Utility Computing and Clusters

CHAPTER **2**

Understanding Grid

There is always a lingering doubt in everyone's mind, as to what Grid is exactly, and how it differs from other well known architectures such as Cluster and P2P architectures. In this chapter, we will highlight the true nature of Grid and Clustering architectures. We will also look at other buzzwords such as 'on-demand' and 'utility' computing.

There is a somewhat hazy notion and understanding of Grid and the types of Grids among many IT technologists. Sometimes they do not find a clear demarcation between grids and other related technologies like clusters. We will examine the differences between the Clusters and Grid.

"Oops … Am I in Grid now?"

Another thing that often needs explanation is the concept of On-Demand and Utility Computing Model, which is often termed as Grid Computing.

The Clustering of Servers has been with us for about two decades and it is one of the most widely understood, being deployed in scientific, research and commercial worlds. We are well versed with the concept of aggregating the servers, and view them as the single system image (SSI). Now, with the gradual adoption of grid technologies, one often wonders if grid works with clusters or if it replaces the clustering. How does grid change the whole perspective of the sharing of the servers?

The Nature of Grid

Grid Computing is an emerging infrastructure that aims at providing a mechanism for sharing and coordinating the use of diverse computing resources. The word '**Grid'** is often used as an analogy with the electric power grid, which provides access to electricity.

As Grid Computing makes long strides and impacts many organizations in the IT world with its utility-like access to computational resources, a question remains in everyone's mind if Grid Computing can become similar to the Electric Power Grid of the 20th century. While the state of Grid Computing is still in its infancy, there are definite signs of similarities according to Rajkumar Buyya and Madhu Chetty [1.2], the researchers from University of Melbourne, Australia.

As seen in Table 2.1, there many apparent similarities in the electric power grid infrastructure and computational grid structure. However, the computational grid is more varied and more complex than electric grid. The existence of hardware components and user specific software components and applications make all the difference. Nevertheless, the comparison is worth noting.

Parameter	Electrical Power grid	Computational Power grid
Resources	Heterogeneous: thermal, hydro, wind, solar, nuclear, others	PCs, workstations, clusters, and others; driven by different operating and management systems
Network	Transmission lines, underground cables. Various sophisticated schemes for line protection.	Internet is the carrier for connecting distributed resources, load, and so on.
Analogous quantities	Bus Energy transmission Voltage	Node Computational transmission Bandwidth
Power source	Power station (turbo generators, hydro generators), windmill	Grid resource (computers, data sources, Web services, databases)
Load type	Heterogeneous application devices: for example, mechanical energy for fans electricity for TVs, heat for irons	Heterogeneous applications: for example, graphics for multimedia applications, problem solving for scientific or engineering applications
Security / safety	Fuses, circuit breakers, and so on	Firewalls, public-key infrastructure, and PKI-based grid security
Storage	Only storage for low-power DC using batteries.	No storage of computational power is possible.

Parameter	Electrical Power grid	Computational Power grid
Automated accounting	Advanced metering and accounting mechanisms are in place	Local resource management systems support accounting. Resource brokers can meter resource consumption
Standards body	Many standardization bodies exist for various components, devices, system operation, and so on. (For example, the IEEE publishes standards on transformers, harmonics, and so on.)	Forums such as Global Grid Forum and the P2P Working Group promote community practices. The IETF and W3C handle Internet and Web standardization issues.

Source: Rajkumar Buyya and Madhu Chetty

Table 2.1 *Electrical and computational Power grids: A comparison.*

With sudden interest in the grid and grid-related technology, many IT vendors and analysts are creating their own vision, definition and solutions in the grid space. Grids have moved from the obscurely academic to the highly popular. We read about Compute Grids, Data Grids, Science Grids, Access Grids, Knowledge Grids, Bio Grids, Sensor Grids, Cluster Grids, Campus Grids, Tera Grids, and Commodity Grids as proposed by various IT companies and researchers. There are so many flavors and variations, based on the functionality and sometimes based on the understanding. Many IT vendors freely term their solutions as Grid technologies and try to fit them into some category.

Let's ponder what Ian Foster has to say, *"Ultimately the Grid must be evaluated in terms of the applications, business value, and scientific results that it delivers, not its architecture. Nevertheless, the questions above must be answered if Grid computing is to obtain the credibility and focus that it needs to grow and prosper."*

Ian Foster's 3-point checklist for a grid computer:

Coordination of Distributed Resources - *Grid controls and integrates different resources and users within different control domains – for example desktops versus large computers, different units of same enterprise and different enterprise. It also addresses the issues of security, policy, membership and payment.*

Using Standard Pen, General Purpose Protocols and Interfaces - *Grid Computing is based on various protocols and interfaces. These protocols and interfaces control the authentication, resource discovery, and resource access.*

Quality of Service - *Grid aims at delivering at non-trivial quality services in terms of response time, throughput, availability and security. This is the motivation for the community to move towards the grid-computing era and meet the ever-increasing user core application demands. Grid computing becomes more of a utility from the user's perspective.*

As noted, Ian Foster's 3-point checklist provides a broad guideline. With these definitions in mind, many IT vendors are coming up with varied Grid Architecture solutions. At the same time, some vendors are pitching their cluster solutions. This happens because the clustering (sharing servers) concept and clustering solutions loosely fit into the broader sense of grid architecture.

