

Grid Computing- An Emerging Technology that enables large-scale resource sharing

Prof. R. S. Kamble¹, Prof. Mrs. D. A. Nikam²

¹(Department of Computer Science & Engg., Dr. J. J. Magdum College of Engineering, Jaysingpur, India)

²(Department of Computer Science & Engg., Dr. J. J. Magdum College of Engineering, Jaysingpur, India)

Abstract: In the last few years there has been a rapid exponential increase in computer processing power, data storage and communication. But still there are many complex and computation intensive problems, which cannot be solved by supercomputers. These problems can only be met with a vast variety of heterogeneous resources. The increased use and popularity of the Internet and the availability of high-speed networks have gradually changed the way we do computing. These technologies have enabled the cooperative use of a wide variety of geographically distributed resources as a single more powerful computer. This new method of pooling resources for solving large-scale problems is called as grid computing. Grid computing is a form of distributed computing in which an organization (business, university, etc.) uses its existing computers (desktop and/or cluster nodes) to handle its own long-running computational tasks. Grid computing combines computers from multiple administrative domains to reach a common goal, to solve a single task, and may then disappear just as quickly. Grid computing is managed by Global Grid Forum (GGF). Grid includes various protocols, topologies, standards and layers for implementing grid related applications. This paper describes the concepts underlying grid computing.

Keywords: CPU scavenging, Open Grid Forum (OGF), Object Management Group (OMG), Sensor grid, Supercomputers.

I. Introduction

1.1 History

The term grid computing originated in the early 1990s as a metaphor for making computer power as easy to access as an electric power grid. The power grid metaphor for accessible computing quickly became canonical when Ian Foster and Carl Kesselman published their seminal work, "The Grid: Blueprint for a new computing infrastructure" (1999). CPU scavenging and volunteer computing were popularized beginning in 1997 by distributed.net and later in 1999 by SETI@home^[1] to harness the power of networked PCs worldwide, in order to solve CPU-intensive research problems.

The ideas of the grid (including those from distributed computing, object-oriented programming, and Web services) were brought together by Ian Foster, Carl Kesselman, and Steve Tuecke, widely regarded as the "fathers of the grid". They led the effort to create the Globus Toolkit incorporating not just computation management but also storage management, security provisioning, data movement, monitoring, and a toolkit for developing additional services based on the same infrastructure, including agreement negotiation, notification mechanisms, trigger services, and information aggregation. While the Globus Toolkit remains the de facto standard for building grid solutions, a number of other tools have been built that answer some subset of services needed to create an enterprise or global grid.

In 2007 the term cloud computing came into popularity, which is conceptually similar to the canonical Foster definition of grid computing (in terms of computing resources being consumed as electricity is from the power grid). Indeed, grid computing is often (but not always) associated with the delivery of cloud computing systems as exemplified by the AppLogic system from 3tera.

1.2 Definitions of a Grid

- "... a network of conductors for distribution of electric power; also : a network of radio or television stations" – Merriam-Webster
- "... the illusion of a simple yet large and powerful self-managing virtual computer out of a large collection of connected heterogeneous systems sharing various combinations of resources" – IBM Redbooks
- "Grid Computing enables virtual organizations to share geographically distributed resources as they pursue common goals, assuming the absence of central location, central control, omniscience, and an existing trust relationship." – Globus Alliance
- "The Web provides us information — the grid allows us to process it." - Ahmar Abbas.

1.3 What is Grid computing?

Today there are many definitions of grid computing:

- In his article “What is the Grid? A Three Point Checklist”, Ian Foster lists these primary attributes:
 - Computing resources are not administered centrally.
 - Open standards are used.
 - Nontrivial quality of service is achieved.
- Plaszczak /Wellner define grid technology as "the technology that enables resource virtualization, on-demand provisioning, and service (resource) sharing between organizations."
- IBM defines grid computing as “the ability, using a set of open standards and protocols, to gain access to applications and data, processing power, storage capacity and a vast array of other computing resources over the Internet. A grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of resources distributed across ‘multiple’ administrative domains based on their (resources) availability, capacity, performance, cost and users' quality-of-service requirements”.
- Buyya/Venugopal ^[2] define grid as "a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed autonomous resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements".
- CERN, one of the largest users of grid technology, talk of The Grid: “a service for sharing computer power and data storage capacity over the Internet.”

