Towards the design of an Active Grid

Jean-Patrick Gelas, Laurent Lefèvre

RESAM Laboratory UCB - Action INRIA RESO
Ecole Normale Supérieure de Lyon
46, allée d'Italie 69364 LYON Cedex 07 - France
Jean-Patrick.Gelas@ens-lyon.fr, Laurent.Lefevre@inria.fr

Abstract. Grid computing is a promising way to aggregate geographically distant machines and to allow them to work together to solve large problems. After studying Grid network requirements, we observe that the network must take part in the Grid computing session to provide intelligent adaptative transport of Grid data streams.
By proposing new intelligent dynamic services, active network can easily and efficiently deploy and maintain Grid environments and applications. This paper presents the Active Grid Architecture (A-Grid)[1] which focuses on active networks adaptation for supporting Grid middlewares and applications.
We describe the architecture and first experiments of a dedicated execution environment dedicated to the Grid: Tamanoir-G.

1 Introduction

In recent years, there has been a lot of research projects on Grid computing which is a promising way to aggregate geographically distant machines and to allow them to work together to solve large problems ([9, 16, 5, 7, 14, 2, 8, 6]). Most of proposed Grid frameworks are based on Internet connections and do not make any assumption on the network. Grid designers only take into account of a reliable packet transport between Grid nodes and most of them choose TCP/IP protocol.

This leads to one of the more common complaint of Grid designers : networks do not really support Grid applications.

Meantime, the field of active and programmable networks is rapidly expanding [17]. These networks allow users and network designers to easily deploy new services which will be applied to data streams. While most of proposed systems deal with adaptability, flexibility and new protocols applied on multimedia streams (video, audio), no active network efficiently deal with Grid needs.

In this paper we propose solutions to merge both fields by presenting The Active Grid Architecture (A-Grid) which focus on active network adaptation for supporting Grid environments and applications. This active Grid Architecture

[1] This research is supported by French Research Ministry and ACI-GRID project JE7 RESAM.

proposes solutions to implement the two main kind of Grid configurations : meta-cluster computing and global computing. In this architecture the network will take part in the Grid computing session by providing efficient and intelligent services dedicated to Grid data streams transport.

This paper reports on our experience in designing an Active Network support for Grid Environments. First, we classify the Network Grid requirement depending on environments and applications needs (section 2). In section 3 we propose the Active Grid Architecture. We focus our approach by providing support for the most network requirements from Grid. In section 4, we describe Tamanoir-G, the Active Grid framework and first experiments. Finally, in the last section we present our future works.

2 Network requirements for the Grid

A distributed application running in a Grid environment requires various kinds of data streams: Grid control streams and Grid application streams.

2.1 Grid control streams :

We can classify two basic kinds of Grid usage :

- Meta cluster computing :
  A set of parallel machines or clusters are linked together over Wide Area Networks to provide a very large parallel computing resource. Grid environments like Globus[9], MOL[16], Polder[5] or Netsolve[7] are well designed to handle meta-cluster computing session to execute long-distance parallel applications.
  We can classify various network needs for meta-clustering sessions :
  - Grid environment deployment : The Grid infrastructure must be easily deployed and managed : OS heterogeneity support, dynamic topology re-configuration, fault tolerance...
  - Grid application deployment : Two kind of collective communications are needed : multicast and gather. The source code of any applications is multicast to a set of machines in order to be compiled on the target architectures. In case of Java based environments, the *bytecode* can be multicast to a set of machines. In case of an homogeneous architecture, the binaries are directly sent to distant machines. After the running phase, results of distributed tasks must be collected by the environment in a gathering communication operation.
  - Grid support : The Grid environment must collect control data : node synchronization, node workload information... The information exchanged are also needed to provide high-performance communications between nodes inside and outside the clusters.

– Global or Mega-computing : These environments usually rely on thousand of connected machines. Most of them are based on computer *cycles stealing* like Condor[14], Entropia[2], Nimrod-G[8] or XtremWeb[6].
  We can classify various network needs for Global-computing sessions :
  - Grid environment deployment : Dynamic enrollment of unused machines must be taken into account by the environment to deploy tasks over the mega-computer architecture.
  - Grid application deployment : The Grid infrastructure must provide a way to easily deploy and manage tasks on distant nodes. To avoid the restarting of distributed tasks when a machine crashes or become unusable, Grid environments propose check-pointing protocols, to dynamically re-deploy tasks on valid machines.
  - Grid support : Various streams are needed to provide informations to Grid environment about workload informations of all subscribed machines. Machine and network sensors are usually provided to optimize the task mapping and to provide load-balancing.