Grid Architecture

At a very high level, Grid Architecture can be best represented in component layers. This layered architecture is by no means a rigid dictation of the components, but it is extensible and follows the open architectural framework.

As shown in the Figure 2.1, the Layered Grid Architecture, Grid Architecture follows the hourglass model, where the narrow neck of hourglass defines a small set of core abstractions and protocols such as TCP and HTTP.

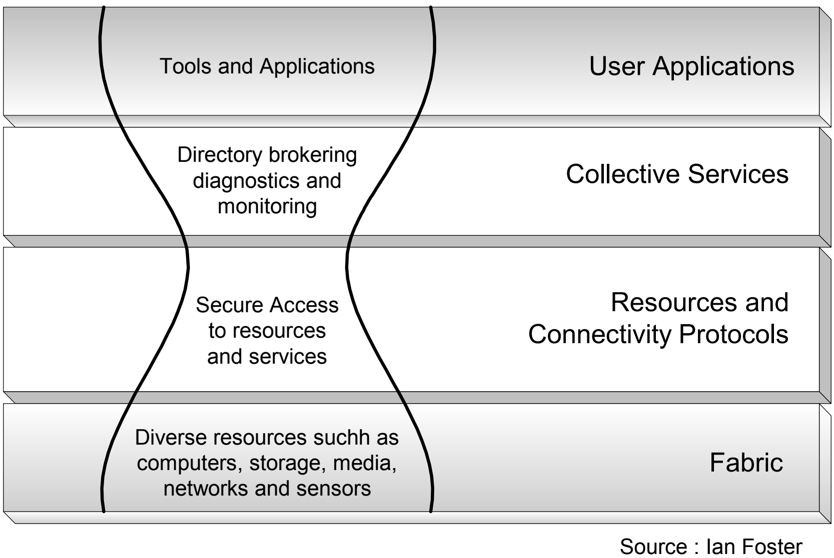


Figure 2.1 *The Layered Grid Architecture*

While the base of the model conveys the different underlying technologies, the top of model shows high-level behaviors that translate into services and user applications.

The *fabric* layer provides the resources to which the shared access is controlled by the grid protocols. The resources normally include physical and logical entities. Physical entities are resources like storage systems, catalogs, servers and network resources. The resource may be a logical entity like distributed file system, computer cluster or distributed computer pool and database systems to store structured data. The Grid mechanism normally permits the capability for the resource management, which involves discovery and control.

The *connectivity* layer defines core communications and authentication protocols required for Grid specific network transactions. These protocols enable the exchange of data between fabric layer resources. The *resource* layer, based on the connectivity and authentication protocols, controls the access resources.

The *collective services* layer deals with the directory brokering services, scheduling services, data replications services and diagnostics/monitoring services. These services are not associated with any one specific resource but focus on interactions across resources. The programming models and tools define and invoke the collective layer functions. This layer is a key component in the whole grid architecture and its functioning. This is the layer that glues all the resources together in expedient exchange.

The top layer, *User Applications*, comprises the user applications that operate within a VO (virtual organization) environment.

The components in each layer share common characteristics but can build on new capabilities and behaviors provided by the lower layer. What we get out of this model is the flexibility with which Grid Architecture can be extended and evolved. This is the precise reason why grid architecture is taking shape and form based on the guidelines developed by the visionaries and grid standard forums.

While looking at vision and broad framework the Grid Architecture provides, many of the IT vendors are steadily designing their own strategy and services. Many IT leading vendors such as IBM, SUN, Oracle, Dell and HP offer various and diversified solutions that enable grid computing. In addition, there are many smaller companies, which focus on an innovative and specific resource, or issue and they are able to design many solutions. Such groups include companies like Avaki, Axceleon, DataSynapse, Ejasent, Enigmatec, Entropia, GridFrastructure, GridIron, GridSystems, GridXpert, Powerllel, Tsunami Research, The Mind Electric and United Devices.

With Grid Technology being a new and unexplored arena for many of us, we tend to associate the technology with variations of what we have seen and what makes us feel comfortable. One such confusion is the perception of a large cluster to be a synonym to the grid framework. Clusters have been with us good number of years; we understand very well the architecture and functionality. Clusters focus on a single objective and are a collection of servers with homogenous nature.