Grid computing is the collection of computer resources from multiple locations to reach a common goal. The grid can be thought of as a distributed system with non-interactive workloads that involve a large number of files. What distinguishes grid computing from conventional high performance computing systems such as cluster computing is that grids tend to be more loosely coupled, heterogeneous, and geographically dispersed. ^[3] Although a single grid can be dedicated to a particular application, commonly a grid is used for a variety of purposes. Grids are often constructed with general-purpose grid middleware software libraries.

Grid size varies a considerable amount. Grids are a form of distributed computing whereby a “super virtual computer” is composed of many networked loosely coupled computers acting together to perform large tasks. For certain applications, “distributed” or “grid” computing, can be seen as a special type of parallel computing that relies on complete computers (with onboard CPUs, storage, power supplies, network interfaces, etc.) connected to a network (private, public or the Internet) by a conventional network interface, such as Ethernet. This is in contrast to the traditional notion of a supercomputer, which has many processors connected by a local high-speed computer bus.

This technology has been applied to computationally intensive scientific, mathematical, and academic problems through volunteer computing, and it is used in commercial enterprises for such diverse applications as drug discovery, economic forecasting, seismic analysis, and back office data processing in support for e-commerce and Web services. Coordinating applications on Grids can be a complex task, especially when coordinating the flow of information across distributed computing resources. Grid workflow systems have been developed as a specialized form of a workflow management system designed specifically to compose and execute a series of computational or data manipulation steps, or a workflow, in the Grid context.

1.4 Grid computing characteristics

Grid characteristics may be described as follows:

- **Large scale:** A grid must be able to deal with a number of resources ranging from just a few to millions. This raises the very serious problem of avoiding potential performance degradation as the grid size increases.
- **Geographical distribution:** Grid’s resources may be located at distant places.
- **Heterogeneity:** A grid hosts both software and hardware resources that can be very varied ranging from data, files, software components or programs to sensors, scientific instruments, display devices, personal digital organizers, computers, super-computers and networks.
- **Resource sharing:** Resources in a grid belong to many different organizations that allow other organizations (i.e. users) to access them. Nonlocal resources can thus be used by applications, promoting efficiency and reducing costs.
- **Multiple administrations:** Each organization may establish different security and administrative policies under which their owned resources can be accessed and used. As a result, the already challenging network security problem is complicated even more with the need of taking into account all different policies.
- **Resource coordination:** Resources in a grid must be coordinated in order to provide aggregated computing capabilities.
- **Transparent access:** A grid should be seen as a single virtual computer.
- **Dependable access:** A grid must assure the delivery of services under established Quality of Service (QoS) requirements. The need for dependable service is fundamental since users require assurances that they will receive Predictable, sustained and often high levels of performance.

- **Consistent access:** A grid must be built with standard services, protocols and inter-faces thus hiding the heterogeneity of the resources while allowing its scalability. Without such standards, application development and pervasive use would not be possible.
- **Pervasive access:** The grid must grant access to available resources by adapting to a dynamic environment in which resource failure is commonplace. This does not imply that resources are everywhere or universally available but that the grid must tailor its behavior as to extract the maximum performance from the available re-sources.

An application characteristic of Grid Computing is shown in figure 1.

