Toward the Implementation of Production Grid Systems: The Hungarian ClusterGrid Infrastructure Project

P Stefán, F Szalai, G Vizéz
Office for National Information and Infrastructure Development
NIIF/HUNGARNET
Victor Hugo str 18-22, H-1132 Budapest, Hungary
Phone: +36 1 4503070, Fax: +36 1 3506750, E-mail: grid-tech@niifhu ,
Web: http://wwwclustergridiif hu

Keywords: grid, cluster, high-performance computing

Abstract

In the paper the key differences between classical computational grid development approaches and the approach used in developing the Hungarian ClusterGrid infrastructure will be compared While the classical grid approaches often make the assumption that certain resources exist and aiming at developing the pure connection among the resources, the ClusterGrid infrastructure is a user-driven, production-like system which maps into a grid infrastructure modular reference layout model initiated by the Global Grid Forum, as well as builds a system from the bottom to the top

1 INTRODUCTION

The Hungarian ClusterGrid project which dates back into July, 2002, and which has been operating as a production grid system since July 2003, was originally proposed to connect more than 100 personal computer (PC) labs located at Hungarian universities, polytechnics and public libraries, donated by the Hungarian Ministry of Education The PC labs involved were not considered to be grid resources in the classical terms rather they were treated as raw resources , hardware which should serve multiple purposes such as education or library services

On the other hand, the PC labs were never considered to fulfill their fundamental purpose 24 hours a day, so it was pretty straightforward to utilize the computational power of these machines whenever they are not used for education, say, during the nights and during the week-ends

2 THE KEYWORD: SEPARATION

Unfortunately the technical overlapping among the different functionalities is considerably small On one hand, when the labs are used for education, they often operate in an office environment executing Windows and office tools (Whatever is used, the exact software layout and configuration are always up to the individual institutes who operate the lab) On the other hand high performance and throughput computation (HPC) has a traditional background in the UNIX environment where the executable software can be easily developed, ported or optimized Since the two are structurally different, instead of forcing them to approach together, it is a much more clean way to separate the two on the same hardware both in space and in time

Temporal separation in this environment means that the machines in any lab are dual boot machines, and there are time slots for educational operation, and supercomputing or grid operation The two are referred to as day-time operation and night-time operations respectively regarding the typical period of the day the machines used in this way

Spatial separation is a compound principle: it basically means using separate operating system, on separate disk partitions in a separate network segment Spatial separation can be extended to all layers of the grid reference model used

3 THE GRID REFERENCE MODEL

While the classical grid approaches focus mainly on the connectivity among super-computational resources and on the application layer, contemporary grid development principles gradually start to recognize that many problems emerging at the production grid design cannot be solved only at the application layer Not even the connectivity, which following the separation principle should also be accomplished in a separated network segment Therefore, a grid reference model shown in Figure 1 is proposed in [1]

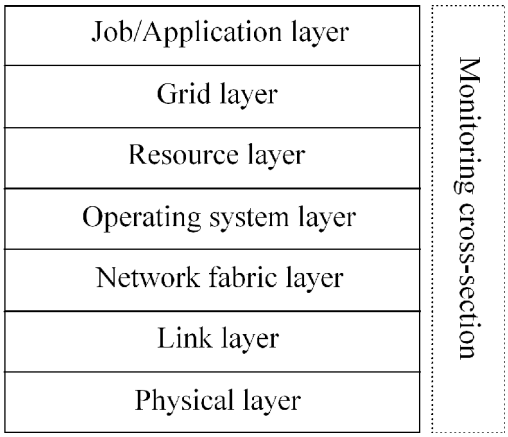

Figure 1: The grid reference model

The grid reference model (GRM) is a layered modular model design and shows many similarities to the networking reference models, such as ISO/OSI, or TCP/IP The similarity is pretty straightforward, since the compute grid as an interconnection of computational resources use the network in an organic way