We will cover a more detailed examination of cluster and its architecture in later sections of this chapter.

Grid Concepts and Components

Grid is a collection of machines, nodes, resources, members, donors, clients, hosts, engines and many other such items. Grid architecture provides the necessary opportunity for executing applications with optimum exploitation of grid resources. Next, we look at these resources. Figure 2.2 shows the key components in the grid environment.

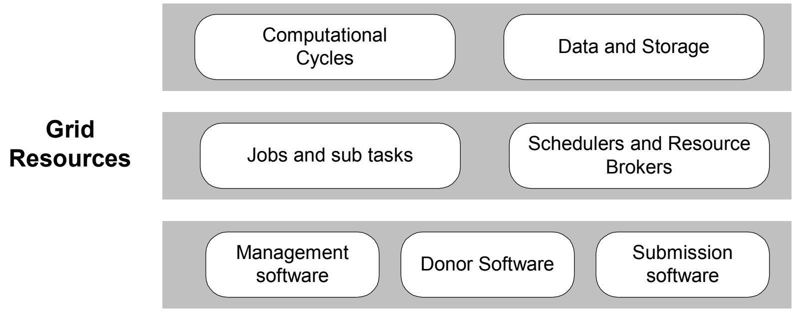


Figure 2.2 *Grid Components*

Computational Power

Processors are the main components that provide the computing cycles. Processors come in different architecture, speed and instructional set design. The computing cycles are the main target for exploitation in a typical grid environment. There are three ways in which the applications utilize the computing cycles.

- First, the processing power can be harnessed by running an application on available machines in the grid rather than locally.
- Second, the user application can be designed to split its computation tasks or work and execute them in parallel running on different machines or processors.
- The third, involves running an application many times on many different machines in the grid. Grid architecture provides a suitable software and framework to make this work.

Data and Storage

The next significant resource that an application would like to utilize is the data or storage resource. The data is available in a variety of forms such as structured databases, data files, XML documents and others. Storage is attached to various machines (or servers) in the form of directly attached file systems, or network attached file systems (NFS). By providing a robust storage array attached to multiple servers in the grid environment and creating a unified file system that is attached to multiple servers, the data or files can be made accessible to many servers and thereby to the user applications. Users do get local access to such data. It also becomes easier for the users to reference the data in the grid.

In the case of structured database systems, the data can be made available in two forms: the Clustered Database and the Federation of Databases. In a clustered solution, the same data can be accessed by referring to any server in the cluster configuration. Federation of database is more like an association of data to form a comprehensive database. In a single query, user application is able to access multiple database resources.

At the same time, many grid software solution vendors are designing some kind of unifying software, which controls multiple data sources, and project a single data view for the application or user access.

Communication

In the recent past, we have witnessed a rapid growth in server-to-server and server-to-storage communication capability. 10GbEthernet and Inifniband technologies are becoming very common. With the availability of these high bandwidth and low latency communication channels, huge data is moved across the servers in the grid environment. Communication within the grid for the purpose of sending jobs and data required for those jobs is now faster. This facilitates the transfer of job execution to other machines in the grid. Bandwidth is a critical resource for effective communication and optimum functioning of grid.

Jobs and Applications

Even though we talk of applications and transactions, ultimately the application is executed in terms of jobs and sub-jobs. As the application execution is split into multiple jobs, those jobs can effectively be moved to different processing centers in the grid. In the grid environment, processing centers are the servers or processors. Thus, a grid application that is designed or organized as a collection of jobs is quite suitable for the grid environment. These jobs can suitably be scheduled and executed within the grid and take advantage of multiple processing resources.

Schedulers / Resource Brokers

Schedulers are other key components in the grid environment. They are also sometimes called 'resource brokers'

The user may select a machine suitable for running his/her job and then execute a grid command that sends the job to the selected machine. More advanced grid systems would include a job "scheduler" of some kind that automatically finds the most appropriate machine on which to run any given job that is waiting to be executed. Schedulers react to current availability of resources on the grid. These schedulers, besides carrying the job to other machines in the grid environment, have the capability to understand the resource availability on the idle machines.

In the grid system, any machine that becomes idle would typically report its idle status to the grid management node. This management node would assign the idle machine the next job that is satisfied by the machine's resources. Scavenging of CPU cycles is usually implemented in a way that is unobtrusive to the normal machine user. If the machine becomes busy with local non-grid work, the grid job is usually suspended or delayed. This situation creates somewhat unpredictable completion times for grid jobs, although it is not disruptive to those machines donating resources to the grid.

Grid Software Components

It is important to understand that it is the software that makes the realization of Grid Computing. Even though we have many suitable hardware components lined up in the grid environment, it is the software component that glues all of these hardware resources into one usable grid framework. There are various software components, which run the basic functioning of the grid system. The important software components include management software, donor software and submission software.

