# 1. Introduction to Grid Computing

In today's pervasive world of needing information anytime and anywhere, the explosive Grid Computing environments have now proven to be so significant that they are often referred to as being the world's single and most powerful computer solutions.. As a matter of fact the complexity and dynamic nature of industrial problems in today's world are much more intensive to satisfy by the more traditional, single computational platform approaches. Grid computing enables the virtualization of distributed computing and data resources such as processing, network bandwidth and storage capacity to create a single system image, granting users and applications seamless access to vast IT capabilities. Just as an Internet user views a unified instance of content via the Web, a grid user essentially sees a single, large virtual computer.

Grid computing is concerned with "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations." The key concept is the ability to negotiate resource-sharing- arrangements among a set of participating parties (providers and consumers) and then to use thee resulting resource pool for some purpose..

At its core, grid computing is based on an open set of standards and protocols - e.g., Open Grid Services Architecture (OGSA) - that enable communication across heterogeneous, geographically dispersed environments. With grid computing organizations can optimize computing and data resources, pool them for large capacity workloads, share them across networks and enable collaboration.

# GRID COMPUTING

Computational Grid is a collection of distributed, possibly heterogeneous resources which can be used as an ensemble to execute large-scale applications

"Grid" computing has emerged as an important new field, distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and, in some cases, high-performance orientation. In this article, we define this new field.

Grid computing is a very hot topic these days. Many major IT vendors are promoting and announcing "grid," "on-demand," "adaptive infrastructure" or some closely related initiative. It's likely the buzz will only increase as these firms reorient themselves to this emerging market.

Though it may seem to be yet another "next big thing," grid computing is in fact. bringing real benefits to commercial enterprises. That's why enterprises and the software vendors that serve the analytics/business intelligence (BI) sectors are now partnering with the 'technology specialists in this space - or pushing initiatives of their own. It's particularly relevant in today hyper-competitive yet cost constrained times when companies truly do need to do more with less.

A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities. "

grid computing is based on the concept of coordinated shared use of computers grid computing is a way to create a virtual supercomputer by connecting large numbers of  pcs in different locations over a shared network grid computing is

applying the resources of many computers in a network to a single problem at the same time.

Grid computing is concerned with "coordinated resource sharing and problem solving in dynamic; multi-institutional virtual organizations." The key concept is the ability to negotiate resource-sharing arrangements among a set of participating parties (providers and consumers) and then to use the resulting resource pool for some purpose. We noted: "The sharing that we are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering. This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. A set of individuals and/or institutions defined by such sharing rules form what we call a virtual organization."

A Grid Checklist I suggest that the essence of the definitions above can be captured in a simple checklist, according to which a Grid is a system that:
1. coordinates resources that are NOT subject to centralized control
2. uses standard, open, general purpose protocols and interfaces
3. delivers non-trivial qualities of service

## Expansion:

- Coordinates resources that are not subject to centralized control - (A Grid integrates and coordinates resources and users that live within different control domains for example, the user's desktop vs. central computing; different administrative units of the same company; or different companies; and addresses the issues of security, policy, payment, membership, and so forth that arise in these settings. Otherwise, we are

dealing with a local management system.).

- Using standard, open, general-purpose protocols and interfaces - (A Grid is built from multi-purpose protocols and interfaces that address such fundamental issues as authentication, authorization, resource discovery, and resource access. As I discuss further below, it is important that these protocols and interfaces be standard and open. Otherwise, we are dealing with an application-specific system.).

- To deliver nontrivial qualities of service - (A Grid allows its constituent resources to be used in a coordinated fashion to deliver various qualities of service, relating for example to response time, throughput, availability, and security, and/or co-allocation of multiple resource types to meet complex user demands, so that the 'utility of the combined system is significantly greater than that of the sum of its parts.)

## 2. Grid Computing equates to the world's largest computer...

"Grid computing is based on the concept of coordinated shared use of computers. Grid computing is a way to create a virtual supercomputer by connecting large numbers of PCs in different locations over a shared network".

