DRAFT
GGF Informational Document

Reliability in Grid Computing Systems

Christopher Dabrowski and others?
(author or editor)
U.S. National Institute of Standards and Technology
cdabrowski@nist.gov
http://gridreliability.nist.gov/

Abstract: This informational document surveys the state of current work in grid system reliability and identifies major issues of concern to practitioners and researchers. The survey identifies contemporary practices for ensuring reliability in both scientific and commercial grids and describes research intended to support development of reliable future, large-scale global grids. The survey helps bring to light technical issues and research problems that need to be addressed in order to implement more reliable and robust grid systems. This provides a basis for identifying requirements for capabilities needed to ensure high levels of reliability in current and future large-scale grid systems. It also provides a basis for preliminary requirements for methods and tools to measure grid and WS system reliability. Of special interest are current practices and research that provide insight on how use of WS and grid standard specifications may affect system reliability. Specifications that may need to be evolved to better support grid reliability are identified. This document is intended to serve as a resource for grid reliability issues for researches, implementers, and specification developers.

Reliability in Grid Computing Systems

1. Introduction

In recent years, grid technology has emerged as an important tool for solving compute-intensive problems within the scientific community and industry. To further the development and adoption of this technology, researchers and practitioners from different disciplines have collaborated to produce a set of standard specifications for web and grid services. These specifications are intended for use in creating software components that will enable development and operation of large-scale, interoperable grid systems. However, if the potential of grid technology is to be realized in large-scale industrial and scientific computing environments, it also will be critically important to develop methods for ensuring the reliability of large-scale grid systems and to ensure specifications are created that support high reliability. Predictably and somewhat understandably, the investigation of reliability issues for grid systems has been less extensive thus far, than efforts to develop basic capabilities. This document addresses this gap by providing a survey of current work on grid reliability, undertaken by researchers and practitioners in academe, industry, and government[1], and identifying technical issues and problems of special concern. This current work provides a basis for identifying preliminary requirements for capabilities that need to be developed in order to establish and maintain high levels of reliability in large-scale grid systems. This document also provides preliminary requirements for metrics on grid and WS system reliability. Also of special interest are current practices and research results that provide insight on how use of WS and grid standard specifications may affect system reliability. Specifications that may need to be evolved to better support reliability are identified. Once finalized, this document will serve as a guide on reliability issues to researchers in grid reliability, implementers of grid systems, and to working groups developing specifications.

The term reliability, as used in this document, refers to the ability of a grid system to effectively provide its intended service over time. Fault tolerance refers to the ability to provide this service in the face of faults among system components and is a key strategy for improving system reliability. The survey of grid reliability provided in this document is facilitated by decomposing a grid system into four major topical areas of concern that can be treated separately.

- First, is the reliability of the hardware and software computing resources accessible through the grid. This includes grid computing resources such as processor clusters, supercomputers, storage devices, and related hardware, together with the software for managing these resources. Grid resources also include software components accessible through the grid that are engaged by users to perform various functions, such as data mining and other analysis. In data

[1] Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

grids, data stores and data sets are grid resources as well. Using the WSRF framework [Ogf2006a] as a basis, grid resources will be implemented in future grids as web services. Hence, grid resources are often viewed simply as services.

- Second are the core grid infrastructure and resource management services whose responsibility is to manage allocation of, and access to, grid resources by users and user applications. This includes a sizable collection of management services for discovery, negotiation, execution management, status notification, security, and related functions. The infrastructure and resource management services are the core of the grid function whose reliability is essential to the overall reliability of the grid. These also are likely to be implemented as web services.
- Third, there is a need to address the reliability of underlying connection and data transport facilities used by the grid, or the grid network. Discovering and securing necessary grid resources and executing grid applications require substantial numbers of individual messages. Executing grid applications may require reliable transport of large data sets among distributed application components. Therefore, grid message exchange and data transport must be reliable and operate in the face of faults in lower-level transport components.
- Fourth, reliability of a grid system should also be viewed from an overall system perspective. There are two approaches to viewing the grid from an overall perspective: the first is to consider the effect on reliability of the overall design, or architecture, of the grid. The second is to view the grid as a complex system, in which the cumulative actions of large numbers of components can lead to emergent global behaviors that might have unexpected impacts on reliability.

The four topical areas are based on the assumption that grid system reliability is provided by the grid itself. That is, ensuring the reliability of grid resources, infrastructure and management services, and the grid network is the responsibility of the providers or managers of these various resources and services. However, reliability may also be provided by the end user or user application. The existence of these two alternative sources for providing reliability raises significant issues that are addressed in this study.

This document is organized as follows. Section 2 discusses definitions of major terms and concepts that underlie this study and are used in subsequent sections. In section 3, the four major areas of grid reliability identified above are surveyed. In each area, known state-of-the-art practices and solutions are summarized, together with research into major issues and problems of concern. In section 3, the grid service provider is assumed to be source of reliability capabilities. Section 4 discusses the role of the user or application in providing reliability. Section 5 uses the discussion in the preceding two sections to propose preliminary requirements for reliability capabilities that will be needed for future large-scale grid computing systems, including such items as specification development and component testing. These requirements are not intended to be definitive, but rather to stimulate discussion on the topic. Section 6 presents initial work on reliability metrics and provides some preliminary requirements for measuring reliability. Section 7 summarizes and concludes. Section 8 lists references. The appendix provides a cross-reference list for references and the various grid reliability subject areas. Finally, it is important to note areas that are out of the scope of this document. These are security, which is an extensive subject that should be treated separately and integrity of physical sites, which is similarly studied extensively elsewhere.

2. Definitions

This section reviews key definitions and terminology used in this informational document. Avizienis and his colleagues [Aviz2004] provide an extended set of definitions and taxonomy of reliability concepts that provide a basis for this terminology. This section also identifies terms that must, of necessity, be sometimes used differently.

2.1 Definitions of Grid Services and Grid Resources

The definitions of *grid services* and *grid resources* are key in this document. In [Aviz2004], a "*correct service* is delivered when the service implements the system function," where a *system* is an entity that interacts with other entities. The "*function* of a system is "what the system is intended to do and is described by the functional specification in terms of functionality and performance." Further, [Aviz2004] goes on to state that "a *service* delivered by a system (in its role as a provider) is its behavior as it is perceived by its user(s)." The term resource, or *grid resource*, as used here in this document, corresponds to the term *system* as used in [Aviz2004]. As described above, grid resources include processors that execute code and data, together with the software for managing execution. Grid resources also include software components that are implemented as web services, using the WSRF framework [Ogf2006a], as well as data sets in data grids. It appears intuitively obvious and consistent to say that grid resources are systems that provide services to users of a grid, in the sense used in [Aviz2004]. However, this document is also based on the works of authors who may use these terms slightly differently. For instance, because of the prevalence of the term grid service and web service in the literature, this document will, on occasion, use the term grid service in place of grid resource. This is necessary when reviewing work where the term service is used, rather than resource. This is similarly necessary when discussing research work that assumes concepts such as services *encapsulating* resources. It is also desirable when discussing widely-used, but perhaps not well fleshed out concepts, such as grid infrastructure and management services or *middleware services*.

2.2 Definitions of Reliability, Availability, Dependability, and Fault Tolerance

In this document, the key term *reliability* is the "continuity of correct service" in the meaning of [Aviz2004], while *availability* is "readiness for correct service." Reliability can be therefore be thought of as the ability of a resource to provide correct service over time, while availability is the ability, or potential, to provide correct service at any particular time. Perhaps it is easiest to visualize reliability as the proportion of time a resource provides correct service, while availability is the probability or likelihood of doing so. In this study, the term reliability generally will be used in preference to *dependability*, which has a more subjective definition. In [Aviz2004], dependabilit*y* "is the ability to deliver service that can justifiably be trusted. This definition stresses the need for justification of trust. The alternate definition that provides the criterion for

deciding if the service is dependable is the dependability of a system is the ability to avoid service failures that are more frequent and more severe than is acceptable." The more concretely defined term *reliability* is preferable because it allows unreliable systems and behavior to be more readily discerned. However, the term dependability will be retained when reviewing work of researchers who also use this term.
Finally, *faults* are deemed to be causes of errors in the behavior of systems that lead to providing incorrect service and failures. The term *fault tolerance* refers to the ability for a grid resource or resources to continue to provide service and avoid failure in the presence of faults. Fault tolerance can be roughly equated to the concept of *resilience*, a term which is used less often in the literature. The term *robustness* is sometimes used to indicate fault tolerance in the face of faults that are external to a system. *Fault prevention* refers to preventing the occurrence or introduction of faults, which for software components may be accomplished through rigorous developmental methodology. *Fault removal* is the process of reducing the number of faults, which can be accomplished either through rigorous testing procedures or after detection and isolation of faults in an operational system. In this study, fault prevention, removal, and tolerance are considered to be means by which grid reliability is achieved.

For purposes of this document, any term encountered below that is not explicitly defined above will be assumed to be defined as in [Aviz2004].

3. Current Practices and Research on Grid System Reliability

This section surveys research and methods for improving reliability in the four major subject areas identified in the introduction. In some cases, reliability solutions have been developed with a view toward future large-scale grid systems in which grid resources and management services are of heterogeneous origin and may be separated by administrative an firewall barriers. These services will be implemented as web services. Moreover, grid systems will be increasingly dynamic, in the sense that resources will be constantly joining and leaving the grid, so that characteristics of a resource being used cannot always be fully known. Under these circumstances, the chances of executing typically long-running grid applications involving non-trivial workflows and many resources without encountering a fault diminishes. For this reason, real-time fault tolerance is seen as very important for ensuring reliability in grid computing. Researchers have therefore focused on methods for improving fault tolerance of grid resources, resource allocation and management services, and grid networks. Fault removal, through testing and certification, has thus far received less attention, an issue which is also addressed below.

Fault tolerance consists of fault detection and recovery. [Aviz2004]. Fault detection involves isolation and identification of a fault so that the proper recovery actions can be initiated. Generally, in computer systems, recovery is based on redundancy that can take either temporal or spatial form. Temporal redundancy involves repeated attempts to restart failed resources or services. Spatial redundancy involves leveraging multiple copies of resources. Spatial redundancy has been of most concern to grid system researchers, who have focused on two basic approaches: (1) migration of a process from failed to an operating environment, and (2) maintaining a sufficient number of replicas of a process, executing in parallel with identical state, so that one is always available to continue a process.

This section will review fault tolerant solutions to enhance reliability of grid resources (section 3.1), grid resource allocation and management services (section 3.2), and grid communications and data transport services (section 3.3). Section 3.4 will discuss methods for ensuring reliability that are based on an overall system perspective. These discussions are based on the assumption that reliability capabilities are provided from within the grid system itself. Section 4 covers reliability originating in the user application.

3.1 Fault Tolerance of Grid Resources.

This section treats the topic of current methods for fault tolerance in grid resources being developed within academe and industry. Not surprisingly, fault tolerance in grid systems are based on methods used in previous generations of high-performance (HP) computing systems. Therefore, of special interest here is new work that is specific to grids or that has been adapted for grid systems from HP computing systems and cluster computing as well as from related technologies such as web services and Internet services. Of particular importance are web services, which by virtue of the WSRF, provide the technical base for future grid computing. The section first addresses the topic detecting faults in grid resources in section 3.1.1. Section 3.1.2 addresses methods for recovering from faults. Section 3.1.3 then addresses fault removal through component testing.

3.1.1 Fault Detection

Fault detection is an important area of research for future large-scale grids. In today's grids, fault detection is often accomplished using contemporary network monitoring tools such as SNMP [Ietf2002a]. As grid systems become larger, increasingly dynamic and heterogeneous, and more distributed, fault detection systems must be developed that are scalable and efficient. The traditional mode of monitoring components with recurring heartbeat messages may not be scalable under all circumstances in a grid system. Similarly, problem diagnosis based on detailed knowledge of network structure and service operation, prevalent in today's SNMP-based systems, is not likely to scale or be effective in heterogeneous, dynamic environments. In large-scale grids, administrative boundaries and firewalls may pose problematic barriers to efficient fault detection. In addition, the fault detection systems must themselves be resilient and not be vulnerable to a single-point of failure; they must also be able to automatically reconfigure in the face of changing circumstances. Many efforts therefore focus on developing fault detection solutions for grid systems that are scalable, efficient, and work in the presence of administrative barriers. Further, these fault detectors must themselves be highly reliable. Other efforts concentrate on detecting faults that are hard to isolate, but could seriously impair grid computations. Some work also focuses on distinguishing between different types of faults in order to effect alternative recovery actions. These challenges also suggest that research is needed on scalable methods for fault isolation and diagnosis specific to grids that are not affected by dynamism or administrative barriers.

Early work on Globus systems resulted in a fault detector, which decoupled monitoring, detection, and notification functions in order to provide greater deployment flexibility and efficiency [Stel1999]. Later, Horita and his associates [Hori2005] proposed a fault detection system which leverages earlier work on group membership protocols [Gupt2001], [Das2002]. In this system, individual processes in a computing grid are monitored by a small group (4 or 5) of randomly chosen processes on remote nodes. The monitoring processes automatically establish a Transmission Control Protocol (TCP) connection to the monitored process and periodically transmit short messages (heartbeats) to check if the connection is alive, thus creating a kind of virtual monitoring network within a grid. When a monitored connection fails, notifications are propagated through the monitoring network. Experimental results demonstrate scalability within small clusters of nodes on the order of hundreds (approximating grid sites containing LANs). In [Jain2004], a failure detection protocol is presented in which grid resources are organized in heartbeat groups on the basis of physical network topology reflected in Internet addresses. Each group member is monitored by a leader node, which is made redundant for fault tolerance purposes. The total number of heartbeats required to monitor all resources is shown to scale with a computational complexity of $O(n)$, where n is the number of heartbeat groups in a grid.

Work has also begun on detection techniques that differentiate between fault types that occur in grids. The OASIS specification *WS-BaseFaults* [Oasi2004b] provides a basis for standardizing different fault types. Jitsumoto and colleagues [Jits2007] have developed a detector that differentiates between hardware faults, process faults, and transmission faults; users are allowed to pre-select a recovery procedure to be invoked in response to occurrences of particular fault types. A combined fault detection and recovery method for transient process faults is presented in [Xian2006] that is based on an adaptive checkpointing scheme. In this approach, checkpoints are taken on replicated processes executing in parallel to compare states and discover faults that produce erroneous computations; if a fault is found, both processes are rolled back to the last consistent state. Checkpoint intervals are dynamically varied and determined by the frequency of detected faults. This approach was found to reduce overall process execution time in grid workflows consisting of multiple tasks. Other researchers are working on methods to isolate faults that originate at a particular component and propagate across a grid network [Li2006]. Kola and others [Kola2005] report work on hard-to-detect faults in the Condor distributed computing system, developing a model of "silent" fault types, which are characterized by not immediately indicating their presence after occurrence. In [Duar2006], a diagnostic approach is proposed for executing real-time tests on groups of interdependent grid components in order to isolate the origin of a fault and determine recovery actions. Results of tests are reported in a Globus-based system.

