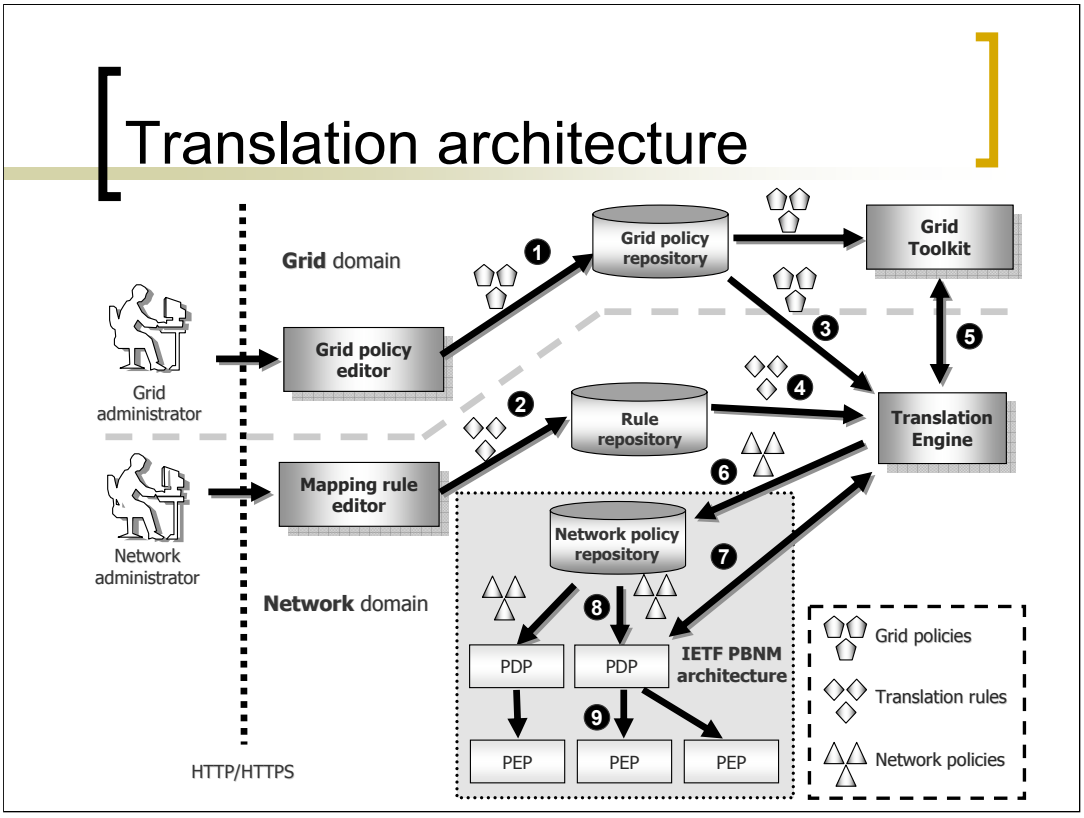
```
if (srcResource.address/24 == 143.54.47.0/24 and
     dstResource.address/24 == 143.54.48.0/24 and
     dstResource.port == 80 and dstResource.protocol == TCP)
{
    p1 = new NetworkPolicy();
    p1.addCondition(startTime,">=",schedule.startTime);
    p1.addCondition(endTime,"<=",schedule.endTime);
    p1.addCondition(srcAddress,"==",srcResource.address);
    p1.addCondition(dstAddress,"==",dstResource.address);
    p1.addCondition(dstPort,"==",dstResource.port);
    p1.addCondition(dstProtocol,"==","tcp");

    p1.addAction(DSCP,1);
    p1.addAction(bandwith,requiredQoS.requiredBandwidth);

    inPEPs = select pep within[srcResource.address, 143.54.47.1]
             direction["in"] from device.type["DiffServDevice"];
    outPEPs= select pep within[srcResource.address, 143.54.47.1]
             direction["out"] from device.type["DiffServDevice"];

    inPEPs[0].deployPolicy(p1);
    outPEPs.deployPolicy(p1);
}
```

3.3 Translation rules

Translation rules are defined by network administrators of each administrative domain that joins the grid environment. In the translation rules, the network administrator defines the appropriate configuration for his managed network considering the requirements defined in the grid policies created by grids administrators. It is interesting to notice that one grid policy may have different associated translation rules due to network infrastructures constraints.

The translation rule shown in the example above defines in the network policy `p1` the marking of packets and allocation of bandwidth in the underlying network, operating with the IETF DiffServ architecture, presented in the slide 7. However, `p1` is only created if the original grid policy states that the source resource is located in the administrative domain `143.54.47.0/24` and the destination resource belongs to the administrative domain `143.54.48.0/24`. The network policy `p1` verifies the local and remote addresses, the remote port (80), and the transport protocol (TCP) of the network packets in order to mark the DS field with the DSCP 1 and as well as to guarantee the required bandwidth determined in the original grid policy.

