



Project no. FP6-004265

CoreGRID

European Research Network on Foundations, Software Infrastructures and Applications for large scale distributed, GRID and Peer-to-Peer Technologies

Network of Excellence

GRID-based Systems for solving complex problems

D.SA.01 – Roadmap version 1 on System Architecture

Due date of deliverable: February 28, 2005

Actual submission date: April 15, 2005

Start date of project: 1 September 2004

Duration: 48 months

Organisation name of lead contractor for this deliverable: Zuse Institute Berlin (ZIB)

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	PU

Keyword List:

Table of content

1. Executive Summary	4
2. Introduction	4
Context	4
Problem(s)	4
Objectives	5
Tasks	5
Drivers	6
3. Positioning	6
State of the art.....	6
State of the art – Scalability	6
State of the art - Adaptability	7
State of the art - Dependability	7
Extended Context.....	8
4. Vision, Strategy and Roadmap.....	8
Vision and Scenarios (end-users, technologies, computer science)	8
Application Scenarios	9
General Aspects	10
Scalability.....	11
Adaptability.....	12
Dependability.....	12
Strategy	13
Roadmap	14
Phases of the roadmap	14
Definition of capabilities and requirements of future GRIDs (Task 4.1).....	15
A Common Experimental and Benchmarking Environment (Task 4.2)	15
Scalability (Task 4.3).....	15
Scalable Grid Services	15
Scalable Storage and Publishing Systems.....	15
Resource Discovery and Searching	16
Resource Access and Service Provision	16
Scalability vs. Reliability	17
Scalability vs. Consistency	17
Scalability in Insecure World	17
Multicast Services	17
Advance Resource Location Mechanisms	17
Distributed Catalogue to Store Metadata	17
Recommended focus areas within the domain of scalability.....	18
Research projects related to scalability	19
Adaptability (Task 4.4).....	21
1 - Modelling and Prediction of Demand/Capacity.....	21
2 - Adaptive Workflows for System Management.....	21
3 - Composition of Web Services.....	22
4 - Adaptive Communication Layers in Grid Systems	22

Research projects related to adaptability	23
Dependability (Task 4.5)	25
1- Failure Detection and Diagnosis	25
2- Checkpointing-and-Recovery	25
3- Fault-Tolerant MPI	26
4- Dependability for Data Grids	27
5- Fault-tolerant Global Computing	27
Recommended focus areas within the domain of dependability	27
Research projects related to dependability	29
Integration of the proposed methods (Task 4.6)	31
Mechanisms	31
Future steps	32
5. Links with other CoreGRID scientific workpackages	32
6. References	33
6. Participants	35

1. Executive Summary

This document is the first version of the roadmap of the CoreGRID Virtual Institute on System Architecture. This version of the roadmap is to be delivered as D.SA.01 due February 2005. This roadmap will be expanded and refined in its second version D.SA.03 due M18 (February 2006). In particular, the next version will address the second part of the project.

The purpose of the CoreGRID Network of Excellence is strengthening and advancing scientific and technological excellence in the area of GRID and Peer-to-Peer computing. The technical work program of CoreGRID is structured into six complementary research areas, and is conducted by corresponding Virtual Research Institutes. The Virtual Institute on System Architecture (CoreGRID Work Package 4) focuses on the architectural principles of Grid applications and infrastructure that meet the mandatory properties on Next Generation Grids. Grid architecture is one of the cornerstones for successful research, development and proliferation of Grid computing as it is supposed to define basic building blocks and their major interfaces and communication mechanisms. The main objective of the Virtual Institute is to exploit the synergy of research projects conducted by partners in the Virtual Institute. By exploiting the cooperation between partners, we aim at creating a critical mass of participants that will increase the cumulative efficiency of research. The additional objective is to continuously identify the gaps in research and provide the feedback to partners enabling them to adjust their own research directions and to fork off new research projects. The partners of the Virtual Institute conduct research in three main directions: scalability, adaptability and dependability of Grid architectures and Grid services. The partners contribute their expertise and are committed to strengthen the collaboration beyond the NoE.