Finally, work has been reported on detection of Byzantine faults in grid systems, which disrupt computations, but are not caused by events such as wholesale crashes or link failures and are not easily traceable by examining related processes, messages, or data [Mogi2006], [Wang2006]. Byzantine faults are caused, for example, when equipment periodically or randomly malfunctions due to aging, sabotage or external damage or is subjected to transient events such as electromagnetic interference. Byzantine faults are potentially dangerous to long-running, parallelized applications because hard-to-detect, recurring errors to a single component can disrupt an entire computation. This, in turn, can result in lower user confidence in a grid system.

In research on technologies related to grids, a fault detection scheme was developed for large, dynamic Internet service environments [Chen2002a] that is likely to be relevant to grid systems. Here, data clustering analysis of failed client request messages to remote services is used to diagnose faulty component services. This system does not require knowledge of network structure or services being analyzed, in contrast with current technology that uses event correlation and detailed component dependency graphs, such as [Choi1999], [Grus1998], [Yemi1996]. In [Keya2002], an approach is presented for detecting and reacting to malicious attack in Gnutella-based peer-to-peer networks.

3.1.2 Research in Recovery Methods for Grid Resources

As in other computing systems, exploitation of component redundancy is the basis for fault tolerance and recovery in both research and commercial grid systems. This section begins with a discussion of checkpoint and process migration, a well-known technique for leveraging resource redundancy to move a process from a failed to an operational computing environment where it can be resumed. The section then describes research on replicating grid resources and services, in which multiple copies of a process or service run simultaneously to perform a grid computation. In principle, failure of a primary copy allows a replica to seamlessly take its place. Finally, the section reviews some of the considerable body of work on replication of data in data grids.

3.1.2.1 Checkpoint and Recovery through Process Migration

Taking checkpoints is the process of periodically saving the state, or snapshot, of a running process to durable storage. If the process is unable to complete, it can be restarted from the point at which it was last saved, known as its *checkpoint*. Processes whose checkpoints are taken can be later restarted on the same processor, or *migrated* to a different processor. The migration of a process, that is unable to continue on its original processor to another processor, is sometimes known as *failover*. While checkpoint procedures can be used in connection with other fault tolerant methods (as for failure detection), they are most often employed with process migration. This section considers checkpoint and process migration methods that are provided by the grid system and are transparent to the application and the user. The OGF checkpoint and recovery specification *GridCPR* [Ogf2005a] also addresses this function but appears to be written from the point of view of the application. Methods that originate from the application are discussed in section 4.

Many scientific and commercial grid systems provide checkpoint and process migration techniques that involve adaptation and extension of methods used in current high-performance cluster computing systems. These grid systems consist of interconnected clusters controlled by servers. In some cases, a fault-tolerant grid infrastructure is provided where replicated server managers supervise a compute node cluster. Failure of any single manager results in transfer of its function to another manager, while failure of a compute node results in similar transfer of an ongoing process to another node. In this manner individual clusters preserve a logical structure in which a manager continues to supervise a set of compute nodes. Some commercial products include cluster computing components that can be assembled into grid systems as for example. Here, fault tolerant grid capabilities can be provided for controller and compute nodes by deploying and configuring components that monitor processes, take checkpoints, detect faults, and migrate processes. Using such components, a number of deployed grid systems have been described at vendor web sites that provide fault-tolerant capabilities.

In research grids based on cluster computing systems, early efforts at using checkpoint and process migration in grid systems were reported in [Lanf2002]. The Condor distributed processing system [Cond2007] provides site server fault tolerance by replicating servers and employing process migration when the primary fails. Earlier work on enhancing checkpoint and process migration techniques in Condor to permit dynamic relocation and re-linking of proprietary executables in foreign administrative domains is discussed in [Zand1999]. A survey of research on process migration methods in high-performance computing environments is provided in [Milo2000]. In the HA-OSCAR research system [Lean2004] [Liu2005] [Lima2005a] [Lima2005b], fault tolerance in grid site servers, or cluster head nodes is improved by taking checkpoints of job-queue information and updating a hot-standby backup server. If the primary server fails, the backup is provided with more up-to-date job-queue information, which allows faster restart of in-progress jobs. A similar fault-tolerant feature is added to the implementation of GridFTP. In [Liu2005], a scheme for taking coordinated checkpoints is employed for parallel processes managed by a server, in which failure of one process necessitates rollback of all processes. The ABARIS system [Jits2007] flexibly responds to different types of faults (process, hardware, etc.) by allowing the user to select different recovery strategies (process restart, process migration, or substitution of a replica process) to respond to occurrence of different fault types. Grid systems composed of computing clusters, such as OSCAR, ABARIS, and others, often use the Message Passing Interface (MPI) specification for parallel computing systems [Mpi2003] to enable communication between server and processes, as well as between processes. MPI provides basic error handling features, for which a number of researchers have proposed extensions [Louc1998], [Bosc2002], [Grah2002] [Batc2004], [Gabr2003], [Woo2003], [Yeom2006], and [Bout2005].

A particularly important issue for parallel processes in grids is synchronization of checkpoints among multiple parallel processes that continuously interact through message passing. As these processes send messages to each other, they also cause changes to each others internal states, which together form a larger, collective state that rapidly evolves over time. These situations require coordinated checkpoint schemes to capture a consistent view of the larger, collective state. Taking coordinated checkpoints is a difficult problem because it must resolve messages in transit at the time the checkpoint operation takes place. These in-transit messages must be accounted for to save a consistent collective state and provide a common restart point in the event of failure.

Elnozany and others survey coordinated checkpoint schemes for distributed systems in [Elno2002]. A proposal for a fault-tolerant version of MPI, FT-MPICH that employs coordinated checkpoints was developed for grid environments by Yeom and other researchers [Yeom2006] [Woo2003]. FT-MPICH takes coordinated checkpoints of interacting parallel processes, requiring processes to block (halt) during the checkpoint procedure in order to synchronizing states. An MPI extension that also employs coordinated checkpoints and blocks processes was proposed by Bouteiller and colleagues [Bout2005], while [Jits2007] implements a coordinated checkpoint scheme from [Elno2002]. In [Bunt2007] different coordinated checkpoint protocols were compared. The results of this analysis indicated that blocking processes to coordinate checkpoints reduced efficiency and required more overhead; however, protocols that did not block processes suffered from implementation issues. The requirements of scientific and commercial process workflows in future grid systems suggests that additional investigation will be needed to assess the impact of scalability and physical distance on different checkpoint coordination schemes and to examine such issues as clock synchronization across time zones.

3.1.2.2 Grid Resource Replication

Grid resource replication, as used here, assumes that redundant grid resources simultaneously perform an identical computation and have identical state. The goal of replication is to ensure at least one replica is always able to continue the computation if a failure occurs. This manner of replicating processes is sometimes known as providing a *hot standby*. This section will review work on use of replicated resources to improve fault tolerance. To date, researchers have investigated algorithms for determining optimal (or near-optimal) number and placement of replicas intended to increase fault tolerance and lessen management overhead. Also critical is the issue of synchronizing replica computations to ensure they are the same. Thus far, a comprehensive understanding is lacking of tradeoffs between the increases in fault tolerance gained through replication versus overhead necessary to manage and synchronize replicas. Similarly, there has been no comparative analysis of the combination of checkpoint and process migration resource replication, to determine when best to use either technique.

Lee and Weismann describe a dynamic service replication approach that adapts to changing user demand [Lee2001], implemented in the Legion distributed computing system [Natr2001]. An early attempt by industry researchers to compare different algorithms for placing services in dynamic, distributed systems from the standpoint of overall system fault-tolerance is [Andr2002]. In [Verm2003], the SRIRAM system is presented for automatic replication of computing resources in distributed environments. Here, resources are members of networks, or meshes, which can be searched to find nodes on which grid processes can be replicated. Search of large meshes is made more efficient through organization of its participant resources in a spanning tree structure and through intermediate caching of query results for reuse. Participants in the mesh operate securely and anonymously, allowing some possibility of operating across administrative boundaries. The approach is intended for use in grids and peer-to-peer networks to enable deployment of services with higher availability and better fault tolerance.

Within the e-Demand project, [Town2005] proposed a replication method for web-service-based grids that detects faulty computations. Here, a computation is executed by parallel multiple service replicas. The results are evaluated through a voting process to select which replica should return its result to the user. Since a single service replica may itself be composed of a workflow consisting of multiple services, this approach identifies faulty services used in more than one replica workflow and eliminates the related service replicas from the voting. Experiments using a testbed demonstrated this approach improves fault tolerance in service compositions.

[Zhan2006a] address the problem of efficiently coordinating consistent states among service replicas that behave non-deterministically in asynchronous commercial and scientific grid systems. The researchers propose an optimized version of the Paxos algorithm for synchronizing replicas in distributed environments [Lamp2001] and demonstrate the efficiency of their approach under both local and wide-area conditions. . In earlier work [Zhan2004], a more traditional primary-backup approach was used to investigate replication of Grid Services implemented with the Open Grid Services Infrastructure (OGSI) and the Globus toolkit. Here, it was found that the strategy could be readily implemented and resulted in higher service availability in local area environments; however, the overhead costs imposed by OGSI notification were significant.

Valcarenghi [Valc2005] presents a service replication approach in which replicas are located in proximity to each other to form *service islands* in a network. Different replica configurations are evaluated using a Mixed Integer Linear Programming model to determine which choice of islands exhibits higher fault tolerance. The approach is shown to enable recovery of a high percentage of long distance inter-service connections and has advantages in minimizing the number of replicas needed, thus simplifying the process of generating and synchronizing replicas. In [Lac2006], work on an experimental resource allocation system in a telecommunications grid is reported that employs dynamic process replication to provide fault tolerance and enable fulfillment of terms of service level agreements. In [Abaw2004], a method for scheduling jobs redundantly at different sites in a compute grid is presented, assuming a grid in which processors are underutilized. The method reported here does not consider cost of resource use.

In the area of web service research on replication, Hillenbrand and colleagues [Hill2005] explore dynamic binding of web service replicas to ensure selection of operational services at run time and thus achieve greater service availability in voice-over-IP environments. In [Sant2005], a system for managing replica web services is proposed as part of an overall scheme for web service fault tolerance that is based on FT-CORBA, while in [Marc2001], a CORBA service replication management system is described. Also related is work by Microsoft researchers [Qui2001] comparing alternative algorithms for placement of replicated web servers.

3.1.2.3 Replication in Data Grids

Data replication and replication management have been issues of long standing in grid systems and have also been implemented commercially. Early research on data grids predicted the benefits from data replication for performance, data availability, and fault tolerance [Cher1999], [Hosc2000], [Stoc2001]. However, many studies emphasize performance and efficiency improvements obtained through data replication rather than improved reliability. In contrast, a number of studies have sought to improve reliability of the replica management service, rather than the data itself, by decentralizing these management services and thereby making these services more fault tolerant. Hence in data replication, both redundancy of the data and the data replication services are research issues. However, here as in resource replication, no work appears to address the issue of replication across administrative or firewall boundaries.

In [Rang2001], a simulation system is described for evaluating performance, expressed in terms of response times, of different data replica placement strategies in a data grid. In [Dull2001], an approach is proposed for maintaining consistency of replicated data. In [Lame2002], a data replica management system is described that is intended to improve access efficiency, partly by using a cost function to place replicas at locations so as to minimize access and storage costs. Simulation results are provided that show improved query response times. Bell and his colleagues [Bell2002], [Bell2003] examined different data file replica placement algorithms, including an economic approach, comparing these approaches on the basis of job throughput. Another study presents a scalable approach to replication aimed at improving efficiency of large-scale data access while reducing replication overhead cost [Taki2005]. In [Lui2006], two algorithms are examined for determining number and location of data replicas in order to balance workload in data grid environments where databases are hierarchical. The algorithms are shown to be scalable with a computational complexity of $O(n \log n)$, where n is the number of databases. A few studies have focused on improving fault tolerance through data replication. A decentralized strategy for replica generation and placement in a peer-to-peer network is presented in [Rang2002]. This study provides simulation results showing the approach improves data availability. Lei and others [Lei2007] describe 3 alternative data replica placement optimization algorithms that provide improved data availability.

A number of researchers investigated decentralization of services that manage replication of data in order to improve service fault tolerance. In [Cher2002] [Cher2004], a fault tolerant, decentralized replica location service for the Globus toolkit is described, which is designed to avoid a single point of failure. Here, decentralized and redundant replica indexes maintain consistent information about data replicas and their location. These studies report testbed results that document performance and scalability of the approach. Subsequently in [Cher2005], the design and implementation of a Globus web service-based data replication service is presented and preliminary performance test results are provided using scientific datasets in wide-area environments. In related work, [Ripe2002] describes a decentralized replica location service that is also intended to achieve robustness by avoiding a single point of failure through redundant, distributed replica management services. This work adds other features to improve service operation, such as modulation of update frequency in response to network traffic. Results are provided on performance of this system, but fault tolerance characteristics are not documented. In [Deri2004], a quorum-based protocol is described for maintaining replicated data in distributed environments, such as data grids, that is designed to enhance fault tolerance and data availability. Here, it is shown that data retrieval can succeed when as many as 75% of replicas have failed. More recently, Zhang and colleagues [Zhan2006b] propose an algorithm for dynamically locating data replica servers within a grid in order to optimize performance, and improve fault tolerance of grid data replication services.

3.1.3 Fault Removal through Testing and Code Certification

Testing of components to find and remove potential future faults is a traditional and proven method of fault removal. Components that have passed tests can be certified as having achieved a level of reliability or dependability. While testing is essential for achieving high levels of reliability, it has been observed in grid and distributed environments that even extensively tested components can produce failures when interacting in complicated workflows over extended time intervals. For this reason perhaps, methods for testing and certifying grid components receive less attention in the grid research community. Nevertheless, software testing to avoid faults is a precondition to, and basis for, fault tolerance. It is anecdotal that recovery cannot succeed if all replicas of a software component contain a fatal error. There is ample evidence of the economic cost of having inadequate methods and tools in place to test of software components prior to operational deployment [Demm1989], including distributed systems for commercial use [Rti2002].