Of course, most of environments work well on both kind of Grid usage like Legion[12], Globus[9], Condor[14], Nimrod-G[8] ...

2.2 Grid application streams

A Grid computing session must deal with various kind of streams :

– Grid application input : during the running phase, distributed tasks of the application must receive parameters eventually coming from various geographically distant equipments (telescopes, biological sequencing machines...) or databases (disk arrays, tape silos...).
– Wide-area parallel processing : most of Grid applications consist of a sequential program repeatedly executed with slightly different parameters on a set of distributed computers. But with the raise of high performance backbones and networks, new kind of real communicating parallel applications (with message passing libraries) will be possible on a WAN Grid support. Thus, during running phase, distributed tasks can communicate data between each others. Applications may need efficient point to point and global communications (broadcast, multicast, gather...) depending on application patterns. These communications must correspond to the QoS needs of the Grid user.
– Coupled (Meta) Application : they are multi-component applications where the components were previously executed as stand-alone applications. Deploying such applications must guarantee heterogeneity management of systems and networks. The components need to exchange heterogeneous streams and to guarantee component dependences in pipeline communication mode. Like WAN parallel applications, QoS and global communications must be available for the components.

Such a great diversity of streams (in terms of messages size, point to point or global communications, data and control messages...) requires an intelligence in the network to provide an efficient data transport.

3 Active Grid Architecture

We propose an active network architecture dedicated to Grid environments and Grid applications requirements : the A-Grid architecture.

An active grid architecture is based on a virtual topology of active network nodes spread on programmable routers of the network. Active routers, also called Active Nodes (AN), are deployed on network periphery (edge equipments).

As we are concerned by a wide active routers approach, we do not believe in the deployment of Gigabit active routers in backbones. If we consider that the future of WAN backbones could be based on all-optical networks, no dynamic services will be allowed to process data packets. So, we prefer to consider backbones like high performance well-sized passive networks. We only concentrate active operations on edge routers/nodes mapped at network periphery.

Active nodes manage communications for a subset of Grid nodes. Grid data streams cross various active nodes up to passive backbone and then cross another set of active nodes up to receiver node. The A-Grid architecture is based on Active Node approach : programs, called services, are injected into active nodes independently of data stream. Services are deployed on demand when streams arrive on an active node. Active nodes apply these services to process data streams packets.

3.1 Active Grid architecture

To support most of Grid applications, the Active Grid architecture must deal with the two main Grid configurations :

- Meta cluster computing (Fig. 1) :
  In this highly coupled configuration, an active node is mapped on network head of each cluster or parallel machine. This node manage all data streams coming or leaving a cluster. All active nodes are linked with other AN mapped at backbone periphery. An Active node delivers data streams to each node of a cluster and can aggregate output streams to others clusters of the Grid.
- Global or Mega computing (Fig. 2) :
  In this loosely coupled configuration, an AN can be associated with each Grid node or can manage a set of aggregated Grid nodes. Hierarchies of active nodes can be deployed at each network heterogeneity point.
  Each AN manages all operations and data streams coming to Grid Nodes : subscribing operations of voluntary machines, results gathering, nodes synchronization and check-pointing...

For both configurations, active nodes will manage the Grid environment by deploying dedicated services adapted to Grid requirements : management of nodes mobility, dynamic topology re-configuration, fault tolerance....

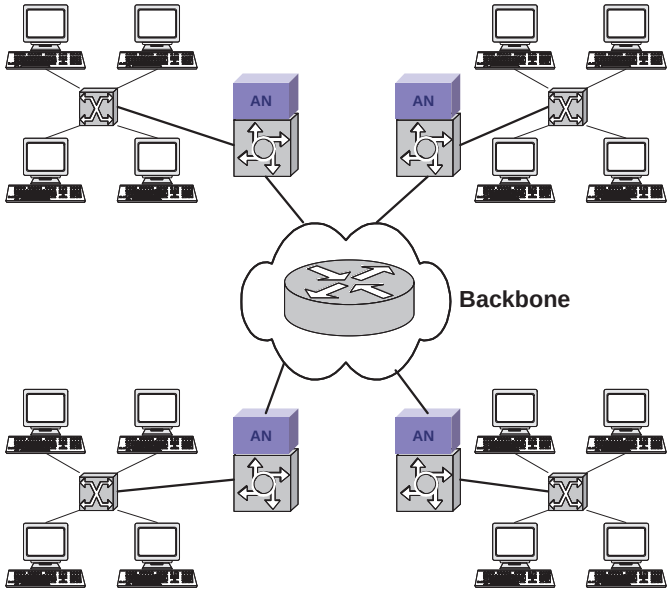

Fig. 1. Meta cluster computing Active Grid Architecture

3.2 Active network benefits for Grid applications

Using an Active Grid architecture can improve the communications needs of Grid applications :

