Command and Control Grid Systems

Grid Paradigm applicability to the U.S. Department of Defense

Version 1.0 final, 15 November 2005

Prepared for:

Defense Information Systems Agency
5275 Leesburg Pike, Falls Church, VA 22041
GCCS-J Program Management Office

Prepared by:

Michael Behrens, R2AD, LLC
Peter Ziu, Northrop Grumman
William Huey, R2AD, LLC

"Possibly the single-most transforming thing in our forces will not be a weapons system, but a set of interconnections and a substantially enhanced capability because of that awareness. "-Secretary of Defense Donald Rumsfeld, August 9,2001.

1 Introduction

The Grid Computing paradigm asserts that it is capable of providing a standard enterprise infrastructure, offering a foundation for services and applications that satisfy DISA's design goals of integrated yet loosely coupled system components, standardized security, availability, and reliability. This study delves into how grid computing can be employed to reduce overall enterprise complexity while at the same time providing increased levels of availability, interoperability and information sharing.

Grid Computing currently provides examples of the above design traits as seen through the use of grids within the scientific and academic communities. In life sciences for example, grids provide valuable insight into weather and earthquake prediction, tools for managing real-time data from many sensor feeds to produce predictions and perform assessments, and to assist in the identification of the human genome, dealing skillfully with huge real-time data sets. The U.S. Government, notably the DOE, NASA, DARPA, and certain branches of the Armed Services, is using grid computing. One question this paper attempts to answer is whether or not it is ready for the DoD-wide decision-making support systems.

Recently grid based technologies have adopted core web services creating new market places that have found a significant role in enterprise computing.[1] Inevitably, either grids shall become the predominate container for applications or the J2EE/.NET or containers shall evolve into a grid. Steps regarding the latter are evident in JSR-208, Java™ Business Integration (JBI).

This paper describes how this convergence of technologies makes grid technology relevant today's climate and improves the knowledge-based decision making process for the war fighters.

1.1 Project and Document Overview

This paper is a deliverable for a grid study and is an *initial* investigation and analysis into the potential uses of grid technologies within DISA's Command and Control (C2) and intelligence systems. The potential impact to infrastructure and applications to current and future systems including the Global Command and Control System Joint (GCCS-J), Joint Command and Control (JC2), Net Centric Enterprise Services (NCES) and other related intelligence and C2 capabilities shall be evaluated.

Section 2 introduces the concept of a grid and provides as a synopsis of the grid standard developments and their impact on distributed computing. Section 3 provides an overview of various grid capabilities that are applicable to the DoD.

Throughout this report are recommended courses of action to best utilize and incorporate grid technology where applicable to command and control intelligence systems.

During the writing of this report and over the past 1.5 years, DISA was represented at the Global Grid Forum (GGF)[2]. Various working group meetings and teleconferences have been attended such as the bi-weekly architecture Open Grid Services Architecture (OGSA) collaboration teleconference.

Note: *Distributed computing philosophies are evolving rapidly. If this document in time misrepresents current technologies or histories, the author apologizes in advance and will*

---

[1] Oracle 10G, Sun Grid, IBM's On-Demand, HP Utility Computing, Sybase Avaki, etc.
Example business adoption: http://www-1.ibm.com/grid/grid_press/pr_104.shtml
[2] Global Grid Forum – http://www.ggf.org and http://www.gridforum.org/GGF15/presentations/GGF101_GGF15.pdf

endeavor to correct immediately in response to audience feedback. Feedback can be provided directly to the GCCS-J PMO or via the collaboration website: http://grid.geden.org.

2 Grid Systems

The term *Grid System* refers to the emerging computational and networking infrastructure that is designed to provide pervasive, uniform and reliable access to data as well as heterogeneous computational resources distributed over a wide area. The grid environment creates a virtual computer out of many computers that enable users at locations throughout the world to share data such as sensors feeds, clusters of servers and workstations, and community datasets stored on network caches and hierarchical storage systems.

A grid system is a collection of distributed heterogeneous resources connected by a network.[3] A Grid is a coordinated distributed computing infrastructure formed by combining *heterogeneous* resources from ad-hoc administrative domains, such as an ensemble to support large-scale, resource-intensive, and distributed applications.

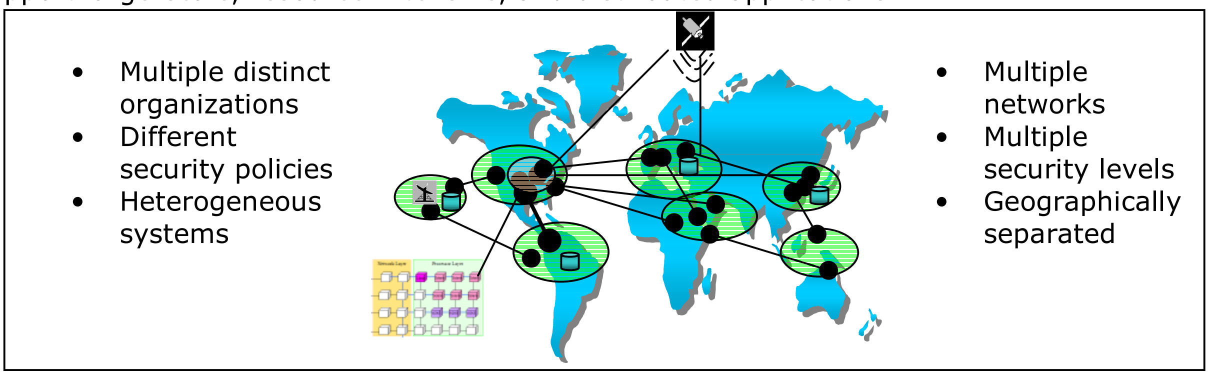


Figure 1 - Enterprise Computing

From a DISA perspective, this would mean that each Combatant Command would participate in one or more grid systems. COCOM resources could be shared within their own grids and externally to other COCOMs or agencies based on security and access policy. Indeed, ad hoc grids can be dynamically established to support collaboration environments that can interoperate with other grid systems, even those in coalition command elements.

Essentially, a grid is a virtual single computer with a controlled sandbox. It extends the popular concept of the Java Virtual Machine to the entire network to span multiple machine resources. *Resources* are allocated to be shared in a grid and normally include computational resources (CPUs), storage (disk space), memory, information (data), and even network services.

When one site needs more resources, it can borrow them dynamically from other sites as needed to accomplish their immediate real-world situation. The resources in a Grid are shared via grid standards that provide mechanisms for local site policies along with a coordinated global grid policy. The combined distributed heterogeneous resources from the administrative domains create a logical resource namespace.