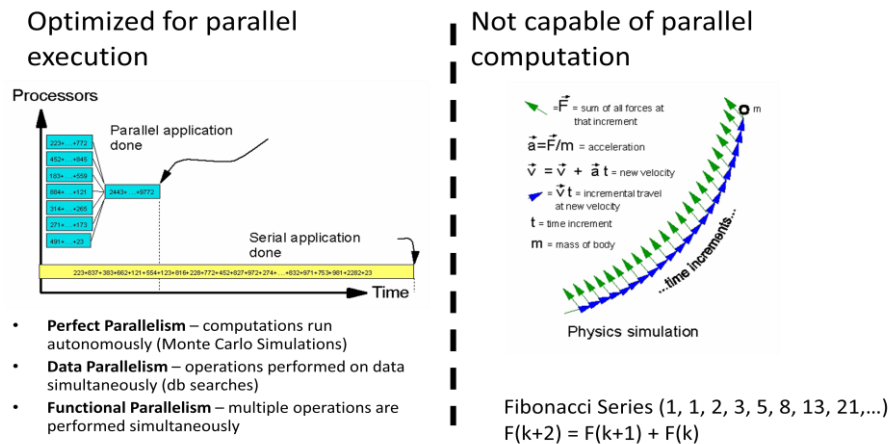


Fig.1. Application Characteristic

II. Importance Of Grid Computing

Grid computing is emerging as a viable technology that businesses can use to wring more profits and productivity out of IT resources -- and it's going to be up to you developers and administrators to understand Grid computing and put it to work.

It's really more about bringing a problem to the computer (or Grid) and getting a solution to that problem. Grid computing is flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources. Grid computing enables the virtualization of distributed computing 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 World Wide Web, a Grid user essentially sees a single, large, virtual computer.

Grid computing will give worldwide access to a network of distributed resources - CPU cycles, storage capacity, devices for input and output, services, whole applications, and more abstract elements like licenses and certificates.

For example, to solve a compute-intensive problem, the problem is split into multiple tasks that are distributed over local and remote systems, and the individual results are consolidated at the end. Viewed from another perspective, these systems are connected to one big computing Grid. The individual nodes can have different architectures, operating systems, and software versions. Some of the target systems can be clusters of nodes themselves or high performance servers.

III. Comparison Of Grids And Conventional Supercomputers

“Distributed” or “grid” computing in general is a special type of parallel computing that relies on complete computers (with onboard CPUs, storage, power supplies, network interfaces, etc.) connected to a network (private, public or the Internet) by a conventional network interface producing commodity hardware, compared to the lower efficiency of designing and constructing a small number of custom supercomputers. The primary performance disadvantage is that the various processors and local storage areas do not have high-speed connections. This arrangement is thus well-suited to applications in which multiple parallel computations can take place independently, without the need to communicate intermediate results between processors. The high-end scalability of geographically dispersed grids is generally favorable, due to the low need for connectivity between nodes relative to the capacity of the public Internet.

There are also some differences in programming and deployment. It can be costly and difficult to write programs that can run in the environment of a supercomputer, which may have a custom operating system, or require the program to address concurrency issues. If a problem can be adequately parallelized, a “thin” layer of “grid” infrastructure can allow conventional, standalone programs, given a different part of the same problem, to

run on multiple machines. This makes it possible to write and debug on a single conventional machine, and eliminates complications due to multiple instances of the same program running in the same shared memory and storage space at the same time.

IV. Market Segmentation Of The Grid Computing Market

For the segmentation of the grid computing market, two perspectives need to be considered: the provider side and the user side:

5.1 The provider side

The overall grid market comprises several specific markets. These are the grid middleware market, the market for grid-enabled applications, the utility computing market, and the software-as-a-service (SaaS) market.

Grid middleware is a specific software product, which enables the sharing of heterogeneous resources, and Virtual Organizations. It is installed and integrated into the existing infrastructure of the involved company or companies, and provides a special layer placed among the heterogeneous infrastructure and the specific user applications. Major grid middlewares are Globus Toolkit, gLite, and UNICORE.

Grid-enabled applications are specific software applications that can utilize grid infrastructure. This is made possible by the use of grid middleware, as pointed out above.

Software as a service (SaaS) is “software that is owned, delivered and managed remotely by one or more providers.” (Gartner 2007) Additionally, SaaS applications are based on a single set of common code and data definitions. They are consumed in a one-to-many model, and SaaS uses a Pay As You Go (PAYG) model or a subscription model that is based on usage. Providers of SaaS do not necessarily own the computing resources themselves, which are required to run their SaaS. Therefore, SaaS providers may draw upon the utility computing market. The utility computing market provides computing resources for SaaS providers.