The Grid Computing discipline involves the actual networking services and connections of a potentially unlimited number of ubiquitous computing devices within a "grid." This new innovative approach to computing can be most simply thought of as a massively large power "utility" grid, such as what provides power to our homes and businesses each and every day. This delivery of utility-based power has become second nature to - many of us, worldwide. We know that by simply walking into a room and turning on the lights, the power will be directed to the proper devices of our choice for that moment in time. In this same utility fashion, Grid Computing openly seeks and is capable of adding an infinite number of computing devices into any grid environment, adding to the computing capability and problem resolution tasks within the operational grid environment.

The worldwide business demand requiring intense problem-solving capabilities for incredibly complex problems has driven in all global industry segments the need for dynamic collaboration of many ubiquitous computing resources to be able to work together. These difficult computational problem-solving needs have now fostered many complexities in virtually all computing technologies, while driving up costs and operational aspects of the technology environments. However, this advanced computing collaboration capability is indeed required in almost all areas of industrial and business problem solving, ranging from scientific studies to commercial solutions to academic endeavors. It is a difficult challenge across all the technical communities to achieve this level of resource collaboration needed for solving these complex and dynamic problems.

## 3. Why is Grid computing important?

Grid computing is about getting computers to work together. Almost every organization is sitting on top of enormous, unused computing capacity, widely distributed. Mainframes are idle 40% of the time. UNIX servers are actually "serving" something less than 10% of the time. And most PCs do nothing for 95% of a typical day. Imagine an airline with 90% of its fleet on the ground, an automakerwith 40% of its assembly plants idle, a hotel chain with 95% of its rooms unoccupied. With Grid computing, businesses can optimize computing and data resources, pool them for large capacity workloads, share them across networks, and enable collaboration. Many consider Grid computing the next logical step in the evolution of the Internet, and maturing standards and a drop in the cost of bandwidth are fueling the momentum we're experiencing today. Virtualization of the computing environment -- Grid computing -- is a key component of the IBM e-business on demand strategy. Virtualizes your computing environment, automate it, and integrate your business processes and information, and you'll have an on demand operating environment that can transform your business.

## 4. What Grid computing_can do? .

When you deploy a grid, it will be to meet asset of customer requirements. To better match grid computing capabilities to those requirements, it is useful to keep in mind the reasons for using grid computing. This section describes the most important capabilities of grid computing.

Exploiting underutilized resources

The easiest use of grid computing is to run an existing application on a different machine. The machine on which the application is normally run might be unusually busy due to an unusual peak in activity. The job in question could be run on an idle machine elsewhere on the grid. There are at least two prerequisites for this scenario.

1. The application must be executable remotely and without undue overhead.
2. The remote machine must meet any special hardware/software, or resource requirements imposed by the application.

For example, a batch job that spends a significant amount of time processing a set of input data to produce an output set is perhaps the most ideal and simple use for a grid. If the quantities of input and output are large, more thought and planning might be required to efficiently use the grid for such a job.

In most organizations, there are large amounts of underutilized computing resources. Most desktop machines are busy less than 5% of the time. In some organizations, even the server machines can often be relatively idle. Grid computing provides a framework for exploiting these underutilized resources and thus has the possibility of substantially increasing the efficiency of resource usage.

The processing resources are not the only ones that may be underutilized. Often,

machines may have enormous unused disk drive capacity. Grid computing, more specifically, a "data grid", can be used to aggregate this unused storage into a much larger virtual data store, possibly configured to achieve improved performance and reliability over that of any single machine.

If a batch job needs to read a large amount of data, this data could be automatically replicated at various strategic points in the grid. Thus, if the job must be executed on a remote machine little grid, the data is already there and does not need to be moved to that remote point. This offers clear performance benefits. Also, such copies of data can be used as backups when the primary copies are damaged or unavailable.

Another function of the grid is to better balance resource utilization. An organization may have occasional unexpected peaks of activity that demand more resources. If the applications are grid enabled, they can be moved to underutilized machines during such peaks. In fact, some grid implementations can migrate partially completed jobs. In general, a grid can provide a consistent way to balance the loads on a wider federation of resources.

