Thinking in Grid - Initiatives for Hungarian AgriGrid System

Miklós Herdon [a], Péter Salga [b], Balázs Kónya [c], Róbert Szilágyi [d]

[a] *University of Debrecen, Hungary, Department of Business- and Agricultural Informatics, herdon@date.hu*

[b] *University of Debrecen, Hungary, Department of Business- and Agricultural Informatics, salga@thor.agr.unideb.hu*

[c] *Lund University, Sweden, Experimental High Energy Physics, balazs.konya@hep.lu.se*

[c] *University of Debrecen, Hungary, Department of Business- and Agricultural Informatics, szilagyi@thor.agr.unideb.hu*

Abstract

Grid is a resource sharing infrastructure providing uniform access to dynamic, heterogeneous systems and a common way of managing distributed computing, data, instrument, and human resources. It can maximize data and application usage without centralization and easily maintain and dynamically integrate different data.

Nowadays in Hungary we have dispersed agricultural and rural systems which have not complete but overlapping data, different databases and information resources which don't communicate with each other. Environmental-, meteorological-, growth-models and decision support often need enormous computing power but farmers have no supercomputing facility.

Emerging grid technology offers a powerful mechanism for assembling and processing the requisite data for different applications e.g. decision support, knowledge base, information- and monitoring systems. The one of most important demand is the information, knowledge and several different types of data sharing. The datagrid concept may provide good solution for the authorized access of data, integration of different resources and searching for data in different databases.

Our University takes part in the ClusterGrid project. The aims of this project to integrate the Intel processor based PCs into a single, large, countrywide interconnected set of clusters. The PCs are provided by participating Hungarian institutes, such as high schools, universities, or public libraries, the central infrastructure and the coordination is provided by NIIF/HUNGARNET. This ClusterGrid gives possibilities for application development.

But the background of our initiative is the NorduGrid system. This paper overviews our work with a focus on the Advanced Resource Connector (ARC) one of the first reliable and efficient production quality grid middleware developed by the Nordugrid Collaboration. This solution is a light-weight grid-package, designed to support a dynamic, heterogeneous grid facility, spanning different computing resources and user communities.

The presentation discusses the usable application development tools for distributed computing and overviews a set of possible application in the topic of agriculture and rural development. Also, the vision of Hungarian agricultural datagrid system is presented.

Key words: grid, cluster programming, decision support, knowledge base, data-integration.

1 Introduction

The last decade has seen a substantial increase in commodity computer and network performance, mainly as a result of faster hardware and more sophisticated software (*Fig. 1*). Nevertheless, there are still problems, in the fields of science, engineering, and business, which cannot be effectively dealt with using the current generation of supercomputers. In fact, due to their size and complexity, these problems are often very numerically and/or data intensive and consequently require a variety of heterogeneous resources that are not available on a single machine. A number of teams have conducted experimental studies on the cooperative use of geographically distributed resources unified to act as a single powerful computer (Buyya, 2002).

This new approach is known by several names, such as metacomputing, scalable computing, global computing, Internet computing, and more recently peer-to-peer or Grid computing. The early efforts in Grid computing started as a project to link supercomputing sites, but have now grown far beyond their original intent. In fact, many applications can benefit from the Grid infrastructure, including collaborative engineering, data exploration, high-throughput computing, and of course distributed supercomputing.

The stated goal of Grid computing is to create a worldwide network of computers interconnected so well and so fast that they act as one. Yet most of the time, this over-hyped catchphrase is used to describe rather mundane improvements that allow companies to manage their workload more flexibly by tapping into idle time on their computers.

The decision to build a distributed computing system to deal with this deluge of data predates the hype about Grid technology and is purely pragmatic: it would be difficult to fund the necessary computational power and storage capacity if it were concentrated on one site. If, on the other hand, the computations are distributed among the hundreds of institutes worldwide that are involved in the Grid, each institute can tap into national or regional funding sources to raise cash, spreading the pain (Economist).

2 What is Grid?

Computer systems using multiple processors have existed for decades in various forms, the most common of which have been multiprocessing servers and mainframes. The advent of inexpensive, high-performance processors has provided impetus to the development of multiprocessor designs.