1-4244-0143-7/06/$20.00 ©2006 IEEE

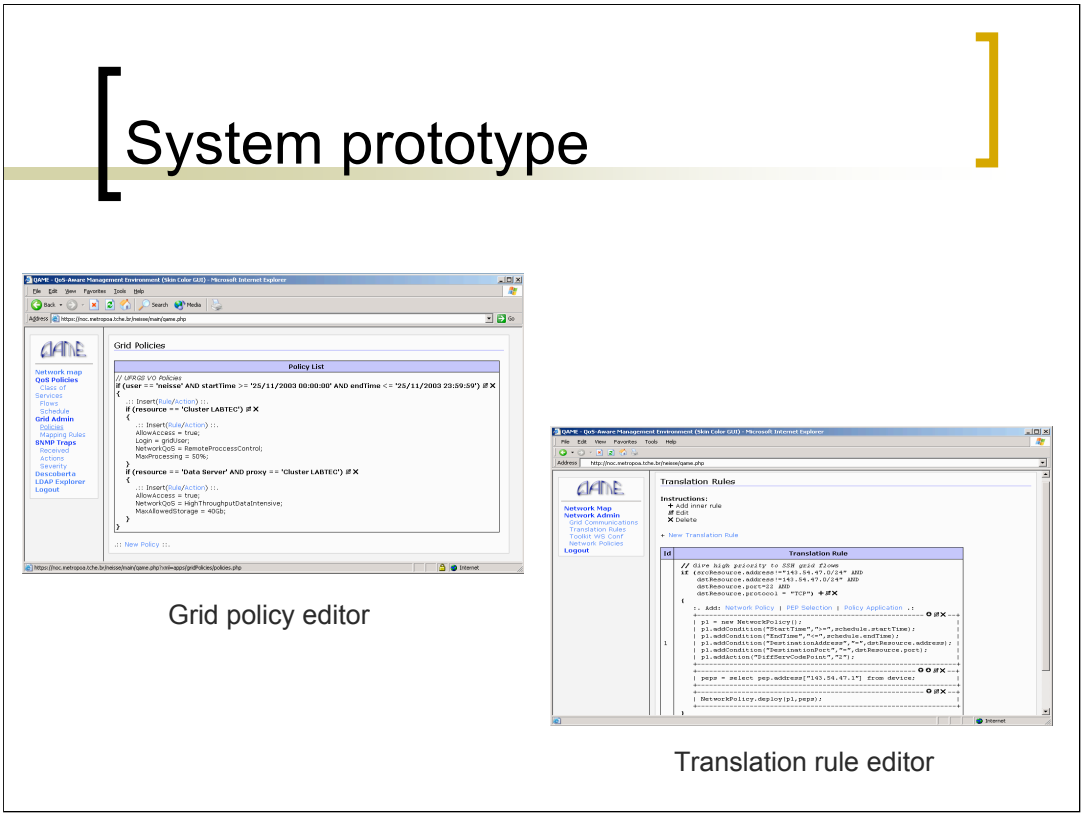


3.4 The translation architecture

The translation architecture operates according to the following steps also depicted in the figure above:

1. The grid administrator defines grid policies and required associated network classes of services through a grid policy editor and stores them in a global grid policy repository;
2. The network administrator of each administrative domain joining the grid environment defines a set of translation rules using a translation rule editor and stores them in a local rule repository;
3. Once the grid administrator wants to deploy a policy, the translation engine retrieves such policy from the global grid policy repository;
4. The translation engine then retrieves the set of translation rules from the local rule repository;
5. The translation engine translates the grid policies based on the translation rules and consults the grid toolkit to discover network addresses and protocols information;
6. Once the translation engine builds up new network policies related to the local domain, these policies are stored back in a local network policy repository;
7. Then the translation engine signals a set of PDPs in the local domain in order to deploy the just created network policies in a set of PEPs;
8. Signaled PDPs retrieve network policies from the local repository;
9. PDPs translate network policies to configuration actions in order to deploy such policies in the local domain PEPs.

1-4244-0143-7/06/$20.00 ©2006 IEEE



4. System prototype

The system prototype was developed as a module of QAME (QoS-Aware Management Environment) [14], which is a Web-based network management system developed in PHP at the Federal University of Rio Grande do Sul. In this Web-based prototype, the grid administrator can define grid policies, grid classes of service and the location of the several translation engines dispersed through the network administrative domains that compose the grid. We also provide in the prototype a graphical user interface that allows the network administrator to create translation rules and configure the translation engine, for instance, specifying which grid toolkit is used.

In the Web browser snapshots shown above, one can check the grid policy and translation rule editors provided by the system prototype. The Grid policy editor snapshot shows a grid policy created by a grid administrator. In order to create a grid policy, the grid administrator must define the policy conditions and its associated actions. Once the grid policy is created and stored in the grid policy repository, network administrators will be able to create associated translation rules.

The Translation rule editor snapshot shows the translation rule editor used by the network administrators. It is interesting to notice that in order to define translation rules the network administrator must only select a list of pre-defined grid policies previously defined by the grid administrator.

The implementation of our system has been based on the Web Services technology to provide the communication among its components. This enables inter-domain communication because Web Services traffic is layered on top of Internet protocols such as HTTP, SMTP, and FTP. Such inter-domain communication is particularly useful in the case of grid management, since grid entities are typically distributed across different domains, as observed before.

