HAL
open science

A Network Topology Description Model for Grid Application Deployment

Sébastien Lacour, Christian Pérez, Thierry Priol

To cite this version:

Sébastien Lacour, Christian Pérez, Thierry Priol. A Network Topology Description Model for Grid Application Deployment. [Research Report] RR-5221, INRIA. 2004, pp.22. inria-00070773

HAL Id: inria-00070773

https://inria.hal.science/inria-00070773v1

Submitted on 19 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A Network Topology Description Model for Grid Application Deployment

Sébastien Lacour — Christian Pérez — Thierry Priol

N° 5221

June 3rd, 2004

Thème NUM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--5221--FR+ENG

A Network Topology Description Model for Grid Application Deployment

Sébastien Lacour, Christian Pérez, Thierry Priol*

Thème NUM — Systèmes numériques
Projet Paris

Rapport de recherche n° 5221 — June 3rd, 2004 — 22 pages

Abstract: Computational grids are probably among the most *heterogeneous* computing systems. However, they are very attractive for their potential computational power. Their heterogeneity is especially perceptible during the deployment of grid applications. In particular, an automatic and efficient deployment of grid applications on resources requires to have access to the characteristics of the resources. This paper presents a description model of (grid) networks which provides a *synthetic* view of the network topology. The simplicity of the proposed model does not hinder the description of *complex* network topologies (asymmetric links, firewalls, non-IP networks, non-hierarchical topologies).

Key-words: Grid Information Service, Network Topology, Description Model, Grid Resource Description, Constrained Application Deployment.

* {Sebastien.Lacour,Christian.Perez,Thierry.Priol}@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 Rennes Cedex (France)
Téléphone : +33 2 99 84 71 00 —Télécopie : +33 2 99 84 71 71

Modèle de description de topologie réseau pour le déploiement d'applications sur grille de calcul

Résumé : Les grilles de calcul font probablement partie des infrastructures de calcul les plus *hétérogènes*. Cependant, leur puissance potentielle de calcul présente un intérêt incontestable. L'hétérogénéité des grilles de calcul se manifeste en particulier au moment du déploiement d'applications sur la grille. Plus précisément, un déploiement automatique et efficace d'applications sur grille de calcul nécessite de connaître les caractéristiques des ressources de la grille. Ce papier présente un modèle de description de réseau (de grille de calcul) qui offre une vision *synthétique* de la topologie du réseau de communication. La simplicitié du modèle proposé permet néanmoins la description de topologies réseau *complexes* (liens asymétriques, pare-feux, réseaux non IP, topologies non hiérarchiques).

Mots-clés : Service d'information sur la grille, topologie réseau, modèle de description de ressources, description de ressources de grille de calcul, déploiement d'applications sous contraintes.

1 Introduction

Computational grids are probably the most *heterogeneous* computing systems: they can be made of computers with different operating systems, various hardware and software, different storage capacities, CPU speeds, network connectivities and technologies. Although computational grids are attractive for their potential computational power, there are still difficulties to exploit such highly heterogeneous resources. Deployment is a very important phase as it bridges the gap between the user (the application) and the grid (the resources). The first step of the deployment phase consists in selecting a set of resources (including computers and network connections) satisfying constraints imposed by the application; in the second step, the application is launched on the selected resources [24]. The first step is a difficult part of application deployment. Ideally, deployment should be as automatic as possible: a user should not have to manually select resources. He or she should just have to specify the application's requirements. In order to achieve *automatic deployment*, the characteristics of both the resources and the constraints of the application need to be described precisely to allow for better, automatic resource selection.

Accurate information about compute nodes, installed software, network connectivity and performance properties is required [39] for a pertinent deployment. Previous works succeed in describing properly the compute nodes (CPU speed, memory size, operating system, *etc.*), but generally fail to describe the network topology and its characteristics in a simple, synthetic, and complete way.

This paper presents a *network topology description model for computational grids*. We target both simple and complex grids, such as those currently deployed (the Grid Physics Network, GriPhyN [20], or TeraGrid [42]), as well as any grid which can be devised. For example, the model should include network connectivity information and support not purely hierarchical networks, asymmetric links, firewalls.

The rest of this paper is divided as follows. Section 2 presents some examples of real-world grid network topologies which we need to be able to describe. The related work with respect to network description is analyzed in Section 3. Our model of a grid network topology description is introduced in Section 4 and an implementation which extends the MDS2 module of the Globus Toolkit is presented in Section 5. Section 6 concludes the paper.

2 Grid Networks

Before deploying the processes of a distributed application on a grid, the compute nodes on which the application will be run must be selected. This node selection can be constrained by both user-level and application-level requirements. A user-level constraint is used for the user's comfort, such as "I want this application to run in less than two hours." An application-level constraint is a requirement imposed by the application developer, like "this application needs 1 GB of memory."

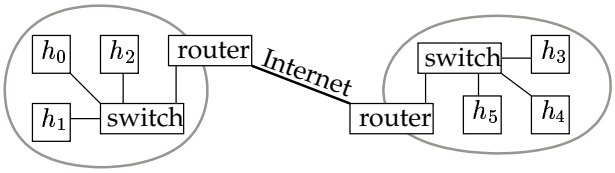

Figure 1: Grid A: a simple grid made of two Fast-Ethernet clusters.

Then, the user-level and application-level constraints must be translated into computer-level and network-level queries. Ideally, this translation process should be automatic, mixing user-level and application-level constraints as well as information on the behavior of the application: this very difficult problem is not the focus of this paper. The computer and network description must make it possible to satisfy such compute-level and network-level queries as "I want 32 computers connected by a Myrinet-2000 network" or "I want 32 computers connected by a network of at least 1 Gb/s, distant from the visualization host of IP a.b.c.d by at most 5 ms of latency".

