The WISENT Grid Architecture: Coping with Firewalls and NAT

Guido Scherp[1], Wilhelm Hasselbring[2], and Jan Ploski[1]

[1] Business Information Management, OFFIS Institute for Information Technology,
Escherweg 2, 26121 Oldenburg, Germany
{guido.scherp|jan.ploski}@offis.de

[2] Software Engineering Group, University of Oldenburg, 26111 Oldenburg, Germany
hasselbring@informatik.uni-oldenburg.de

Abstract

In energy meteorology research, scientists from several domains such as physics, meteorology and electrical engineering work together to obtain information needed to characterize energy production from regenerative energy sources such as wind and solar power. For this purpose, several scientific applications were developed to process large data sets from heterogenous data sources in complex and sometimes long-running process chains. In our project WISENT a Grid infrastructure is created to speed up execution of these applications and to ease access to computational and data resources. To achieve this goal, Grid software such as Globus Toolkit and Condor is employed to connect the existing resources of each project partner. But this ongoing process is hindered by blocking firewalls due to strong security policies and by the use of network address translation (NAT). In this paper we describe the current Grid architecture and focus on problems that occurred due to the use of firewalls and NAT. We contribute our present solutions and also discuss alternative solution ideas. One solution using the so-called “hole punching” technology is described in more detail.

1 Introduction

In the project WISENT (Knowledge Network Energy Meteorology, wisent.d-grid.de) [H^{+}06], funded by the German Federal Ministry of Education and Research (BMBF, www.bmbf.de), six project partners work together to improve processes and applications in the domain of energy meteorology. These partners are the OFFIS Institute for Information Technology, the Energy and Semiconductor Research Laboratory (EHF) of the University of Oldenburg, three institutes of the German Aerospace Center (DLR-TT, DLR-IPA, DLR-DFD) and the commercial partner Meteocontrol. One part of the collaboration relies on the sharing of distributed resources to gain common access to computing resources and large, distributed, and heterogenous data sources. These resources are connected via Grid technologies such as Globus Toolkit 4 (GT4) [GT4] and Condor [Con]. Unfortunately, the construction of this Grid infrastructure is hindered by blocking firewalls due to strong firewall policies and by the use of network address translation (NAT), also known as masquerading. Thus, project partners' Grid middleware installations cannot communicate without restrictions. We consider several approaches to cope with this problem. One approach utilizes the so-called "hole punching", which is a commonly used NAT traversal technique in peer-to-peer systems [For05]. This technique is also applicable to traverse NAT system in Grid environments.

The rest of this paper is organized as follows. In Section 2 we describe the WISENT Grid architecture. In Section 3 we report on problems with firewalls and NAT that occurred at the German aerospace center (DLR) during the construction of the Grid infrastructure. Afterwards, we present current or possible future solutions, including the NAT traversal technique in Section 4. We conclude with an outlook on our future work in Section 5.