For this reason, methods and tools are needed that are geared for measuring quality and discovering defects in software components that undergo interactions that are typical for grid systems. To date, there have been initial efforts in this area. Looker and colleagues [Look2004a], [Look2004b], [Look2005] report preliminary work on use of fault-injection to identify malfunctioning SOAP-based web services components. The approach is intended to analyze system designs and also to certify code. In [Look2007], an ontology-based approach is used to generate fault injection test cases. *In [Khar2004], preliminary work is reported on methods for assessing dependability of web service compositions, focusing on impact of upgrading individual components of commercial off-the-shelf (COTS) web service products. Song and colleagues* [Song2007], [Topk2006] *analyze GridSphere systems by creating component dependency graphs to identify crucial "hub" components, through which a large portion of system messages flow. If "hub" components contain faults, they are more likely to adversely impact overall system operation. Therefore, once identified, "hub" components can become the focus of testing activity. The approach is intended to be generalized for analysis of web-based COTS products and systems. The efforts described here represent a start toward developing testing technology for grids. Perhaps an important step toward developing methods and tools for systematic testing is to first obtain a better understanding of cost-benefits of testing grid components. Such a study could be used to determine which grid functions most require testing (some may have been subjected to extensive testing elsewhere) and what kind of tests would be most cost effective (component tests, integration tests, interaction tests, etc.).*

3.2 Supporting Grid Infrastructure and Resource Management.

A grid system requires infrastructure and management middleware services in order to function as a coherent entity in which users and service providers can interact and complete tasks. Example infrastructure and management services include service discovery (through directories or other service discovery facilities), scheduling and co-allocation of grid services remote from each other, interface services to facilitate job submission and monitoring, remote monitoring and notification services, services for high-speed transfer of data and files (e.g., Grid FTP), usage and accounting services, and security (authentication, authorization, encryption, etc.). Reliability of infrastructure and management services can be improved by the same techniques for fault tolerance and fault removal through testing, which are described above for grid resources in general. However, in contrast to grid resources, infrastructure and management services have a wider scope and more central function. For the grid to operate, the reliability of infrastructure and management services is critical. This is why work within the research community has been focused on methods specifically geared toward this essential class of services.

Work on fault tolerant data replica management services [Cher2002] [Cher2004], [Cher2005], [Ripe2002], [Deri2004] has been described above. Early work on increasing survivability of secure communications services in distributed environments through use of redundancy was reported in [Hilt2001]. This work describes a variety of redundancy techniques for making security services resistant to attack, which can be used in grid environments. In [Juha2003], a fault-tolerant service discovery system is described that is built on top of the Jini Service Discovery protocol [Arno1999]. This approach exploits the inherent redundancy of Jini lookup services to build a distributed, hierarchical index of grid resources in which nodes in the hierarchy are replicated and geographically distributed. Experimental results from a testbed are provided to document system performance; however tests of fault tolerance capabilities are not reported.

The ability to reschedule jobs in the face of failure is potentially an important infrastructure and management service function. Huedo and colleagues describe a fault-tolerant scheduling service for Globus environments used to dynamically reschedule failed jobs [Hued2006]. An important job scheduling function for forming workflows is co-allocation, or co-scheduling, of grid services so that they can be simultaneously available. Different methods of co-allocating resources in grid environments were studied early on in [Czaj1999], [Anan2003] and most recently in [Kuo2005], [Macl2006], [Wald2006], and [Yosh2005]. [Czaj1999] and [Anan2003] discuss mechanisms to allow users to react to, and overcome, failures of the services being aggregated. Fault tolerance of the co-allocation service itself is addressed in [Macl2006]. The co-allocation service is designed to leverage the fault-tolerant properties of Paxos commit algorithm for distributed transactions [Gray2004], which is used here to coordinate scheduling of grid resources that need to be simultaneously available.

3.3 Grid Connection and Transport Reliability

This section identifies work, originating from within the research community, on improving fault tolerance in networks underlying grid systems. Also, this section identifies critical issues in achieving high levels of availability and reliability of network resources that are necessary for grid systems. The OGF informational document [Ogf2004a] sets forth requirements for network transport in grid systems. Among the most important is reliable, rapid transport of bulk data (over 1Gb/s per flow) over dynamically allocated, secure connections. Because grid applications are long running and require data transport capabilities for extended durations, these connections must be reliable and stable for long periods. Another key requirement is multicast transmission of large data sets for processing by remote computing resources that operate in parallel. Here again, connections must be maintained for extended periods and delivery of data must be reliably ensured.

Reliable connectivity and multicast transmission in turn requires highly available and reliable grid networks. In realization of this, a number of specifications for reliable transport have been published. However, maintaining long-term connections and reliable transmission will pose greater challenges in future large-scale global grids. Here greater distances will require more resources to create connections, which correspondingly increase likelihood of failed links that necessitate rerouting. Unfortunately, currently available routing protocols are thought to be inadequate for rerouting failed connections in grid applications [Ogf2004a] [Valca2005]. For these reasons, fault tolerance in networks for large-scale global grids is an important research problem that must be solved to enable implementations to realize reliable transport specifications. The section first considers existing standard protocol specifications for reliable unicast point-to-point connection and data transport that are relevant to grid environments. Current work on evaluating these protocols from the standpoint of reliability is discussed together with implementations that realize and strengthen protocol features related to fault tolerance. Then, the section discusses research on methods for ensuring reliable connectivity and data transport, focusing on the use of overlay, or virtual, networks for grid systems. Finally, current research on reliable multicast transmission in grid environments is addressed.

3.3.1 Specifications for Reliable Connection and Transport

This section discusses specifications developed for reliable point-to-point unicast connection and transmission of data in grid environments. To date, this includes three specifications: the GridFTP specification for bulk data transfer and two web service specifications for reliable point-to-point connection and message exchange by web services, which is also intended for use in grids. A fourth specification, the Simple Object Access Protocol (SOAP), underlies web service reliable messaging specifications, and is also briefly discussed. The section identifies work that evaluates and strengthens these specifications and related implementations from the standpoint of their reliability. Section 3.3.3 addresses the topic of reliable multicast for grid environments. As with grid resources and infrastructure and management services, reliability is often achieved by providing underlying mechanisms to improve fault tolerance. Sometimes these mechanisms were not designed with grid systems in mind and therefore need to be covered here, for instance the significant body of work on fault tolerant versions of TCP such as [Alvi2001] and others.

The GridFTP, version 2 [Ogf2005b] was developed by the Globus alliance and OGF to extend the File Transfer Protocol (FTP) [Ietf1985] to permit point-to-point transfer of larger, "bulk" data over a wide area network. GridFTP, like FTP, is based on the TCP. GridFTP is designed to transfer files by taking advantage of "long fat" communication channels to create multiple data streams to significantly improve aggregate throughput of large files. GridFTP assumes fault-tolerance mechanisms provided by the underlying TCP. In addition, the basic GridFTP specification employs a checksum technique to determine if data was lost during the transfer. The Globus implementation of GridFTP has been extensively used by the scientific community. However, a known problem is that failure of a GridFTP client necessitates a complete restart of a data transmission, a disadvantage for transfer of large data sets. This is overcome by solutions provided in [Lim2004] and the Globus toolkit. In [Lim2004], a recovery mechanism consisting of redundant, intermediate brokers operates on behalf of the GridFTP server; a broker stores and forwards subsets of data streams originating from the client. In this way, a client that fails and recovers can resume transmitting data from an intermediate point rather than having to restart the data stream. Broker redundancy ensures fault tolerance in the event of individual broker failure. The Globus toolkit [Glob2005] also provides a reliable transfer service with transparent, fault-tolerant transfer of data using GridFTP. The service uses an intermediate distributed DBMS to track data movement that can be resumed in the event of failure. In [Matt2006] GridFTP is compared against other bulk data transfer protocols with respect to reliability and other characteristics.

Basic requirements for reliable messaging between web services have been set forth in two specifications. The *Web Services Reliable Messaging Protocol* (*WS-ReliableMessaging*) [Wsrm2005], [Oasi2006b] was originally developed by a group of vendors to define a protocol for guaranteed message delivery. The protocol specifies procedures for connection establishment, message exchange, and connection termination between web services. WS Reliable Messaging also specifies requirements for tracking the status of messages sent between services, guaranteed message ordering, elimination of duplicate messages—to allow guaranteed *at most once* delivery. The OASIS *WS-Reliability* specification [Oasi2004c] provides similar capabilities. Both specifications prescribe a binding that allows its messages to be encoded and transmitted using the XML-based SOAP protocol [W3c2007]. Both specifications are extensible via WSDL [Wsdl2001], to allow them to be composed with other web service specifications to defined new services. The SOAP protocol is evaluated in [Fang2007], who propose a fault-tolerant version of SOAP (FT-SOAP), with features to enable web service replication, taking checkpoints, fault detection, and recovery.

In [Pall2005], a comparison of the two web service reliable messaging specifications concludes that *WS-ReliableMessaging* provides more flexible features for re-initiating erroneous transmissions and also provides more extensive capabilities for reporting faults that occur during transmission. A comparison reported in [Dura2005] found in some cases that the two specifications were geared toward responding to different types of failures. These comparative studies serve to illustrate that subtle differences in assumptions about the kinds of faults that will be encountered may affect error handling in implementations. Initial efforts in implementing *WS-ReliableMessaging* such as [Tai2004], [Pall2005] and *WS-Reliability* have not yet provided information on their effectiveness in grid environments. A comprehensive analysis based on actual operational experience is needed to evaluate reliability aspects of these specifications for web-based grid applications and to bring to light specific issues that need attention. A performance analysis of *WS-ReliableMessaging* appears in [Pall2005]. Finally, subsequent OASIS standardization of *WS-ReliableMessaging* [Oasi2006b] seems to indicate growing use of this specification.

3.3.2 Research in Fault Tolerant Grid Networks

The specifications described in the previous sections provide requirements for reliable connectivity and transport of data in grid environments. For implementations to achieve reliable behavior called for in these specifications requires effective mechanisms for fault detection and recovery within the network. To date, research on grid networks has looked toward use of overlay, or virtual, networks dedicated to grid systems as the best solution for providing fault tolerance. Here, as with grid resources, the focus has been on recovery techniques based on rerouting through redundant network resources reserved for grid use. There appears to have been less work on fault detection.

In [Fox2006], [Fox2005], a messaging infrastructure is presented for support of communication and large-scale data transfer in web service based grid systems. The infrastructure employs redundant distributed intermediate brokers to form a virtual software overlay network, the *NaradaBrokering* system, for managing large data streams. The infrastructure supports multiple protocols (including UDP, TCP, and parallel TCP) and web service communication through its support of SOAP, WS-Eventing, and WS-Addressing. The infrastructure implements reliable messaging through its support of *WS-Reliable Messaging*, and *WS-Reliability* to facilitate ordered and guaranteed at-most once delivery of messages and events. The underlying system of redundant brokers and links is intended to exhibit fault tolerance and guarantee delivery of messages in the face of broker and link failure, failure or disconnect of communicating services, link failure, and failure of storage devices. The viability of this approach is demonstrated through prototype grid applications involving streaming audio and video data and implementation of the Grid-FTP recovery mechanism described above [Lim2004].

The OGF informational document [Ogf2004a] identifies efficient routing as a key to achieving high availability in networks that serve grid systems. An important aspect of routing is traffic engineering [Ietf2002b], the selection of paths through a network to maximize data flow within available bandwidth without violating administrative constraints. Effective routing mechanisms are important to dynamically reroute grid data flows around failed network components. The identification of routing mechanisms which will perform well in grid environments is an important topic of research, since current Interior Gateway Protocols (IGPs), such as OSPF/IS-IS [Ietf2007a] and BGP [Ietf1995], are believed to be insufficient [Ogf2004a]. One solution involves creation of virtual overlay networks on top of existing physical networks using Multi-Protocol Label Switching (MPLS) [Ietf2001], [Ietf2002c] to provide better bandwidth availability and predictable performance. In related work, design of a dynamic grid networking layer that provides automatic bandwidth on demand is described by Clapp and colleagues in [Clap2004]. Whether these solutions improve network reliability is not known.

Given the research on overlay networks to promote fault tolerance at two different logical levels of the network represented by such systems as NaradaBrokering [Fox2006] (at a high level) and overlay networks (at a low level), an interesting question to consider is whether a combined solution is possible that leverages more than one. For instance, would mapping a software overlay represented by a broker network over lower-level virtual paths belonging to an overlay network lead to higher levels of availability in a grid network? Similarly, would fault tolerance be enhanced by mapping long-distance connections between the service islands as described in [Valc2005] onto an overlay network? These questions may be future topics of research. More generally, additional work on overlay networks is needed to determine how best to deploy these solutions for grid environments. It is important to know how management of overlay networks might differ for grid applications, particularly with respect to key functions relating to fault tolerance

The idea of grid overlay networks brings up the important issue of the degree of control by the overlays of the physical network resources that the overlays use and depend on. Because of the dependence of overlays on physical network resources, the reliability of the overlay also depends on the reliability of the physical resources. For this reason, overlays may wish to request allocation of physical resources or be notified of the resource failure in order to take appropriate recovery actions. Thus, from a reliability standpoint, understanding and controlling interactions between overlays and the supporting physical network becomes an area of needed research, especially if multiple overlay technologies are employed together as suggested above. Finally for use in grids, it is necessary to investigate allocation of dedicated network resources, rather than shared resources. Given the heavy demands for data transport in grid systems, ultimately it may not be possible to fulfill grid networking requirements without dedicated network resources.

3.3.3 Reliable Multicasting

Multicast transmission in grid environments is necessary point-to-multipoint dissemination of data across a grid. For instance, scientific grid systems may require transmission of instrument or simulation data originating at one site to multiple, remote storage sites. Perhaps the most popular use of multicast is the Access Grid [Acce2007], where large audio and video datasets are broadcast regularly to large groups of participants. However, here and in other cases such as [Fox2005], best effort multicast is used, which provides high-throughput and low end-to-end delay, but does not provide guaranteed delivery [Neko2005].

