

# A Network Topology Description Model for Grid Application Deployment

Sébastien Lacour, Christian Pérez, Thierry Priol

IRISA / INRIA, Campus de Beaulieu, 35042 Rennes Cedex, France

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

## 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 functional 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).*

## 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 [17]. 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.

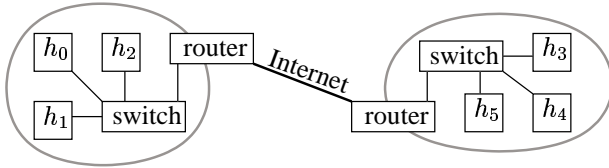
Accurate information about compute nodes, installed software, network connectivity and performance properties is required [28] for a pertinent deployment. Previous works succeed in describing the compute nodes properly (CPU speed, memory size, operating system, *etc.*), but generally fail to describe the network topology and its characteristics in a simple, functional, 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 (GridPhyN [15], or TeraGrid [31]), as well as any grid which can be devised. For example, the model should include network connectivity information and it should 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 description model of grid network topology is introduced in Section 4 and an implementation which extends the MDS module of the Globus Toolkit is presented in Section 5. Section 6 concludes the paper and details our perspectives.

## 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 computer-level and network-level requirements 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 100 ms of latency”. 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 [12], Sekitei [16], or Condor's matchmaking [25].



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

To satisfy the requirements mentioned above, the network must be described *accurately* 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 requirements are made of both network constraints and computer constraints.

### 2.1. Simple Grids

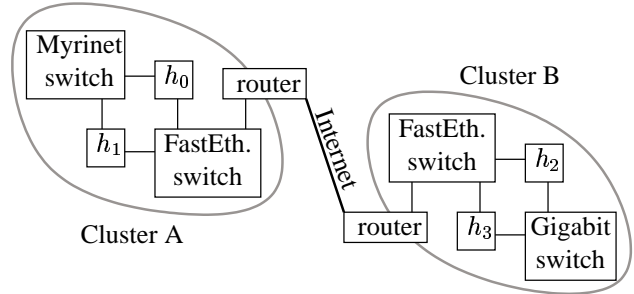
Figure 1 shows the physical description of a simple grid (Grid A), made of two Fast-Ethernet clusters connected together through the Internet. Each cluster is made of three hosts ( $h_i$ ). The description of such a grid should be as simple as the grid itself.

### 2.2. Connection to Various Networks

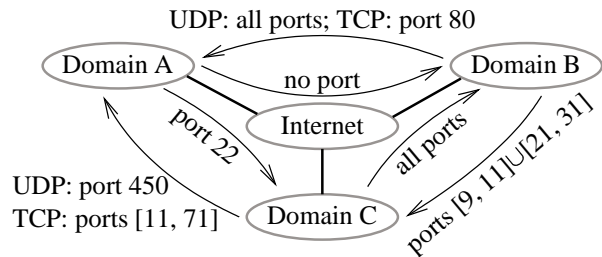
Figure 2 shows the physical description of a grid made of two clusters (Grid B). Both clusters are equipped with two different networks. Cluster A has Fast-Ethernet and a Myrinet network; Cluster B has Fast-Ethernet and a Gigabit Ethernet network. Note that the Gigabit Ethernet cards have IPs which cannot 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 [22], BIP [24] or MX [23]. This case also encompasses clusters made of nodes with private IPs (*i.e.*, locally valid IPs), should they resort to network address translation (NAT, [27]) 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 of-



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

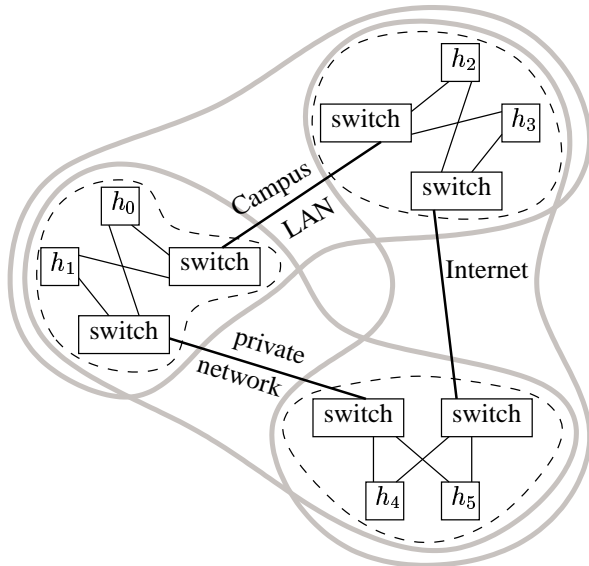


**Figure 3. Grid C: firewalls limiting network traffic selectively.**

ten protected against intrusion by firewalls, closing certain TCP/UDP ports for network traffic coming from certain domains, or resorting to NAT (Network Address Translation [27]). 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.