Note that the process of automatically selecting the resources which will execute the application to deploy, *i.e.* mapping the processes onto the compute nodes of a grid, has already been studied and is out of the scope of this paper: see the work done by the ICENI project [17, 16], Sekitei [23], Condor's matchmaking [36] and an extension [25], or [30] for algorithms which dynamically remap the subtasks of a running application.

Note also that we do not address the issue of feeding the grid information service (GIS) automatically, or monitoring the network. The network description might be entered manually by administrators at distributed grid sites, or automatically by some monitoring tool (see subsections 3.4 and 3.8, as well as [6, 11]).

The network must be described *accurately* enough to answer these queries, and the *organization of the information* must make it as easy as possible to satisfy them. As the properties of the *compute nodes* of a grid are usually described properly and accurately (see Section 3), this paper focuses on the description of the *network topology*, and this section presents only examples of real-world grid network topologies which we need to be able to describe. However, we keep in mind that realistic queries are made of both network constraints and computer constraints.

2.1 Simple Grids

Figure 1 shows the physical description of a simple grid (Grid A), made of two Fast-Ethernet clusters connected together through the Internet. Each cluster is made of three hosts (h_i). The description of such a grid should be as simple as the grid itself.

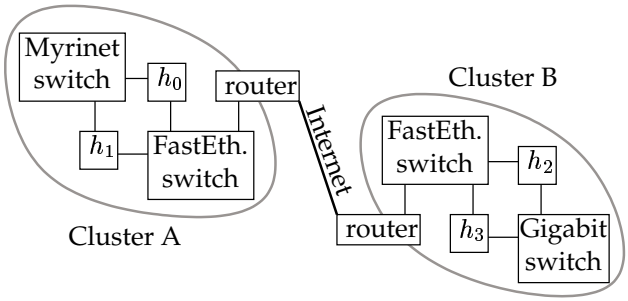

Figure 2: Grid B: overlapping of various network technologies.

2.2 Connection to Various Networks

Figure 2 shows the physical description of a grid made of two clusters (Grid B). Both clusters are equipped with two different networks. Cluster A has Fast-Ethernet and a Myrinet [5] network; Cluster B has Fast-Ethernet and a Gigabit Ethernet network. Note that the Gigabit Ethernet cards have IPs which cannot be routed from outside the Gigabit Ethernet cluster. Note also that Myrinet is not an IP network: the software which grants an application access to this network may be GM [31], BIP [35] or MX [32]. This case also encompasses clusters made of nodes with private IPs (*i.e.*, locally valid IPs), should they resort to network address translation (NAT, [38]) or not: routing is not the focus of our work, but the network description must enclose enough information to start a "forwarder" on the front-end node of such a cluster at deployment time in order to relay messages to/from the nodes with private IPs.

2.3 Firewalls, NAT and Asymmetric Links

Grids encompass several administrative, organizational domains. For security reasons, local networks are very often protected against intrusion by firewalls, closing certain TCP/UDP ports for network traffic coming from certain domains, or resorting to NAT (Network Address Translation [38]). As Figure 3 illustrates (Grid C), the list of open or closed ports can be arbitrary for network traffic from/to any specific domain: a realistic, accurate network description must include firewall-related information. Note also that the network traffic permitted on a link is not symmetric: the open ports on a link may be different depending on the direction of the flow. The same remark applies for network performance properties, such as bandwidth on asymmetric links.

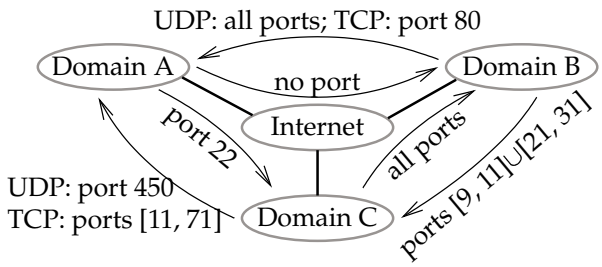

Figure 3: Grid C: firewalls limiting network traffic selectively.

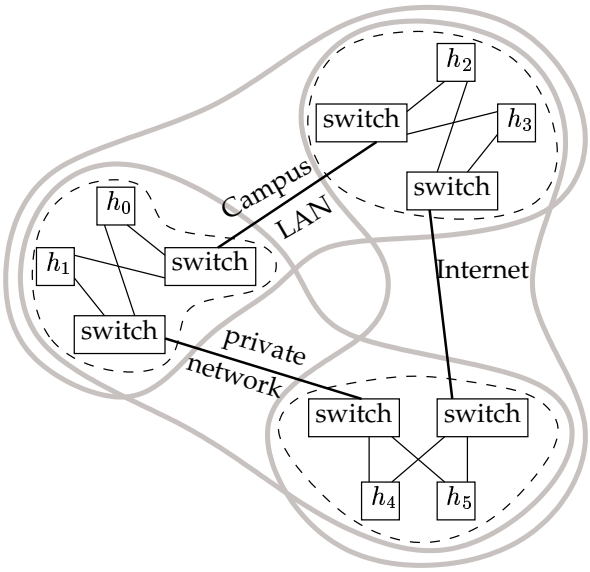

Figure 4: Grid D: non-hierarchical network.

2.4 Non-Hierarchical Network Topologies

Figure 4 shows an example of non-hierarchical network (Grid D). This network is made of three Fast-Ethernet clusters. Two clusters only are connected to the Internet. Another pair of clusters is connected through a private, dedicated high-performance network. Two clusters are located in the same institute, connected over a Local-Area Network (LAN). This network configuration is complex, but it is realistic (the authors actually have access to a similar platform), so the network description must be able to represent a grid such as Grid D.