This applies to CPU, storage, and many other kinds of resources that may be available on a grid. Management can use a grid to better view the usage patterns in the larger organization, permitting better planning when upgrading systems, increasing capacity, or retiring computing resources no longer needed.

## 5. Parallel CPU capacity

The potential for massive parallel CPU capacity is one of the most attractive features of a grid. In addition to pure scientific needs, such computing power is driving a new evolution in industries such as the bio-medical field, financial modeling, oil exploration, motion picture animation, and many others. The common attribute among such uses is that the applications have been written to use algorithms that can be partitioned into independently running parts.

A CPU intensive grid application can be thought of as many smaller "sub jobs," each executing on a different machine in the grid. To the extent that these sub jobs do not need to communicate with each other, the more "scalable" the application becomes. A perfectly scalable application will, for example, finish 10 times faster if it uses 10 times the number of processors. Barriers often exist to perfect scalability. The first barrier depends on the algorithms used for splitting the application among many CPUs. If the algorithm can only be split into a limited number of independently running parts, then that forms a scalability barrier. The second barrier appears if the parts are not completely independent; this can cause contentions, which can limit scalability.

# 6. Grid Computing Concepts

Grid computing environments must be constructed upon the following foundations:

Coordinated resources. We should avoid building grid systems with a centralized control; instead, we must provide the necessary infrastructure for coordination among the resources, based on respective policies and service-level agreements.

Open standard protocols and frameworks. The use of open standards provides interoperability and integration facilities. These standards must be applied for resource discovery, resource access, and resource coordination.

Another, basic requirement of a Grid Computing system is the ability to provide the quality of service (QoS) requirements necessary for the end-user community. These Qos validations must be a basic feature in any Grid system, and must be done in congruence with the available resource matrices. These QoS features can be (for example) response time measures, aggregated performance, security fulfillment, resource scalability, availability, autonomic features such as event correlation and configuration management, and partial fail over mechanisms.

There have been a number of activities addressing the above definitions of Grid Computing and the requirements for a grid system. The most notable effort is in the standardization of the interfaces and protocols for the Grid Computing infrastructure implementations.

# 7. Types of Grids

Grid computing .can be used in a variety of ways to address various kinds of application requirements. Often, grids are categorized by the type of solutions that they best address. The three primary types of grids are summarized below. Of course, there are no hard boundaries between these grid types and often grids may be a combination of two or more of these_ However, as you consider developing applications that may run in a grid environment, remember that the type of grid environment that you will be using will affect many of your decisions.

## 1. Computational grid

"A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities".

## 2. Scavenging grid

A scavenging grid is .most commonly used with large numbers of desktop machines. Machines are scavenged for available CPU cycles and other resources. Owners of the desktop machines are usually given control over when their resources are available to participate in the grid.

## 3. Data grid

A data grid is responsible for housing and providing access to data across multiple organizations. Users are not concerned with where this data is located as long as they have access to the data. For example, you may have two universities doing life science research, each with unique data. A data grid would allow them to share their data, manage the data, and manage security issues such as who has access to what data.

# 8. The key components

Security, because you have to take care that only authorized users can access and use Grid resources. The heterogeneous nature of resources and their differing security policies are complicated and complex in the security schemes of a Grid Computing environment. These computing resources are hosted in differing security domains and heterogeneous platforms. Simply speaking, our middleware solutions must address local security integration, secure identity mapping, secure access/authentication, secure federation, and trust management.

Data management, because data must be transported, cleansed, parceled, and processed. Data forms the single most important asset in a Grid Computing system. This data may be input into the resource, and the results from the resource on the execution of a specific task. If the infrastructure is not designed properly, the data movement in a geographically distributed system can quickly cause scalability problems. It is well understood that the data must be near to the computation where it is used. This data movement in any Grid Computing environment requires absolutely secure data transfers, both to and from the respective resources.