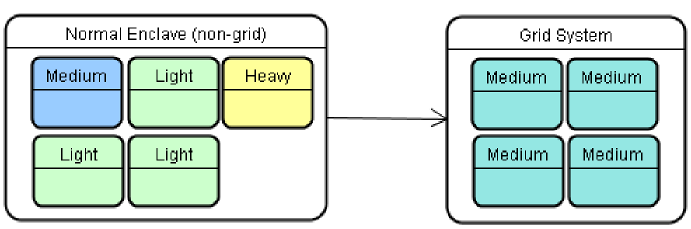


Figure 2 - fewer nodes, greater efficiency, lower cost of ownership

[3] OGSA – The Architecture for Grids, Andrew Gimshaw, David Snelling, Sep 2005

All the grid resources can be thought of as belonging to a pool of resources, ready to be used as needed or On-Demand (as IBM states). In this figure, note how the overall resource utilization can be managed.

2.1 Use Case

The specifications for grids are driven by use cases, sometimes referred to as scenarios. Therefore a use case is presented here to provide an example of how grid systems might be applied to a particular problem space within the command and control systems.

Radars and other sensors are scanning the world and picking up tracks which are recorded by many different server nodes. Properties of the sensor data can then be analyzed and correlated and disseminated everywhere via channels. Each recorded track is comprised of a large number of data attributes. Thousands to millions of events are recorded each day yielding a total dataset size of hundreds of thousands of Gigabytes (or hundreds of Terabytes) if they were to be all kept (currently, all this data is not retained).

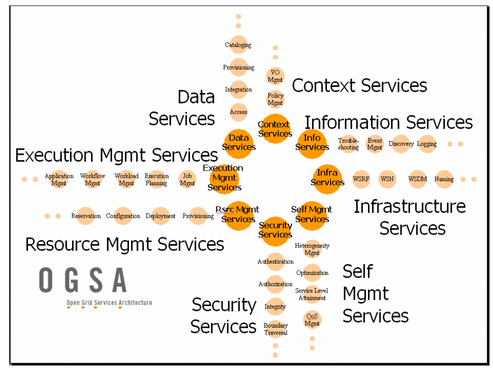


Figure 3 - OGSA Services

A War Fighter might be interested in performing an analysis to assist with their mission, such as a route calculation. Such an analysis is computationally intensive owing to the size and number of the events and data elements involved. A good route algorithm would utilize the tracks and their histories, current terrain data, current weather, intelligence information, logistics/planning data, and readiness information.

If a route analysis is requested on a single computational node it would take a long time, especially if all the information were required to be obtained on that single node. If this algorithm were requested for multiple tracks, then the computational and data throughput requirements are increased. Compounding this would be if there are many users also desiring a route analysis. The amount of time required can be overwhelming.

Grid computing provides an architected solution. Grid middleware, working as an enterprise Operating System (OS), provides the necessary infrastructure:

- many worker nodes managed by a distributed grid engine
- requests directed by resource brokers and schedulers
- service metadata managed by the location services
- high volume data are kept on storage elements by the data services

The route analysis can be divided among all available nodes. Storage space can be allocated to keep pertinent data available based on usage and data pedigree. High performance parallel transfer tools can be used to keep data cached and up to date without each application knowing where the data actually resides. Furthermore, the best route can be continuously updated as a background task using computation resources when and where needed.

2.2 Grid Standards

Many grid systems today do not interoperate because different grid systems incorporate different grid middleware (infrastructure) that do not implement the same interface protocols.

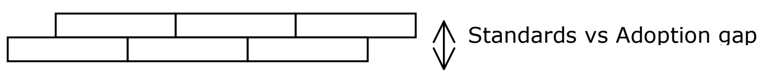
The Global Grid Forum (GGF) is an open standards organization that has the goal of standardizing a grid framework to achieve interoperability between and amongst virtual organizations [Anatomy]. As stated on their website (http://www.ggf.org), "*As a standards-*

development organization, GGF has a well-defined, open standards development process; strong ties to other standards development organizations such as IETF, OASIS, DMTF, W3C and SNIA and a growing pipeline of specifications under development." The OGSA-WG within GGF defines the flagship architecture and the blueprint for industry standard grid computing.

Each standards body generally focuses on a specific technology area. Grid technology tends to overlap many of the areas. The GGF has turned over specifications to some of the other organizations for final ratification. Likewise, members of the GGF interact and collaborate with the other bodies, promoting a unification of standards. This is especially true with regard to the Enterprise Grid Alliance (EGA) which might be merging with the GGF in the coming year[4].

Standards help mitigate risk and promote interoperability. This is especially important for the DoD where the abilty to make the best decision with the best information is critical to success. At the same time however, the evolution of standards is a moving target. At times it may be necessary, although not ideal, to create a DoD proprietary specification or to adopt a commercial product which is not standards based.

It is important to note that standards evolve. They tend to reinvent themselves over time. The DoD can either follow the standards organization's lead, or get involved and participate in the standards process, or generate DoD proprietary standards. We recommend that involvement would provide the best outcome for the defense community.

Specification Alignment: 

In the past few years, GGF has adopted SOA and web services. This decision has created a convergence between the vast amounts of grid technologies with the emerging web-service world. This convergence means that grid computing is now "net-centric".

2.3 Globus Toolkit

The Globus toolkit is an open source grid implementation by the Globus Alliance, which includes Argonne National Laboratory and others. Globus has been in development since the late 1990s to support service-oriented distributed computing. It currently leverages web services such as Web Service Resource Framework (WSRF) (described later) and therefore provides a Service Oriented Infrastructure (SOI) for grid systems. Globus has close ties with the GGF and a future GGF conference will co-exist with GlobusWorld/GridWorld[5].

The Globus Consortium[6] is an industry consortium that helps accelerate the adoption of Globus. Univa is a commercial support mechanism for the Globus Toolkit that provides support and can be engaged for specific tasks and can even ensure that the requirements of the DoD are incorporated into Globus in a timely manner. Globus has its roots in funding from the Department of Defense via DARPA.