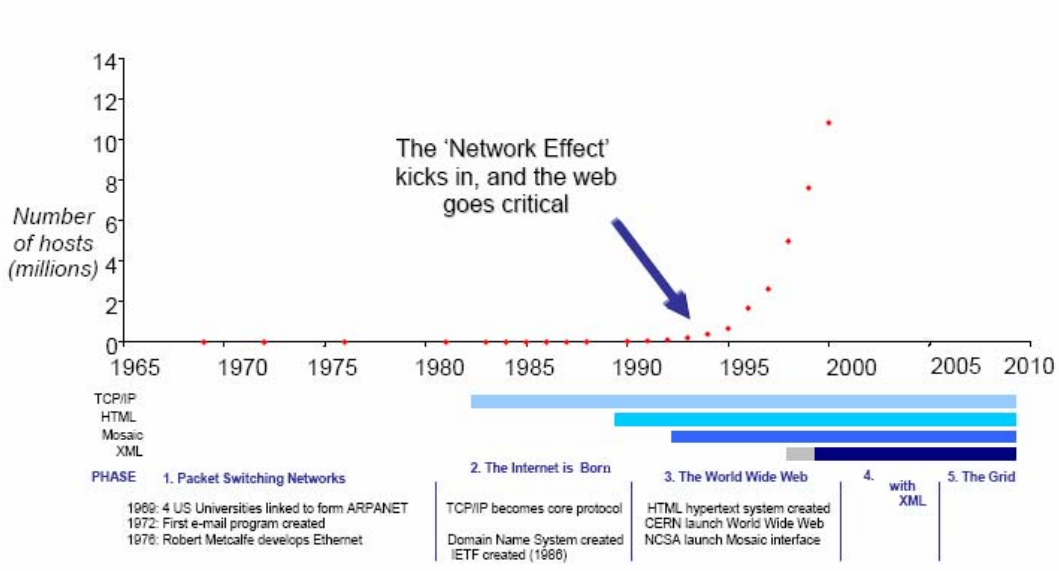

Fig. 1. Past, Present, Future (Buyya, 1999)

The terms *'Parallel processing', 'parallelization'* or *'distributed programming'* all refer to the system where a complex task is broken up into many subtasks that are to be run in parallel. Each subtask is then assigned to a CPU on the network and the results are combined.

Distributed programming uses a collection of computers connected over a network to solve a single problem. Programming multi-computers requires models which are different from normal systems. The programmer must be able to transfer data between different parts of the program through a shared memory space and to coordinate efforts through an inter-process communications system capable of communication between interconnected CPUs.

Distributed programs achieve the following:

- Increased processing speed by using more than one computer at a time.
- Potential for improved reliability when additional computers can compensate for the failure of one
- Allowance for some problems, like remote data acquisition, to succeed in a distributed environment

A *cluster* is a group of individual, stand-alone computers that work together in parallel way and that outside systems view as a single computing resource. The individual systems (*nodes*) that make up the cluster communicate with each other via high-speed connections such as Gigabit Ethernet, ATM or a proprietary link. For easier management, clusters use special software to coordinate and manage their activities, depending on how they are used. Clusters are particularly well suited to meeting the needs of high-availability, load balancing and scientific computing (Binstock).

Cluster computing using machines connected in a finite physical network. These are PCs similar in both hardware and software. In order to solve massive computational problems most networks are not big enough. The Grid computing is the answer to this problem.

While clusters are groups of computers tied together as a single device, *Grid*s consist of multiple systems that work together while maintaining their distinct identities.

Grids virtualize resources, such that a user logging on at a site would view the processors at the other sites as though they were part of the local system. In fact, of course, they are not. They are in a separate administrative domain but available as a virtual local resource to any node on the Grid. *The constituent systems are administered separately and are physically distinct from each other.* This is not true for any of the previous designs developed so far.

Like clusters, Grids use high-speed interconnects to link the various systems. Because of the latencies inherent in long distances, however, Grids often partition tasks to minimize the need for long-distance messaging. Grids enjoy common benefits with clusters: they can be built from inexpensive, off-the-shelf parts, and they can be expanded nearly endlessly, that is why often called the poor man's Supercomputer (Burger).

3 Application of Grid technology

Grid computing is made up of computational and data intensive problems, and providing common interface for instruments. The computational aspect focuses on reducing execution time of applications that require large amounts of computer processing cycles. Data intensive problems require large scale data management methods to transfer the data needed for solving the problem to the machine assigned to solve it. Data intensive applications such as High Energy Physics and Bioinformatics require both computational and data management solutions to be present in Grid computing solutions. Another power of Grid - which emphasized rarely in literature - that it can provide a uniform interface for researchers to common use instruments and infrastructure.