31 The physical layer

At the physical layer there are machines, PCs and servers, in collective terms, the hardware The reason why the physical layer is so important is that there could be huge differences betweenthe different hardware layouts Most precisely a piece of hardware may be appropriate for one purpose, but may be inappropriate for another one There are five basic roles (and much more elaborated functions) in a possible grid implementation:

- Resource nodes: Resource nodes are compute nodes which need to have good CPU power and memory enough to store temporary data Resource nodes are similar to dumb terminals which do not need any intelligence but raw power to make fast computations
- Local master nodes: Local master nodes are specific nodes, being located in the same segment as the resource nodes, but serving different purposes They basically provide

services to the resource nodes, such as network boot service, network file system service, or connectivity service (see later in section 33) They do not accomplish computation at all

- Service nodes: Service machines basically provide grid-level services to all resources, machines, or labs For example: operating system mirroring, maintenance, statistics, logging or monitoring
- Entry nodes: Entries provide controlled access to the grid resources, carry out the user authentication, and maintain jobs and resources
- Grid job gateway nodes: Job gateways have the same basic functionality as entry nodes, with the difference that they are expected to be capable to handle, and transfer jobs from one grid system to the other one

Following the separation principle, all of these roles are expected to run on separate physical or virtual machines

32 The link layer

The link layer corresponds to the data link layer in the ISO/OSI networking reference model, which in the contemporary network design means connecting the different nodes with special network devices, called hubs or switches

From the grid perspective, again following the separation issue, the link layer deals with the following question: How can the grid traffic be separated from the ordinary Internet traffic? The reason why separation is so important at this low level of connectivity has security and policy reasons as well

Separation is carried out on the link layer level by using virtual local area networks (VLAN), or secure networks satisfying 8021q and 8021x standards In the ClusterGrid infrastructure design, the data link network configuration of each PC lab is configured to treat four networks:

- Public Network (PNET): A virtual segment that is routed, and used for ordinary communication
- Local Network (LNET): A virtual segment that is not router (it is either address translated, or proxied), and also used for Internet traffic communication
- Virtual Network (VNET): A virtual segment that is used for connecting the resource with the other resources
- Grid Network (GNET): A virtual segment that is used for grid mode communication

Out of the four virtual data link network segments, only GNET and VNET are mandatory

33 The network fabric layer

Using a separated network segment comes as a direct consequence from the spatial separation principle, and is built up on top of the link layer separation In the original ClusterGrid design, it was quite unreasonable to suppose that all grid resources are well-monitored, well-controlled Otherwise, the security risk of allowing each grid resource to access the public data network was large enough to drop the idea of connection through the public Internet and to search for other solutions The situation was even made worse by the grid software which used such communication technology (dynamic port-mapping, remote procedure calls) which does not allow low level data protection at all

In this environment, the only secure way of connecting resources is to use some sort of virtual computer networking, which operate on the same active network devices as the public Internet, but are separated in different data segments (routing tables, access lists), thus, they are safe enough to be considered as a separated data connection

Virtual private networks (VPN) can be implemented in many ways: One possible implementation is to use multi-protocol label switching (MPLS) technology by exploiting the good quality academic network The customer edge (CE) - provider edge (PE) layout used in MPLS [2] fits perfectly to the grid network fabric requirements, with the only extension of connecting the CE and the PE routers with each other via either intra-institutional VLAN or via tunneling

34 The operating system layer

When considering a large amount of interconnected compute nodes, the ease of management immediately comes to the focus Managing just a couple of PCs and administering several hundreds or thousands of PCs are totally different, and, thus, requires different approach: batch management of PS

Batch management, which is different from batch job management, involves finding the minimally common elements in the operating system (OS) for each roles mentioned section 31, determining the system used, and the package management policy which can help to keep the system clean and manageable

In the ClusterGrid infrastructure implementation, Linux is used as the grid OS, and Debian n package management has been found to bind files to managable packages By using network root file system on the clients the number of OS images can be significantly reduced, at the price of putting a special file server into the system