2 Architecture

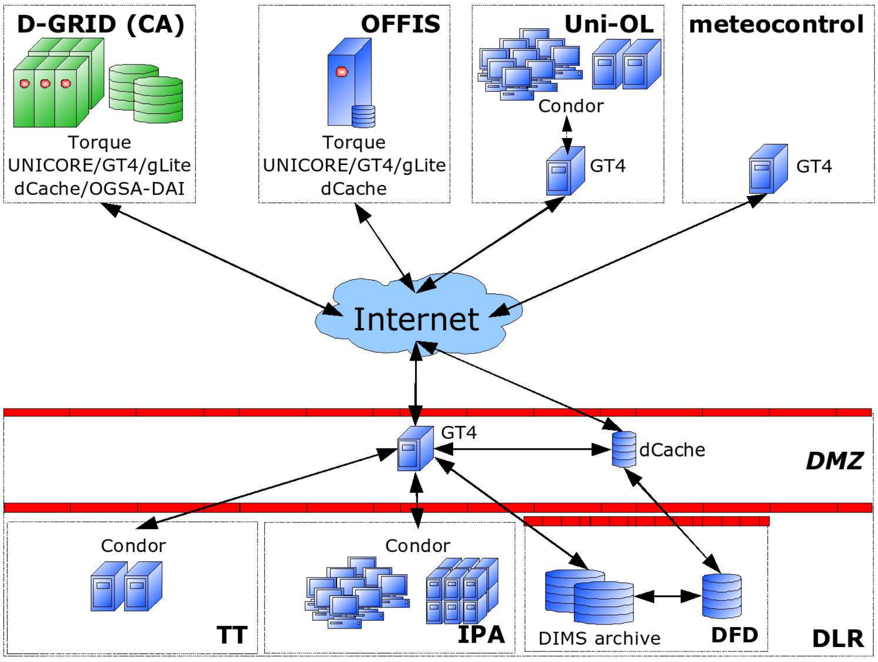

Figure 1: Grid infrastructure in WISENT

The WISENT Grid infrastructure is shown in Figure 1. The blue resources exist at our project partners, and the green resources are available in the German D-Grid infrastructure. To construct the Grid infrastructure, we have chosen a bottom-up approach. First, all Grid resources within the internal network of each project partner are connected or prepared for external connection. Afterwards, the obtained six separate networks are connected together to form the planned Grid infrastructure, which is also called a Community Grid. The last step consists of integrating with the German D-Grid infrastructure, where D-Grid provides a certificate authority (CA) as a basis for a security infrastructure.

One task for the internal networks is to support compute-intensive applications with existing CPU resources available at DLR-TT, DLR-IPA and the University of Oldenburg. These resources are a dedicated cluster, several servers, and desktop PCs. A promising approach comes with Condor, a batch system designed for using idle CPU cycles ("cycle-scavenging") of regulars PCs. It is well-suited for building desktop Grids consisting of regular PCs, in which computations are performed on separate data sets with no message exchange between distributed processes. Condor jobs are unobtrusive for desktop PC users, as they are suspended when Condor determines a user's activity by observing mouse and keyboard events.

Besides of the already deployed Condor pools, the WISENT project's further internal resources consist of a HPC cluster with data storage at OFFIS and dedicated data storage at DLR-DFD. As recommended by the D-Grid integration project DGI, we are using the batch system TORQUE [tor] for the HPC cluster and plan to deploy the data management software dCache [dCa] for the data storages, excluding the Data Information and Management System (DIMS) at DLR DFD. The DIMS archive is a tape storage which can be accessed only through proprietary services supporting data retrieval and post-processing. Its complex architecture does not permit direct access through dCache.

All Grid resources in the six internal networks are externally connected to support data transfers and to enable access to external computing resources. Our project partners are connected using GT4 because of its comprehensive support for data management and data transfers provided by the GridFTP and the Reliable File Transfer (RFT) services.

GT4 also offers functionality to access Condor and TORQUE pools via the resource manager WS-GRAM, deployed as a WSRF Grid service. However, the access to WS-GRAM differs from Condor and users who are familiar with Condor have no interest in learning another job description language and further tools. This problem is solved by Condor-G, which allows Condor jobs to be submitted to Grid middleware systems such as GT4. Thus, computing resources can be accessed via WS-GRAM directly or via the Condor-G bridge. The HPC cluster at OFFIS is accessible through GT4, UNICORE, and gLite, as recommended by DGI.

3 Problems with firewalls and NAT

As described in Section 2 the WISENT Grid infrastructure mainly relies on Condor and GT4. We have no firewall or NAT problems when using a Condor pool at a single location, because no blocking firewalls or a NAT system exist between the participating hosts. However, GT4 is deployed in and accessed from different locations whose networks are protected by firewalls and/or use NAT. This architecture does not allow unhampered communication and thus a GT4 host cannot be deployed in and accessed from every network. These problems mainly occur in the DLR network, where multiple strong firewalls exist due to restrictive security policies and NAT is used.

In a typical firewall configuration all incoming connections are blocked except for a few ports belonging to mail or web servers. Outgoing connections, however, can often pass the firewall unblocked. In some cases, a facility's network is divided into a so-called "demilitarized zone" (DMZ) and an internal network. Servers that must be reachable from the outside are located in the DMZ. All other hosts, such as users' PCs or security relevant systems with company-internal data, are in the internal network. Incoming connections to the internal network are usually completely blocked, except for a few explicitly allowed connections from the DMZ. The DLR network is divided in such a DMZ and an internal network with corresponding firewall policies (Figure 1). DLR-DFD is additionally protected by an internal firewall, behind which the DIMS archive is located. Thus, the deployment of GT4 is only possible in the DMZ, as direct connections to resources in the internal DLR network are not possible.