- Application deployment : To efficiently deploy applications, active reliable multicast protocols are needed to optimize the source code or binary deployment and the task mapping on the Grid configuration accordingly to resources managers and load-balancing tools. An active multicast will reduce the transport of applications (source code, binaries, bytecode...) by minimizing the number of messages in the network. Active node will deploy dedicated multicast protocols and guarantee the reliability of deployment by using storage capabilities of active nodes.
- Grid support : the Active architecture can provide informations to Grid framework about network state and task mapping. Active nodes must be open and easily coupled with all Grid environment requirements. Active nodes will implement permanent Grid support services to generate control streams between the active network layer and the Grid environment.
- Wide-area parallel processing : with the raise of grid parallel applications, tasks will need to communicate by sending computing data streams with QoS requests. The A-Grid architecture must also guarantee an efficient data transport to minimize the software latency of communications. Active nodes

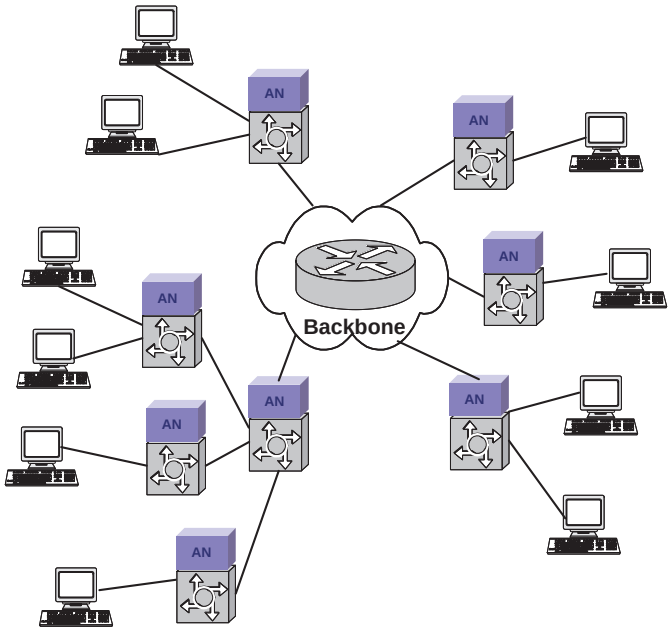

Fig. 2. Global Computing Active Grid Architecture

deploy dynamic services to handle data streams : QoS, data compression, "on the fly" data aggregation...

– Coupled (Meta) Application : the Active architecture must provide heterogeneity of services applied on data streams (data conversion services...). End to end QoS dynamic services will be deployed on active nodes to guarantee an efficient data transport (in terms of delay and bandwidth).

Most services needed by Grid environments : high performance transport, dynamic topology adapting, QoS, on-the-fly data compression, data encryption, data multicast, data conversion and error management must be easily and efficiently deployed on demand on an Active Grid architecture. To allow an efficient and portable service deployment, we will present in next section our approach to propose an active network framework easily mergeable with a Grid environment : The Tamanoir-G Framework.

4 Tamanoir-G : a Framework for Active Grid support

We explore the design of an intelligent network by proposing a new active network framework dedicated to high performance active networking. The Tamanoir-G framework [11] is an high performance prototype active environment based on

active edge routers. Active services can be easily deployed in the network and are adapted to architecture, users and service providers requirements.

A set of distributed tools is provided : routing manager, active nodes and stream monitoring, web-based services library... Tamanoir-G is based on a JAVA multi-threaded design to combine performance and portability of services, applications can easily benefit of personalized network services through the injection of Java code.

4.1 Overview of a Tamanoir-G node

An active node is a router which can receive packets of data, process them and forward them to other active nodes.