5.2 The user side

For companies on the demand or user side of the grid computing market, the different segments have significant implications for their IT deployment strategy. The IT deployment strategy as well as the type of IT investments made are relevant aspects for potential grid users and play an important role for grid adoption.

V. Types Of Grid

The three primary types of grids and are summarized below:

5.1 Computational Grid

A computational grid is focused on setting aside resources specifically for computing power. In this type of grid, most of the machines are high-performance servers.

5.2 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.

5.3 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.

VI. Grid Related Concepts

6.1 Grid Topologies

Grid topologies are described as follows:

6.1.1 Departmental Grids

- Localized to a specific group of people.
- Generally, same hardware and software.
- Designed for high throughput and high performance over a dedicated network.

6.1.2 Enterprise Grids

- Service to numerous groups within a single company or campus.
- Resource heterogeneity increases.
- Company-wide local area network.

6.1.3 Extraprise Grids

- Service to multiple companies, partners, and customers within a particular domain.
- Domain based private network.

6.1.4 Global Grids

- Established over the public-Internet.

6.2 Grid Protocols

Grid uses various protocols these are described below:

6.2.1 Security Infrastructure (GSI)

- Extended from SSL/TLS and X.509 protocols
- Utilizes PKI for Certificate Authority
 - Primary objective is “Authorization”
 - Generates primary credential
 - Generates temporary proxy credential
- Certificate Authority
 - Positively identify entities requesting certificates
 - Issuing, removing, and archiving certificates
 - Protecting the Certificate Authority server
 - Maintaining a namespace of unique names for certificate owners
 - Serve signed certificates to those needing to authenticate entities
 - Logging activity

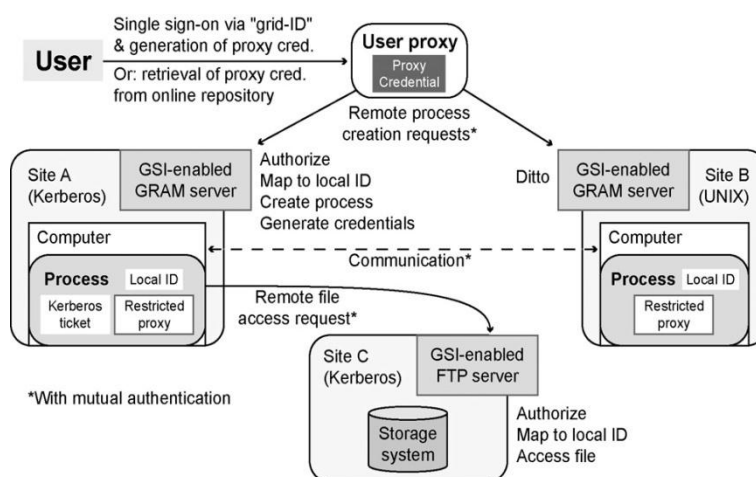


Fig. 2. Grid Security Infrastructure

6.2.2 Grid Resource Allocation and Management (GRAM)

- Allows programs to be started on remote resources
- Resource Specification Language (RSL)
 - Resource requirements
 - Machine type, number of nodes, memory, etc...
 - Job configuration
 - Directory, executable, arguments, environment
- Communication protocols
 - HTTP-base RPC (early protocol)
 - Web-services (WSDL, SOAP)

6.2.3 Grid File Transfer Protocol (GridFTP)

- Providing high-speed and reliable transfer of large volume data (petabytes)
- Extension of standard FTP to include
 - Striped/parallel data channels
 - Partial files
 - Automatic and manual TCP buffer size settings
 - Progress monitoring
 - Extended restart functionality

6.2.4 Grid Information Services (GIS)

- Grid Resource Information Service (GRIS)
 - Provides resource specific information
- Grid Resource Registration (GRR)
 - Updates GRIS about resource status
- Grid Index Information Service (GIIS)
 - An aggregate directory service
 - Provides a collection of information that has been gathered from multiple GRIS servers
- Grid Resource Inquiry (GRI)
 - Queries GRIS server for resource information
 - Queries GIIS server for information