Another problem with the DLR (DMZ) firewall is caused by the use of so-called dynamic or ephemeral ports ([Wel06]). These ports typically belong to the "untrusted" port range (>1024) and are blocked because administrators generally disapprove of wide port ranges remaining open in the firewall. We are using two services of GT4 that rely on ephemeral ports, GridFTP and WS-GRAM. GridFTP uses a static port (2811) for a control channel and multiple ephemeral ports for data channels. WS-GRAM itself is running on a static port (8443) too, but ephemeral ports are needed on the client side for file staging and callbacks during job execution. About 20 ephemeral ports are needed per user in general [Wel06]. Thus, the port range needed by GT4 grows if many users are working simultaneously. Fortunately, GT4 offers an option to limit the used range of ephemeral ports. This option is supported by WS-GRAM and GridFTP and is already used in the D-Grid infrastructure [VG06].

The main purpose of NAT is to cope with the shortage of globally unique IP adresses. Hosts in an internal network are assigned IP addresses from a non-unique, private address range and configured to route their external communication through the NAT system. NAT transparently translates the internal source IP address and port number contained in an outgoing packet to a single externally known IP address and associated port numbers. Likewise, the external destination IP address and port number from an incoming packet is translated back to the internal pair. This technique is also known as "masquerading" because it hides the internal IP range from the outside world. A pair consisting of an IP address and port number is also called an "endpoint". The role of NAT is thus translating between private and public endpoints. At any time, the NAT system maintains a table of sessions, each session representing a translation rule. A NAT session is established upon receiving an outgoing packet from the internal network. It ends after receiving an incoming packet which terminates a TCP connection or after an idle timeout occurs.

All external traffic to a port number with no valid NAT session is discarded. This obviously precludes connection attempts to internal hosts from the outside network. To work around this restriction, most NAT system support static configuration of public-private endpoint pairs. At DLR this NAT technique is used for parts of the network, one exception is the DMZ.

The NAT technique conflicts with some properties of GT4 [Wel06]. The Grid Security Infrastructure (GSI) demands that each GT4 host is directly reachable through a so-called fully qualified domain name (FQDN), which does not work with NAT systems. This is not a problem since the GT4 host at DLR is located in the DMZ with a globally unique IP address and a FQDN. But each connection attempt from the GT4 host to the internal DLR network is blocked, so internal batch systems such as Condor are unreachable, which makes job submission impossible. Furthermore, NAT systems hinder the use of WS-GRAM client's full functionality within these networks. When a WS-GRAM client submits a job to an external WS-GRAM service, it opens an outgoing connection to the service as well as an additional local ephemeral port for callbacks from the service. However, the NAT system is unaware of the client's ephemeral port and thus rejects callback connections (without NAT, they would be blocked by a firewall). Thus, the WS-GRAM service and others cannot perform any callbacks to a client behind a NAT system or firewall.

In summary, connecting distributed Grid resources is challenging due to strong firewall policies and the use of NAT systems. Workarounds and other techniques that can be used to cope with the presented problems are described in the following section.

4 Coping with firewalls and NAT

Initially, we considered the following possible solutions to cope with the firewall problems mentioned in Section 3:

1. An extra "Grid zone" constructed beside the DLR DMZ.
2. A tunnel via virtual private network (VPN) established with each external project partner.
3. An application level gateway (ALG) used as a transparent proxy.

The first two solutions are very suitable to solve the firewall problems. Yet, these solutions were rejected because they are associated with additional costs, such as purchasing extra hardware. Additionally, the creation of a VPN with each project partner decreases the flexibility of the Grid infrastructure. The third solution is to place an application level gateway (ALG), a software component developed at DLR Sistec, into the DMZ as a transparent proxy between internally and externally installed Grid software. This is a good approach as it allows a secure and filtered communication across firewalls. However, each packet must pass the ALG, which would reduce the throughput of data transfers significantly. Furthermore, the ALG is still under development.