Resource management, because the grid must know what resources are available for different tasks. The tremendously large number and the heterogeneous potential of Grid sources cause the resource management challenge to be a significant effort topic in Grid Computing environments. These resource management scenarios often include, resource discovery, resource inventories, fault isolation, resource provisioning, resource monitoring, a variety of autonomic capabilities, and service-level management activities. The most interesting aspect of the resource management area is the selection of the correct resource from the grid resource pool, based on the service level requirements, and then to efficiently provision them to facilitate user needs. Information services, because users and applications must be able to query the grid efficiently. Information

services are fundamentally concentrated on providing valuable information respective to the Grid Computing infrastructure resources. These services leverage and entirely depend on the providers of information such as resource availability, capacity, and utilization, just to name a few. This information is valuable and mandatory feedback respective to the resources managers discussed earlier in this chapter. These information services enable service providers to most efficiently allocate resources for the variety of very specific tasks related to the Grid Computing infrastructure solution.

## 9. The Nature of Grid Architecture

The establishment, management, and exploitation of dynamic, cross-organizational VO sharing relationships require new technology. In defining Grid architecture, we start from the perspective that effective VO operation requires that we be able to establish sharing relationships among any potential participants. Interoperability is thus the central issue to be addressed. In a networked environment, interoperability means common protocols. Hence, our Grid architecture is first and foremost a protocol architecture, with protocols defining the basic mechanisms by which VO users and resources negotiate, establish, manage, and exploit sharing relationships. A standards-based open architecture facilitates extensibility, interoperability, portability, and code sharing; standard protocols make it easy to define standard services that provide enhanced capabilities.

Why is interoperability such a fundamental concern? At issue is our need to ensure that sharing relationships can be initiated among arbitrary parties, accommodating new participants dynamically, across different platforms, languages, and programming environments.
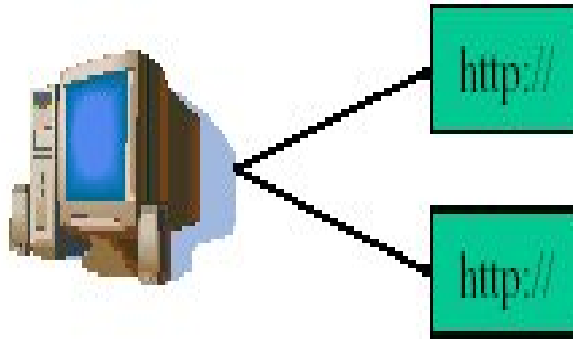
In this context, mechanisms serve little purpose if they are not defined and implemented so as to be interoperable across organizational boundaries, operational policies, and resource types. Without interoperability, vo applications and participants are forced to enter into bilateral sharing arrangements, as there is no assurance that the mechanisms used between any two parties will extend to any other parties. Without such assurance, dynamic vo formation is all but impossible, and the types of vos that can be formed are severely limited. Just as the Web revolutionized information sharing by providing a universal protocol and syntax (HTTP and HTML) for information exchange, so we require standard protocols and syntaxes for general resource sharing.

## 10. Grid Architecture Description

Our architecture and the subsequent discussion organize components into layers. Components within each layer share common characteristics but can build on capabilities and behaviors provided by any lower layer.
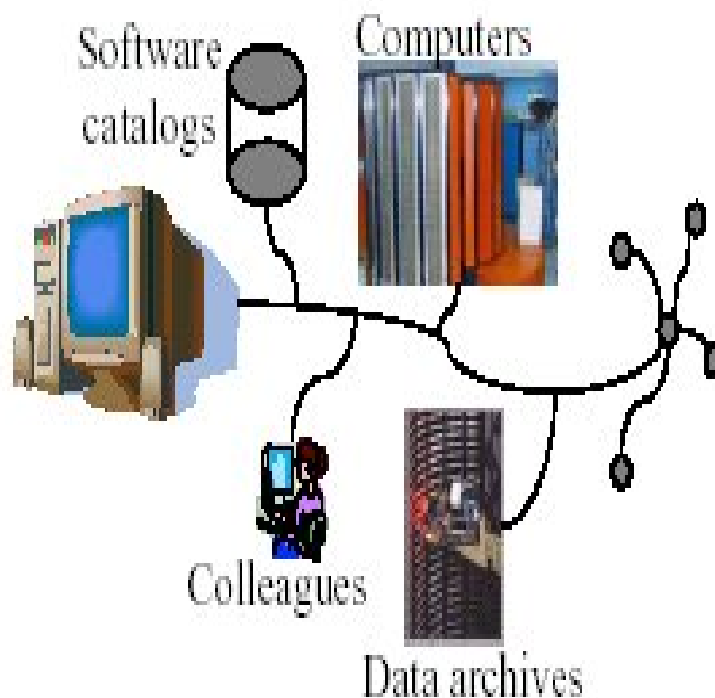
In specifying the various layers of the Grid architecture, we follow the principles of the "hourglass model" .The narrow neck of the hourglass defines a small set of core abstractions and protocols (e.g., TCP and HTTP in the Internet), onto which many different high-level behaviors can be mapped (the top of the hour glass).