2.5 Heterogeneous Network Description

To put it in a nutshell, a useful, relevant network description must include the following three pieces of information.

- First, *network topology* must describe which computers are connected with each other through a particular network, along with the firewalls, and allowing non-hierarchical topologies.
- Second, the *numerical network performance characteristics* must be provided, in terms of bandwidth (maximum, average, *etc.*), latency, jitter, loss, *etc.*
- Third, the resource description must include the various network *software* or drivers which enable an application to access a particular (potentially non-IP) network technology, such as Myrinet [5].

This paper focuses on the first aspect of network description, *i.e.* network *topology*.

3 Related Work

The issue of describing networks and grid resources has already been tackled. But new challenges arise with grids: grid network topologies are not limited to the Internet topology, including firewalls, non-IP networks, non-hierarchical topologies, where some computers cannot communicate directly with certain others. Grid networks must be described in a scalable (likely distributed) manner and with enough accuracy for a relevant deployment, including network software. Approaches based on a per link description miss synthetic, functional topology information. As this synthetic information is usually needed during the deployment phase [39], it must often be computed. However, this resource consuming computation can be avoided if the information service directly provides a synthetic view of the network topology.

3.1 Globus MDS

MDS (Monitoring and Discovery Service, [12]) is the Grid Information Service (GIS) of the Globus Toolkit [18]. MDS allows for flexible, hierarchical, distributed (scalable), extensible information storage; it is based on an LDAP directory. MDS includes robust authentication and authorization mechanisms. MDS also supports dynamic sources of information, such as the Network Weather Service (NWS, [44, 40]).

MDS describes the compute nodes of a grid conveniently and accurately (CPU speed and number, memory size, disk space, operating system type and version, *etc.*), but it does not currently describe network interconnections between computers. Neither does MDS hold information about the network performance properties between the nodes of a grid.

3.2 RSD from ZIB

RSD (Resource Service Description, [7, 22]) is "a software architecture for specifying, registering, requesting and accessing resources and services in complex heterogeneous computing environments." It is developed at Konrad-Zuse-Zentrum für Informationstechnik in Berlin (ZIB).

RSD allows for dynamically changing performance data of the network resources and it is not restricted to IP networks. RSD seems to be able to describe any network topology by specifying each physical network link between any individual computer, router or switch on a pairwise basis. However, RSD is designed for purely hierarchical network topologies, so RSD is ill-suited to represent Grid D presented in Subsection 2.4. RSD does not either take firewalls into account, and its description of each physical link may not be scalable in a grid environment.

3.3 The NMWG of the GGF

The Network Measurement Working Group (NMWG, [28]) of the Global Grid Forum (GGF) provides a sound, interoperable way of describing network performance characteristics, but their work does not focus on network topology description. The proposed recommendation [26] shows a good work on the classification of network characteristics and measurement methodologies, but it does not specify anything precise about network topology description, while topology description should come before network characteristics description. The NMWG only considers nodes and paths, providing too many details on the physical connectivity without discussing scalability in a grid environment: for application deployment, we need a higher-level, logical topology description of a grid.

However, our network topology description model and the performance characteristics description proposed by the NMWG are complementary.

3.4 Remos

Remos (REsource MOnitoring System, [27]) is a piece of software which allows network-aware applications to obtain relevant network information. Remos represents the logical network topology using a graph. The nodes of the graph are the computers, and the edges are the network links. Remos automatically derives the topology of IP-based Local-Area Networks by performing network performance measurements: Remos is concerned only with LANs, not grids. Remos does not take firewalls into account, and it is restricted to IP networks.

3.5 GridLabMDS

GridLab [1] is a project which aims to provide new capabilities for applications to exploit the power of grid computing. GridLab relies on Globus/MDS (see Subsection 3.1) and states that the bare MDS system lacks information about network and installed software [2].

GridLabMDS [3] extends Globus/MDS to describe available software and firewalls by specifying the open ports on each host. But this is not enough to describe the grid network precisely: a firewall can be open or closed *with respect to specific domains*, while GridLabMDS cannot describe this sort of firewall. Neither does GridLabMDS hold information about the network topology or the nature of the network links.

3.6 ENV and GridML from SDSC

ENV (Effective Network Views, [37]) is a project developed at San Diego Supercomputer Center (SDSC). ENV uses an XML dialect (GridML, [19]) to describe a network. This software can represent clusters of computers connected over IP networks only within Local-Area Networks (LANs). Thus, it is not adapted to describe computational grids. GridML also assumes purely hierarchical network topologies only, it does not take firewalls into account and it is restricted to IP networks.

3.7 GridG and RGIS Relational Database

GridG [29] is a grid network topology generator useful to realistically simulate the resources of a grid and designed to evaluate middleware systems for computational grids. However, GridG only considers purely hierarchical IP-based networks. GridG does not either take firewalls into account.

RGIS [14] is a Relational Grid Information Service system based on a relational database. This approach allows *composition* of information ("joins" in the database language) to answer such complex queries as those presented in Subsection 2 in order to map the processes of an application to the resources of a grid. RGIS describes the network topology in a point-to-point manner, specifying every network link between any two computers. However, joins over numerous database tables at distributed locations have the potential for introducing serious performance problems. As RGIS relies on GridG, it is restricted to IP networks and does not take firewalls into account.

3.8 TopoMon

TopoMon [13] is a monitoring tool for grid networks which augments NWS [44, 40] with topology information: it uses `traceroute` to discover the network topology between the monitored sites, tracking down shared network paths over the Internet. As TopoMon relies on NWS, it is restricted to IP networks. The network topology is described on a pairwise basis, TopoMon even describes the intermediate hops on a path from one machine to another machine of the Internet in order to capture shared network links: it does not provide a *scalable, logical* description of the network.