6.3 Grid Standards & Layers

Perhaps the most important standard that has emerged recently is the Open Grid Services Architecture (OGSA), which was developed by the GGF. OGSA is an Informational specification that aims to define a common, standard and open architecture for Grid based applications. The goal of OGSA is to standardize almost all the services that a grid application may use, for example job and resource management services, communications and security. OGSA specifies a Service-Oriented Architecture (SOA) for the Grid that realizes a model of a computing system as a set of distributed computing patterns realized using Web services as the underlying technology. Basically, the OGSA standard defines service interfaces and identifies the protocols for invoking these services.

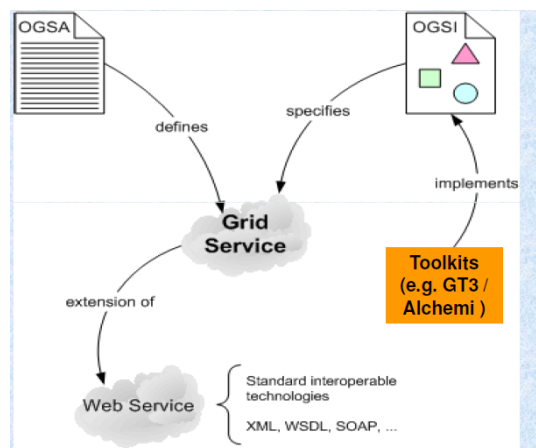


Fig.3. Grid standards

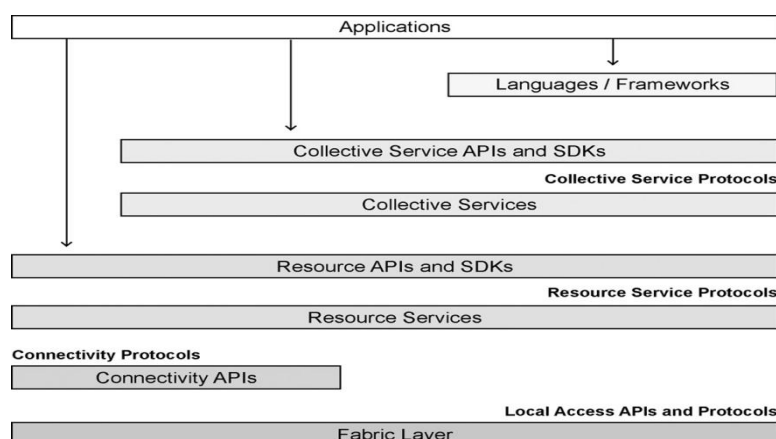


Fig.4. Grid Components & Layers

6.4 CPU scavenging

CPU-scavenging, cycle-scavenging, or shared computing creates a “grid” from the unused resources in a network of participants (whether worldwide or internal to an organization). Typically this technique uses desktop computer instruction cycles that would otherwise be wasted at night, during lunch, or even in the scattered seconds throughout the day when the computer is waiting for user input or slow devices. In practice,

participating computers also donate some supporting amount of disk storage space, RAM, and network bandwidth, in addition to raw CPU power.

Many volunteer computing projects, such as BOINC, use the CPU scavenging model. Since nodes are likely to go "offline" from time to time, as their owners use their resources for their primary purpose, this model must be designed to handle such contingencies?

VII. The Architecture Of The Grid

This description of grid architecture does not provide a complete enumeration of all required protocols and services but it identifies the requirements for general class of components. This architecture organizes the components into layers as shown in Figure 5.