A typical application when executed, functions on a server and uses the resources such as memory, processes and storage attached to the server. When the same application intends to utilize the resources from other servers (machines) in the grid, it needs to interact with the middleware and its API's in order to extend its reach to utilize the resources.

Management Software

The Management component keeps track of resources available to the grid and also keeps track of the users or applications that can use them. This component decides the computing locations, which are available for the jobs execution, decides the assignment pattern and keeps track of utilization levels of the servers.

This information is used to schedule jobs in the grid. Such information is also used to determine the health of the grid, alerting personnel to problems such as outages, congestion, or over commitment.

Donor Software

Each of the Grid machines participating in the grid framework typically contributes the resources. It needs to have some kind of agent grid software that manages the grid's use of its resources. That software also manages the authentication procedures. The participating machine is a donor machine and it will have donor software.

Submission Software

Usually any member machine of a grid can be used to submit jobs to the grid and initiate grid queries. However, in some grid systems, this function is implemented as a separate component installed on "submission nodes" or "submission clients." When a grid is built using dedicated resources rather than scavenged resources, separate submission software is usually installed on the user's desktop or workstation.

Open Grid Services Architecture

The Grid software is a crucial piece in combining multiple and varied resources into a homogenous and usable computing platform. Since there are many different operating systems and hardware components involved, standards and suitable API for the applications to interact are essential.

Over the years, many standards for grid computing have evolved and many leading researchers and IT vendors have contributed common specifications. The Open Grid Services Architecture (OGSA) represents an evolution towards a Grid system architecture based on Web services concepts and technologies. With the release of Globus Tool Kit 3.0 (GT3), the

Globus Project now offers an open source implementation of version 1 of the OGSI (Open Grid Services Infrastructure) Specification, a key building block in the OGSA framework.

GT3 includes standards and software that provides Grid Security (GSI), remote job submission and control (GRAM), high-performance secure data transfer (GridFTP), and consistent interfaces to system and service information (MDS). All of the components provided by the Globus Tool Kit Version-2 for building Grid infrastructures and for developing Grid applications remain available in GT3. GT3 has strong support from vendors.

Thus, the OGSA effort provides basic services and the grid community is developing services to sit on its top. OGSA describes a set of implementation and platform independent protocols and services. These define technology and infrastructure of the next generation of Grids defined in terms of Grid Services, closely allied to Web Services, which supports the sharing and coordinated use of diverse resources in dynamic distributed virtual organizations.

What is WebService?

A Web Service is an interface that describes a collection of operations that are network accessible. They are built on XML based technologies and messaging. They use XML schemas to mark-up and describe services and their operations. Web Services commonly use the Simple Object Access Protocol (SOAP) as a communication protocol over HTTP- when this is not appropriate for performance reasons, a different protocol may be used.

It is the GSA that pulls together Open Grid Architecture and Web Services to form Grid Services. Grid Services are dynamic, transient, and have state (i.e., have a finite lifetime) and are defined by a well-defined set of interfaces and behaviors. OGSA interfaces and behaviors describing Grid Services can be written in an XML schema based WSDL document – other implementations are possible.

Perception of Grid

Since the Grid Architecture involves interoperability and sharing of resources within enterprise and across enterprises, we often wonder if the Grid is synonymous with Internet or if the grid is alternative to internet. Rather, the Grid is a set of additional protocols and services that build on Internet protocols and services to support the creation and use of computation and data-enriched environments. In addition, we need to note that any resource that is "on the Grid" is also, by definition, "on the Net."

The Grid is perceived as a source of free CPU cycles. However, Grid computing does not imply unrestricted access to resources; rather it is about controlled sharing. The CPU cycles are normally used and controlled by resource policies defined. Usage of such resources depends on group membership, ability to pay, and so forth.

Grid experts believe that the Grid makes high-performance computers superfluous. The thousands of processors that may be accessible within a Virtual Organization represent a significant source of computational power. This does not imply, however, that traditional high-performance computers are obsolete.

Where Are We on the Grid?

As the Grid is gradually moving into the commercial world, many of the standards are finalized and many of niche software players are churning out software components needed for grid integration.

There are many IT vendors, in every aspect of the grid technology, who have released various strategies and solutions. The Figure 2.3 gives an idea of various players in the grid technology.