It provides all the necessary components (shown in figure 4 below) [GT4] to establish a complete grid system, namely:

- Security infrastructure
- Resource access and management
- Data transport
- Task execution

[4] Mark Linesh's announcement: http://www.ggf.org/Announcement/ggf_announce.php
[5] Globus Implementation: http://www.globusworld.org/
[6] http://www.globusconsortium.org/

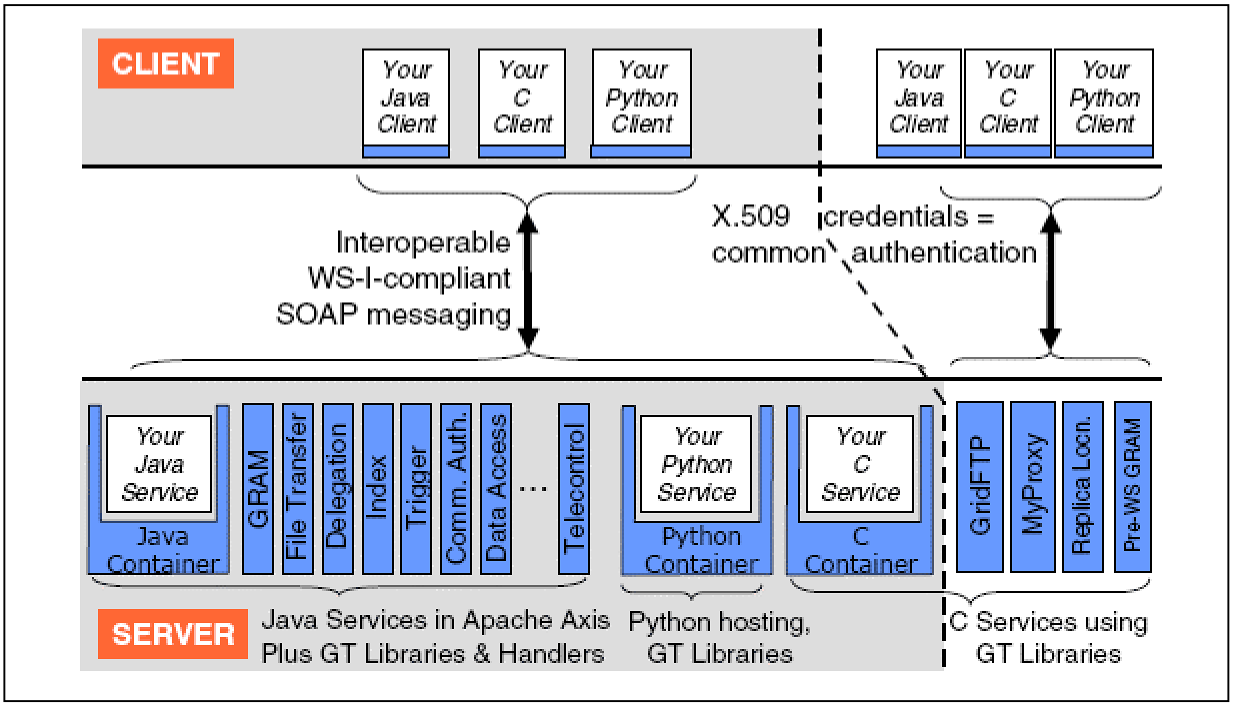


Figure 3 - Globus Toolkit 4

There are other grid middleware[7] implementations as well, such as UNICORE which is funded by Germany and other corporations. Gridbus is another, which is led by the University of Melbourne. The Enabling Grids for E_SciencE (EGEE) project also produces grid middleware such as GLite and is co-funded by the European Commission. The Open Middleware Infrastructure Institute (OMII) provides a web service infrastructure for building grid applications.[8]

All grid projects do share the same vision that is to turn the global network of computers into a vast virtual machine[9]. In time we expect that the variants will unite under a future single standards banner and will all implement the specifications that will enable interoperability. For the purposes of GCCS-J, JC2, and NCES, the Globus toolkit is recommended since it has excellent support within the United States and is currently being supported and used by the U.S. Government.

3 Grid Features

There are many features and capabilities offered by grid technologies. Generally, standing up a grid infrastructure today is a daunting integration task since there are so many choices and components that can be composed to meet different needs. The following sections contain information on specific aspects of grid computing technologies that are relevant to the GCCS Family of Systems (FoS) and future systems of records such as JC2 and NCES.

3.1 Parallel and Non-Parallel applications

Most grid systems expect the applications to be developed with the grid in mind and therefore must be made parallel; meaning the problem being solved is separated into asynchronous tasks and executed at the same time. The answer is then attained much

[7] Grid middleware comparison: http://www.gridbus.org/papers/gmchapter.pdf
[8] OMII Implementation: http://www.omii.ac.uk/
[9] http://gridcafe.web.cern.ch/gridcafe/index.html

faster and more reliably. The use case mentioned earlier is an example of a problem that could be divided up between multiple computers or grid nodes based on geographic location.

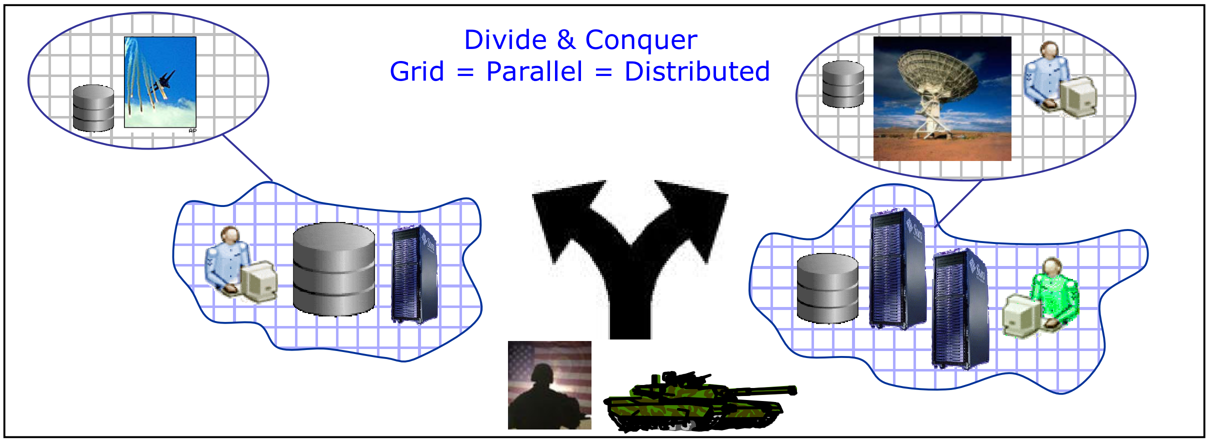


Figure 4 - Reliability and Availability

However even those tasks which are not parallel can be still executed on a grid. The application meta-data would detail that it requires only a single node. Furthermore, applications of this nature could have an indefinite lifetime. This would provide a way to execute legacy applications within a grid with little difficultly.

For best results it is recommend that applications be designed with the grid nature in mind.