2. Introduction

Context

The purpose of the CoreGRID Network of Excellence is strengthening and advancing scientific and technological excellence in the area of GRID and Peer-to-Peer computing. The technical work program of CoreGRID is structured into six complementary research areas that have been selected on the basis of their strategic importance as recognised by the leading European expertise. Research in each of the areas is conducted by Virtual Research Institutes that form together the CoreGRID's Research Laboratory. This document describes the research roadmap of the Virtual Institute on System Architecture (CoreGRID Work Package 4). This Virtual Institute coordinates research and promotes propagation of scientific expertise between its members. Grid architecture is one of the cornerstones for successful research, development and proliferation of Grid computing as it is supposed to define basic building blocks and their major interfaces and communication mechanisms. Grid architecture has to meet certain demands that are novel for the computing science. Because of the central role of architecture in Grid computing research, this Virtual Institute on System Architecture will have close connections with other CoreGRID Virtual Institutes.

Problem(s)

Every Grid system, as every non-trivial system in general, features an architecture – the definition of its individual building blocks, interconnections between them, and the general principles how blocks are composed and interoperate. Grid systems will be composed from many components which number and diversity will only increase over time. The vision of “invisible Grid”, as pioneered by experts in the report “Next Generation Grid(s), European Grid Research 2005-2010” (also known as the NGG report), states that the complexity of the Grid is to be fully hidden from users and developers through complete virtualization of Grid resources. Different Grid systems can be composed from the same reusable, usually pre-existing resources and components. Virtualization of resources demands certain uniformity and standardization, which further increases the role of architecture in the Grid. The scale, dynamism and openness of the Grid, together with demands on its reliability, security and manageability, poses new, unique challenges on software architecture. While it is not realistic to initiate, coordinate and integrate research on all aspects of Grid architecture, we envision a vast improvement of

architectural designs of future Grids by focusing on three particular key aspects: scalability, adaptability and dependability.

Objectives

This Virtual Institute focuses on the architectural principles of Grid applications and infrastructure that meet the mandatory properties on Next Generation Grids. The main objective of the Virtual Institute is to exploit the synergy of research projects conducted by partners in the Virtual Institute. By exploiting the cooperation between partners, we aim at creating a critical mass of participants that will increase the cumulative efficiency of research. The additional objective is to continuously identify the gaps in research and provide the feedback to partners enabling them to adjust their own research directions and to fork off new research projects. The partners of the Virtual Institute conduct research in three main directions: scalability, adaptability and dependability of Grid architectures and Grid services. These research directions address the following mandatory architectural properties of Next Generation Grids as identified by NGG reports: simplicity, resilience, scalability of services, and straightforward administration and configuration management.

Tasks

Next Generation Grids will be open, large-scale, pervasive and heterogeneous, and will have to deal with diverse types of resources. Yet, in order to exploit Grid's full potential, Grids have to be simple, transparent, reliable, persistent, secure, and easily configurable and manageable. This is clearly a novel combination of demands on software architecture, and guiding principles for building such systems are not known. We can expect all kinds of applications on the Grid as it becomes more and more pervasive, but the exact scope of applicability of Grid, together with specific demands those future applications impose on it, remain to be seen. It is apparently unrealistic to attempt to derive generic architectural principles in this situation.

This Virtual Institute brings together research projects that focus on specific deficiencies of present day Grid systems, and try to solve the problems while keeping in mind the vision of Next Generation Grid and its mandatory architectural properties. We believe this process, with the involvement of CoreGRID Network of Excellence, will collect experience, generalize it, and eventually distil generic architectural principles.

Three groups of deficiencies, namely scalability, adaptability and dependability, are targeted in the following three main tasks of the Virtual Institute.

The '**scalability**' task comprises partners that study the applicability of peer-to-peer techniques to generic Grid services that will certainly benefit from good scalability properties:

- Scalable Storage and Publishing Systems
- Resource Discovery and Searching
- Resource Access and Service Provision
- Multicast Services
- Advance Resource Location Mechanisms
- Distributed Catalogue to Store Metadata.

In addition to these specific services, some partners also study the following properties of scalable, Grid-enabled p2p-based architectures:

- Scalability vs. Reliability
- Scalability vs. Consistency
- Scalability in Insecure World.