3.9 Miscellaneous Related Works

A few other works [8, 15] consider the topology of the Internet and represent it in a hierarchical manner with Wide-Area, Metropolitan-Area and Local-Area Networks (WANs, MANs, LANs). A computational grid cannot be described like the Internet, because it also includes dedicated, high-performance, specific (potentially non-IP-based) networks such as Myrinet clusters.

Topology-d [33] computes logical network topologies automatically, but does not support firewalls and it is restricted to IP-based networks.

4 Our Proposed Grid Network Description Model

4.1 Logical Topology and Network Grouping

Since we aim to use our grid network description in order to select nodes before mapping an application to the resources of a grid, we need a functional description of the network. This goal is achieved by representing a *logical* network topology and by *grouping* together the computers with common network characteristics.

As most related works, we describe a *logical* network topology, in contrast with the *physical* network topology. That consists in not representing all the physical network connections. For instance, Figures 1, 2, 3, 4 model the Internet connection using just *one logical link*, while *multiple physical paths* may interconnect different Internet domains.

To serve our purpose of application deployment, the network topology description does not need to be aware of any switch or router, neither does it need to represent every single network link. This assertion contradicts the assumptions made by TopoMon (see Subsection 3.8) which claims that representing shared network links is essential for a good grid network topology description. First, the effect of shared links is that the communication performance can decrease over certain network paths at certain time periods: rather than including this effect in the network topology, we choose a more functional network description by including the effect of shared links in the numerical network properties, such as jitter, bandwith variance (in time) with respect to its average value, *etc.* all the more so as our description model accepts dynamic values. Second, network congestion may not come from the shared link on a network path, which is usually a backbone link, but it may stem from a lower performance connection of an institution to the Internet backbone. Third, accurate topological knowledge about shared network links might reveal useless because traffic on the shared links of the Internet's backbone is totally unpredictable, and we have no control over it.

All the grid network topology description needs to include is the fact that a certain set of computers are connected together over the same sub-network, and that those computers have roughly the same communication characteristics while communicating with each other. So the computers are registered to network groups, depending on how many sub-networks they belong to. Thus, the common communication capabilities (end-to-end

bandwidth, latency, loss) of the computers belonging to the same sub-network are entered only once as attributes of the network group, as well as the software available to access particular network technologies (BIP, GM or MX for a Myrinet network, for instance).

4.2 Benefits of Network Grouping

Network grouping makes the node selection phase easier because it supplies a *synthetic* view of the network topology. Intuitively, grouping replaces a "join" (in the database language) for free, since the information about the network is already "pre-compiled". As we do not describe each individual link, our description is more compact. We also claim that network grouping makes sense because end-to-end network performance properties are roughly the same between any two computers of a sub-network like a dedicated cluster or a Local-Area Network (see Subsection 4.4 for exceptions): the network performance characteristics can be described as attributes of the network groups using the results of the NMWG from the GGF (see Subsection 3.3). Finally, a network group may be mapped to a multicast group, specifying in just one place that all hosts and/or sub-networks belonging to the network group can communicate together by using multicast messages.

4.3 Graph of Network Groups

We describe the topology of a grid using a directed acyclic graph (DAG). The nodes of our grid network description graph correspond to the network groups introduced in Subsection 4.1 or to the computers (which can be considered as network groups made of just one host).

The oriented edges of our description graph correspond to network group inclusions: network groups can have parent or child network groups and the edges are oriented from a parent network group to a child network group. In other words, a child network group represents a sub-network of its parent network group.

For instance, Grid B shown in Figure 2 would be described with the graph of Figure 5. Myrinet in Cluster A, Gigabit in Cluster B and Internet are three independent network groups available to Grid B. Figure 5 also contains a particular node named "root node" whose goal is to solve the problem of non-hierarchical grids.

A non-hierarchical grid is a grid composed of independent networks. For example, Grid D of Subsection 2.4 contains three independent networks. To handle this sort of situation, a special node is always present in our proposed model: the "root node". This particular node represents the whole grid by listing all the independent networks which belong to the grid. The "root node" has no parent, and all its children are *independent* network groups: this organization reflects the non-hierarchical nature of a grid topology and permits to describe Grid D of Subsection 2.4 as shown in Figure 6. The leaves of the graph, defined as the graph nodes which have one or more parents but no children, correspond to the computers of the grid.

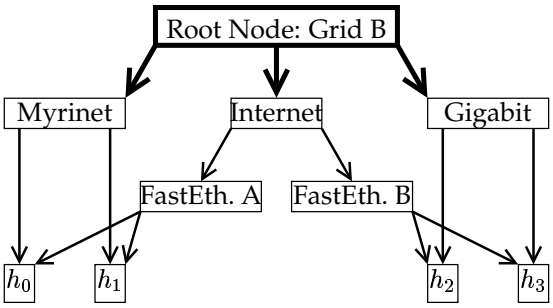

Figure 5: Network graph describing the topology of Grid B.

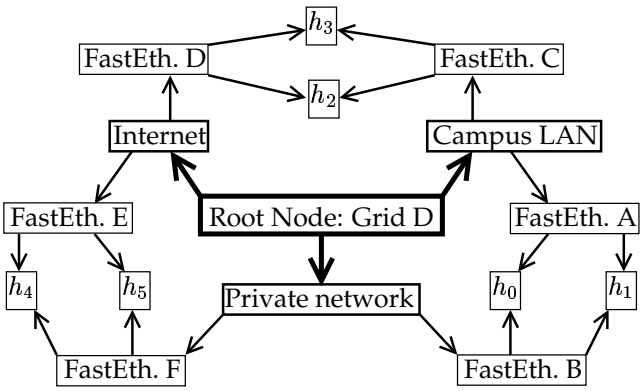

Figure 6: Network graph describing the topology of Grid D.