3.2 Clustering vs Grids

Clusters, such as J2EE clusters, provide a way to distribute HTTP requests between two nodes to ensure availability in the event one node crashes. The cluster model is one that supports many users to one cluster. If a cluster consists of more than two nodes, then as more users login, the load is spread across the cluster to help distribute the load. In general terms requests from a single user execute identically on two nodes. In addition to J2EE clusters, database clusters can also help with data access performance and availability.

Grids are more abstract than clusters. The grid model conversely is one in which many users interact with grid nodes and therefore to many resources. A grid resource could be a cluster or a single resource. J2EE containers (clustered or not) are in fact needed by grid systems since most grids use web services hosted mainly in J2EE containers. To a grid, a cluster is just another available resource onto which applications and services can be executed. Grid nodes expose a grid compliant manageability and can be monitored and managed as needed to ensure QoS.

J2EE applications themselves could invoke the grid to perform tasks or workflows on behalf of a user. Indeed, most grid systems today use a web portal as the primary user interface. However in order to take advantage of grids to the fullest, grid tasks should be written to be parallel in nature. An application might, for example, be crafted to execute in many J2EE instances simultaneously.

In GCCS-J today, the I3 IMOM software has already been designed to be parallel, however, the infrastructure is not present in the standard global build to provide the engine. If the Globus software suite were segmented and made available for developers, then the infrastructure would be present and thus design changes could be made to take advantage of it. In addition to IMOM, many other applications can also be parallelized such as correlation, image processing, route analysis, document/file conversion/transformation[10], and weather modeling, to name a few.

3.3 Stateful Web Services

Web Services Resource Framework (WSRF)[11] is an OASIS specification that defines an open framework for modeling and accessing stateful resources using Web Services. Understanding WSRF enables one to understand current grid technology and its influence on web services. Web Services are traditionally stateless. However, many services need to access state eventually, typically stored in back-end database. WSRF is a combination of a "stateless" web service and introduces "stateful" web service resources (WS-Resource). It provides a standardize way to provide and set state which also includes the lifecycle of the state in grid systems.

Essentially, WSRF brings a level of object orientation to web services since it provides for a concept of an object (the WS-Resource) and its methods (the service interface). It also defines a life cycle to control the instantiation process. The resource properties can be obtained via the specified getter/setter pattern. While WSRF does not define or create a method, the resources are referenced via an Endpoint Reference (EPR). The EPR created via a factory pattern portType implementation acts like a constructor method. EPRs are essentially pointers to a specific instance of a resource or object and are defined by WS-Addressing[12].

Resources can be grouped together using the WS-ServiceGroup specification to create object containers or aggregations. Furthermore, OASIS provides for a standard way to notify interested parties of changes to properties via the Web Service Notification (WSN) specifications that define, publish and subscribe asynchronous notification mechanisms for status change.

Couple all this with a globally unique time safe abstract name as being proposed by the WS-Naming working group, one can see how a grid infrastructure based on web services can provide a robust scalable distributed information grid.

This is directly applicable to many functions, most notably for implementing the concept of Operational Context in NCCP/JC2 in a standard way. Every track, for instance, can be a resource which is directly accessible via a network interface. This software stack (and other similar ones[13]) can be used to provide great advantages to the challenges facing DISA.

3.4 Enterprise Management

The grid views all resources (computers, routers, hard drives, etc) as resources. Furthermore, the grid views all applications as resources too. Since all resources should expose a manageability interface as per the grid profile specifications, enterprise management becomes possible. The related specifications include the Web Services Distributed Management (WSDM) from OASIS, Web-Based Enterprise Management (WBEM), and the Common Information Model (CIM) schema. The latter two are defined by the Distributed Management Task Force (DMTF)[14].

Resources are defined to have a lifecycle that can be monitored and changed. All software applications also have a lifecycle, however most current applications do not implement this level of manageability and administrators are therefore forced to "kill" the application as a way of restarting.

---

[10] Sun Grid Transformation Services: http://java.sys-con.com/read/148842.htm
[11] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf
[12] Web Service Addressing working group: http://www.w3.org/2002/ws/addr/
[13] http://www.cs.virginia.edu/~humphrey/papers/AlternativeSoftwareStacksOGSA.pdf
[14] Report on web base management: http://xml.coverpages.org/dmtf-cim.html

Java also provides a Java specific management interface available now called Java Management Extensions (JMX). The grid interfaces are language impendent and are exposed as web services. Also, many existing grid systems present a web portal to users on which they depict a status table indicating near-realtime up/down status of the key components in the grid (i.e.: stoplight chart).

Automation is a key capability desired by the IT world. GCCS-J today, for example, is manually managed for the most part. All software is manually installed. If a system crashes, it is manually swapped. Redundant devices such as RAID enabled storage and redundant power supplies, help provide robustness. The grid extends the same concept to all major components by providing access to distributed resources. The grid provides a common management interface that enables all resources to be utilized in a systematic way throughout the enterprise.

In a grid system, the grid engine or middleware is able to ensure that systems are executing and can take corrective action in the event of a failure or surges in load. Since every machine on the grid is an available resource, it can be allocated as needed to take advantage of available processor, disk, memory, or other resources.

Management of the grid is a reason by itself for wide adoption since it will drive the cost of operation down. For GCCS-J and JC2, this means that the many fielding teams will not need to leave DISA. Instead, the systems can be provisioned and managed using the grid infrastructure.

Since the grid is a logical single virtual computer containing virtual resources, tasks can execute on it to perform many different management tasks such as provisioning, backups, cleanup, log searches, processing checking, health monitoring. Administrative domains could be either different organizations (forming an inter-organizational grid or virtual organization), or within the same organization (forming an intra-organizational grid or enterprise grid).

3.5 Software Provisioning

Programs running on the grid are generally installed automatically, initiated by user interaction. Current scientific grids, for example, use the notion of "jobs" to submit the code to be executed in computation grids. In the mainframe era of computing, jobs were also used to submit applications to run on the system where it was queued and executed based on its priority and parameters specified in the Job Control Language (JCL). Grid systems today have a similar construct, however the jobs can be long-term running services. Each job or task is specified with dependencies, meta-data exposure of properties like ports used, memory needed, disk space required, version #, etc. All of these are similar to what the current COE Kernel provides (boot processes, login processes, and the COE Installer). Constraints also include service level agreements (SLA), such that if a system is not performing at an acceptable SLA, autonomic steps can be taken, such as auto scaling to meet the agreement.

GCCS-J System Engineering is studying the installer issue and will work with the COE transition team to effect a positive change that meets system objectives. The installation process needs to be capable of providing support for: security, functional oriented menus for users (i.e: Start | Program | GCCS-J | Intelligence), location of key standardized files, Metadata location for external inspection, Solaris support, registration of names, and ports to prevent duplication (prefixed?), etc.