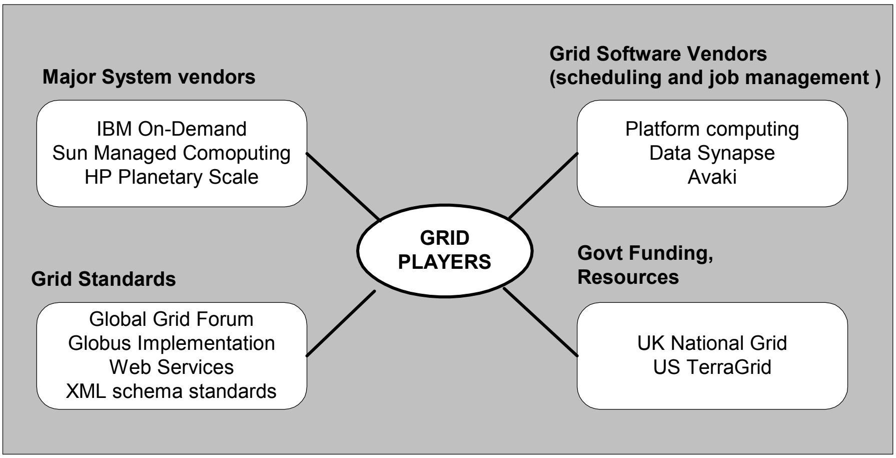


Figure 2.3 *Grid Universe and the Key Players*.

For the success and wide adoption of any new technology or new orientation, commercialization efforts are very important. For the Grid to become a widespread technology, in the same way the Internet became a mainstream technology and thee communication media, it needs to enter or penetrate into the commercial world. Commercialization provides necessary impetus for growth and funding.

There are many signs of Grid commercialization, even though they are at early stages. Some of examples of major business areas where Grid computing is seen as a good fit:

- Life Sciences – Grid is used for analyzing strings of biological and chemical information.
- Financial Services – For running long running, complex financial models and arriving at decisions that are more accurate.
- Higher Education – For enabling advanced, data and compute intensive research.
- Engineering Services – In the automotive and aerospace industry, for collaborative design and data intensive testing.
- Government – For enabling multi-departmental computing projects in both civil and military agencies.
- Collaborative Games – where many players participate simultaneously and use the computing resources for playing online games.

Next, we narrate an interesting case study, which shows the gradual adaptation of this new and revolutionary grid technology into the commercial world.

Compute Backbone - JP Morgan Chase's Grid Story

Grid Computing is extending its reach beyond scientific and technical users towards large-scale commercial enterprise, as evident from the recent adoption of grid technologies at JP Morgan Chase. JP Morgan is one of the largest financial services companies in the US. With technology partners, Platform Computing and Egenera, JP Morgan has been working on what it calls 'Compute Backbone' initiative, service based architecture intended to support its capital market business.

JP Morgan and Platform Computing has collaborated and

Compute Backbone - JP Morgan Chase's Grid Story

developed Platform Symphony product, which is a policy driven, real-time application execution layer, including the workload orchestration and service provisioning capabilities. It is used for grid enabling, testing and deploying the applications, and then managing them over a virtualized production environment, or 'service layer' as JP Morgan terms it. While the Platform computing contributed to the grid software layer, the other two significant contributors are the Egenera Blade servers and Linux operating system. Introduction of Linux operating system helped JP Morgan to standardize its software builds. The adoption of blade server technology provided JP Morgan a modular approach to grow its server hardware farm. With the blades platform, they are able to add blade-by-blade as the computing power need grows.

The grid set up is run as a service for internal users, and is charged for by usage. Users can buy CPU time on an hour-by-hour basis rather than having to own the CPU's for 24 hours a day, 365 days a year. It has opted for electricity-style pricing model – pay only for what you use. There are currently 500 Intel processors running the Computer Backbone, divided between New York and London. The Platform Symphony product gathers the utilization and allocation in the grid and the helps in decision to manage and extend the grid resources.

Source : Grids 2004: From Rocket Science to Business Service – The 451 Group

Now that we have a fair idea of what Grid computing is about, it is time to look at the buzzwords or technologies that often go hand in hand with grid. They include utility computing and on-demand computing and they often fit into the grid framework too.

Utility and On-Demand Computing

Many pundits in the IT industry agree that the computing trend is moving away from compute-inside-the-box to compute-outside-the-box, which is also called service-oriented computing. Computing is seen more of a service and accordingly it needs to be changed. This kind of trend is very much seen in the current business policies at leading firms like IBM, HP, Sun and EMC. The so-called On-Demand Computing is seen as a perfect alignment for the business needs and current stressed conditions.

Where's Tiger Woods right now?

Steve Evans, VP of information services for the PGA Tour, is tapping hundreds of volunteers to follow every golf pro around with Palm Pilots and survey-grade range finders. The resulting pgatour.com service offers real-time coverage of every player -- not just the front-runners. So how do you handle 100,000 subscribers during the golf season without buying boxes that sit idle most of the year? Outsource it to IBM's Virtual Linux Services, of course. If you can't tell the difference between a virtual server and a real server, then you might as well rent the server capacity on demand.