### 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.



**Figure 4. Grid D: non-hierarchical network.**

## 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 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.

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. 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 combined information is usually needed during the deployment phase [28], 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.

MDS (Monitoring and Discovery Service, [9]) is the Grid Information Service (GIS) of the Globus Toolkit version 2 [13]. MDS allows for flexible, distributed, 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, [29]). MDS describes the compute nodes of a grid conveniently and accurately (CPU speed and number, memory size, disk space, operating system type, *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.

RSD (Resource Service Description, [6]) is a software architecture for specifying resources and services in complex heterogeneous computing environments. RSD allows for dynamically changing performance data of the 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 not well adapted to represent Grid D presented in Section 2.4. RSD does not either take firewalls into account, and its description of each physical link may not be scalable for a grid.

The Network Measurement Working Group (NMWG) of the Global Grid Forum (GGF) provides a sound way of describing network performance characteristics [20], but their work does not focus on network topology description. The proposed recommendation [18] shows interesting results on the classification of network characteristics and measurement methodologies, but it does not specify anything precise about network topology description and its scalability in a grid environment. However, our network topology description model and the performance characteristics description proposed by the NMWG are complementary.

Remos (REsource MONitoring System, [19]) 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.

GridLab [1] relies on Globus/MDS 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.

ENV (Effective Network Views, [26]) uses an XML dialect (GridML, [14]) 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 topologies only, it does not take firewalls into account and it is restricted to IP networks.

GridG [21] 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 and does not take firewalls into account.

RGIS [11] 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 requirements as those presented in Section 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.

TopoMon [10] is a monitoring tool for grid networks which augments NWS [29] 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 pair-wise basis, including 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.

## 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 amounts to 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 Section 3) which claims that representing shared network links is essential for a good description of grid network topology. Indeed, the effect of shared links is that 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, bandwidth variance (in time), *etc.* all the more so as our description model accepts dynamic values.

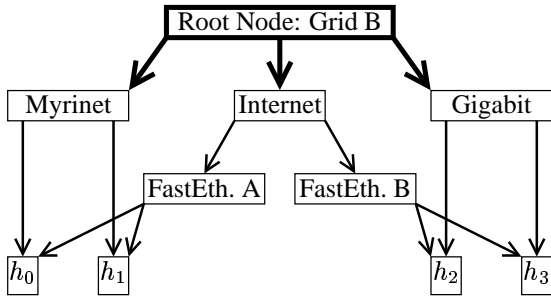
All the grid network topology description needs to include is the fact that a certain set of computers are connected together over the same sub-network, and that those computers have roughly the same communication characteristics while communicating with each other. So the computers are registered to network groups, depending on how many sub-networks they belong to. Thus, the common communication capabilities (end-to-end bandwidth, latency) 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 *functional* view of 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. Network grouping also 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 LAN (see Section 4.5 for exceptions): network performance characteristics can be described as attributes of network groups using the results of the NMWG from the GGF (see Section 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 de-



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

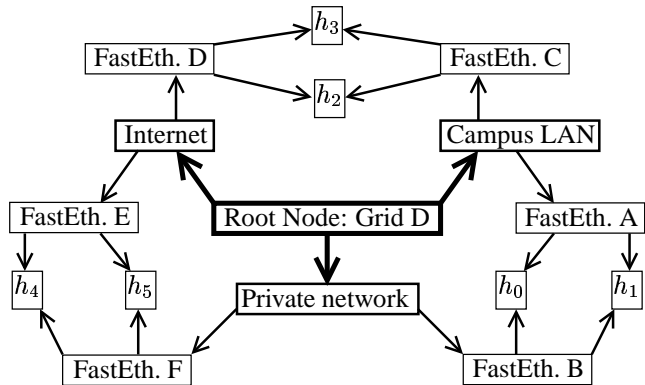
scription graph correspond to the network groups introduced in Section 4.1 or to the computers (which can be considered as network groups made of just one host).

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 could 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 allow for the description of non-hierarchical grids.

A non-hierarchical grid is a grid composed of independent networks. For example, Grid D of Section 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 Section 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 compute nodes of the grid. By “compute node”, we mean a single PC, a supercomputer, the front-end computer of a cluster with a PBS server or a Globus2 gatekeeper accepting job submissions, etc.