The properties of a network group (available software for network access, network performance characteristics) can be specified as attributes of the corresponding node of the graph. If a child network group does not define a property among the attributes of its node, then this property is inherited from its parent nodes. As shown on Figure 6, a child network node may have several parents, meaning that the child network group is included in several parent network groups. The network description graph must be consistent, preventing different properties from being inherited from the possibly various parents of a child node.

The *fundamental difference* between related works and our description graph is the nature of the objects which we map to the nodes and edges of the network description graph. Related works map the computers to the nodes of a graph and the network links to the edges of the graph, while our network grouping allows for another mapping where the

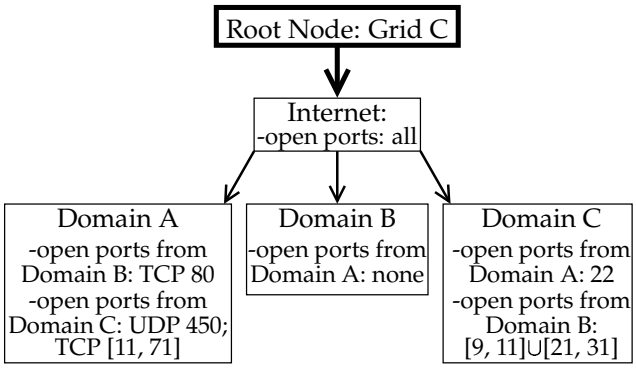

Figure 7: Network graph describing the topology of Grid C.

edges of the graph represent network group inclusions. Our mapping is more scalable because it yields less edges: several network links are implicitly gathered into a single network group.

4.4 Asymmetric Links, Firewalls and NAT

To support asymmetric network links (in terms of open or closed ports with firewalls or in terms of network performance properties, see Subsection 2.3), *exceptions* can be declared within a network group enumerating its child nodes. An exception overrides (possibly partially) the parent network groups' inherited properties. Those exceptions can even be network group specific, meaning that an exception may not concern all the sibling network groups, but only one or a few.

Grid C illustrated on Figure 3 is described by the graph on Figure 7: all three domains (A, B, C) are registered to the (parent) Internet network group. As the Internet network group declares that there are no firewalls restricting the communications (as a general rule), exceptions must be specified while registering the three child network groups. Domain B will mention that it accepts no network connections from Domain A, but it will not need to specify that it accepts all connections from Domain C because that is the general rule, inherited from the parent Internet network group.

To describe a set of computers subject to network address translation (NAT), a network group can include hosts with private IP addresses and hold a pointer to one or more NAT gateways (or relay hosts) which are responsible for network address translation and message forwarding.

5 Implementation

This section deals with practical issues. First, it shows that the proposed model is quite simple to integrate into an existing resource information service. Second, the implementation and the model are validated by registering examples of grids.

We chose to extend version 2 of MDS to describe grid network topologies following our model because MDS2 is widely deployed and applied, flexible and easily extensible. MDS2 also permits new network groups to register dynamically to the grid. However, our grid network description model is not bound to MDS2: we simply handle XML descriptors, which can be stored by any means, including MDS, description files accessed through the HTTP protocol (possibly HTTPS), *etc.* We could also have expressed our description by using an extension of RDF (Resource Description Framework, [21]), an XML dialect used to describe the metadata information about the resources of a grid.

5.1 Network Group Manager

The information concerning a network group is held by a computer called the "network group manager" chosen by a network group administrator. This manager must be accessible (*i.e.*, with no firewall obstructing communications) from the application deployment client which needs the information held by the manager to select resources.

Updating and maintaining the network topology and performance properties is easier as information is gathered in one point (thanks to network grouping) for a whole network group rather than scattered on each network link description. More than one manager may be chosen for fault tolerance reasons, but that would require to maintain consistency among the information held by the all the managers of a network group.

5.2 MDS2-only Implementation

In an earlier implementation, the network topology description graph was mapped to a hierarchical MDS: the child network groups maintained a list of parent nodes which they were registered to, and parent network groups also maintained a list of their child nodes. MDS2 already provides the list of child nodes registered to a parent node, using `grid-info-search -s base giisregistrationstatus`. We just needed to add a simple Information Provider script into MDS2 to find out the list of the parent nodes by reading an already existing MDS2 configuration file, namely `etc/grid-info-resource-registration.conf`.

We described the new data we needed to add into MDS2.4 by defining LDAP schema entries: around 40 LDAP schema entries were added to describe the network topology. The extended version of MDS2 remained *inter-operable* with the original MDS2.

5.3 Implementation Indenpendent of Grid Technology

The earlier implementation required consistency to be maintained among the information held by the parent and child network groups, since there was a double linkage from the parent groups referencing their children *and* from the child groups referencing their parents. So we moved to another implementation, which is not bound to MDS2 any longer. The latter implementation does not map the network topology description graph to a hierarchical MDS: the description graph is made of a set of distributed XML descriptors, possibly hosted by a *single binary LDAP* entry in MDS2. By doing so, we lose the filtering capabilities of LDAP, since all the information held by an MDS server is contained in an XML file which an LDAP server cannot parse. However, we make it easier for a future transition to Globus3/OGSI or Globus4/Web Services, and we find it more convenient to handle XML rather than LDAP format, especially with the powerful XML filtering capabilities of Xpath or Xquery.