Apart from file sharing and auxiliary boot support services there is an additional function to deal with at the OS level: authentication The authentication process can be divided up to two sub-processes: to authenticate a user and to authenticate a job Following the separation principle these two processes are also distinguished, while user authentication is always done on the entry points of the system and nowhere else, job authentication is accomplished on the local masters, execution nodes There are numerous advantages of separating job and user entities from each other and maintain the link among them only on determined places

35 The resource layer

The resource layer basically refers to all software that is used in the classical terms of HPC computing, ranging from the local resource managers to the library routines that support utilizing the distributed environment, such as parallel virtual machine (PVM) or message passing interface (MPI) routines

The local job manager organizes a resource pool out of the individual PCs and manages the jobs that has been scheduled on them In most of the cases all machines should use a unified view on the file system through network or clustrer file systems This is one of the key reasons why local schedulers cannot step out of their intended local nature of use, and be used for building up a general grid

The purpose of the resource layer is to offer a single resource by transferring the PC lab into a managed computational resource, and there are a great variety of software that can be used such as Condor job manager or Sun Grid Engine (SGE)

36 The grid layer

By using a grid layer, several resources, like supercomputers and PC clusters can be joined together to form a single large computational grid The grid layer mainly binds the local clusters together and gathers, shares resource and job information among them via the local masters

In some particular cases the grid layer functionality can be implemented by using the lower resource layer s protocols such as Condor s flocking mechanism The advantage of the solution that it fits almost perfectly to the philosophy of the resource manager, but it may

introduce serious bottlenecks into the whole grid In the earlier implementation of the ClusterGrid infrastructure Condor s flocking left a great deal of IO operation burden on the entry machines, and therefore has been replaced by a separated grid layer protocol, and a communication entity which is referred to as the grid resource broker

(ide kellene valami Feritol)

The grid layer should also involve user access interfaces, such as command line interface (CLI) or some kind of graphical collaboration interface, called grid portals A good starting point, which is used in production in the ClusterGrid, can be found in [3]

37 The application layer

The users jobs are formulated at the application layer The number of tasks at the application layer is large ranging from software development (graphical parallel software development environment, distributed make utilities, etc), compilation, or in a more compact view the preparation of the executables, like code optimizing, packaging, and job handling (job submit, job query) Unfortunately activities at the application layer can hardly be automated and an exercise, almost always requires serious human intervention

From this perspective the term job also appears in a different way then in the case of classical grid implementations Classical approaches consider a job to be a single large statically linked binary executable fed by an input file, providing an output This approach is quite restrictive, and is not considered to be economic at all

On the other hand, if dynamic linking is used, there is no guarantee for the job to face the same execution environment as the one in which it has been complied The solution is pretty straightforward: Allow a directory structure to constitute a job The key advantage of the jobdir job format is that by involving sub-directory structures it allows much more freedom to the user, than a single executable, while it is possible to transfer not only a binary, but libraries, environmental variables, multiple executables, workflow information, or even license files to the place of the real execution

Figure 2 shows the general view of the ClusterGrid infrastructure which is one possible implementation of the GRM The core part of the infrastructure involving all machines, except the PCs, is referred to as the grid backbone infrastructure, or simply GBone The quality of implementation is determined by the software and hardware layout of the grid backbone

4 MONITORING

Monitoring is one of the most important functions that a grid operator has to deal with Monitoring, as a cross-section in the grid reference model, spans over multiple layers providing relevant information about the appropriate operation in real-time Monitoring helps forecasting errors, finding and repairing malfunctioning components and measuring the overall usability of the whole system

In the ClusterGrid infrastructure project an open-source monitoring system, MON [4] is used for supervising various aspects of the production grid: running services, system utilization, network utilization, free disk space, CPU accounting (which is tightly coupled to the monitoring system), availability of resources, etc

For storing the CPU accounting information RRD database is used RRD implements good quality data storage, and is also redundant against missed database updates without explicit user-level coding