<u>A network with a very high operative capacity.</u>

What is it?

"Grid computing" is the new network concept, as a natural evolution of the basic computing concept. No longer a single resource, but several heterogeneous resources which share computing in order to arrive at a common result.
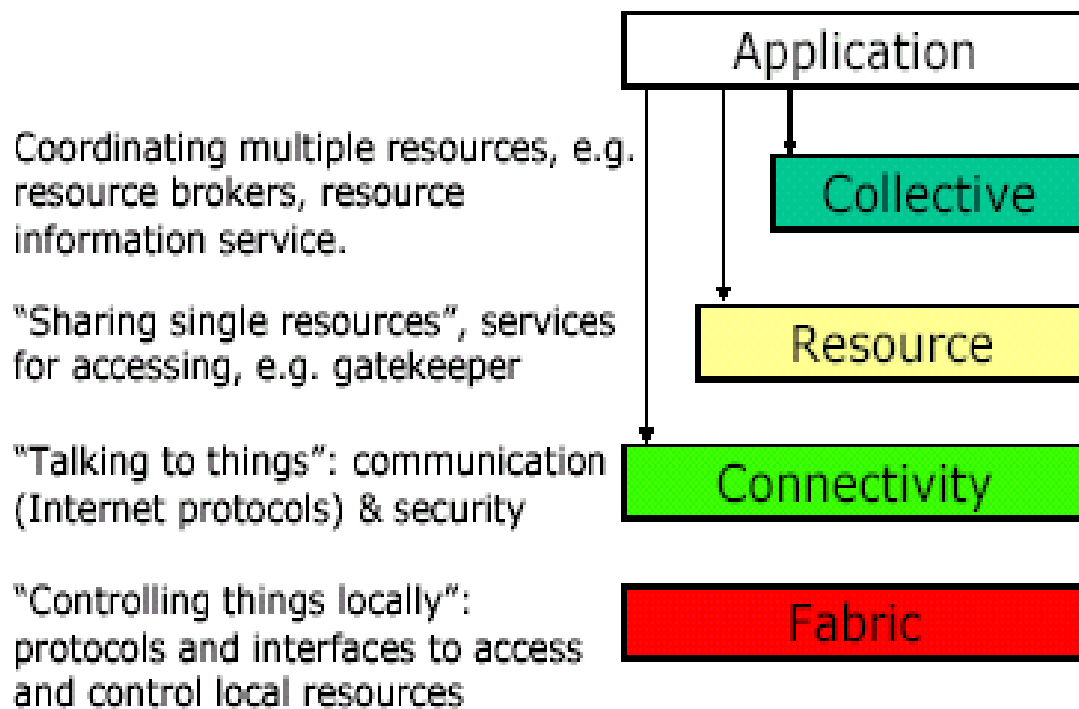
The term grid computing identifies an infrastructure of distributed calculation: thousands of heterogeneous units ranging from programmes, files and data (logical resource;) to computers, sensors and networks, physical resources connected to each other telemetrically with highly flexible sharing relationships (client-server, peer-to-peer and Mesh architectures).