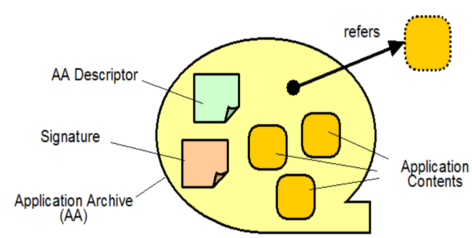


Figure 5 - ACS Trusted Archive

The GGF has published specifications to help provide for a standard approach to installing components into a grid system. For instance, the Application Content Services (ACS) specification provides a foundation for a standardized secure software repository for all segments that would be supported by true network installation. ACS defines a trusted hierarchical repository for grid systems. It recommends the use of OASIS SDD compliant packaging provided by tools such Macrovisions's Software Architect[15]. CDDLM is another GGF specification that helps to deploy services automatically and provides function similar to J2EEDT.

Incorporating this technology into GCCS-J/JC2/NCES help transform current processes into a net-centric accountable system.

3.6 Bandwidth Expansion

The Global Information Grid (GIG) BE (Bandwidth Expansion) effort establishes a huge fiber optic network pipe for use by the collection of applications and systems which shall comprise the Global Information Grid Enterprise Services (GES).
Grid technology provides the management and application infrastructure for the GES. The security, provisioning, parallel data transfers, CPU economy, storage digitization, and other advances will help GES achieve these goals.

This technology is expected to provide a capabilities-based infrastructure required for timely, secure, and ubiquitous edge user access to decision-quality information. This information can then be used by dynamically formed collaboration groups (virtual organizations) to solve problems. GCCS-J and the follow-on program JC2 will be a critical part of the GES for the purpose of strategic command and control.
Grid technology is most likely initially fielded DoD data centers such as the Defense Engineering Computing Centers (DECCs) and the Regional Support Centers (RSCs).

3.7 Data Access

Data grids offer the potential access to all data as if it were local. Virtualized data access is the primary purpose of the Grid Data Services (GDS). This exists in Globus and is being standardized by working groups such as OGSA-D and the Grid File System (GFS) working groups. Current implementations of the services provide an abstraction layer around databases such as Oracle or MySQL. Commercial products also provide this capability, notably the Avaki product that was recently acquired by Sybase. This means that data grid technology is now being integrated into Sybase's core capabilities and is available today. I^3 could take advantage of this, for instance, to establish a globally replicate scalable data grid.

Recent efforts have expanded the notion of mediation and embrace the global distribution and access to data regardless of underlying storage or transport protocol.

Grid users and applications require a logical namespace for using and managing distributed data and other shared resources within the grid. To help with this, the "context" is also being standardized which includes the simple way to access data using a multi-tier naming system with lookup services. The appropriate analogy would be DNS, which is used to distribute IP address and their names. Using the same pattern, network references to data can be used and shared, knowing that they can be resolved when and where needed.

"*As more and more applications move towards distributed storage and processing solutions, new and interesting issues arise. Applications must handle varying, sometimes extremely large data sizes and increasingly complex data models*."[16]

---

[15] http://www.macrovision.com/products/solution_architect/index.shtml
[16] Joel Saltz from The Ohio State University, DIALOGUE workshop interview

3.8 Edge Caching

Data grid systems generally advocate the need for data caching. Caching provides standardized transport layers, which support the massive movement of data between nodes. In a typical grid scenario, data is pulled where it is needed. The data services can then cache the data and manage it based on attributes such as time to live, importance, and frequency of use, etc.

It is interesting to note that the policy language described in the previous section can be used to declare caching properties as well as the quality of data or Quality of Service (QoS) when viewing data as a service.

In the real world, we are constantly being bombarded by many signals (TV, Radio, Cell, and Satellite). Imagine the amount of data that is around each of us everywhere. A TV as a client can consume large amounts of information and keep it displayed in real-time. Although that is a one-way medium, the COP for the most part is a one-way medium. There are some interesting grid transport and replication technologies that can be used to standardize dissemination of data and access to information.

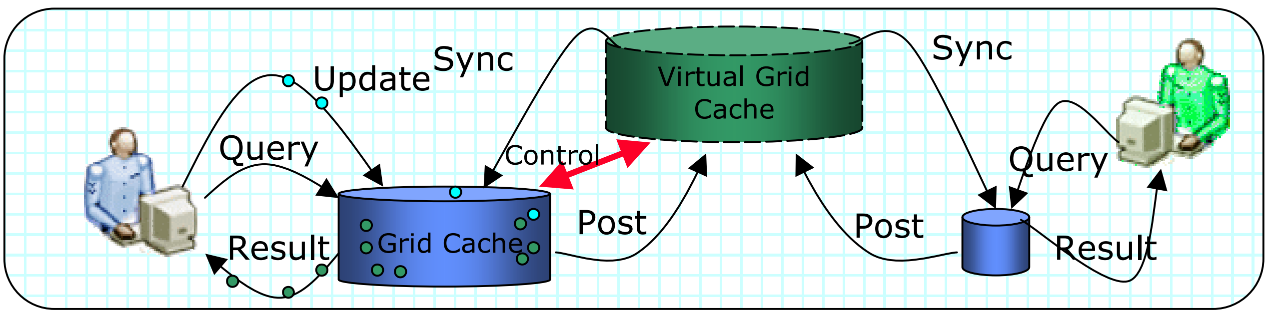


Figure 6 - Data Grid

This brings to mind the related concept of replaying situational awareness. Since the client will be capable of accessing hundreds of thousands of data points - what is preventing that data from being processed on the grid into a form that enables replays in a quick manner? Globus and partner toolsets provide technology to collect, process, and access data using broadband parallel distribution transports such as Grid-FTP.

There are software patterns that lend themselves to be solved via standardized appliances. These can cache data, thereby reducing bandwidth or using it at more appropriate times. Commercial products are available which support the extension of caching beyond the database tier[17]. These products generally are data-grid enablers that help to create an information-fabric for applications to access.

Caching is also another example of the "Meta-OS" model evolution prevalent in grid architecture from which the benefits of years of advancement in cache coherence and other algorithms derived from the processor, memory, and database markets continue to affect the distributed computing world.

It is important however to realize that in the command and control world, all data does not "have" to be kept synchronized "everywhere". This is common pitfall which modern data grids avoid. The important notion is that data is "available everywhere". Availability means that if needed, it can be obtained and cached locally for efficiency purposes. Changes to the data only need to be disseminated to those locations where it is being used. Furthermore, with grid distributed processing, the data may not have to be moved at all, since the results of calculations can be moved instead (derived second order or tertiary data). Bringing back

[17] For example, GemStone's Gemfire or Tangosol's Coherent

the use-case of a route calculation, the requestor of the route does not need all the data behind the decision of which is the best route. This is important to understand as it affects the entire enterprise design and therefore optimal flow of information.