Reliable multicast protocols have been the subject of research for years, both within grid settings and for more general purpose use. The Nack-oriented reliable multicast (NORM) is currently being developed as an IETF standard [Ietf2007b]. In [Neko2005] [Barc2005], a series of trials were conduced in a grid testbed that compared performance of NORM, the Multicast Dissemination Protocol (MDP) reliable multicast protocol [Ietf1999], and a variant of TCP extended for multicasting. The results showed that all three protocols exhibited unsatisfactory performance in grid environments. To date, no known studies have been conducted of the reliability levels achieved by implementations of NORM or of other multicast protocols.

Other researchers [Wate2004] [Bane2002] have proposed use of multicast overlay networks for grid applications, using clustering algorithms to organize multicast groups into multi-level hierarchical trees for efficient transmission. While these solutions provide good performance for moderately-sized multicast groups in wide area networks, they do not provide guaranteed reliable delivery. In [Jo2005], a diagnostic tool is described for analyzing multicast transport problems in the Access Grid. In [Rena2006], progress is reported in using the TRAM (*Tree-based Reliable Multicast)* protocol [Chiu1998] in a hierarchically organized compute grid designed for task farming. A survey of current reliable multicast technology [Pope2007] reviews alternative solutions and research problems under investigation. This report and the current status of NORM appears to indicate that reliable multicast transmission technology is essentially still a work in progress, with no off-the-shelf solution yet available for grid computing environments that provides both reliability and performance at large scale.

3.4 Reliability Concerns from an Overall System Perspective

This section discusses approaches to grid reliability that consider a grid system as a whole, rather than focusing on individual grid resources or collections of resources. There are two classes of such approaches. The first considers the grid from an architectural standpoint. A grid architecture can be viewed as a high-level design or blueprint of a grid system consisting of sites and their interconnections. An architectural approach seeks to analyze this high-level design to determine how to improve the overall reliability of the grid, as for instance, by identifying architectural alternatives that are more fault tolerant. The second approach involves viewing the grid as a complex system, in which the individual behaviors of large numbers of components may collectively produce a global behavior that could not have been easily predicted by understanding the behavior of the components. If the resulting global behavior results in degradation of the overall performance of the grid system, this obviously represents a fault which reduces overall reliability. For this reason, the study of grids as a complex system is important for improving grid reliability.

3.4.1 Grid Reliability from an Architectural Perspective

In this document, the term *grid system architecture* refers to the structure of a grid system, which is composed of various interconnected sites consisting of nodes[2] on which grid resources reside. Architectures may be differentiated by different topologies of sites. For instance, one can contrast a hierarchical architecture in which sites are organized in a logical tree with a decentralized architecture in which interconnections between sites assume no particular pattern. Architectures can also be distinguished by choice of geographic location and physical distance between sites. They may also be distinguished by the distribution of nodes across sites: e.g., a few sites with many nodes vs. many sites, each with few nodes. In addition, architectures can be differentiated by the number and location of key management and infrastructure services, an aspect that is closely related to the issue of service replication, discussed above. The organization of these services and their relationship to each other, in part, determines the paths for messages required to manage the grid system, which is also an aspect of architecture that influences reliability.

From the perspective of grid systems reliability, an important and necessary question to ask is whether differences in architecture impact system fault tolerance. Reliability may be improved by identifying topological alternatives or that are more fault tolerant or by determining locations of infrastructure and management services that are more fault tolerant. Reliability may also be improved by determining which components should be prioritized for testing, as in [Song2007], [Topk2006]. At present, this appears to be an insufficiently investigated question, though preliminary inquiries along these lines also can be found in [Bezz2006], [Fox2006]. A reliability analysis of a grid system that is partly based on an architectural model is presented in [Xie2004]; this is described below in section 5 on reliability measurements. The OGF *Configuration Description, Deployment, and Lifecycle Management (CDDLM)* [Ogf2006b] document describes an architecture for configuration and deployment of grid resources that makes provisions for fault-tolerant behavior by resources.

3.4.2 Grid Reliability from the Complex Systems Perspective

Complex systems are large collections of interconnected components whose interactions lead to emergent global behaviors that are not necessarily predicable from component behaviors. Because of their scale both in terms of number of components and number of interactions, the development of tractable methods to understand causes of emergent global behavior presents a challenge. From the standpoint of grid reliability, the study of grid systems as complex systems seeks to develop analytical methods that reveal global states a grid may enter into in which performance is impaired to the degree that

[2] This concept is distinguishable from a reference model architecture, such as OGSA [Ogf2006c].

constitutes a system-wide fault state. Understanding causes of emergent behavior provides a basis for developing decentralized methods of control, which when implemented by components across a grid, lead to desirable global fault-free states.

Work toward developing simulation tools to study dynamics of grid systems is reported in [Liu2004]. Other work demonstrates the importance of viewing a grid as a complex system using simulation studies. In [Mill2006], it is shown that when randomly subjected to malicious spoofing designed to interfere with resource allocation actions on a global scale, a plausible response intended to isolate spoofed service providers can actually lead to further degradation of global system performance. Subsequent work in [Mill2007] demonstrates feasibility of using current web service and grid specifications to allocate resources on basis of supply and demand in a grid compute economy. Results show decentralized economic methods yield good performance even when service providers are overloaded or subject to random failure.

By studying grids as complex systems and developing methods to understand causes of emergent behavior grids, future administrators will be able to promote global reliability of present and future large, decentralized grids.

4. Reliability of Grid Applications

Reliability can also be considered from the user application perspective—that is, the reliability of the application itself. Grid applications are dynamically enabled by first using discovery services to find appropriate grid resources and then using resource allocation services to schedule execution of application processes and data operations. As applications execute, application data moves to and from assigned resources over the network, often dedicated for grid use. Remote job management functions are used to coordinate execution activity and return results to the user. From the application's point of view, there often are no guarantees as to the reliability of resources which have been allocated for its use. Similarly, the application may itself provide code or data that causes faults. Therefore assuming a worst case scenario, application developers have often taken steps to ensure their applications achieve some degree of reliability by designing their applications to have fault tolerant features.

This section considers efforts undertaken to make grid applications more fault tolerant. The section first considers fault tolerance for individual processes that belong to the application. Here, the focus of efforts within the OGF have centered on the grid checkpoint and recovery specification [Ogf2005a]. The section then focuses on the topic of fault-tolerance in resource compositions created to execute grid workflows using web service technology. Workflows effectively involve many processes executed by a collection of remote, cooperating grid resources implemented as web services. The overall computation must exhibit resilience in face of faults in, and failures of, single or multiple resources organized in the workflow.

4.1 Fault Tolerance of Remote Application Processes

The OGF Grid Checkpoint and Recovery Service Working Group is specifying requirements for a service that can checkpoint individual processes and transfer checkpointed data to new platforms [Ogf2005a] where the process can be restarted. This service includes a function for notification of critical events, including failure and job resubmission. The specification is currently under development. The proposed service excludes checkpoint operations that involve more than one process. The OGF document *Use-Cases and Requirements for Grid Checkpoint and Recovery* [Ogf2007a] provides a set of requirements for an API to the proposed checkpoint service that assumes the service is being accessed and controlled by an end-user grid application. It appears therefore possible that an application organized as a workflow with multiple processes might invoke the checkpoint and recovery service separately for each process as an action in a workflow. Once work on this specification is completed, it is reasonable to expect that the checkpoint and recovery service might be used in this fashion.

4.2 Fault Tolerance of Grid Resource Compositions and Workflows

In recent years, there has been a significant amount of research devoted to developing methods of workflow composition and management in web-service based grid environments. Most of this work appears to be intended to develop basic capabilities for defining workflows. Only one known survey of research grid workflow management tools [Yu2005] reports fault-management characteristics of these systems. In addition, there has been some research on using generic web-services workflow languages and tools in grid environments. This section surveys web service composition and workflow methods developed specifically for grid systems as well as methods developed for generic web service environments that have been used for grids.

4.2.1 Fault Tolerance of workflows composed with languages and tools for grid environments

The fault tolerance of grid resource (or service) compositions and related workflows can be considered from two perspectives: (1) the fault-tolerance of individual grid resources participating in the composition; and (2) fault-tolerance at the level of the composition or workflow, as a whole. These are referred to in [Hwan2003] and [Yu2005] as *task-level* and *workflow-level* perspectives, respectively. Regarding the task-level perspective (1), methods for taking checkpoints, restart, resource replication, and process migration can be used to mask faults of individual resources so that they do not effect the larger computation defined by the workflow. These methods were discussed above. In regards to the workflow-level perspective, an important question is how to respond to the real-time failure of a computation in the workflow so as to minimize impact on the entire computation, defined by the workflow.

For workflow-level failure (2), [Hwan2003] identifies recovery methods defined for, and initiated from, the workflow. These include conditional rerunning of failed tasks on alternative resources that may be slower but more reliable, techniques that exploit redundancy, and user-defined techniques. [Yu2005] identifies workflow management tools for grid systems that support workflow level recovery mechanisms of this type, including [Tann2002], [Deel2003], [Hwan2003], [Yu2004], [Fahr2005], and [Alti2005]. [Ra2005] also appears to provide workflow-level recovery. Other service composition and workflow languages and tools have been developed specifically for scientific grid environments, such as [Kris2002], [vonL2004], [Baus2003], and [Schn2006] that do not appear to provide workflow-level fault handling provisions. Other work relevant to grid workflows includes the adaptive checkpointing and recovery scheme of [Xian2006], as described above.

To date, no standard specification on workflow definition and management has been produced that is specifically designed for grid environments. The OGF has chartered a research group on this topic.

4.2.2 Fault Tolerance of grid workflows composed with languages and tools for generic web services environments

Standard specification languages have been produced for defining and managing generic web service workflows that have been used in grid systems with some success. These languages do not specifically take into consideration fault-tolerance in grid environments. The OASIS BPEL4WS [Oasi2006a] specification language provides extensive workflow-level mechanisms for fault handling within web service compositions and workflows, using traditional throw and catch semantics to react to faults. Several researchers have demonstrated the feasibility of using this specification to compose grid resources [Tart2003], [Emme2005], [Cybo2006], [Leau2006], [Turn2007], citing as an advantage the reusability of workflow definitions. However, in [Tart2003], it is observed that in cases where members of BPEL4WS compositions compute a result concurrently, as is often the case in a grid system, failure of one member requires termination of all,[CED1] requiring restart of the computation. In response [Tart2003] propose mechanisms that, in certain cases, would permit a concurrent aggregated computation to continue. An XML-based specification language is provided to allow specification of such actions as part of the web service definition.

Other web service specifications for coordination of distributed web services address fault-tolerance in a general, web service context. The OASIS standard WS-Coordination [Oasi2007] specifies handling of a limited number of fault types, not specific to grid systems. The OASIS Business Transaction Protocol [Oasi2004a] and WS-Transaction specification [Cox2002], developed by a group of vendors for web service business applications, also do not address faults in grid environments. At this time, there is no known work on fault tolerant aspects of using WS-Coordination and WS-Transaction in grid environments. Works such as [Koeh2003], which compare different approaches to forming web service workflows and identify open problems in service composition, do not treat reliability issues extensively.

4.3 Merging the Application with Grid Resource Fault Tolerance Strategies

The preceding discussions have contrasted two different approaches to ensuring fault tolerance in grid systems. The first approach is to ensure that grid resources, grid management services, and network resources serving the grid are themselves fault tolerant. Section 3 discusses methods and research related to this approach. The second is to attempt to ensure fault tolerance in grid applications, discussed in this section. Grid system designers, users, and providers need to consider the relationship between the two approaches. Determining when to use either approach is necessary to prevent unnecessary redundancy. Applications and application workflows may not need to take separate fault-tolerant actions if the grid resources they use already provide adequate fault-tolerant capabilities. For instance, a grid workflow need not prescribe recovery actions for a set of nested parallel processes if the grid resources hosting these processes take coordinated checkpoints. Yet when grid resources are known to be unreliable, as is often the case, it would be prudent for applications to attempt to ensure fault tolerance themselves. For this, applications will have to obtain cooperation of providers for checkpoint and failover operations. One can envision users and grid resource providers negotiating how fault tolerance is to be provided as part of creating a service level agreement [Ogf2007b]. To enable this coordination will require standardized conventions for describing and negotiating fault tolerance capabilities that can provide a basis for automating this process in future systems. The development of standardized conventions for this purpose is therefore a topic for future research.

5. Reliability Issues and Preliminary Requirements.

This section summarizes requirements for capabilities that need to be developed to ensure reliability in current and future grid systems and identifies needs for evolving current standard specifications to support reliability and fault tolerance. These requirements identified here are based on problems being addressed by the work of researchers and practitioners discussed above as well as issues that are raised and implied by this work.

5.1 Fault Removal through Development of Methods and Tools for Testing

Fault removal is a necessary prerequisite to fault tolerance. As discussed in section 3.1.4, there is a strong argument for the economic benefits of testing. As first step, an understanding is needed of cost-benefits of testing grid components to determine which grid functions and what kind of tests (component tests, integration tests, interaction tests, etc.) would be most cost effective. Areas of special consideration should be testing of grid infrastructure and management services and testing of grid workflow compositions. To date, there has been little work done on developing testing methods and tools for grid systems.

5.2 Fault Detection for Grid Resources

Over the longer term, large-scale global grids will require fault detection systems that are scalable with respect to number of resources to be monitored and the physical distances in wide area networks. Fault detection systems must assume that grids are dynamic and must operate in the face of administrative boundaries and firewalls. Further, grid fault detection systems must themselves be fault tolerant and not be subject to a single point of failure. The authors of [Mogi2006], [Wang2006] and others have argued that increasing scale of grids will result in greater frequency of hard-to-diagnose faults, such as Byzantine faults or faults that propagate across grid components. Additional research will be necessary to categorize the relevant fault types and to understand their impact on the grid and how best to effect recovery. The potential difficulty presented by occurrence of faults in grid systems argues for research on methods for fault isolation and diagnosis specific to grids. In current systems, isolation and diagnosis relies on static topological models of a network that are likely to be inadequate in the face of the scale and dynamism of future grid systems. It is also notable, that there has been little work thus far on the issue of detecting and diagnosing external attacks and malicious activity in grids.