Source: Forester Research, 2003

Now there is another buzzword "Utility Computing" that is often used now. We have seen in the press, magazines and IT vendor-marketing brochures, the term Utility computing, and Grid is often used as if it is a synonym. Either there is haziness in understanding and/or there is belief that one technology leads to another.

Although there is considerable overlap in what grids do and what utility computing can enable an enterprise to do, the terms "grid computing" and "utility computing" are not synonyms.

It is understandable why many view them as synonyms. Both these technologies deal with using computing resources. Both can be deployed internally, or can make use of external resources. Both provide enterprises with opportunities to reduce computing costs. In addition, the utility computing can make use of grids to provide computing power to users and applications.

However, the differences lie in the architecture and the purpose or objective.

- A Grid is a network infrastructure that exploits computing resources and storage; it is based on the idea of sharing resources. Grid finds and exploits unused computing and storage resources that reside within a distributed computing environment.
- The utility computing is model or approach that enables computing resources and storage to be purchased or acquired on an "as-needed" basis (it's based on the idea of acquiring resources). When a computer is in need of additional resources, it acquires them from another source; this usually works on a pay-as-you-go basis.

Thus, utility computing is about availability of infrastructure; switching resources in a dynamic manner. In the utility computing, there are many ways the resource acquisition process works. Additional computing, memory, or storage resources are usually acquired using one or many of the following approaches:

- Activation of existing resources
- Reconfiguration of existing resources - to provide power/ storage to prioritized applications

- Being a part of the Grid and exploiting the extra resources
- By purchasing computing power and storage on an as needed basis from another source

Activating Existing Resources

In this approach, users activate the existing or embedded capacity that may already be available on the enterprise machines. A good example would be HP Superdome computer system. HP often ships such "built-in" capacity with its Superdome computer systems — and makes it possible for users to activate that capacity when needed to acquire additional computing power. Using this approach, HP charges its customers on the basis of CPU utilization.

In another example, IBM's case, additional CPU power is shipped with certain IBM servers and that CPU power can be activated or "turned-on" using a software key. IBM's approach packages what it calls "CUoD" (capacity upgrade on demand) on many of the servers.

Reconfiguration of Existing Resources

In this approach, the applications can be prioritized — and when an application requires additional computing power, storage, or memory it can acquire those resources by dynamically reconfiguring other systems to support its computing requirements. A good example is seen in "blade" servers. They can be re-configured with required system image on the fly.

Using Grid Resources

When used in the context of utility computing, an application would demand and receive additional computing power from a grid network.

Service Provisioning

The final way that utility services can be obtained is to purchase the services from an external source (called "utility service provisioning").

IBM has been spearheading the On-Demand vision for quite sometime. It follows a new approach to system management, leveraging its WebSphere's enterprise services bus and supported by techniques inherited from grid computing, involving virtualization and dynamic configurable systems. Grid computing techniques such as virtualization, dynamic provisioning, self-discovery and service isolation are the key elements in delivery of on-demand solutions.

There are many companies that are designing and marketing utility computing solutions. IT vendors such as Veritas, Sun, and HP etc., are quite active in this space. Many research analysts believe On-demand computing is on the rise.

A recent survey by Saugatuck Technology indicates that the use of on-demand or "pay-as-you-go" (PAYG) IT services has grown tremendously in the past year, to the point where more than 20 percent of firms recently surveyed report using one or more such services. An additional 45 percent of firms are considering using PAYG services, with the majority of those expecting to use pay as you go IT and business services within 24 months.

Grid Enablers

So far, we have covered many aspects of the grid and grid requirements. In the next section, we will examine some significant grid enabling technologies that are making a big difference. Particularly, the growth of low cost blade servers, Infiniband communication infrastructure and commodity operating such as Linux are very significant. The helping factors can be briefly summarized as below:

- Availability of inexpensive, commodity blade servers
- Inexpensive OS optimized for 1 to 4 CPUs such as Linux
- Storage no longer tied to a single server: NAS and SANs
- Fast interconnect technologies such as Gigabit Ethernet and Infiniband

Blade Servers

Buying the large SMP servers involves substantial capital costs. Many organizations over the period have ended up buying and not using to its fullest extent. With blade servers, purchases can be made incrementally blade-by-blade or frame by frame. Blades are also replaceable because they can be added and removed on an individual basis without having to shutdown the whole blade system or blade frame, thereby providing the online server management. Blades within a unified chassis are easier to manage.

Blade Server is a thin board containing one or more microprocessors. Blades offer power and memory similar to that available in typical 1U (1.75 inch high) servers, but squeeze vertically or horizontally into chassis, which include cabling, fans and power supplies typically found on individual servers.

InfiniBand

The InfiniBand architecture provides an industry-standard technology for "scaling out" computer platforms and is an ideal interconnect for developing high performance Itanium 2 processor-based compute grid clusters.