At this point the last possibility was to place all gateway components—in our case a host with GT4—into the DMZ. A fixed port range of so-called controllable ephemeral ports is opened for explicitly named external hosts, so that each project partner outside the DLR network can access the GridFTP server and the WS-GRAM service [VG06]. Even then we must cope with the fact that the gateway cannot directly connect to resources in the internal DLR network such as data storages and Condor pools. Only outgoing connections in the other direction are allowed by the internal firewall. Respecting this policy, DLR currently delivers data products for external partners through a pick-up point in the DLR DMZ. The requested files can be fetched via a FTP server in this setup. Requests for data products arrive at a host located in the DLR DMZ. Dedicated hosts within the internal DLR network retrieve these requests, extract the desired data product from the DIMS archive and deliver it to the pick-up point. For technical reasons, it is not possible to substitute this chain completely using Grid technologies. Thus, our task is to integrate it with Grid components installed in the DLR DMZ. A similar challenge exists for the WS-GRAM service. At this moment it is not possible to access any Condor pool in the DLR network via WS-GRAM in the DLR DMZ due to the firewall restrictions. The solution could be analogous to the delivery of data products. A job submission with input data is fetched from the DLR DMZ, forwarded to the Condor pool, and the output is delivered to the pick-up point. However, external monitoring of job execution would not be possible using this approach. In general, these approaches for data delivery and job submission, where only outgoing connections are initiated, can be extended to further scenarios. For example, each data connection with a GridFTP server behind a firewall could be initiated by the server itself, if the application scenario allows it. In this case only outgoing connections are using ephemeral ports and the port range for incoming connections can remain closed. This works for GridFTP as long as no parallel data channels are used.

The NAT problem is similar in nature to the firewall problem and permits an analogous solution: if possible, use only outgoing connections. For example, the GT4 client can submit jobs without opening a local port for callbacks. Unfortunately, this solution has drawbacks. Job monitoring is then only possible by fetching the status periodically. Therefore, we consider another approach that is already used in peer-to-peer systems.

When two peers or hosts are behind a NAT system, they cannot establish direct communication using just the standard network protocols. A technique called "hole punching" relies on their ability to communicate with a broker (which has a globally unique IP address) to establish a direct communication link between them. The technique establishes a NAT session at each of the two parties by reusing the public endpoints assigned to the their respective NAT sessions with the broker S. This technique works with UDP as well as with TCP connections [For05].

The following simplified example, some technical details are omitted, describes a typical scenario. Host A and Host B are behind a NAT system and they cannot communicate directly. However, each host maintains an outgoing connection with the broker Host S. Host A and Host B each have a public-private endpoint session with their NAT systems for the communication with Host S. Host S itself knows their public endpoints. We assume that the public endpoint of Host A uses the port 4000 and Host B has one with port 5000. Further used ports are not essential for understanding this technique and are thus omitted. If Host A wants to open a connection with Host B the following steps are performed. The step's numbers correlate with Figure 2. Those port numbers that are not essential for understanding, are marked with the character string "xxxx".

1. Host A submits the request for communication with Host B to Host S
2. Host S submits the public endpoint (port 4000) of Host A to Host B.
3. Host B sends a packet via it's public-private endpoint (port 5000) to the public endpoint of Host A (4000). The NAT system of Host A rejects the packet, but a NAT session on B's side is now established ("a hole is punched"), ready to translate any packets from A's public endpoint (port 4000) to B's public endpoint (port 5000) into B's corresponding private endpoint.
4. Host B notifies Host S and awaits an incoming connection from Host A.
5. Host S submits the public endpoint of Host B (port 5000) to Host A.
6. Host A uses its public-private endpoint (port 4000) to establish a connection with Host B's public endpoint (port 5000).