5.3 Recovery Methods for Grid Resources

In the area of recovery methods, general guidelines are needed for selecting different recovery methods under different circumstances and in response to different fault types; as for instance, when to use checkpoint and process migration rather than replicated resources on hot standby that compute in parallel. Individual recovery methods need research. Future grid systems will require efficient methods for taking coordinated checkpoints of interacting, parallel processes that are also scalable and robust in the face of clock synchronization and time zone issues. Another issue is finding methods for determining the optimal number and location of resource replicas to maximize overall system benefits. To do this requires better understanding of the tradeoff between improved fault tolerance versus overhead in managing replicas. Closely related is the question of developing efficient methods for correctly synchronizing replicas. Dynamism of the grid and the ability to cross administrative boundaries and firewalls must also be considered in replica coordination. Additional work is required on how to best to implement replica coordination solutions using web service specifications. Finally, the MPI specification for communicating parallel processes [Mpi2003] needs to be examined to determine (1) if this specification is to be the basis for future grid systems; and (2) if so, to decide what extensions are needed to enable respective fault tolerance operations.

In the area of data replication in grids, most work has considered how to improve system performance rather than fault tolerance. Therefore, as in resource replication, it may be desirable to have guidelines on selecting the number and location of replicas to maximize tradeoff between improvement in fault tolerance and additional overhead replica management. Here, issues of scalability, extension of replication activities over administrative boundaries and firewalls, and system dynamism must also be considered.

5.4 Special Requirements for Infrastructure and Resource Management Services

Since infrastructure and management services are implemented in the manner and other services, the requirements listed in section 5.1 for grid resources apply here. One important difference, however, is the criticality of infrastructure and management services to the grid. Ensuring their reliability has a higher priority. Hence, tradeoffs involving level of fault tolerance versus overhead costs will be different. A system should be willing to tolerate greater cost in employing mechanisms such as process replication or taking coordinated checkpoints to ensure reliability of grid infrastructure and management services. For instance, a grid system should be willing to incur the additional overhead of providing replicated hot standbys for key inter-side schedulers or authentication services. For this reason, separate studies of tradeoffs between fault tolerance and overhead costs are needed for this special class of services. Similarly, it would be desirable to further specialize fault-tolerance techniques considered for grid resources (section 3.1) for grid infrastructure and management services. Finally, developing test and certification methods should have a higher priority for infrastructure and management services.

5.5 Grid Networking and Data Transport

In principle, networking resources also have requirements for fault detection and recovery, as do other kinds of resources. The use of overlay techniques to link network resources into virtual grid networks raises issues related to techniques for fault tolerance that are specific to virtual networks. Further efforts will be necessary to identify and develop methods for fault detection and isolation, troubleshooting, and dynamic rerouting that meet the requirements for availability and transport of large data sets identified in large-scale global grids [Ogf2004a]. Another important question is whether it is advantageous and possible to combine overlay systems designed to operate at different logical network levels to achieve higher overall levels of fault tolerance. Research is needed into methods for control by grid overlays of the physical network components the overlays depend on. Similarly, the use of dedicated physical network components for grid networks needs to be examined. These issues are discussed at greater length in section 3.3.2. Finally, current standard network specifications that impact grid computing need to be examined to determine if extensions are needed to enhance reliability. One possibility is making permanent fault-tolerant extensions to GridFTP by specifying an API for a service that provides additional fault tolerant capabilities or directly extending the basic specification. Another important area is determining requirements for reliable multicasting and completing a standard specification that provides needed capabilities.

5.6 Fault Tolerance Requirements from the application perspective

Application fault tolerance in grid systems is both potentially complimentary to fault tolerance provided from within grid systems as well as unnecessarily redundant. Section 4.3 discussed the need to coordinate fault tolerance actions originating from applications with actions originating from providers of grid resources. Coordination between application and provider requires development of standardized conventions for describing and negotiating fault tolerance. A second area of requirements involves creation or extension of standard workflow definition and control languages that are specific to grid systems that provide fault tolerance for nested parallel processes. Finally, there is a need to develop generic test methods specifically for workflow compositions, focusing on identifying faults that are likely to arise when components interact.

6. Current Work on Reliability Metrics and Preliminary Measurement Requirements.

Given the preliminary nature of work on reliability of grid computing systems, it is not surprising that work on identifying specific requirements for grid reliability metrics is also just beginning. A grid system presents a myriad of potential measurements that one can make which arguably impact reliability. This section discusses initial work on grid reliability metrics and identifies some preliminary requirements for grid system metrics.

6.1 Work on Grid Reliability Metrics

Early attempts at defining reliability metrics for grid networks were made by [Lowe2003] and the OGF network measurement group [Ogf2004b] whose work led to a proposed standard for representing network entities and measurements of their properties. Aside from this work, there have been several efforts to develop metrics on grid systems. In [Chun2004], performance and robustness of a Globus-based grid system was tested by measuring responses to a patterned series of query and file-transfer operations, or probes. Failed or delayed responses to probes are used as basis for quantitative estimates of robustness and stability of a grid. More recently, [Coll2007] described a framework for evaluating quality-of-service (QoS) provided by grid systems that is based partly on use of service-level agreements. To gauge QoS, this work relied on traditional network performance-based metrics such as network delay, response time, and throughput, but also introduced service-level metrics for service accessibility and availability. In [Lei2007], data replica optimization algorithms are evaluated using data availability measures, expressed as ratio of files unavailable to files requested and the ratio of bytes unavailable to bytes requested.

A comprehensive analysis method for measuring the reliability of a grid system in the face of known component failure rates is presented in [Xie2004]. Here, the reliability of a grid system is represented by the probability that a set of grid applications, or programs executing on the grid, will complete. This method assumes a grid system model that consists of a physical grid network comprised of nodes connected with links, software programs running on nodes, a set of grid resources associated with each node, and a resource management system which maps user resource requests to existing grid resources. The method further assumes knowledge of failure rates of each these components and communication delays. Distributed grid programs, or workflows, are represented by a set of alternative minimal resource spanning trees (MRSTs), where each MRST is composed of a combination of nodes, links, and processes running on nodes. For a workflow to succeed, only one MRST associated with the workflow must succeed. Given an instantiation of the grid system model as a set of nodes, links, and resources that have known failure rates together with a set of assigned grid workflows represented by alternative MRSTs that employ a subset of the model, Xie derives an equation for a total, or comprehensive, grid system reliability—expressed as the probability that all grid programs will complete. While this approach certainly provides a useful measurement, it requires firm knowledge of failure rates of all system components. Further work on developing reliability models such as this would seem to be desirable.

6.2 Preliminary Requirements for Metrics

Clearly, more work is needed on metrics to measure aspects of grid systems other than network properties. However, based on what efforts have been made, three initial classes of metrics appear to be of interest to grid researchers, which may form the basis for initial requirements.

- The first are grid network metrics covered in [Ogf2004b].
- The second, partly based on the very preliminary body of work represented by [Chun2004] and [Coll2007], is the need for formalizing operational definitions of metrics to measure both availability and reliability of individual grid resources, as these concepts are defined in section 2. The need for these types of metrics is also supported indirectly by work on fault tolerance for individual grid resources described at length in section 3. As a practical matter, some basic measure of the effectiveness of different solutions is needed to provide a baseline for evaluation and comparison. Similarly, grid users, even at this early stage in the development of grid technology, need standard reliability and availability metrics to evaluate potential use of different resources they might find in a grid.
- The third is a broader measure of reliability that can be applied to larger portions of a grid system. The one example of [Xie2004] involves individual grid workflows and collections of workflows that comprise an operational grid system at any time. The requirement for this type of metric is also underscored by needs of users who must be able to evaluate reliability of a grid system as a whole.

The second and third classes of metrics are based on records of observed incidences of faults occurring in individual components or groups of components. A fourth possible class of metrics can be considered that measures potential for global system failure or performance degradation that stems from hard-to-predict emergent behaviors precipitated by the actions of many individual grid components. Here, measurement methods are needed to evaluate, and predict, behavior of grid systems as complex systems [Mill2006]. This class of metrics, in contrast to the first three, appears to be more the subject of basic research.

7. Summary and Future Work.

To be written -- this section summarizes the document and discusses future work, if any.

8. Resources.

[Abaw2004] J. H. Abawajy, J., "*Fault*-Tolerant Scheduling Policy for Grid Computing systems", *18th International Parallel and Distributed Processing Symposium*, April, 2004, Santa Fe, New Mexico.

[Acce2007] AccessGrid Home Page. http://www.accessgrid.org/, 2007.

[Alti2005] Altintas, I. et al., "A Framework for the Design and Reuse of Grid Workflows," *Lecture Notes on Computer Science* 3458, 2005.

[Anan2003] Anand, S., et al., "Flow-based Multistage Co-allocation Service," The 2003 International Conference on Communications in Computing, Las Vegas, Nevada, USA, June 2003.

[Andr2002] Andrzejak, A., Graupner, S., Kotov, V., and Trinks, H., "Algorithms for Self-Organization and Adaptive Service Placement in Dynamic Distributed Systems," Hewlet Packard Corporation, *HPL-2002-259*, 2002.

[Alvi2001] Alvisi, L., and et. al. "Wrapping Server-Side TCP to Mask Connection Failures," *in INFOCOM 2001*, 22-26 April 2001, vol. 1, pp. 329-337.

[Arno1999] Arnold, K., et al, *The Jini Specification, V1.0* Addison-Wesley 1999. (Latest version is 1.1 available from Sun)

[Aviz2004] Avizienis, A., Laprie, J., Randell, B., and Landwehr, C. "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transactions on Dependable and Secure Computing*, Volume 1, Number. 1, January-March 2004.

[Bane2002] Banerjee, S., Bhattacharjee, B., and Kommareddy, C., "Scalable Application Layer Multicast," *ACM SigComm*, 2002.

[Barc2005] Barcello, M., "Evaluating High-Throughput Reliable Multicast for Grid Applications in Production Networks," *2005 IEEE International Symposium on*
.

[Bart2003] Bartolini, N., Presti, F.L. , and Petrioli, C. "Optimal Dynamic Replica Placement in Content Delivery Networks," *The 11th IEEE International Conference on Networks, ICON 2003*, 2003, pp. 125-130.

[Batc2004] R. Batchu, Y. Dandass, A. Skjellum, and M. Beddhu, ''MPI/FT: A Model-Based Approach to Low-Overhead Fault Tolerant Message-Passing Middleware,'' *Cluster Computing*, pp. 303–315, Oct. 2004.

[Baus2003] Bausch, W., Pautasso, C., and Alonso, G., "Programming for Dependability in a Service-based Grid," *Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID.03)*, 2003.

[Bell2002] Bell, W., et al. "Simulation of dynamic grid replication strategies in optorsim," *Proceedings of 3rd International IEEE Workshop on Grid Computing*, pp. 46–57, 2002.

[Bell2003] Bell, W., et al., "Evaluation of an economy-based file replication strategy for a data grid," *International Workshop on Agent based Cluster and Grid Computing*, pp. 120–126, 2003.

[Bezz2006] Bezzine, S., et al., "A Fault Tolerant and Multi-Paradigm Grid Architecture for Time Constrained Problems: Application to Option Pricing in Finance," Second IEEE International Conference on e-Science and Grid Computing, 2006, p. 49, December 2006.

[Bosc2002] Bosilca, G., et al., "MPICHV: Toward a Scalable Fault Tolerant MPI for Volatile Nodes", *Proceedings of IEEE SuperComputing*, November 2002.

[Bout2005] Bouteiller, A., et al., "MPICH-V: a Multiprotocol Automatic Fault Tolerant MPI," International Journal of High Performance Computing and Applications, Volume 20, Issue 3, pp. 319-330, 2006.

[Bunt2007] Buntinas, D., et al., "Blocking vs. Non-Blocking Coordinated Checkpointing for Large-Scale Fault Tolerant MPI," Accepted for publication in *Future Generation Computer Systems*, Elsevier Press, 2007

[Chen2002a] Chen, M., Kiciman, E., Fratkin, E., Fox, A., and Brewer, E. "Pinpoint: Problem Determination in Large, Dynamic Internet Services", *Proceedings of 2002 International Conference on Dependable Systems and Networks (DSN),* IPDS track, Washington, DC, June 23-26, 2002.

[Cher1999] Chervenak, A., et al., "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Data Sets," *Journal of Network and Computer Applications*, 2001(23): pp. 187-200.

[Cher2002] Chervenak, A., et al., "Giggle: A Framework for Constructing Scalable Replica Location Services," *SC2002 Conference*, Baltimore, MD USA, 2002.

[Cher2004]. Chervenak, A.L., et al., "Performance and Scalability of a Replica Location Service," *Thirteenth IEEE Int'l Symposium High Performance Distributed Computing (HPDC-13*), Honolulu, HI USA, 2004.

[Cher2005] Chervenak, A.,Schuler, R., Kesselman, C., Koranda, S., and Moe, B. "Wide area data replication for scientific collaborations," *Proceedings of the 6th International Workshop on Grid Computing*, November 2005.

[Chiu1998] Chiu, D., Hurst, S., Kadansky, M., and Wesley, J., "TRAM: A Tree-based Reliable Multicast Protocol," Sun Microsystems Laboratories. SMLI TR-98-68, 1998.

[Choi1999] J. Choi, M. Choi, and S. Lee. An Alarm Correlation and Fault Identification Scheme Based on OSI Managed Object Classes. In IEEE International Conference on Communications, Vancouver, BC, Canada, 1999.

[Chun2004] Chun, G., et al., "Benchmark Probes for Grid Assessment," *The 18th International Parallel and Distributed Processing Symposium (IPDPS'04)*, p. 276a, 2004.

[Clap2004] Clapp, G., Gannet, J., and Skoog, R., "Requirements and Design of a Dynamic Grid Networking Layer," *2004 IEEE International Symposium on Cluster Computing and the Grid*, 2004.

[Coll2007] Colling, D., et al., "On Quality of Service Support for Grid Computing," *The 2nd International Workshop on Distributed Cooperative Laboratories and Instrumenting the GRID (INGRID 2007)*, April, 2007

[Cond2007] "Adding high availability to Condor Central manager," See http://dsl.cs.technion.ac.il/projects/gozal/project_pp./ha/ha.html.

[Cox2002] Cox, W., et al, *Web Services Transaction (WS-Transaction)*, 2002. See http://dev2dev.bea.com/pub/a/2004/01/ws-transaction.html.

[Cybo2006] Cybok, D., "A Grid workflow infrastructure," *Concurrency and Computation: Practice And Experience*, Volume 18, Issue 10, pp. 1243–1254, 2006.

[Czaj1999] Czajkowski, K., Foster, I., and Kesselman, C., "Resource Co-Allocation in Computational Grids," *IEEE International Symposium on High Performance Distributed Computing (HPDC-8)*, August 1999, pp. 219-228.