As shown on Figure 8, the XML descriptor essentially holds two kinds of elements: *grid nodes* and *GIS references* (grid information service). *Grid nodes* correspond to *network groups*: they can be structural or represent a computational resource, accessed using a Globus2 gatekeeper for instance, but other job submission methods are also possible (like SSH). A structural grid node is simply a list of child network groups: those child grid nodes may be described in the same XML file or in an XML descriptor which can be obtained using a GIS reference. A *GIS reference* is a method for retrieving an XML descriptor, using either MDS2 or HTTP, *etc.*

In the XML descriptor, the parents have pointers to their child network groups. The program which reads this information and parses the XML data maintains a double linkage from the parent to the child grid nodes, and from the children to the parent network groups. This double linkage allows to find out whether and how two computers can communicate with each other (we just need to scan all their common parent network groups) and to retrieve all the computers which belong to a network group (using the list of child grid nodes).

A network topology XML descriptor can be retrieved from MDS2, from a local file, from HTTP or HTTPS protocols. Future extensions may allow more XML information sources. Currently, an XML descriptor may contain the description of one or several grid nodes, and it may also point to other, distributed XML information sources (HTTP reference, MDS2 pointer, *etc.*) Figure 8 shows a partial description of Grid B (see Subsection 2.2). Two grid nodes are defined in this XML file:

- "root_grid_B" is a structural network group which has three child network groups, "myrinet", "internet", "gigabit";
- "myrinet" is a grid node which represents a computational resource with 128 CPUs and a Globus2 gatekeeper for job submission.

The child network nodes "internet" and "gigabit" are described in other XML descriptors, which can be obtained respectively through MDS2 and HTTP, as specified by the two GIS

```
<network_description>

  <grid_node id="root_grid_B" type="struct">
    <child_node id="myrinet"/>
    <child_node id="internet" gis="inet_mds2"/>
    <child_node id="gigabit" gis="Gbit_http"/>
  </grid_node>

  <grid_node id="myrinet" type="globus2">
    <node_count>64</node_count>
    <cpu_count>128</cpu_count>
    <gatekeeper_host>a.b.c.d</gatekeeper_host>
    <gatekeeper_port>2119</gatekeeper_port>
  </grid_node>

  <gis id="inet_mds2" type="globus_mds_2">
    <mds_host>paracisrv.irisa.fr</mds_host>
    <mds_port>2135</mds_port>
  </gis>

  <gis id="Gbit_http" type="http">
    <url>http://www.irisa.fr/Gbit_netw.xml</url>
  </gis>

</network_description>
```

Figure 8: XML partial description of Grid B.

XML elements. This allows the deployment planner (or resource selection algorithm) to get information on more resources if needed (upon request) instead of having all information about all the grid resources right from the start. For instance, we have implemented a simple, round-robin deployment planner which allocates processes to be deployed to the first grid resources discovered: only the needed resources are queried, thus avoiding to handle a huge mass of data.

Figure 9 shows an excerpt from the XML descriptor of Grid C (see Figures 3 and 7): it illustrates how attributes can be attached to grid nodes as well as the mechanism to describe exceptions within a network group (see Subsection 4.4). This XML file partially describes the grid node corresponding to network group "Internet". This group has three children: "Domain_A" through "Domain_C". The first "network_properties" XML element specifies the general properties of the "Internet" network group since no "source" or "destination" attribute is specified for this XML element: all TCP and UDP ports are open. The second "network_properties" XML element means that all TCP and UDP communications from "Domain_A" are rejected by "Domain_B". The third "network_properties" XML element restricts the possible communications from "Domain_C" to "Domain_A", opening only UDP port 450 and TCP ports ranging from 11 to 71. The last "network_properties" XML element specifies that only the union of ports $[9, 11] \cup [21, 31]$ is open for TCP.

```
<network_description>

  <grid_node id="Internet" type="struct">

    <child_node id="Domain_A"/>
    <child_node id="Domain_B"/>
    <child_node id="Domain_C"/>

    <network_properties udp_port_policy="allopen" tcp_port_policy="allopen"/>

    <network_properties source="Domain_A" destination="Domain_B"
                        udp_port_policy="allclosed" tcp_port_policy="allclosed"/>

    <network_properties source="Domain_C" destination="Domain_A"
                        udp_port_policy="allclosed" tcp_port_policy="allclosed">
      <udp_ports>
        <min_port>450</min_port>
        <max_port>450</max_port>
      </udp_ports>
      <tcp_ports>
        <min_port>11</min_port>
        <max_port>71</max_port>
      </tcp_ports>
    </network_properties>

    <network_properties source="Domain_B" destination="Domain_C"
                        tcp_port_policy="allclosed">
      <tcp_ports>
        <min_port>9</min_port>
        <max_port>11</max_port>
      </tcp_ports>
      <tcp_ports>
        <min_port>21</min_port>
        <max_port>31</max_port>
      </tcp_ports>
    </network_properties>

  </grid_node>

</network_description>
```

Figure 9: XML partial description of Grid C.

Our network topology description model could host the description of any mix of Grid A, Grid B, Grid C, Grid D (see Section 2) we could imagine.

6 Conclusion and Future Work

The actual, efficient utilization of grids depends on an effective deployment of applications on grid resources. One major source of difficulty stems from the *heterogeneity of the resources* including compute nodes and networks. While the description of the *compute nodes* of a grid is relatively well mastered, the *network description* of grids is not yet suitable for constrained deployments of applications on grids.

This paper has presented a description model for grid networks. This model provides a *synthetic* view of the network topology. The model which we propose is also *simple*, namely thanks to the possibility for a network group to inherit properties from its parent network groups. However, this simplicity does not hinder the description of *complex* network topologies (asymmetric links, firewalls, non-IP networks, non-hierarchical topologies). Finally, our description model aims to be *complete* by including the necessary information about the software available to access particular network technologies and allowing for specification of network performance properties.

This model is useful not only to deploy applications (resource selection, see [39]), but also to schedule algorithms by making proper decisions [10, 9, 43] as well as for grid generators designed for simulation [29].