Grid computing can use the Internet to borrow unused CPU cycles and storage from millions of systems

across a worldwide network. This flexible, readily accessible pool can then be harnessed by anyone who needs it, much as power companies and their users share the electrical grid. Grid computing leans more to dedicated tasks, such as single large medical and engineering problems, rather than for general, everyday jobs. Sun defines a computational grid as "a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to computational capabilities." The computers on a Grid can be of many different OS and hardware platforms.

Generally Grids are classified by function:

- *Computational Grid*s (including CPU scavenging grids) referred - Computational Grids typically gain and lose machines at unpredictable times as interactive users start or stop using their machines, new machines are purchased, machines are removed from the network, or break down. Cycle-scavengers move jobs from machine to machine as necessary to allow the smooth running of the job and the network being scavenged;

- *Data Grid*s - Cycle-scavenging systems use machines purchased for other purposes to run batch jobs at night, weekends, and other idle times. A data grid is a Grid computing system that considers access to distributed data as important as access to distributed computational resources. Many distributed scientific and engineering applications require access to large amounts of data -- often terabytes or even petabytes of data.

It's expected that in the future applications will require even more widely distributed access to data. Data Grids will have to support scientific collaboration in a virtual environment allowing access around the world by many people. A Grid is a distributed collection of computer and storage resources maintained in a Virtual Organization (VO). Any of the authorized users within that VO has access to all or some of these resources, and is able to submit jobs to the Grid and expect responses. The data-grid concept is very important in agricultural and rural application, where the sharing and integrating of data have much more importance than the processing power (Burger).

4 ClusterGrid in Hungary

The ClusterGrid project aims to integrate the Intel processor based PCs into a single, large, countrywide interconnected set of clusters. The PCs are provided by participating Hungarian institutes, such as high schools, universities, or public libraries (among them two clusters at University of Debrecen), the central infrastructure and the coordination is provided by NIIF/HUNGARNET. Every contributor uses their PCs for their own purposes during the official work hours, such as educational or office-like purposes, and offers the infrastructure for high-throughput computation whenever they do not use them for any other purposes, i.e. during the nights and the unoccupied week-ends. The combined use of "day-shift" (i.e. individual mode) and "night-shift" (i.e. grid mode) enables us to utilize CPU cycles (which would have been lost anyway) to provide firm comutational infrastructure to the national research community.

The infrastructure is built on a grid backbone (GBone) which consist of local masters, i.e. servers deployed at all institutes, and a central entry point which provides central services and user interface to the grid. The PCs of the individual clusters (computer labs) "hang" on local masters using network root file system based worker layout. ClusterGrid utilizes the good quality Hungarian Academic Network and it is located in a separated network segment, i.e. Multi-protocol Label Switching Virtual Private Network.

5 Why NorduGrid?

The main Grid systems in Europe were observed by experts of our workgroup. Our main aspects were the international connectivity, functionality and functioning, and the user- and admin-friendly configuration and management. The *NorduGrid* project and the *NorduGrid ARC* (Advanced Resource Connector) fulfil all of the conditions, and we found it is the first reliable and efficient production quality grid middleware in Europe.

The NorduGrid architecture's basic components are the user interface, information system, computing cluster, storage element, and replica catalog. The NorduGrid's user interface includes high-level

functionality namely resource discovery and brokering, Grid job submission and job status querying. NorduGrid thus does not require a centralized resource broker. The user interface communicates with the NorduGrid grid manager and queries the information system and replica catalog. Users can install the user interface client package on any machine, using as many interfaces as they need.

The information system is a distributed service that serves information for other components, such as monitors and user interfaces. The information system consists of a dynamic set of distributed databases that are coupled to computing and storage resources to provide information on a specific resource's status and operates on a pull model: when queried, it generates requested information on the resource locally (optionally caching it afterward). Local databases register to a global set of indexing services via a soft-state registration mechanism. For direct queries, the user interface or monitoring agents contact the indexing registries to find contact information for local databases. The computing cluster consists of a front-end node that manages several back-end nodes, typically through a private closed network (Kónya, 2004).

The software component is a standard batch system, with an extra layer that acts as a grid interface and includes the grid manager, the GridFTP server and the local information service. Although Linux is the operating system of choice, Unix-like systems, including Hewlett-Packard's UX and Tru64 Unix, can be used as well. The NorduGrid does not dictate batch system configuration; its goal is to be an add-on component that hooks local resources onto the Grid and lets Grid jobs run along with conventional jobs, respecting local setup and configuration policies. The cluster has no specific requirements beyond a shared file system (such as Network File system) between the front- and back-end nodes. The back-end nodes are managed entirely through the local batch system; no grid middleware is required on them. To register and locate data sources, the Globus project's replica catalog was modified to improve its functionality. The catalog's records are primarily entered and used by the grid manager and the user interface. The user interface can also use the records for resource brokering (Kacsuk et al, 2004).