[Das2002] A. Das, I. Gupta, and A. Motivala, "Swim: Scalable weakly-consistent infection-style process group membership protocol," in Proc. of Intl. Conf. on Dependable Systems and Networks (DSN'02), pp.303–312, June 2002.

[Deel2003] Deelman, E., et al., "Mapping Abstract Complex Workflows onto Grid Environments," *Journal of Grid Computing*, Volume 1, pp. 25-39, 2003.

[Demm1989] Demmy, W. and Petrini, A., "Statistical Process Control in Software Quality Assurance," *Proceedings of the 1989 National Aerospace and Electronics Conference*, Dayton, Ohio, pp. 1585-1590, May 1989.

[Deri2004] Deris, M., Abawajy, J., Suzuri, H. "An efficient replicated data access approach for large-scale distributed systems," *IEEE International Symposium on Cluster Computing and the Grid*, April 2004.

[Dull2001] Dullman, D. et al., "Models for Replica Synchronisation and Consistency in a Data Grid," *Proceedings. 10th IEEE International Symposium on High Performance Distributed Computing,* pp.67-75, 2001.

[Duar2006] Duarte, A., Brasileiro, F., Cirne, W., an dFilho, J., "Collaborative Fault Diagnosis in Grids through Automated Tests," *Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06)*, 2006.

[Dura2005] Durand, J., and Karmarkar, A., "Message Reliability Protocol Standards for Web Services : An Analysis," *The 3rd IEEE European Conference on Web Services (IEEE ECOWS 2005)*, November 2005, Växjö, Sweden

[Elno2002] Elnozahy, E., Johnson, D., and Wang, Y., "A survey of rollback recovery protocols in message-passing systems," *ACM Computing Surveys*, Volume 34, Issue3, pp. 375–408, 2002.

[Emme2005] Emmerich, W., et al., "Grid Service Orchestration Using the Business Process Execution Language (BPEL)," *Journal of Grid Computing*, Volume 3, pp. 283–304, 2006.

[Fahr2005] Fahringer, T., et al., "ASKALON: a tool set for cluster and Grid computing," *Concurrency and Computation: Practice and Experience*, Volume 17, pp. 143-169, 2005.

[Fang2007] Fang, C., et al. "Fault tolerant Web Services," *Journal of Systems Architecture*, Volume 53, Issue 1, January 2007, pp. 21-38 (Request #45405, received /22/07)

[Fost2005] Foster et al., A Globus Primer Describing Globus Toolkit Version 4, Draft May 8, 2005. http://www.globus.org/toolkit/docs/4.0/key/GT4_Primer_0.6.pdf

[Frey2001] Frey, J., Tannenbaum, T., Livny, M., Foster, I., Tuecke, S., "Condor-G: A Computation Management Agent for Multi-Institutional Grids," *Proceedings of the Tenth IEEE International Symposium on High Performance Distributed Computing*, San Francisco, CA, USA, August 7-9, 2001, pp. 55-67.

[Fox2005] Fox, G., Pallickara, S., Pierce, M., and Gadgil, H., "Building Messaging Substrates for Web and Grid Applications," *Philosophical Transactions of the Royal Society: Mathematical, Physical and Engineering Sciences (Scientific Applications of Grid Computing Special Issue)*, Volume 363 Issue 1833, pp.1757–1773, 2005

[Fox2006] Fox, G., "Collaboration and Community Grids," *International Symposium on Collaborative Technologies and Systems*, pp. 419- 428, May 2006.

[Gabr2003] Gabriel, E., Fagg, et al., "A Fault-Tolerant Communication Library for Grid Environments," *Seventeenth Annual ACM International Conference on Supercomputing (ICS'03), International Workshop on Grid Computing and e-Science*, San Francisco, June 2003

[Glob2005] *Reliable File Transfer (RFT) Service*, Globus Toolkit, version 4.0, http://www.globus.org/toolkit/docs/4.0/data/rft/.

[Grah2002] Graham, R., et al., “A Network-Failure-Tolerant Message-Passing System For Terascale Clusters,” *Proceedings of the 16th international conference on Supercomputing*, New York, USA, pp. 77 – 83, June 2002.

[Gray2004] Gray, J. and Lamport, L., “Consensus on Transaction Commit,” *Microsoft Research Corporation*, MSR-TR-2003-96.

[Grus1998] Gruschke, B., “A New Approach for Event Correlation based on Dependency Graphs,” *Fifth Workshop of the OpenView University Association: OVUA’98*, Rennes, France, April 1998.

[Gupt2001] Gupta, T. D. Chandra, and G. S. Goldszmidt, “On scalable and efficient distributed failure detectors,” *Proceedings of 20th Annual ACM Symposium on Principles of Distributed Computing*, pp. 170–179, 2001.

[Hill2005] Hillenbrand, M., Götze, J., and Müller, P., “Creating Dependable Web Services Using User-transparent Replica,” *Proceedings of the International Conference on Next Generation Web Services Practices (NWeSP’05)*, 2005.

[Hilt2001] Hiltunen, M.A.; Schlichting, R.D.; Ugarte, C.A., “Enhancing survivability of security services using redundancy,” *Proceedings of the 2001 International Conference on Dependable Systems and Networks*, pp.173–182, July 2001.

[Hori2005] Horita, Y., Taura, K., and Chikayama, T. A Scalable and Efficient Self-Organizing Failure Detector for Grid Applications, *Grid Computing Workshop*, 2005.
[Hosc2000] W. Hoschek, W., et al. “Data management in an international data grid project,” *Proceedings of GRID Workshop*, pp. 77–90, 2000.

[Hued2006] Huedo, E., Montero, R. S., and Llorente, I. M. “Evaluating the reliability of computational grids from the end user's point of view. *Journal of Systems Architecture*, Volume 52, Issue 12, pp. 727-736, December 2006. (request #45394, arrived 8/20)

[Hwan2003] Hwang, S., and Kesselman, C., GridWorkflow : A Flexible Failure Handling Framework for the Grid,” In *Proceedings of the 12th IEEE Intl. Symposium on HPDC*, 2003.

[Iamn2000] Iamnitchi, A. and Foster, I. "A problem specific fault tolerance mechanism for asynchronous, distributed systems," in *Proceedings of 2000 International Conference on Parallel Processing (29th ICPP'00)*, Toronto, Canada, August 2000, IEEE.

[Ietf1985] File Transfer Protocol, Internet Engineering Task Force (IETF), http://www.ietf.org/, RFC 959, October 1985.

[Ietf1995] A *Border Gateway Protocol 4 (BGP-4)*, Internet Engineering Task Force (IETF), http://www.ietf.org/, RFC 1771, March 1995.

[Ietf1999] *Multicast Dissemination Protocol version 2 (MDPv2) – Internet Draft*, Internet Engineering Task Force, October 1999.

[Ietf2001] *Multiprotocol Label Switching Architecture*, Internet Engineering Task Force (IETF), http://www.ietf.org/, RFC 3031, January 2001.

[Ietf2002a] Version 2 of the Protocol Operations forthe Simple Network Management Protocol (SNMP),Internet Engineering Task Force (IETF), http://www.ietf.org/, RFC 3416, December 2002.

[Ietf2002b] Overview and Principles of Internet Traffic Engineering, Engineering Task Force (IETF), http://www.ietf.org/, RFC 3272, May 2002.

[Ietf2002c] *Applicability Statement for Traffic Engineering with MPLS*, Internet Engineering Task Force (IETF), http://www.ietf.org/, RFC 3346, August 2002.

[Ietf2007a] *Open Shortest Path First IGP (Interior Gateway Protocol)*, Internet Engineering Task Force (IETF), http://www.ietf.org/ 2007.

[Ietf2007b] *NACK-Oriented Reliable Multicast (NORM) Protocol*, Internet Engineering Task Force (IETF), http://www.ietf.org/, March 2007.

[Jain2004] Jain, A. and Shyamasundar, R., Failure Detection and Membership Management in Grid Environments, in *Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing (GRID'04)*, 2004.

[Jits2007] Jitsumoto, H., Endo, T., Matsuoka, S., "ABARIS: An Adaptable Fault Detection/Recovery Component Framework for MPIs," *IEEE International Parallel and Distributed Processing Symposium (IPDPS 2007)* pp.1-8, March 2007.

[Jo2005] Jo, J., Seok, W., Kwak, J. and Byeon, O., "Design and Implementation of QoS Measurement and Network Diagnosing Framework for IP Multicast in Advanced Collaborative Environment," *Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science (ICIS'05),* 2005.

[Juha2003] Juhasz, Z., Andics, A., and Szabolcs P., "Towards a Robust and Fault-Tolerant Discovery Architecture for Global Computing Grids" *Scalable Computing: Practice and Experience*, Volume 6, Number 2, pp. 22-33. 2003.

[Keya2002] Keyani, P., Larson, B., and Senhil, M. "Peer Pressure: Distributed Recovery from Attacks in Peer-to-Peer Systems", in *Web Engineering and Peer-to-Peer Computing*, Gregori, E. et al. (eds.), NETWORKING 2002 Workshops, Pisa, Italy, May 19-24, 2002, Revised Papers, Lecture Notes in Computer Science 2376 Springer 2002, ISBN 3-540-44177-8, pp. 306-320.

[Khar2004] Kharchenko, V.,Popov, P., andRomanovsky, A., "On Dependability of Composite Web Services with Components Upgraded Online," *Proceedings of the International Conference on Dependable Systems and Networks (DSN 2004)*, Florence, Italy, pp. 287–291, June 2004.

[Koeh2003] Koehler, J., and Srivastava, B., "Web Service Composition - Current Solutions and Open Problems." ICAPS 2003 Workshop on Planning for Web Services, pp. 28 – 35, 2003.

[Kola2005] Kola, G., Kosar, T., and Livny, M., "Faults in Large Distributed Systems and What We Can Do About Them", *Proceedings of 11th European Conference on Parallel Processing (Euro-Par 2005)*, pp. 442-453, Lisbon Portugal, August 2005.

[Kris2002] Krishnan S., Wagstrom P., and von Laszewski G., "GSFL: A workflow framework for Grid services," http://www-unix.globus.org/cog/papers/gsfl-paper.pdf, January 2004.

[Kuo2005] Kuo, D. and Mckeown, M., "Advance Reservation and Co-Allocation Protocol For Grid Computing," *Proceedings of the First International Conference on e-Science and Grid Computing (e-Science'05)*, 2005.

[Lac2006] Lac, C. and Ramanathan, S., "A Resilient Telco Grid Middleware," *Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC'06)*, pp. 306-311, 2006

[Lame2002] Lamehamedi, H., Szymanski, B., Shentu, Z., Deelman, E. , "Data replication strategies in grid environments," *Proceedings of the Fifth International Conference on Algorithms and Architectures for Parallel Processing*, 2002, pp. 378- 383.

[Lamp2001] L. Lamport, L., Paxos made simple. *ACM SIGACT News (Distributed Computing Column)*, Volume 32, Number 4, pp. 18-25, 2001.

[Lan2002] Lan, J. *Cache Consistency Techniques for Peer-to-Peer File Sharing Networks*, Master's Thesis, Department of Computer Science, University of Massachusetts Amherst, June 2002.

[Lanf2002] Lanfermann, G., Allen, G., Radke, T., and Seidel, E., "Nomadic Migration: Fault Tolerance in a Disruptive Grid Environment," *Second IEEE/ACM International Symposium Cluster Computing and the Grid, 2002*, pp. 280, May 2002.

[Leau2006] Leai, K., Tan, L., Turner, K. "Orchestrating Grid Services using BPEL and Globus Toolkit 4," *Proceedings of the 7th PGNet Symposium*, pp. 31-36, 2006.

[Lean2004] Leangsuksun, C., et al., "A Failure Predictive and Policy-Based High Availability Strategy for Linux High Performance Computing Cluster," *The Fifth LCI*

International Conference on Linux Clusters: the HPC Revolution 2004, Austin TX USA, May 2004.

[Lee2001] Lee, B. and Weissman, J. B. "Dynamic Replica Management in the Service Grid," in *IEEE 2nd International Workshop on Grid Computing*, November, 2001.

[Lee2003] Lee H., and et. al.,"Grid Fault Tolerance Service for Quality of Service", *The 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2003),* 2003.

[Lei2007] Lei, M.; Vrbsky, S.V.; Zijie, Q., "Online Grid Replication Optimizers to Improve System Reliability," *IEEE International Symposium on Parallel and Distributed Processing Symposium*, pp. 26-30 March 2007

[Li2006] Li, Q., Xu, M., and Zhang, H., "A Root-fault Detection System of Grid Based on Immunology. *Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC 2006)*, Changsha, China, October 2006. pp. 369-373

[Lim2004] Lim, S., Fox, G., Pallickara, S., and Pierce, M., "Web Service Robust GridFTP", *The 2004 International MultiConference in Computer Science and Computer Engineering*, Las Vegas, NV USA, June 2004.

[Lima2005a] K. Limaye, C. B. Leangsuksun, et. al, "Job-Site Level Fault Tolerance for Cluster and Grid environments", *The 2005 IEEE Cluster Computing*, Boston, MA, September, 2005.

[Lima2005b] Limaye, K. Tikotekar, A., and Leangsuksun, B. "Fault tolerance-enabled HPC resource management with HA-OSCAR framework," *High Availability and Performance Computing Workshop*, Santa Fe, NM USA, October 2005.

[Liu2005] Liu, Y. Leangsuksun, C., Song, H., and Scott, S., "Reliability-aware Checkpoint/Restart Scheme: A Performability Trade-off," *Proceedings of IEEE International Conference on Cluster Computing,* September 2005

[Look2004a] Looker, N., Munro, M., and Xu, J., "Practical Dependability Analysis of SOAP Based Systems," *Proceedings of the UK e-Science All Hands Meeting*, Nottigham, UK, pp. 1126–1129, August, 2004.

[Look2004b] Looker, N., Munro, M., and Xu, J., "WS-FIT: A Tool for Dependability Analysis of Web Services," *Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC)*, Hong Kong, pp. 120–123, September, 2004.

[Look2005] Looker, N., Burd, S., Drummond, M., and Munro, M., "Pedagogic Data as a Basis for Web Service Fault Models," *IEEE International Workshop on Service-Oriented System Engineering*, Beijing, China, October 20-21, 2005.

[Look2007] Looker, N., Munro, M., and Xu, J., "Determining the Dependability of Service-Oriented Architectures," *Submitted to the International Journal of Simulation and Process Modelling*, 2007.

[Loug2002] Loughran, *Making Web Services that Work*, HP Laboratories, Hewlet-Packard Corporation, HPL-2002-274, 2002.