Furthermore, incorporating good data pedigree (or providence) on how the data is modified or accessed helps with the overall integrity and efficiency of the data grid. Such attributes would include read-only, mostly-read, migratory, access frequency, importance, and time to live.

It is interesting to consider that future grid systems could be smarter by providing a dynamic runtime determination of how to best handle data. Data sources that do not provide the meta-information can be analyzed or observed in order to determine data behavior patterns. WSRF provides the hooks for this level of monitoring today. Furthermore, aspect oriented programming techniques could also be employed for this purpose.

Therefore, meta-information is important for the efficient distribution of data.

3.9 Transport

Grid transport technologies such as GridFTP offer a way to keep entire disks synchronized between sites in a fast and reliable manner [GridFTP]. GridFTP is also secure and is an implementation of the published standards. It can utilized multiple computers and Network Interface Cards (NICs) at the same time (parallel) to maximize transfer efficiency. It is recommended that the PMO incorporate this technology since it is available now and will assist with database replication needs (I3 and JOPES) and Image/Video exploitation.

GridFTP is built on a basic transport layer called Extensible Input/Output (XIO) that provides an abstraction layer between applications and the transport layer. This is an evolutionary technology that recognizes patterns in data access and builds an abstract layer around it (common in grid technology). This type of library is one that can replace many legacy file and network data transfers. Applying this to command and control would make information dissemination largely based on a single over-arching mechanism for distributing all data types in addition to track sources.
This diagram depicts how a GCCS-J application can be isolated from the details of various transport or storage protocols:

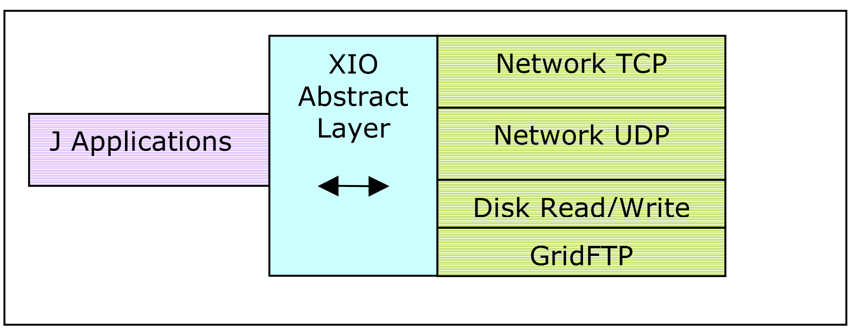


Figure 7 - Extensible I/O Usage

Another interesting capability of transport is the Grid File System (GFS)[18] component. It is similar to UFS or Windows Shared directories, however the underlying physical storage can be anywhere in the grid. The applications are not concerned with physical location or whether it is part of a windows box or a UNIX box.

[18] GFS Working Group: https://forge.gridforum.org/projects/gfs-wg

3.10 Quality of Service (QoS)

One benefit of grid technologies is robustness. When a resource (i.e. service) goes down, the grid monitor can determine whether or not to start another instance somewhere else or not bother, depending on what the QoS agreement is. This is currently the model used by most clustering systems. The difference between grids and clusters is the heterogeneity and geographic dispersion factors.

The grid engine contains a component called Execute Monitoring Services (EMS). EMS is responsible for ensuring that the qualities of service agreements are met. It does this by querying the management mechanisms built into all grid resources and ensures that they are performing within the required levels.

The WS-Agreement or WS-Policy standards are most likely to be used to convey the policy; however there are other possibilities as well.