Our next step is to focus on a model for automatic deployment of component-based applications on a computational grid [24]. The software component model [41, 4, 34] which advocates a programming model based on the composition of (reusable and) *independent units of deployment* emphasizes the deployment phase as a separate step, independent of the programming phase. To this end, an adequate network topology model is required. However, we will probably need to extend the component assembly description as network constraints are generally not taken into account.

References

[1] Gabrielle Allen, Kelly Davis, Konstantinos N. Dolkas, Nikolaos D. Doulamis, Tom Goodale, Thilo Kielmann, André Merzky, Jarek Nabrzyski, Juliusz Pukacki, Thomas Radke, Michael Russell, Ed Seidel, John Shalf, and Ian Taylor. Enabling applications on the grid: a GridLab overview. *International Journal of High Performance Computing Applications (JHPCA), special issue on Grid Computing: Infrastructure and Applications*, 17(4), August 2003.

[2] Giovanni Aloisio, Massimo Cafaro, Italo Epicoco, and Sandro Fiore. Analysis of the Globus Toolkit grid information service. Technical Report GridLab-10-D.1-0001-1.0,

HPCC, University of Lecce, Italy, 2002. available at `http://www.gridlab.org/Resources/Deliverables/D10.1.pdf`.

[3] Giovanni Aloisio, Massimo Cafaro, Italo Epicoco, Daniele Lezzi, Maria Mirto, Silvia Mocavero, and Serena Pati. First GridLabMDS release. Technical Report GridLab-10-D.3-0001-1.0, HPCC, University of Lecce, Italy, 2002. available at `http://www.gridlab.org/Resources/Deliverables/D10.3c.pdf`.

[4] Rob Armstrong, Dennis Gannon, Al Geist, Katarzyna Keahey, Scott Kohn, Lois McInnes, Steve Parker, and Brent Smolinski. Toward a common component architecture for high-performance scientific computing. In *Proc. of the 8th IEEE International Symp. on High Performance Distributed Computing (HPDC'99)*, pages 13–22, Redondo Beach, CA, August 1999.

[5] Nanette J. Boden, Danny Cohen, Robert E. Felderman, Alan E. Kulawik, Charles L. Seitz, Jakov N. Seizovic, and Wen-King Su. Myrinet: A gigabit-per-second local area network. *IEEE Micro*, 15(1):29–36, February 1995.

[6] Yuri Breitbart, Minos N. Garofalakis, Cliff Martin, Rajeev Rastogi, S. Seshadri, and Abraham Silberschatz. Topology discovery in heterogeneous IP networks. In *Proc. of IEEE INFOCOM'2000*, pages 265–274, Tel-Aviv, Israel, March 2000.

[7] Matthias Brune, Alexander Reinefeld, and Jörg Varnholt. A resource description environment for distributed computing systems. In *Proc. of the 8th IEEE International Symp. on High Performance Distributed Computing (HPDC'99)*, pages 279–286, Redondo Beach, CA, August 1999.

[8] Kenneth L. Calvert, Matthew B. Doar, and Ellen W. Zegura. Modeling internet topology. *IEEE Communications Magazine*, 35(6):160–163, June 1997.

[9] Henri Casanova and Fran Berman. *Grid Computing: Making the Global Infrastructure a Reality*, chapter 33 (Parameter Sweeps on the Grid with APST). Wiley & Sons, April 2003.

[10] Henri Casanova, Graziano Obertelli, Francine Berman, and Richard Wolski. The AppLeS parameter sweep template: User-level middleware for the grid. In *Proc. of Supercomputing 2000*, pages 75–76, Dallas, TX, November 2000.

[11] R. L. Cottrell, Connie Logg, and I-Heng Mei. Experiences and results from a new high performance network and application monitoring toolkit. In *Proc. of the Passive and Active Measurement Workshop (PAM2003)*, pages 205–217, La Jolla, CA, April 2003.

[12] Karl Czajkowski, Steven Fitzgerald, Ian Foster, and Carl Kesselman. Grid information services for distributed resource sharing. In *Proc. of the 10th IEEE International Symp. on High-Performance Distributed Computing (HPDC-10'01)*, pages 181–194, San Francisco, California, August 2001. IEEE Computer Society.

[13] Mathijs den Burger, Thilo Kielmann, and Henri E. Bal. TopoMon: A monitoring tool for grid network topology. In *Proc. of the International Conf. on Computational Science, part 2 (ICCS2002)*, number 2330 in LNCS, pages 558–567, Amsterdam, The Netherlands, April 2002. Springer.

[14] Peter Dinda and Beth Plale. A unified relational approach to grid information services. Informational Draft GWD-GIS-012-1, Global Grid Forum, February 2001.

[15] Matthew B. Doar. A better model for generating test networks. In *Proc. of the IEEE Global Telecommunications Conference / Globecom'96*, London, UK, November 1996.

[16] Nathalie Furmento, Anthony Mayer, Stephen McGough, Steven Newhouse, Tony Field, and John Darlington. Optimisation of component-based applications within a grid environment. In *Proc. of the 2001 ACM/IEEE Conf. on Supercomputing*, page 30, Denver, CO, November 2001. ACM Press, New York, NY, USA.

[17] Nathalie Furmento, Anthony Mayer, Stephen McGough, Steven Newhouse, Tony Field, and John Darlington. ICENI: Optimisation of component applications within a grid environment. *Journal of Parallel Computing*, 28(12):1753–1772, 2002.

[18] The Globus Alliance: `http://www.globus.org/`.

[19] Web site of the ENV project at SDSC, CA: `http://grail.sdsc.edu/projects/env/GridML.html`.

[20] The Grid Physics Network (GriPhyN) web site: `http://www.GriPhyN.org/`.

[21] Dan Gunter and Keith Jackson. The applicability of RDF-schema as a syntax for describing grid resource metadata. Informational Draft GWD-GIS-020-1, Global Grid Forum (GGF), June 2001.