[Louc1998] Louca, S., Neophytou, N., Lachanas, A., and Evripidou, P., "MPI-FT: A portable fault tolerance scheme for MPI," *Proceedings of the PDPTA '98 International Conference*, Las Vegas, Nevada 1998.

[Lowe2003] Lowekamp, B., et al., "Enabling Network Measurement Portability Through a Hierarchy of Characteristics," *Proceedings of the Fourth International Workshop on Grid Computing (GRID'03)*, 2003.

[Liu2004] Liu, X., Xia, H., and Chien, A., "Validating and Scaling the MicroGrid: A Scientific Instrument for Grid Dynamics," *Journal of Grid Computing*, Volume 2, Number 2, pp. 141-161, 2004.

[Lui2006] Lui, P. and Wu, J. J., "Optimal Replica Placement Strategy for Hierarchical Data Grid Systems," *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)*, pp. 417-420, 2006

[Macl2006] MacLaren, J., Keown, M., and Pickles,S., "Co-Allocation, Fault Tolerance and Grid Computing," *Proceedings of the UK e-Science All Hands Meeting* pp. 155–162, 2006.

[Marc2001] Marchetti, C., Virgillito, A., and Baldoni, R. "Design of an Interoperable FT-CORBA Compliant Infrastructure," *Proceedings of the European Research Seminar on Advances in Distributed Systems* (ERSADS), 2001.

[Matt2006] Mattmann, C., et al., "A Classification and Evaluation of Data Movement Technologies for the Delivery of Highly Voluminous Scientific Data Products," National Aeronautics and Space Administration, Document 20060044153, 2006.

[Milo2000] Milojicic, D., Douglis, F., Paindaveine, Y., Wheeker, R., and Zhou, S. "Process Migration Survey," *ACM Computing Surveys*, September, 2000.

[Mill2006] Mills, K. and Dabrowski, C. "Investigating Global Behavior in Computing Grids." *Self-Organizing Systems, Lecture Notes in Computer Science*, Vol. 4124, pp. 120-136, 2006.

[Mill2007] Mills, K. and Dabrowski, C., "Can Economics-based Resource Allocation Prove Effective in a Computation Marketplace?" accepted for publication to the *Journal of Grid Computing*, 2007.

[Mogi2006] Mogilevsky, D., Koenig, G., Yurcik. W., "Byzantine Anomaly Testing for Charm++: Providing Fault Tolerance and Survivability for Charm++ Empowered Clusters," *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid Workshops (CCGRIDW'06)*, p. 30, May 2006.

[Mpi2003] MPI*: A Message-Passing Interface Standard*, Message Passing Interface Forum, http://www.mpi-forum.org/, 2003.

[Oasi2004a] *Business Transaction Protocol (BTP) Version 1.1*, Committee Draft, June 2004.

[Oasi2004b] *Web Services Base Faults (WS-BaseFaults)*, OASIS, 2004.

[Oasi2004c] *WS-Reliability 1.1*, OASIS, Committee Draft 1.086, August 2006.

[Oasi2006a] *Web Services Business Process Execution Language (WSBPEL)*, OASIS WS-BPEL 2.0 Committee Draft, May 2006.

[Oasi2006b] *Web Services Reliable Messaging (WS-ReliableMessaging)*, Committee Draft 04, wsrm-1.1-spec-cd-04, August 2006

[Oasi2007] *Web Services Coordination (WS-Coordination), Version 1.1* OASIS Standard, April 2007.

[Ogf2004a] *Networking Issues for Grid Infrastructure*, Open Grid Forum Informational Document, GFD-I.037, November 2004.

[Ogf2004b] *A Hierarchy of Network Performance Characteristics for Grid Applications and Services*, Open Grid Forum, GFD-R-P.023 (Proposed Recommendation), May 2004.

[Ogf2005a] *An Architecture for Grid Checkpoint and Recovery (GridCPR) Services and a GridCPR Application Programming Interface*, Draft Document, Global Grid Forum, 2005.

[Ogf2005b] *GridFTP v2 Protocol Description*, GFD-R-P.047, Open Grid Forum, May 2005.

[Ogf2006a] *OGSA WSRF Basic Profile 1.0*, Open Grid Forum, GFD.72, September 2006.

[Ogf2006b] *Configuration Description, Deployment, and Lifecycle Management CDDLM Deployment API*, Open Grid Forum, GFD.69, April 2006.

[Ogsa2006c] *The Open Grid Services Architecture, Version 1.5*, Open Grid Forum, GFD.80, September 2006.

[Ogf2007a] *Use-Cases and Requirements for Grid Checkpoint and Recovery*, Version 1.0, Open Grid Forum, GFD-I.92, May 2007.

[Ogf2007b] *Web Services Agreement Specification (WS-Agreement)*, Open Grid Forum, GFD.107, May 2007.

[Natr2001] Natrajan, A., Humphrey, M., and Grimshaw, A., "Capacity and Capability Computing in Legion," *The 2001 International Conference on Computational Science*, May 2001

[Neko2005] Nekovee, M., Barcellos, M., and Daw, M., "Reliable multicast for the Grid: a case study in experimental computer science," *Philosophical Transactions of the Royal Society A*, Volume 10, Number 1098, 2005.

[Pall2005] Pallickara, S., Fox, G., and Pallickara, S.L., "An Analysis of Reliable Delivery Specifications for Web Services", *International Conference on Information Technology: Coding and Computing, 2005 (ITCC 2005)*, Volume1, pp. 360-365, April 2005.

[Pope2007] Popescu, A., Constantinescu, D., Erman, D., Ilie, D., "A Survey of Reliable Multicast Communication," *Third EuroNGI Conference on Next Generation Internet Networks,* pp.111-118, May 2007.

[Qui2001] Qiu, L., Padmanabhan, V., and Voelker, G. "On the Placement of Web Server Replicas", *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies - INFOCOM 2001*, pp. 1587-1596.

[Ra2005] Ra, D., et al., "Scalable Enterprise Level Workflow Manager for the Grid," *Proceedings of the Fifth International Conference on Quality Software (QSIC'05)*, pp. 341-348, September 2005.

[Rang2001] Ranganathana, K., and Foster, I. "Identifying dynamic replication strategies for a high performance data grid," *Proceedings of the International Grid Computing Workshop*, pp. 75–86, 2001.

[Rang2002] Ranganathan K., Iamnitchi, A., and Foster, I., "Improving Data Availability through Dynamic Model-Driven Replication in Large Peer-to-Peer Communities," in *Global and Peer-to-Peer Computing on Large Scale Distributed Systems Workshop*, Berlin, May 2002, p. 376.

[Rena2006] Ranaldo, N., Tretola, G., and Zimeo, E., "Hierarchical and Reliable Multicast Communication for Grid Systems," *Current & Future Issues of High-End Computing, Proceedings of the International Conference ParCo*, pp. 137-144, 2005

[Ripe2002] Ripeanu, M., and Foster, I., "A Decentralized, Adaptive Replica Location Mechanism," *Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing*, 2002.

[Rti2002] Research Triangle Institute, *The Economic Impacts of Inadequate Infrastructure for Software Testing*, May 2002.

[Sant2005] Santos, G. T., Lung, L.C., and Montez, C., “FTWeb: A Fault Tolerant Infrastructure for Web Services,” *Proceedings of the 2005 Ninth IEEE International EDOC Enterprise Computing Conference (EDOC’05).*

[Schn2006] Schneider, J., Linnert, B., and Burchard, L., "Distributed Workflow Management for Large-Scale Grid Environments," *International Symposium on Applications and the Internet (SAINT'06)*, 2006, pp. 229-235.

[Song2007] Song, C.X., Topkara, U., Woo, J., and Park, S.K., "Assessing Reliability of Grid Software Systems Using Emergent Features," *The 2nd Workshop on Reliability and Robustness in Grid Computing Systems, the 19th Open Grid Forum (OGF19)*, Chapel Hill, NC. January, 2007

[Stel1999] Stelling, P., Foster, I. Kesselman, C., Lee, C., and von Laszewski, G. “A Fault Detection Service for Wide Area Distributed Computations”, *Cluster Computing*, Volume 2, Number 2, 1999, pp. 117-128.

[Stoc2001] Stockinger, H., et al., “File and object replication in data grids,” *Tenth IEEE Symposium on High Performance and Distributed Computing*, pp. 305–314, 2001

[Sun2005] *N1 Grid Engine User’s Guide*, Sun MicroSystems, Inc., May 2005.

[Tai2004] Tai, S., Mikalsen, T., and Rouvellou, I., “Using Message-Oriented Middleware for Reliable Web Services Messaging,” *Lecture Notes on Computer Science*, Number 3095, July 2004

[Taki2005] Takizawa, S. et al., “A Scalable Multi-Replication Framework for Data Grid,” *Proceedings of the 2005 Symposium on Applications and the Internet Workshops (SAINT-W’05)*, 2005.

[Tann2002] Tannenbaum, T., Wright, D., Miller, K, and Livny, M. “Condor - A Distributed Job Scheduler,” In *Beowulf Cluster Computing with Linux*, The MIT Press, MA, USA, 2002.

[Tart2002] Tartanoglu, F., Issarny, V., Romanovsky, A., Levy, N., “Dependability in the Web Service Architecture.” Proceedings of the ICSE 2002 Workshop on Architecting Dependable Systems (Orlando, Florida, USA), May 2002.

[Tart2003] Tartanoglu, F., Issarny, V., Romanovsky, A., Levy, N., “Coordinated Forward Error Recovery for Composite Web Services,” Proceedings of the 22nd International Symposium on Reliable Distributed Systems, SRDS (Florence, Italy), October 2003.

[Tauf2005] Taufer, M.,Teller, P., Anderson, D., Brooks, C., “Metrics for Effective Resource Management in Global Computing Environments,” *First International Conference on e-Science and Grid Computing,* p. 8, December 2005.

[Topk2006] Topkara, U., Song, C.X., Woo, J., and Park, S.K., "Connected in a Small World: Rapid Integration of Biological Resources", *Grid Computing Environments Workshop (in conjuction with Supercomputing'06)*, Tampa, FL, November, 2006

[Town2005] Townend, P., Groth, P., Looker, N. and Xu, J. FT-Grid: A Fault-Tolerance System for e-Science, *Proceedings of the UK oST e-Science Fourth All Hands Meeting (AHM05)*, September 2005.

[Turn2007] Turner, K. and Tan, K., “Graphical Composition of Grid Services,” *Lecture Notes in Computer Science* 4401, pp. 1-17, Springer, Berlin, May 2007.

[Yemi1996] A. Yemini and S. Kliger. High Speed and Robust Event Correlation. *IEEE Communication Magazine*, Volume 34 Number 5, pp. 82–90, May 1996.

[Urga2001] Urganonkar, B. et al. “Maintaining Mutual Consistency for Cached Web Objects”, *Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS-21*), Phoenix, Arizona, April 2001

[Valc2005] Valcarenghi, L. and Piero C. “QoS-Aware Connection Resilience for Network-Aware Grid Computing Fault Tolerance”, *Proceedings of 2005 7th International Conference on Transparent Optical Networks*, July 3-7 2005, Barcelona Spain, Volume 1, pp. 417-422.

[Verm2003] Verma, D., and et al. “SRIRAM: A scalable resilient autonomic mesh”, *IBM SYSTEMS JOURNAL*, Volume 42, Number 1, pp. 19-28, 2003.

[vonL2004] von Laszewski, G., et al., “GridAnt: A Client-Controllable Grid Workflow System,” *Argonne National Laboratory Preprint* ANL/MCS-P1098-1003 and *Thirty-seventh Hawai'i International Conference on System Science*, Island of Hawaii, Big Island, January 2004.

[Wald2006] Waldrich, O. Wieder, P. Ziegler, W., “A Meta-Scheduling Service for Co-allocating Arbitrary Types of Resources,” *Lecture Notes on Computer Science* 3911, pp. 782-791, 2006.

[Wang2006] Wang, X., Zhuang, Y., Hou, H., "Byzantine Fault Tolerance in MDS of Grid System," *International Conference on Machine Learning and Cybernetics,* pp.2782-2787, August 2006.

[Wate2004] Waters, G., Crawford, J., and Lim, S., “Optimising Multicast Structures for Grid Computing,” *Computer Communications*, Volume 27, pp. 1389-1400, September 2004.

[Woo2003] Woo, N., et al., MPICH-GF: Providing fault tolerance on grid environments", *The 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid2003)*, May 2003.

[Wsdl2001] *Web Services Description Language (WSDL) 1.1*, "http://www.w3.org/TR/2001/NOTE-wsdl-20010315"

[Wsrm2005] *Web Services Reliable Messaging Protocol (WS-ReliableMessaging), BEA Systems*, IBM, Microsoft Corporation, Inc, and TIBCO Software Inc., 2005.

[W3c2007] *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*, World Wide Web Consortium (W3C), W3C recommendation, April 2007.

[Xian2006] Xiang, Y., Li, Z., and Chen, H., "Optimizing Adaptive Checkpointing Schemes for Grid Workflow Systems," *Proceedings of the Fifth International Conference on Grid and Cooperative Computing Workshops (GCCW'06)*, 2006.

[Xie2004] Xie, M., Dai, Y., and Poh, K., *Computing Systems Reliability*, Kluwer Academic Publishers: New York, NY, U.S.A., 2004.

[Yeom2006] Yeom, H., "Providing Fault-tolerance for Parallel Programs on Grid (FT-MPICH)", presented at the GGF First Workshop of Reliability and Robustness in Grid Computing Systems, Athens, Greece, February 2006.

[Yosh2005] Yoshimoto, K., Kovatch, P., Andrews, P. "Co-Scheduling with User-Settable Reservations," LNCS, 3834 ed., Workshop on Job Scheduling Strategies for Parallel Processing, Jun. 2005, pp. 146-156.

[Yu2004] Yu, J., Buyya, R., "A Novel Architecture for Realizing Grid Workflow using Tuple Spaces," *The 5th IEEE/ACM International Workshop on Grid Computing (Grid 2004)*, Pittsburgh, USA, November 2004.

[Yu2005} Yu, J., and Buyya, R., *A Taxonomy of Workflow Management Systems for Grid Computing*. Technical Report GRIDS-TR-2005-1, University of Melbourne, Australia, March 10 2005.

[Zand1999] Zandy, V., Miller, B., and Livny, M. "Process Hijacking," *The Eighth International Symposium on High Performance Distributed Computing*, pp. 177-184, August 1999.