A Tamanoir-G Active Node (TAN) is a persistent dæmon acting like a dynamic programmable router. Once deployed on a node, it is linked to its neighbors in the active architecture. A TAN receives and sends packets of data after processing them with user services. A TAN is also in charge of deploying and applying services on packets depending on application requirements. When arriving in a Tamanoir-G dæmon, a packet is forwarded to service manager. The packet is then processed by a service in a dedicated thread. The resulting packet is then forwarded to the next active node or to the receiver part of application according to routing tables maintained in TAN.

4.2 Dynamic service deployment

In Tamanoir-G, a service is a JAVA class containing a minimal number of formatted methods. If a TAN does not hold the appropriate service, a downloading operation must be performed.We propose three kind of service deployment. The first TAN crossed by a packet can download the useful service from a service broker. By using an *http address* in service name, TAN contact the web service broker, so applications can download generic Tamanoir-G services to deploy non-personalized generic services. After, next TANs download the service from a previous TAN crossed by packet or from the service broker.

4.3 Multi-protocols support

Most of existing active frameworks propose active services dedicated to UDP streams (like ANTS [18], PAN[15]...). Tamanoir-G environment has been extended to support various transport protocols and specially TCP protocol used by most of Grid middlewares and applications (see figure 3). By this way, Tamanoir-G services can be easily adapted and merged in a Grid Middleware to provide adaptative network services.

4.4 Tamanoir-G Performances

We based our first experiments of Tamanoir-G on Pentium II 350 MHz linked with Fast Ethernet switches. These experiments show that the delay needed to

cross a Tamanoir-G active node (latency) remains constant (under 750 μs for simple forwarding operation).

Results presented in figure 3 show bandwidth comparisons of forwarding service of a Tamanoir-G node with UDP and TCP transport. While kernel passive forward operations (*kr*) provide around 50 Mbits/s, active forwarding services running in a JVM obtain good results (around 30 Mbit/s) with Kaffe(*kaffe*) [4], Blackdown(*jvmBkDwn*) [1] and IBM(*jvmIBM*) [3] Java Virtual Machines. We note poor performances obtained with GCJ(*gcj*) [10] compiled version around 17 Mbit/s due to lack of optimizations in GCJ compiler.

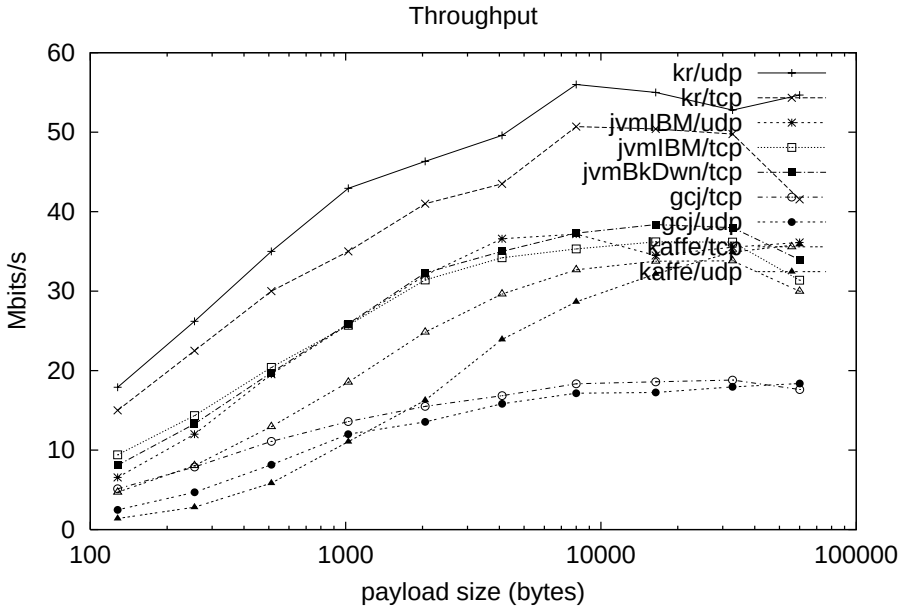

Fig. 3. Bandwidth of a passive (*kr*) or active forwarding operation with Tamanoir-G in UDP or TCP. Comparisons between Java Virtual Machines (*kaffe, jvmBkDwn, jvmIBM*) and compiled approach (*gcj*).

These first experiments show that Tamanoir-G framework can support a Grid environment without adding too much latency to all data streams. So Tamanoir-G can efficiently deploy services on active nodes depending on Grid requirements : QoS, data conversion, multicast, filtering operations, "on the fly" data compression...