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 edges of the graph represent network group inclusions. Our



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

mapping is more scalable because it yields less edges: several network links are implicitly gathered into a single network group.

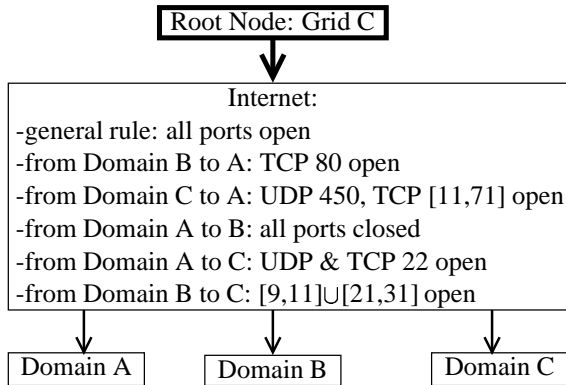
#### 4.4. Network Properties

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 (like a transport protocol or a list of closed IP ports), then this property is inherited from its parent nodes. To avoid inconsistencies, a child node cannot redefine a property which has already been defined by one of its parents.

As shown on Figure 6, a child 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. In case of ill-formed network descriptions, there might be conflicting values for an inherited property: the software exploiting the network information can detect this situation, and it should signal the error and cancel all conflicting values for the property.

#### 4.5. 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 Section 2.3), *exceptions* can be declared within a network group enumerating its child nodes. Exceptions override (possibly partially) the general network properties defined for the whole group. Those exceptions are network group specific, meaning that they do not con-



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

cern all the sibling network groups, but only one or a few. To avoid inconsistencies, all exceptions are gathered in one point, *i.e.* within the parent network group definition, and property values are overridden by exceptions, not merged.

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 accepts no network connections from Domain A, but it is useless to specify that it accepts all connections from Domain C because that is the general rule within the Internet network group. All those exceptions are specified in one place, *i.e.* within the definition of a 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.

#### 4.6. Designing a Network Description Graph

This section discusses the design of network description graphs. 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 a monitoring tool (see Section 3 and [5, 8]).

For a given grid infrastructure, the network description graph is not unique. However, the shape of a graph should reflect the physical organization of the network as closely as possible: network groups with common property values (open ports to/from the Internet, *etc.*) should be gathered within a common parent network group.

For instance, the graph of Figure 5 is not the only way to describe Grid B. The network groups FastEth. A or B could have been declared as direct children of the Root Node. But this would not have been a good idea, since they share common properties (IP network, open ports, *etc.*) with the Internet node. Declaring those common properties in one place yields better scalability and easier maintenance. However, the Myrinet network node cannot be a child of the Internet node, because Internet is an IP network while Myrinet is not, and a child node cannot redefine inherited network properties. Likewise, the Gigabit network group cannot be a child of the Internet node because this network cannot route messages to/from other hosts of the Internet ( $h_0, h_1$ ).

## 5. Implementation

This section shows that the proposed model is quite simple to integrate into an existing resource information service and presents an example of grid description.

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. 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(S), (GSI)FTP protocols, *etc.*

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 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.

In our implementation, 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/WS-RF 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, HTTP(S) or (GSI)FTP, 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).

Figure 8 shows a partial description of Grid B (see Section 2.2). Two grid nodes are defined in this XML file: first, “root\_grid\_B” is a structural network group which has three child network groups, “myrinet”, “internet”, “gigabit”; second, “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 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, avoiding to handle a huge mass of data.