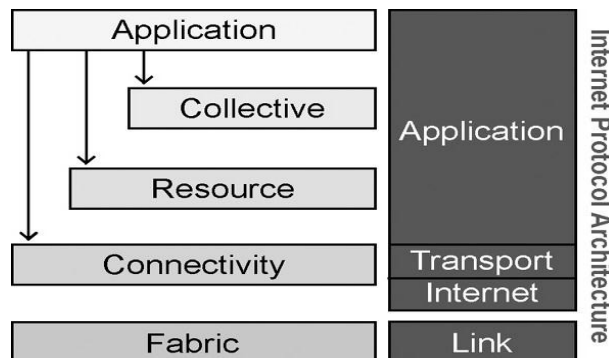


Fig.5.The Architecture of the Grid

7.1 Fabric Layer

- It provides protocols and interfaces to resource being shared.
- This layer provides the resources, which could comprise computers (PCs running Windows NT or UNIX), storage devices and databases. The resource could also be a logical entity such as a distributed file system or computer pool. Excellent fabric functionality could mean that sophisticated sharing operations can be accomplished. For this, it should support enquiry mechanisms to discover their state, structure and capabilities. It should also have resource management mechanisms that provide some control of delivered quality of service.

7.2 Connectivity Layer

- It provides protocols for grid-specific network transactions (IP, DNS, WSDL); Security implementation (GSI).
- This layer consists of the core communication and authentication protocols required for transactions. Communication protocols enable the exchange of data between fabric layer resources. Authentication protocols provide secure cryptographic mechanisms for identifications of users and resources. For communication transport, naming and routing are required. These protocols can be drawn from TCP/IP protocol stack.

7.3 Resource Layer

- It provides protocols to initiate and control sharing of local resources (GRAM, GridFTP and GRIS).
- This layer builds on the Connectivity layer communication and authentication protocols to define Application Program Interfaces (API) and Software Development Kit (SDK) for secure negotiation, initiation, monitoring, control, accounting and payment of sharing operations. The protocols, which the resource layers implement to achieve the above functionality are implemented with the help of functions provided by the Fabric layer. Resource layer protocols can be distinguished primarily into two classes, which are Information Protocols and Management Protocols

7.4 Collective Layer

- It provides protocols for system-wide deployment.
- This layer is different from the resource layer in the sense, while resource layer concentrates on interactions with single resource; this layer helps in coordinating multiple resources. Its tasks can be varied like Directory Services, Co-allocation and scheduling, monitoring, diagnostic services, and software discovery services.

7.5 Application Layer

- It provides protocols targeted at a specific application or class of applications.
- This layer consists of the user applications and programs and which call upon another layer.

VIII. The Sensor Grid

A sensor grid integrates wireless sensor networks with grid computing concepts to enable real-time sensor data collection and the sharing of computational and storage resources for sensor data processing and management. It is an enabling technology for building large-scale infrastructures, integrating heterogeneous sensor, data and computational resources deployed over a wide area, to undertake complicated surveillance tasks such as environmental monitoring.

The concept of a sensor grid was first defined in the Discovery Net project where a distinction was made between “sensor networks” and “sensor grids”.^{[3][4]} Briefly whereas the design of a sensor network addresses the logical and physical connectivity of the sensors, the focus of constructing a sensor grid is on the issues relating to the data management, computation management, information management and knowledge discovery management associated with the sensors and the data they generate, and how they can be addressed within an open computing environment.

In particular in a Sensor Grid is characterized by:

- **Distributed Sensor Data Access and Integration:** relating to both the heterogeneity and geographic distribution of the sensors within a sensor grid and how sensors can be located, accessed and integrated within a particular study.
- **Large Data Set Storage and Management:** relating to the sizes of data being collected and analyzed by multiple users at different locations for different purposes.
- **Distributed Reference Data Access and Integration:** relating to the need for integrating the analysis data collected from a Sensor Grid with other forms of data available of the Internet.
- **Intensive and Open Data Analysis Computation:** relating to the need for using a multitude of analysis components such as statistical, clustering, visualization and data classification tools that could be executing remotely on high performance computing servers on a computational Grid.

8.1 Uses of the Sensor grid

The sensor grid enables the collection, processing, sharing, visualization, archiving and searching of large amounts of sensor data.