5 Conclusion and future works

We have analyzed the Grid computation models in order to determine the main Grid network requirements in terms of efficiency, portability and ease of deployment.

We then studied a solution to answer these problems : the active networking approach, where all network protocols can be deported in the network in a transparent way for the Grid designer and the Grid user.

Most of services needed by Grid environments : high performance transport, dynamic topology adapting, QoS, on-the-fly data compression, data encryption, data multicast, data conversion, errors management must be easily and efficiently deployed on demand by an Active Grid architecture.

We proposed an active network support : the Tamanoir-G Framework. The first results are promising and should lead major improvements in the behavior of the Grid when the active support will be deployed.

Our next step will consist of merging the Tamanoir-G framework with a Middleware Grid environment in order to provide an environment for Active Grid services (reliable multicast, Active DiffServ [13]...) for middlewares and applications.

References

[1] Blackdown JVM. http://www.blackdown.org/.

[2] Entropia : high performance internet computing. http://www.entropia.com.

[3] IBM Java Developer Kit for Linux. http://www.alphaworks.ibm.com/tech/linuxjdk.

[4] Kaffe : an open source implementation of a java virtual machine. http://www.kaffe.org/.

[5] The Polder Metacomputing Initiative. http://www.science.uva.nl/projects/polder.

[6] Xtremweb : a global computing experimental platform. http://www.xtremweb.net.

[7] M. Beck, H. Casanova, J. Dongarra, T. Moore, J. Planck, F. Berman, and R. Wolski. Logistical quality of service in netsolve. *Computer Communication*, 22(11):1034–1044, july 1999.

[8] Rajkumar Buyya, Jonathan Giddy, and David Abramson. An evaluation of economy-based resource trading and scheduling on computational power grids for parameter sweep applications. In C. S. Raghavendra S. Hariri, C. A. Lee, editor, *Active Middleware Services, Ninth IEEE International Symposium on High Performance Distributed Computing*, Pittsburgh, Pennsylvania, USA, aug 2000. Kluwer Academic Publishers. ISBN 0-7923-7973-X.

[9] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *Intl J. Supercomputing Applications*, 11(2):115–128, 1997.

[10] GCJ. The GNU Compiler for the Java Programming Language. http://sourceware.cygnus.com/java/.

[11] Jean-Patrick Gelas and Laurent Lefèvre. Tamanoir: A high performance active network framework. In C. S. Raghavendra S. Hariri, C. A. Lee, editor, *Active Middleware Services, Ninth IEEE International Symposium on High Performance Distributed Computing*, pages 105–114, Pittsburgh, Pennsylvania, USA, aug 2000. Kluwer Academic Publishers. ISBN 0-7923-7973-X.

[12] Andrew Grimshaw, Adam Ferrari, Fritz Knabe, and Marty Humphrey. Legion: An operating system for wide-area computing. *IEEE Computer*, 32(5):29–37, May 1999.

[13] L. Lefèvre, C. Pham, P. Primet, B. Tourancheau, B. Gaidioz, J.P. Gelas, and M. Maimour. Active networking support for the grid. In Noaki Wakamiya Ian W. Marshall, Scott Nettles, editor, *IFIP-TC6 Third International Working Conference on Active Networks, IWAN 2001*, volume 2207 of *Lecture Notes in Computer Science*, pages 16–33, oct 2001. ISBN: 3-540-42678-7.

[14] Miron Livny. Managing your workforce on a computational grid. In Springer Lecture Notes in Computer Science, editor, *Euro PVM MPI 2000*, volume 1908, Sept 2000.

[15] Erik L.Nygren, Stephen J.Garland, and M.Frans Kaashoek. PAN: A High-Performance Active Network Node Supporting Multiple Mobile Code Systems. In *IEEE OPENARCH '99*, March 1999.

[16] A. Reinefeld, R. Baraglia, T. Decker, J. Gehring, D. Laforenza, J. Simon, T. Romke, and F. Ramme. The mol project: An open extensible metacomputer. In *Heterogenous computing workshop HCW'97,IPPS'97*, Geneva, April 1997.

[17] D. L. Tennehouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Winden. A survey of active network research. *IEEE Communications Magazine*, pages 80–86, January 1997.

[18] David Wetherall, John Guttag, and David Tennenhouse. ANTS : a toolkit for building and dynamically deploying network protocols. In *IEEE OPENARCH '98*, April 1998.