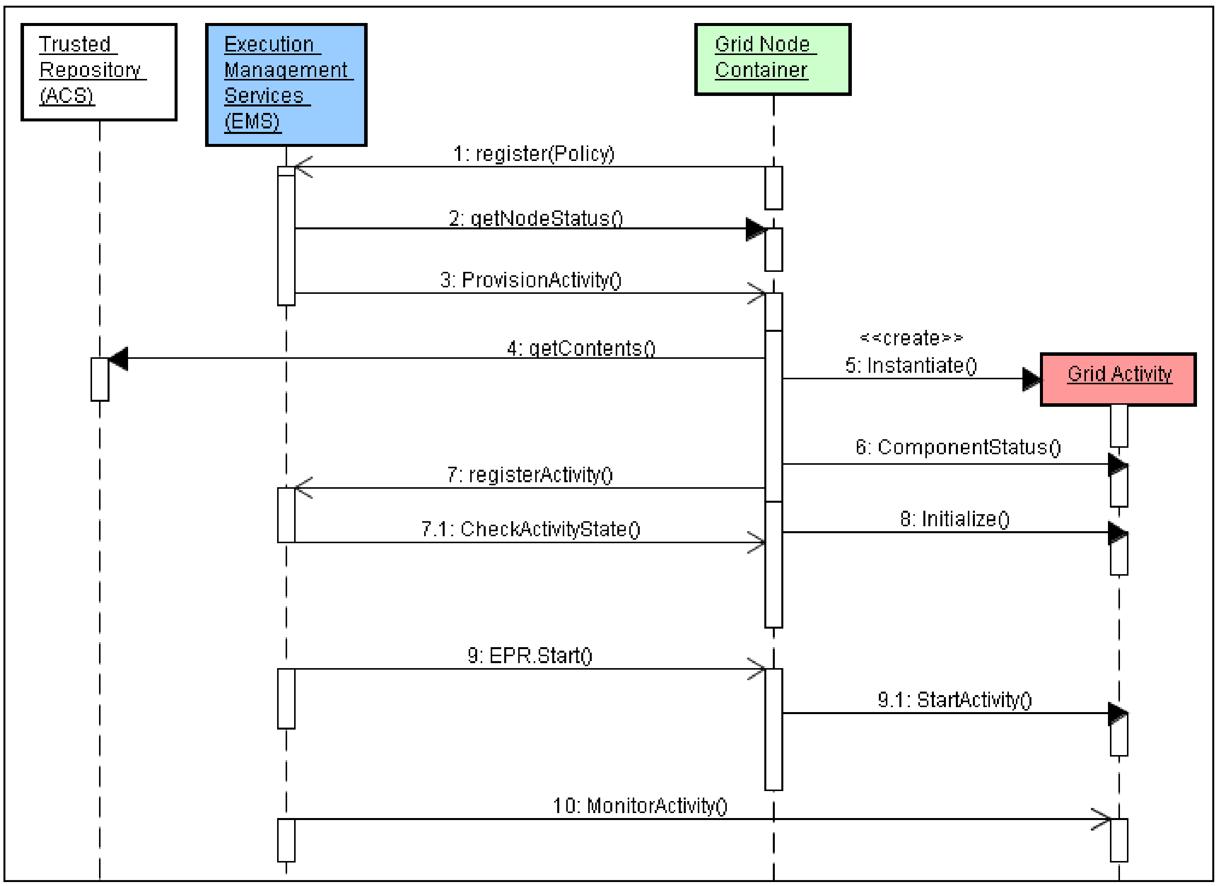


Figure 8 - Typical Execution Management Services

While most grid engines currently provide a monitoring service, they are not based on a standard. The OGSA working group in GGF is working to define this core specification that would provide Execution Management Services (EMS). EMS is responsible for ensuring Quality of Service (QoS) on the system. It has the ability to start, stop, and even migrate resources on demand when needed. Systems from HP, IBM, Sun, and others already advertise this capability for grid systems. The challenge however is enabling resources in a standardized manner.

3.11 Service Publication and Discovery

The grid community does recognize the need for discovery, however not always in the form of UDDI. This is interesting because it opens the possibility for multiple mechanisms for discovery based on consumer requirements.

NCES is counting on using UDDI in order to provide a mechanism for discovering services. It is interesting to note that most of the current code in NCES uses hard-wired IP addresses. The issue of service migration is covered by grid systems, especially those which implement the WSRF specification.

A catalog of services and related information does not help locate the data. A service can provide data and search capabilities (i.e., where are the images of tanks in Iraq) however UDDI is for pre-positioned services. It is not Google. UDDI should not be used for policy enforcement, despite the attempts to market that ill-fated concept.

In the grid space, UDDI might be used to find the location of the dozen or so major services such as security or the scheduler. It would not be used to locate the thousands of other services that are operating to search for data and operate on this data. The grid would use naming services to allow handles to be registered and found as needed, much like a pointer or reference is used in programming languages today. In distributing computing, handles or external references are created and passed around in order for other systems to perform operations needed to accomplish the task. Only those involved in that thread need to know the location of the services and therefore they would not be registered into a UDDI service.

3.12 Security

While security is important in any system, Grid systems take security very seriously since the nature of grids means distributed processing and management. Current specifications for security in grid systems are aligned with the WS-I Basic Security Profile[19]. Grids rely upon PKI technology and have also advanced the use of PKIs in many areas such as proxy certificates and attribute based access controls.[20] Every user of the grid is provided a PKI certificate which enables them access to the virtual system.

Grid Security is based on Web Services Security (WSS). The profile specified by OGSA provides an agreement of security which is required for interoperability. The Globus implementation for instance, defines a near compliant set of services called Grid Security Infrastructure (GSI)[21]. GCCS-J can build on this experience and transform its current systems sooner by using the current technology proven in products such as Globus.

One current use of GSI is to provide single-sign-on capabilities by creating a grid session user ID with a default lifespan of 12 hours. This certificate (x509) allows the user to access the grid resources for the duration of their session. Browsers can be closed and re-opened, for example, and the session will not be orphaned as current thin-client applications behave in other web-enabled applications (including GCCS-J).

Auditing is also performed as a distributed mechanism. By using certificates, identities of individuals and processes can be traced and monitored.

The security aspects of grid computing need to be understood by NSA and the DIA in order for their respective IA policies to evolve to support the net-centric behavior of grid systems. Currently Java Applets are allowed as part of the mobile code policy, dated November 7, 2000. Applets are acceptable from a policy point of view because the sandbox in which they

---

19 http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html
20 http://www.cs.virginia.edu/~humphrey/papers/humphrey_security.pdf
21 GT4 Security: http://www.globus.org/toolkit/docs/4.0/security/

execute is trusted. Grid infrastructure provides a similar sandbox via its virtual nature. For the entire vision of grid computing to be accepted in full and in time, a complete analysis by a security team is recommended, as it is beyond the scope of this paper. However it is important to understand that the very nature of grids depends on being able to operate securely, much like a VPN. So while the security policy challenge is present, it is not that great of a risk and might even be welcomed by the IA policy makers of the DoD.

3.13 Access Control

One of the applications challenges being addressed is fine-grained access control. The current specifications are advocating a policy document that would be associated with the data source in order to describe access rules. Additionally policies would state accuracy and accessibility requirements to ensure Service Level Agreements (SLA).

This is directly in line with actual requirements for track data in GCCS-J as brought out in the 100k track discussions during the CDS CDR at Northrop's facility in San Diego last August 2005. This also pertains to the databases such as MIDB, SORTS, and JOPES. Standard Policy language should be used as a management layer built into the data infrastructure, which would include access to metadata and pedigree. Examples of standardized policy language include XACML[22].

The track data policy or MIDB access policy could be expressed using XACML[23] which provides an XML description of security rules. Track security could be expressed as an XACML document that would include an originator rule as a primary policy attribute. Efforts toward this end could yield a definition of a Track Path metadata that could be used to determine how a track made it into a particular node. Based on this information, the security policy could be defined and enforced wherever the track resides to ensure, for instance, that it can't be deleted from certain locations. This means that the policy would be referenced or would migrate with the data or data collection. This concept should probably be explored further and brought to the attention of the COP working group at the J3 level (Wayne Parks).

The SAFE segment in GCCS-J provides current applications with authentication and authorization and 8500.2 compliant PKI support for thick/rich clients and web instances. This paper recommends that SAFE continue to evolve to adopt the WSS/OGSA Basic Security profile via an implantation of the Globus GSI to become the security infrastructure for a future command and control grid system.

3.14 Mobile Services & Mobile WS-Resources (future)

Implementing current web service technology results in static services that are deployed manually (mostly) and registered with a fixed address or End Point Reference (EPR). The WSRF set of specifications provides a way to obtain the current EPR for a service that is no longer available.

This paper introduces the concept of mobile services, which are dynamically deployed or migrated between machines. They are analogous to mobile agents, except that in addition to executing logic, they expose network services that accept input or provide output. Service Repositories, such as the GGF Application Container Services (ACS) mentioned earlier, provide a residence for services when they are not deployed. Replication tools such as those in Globus can be used move services to the edge, providing quick access when required.

---

[22] http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf
[23] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

While WSRF's Renewable References provide the mechanism to locate where the resource has been moved to, it does not currently provide for the physical transport/relocation of the resource. The OGSA Execution Management Services (EMS) would generally have the ability and security access rights to move a service.