## 6. Conclusion and Future Work

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

This paper has presented a description model for grid networks. This model provides a *functional* 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 par-

```
<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.**

ticular network technologies and allowing for specification of network performance properties.

This model is useful not only to deploy applications (resource selection, see [28]), but also to schedule algorithms by making proper decisions [7, 32] as well as for grid generators designed for simulation [21].

Our next step is to focus on a model for automatic deployment of component-based applications on a computational grid [17]. The software component model [30, 4] 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 description is required. However, we will probably need to extend the component assembly description as network constraints are not expressed in general.

## References

- [1] G. Allen, K. Davis, K. N. Dolkas, N. D. Doulamis, T. Goodale, T. Kielmann, A. Merzky, J. Nabrzyski, J. Pukacki, T. Radke, M. Russell, E. Seidel, J. Shalf, and I. 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), Aug. 2003.
- [2] G. Aloisio, M. Cafaro, I. Epicoco, and S. Fiore. Analysis of the Globus Toolkit grid information service. Technical Report GridLab-10-D.1-0001-1.0, HPCC, University of Lecce, Italy, 2002.

- [3] G. Aloisio, M. Cafaro, I. Epicoco, D. Lezzi, M. Mirto, S. Mocavero, and S. Pati. First GridLabMDS release. Technical Report GridLab-10-D.3-0001-1.0, HPCC, University of Lecce, Italy, 2002.
- [4] R. Armstrong, D. Gannon, A. Geist, K. Keahey, S. Kohn, L. McInnes, S. Parker, and B. 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, Aug. 1999.
- [5] Y. Breitbart, M. N. Garofalakis, C. Martin, R. Rastogi, S. Shadri, and A. Silberschatz. Topology discovery in heterogeneous IP networks. In *Proc. of IEEE INFOCOM'2000*, pages 265–274, Tel-Aviv, Israel, Mar. 2000.
- [6] M. Brune, A. Reinefeld, and J. 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, Aug. 1999.
- [7] H. Casanova, G. Obertelli, F. Berman, and R. Wolski. The AppLeS parameter sweep template: User-level middleware for the grid. In *Proc. of Supercomputing 2000*, pages 75–76, Dallas, TX, Nov. 2000.
- [8] R. L. Cottrell, C. Logg, and I.-H. 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, Apr. 2003.
- [9] K. Czajkowski, S. Fitzgerald, I. Foster, and C. 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, Aug. 2001.
- [10] M. den Burger, T. Kielmann, and H. 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, Apr. 2002. Springer.
- [11] P. Dinda and B. Plale. A unified relational approach to grid information services. Informational Draft GWD-GIS-012-1, Global Grid Forum, Feb. 2001.
- [12] N. Furmento, A. Mayer, S. McGough, S. Newhouse, T. Field, and J. Darlington. ICENI: Optimisation of component applications within a grid environment. *Journal of Parallel Computing*, 28(12):1753–1772, 2002.
- [13] The Globus Alliance: <http://www.globus.org/>.
- [14] Web site of the ENV project at SDSC, CA: <http://grail.sdsc.edu/projects/env/GridML.html>.
- [15] The GriPhyN web site: <http://www.GriPhyN.org/>.
- [16] T. Kichkaylo, A.-A. Ivan, and V. 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, Apr. 2003.
- [17] S. Lacour, C. Pérez, and T. Priol. Deploying CORBA components on a computational grid: General principles and early experiments using the Globus Toolkit. In *Proc. of the 2nd International Working Conf. on Component Deployment (CD 2004)*, volume 3083 of LNCS, pages 35–49, Edinburgh, Scotland, UK, May 2004. Springer-Verlag.
- [18] B. Lowekamp, B. Tierney, L. Cottrell, R. Hughes-Jones, T. Kielmann, and M. Swany. A hierarchy of network performance characteristics for grid applications and services. Proposed Recommendation Global Grid Forum (GGF), Network Measurement Working Group (NMWG), Jan. 2004.
- [19] B. B. Lowekamp, N. Miller, D. Sutherland, T. Gross, P. Steenkiste, and J. Subhlok. A resource query interface for network-aware applications. *Journal of Cluster Computing*, 2(2):139–151, 1999.
- [20] B. B. Lowekamp, B. Tierney, L. Cottrell, R. Hughes-Jones, T. Kielmann, and M. 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, Nov. 2003. IEEE.
- [21] D. Lu and P. A. Dinda. Synthesizing realistic computational grids. In *Proc. of SuperComputing 2003 (SC'03)*, Phoenix, AZ, Nov. 2003.
- [22] Myricom. *GM: A Message-Passing System for Myrinet Networks*. reference manual, available at <http://www.myri.com/scs/GM-2/doc/html/>.
- [23] 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>.
- [24] L. Prylli and B. 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, Apr. 1998. Held in conjunction with IPPS/SPDP 1998.
- [25] R. Raman, M. Livny, and M. 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.
- [26] G. Shao, F. Berman, and R. 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 (PDP'99)*, June 1999.
- [27] P. Srisuresh and M. Holdrege. IP network address translator (NAT) terminology and considerations. Informational RFC 2663, IETF Network Working Group, Aug. 1999.
- [28] J. Subhlok, P. Lieu, and B. 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.
- [29] M. Swany and R. 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 (CC-Grid'02)*, pages 48–56, Berlin, Germany, May 2002.
- [30] C. Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley / ACM Press, first edition, 1998.
- [31] TeraGrid web site: <http://www.TeraGrid.org/>.
- [32] J. B. Weissman and X. Zhao. Scheduling parallel applications in distributed networks. *Journal of Cluster Computing*, 1(1):109–118, May 1998.