Low cost InfiniBand silicon that supports 10 Gb/sec RDMA transfers is shipping today providing eight times the bandwidth of Ethernet and three times the bandwidth of proprietary clustering interconnects. With an approved specification for 30 Gb/sec, InfiniBand is at least a generation ahead of competing fabric technologies today and in the foreseeable future.

Leveraging the 10 Gb/sec throughput and low latency capabilities of the InfiniBand technology, the InfiniBand enabled clusters are increasingly built in research and commercial institutions. They are showing a significant improvement in rendering speed (3x to 4x) over the Gigabit Ethernet enabled grid cluster.

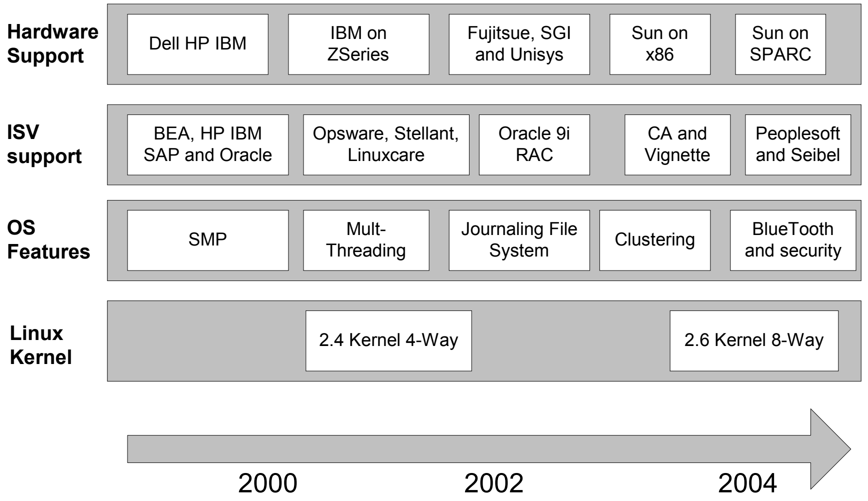
The Role of Linux

Linux operating system, which became quite popular for hosting web servers and firewalls is now increasingly picked up or selected for mission critical servers such as the application clusters and databases.

“Linux is able to provide the UNIX reliability at Intel prices,” says a recent report by Forester Research Inc. For example, Linux based 2-way Dell machines can handle the same wok load as the Solaris 4-way Sun servers and at a fraction of the cost.

Linux has achieved other technological advances such as 4-way CPU support and sophisticated threading model and enterprise class security. Linux based Intel servers and clusters are now being increasingly used for IBM Webspehere servers, BEA Web Logic Servers and for Oracle Real Application Clusters.

Figure 2.4 shows the gradual growth of Linux and its footing into the corporate IT data centers. As the Linux operating system is gaining ground and maturing as commodity operating system, it is becoming a key resource for Grid infrastructure.



Source : Forester Research Inc

Figure 2.4 *Linux Adoption and Growth*

Clusters

When the IT technologists began examining grid computing, many mistakenly viewed the clusters as the grid environments. Adding to this confusion, vendors such as Sun Microsystems began calling their cluster offerings Grid Clusters. However, the clusters have different architecture, objective and purpose. Next, we examine the true nature of the clustering technology.

What Is Cluster?

According to Greg Pfister, the guru of clustering,

> *"A cluster is a type of parallel system that consists of interconnected whole computers and is used as a single, unified computing resource".*

The whole computer is a normal combination of parts that comprise a stand-alone, usable computer. The components include one or more processors (including SMP and NUMA), memory, and I/O facilities. The whole computer is also referred to as a 'Node'.

Thus, the cluster is a configuration of a group of independent servers, so that they appear on a network as a single machine. This group can be managed as a single system, shares a common namespace, and is designed specifically to tolerate component failures and to support the addition or subtraction of components in a way that is transparent to users. Figure 2.5 represents a 3-Node Cluster hosting a parallel database with three database instances residing on three hosts and providing jointly the database service.

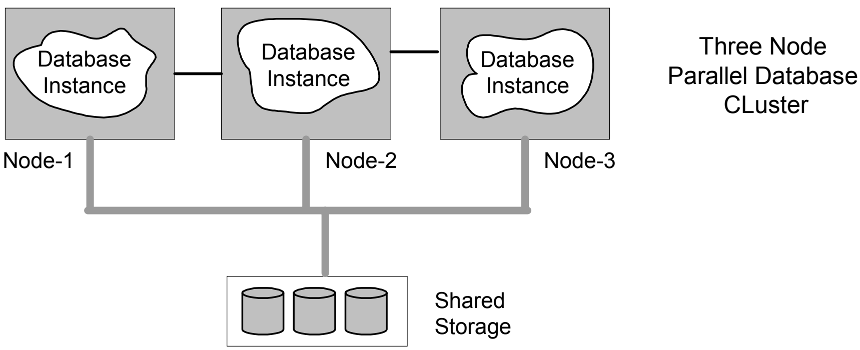


Figure 2.5 *A typical Cluster with 3-Nodes supporting Database Service*

Why Clusters?