It would be expected that EMS shall be capable of detecting the failure of a grid container and then create a new instance of the WS-resource (stateless service and stateful resource) if QoS warrants. EMS would have to communicate this to the interested parties such as registries and grid activity managers and perhaps provide state information to the newly spawned service(s).

These are a few scenarios where this capability would be needed:

1. If a container stops and the WS-Resource stateful information needs to be recovered and made available elsewhere.
2. The designer of the job intentionally wants it to move around - like a classical mobile-agent. This would be a mobile WS-Resource. Instead of the grid deciding where to run it, the resource moves itself or copies itself to where it needs to go or where other resources are located.
3. Edge caching of resources would require movement of not only the resource but also the service (WSDL + factory). This introduces the concept of mobile web-services, which can greatly increase the efficiency and robustness of a grid system.

The GGF is helping to define these capabilities as part of several working groups: OGSA-Naming, OGSA-BES, and OGSA-EMS. Perhaps a future specification might evolve around the notion of standardizing WS migration.

4 Summary

A key benefit of implementing grid technologies in command and control systems would be a standardized infrastructure environment in support of applications to:

- Globally distribute and access data
- Distribute computational/storage load
- Unify management the enterprise
- Provide service availability

The grid specifications are being written using web services as a foundation, building a grid based Service Oriented Infrastructure (SOI) which represents a convergence of technologies. Command and Control and Intelligence mission applications can take advantage of the grid in several areas such as data distribution & edge caching (data grids) and massive computational problems in the areas of image processing, intelligence preparation of the battlefield (IPB), entity correlation (computational grids), and analysis functions. The grid can capture unused computer resources to increase efficiency by providing a secure means to share data resources, the ability to create interoperable virtual command and control centers, and provide a high enterprise availability and scalability at a lower total cost of ownership.

4.1 Road Ahead

Grid technology is not a silver bullet. It does seem however that it is the evolutionary end-point of distributed computing for the enterprise. The National Science Foundation (NSF), which has helped to fund grid technology, states that grids will be a major focus in the next decade (decade of the grid) as part of their Cyberinfrastructure[24], which will be a critical national resource. The United States is currently spending millions of dollars in grid technology for academic, scientific, and military purposes[25]. Their funding is helping to solidify and commercialize this technology. IT research firm International Data Corp. of Framingham, Mass., expects the grid marketplace to generate $10.3 billion in sales by 2007.

It is important to first understand this new paradigm and then to reap the benefits of the technologies as found in many commercial products available now from corporations like Oracle, IBM, HP, Sun Microsystems, Platform Computing and Sybase, to name a few. Ultimately, standards based grid infrastructure should be the framework for C2 systems.

The recommended priorities for GCCS-J, JC2, and NCES would be to introduce a grid system into the core system and to allow developers time to understand it and adjust their designs to take advantage of the technology and standards as they evolve. Simple grid tools such as GridFTP and grid management can be used immediately and without risk to the current architecture. A segmented Globus could become a core component of GCCS-J/JC2 and be made available on every C2 node. A small team can start building specific grid applications such as route processing, image processing, video mosaic creation, or distributed correlation.

Additionally, it is recommended that grid technologies (GGF, Java with JBI, etc) continue to be studied and monitored in order to continue to bring the advances of distributed computing to the war-fighter.

[24] http://www.nsf.gov/od/oci/CI-v40.pdf
[25] e.g. NASA (http://www.nas.nasa.gov) and the TeraGrid project (http://www.teragrid.org/)

5 Addendum

5.1 Acknowledgements

This effort could not have been realized without the support from the Defense Systems Information Agency (DISA), particularly the PMOs for NCES, GCCS-J, and JC2. Thanks to the GCCS-J I^3 team and their Product Branch Chief, Richard Gragg. Thanks to William Huey, R2AD West Coast for the remote editing and Q&A review. Finally, the guidance and support from Northrop Grumman has been most helpful.

5.2 References

[Anatomy]
I. Foster, C. Kesselman and S. Tuecke, "The Anatomy of the Grid: enabling Scalable Virtual Organizations", Int. J. Supercomputing Apps. 2001.
http://www.globus.org/alliance/publications/papers/anatomy.pdf
[DataMgt]
http://www.globus.org/research/papers/dataMgmt.pdf
[Elephant]
Describing the Elephant: The Different Faces of IT as Service
http://acmqueue.com/modules.php?name=Content&pa=printer_friendly&pid=319&page=1
[GGF]
Global Grid Forum
http://www.ggf.org/
[GridFTP]
The Globus Striped GridFTP Framework and Server. W. Allcock, J. Bresnahan, R. Kettimuthu, M. Link, C. Dumitrescu, I. Raicu, I. Foster. Proceedings of Super Computing 2005 (SC05), November 2005.
[GT4]
Globus Toolkit Version 4: Software for Service-Oriented Systems. I. Foster. IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13, 2005.
[OGSA]
http://forge.gridforum.org/projects/ogsa-wg
[SOAP]
The fundamental message enveloping mechanism in Web services.
http://www.w3.org/TR/SOAP
[WS-Security]
The roadmap to the various security related Web services standards. See
http://www.oasis-open.org
[WSRF]
The Web Services Resource Framework
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf

5.3 Additional Information

Readers of this paper are encouraged to learn more about Grid Computing from the various journals and web sites. Public comments are also welcome in the standards processes and readers are encouraged to participate by reading[26] and making public comments. Public comments are a very important part of the standards document process. DISA can help drive the process more instead of generating proprietary formats or protocols. In general, DISA might also want to have permanent representation in various working groups.

5.3.1 Other grid standard bodies or organizations

Enterprise Grid Alliance (EGA)
http://www.gridalliance.org/en/index.asp

Distributed Management Task Force (DMTF)
http://www.dmtf.org/home

The Internet Engineering Task Force (IETF)
http://www.ietf.org/

Storage Networking Industry Association (SNIA)
http://www.snia.org/home

NFSv4 (IETF)
http://www.nfsv4.org/

CIM (DMTF)
http://www.dmtf.org/standards/cim/

Globus Publications
http://www.globus.org/alliance/publications/papers.php#Data%20Grid%20Components

GEDEN
http://grid.geden.org
R2AD's and Northrop Grumman's active involvement in open-source and standards bodies underline their commitment to open standards and the importance of their contribution evolution of computer science. They are associated with the Global Grid Forum (GGF) as an active member of the Open Grid Services Architecture (OGSA) Working Group. R2AD is helping to bridge the DoD to the future of grid computing by support the grid.genden.org site in an effort to accelerate the deployment of grid computing in command and control environments.

[26] http://www.ggf.org/ggf_docs_public.htm