Created to respond to the needs of calculating and processing enormous quantities of data by universities and other bodies for scientific purposes and planning. Its function is therefore to pool calculating and heterogeneous memory resources, so as to reach power levels that otherwise would be impossible.

In this new context, sharing becomes direct access to computers, data, software and other resources ad no longer a simple exchange of files. Grid Computing is derived from Parallel Computing, applied to real cases and for more varied objectives.

Interoperability becomes the fundamental aspect and in a network environment this is translated into common protocols. So the grid architecture is first of all a protocol architecture made of the exchange ,of data, electronics and hardware, firmware and management software. The pooling of resources and the use of common communication protocols are in fact one of the fundamental characteristics of this

innovative form of

| | |
|---|---|
| | **Application** |
| Coordinating multiple resources, e.g. resource brokers, resource information service. | **Collective** |
| "Sharing single resources", services for accessing, e.g. gatekeeper | **Resource** |
| "Talking to things": communication (Internet protocols) & security | **Connectivity** |
| "Controlling things locally": protocols and interfaces to access and control local resources | **Fabric** |

# 11. Relationships with Other Technoloaies

The concept of controlled, dynamic sharing within vos is so fundamental that we might assume that Grid-like technologies must surely already be widely deployed. In practice, however, while the need for these technologies is indeed widespread, in a wide voriety of different areas we find only primitive -and inadequate solutions to vo problems. In brief, current distributed computing approaches do not provide a general resource-sharing framework that addresses vo requirements.

Grid technologies distinguish themselves by providing this generic approach to resource sharing. This situation points to numerous opportunities for the application of Grid technologies.

### 1. World Wide Web

The ubiquity of Web technologies (i.e., IETF and W3C standard protocols TCP/IP, HTTP, SOAP, etc.-and languages, such as HTML and XML) makes them attractive as a platform for constructing vo systems and applications. However, while these technologies do an excellent job of supporting the browser client-to-web-server interactions that are the foundation of today's Web, they lack features required for the richer interaction models that occur in vos. For example, today's Web browsers typically use TLS for authentication, but do not support single Sign-on or delegation. Clear steps can be taken to integrate Grid and Web technologies. For example, the single sign-on capabilities provided in the GSI extensions to TLS would, if integrated into Web browsers, allow for single sign-on to multiple Web servers. GSI delegation capabilities would permit a browser client to delegate capabilities to a Web server so that the server could act on the client's behalf.

These capabilities, in turn, make it much easier to use Web technologies to build "vo Portals" that provide thin client interfaces to sophisticated vo applications. WebOS

addresses some of these issues.

## 2. Application and Storage Service Providers

Application service providers, storage service providers, and similar hosting companies typically offer to outsource specific business and engineering applications (in the case of ASPs) and storage capabilities (in the case of SSPs). A customer negotiates a service level agreement that defines access to a specific combination of hardware and software. Security tends to be handled by using VPN technology to extend the customer's intranet to encompass resources operated by the ASP or SSP on information, these Intergrid protocols enable different organizations to interoperate and exchange or share resources.

The customer's behalf. Other SSPs offer file sharing services, in which case access is provided via HTTP, FTP, or Web DAV with user ids, passwords, and access control lists controlling access.

From a vo perspective, these are low-level building-block technologies. VPNs and static configurations make many vo sharing modalities hard to achieve. For example, the use of VPNs means that it is typically impossible for an ASP application to access data located on storage managed by a separate SSP. Similarly, dynamic reconfiguration of resources within a single ASP or SPP is challenging and, in fact, is rarely attempted. The load sharing across providers that occurs on a routine basis in the electric power industry is unheard of in the hosting industry. A basic problem is that a VPN is not a VO: it cannot extend dynamically to encompass other resources and does not provide the remote resource provider with any control of when and whether to share its resources.

## 3. Enterprise Computing Systems

Enterprise development technologies such as CORBA, Enterprise Java Beans,

Java 2 Enterprise Edition, and DCOM are all systems designed to enable the construction of distributed applications. They provide standard resource interfaces, remote invocation mechanisms, and trading services for discovery and hence make it easy to share resources within a single organization. However, these mechanisms address none of the specific VO requirements listed above. Sharing arrangements are typically relatively static and restricted to occur within a single organization. The primary form of interaction is client-server, rather than the coordinated use of multiple resources. These observations suggests that there should be a role for Grid technologies within enterprise computing. .