[22] Axel Keller and Alexander Reinefeld. Anatomy of a resource management system for HPC clusters. Technical Report 00-38, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Germany, November 2000.

[23] Tatiana Kichkaylo, Anca-Andreea Ivan, and Vijay Karamcheti. Constrained component deployment in wide-area networks using AI planning techniques. In *Proc. of the 17th International Parallel and Distributed Processing Symp. (IPDPS'2003)*, page 3, Nice, France, April 2003.

[24] Sébastien Lacour, Christian Pérez, and Thierry Priol. Deploying CORBA components on a computational grid: General principles and early experiments using the Globus Toolkit. In Wolfgang Emmerich and Alexander L. Wolf, editors, *Proc. of the 2nd International Working Conference on Component Deployment (CD 2004)*, volume 3083 of *LNCS*, pages 35–49, Edinburgh, Scotland, UK, May 2004. Springer-Verlag. Held in conjunction with the 26th International Conference on Software Engineering (ICSE 2004).

[25] Chuang Liu, Lingyun Yang, Ian Foster, and Dave Angulo. Design and evaluation of a resource selection framework for grid applications. In *Proc. of the 11th IEEE Symp. on High Performance Distributed Computing (HPDC-11)*, pages 63–52, Edinburgh, Scotland, July 2002.

[26] Bruce Lowekamp, Brian Tierney, Les Cottrell, Richard Hughes-Jones, Thilo Kielmann, and Martin Swany. A hierarchy of network performance characteristics for grid applications and services. Proposed Recommendation Global Grid Forum (GGF), Network Measurement Working Group (NMWG), January 2004.

[27] Bruce B. Lowekamp, Nancy Miller, Dean Sutherland, Thomas Gross, Peter Steenkiste, and Jaspal Subhlok. A resource query interface for network-aware applications. *Journal of Cluster Computing*, 2(2):139–151, 1999.

[28] Bruce B. Lowekamp, Brian Tierney, Les Cottrell, Richard Hughes-Jones, Thilo Kielmann, and Martin Swany. Enabling network measurement portability through a hierarchy of characteristics. In *Proc. of the 4th International Workshop on Grid Computing (Grid2003)*, pages 68–75, Phoenix, AZ, November 2003. IEEE.

[29] Dong Lu and Peter A. Dinda. Synthesizing realistic computational grids. In *Proc. of SuperComputing 2003 (SC'03)*, Phoenix, AZ, November 2003.

[30] Muthucumaru Maheswaran and Howard Jay Siegel. A dynamic matching and scheduling algorithm for heterogeneous computing systems. In *Proc. of the 7th Heterogeneous Computing Workshop, held in conjunction with IPPS/SPDP'98*, pages 57–69, Orlando, FL, March 1998.

[31] Myricom. *GM: A Message-Passing System for Myrinet Networks*. reference manual, available at `http://www.myri.com/scs/GM-2/doc/html/`.

[32] Myricom. *Myrinet Express (MX): A High-Performance, Low-Level, Message-Passing Interface for Myrinet*, July 2003. pre-release, available at `http://www.myri.com/scs/#documentation`.

[33] Katia Obraczka and Grig Gheorghiu. The performance of a service for network-aware applications. In *Proc. of the 2nd ACM Symp. on Parallel and Distributed Tools (SPDT'98)*, pages 81–91, Welches, Oregon, USA, August 1998.

[34] Christian Pérez, Thierry Priol, and André Ribes. A parallel CORBA component model for numerical code coupling. *The International Journal of High Performance Computing Applications (IJHPCA)*, 17(4):417–429, 2003. Special issue Best Applications Papers from the 3rd International Workshop on Grid Computing.

[35] Loïc Prylli and Bernard Tourancheau. BIP: a new protocol designed for high performance networking on Myrinet. In *Proc. of the 1st Workshop on Personal Computer based Networks Of Workstations (PC-NOW'98)*, Lecture Notes in Computer Science, pages 472–485. Springer-Verlag, April 1998. Held in conjunction with IPPS/SPDP 1998.

[36] Rajesh Raman, Miron Livny, and Marvin Solomon. Matchmaking: Distributed resource management for high throughput computing. In *Proc. of the 7th IEEE International Symp. on High Performance Distributed Computing (HPDC7)*, pages 140–146, Chicago, IL, July 1998.

[37] Gary Shao, Fran Berman, and Rich Wolski. Using effective network views to promote distributed application performance. In *Proc. of the 1999 International Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)*, June 1999.

[38] Pyda Srisuresh and Matt Holdrege. IP network address translator (NAT) terminology and considerations. Informational RFC 2663, IETF Network Working Group, August 1999.

[39] Jaspal Subhlok, Peter Lieu, and Bruce Lowekamp. Automatic node selection for high performance applications on networks. In *Proc. of the 7th ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming (PPoPP'99)*, pages 163–172, Atlanta, GA, May 1999.

[40] Martin Swany and Rich Wolski. Representing dynamic performance information in grid environments with the network weather service. In *Proc. of the 2nd IEEE/ACM International Symp. on Cluster Computing and the Grid (CCGrid'02)*, pages 48–56, Berlin, Germany, May 2002.

[41] Clemens Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley / ACM Press, first edition, 1998.

[42] TeraGrid web site: `http://www.TeraGrid.org/`.

[43] Jon B. Weissman and Xin Zhao. Scheduling parallel applications in distributed networks. *Journal of Cluster Computing*, 1(1):109–118, May 1998.

[44] Rich Wolski, Neil Spring, and Jim Hayes. The Network Weather Service: a distributed resource performance forecasting service for metacomputing. *Journal of Future Generation Computing Systems*, 15(5-6):757–768, October 1999.

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399