Symmetric multiprocessors (SMP) and non-uniform memory access (NUMA) systems have grown in size and power. It is very common to see SMP or NUMA systems equipped with 8 to 64 processors. They present an alternative to cluster systems, but clusters have remained competitive and have made inroads into commercial application environments. There are obvious reasons for the survival of clusters: [Pfister]

- **Performance** - Throughput, response time, and turn around time is improved by using several machines at the same time. Clusters generate higher levels of performance.
- **Availability** - Clusters provide uninterrupted service by redistributing work, or through shifting the application services to surviving nodes by way of a failover process.
- **Price/Performance** - Clusters or other forms of computer aggregations are typically collections of machines that individually have very good performance for their price.
- **Incremental Growth** - It is easier to justify adding to a cluster than to buy a whole new computing facility.
- **Scaling** - Clusters have the ability to add capacity as needed.
- **Manageability** - Clusters have the ability to be managed as a single system.

Cluster Objectives

Traditionally, the term cluster was used to represent '**Server Clusters**'. Clustered systems are synonymous with a group of servers. The server, being the main layer or platform where the database or application service resides, is the most important component in providing availability and high performance.

Clusters, with multiple nodes, primarily aim at protecting server availability. Any failure in a server is transparent to end-users and is hidden by the failover of the application or database to a surviving node. End users or clients have access to the surviving node, thus allowing processing to continue. In another situation, a group of nodes are joined together to provide database services, as in a parallel database. Failure of a single node does not interrupt access to the database since the secondary nodes are still active in providing database access.

Thus, the cluster technology focuses on providing an alternative to a failed server. However, we have to realize that there are many other layers or components, each of which is significant in maintaining overall availability of the database or application service. Although the server (node) is a very crucial component and plays the key role in running the database or application, there are other components, such as the disk storage units and networking equipment for which we need to provide alternatives or backups to meet the failure conditions. The traditional concept of clustering revolves around server clustering only.

With this in mind, administrators and managers should provide adequate redundancy for other components in order to have an effective high availability environment.

Clusters vs. Grids

As evident from the close examination of Grid and Clusters, there are many differences. The following table shows comparison of Grid and Clusters.

Characteristic	Cluster	Grid
Population	Commodity Computers	Commodity and High-end computers
Ownership	Single	Multiple
Discovery	Membership Services	Centralized Index and Decentralized Info
User Management	Centralized	Decentralized
Resource management	Centralized	Distributed

Characteristic	Cluster	Grid
Allocation/ Scheduling	Centralized	Decentralized
Inter-Operability	VIA and Proprietary	No standards being developed
Single System Image	Yes	No
Scalability	100s	1000?
Capacity	Guaranteed	Varies, but high
Throughput	Medium	High
Speed(Lat. Bandwidth)	Low, high	High, Low

Grid represents a bigger framework and architecture, and focuses on the broader scope or objective. Grid incorporates many varied computing resources and the clusters often become one of the many components. A Grid enables the sharing, selection, and aggregation of a wide variety of geographically distributed resources including supercomputers, storage systems, data sources, and specialized devices owned by different organizations for solving large-scale resource intensive problems in science, engineering, and commerce. Clusters, on the other hand, focus on a specific objective and purpose, such as a database service or a web logic application server. These clusters fit into grid architecture very well for ultimate sharing of resources at a higher level of aggregation.

With technological advances we have witnessed in recent times, we may not be far from a time period where large number of nodes (say, 50 to 100 nodes) form a database cluster incorporating a variety of data stores consolidated into a huge data sources. That may lead to a true Data Grid of different stores supported in a single large cluster.

In the next chapter, *High Performance and Highly Available Database Clusters,* we will cover a detail account of Database Clusters and how they fit into the high performance computing arena. We will also examine the types of clusters, simple fail-over clusters and complex parallel database clusters.

Conclusion

In this chapter, we have attempted to look at differences between the aggregation of servers by way of clustering and by way of setting up of a Grid framework.

Clustering technology has been around for many years and it focuses on a specific objective. It is usually an aggregation of a small number of servers supporting an application or a database.

Grid Technology focuses on wider scale issues and is usually the aggregation of many varied computing resources, in which even the clusters become one of the many critical components. Grid enables the aggregation of a wide variety of geographically distributed resources including supercomputers, smaller computers, storage systems and many data sources.

We have also examined the difference between Grid and Utility Computing. Utility computing is more of a perception from the client or consumer of the grid resources.