The NorduGrid Toolkit is freely available at www.nordugrid.org as RPM distributions, source tar-balls, and as CVS snapshots and nightly builds.

Fig. 2. – the Grid Monitor - shows the actual activity of clusters involved in NorduGrid and other technical data. In Nordugrid involved 40 clusters, ~4000 CPUs, 10 countries. The storage capacity: 42 TB total, 28 TB free and there are more than 1000 ARC-users worldwide.

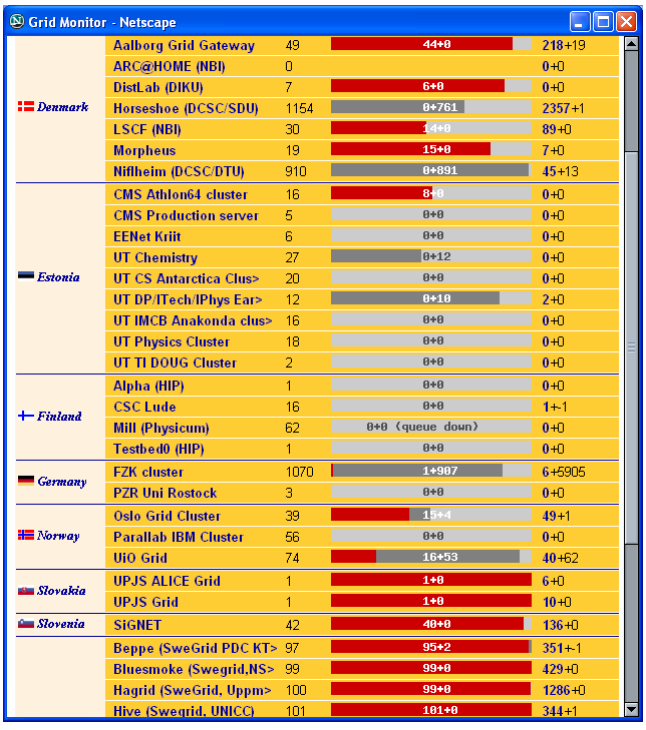

Fig. 2. The NorduGrid Grid Monitor

6 Application of Grid technology in Agriculture and Rural development

There are lot of decision situations in the life of a farm which needs the knowledge of a complex system of agricultural, meteorological, chemical, and geographical data. Not an easy mission to support similar type of decisions because the necessary data stored in heterogeneous databases managed by several different ownerships. No chance whatever to centralize data-storage units but there is a strong necessity of integration of various databases. Another effect in this topic is the commercial competition which is an additive difficulty for information-support.

The Grid is a plausible way for data integration because it serves:

- Uniform interface for queries
- High storage capacity
- Processing power for optimization programs (neural networks, genetic algorithms)
- Accessibility rights and policies to data
- Accounting system for data-sellers and costumers
- Sharing of instruments

With the help of AgriGrid system the farmer can call information for irrigation or spraying based on images taken by his mobile phone and the geographical position of area as shown in the *AgModel* project (Laurenson et al, 2004). This system adds the following advantages to the Grid:

- Maximize data and application usage without centralization
- Easy Maintenance
- Easy updates of data and applications
- Easy addition of new data and applications
- Flexible and dynamic integration of data and applications

This model can help to integrate the decision data for agriculture and usable also in development phase, in this way we can reduce the costs of development (Ninomiya and Laurenson, 2003).

Relations between rural citizens, business and governments, as well as the relations between rural regions and global market have growing importance and complexity. In this area the information technology plays a very important role to support the desired data. The grid technology enables a knowledge sharing infrastructure (Knowledge Grid) which enables authorized data access and publication for rural population.

7 Initiative of Hungarian AgriGrid

At the University of Debrecen, Department of Business- and Agricultural Informatics a Fedora C2 Linux cluster was built. We use the PBS based *Torque* cluster management tool, and the Advanced Resource Connector and we have connection to the NorduGrid system. In the cluster multiple programming environments were installed: C, C++, MPI and Java. The distributed treatment of Java programs is

managed by *JBoss* application server.