5 USER SUPPORT

One of the key recognitions in the ClusterGrid infrastructure project was the precise definition of motivations: Not all grid providers are eager to use large computational resources On the contrary, there is no motivation power at all for most of the grid resource providers behind joining and also utilizing a large, shared resource at the price of offering self-operated computational power In most of the cases the user community and the provider community is quite different, and has minor overlapping between them Figure 3 charts a little illustration to the fact

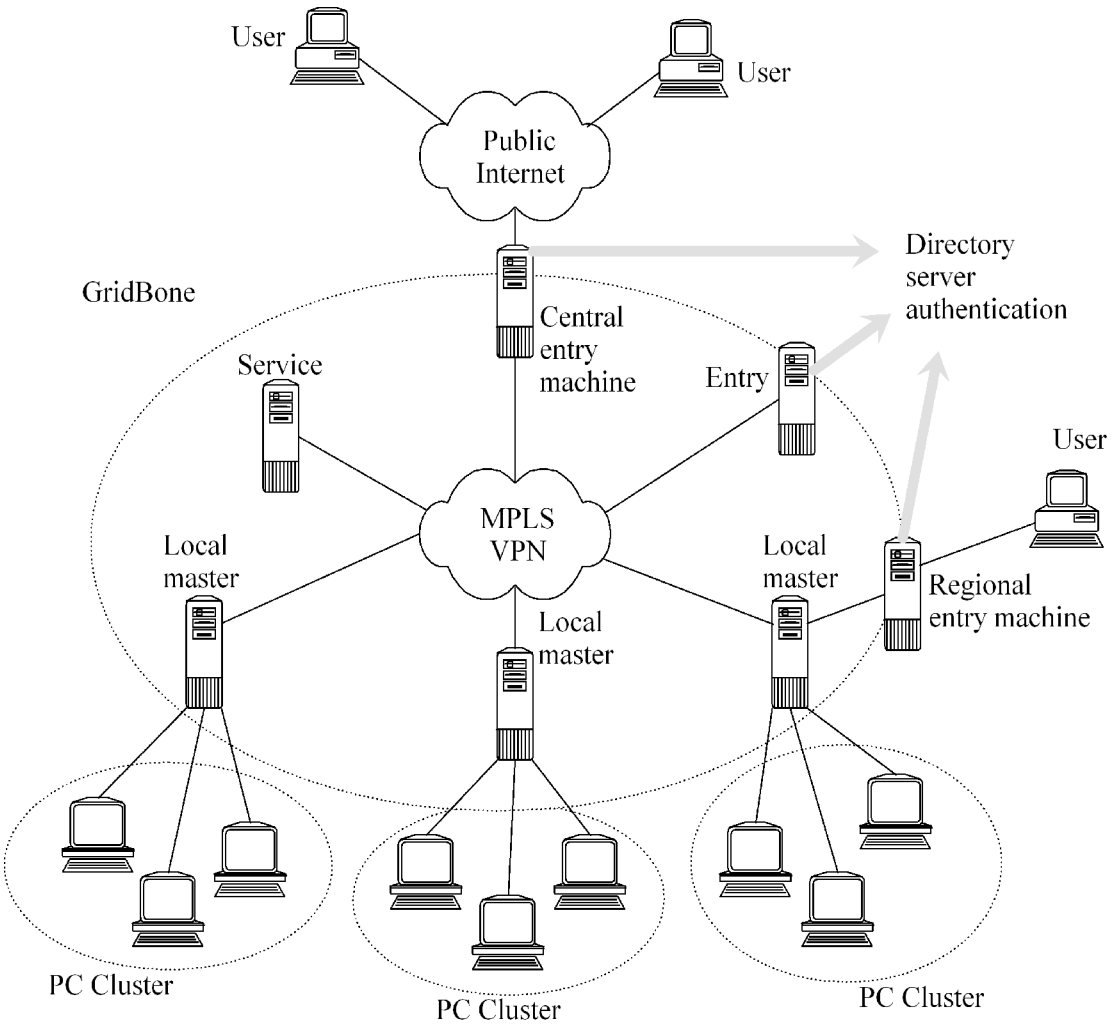

Figure 2: The general view of the ClusterGrid infrastructure project

When talking about a production grid service, the following participants can be identified:

- grid user: a person or an institute who has structurally a great need of compute power and uses the grid resources,
- resource provider: an institute who offers its compute resources into a common pool, and is not motivated by getting larger resource pool than he already has (money, or offering),

- grid service provider: a special institute who plays the role of a resource coordinator, gathers local resources, motivates grid resource providers, adds some extra service value to the raw resources, and integrates them into a single, large computational pool

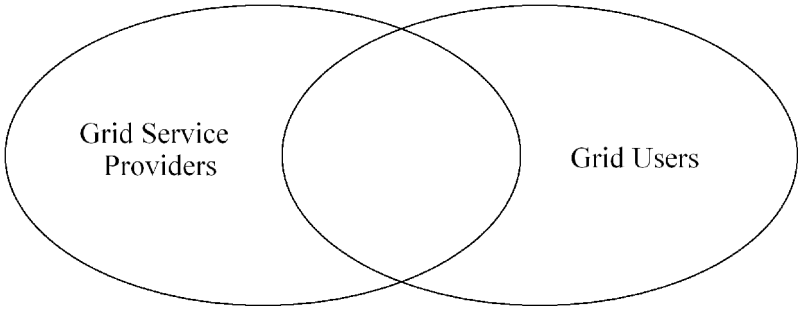

Figure 3: The difference between the grid user and grid provider communities There is a little overlapping between the two communities

6 THE FUTURE PERSPECTIVE

Implementing a production grid is a tedious task, but an elementary requirement not only inside of Europe, but all over the world Grid development must not stay at the level of research, but everything which turns out as a grid research result, should be immediately put into production to allow the users to test, evaluate it and declare if it is usable or not It should also be kept in mind, that the whole grid research is for the users, and every element within the infrastructure should directly or indirectly serve their real needs

Either as a research field, or as an infrastructure implementation and development challenge, the area faces a great perspective: there are a lot of questions need to be addressed, a lot of problems to be solved, and a lot of users to be served

7 CONCLUSIONS

In the paper the key implementation features of a production grid is surveyed with respect to the grid reference model The Hungarian ClusterGrid infrastructure has been built up following the above mentioned principles and as long as it has been a production grid system (the first in Europe, officially opened at July, 2003), it gives real proof to the usability and importance of the modular infrastructure implementation

There are many ways these principles can be elaborated, and there are a lot of questions still left open, but the user-driven grid research provides ambitious goals and requires well-tried principles and clean technical solutions

8 ACKNOWLEDGEMENTS

The authors would like to express their gratitude to the ClusterGrid development community (the former Technical Board) involving the members of the Hungarian Grid Competence

Center: István Botka, István Farkas, Gábos Gombás, Péter Halász, Zoltán Kalmár, Bence Kiss, Tamás Máray and Imre Szeberényi

The work and moral support of NIIF/HUNGARNET colleagues, namely Lajos Bálint and Miklós Nagy is also highly appreciated

The authors would like to say thanks for the efforts of Péter Kacsuk for propagating and advertising the results of the Hungarian ClusterGrid infrastructure projects outside Hungary Many thanks to him!

9 REFERENCES

[1] SEEGRID
[2] MPLS VPN
[3] GridSphere
[4] MON

10 AUTHORS BIOGRAPHY

PÉTER STEFÁN has MSc in Information Engineering, and has been working as a Solaris system administrator for 5 years He joined the NIIF/HUNGARNET Hungarian Supercomputing Center in 2001

From 2001 to 2003 he participated in numerous grid and high availability clustering projects, and experienced supercomputing technology from many aspects, such as network design, code porting and user support

Besides his system administrator activity, now, he is the leader of the Hungarian ClusterGrid infrastructure project, which aims at integrating more than 2000 PCs into a single, homogeneous, country-wide virtual supercomputer

FERENC SZALAI has M Sc in Physics, and

GÁBOR VITÉZ has M Sc in Information Engineering, and