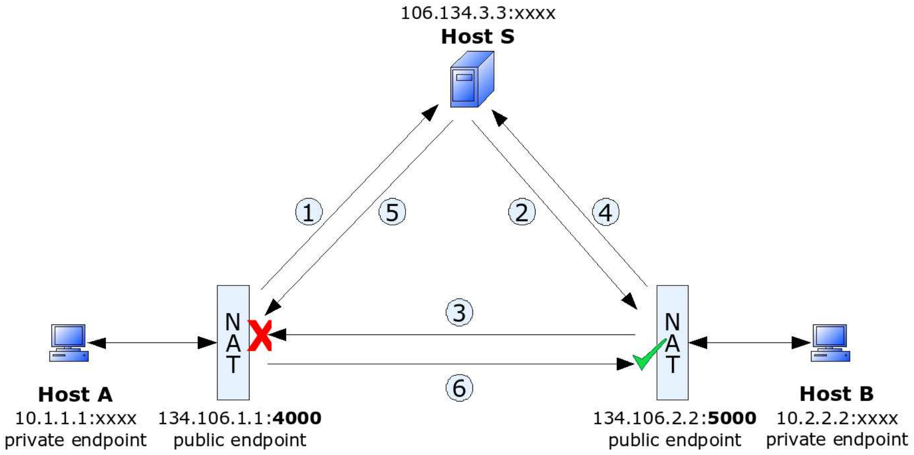

Figure 2: Simplified hole punching example

In fact, the hole punching technique has some variants. One problem with hole punching is that the behavior of NAT systems varies and hole punching does not work with all of them. A comprehensive study has shown that about 82 percent of NAT systems allow hole punching for UDP and about 64 percent allow hole punching for TCP [For05]. TCP hole punching is less often supported than the UDP counterpart because of the protocol's additional complexity. The use of TCP in Grid software increases the risk of incompatibility with NAT systems. Fortunately, the number of NAT systems supporting both hole punching techniques is expected to increase.

Another problem is the FQDN requirement mentioned in Section 3. A single NAT system may be used by several Grid components, deployed on different machines. In this case, each component's server certificate would have to contain the FQDN associated with the external IP address. In principle, this should be possible, but we have not tested such a scenario yet.

Beside NAT systems, hole punching is also suitable for firewalls that block incoming connections and allow outgoing connections. When the two hosts in the scenario above are behind firewalls, this technique is also called dynamic port configuration. Furthermore, the use of a connection broker itself has additional benefits. It can be also used for delegating connection initiation when one host is behind a NAT system or a firewall and the other host is reachable through a globally unique IP address. For example, the hidden host can open outgoing data channels for data transfers via GridFTP to the public host. In general, if allowed by corporate security policies, hole punching is a suitable technology to use in Grid software across firewalls and NAT systems.

We plan to implement a prototype connection broker to test our existing firewalls and NAT systems. Afterwards, an exemplary extension for GT4 will be implemented to use this broker for establishing connections with other hosts behind NAT systems and firewalls. Naturally, the server must support the GSI of GT4 for secure connections. An integration with the ALG proxy is possible as well. At this time, we are not aware of any Grid software that supports hole punching, even though the technique has been considered before. For example, the use of UDP hole punching in Grid environments is discussed in [GMNP06].

5 Conclusions and future work

In this paper we described the WISENT Grid architecture and the firewall and NAT problems encountered during its construction. We introduced different solutions or workarounds that are employed in our Grid infrastructure or that we plan to employ. One focus was the use of a connection broker that can be used for hole punching to traverse firewalls and NAT systems. We plan to implement such a broker in the future and to test it in our Grid environment.

Today, the Grid technology is not ready for unhindered use across organizations with strong firewall policies or NAT systems. Especially the use of ephemeral ports conflicts with the usual firewall configurations and security policies. Regardless of the approach, Grid middleware must account for firewalls and NAT to gain wider acceptance. This topic is particularly important for the adoption of Grid technologies in commercial facilities where the strong firewall policies and NAT systems are even more common than in academic institutions.

References

[Con] Condor High Throughput Computing. http://www.cs.wisc.edu/condor/.

[dCa] dCache. http://www.dcache.org.

[For05] B. Ford. Peer-to-peer communication across network address translators, 2005. USENIX Annual Technical Conference.

[GMNP06] E. Grünter, M. Meier, R. Niederberger, and F. Petri. Dynamic Configuration of Firewalls Using UDP Hole Punching. Technical report, Forschungszentrum Jülich, 2006.

[GT4] Globus Toolkit. http://www.globus.org/toolkit/.

[H$^+$06] Wilhelm Hasselbring et al. WISENT: e-Science for Energy Meteorology. In *Proceedings of 2nd IEEE International Conference on e-Science and Grid Computing (e-Science'06)*, pages 93–100, Amsterdam, Netherlands, December 2006. IEEE Computer Society Press.

[tor] TORQUE. http://www.clusterresources.com/pages/products.php.

[VG06] Gian Luca Volpato and Christian Grimm. Empfehlungen zur statischen Konfiguration von Firewalls im D-Grid. Technical report, D-Grid Integrationsprojekt, 2006.

[Wel06] Von Welch. Globus Toolkit Firewall Requirements. Technical Report 9, Globus Alliance, 10 2006. http://www.globus.org/toolkit/security/firewalls/.