We have been using the Globus toolkit to provide the management of grid resources. The translation mechanism is able to communicate with Globus via Web Services offered by Globus to external clients.

1-4244-0143-7/06/$20.00 ©2006 IEEE

Conclusions

- Grid policies defined in this work allow an integrated management of grids and networks
- Even though mapping rules facilitate the translation process, they are not easily defined because they require:
  - Previous agreement between the grid and network administrators
  - Good knowledge of the network and grid infrastructure

5. Conclusions

We have presented in this paper an architecture that translates grid policies to network policies through a translation mechanism. The translation mechanism uses translation rules defined by network administrators located across the administrative domains of a grid environment. Since each domain has its own network administrator that defines local translation rules a single grid policy is potentially translated to different network policies in each administrative domain due to, for instance, different QoS provisioning architectures.

Although the translation rules are flexible, this forces the network administrators to learn a new language to define more adequate translations. We believe that visual wizards would ease the definition of translation rules. We have also argued that grid policies are supposed to be defined by grid administrators, instead of network administrators. In order to do that, we have presented a set of grid policy definition elements to be used by the grid administrator while defining grid policies. Grid policies defined with such elements, compared to the grid policy definition languages found today, express richer rules, because such grid policies support the concept of proxies and explicitly express network QoS requirements used in the translation process.

The translation mechanism, besides receiving grid policies and translation rules, needs to interact with both grid toolkit and policy-based network management system in order to build and deploy the expected network policies. We have presented a system prototype that uses the QAME management system and the Globus toolkit. Concerning Globus, just a subset of the grid policies can be effectively used, since the toolkit does not fully support the grid policy language yet. The current communication between QAME, Globus, and the translation mechanism is achieved using Web Services.

Finally, we believe that the main contribution of this work is that it simplifies the management of grids and networks through an integrated approach that allows network policies to be automatically generated instead of manually defined by network administrators.

1-4244-0143-7/06/$20.00 ©2006 IEEE

Thanks for your attention!

- Contact:
  - Tiago Fioreze (t.fioreze@utwente.nl)
  - Ricardo Neisse (r.neisse@utwente.nl)
  - Lisandro Granville (granville@inf.ufrgs.br)
  - Maria Janilce Almeida (janilce@inf.ufrgs.br)
  - Aiko Pras (a.pras@utwente.nl)

References

1. K. Nichols, S. Blake, F. Baker and D. Black. "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", Request for Comments 2474, IETF, December 1998.
2. Braden, R., Clark, D. and S. Shenker. "Integrated Services in the Internet Architecture: An Overview", Request for Comments 1633, IETF, July 1994.
3. Globus, "The Globus project", 2003, Availabe in: <http://www.globus.org>.
4. AccessGrid, "The Access grid (ag) user documentation", 2003, Available in: <http://www-fp.mcs.anl.gov/fl/accessgrid/>.
5. I. Foster, M. Fidler, A. Roy, V. Sander, and L. Winkler, "End-to-End Quality of Service for High-end Applications", *Computer Communications*, 27(14): pp. 1375–1388, 2004.
6. B. Sundaram and B. M. Chapman, "XML-Based Policy Framework for Usage Policy Management in Grids", in *Grid'02 3rd International workshop on Grid Computing*, pp. 194–198, Nov. 2002.
7. D. C. Verma, S. Sahu, S. B. Calo, M. Beigi, and I. Chang, "A Policy Service for Grid Computing", in *Grid Computing - GRID 2002, Third International Workshop*, Lecture Notes in Computer Science, pp. 243–255, Nov. 2002.
8. K. Yang, A. Galis, and C. Todd, "Policy-Based Active Grid Management Architecture", *10th IEEE International Conference on Networks*, pp. 243–248, Aug. 2002.
9. A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser, "Terminology for Policy-Based Management", Request for Comments: 3198, IETF, Nov. 2001.
10. V. Sander, W. Adamson, I. Foster, and R. Alain, "End-to-End Provision of Policy Information for Network QoS", *10th IEEE International Symposium on High Performance Distributed Computing*, pp. 115–126, Aug. 2001.
11. R. Yavatkar, D. Pendarakis, and R. Guerin, "A Framework for Policy-Based Admission Control", Request for Comments 2753, IETF, Jan. 2000.
12. B. Sundaram, C. Nebergall, and S. Tuecke, "Policy Specification and Restricted Delegation in Globus Proxies", in *SuperComputing* Nov. 2000.
13. B. Moore, "Policy Core Information Model (PCIM) Extensions", Request for Comments 3460, IETF, Jan. 2003.
14. L.Z. Granville, M.B. Ceccon, L.M. Tarouco, M. J. Almeida, and A.S. Carissimi, "An Approach for Integrated Management of Networks with Quality of Service Support Using QAME", in *12th International Workshop on Distributed Systems: Operations & Management - DSOM 2001*, pp. 15-17, Oct. 2001.

1-4244-0143-7/06/$20.00 ©2006 IEEE