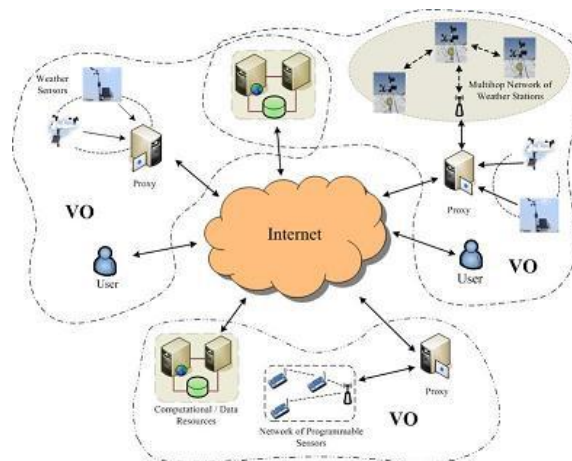


Fig.6. Typical sensor grid architecture

There are several rationales for a sensor grid. First, the vast amount of data collected by the sensors can be processed, analyzed, and stored using the computational and data storage resources of the grid. Second, the sensors can be efficiently shared by different users and applications under flexible usage scenarios. Each user can access a subset of the sensors during a particular time period to run a specific application, and to collect the desired type of sensor data. Third, as sensor devices with embedded processors become more computationally powerful, it is more efficient to offload specialized tasks such as image and signal on the sensor devices. Finally, a sensor grid provides seamless access to a wide variety of resources in a pervasive manner. Advanced techniques in artificial intelligence, data fusion, data mining, and distributed database processing can be applied to make sense of the sensor data and generate new knowledge of the environment. The results can in turn be

used to optimize the operation of the sensors, or influence the operation of actuators to change the environment. Thus, sensor grids are well suited for adaptive and pervasive computing applications.

8.2 Applications of Sensor grid

A sensor grid based architecture has many applications such as environmental and habitat monitoring, healthcare monitoring of patients, weather monitoring and forecasting, military and homeland security surveillance, tracking of goods and manufacturing processes, safety monitoring of physical structures and construction sites, smart homes and offices, and many other uses currently beyond our imagination. Various architectures that can be used for such applications as well as different kinds of data analysis and data mining that can be conducted.

IX. Grid-Related Standards Bodies

For Grid-related technologies, tools and utilities to be taken up widely by the community at large, it is vital that developers design their software to conform to the relevant standards. For the Grid community, the most important standards organizations are the Global Grid Forum (GGF) ^[5], which is the primary standards setting organization for the Grid, and OASIS, a not-for-profit consortium that drives the development, convergence and adoption of e-business standards, which is having an increasing influence on Grid standards.

Other bodies that are involved with related standards efforts are the Distributed Management Task Force (DMTF), here there are overlaps and on-going collaborative efforts with the management standards, the Common Information Model (CIM) and the Web-Based Enterprise Management (WBEM). In addition, the World Wide Web Consortium (W3C) is also active in setting Web services standards, particularly those that relate to XML.

9.1 Open Grid Forum

The Open Grid Forum (OGF) is a community of users, developers, and vendors for standardization of grid computing. It was formed in 2006 in a merger of the Global Grid Forum and the Enterprise Grid Alliance. The OGF models its process on the Internet Engineering Task Force (IETF), and produces documents with many acronyms such as OGSA, OGSi, and JSDL.

The concept of a forum to bring together developers, practitioners, and users of distributed computing (known as grid computing at the time) was discussed at a "Birds of a Feather" session in November 1998 at the SC98 supercomputing conference.^[6] Based on response to the idea during this BOF, Ian Foster and Bill Johnston convened the first Grid Forum meeting at NASA Ames Research Center in June 1999, drawing roughly 100 people, mostly from the US. A group of organizers nominated Charlie Catlett (from Argonne National Laboratory and the University of Chicago) to serve as the initial chair, confirmed via a plenary vote was held at the 2nd Grid Forum meeting in Chicago in October 1999.^[7] With advice and assistance from the Internet Engineering Task Force (IETF), OGF established a process based on the IETF. OGF is managed by a steering group.