[Zhan2004] Zhang, X., Zagorodnov, D. Hiltunen, M., Marzullo, K. and Schlichting, R. "Fault–tolerant Grid Services Using Primary–Backup: Feasibility and Performance", *Cluster 2004*, San Diego, California, September 2004.

[Zhan2006a] Zhang, X., Junqueira, F., Hiltunen, M., Marzullo, K. and Schlichting, R. "Replicating Nondeterministic Services on Grid Environments," *15th IEEE International Symposium on High Performance Distributed Computing*, 2006 June 2006 pp.105 – 116.

[Zhan2006b] Zhang, Q., et al., "Dynamic Replica Location Service Supporting Data Grid Systems," *Sixth IEEE International Conference on Computer and Information Technology (CIT'06)*, p. 61, 2006.

Appendix

This section provides a cross-reference list of major topical areas of grid reliability and references. Each section lists references for works that focus on reliability in grid computing as well references for more general works that are relevant to grid computing.

A.1 Grid Resources (Section 3.1)

Fault Detection

Grid References: [Duar2006], [Hori2005], [Ietf2002a], [Jain2004], [Jits2007], [Kola2005], [Stel1999], [Li2006], [Wang2006], [Xian2006].
General References: [Chen2002a], [Choi1999], [Das2002], [Gupt2001], [Grus1998], [Keya2002], [Mogi2006], [Oasi2004b], [Yemi1996].

Recovery

Checkpointing and Process Migration

Grid References: [Bosc2002], [Bout2005], [Cond2007], [Gabr2003], [Jits2007], [Lanf2002], [Lean2004], [Lima2005a], [Lima2005b], [Liu2005], [Ogf2005a], [Sun2005], [Woo2003], [Yeom2006].
General References: [Batc2004], [Bunt2007], [Elno2002], [Grah2002], [Louc1998], [Milo2000], [Zand1999].

Replication of individual resources (services)

Grid References: [Abaw2004], [Andr2002], [Lac2006], [Lee2001], [Town2005], [Valc2005], [Verm2003], [Zhan2004], [Zhan2006a].
General References: [Marc2001], [Hill2005], [Qui2001], [Sant2005].

Replication in data grids

Grid References: [Bell2002], [Bell2003], [Cher2002], [Cher2004], [Cher2005], [Deri2004], [Dull2001], [Hosc2000], [Lame2002], [Lei2007], [Lui2006], [Rang2001], [Rang2002], [Ripe2002], [Stoc2001], [Taki2005], [Zhan2006b].

Fault Removal through testing and code certification

References: [Demm1989], *[Khar2004],* [Look2004a], [Look2004b], [Look2005], [Look2007], [Rti2002], [Song2007], [Topk2006].

A.2 Grid Infrastructure and Resource Management Services (section 3.2)

Please note that the references listed above in section A.1 also contain methods that apply to grid infrastructure and resource management services. The following references discuss reliability specifically for infrastructure and management services.

Grid References: [Anan2003] [Cher2002] [Cher2004], [Cher2005], [Czaj1999], [Deri2004], [Hilt2001], [Hued2006], [Juha2003], [Kuo2005], [Macl2006], [Ripe2002], [Wald2006], [Yosh2005].
General References: [Gray2004].

A.3 Grid Connection and Transport Reliability (section 3.3)

Specifications for Reliable Connection and Transport

Grid References: [Glob2005], [Lim2004], [Matt2006], [Ogf2005b].
General References: [Alvi2001], [Dura2005], [Fang2007], [Ietf1985], [Oasi2006b], [Oasi2004c], [Pall2005], [Tai2004], [Wsdl2001], [Wsrm2005], [W3c2007].

Research in Fault-Tolerant Grid Networks

Grid References: [Clap2004], [Fox2005], [Fox2006], [Lim2004], [Ogf2004a], [Valc2005].
General References: [Ietf1995], [Ietf2001], [Ietf2002b], [Ietf2002c], [Ietf2007a].

Reliable Multicasting

Grid References: [Acce2007], [Bane2002], [Barc2005], [Fox2005], [Ietf2007b], [Jo2005], [Neko2005], [Rena2006], [Wate2004].
General References: [Chiu1998], [Pope2007].

A.4 Reliability Concerns from an Overall System Perspective (section 3.4)

Grid References: [Bezz2006], [Fox2006], [Liu2004], [Mill2006], [Mill2007], [Ogf2006a], [Ogf2006c], [Song2007], [Topk2006], [Xie2004].

A.5 Reliability of Grid Applications (section 4)

Fault Tolerance of Remote Application Processes (section 4.1)

Grid References: [Ogf2005a], [Ogf2007a].

Fault Tolerance of Grid Resource Compositions and Workflows (section 4.2)

Languages and tools for grid environments.

Grid References: [Alti2005], [Baus2003], [Deel2003], [Fahr2005], [Hwan2003], [Kris2002], [Ogf2005a], [Ogf2007a], [Ra2005], [Schn2006], [Tann2002], [vonL2004], [Xian2006], [Yu2004], [Yu2005].

Web service languages and tools used in grid environments.

General references: [Cox2002], [Cybo2006], [Emme2005], [Koeh2003], [Leau2006], [Oasi2004a], [Oasi2006a], [Oasi2007], [Tart2003], [Turn2007].

A.6 Preliminary Measurement Requirements

References: [Chun2004], [Coll2007], [Lei2007], [Lowe2003], [Xie2004].

References not yet classified

Sahai, A. Graupner, S., Machiraju, V., van Moorsel, A., "Specifying and Monitoring Guarantees in Commercial Grids through SLA," HPL-2002-324 20021206 http://www.hpl.hp.com/techreports/2002/HPL-2002-324.html

References not yet examined

[Abaw2004b] Abawajy, J. (2004) Fault Detection Service Architecture for Grid Computing Systems, *Lecture Notes in Computer Science*, LNCS 3044, pp. 107-115, Springer-Verlag, Germany

[Abaw2004c] J. H. Abawajy, J., "Autonomic job scheduling policy for grid computing," *Springer. Lecture Notes in Computer Science 3516*, 213-220 (2005).

[Caly2007] Prasad Calyam, P., et al., "Orchestration of Network-wide Active Measurements for Supporting Distributed Computing Applications," Accepted for publication in the *IEEE Transactions on Computers*, 2007.

[Chaf2006] Chafle, G., et al., "Adaptation inWeb Service Composition and Execution," *IEEE International Conference on Web Services (ICWS'06),* 2006, pp. 549-557.

[Chang2002] Chang, B. Y., et al., "Trustless Grid Computing in ConCert," *Proceedings of the Grid Computing – GRID 2002: Third International Workshop*, Baltimore MD, USA, November 18, 2002, pp. 112-125.

[Chen2002b] Chen, Y., Katz, R. H., Kubiatowicz, J. "Dynamic Replica Placement for Scalable Content Delivery", *1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*.

[Dai2002] Dai, Y.S., Xie, M., and Poh, K.L., "Reliability analysis of grid computing systems", *Proceedings of the 2002 Pacific Rim International Symposium on Dependable Computing (PRDC 2002)*, IEEE Computer Society Press, pp 97-103, 2002

[Dai2006] Dai, Y.S., Qi, M., Ran, X., Zou, X., "A new approach for reliability and security in grid computing systems", *The 12th ISSAT International Conference on Reliability and Quality in Design,* Aug. 3-5, 2006.

[Dasg2006] Dasgupta, G., Dasgupta, K., Purohit, A., and Viswanathan, B., "QoS-GRAF: A Framework for QoS based Grid Resource Allocation with Failure provisioning",

[Dial2002] Dialani, V., "Transparent Fault Tolerance for Web Services Based Architectures," *Lecture Notes in Computer Science* 2400, 2002

[He2003] He, X., Sun, X., Von Laszewski, G., "A QoS Guided Scheduling Algorithm for Grid Computing," *Journal of Computer Science and Technology*, Special Issue on Grid Computing, Volume 18, Number 4, 2003.

[Hoar2005] Hoara, W.; Tixeuil, S., "A language-driven tool for fault injection in distributed systems," *Sixth IEEE/ACM International Workshop on Grid Computing*, November 2005.

[Jin2006] Jin, H., Shi, X., Qiang, W., and Zou, D., "DRIC: Dependable Grid Computing Framework," *IEICE Transactions on Information and Systems*, Volume E89-D, Issue 2, pp. 612-623, February 2006. (request #45393)

[Kee2006] Kee, Y., Yocum, K., Chien, A., and Casanova, H., "Improving grid resource allocation via integrated selection and binding," *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, Tampa, Florida, November, 2006.

[Lali2006] Laliwala, A. and Chaudhary, S., "Event-Driven Dynamic Web Services Composition: From Modeling to Implementation," *Innovations in Information Technology*, Volume Iss., pp.1-5, November 2006

[Lui2004] Liu, L, Wu, Q., and Zhou, B., "A Fault-Tolerant Architecture for Grid System," *Lecture Notes in Computer Science*, Volume 3251, pp. 58-64, 2004. (Request # 45403-received 8/15/06)

[Maji2004] Majithia, S.,Walker, D., Gray, W., "Automated Web Service Composition using Semantic Web Technologies," *Proceedings of the International Conference on Autonomic Computing (ICAC'04)*, 2004.

[Mori2006] Moritsu, T., Hiltunen, M., Schlichting, R., Toyouchi, J., and Namba, Y. Using Web Service Transformations to Implement Cooperative Fault Tolerance, Lecture Notes in Computer Science, Volume 4328/2006, pp. 76-91. (*Proceedings of the 3rd International Service Availability Symposium (ISAS)* May, 2006, pp.67-81.) (Request 45404-received 8/15/06)

[Pier2007] Peiris, C., et al. "Implementing Reliable Messaging and Queue-Based Communications." in *Pro WCF: Practical Microsoft SOA Implementation*, SpringerLink, 2007.

[Rahm2004] Rahman, R., Barker, K., and Alhajj, R., "Predicting the performance of gridFTP transfers," *Proceedings of the 18th International Parallel and Distributed Processing Symposium,* pp. 238-, April 2004.

[Tang2004] Tang X., and Xu, J. "On Replica Placement for QoS-Aware Content Distribution," in *INFOCOM 2004*, 2004.

[Vlas2006] Vlassov, V.; Dong Li, D., Popov, K.; Haridi, S. A Scalable Autonomous Replica Management Framework for Grids," *IEEE 2006 International Symposium on Modern Computing*, October 2006, pp. 33-40.

[Wass2006] Wassermann, B. and Emmerich, W., "Reliable Scientific Service Compositions?" *Proceedings of 2nd International Workshop on Engineering Service-Oriented Applications: Design and Composition, WESOA'06*, Chicago, USA, December 2006.

[Zhan2006c] Zhang, X., Hiltunen, M., Marzullo, K. and Schlichting, R. "Customizable Service State Durability for Service Oriented Architectures," *Sixth European Dependable Computing Conference, 2006 (EDCC '06*), Oct. 2006, pp.119 - 128

References that were examined but not integrated

[Affa2006] Affan, M. and Ansari, M., "Distributed Fault Management for Computational Grids," *Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC 2006*), Changsha, China, October 2006. pp. 363-368.

[Aung2006] Aung, K., Park, K., and Park, J., "Survival of the Internet applications: a cluster recovery model," *Sixth IEEE International Symposium on Cluster Computing and the Grid*, Volume 2, pp.16-19 May 2006

[Chan2000] Chandrasekaran, S., Madden S., Ionescu, M. "Ninja Paths: An Architecture for Composing Services over Wide Area Networks", CS262 class project writeup, UC Berkeley (2000)

[Chan2006] Chan, P., Lyu, M., Malek, M. Making Services Fault Tolerant, *ISAS 2006*, *LNCS 4328*, pp. 43–61, 2006.

[Deni2004] Denis, A., et al., "Wide-Area Communication for Grids: An Integrated Solution to Connectivity, Performance and Security Problems,"

[Dobs2005] Dobson, G., Hall, S., and Sommerville, I., A Container-Based Approach to Fault Tolerance in Service-Oriented Architectures

[Fagg2004] G. E. Fagg and J. J. Dongarra, "Building and using a Fault Tolerant MPI implementation," *International Journal of High Performance Applications and Supercomputing,* 2004.

[Felb99] Felber, P., et al. Failure Detectors as First Class Objects, *Proceedings of the International Symposium on Distributed Objects and Applications* (DOA'99), IEEE Computer Society Press, September 5-7, 1999, p. 132.

[Frol2000] Frolund, S. et al. "Building Dependable Internet Services with E-speak", *Proceedings of the Workshop on Dependability of IP Applications, Platforms, and Networks*, June 26, 2000, held in conjunction with the 2000 International Conference on Dependable Systems and Networks, IEEE Computer Society, and also available as Hewlett-Packard Labs Technical Report 2000-78.

[Huan2006] Huang, S., Wang, K., Li, S., and Chen, M., "Estimating web services reliability: a semantic approach," *International Journal of Information Systems and Change Management*, Volume 1, Number 1, pp. 82-98, 2006

[Kano2005] Kanamori, M.; Saeki, Y.; Shimojo, S., "An information sharing and analysis method for detection of abnormal behavior in computational grids," *Proceedings of the Eighth International Conference on High-Performance Computing in Asia-Pacific Region*, November 2005.

[Kim2005] Kim, H.S.; Yeom, H.Y., “A user-transparent recoverable file system for distributed computing environment,” *Proceedings of Challenges of Large Applications in Distributed Environments, 2005 (CLADE 2005)*, July 2005, pp. 45- 53.

[LauXXXX] Lau, F., Ho, R., and. Wang, C., “Grid Computing in Hong Kong: Research and Development,”

[Li2006b] Li, Q., Xu, M., and Lui, F., “A Grid Troubleshooting Method Based on Fuzzy Event,” *Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC 2006)*, Changsha, China, October 2006. pp. 374-379

[Rado2001] Radoslavov, P., Govindam, R.,and Estrin, D. "Topology-Informed Internet Replica Placement," in *Sixth International Workshop on Web Caching and Content Distribution*, 2001.

[Weis2000] Weissman, J. "Fault Tolerant Wide-Area Parallel Computing," in *IEEE Workshop on Fault-Tolerant Parallel and Distributed Systems*, International Parallel and Distributed Processing Symposium IPDPS, May, 2000.

[Wu2005] R. Wu, A. Chien, M. Hiltunen, R. Schlichting, S. Sen, “A High Performance Configurable Transport Protocol for Grid Computing”, CCGrid 2005