## 4. Internet and Peer-to-Peer Computing

Peer-to-peer computing and Internet computing (as implemented, for example by the SETI@home system) is an example of the more general ("beyond client-server") sharing modalities and computational structures that we referred to in our characterization of VOs. As such, they have much in common with Grid technologies.

In practice, we find that the technical focus of work in these domains has not overlapped significantly to date. One reason is that peer-to-peer and Internet computing developers have so far focused entirely on vertically integrated ("stovepipe") solutions, rather than seeking to define common protocols that would allow for shared infrastructure and interoperability. Another is that the

, forms of sharing targeted by various applications are quite limited, for example, file sharing with no access control, and computational sharing with a centralized server.

As these applications become more sophisticated and the need for interoperability becomes clearer we will see a strong convergence of interests between peer-to-peer, Internet, and Grid computing. For example, single sign-on, delegation, and authorization technologies become important when computational and data sharing services must interoperate, and the policies that govern access to individual resources become more complex.

## 12. Other Perspectives on Grids

Some perspective on Grids and VOs presented hereby:

1. The Grid is a next-generation Internet. "The Grid" is not an alternative to "the Internet": it is rather 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. Any resource that is "on the Grid" is also, by definition, "on the Net."

.2. The Grid is a source of free cycles. Grid computing does not imply unrestricted access to resources. Grid computing is about controlled sharing. Resource owners will typically want to enforce policies that constrain access according to group membership, ability to pay, and so forth. Hence, accounting is important, and a Grid architecture must incorporate resource and collective protocols for exchanging usage and cost information, as well as for exploiting this information when deciding whether to enable sharing.

3. The Grid requires a distributed operating system. Grid software should define the operating system services to be installed on every participating system, with these services providing for the Grid what an operating system provides for a single computer: namely, transparency with respect to location, naming, security, and so forth. Put another way, this perspective views the role of Grid software as defining a virtual machine.

4. The Grid requires new programming models. Programming in Grid environments introduces challenges that are not encountered in sequential (or parallel) computers, such as multiple administrative domains, new failure modes, and large variations in performance.

5. The Grid makes high-performance computers superfluous. The hundreds, thousands, or even millions of processors that may be accessible within a VO represent a significant source of computational power, if they can be harnessed in a useful fashion. This does not imply, however, that traditional high-performance computers are obsolete. Many problems require tightly coupled computers, with low latencies and high. communication bandwidths; Grid computing may well increase, rather than reduce, demand for such systems by making access easier.

## 13. Applications

Solutions in grid architecture may also be implemented for company situations. Having at one's disposal great calculation powers and storage areas of considerable size, in a manner completely unrelated to the physical distribution of the structures, allows the speeding up of computing processes, optimizes resources, simplifies the management of company information systems and reduces management costs.

The use of suitable instruments at "middleware" level allows the dynamic aggregation of several machines or clusters of machines distributed over the territory in a single virtual information system, where the user may have a great quantity of computing power at his disposal, equal to the partial sum of the powers of the aggregated machine} as though it were one large machine.

It is possible to extend these concepts to networks of sensors and industrial actuators. REI COM technologies contemplate the intelligent and shared use of knowledge also at the level of the single atomic element such as, for example, a radio module or a field sensor.

# 14. Advantages

The extendibility, interoperability, portability and pooling of the code allow rapid access to an other wise impossible number of detecting, processing and storage instruments with consequent optimization of the processes; the greater the amount o( data processed, the more calculation instruments used, the faster and more precise and reliable the results.

Pooling information in a Grid Computing architecture on a Mesh infrastructure offers a 100% fault "tolerant system and the possibility of decentralizing the operative responsibilities for calculation, entrusting the good operation of the system to a more distributed logic, maintaining control only in the points (nodes) where is considered really opportune and important.