9.1.1 OGF Technologies

Some technologies specified by OGF include:

- GridFTP: Extensions to the File Transfer Protocol for high-speed, secure, and reliable data transfer.
- The Grid Laboratory Uniform Environment (GLUE) is an information model, similar to a database schema, for a uniform representation of grid computing resources. It was originally developed as part of the Enabling Grids for E-science (EGEE) project in Europe, which had worked on a grid information system using tools such as the Globus Toolkit through about 2004.^[8] The working group was formed on 28 January 2007. A GLUE schema version 1.3 was published as a draft in February 2007 and final form in August 2008. A 2.0 document was published in March 2009. The abstract schema can be mapped into specific data models using Extensible Markup Language (XML), Lightweight Directory Access Protocol (LDAP), or Structured Query Language (SQL).
- SAGA: The *Simple API for Grid Applications* describes an interface for high-level grid application programming.
- Open Grid Services Architecture (OGSA) describes a service-oriented architecture grid computing environment for business and scientific use.

9.2 Object Management Group

Object Management Group (OMG) is an international, open membership, not-for-profit computer industry standards consortium. OMG Task Forces develop enterprise integration standards for a wide range of technologies and an even wider range of industries. OMG's modeling standards enable powerful visual design, execution and maintenance of software and other processes. Originally aimed at standardizing distributed

object-oriented systems, the company now focuses on modeling (programs, systems and business processes) and model-based standards.

Founded in 1989 by eleven companies (including Hewlett-Packard, IBM, Sun Microsystems, Apple Computer, American Airlines and Data General), OMG's initial focus was to create a heterogeneous distributed object standard. The founding executive team included Christopher Stone, Richard Soley, Bill Hoffman and John Slitz. As of November, 2012, the leadership includes Chairman and CEO Richard Soley, President and COO Bill Hoffman and Vice President and Technical Director Andrew Watson.

Since 2000, the OMG's International Headquarters have been located in Needham, Massachusetts. In November, 2012, the headquarters was moved from 140 Kendrick St to 109 Highland Ave.

The goal was a common portable and interoperable object model with methods and data that work using all types of development environments on all types of platforms.

In 1997, the Unified Modeling Language (UML) was added to the list of OMG adopted technologies. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering.

OMG provides only specifications, and does not provide implementations. But before a specification can be accepted as a standard by OMG, the members of the winning submitter team must guarantee that they will bring a conforming product to market within a year. This is an attempt to prevent unimplemented (and unimplementable) standards.

Other private companies or open source groups are encouraged to produce conforming products and OMG is attempting to develop mechanisms to enforce true interoperability.

X. Conclusion

Grid computing has serious consequences and its implications are enormous in the field of computing. Grid Resources can be used to solve complex problems in many areas like high-energy physics, biophysics, nuclear simulations, weather monitoring and prediction, financial analysis, chemical engineering etc. Projects, such as SETI@Home and Distributed.Net, build grids by linking multiple low-end computational resources, like PCs, from the Internet to detect extraterrestrial intelligence and crack security algorithms respectively. The main strategy of grid computing is to use middleware to divide and apportion pieces of a program among several computers, sometimes up to many thousands. Grid computing involves computation in a distributed fashion, which may also involve the aggregation of large-scale clusters.

References

- [1]. "SETI@Home Credit overview". BOINC. Retrieved April 21, 2010.
- [2]. "A Gentle Introduction to Grid Computing and Technologies" (PDF). Retrieved May 6, 2005.
- [3]. Hingne, V.; Joshi, A.; Houstis, E.; Michopoulos, J. On the Grid and Sensor Networks. In Proc. Of International Workshop on Grid Computing. Pages 166-173. Nov. 2003.
- [4]. H.B. Lim, et al. Sensor Grid: Integration of Wireless Sensor Networks and the Grid, In Proc. of the IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05), October 2005.
- [5]. GGF, <http://www.ggf.org>.
- [6]. "William E. Johnston". ESNNet bio page. Lawrence Berkeley National Laboratory. Retrieved 28 July 2013.
- [7]. "The Grid Forum". Original web site. Archived from the original on 2 October 1999. Retrieved 28 July 2013.
- [8]. "Glue Usage within EGEE". Old project wiki. CERN. 21 June 2011. Retrieved 28 July 2013.