The following is a clustering feature overview for JBoss 3.0:

- Automatic cluster membership discovery.
- Fail-over and load-balancing features for JNDI, RMI, Entity Beans, Stateful Session Beans with in-memory state replication, and Stateless Session Beans.
- Pluggable load-balance policies.
- HTTP session replication with Tomcat and Jetty (CVS HEAD only).
- Dynamic JNDI discovery. JNDI clients can automatically discover the JNDI InitialContext.
- Cluster-wide replicated JNDI tree.
- Network Boot.
- Farming: Distributed cluster-wide hot-deployment.

These features make the JBoss 3.0 application server to a very useful cluster programming tool and enable the application of portable code in distributed environments.

We planned and designed the architecture to satisfy the needs of future users. These needs constitute a general guiding philosophy:

- Start with something simple that works.
- Avoid single points of failure.
- Give resource owners full control over their resources.

Ensure that developed software can use the existing system and, eventually, other preinstalled versions.

8 Conclusions

The potential of computer science has always been hampered by the inability to adequately address massive processing and data volume issues. No matter how fast a CPU is or the data throughput rate, our imaginations come up with new applications that exceed the existing technology or budget.

Grid computing technology has the potential to alleviate processing capacity and cost barriers. A Grid can solve problems that can't be approached without an enormous computing power. Computers will collaborate rather than being directed by one managing computer. Ultimately, the future may bring pervasive computing; computers will be saturating our environment without our direct awareness. Recent Internet over power grid developments may further increase Grid computing use by making high speed connection ubiquitous.

The Java programming language successfully addresses several key issues that accelerate the development of Grid environments, such as heterogeneity and security. It also removes the need to install programs remotely; the minimum execution environment is a Java-enabled Web browser. Java, with its related technologies and growing repository of tools and utilities, is having a huge impact on the growth and development of Grid environments. From a relatively slow start, the developments in Grid computing are accelerating fast with the advent of these new and emerging technologies. It is very hard to ignore the presence of the Common Object Request Broker Architecture (CORBA) in the background. We believe that frameworks incorporating CORBA services will be very influential on the design of future Grid environments. The two other emerging Java technologies for Grid and P2P computing are Jini and JXTA.

The Jini architecture exemplifies a network-centric service-based approach to computer systems. Jini replaces the notions of peripherals, devices, and applications with that of network-available services. Jini helps break down the conventional view of what a computer is, while including new classes of services that work together in a federated architecture. The ability to move code from the server to its client is the core difference between the Jini environment and other distributed systems, such as CORBA and the Distributed Common Object Model (DCOM) (Baker et al, 2002).

The main application areas of today's Grids are the biotechnology, high energy physics, but the structure and demands of agriculture makes it a very appealing field for Grid technology. The enormous processing power of clusters is not the most applicable property in agriculture, the main advantages are the common interface, the sharing of instruments and the data integration facility of data-grid technology.

The Hungarian initiative of Agrarian Grid System was built by our team at the University of Debrecen. This system can be an appropriate ground for integration of Hungarian agricultural data and the development of an agricultural decision support system.

8 References

Baker M., Buyya R., Laforenza D. 2002. Grids and Grid technologies for wide-area distributed computing. Softw. Pract. Exper.

Binstock, A. Multiprocessors, Clusters, Grids, and Parallel Computing: What's the Difference? http://www.intel.com/cd/ids/developer/asmo-na/eng/95581.htm

Burger, T. W. Trends in Distributed Computing, http://www.intel.com/cd/ids/developer/asmo-na/eng/95223.htm

Buyya, R. (Ed). 1999. High Performance Cluster Computing: Architectures and Systems. Volume 1, Prentice Hall, Old Tappan.

Buyya, R., 2002. Economic-based distributed resource management and scheduling for Grid computing. PhD Thesis, Monash University, Melbourne, Australia.

Economist Magazine, Oct 2004. One Grid to Rule Them All.

Kacsuk, P., Kónya, B., Stefán, P., 2004. Production Grid Systems and Their Programming. Lecture Notes in Computer Science, Volume 3241, Page 13.

Kónya, B., 2004. Advanced Resource Connector (ARC) – The Grid Middleware of the NorduGrid. Lecture Notes in Computer Science, Volume 3241, Page 10.

Laurenson, M. R., Yamakawa, A., Meng, H., Kiura, T., Wang, J. and Ninomiya, S., 2004. Integration of Data Broker Web Services for Agricultural Grid. AFITA 2004, Bangkok, Thailand.

Ninomiya, S., Laurenson, M., 2003. A Grid for Efficient Decision Support in Agriculture. International Symponium Grid Computing, Taipei, Taiwan.