The '**adaptability**' task comprises partners that have the following research issues on their agenda:

- Composition of Grid Services
- Modelling and Prediction of Demand/Capacity
- Adaptive Workflows for System Management
- Adaptation Mechanisms of Grid Services.

The '**dependability**' task comprises partners that have the following research issues on their agenda:

- Failure Detection and Diagnosis
- Checkpointing-and-Recovery

- Fault-Tolerant MPI
- Dependability for Data Grids
- Fault-tolerant Global Computing.

Additional tasks are devoted to strengthening of the integration through the following activities:

- analysis of current state, future requirements and scenarios of future Grid systems
- adaptation of the CoreGRID testbed to the specific requirements of the Virtual Institute
- reconciliation of different interdependent technical approaches by partners of this Virtual Institute.

Drivers

Architectural issues pertain nearly all aspects of GRID systems. Therefore, architectural research also suffers substantial fragmentation as being an inherent part of each detailed solution. The need for integration of these architectural approaches is expected to act as a motivating and influencing force of this Virtual Institute. The driving technical aspects tackled in this Virtual Institute will come from the research domains of scalability, adaptability and dependability.

The most influential factors of the work in WP4 will be:

- *User Requirements and Application Scenarios*: architectures of future GRID system must be fitted closely to the arising requirements in terms of scale, functionality and usability.
- *Legacy architectural solutions*: new architectural models must be developed under consideration of existing architectural approaches in order to facilitate interoperability, solution migration and user acceptance.
- *Research community*: different approaches and perspectives on the problems of scalability, adaptability and dependability must be analysed, linked and possibly consolidated within the research community.
- *Economic forces*: costs of system operation and management, cost of system failures and also overhead caused by scalability deficiencies are substantial parts of the total cost of ownership of GRID systems, and must be taken into account in the architectural research.

3. Positioning

State of the art

We present in the following a snapshot of the current state of technology in the Grid architecture research with specific focus on scalability, adaptability and dependability.

State of the art – Scalability

The concept of the Grid has evolved significantly in the past years and made substantial contributions to standards resulting in Open Grid Services Architecture (OGSA), an architecture where heterogeneity is no longer a crucial problem. One of the main challenges regarding the Grid is self-organization, which can be achieved by using Peer-to-Peer techniques. While the Grid is currently distributed and semi-decentralized, individual services are still highly centralized, static, and not self-organizing. If this trend continues, substantial amount of administration and management will be required to setup and maintain a Grid infrastructure, which is an obstacle if the Grid is going to be ubiquitously deployed. Furthermore, individual services will not be scalable and fault-tolerant. Therefore, it is important for services to become scalable, decentralized, and, most importantly, self-organizing.

Grid computing infrastructure allows on-demand construction of virtual organizations (VO), each one spreads across multiple physical organizations, sharing resources such as storage, capacity, and computational power. Resources are accessed on demand by the members of each VO through appropriate Grid services. The process of accessing or offering resources in a Grid infrastructure can be decomposed in several distinct steps such as registering to the VO (i.e., obtaining a certificate), publishing offered resources, looking up and accessing offered resources. Grid computing currently offers a standard framework to deploy and run distributed applications. Paradoxically, its management mechanisms, such as resources access and discovery, are still mainly centralized. There is an opportunity to resolve such scalability issues by incorporating mechanisms from Peer-to-Peer computing.

State of the art - Adaptability

Adaptability is understood as the ability of automatic adaptation of Grid systems to the changes in internal and the external system state. Current research covers several aspects, including web service composition, self-management of resources, and modelling and prediction of demand (as a tool for better scheduling and capacity planning). Further special issues concern how the middleware can take advantage of the communication layer without knowing anything about it before actual execution, and how one can adapt the application running to the environment in order to maximize its performance.

In the context of Web Services, available information about status of resources is restricted to static service description (offered functionality, performance, cost of use, possible restrictions etc.) or the dynamic resource status (current load, open connections, availability etc). However, adaptability is only possible if the overall state information about the Grid (interdependencies, current health of the underlying infrastructure etc.) is fully made available and utilized. When applications are composed of components and services, automatic reconfiguration of these compositions require careful manipulation of the overall state information. While there exist schemes for collection and analysis of overall system state, these are restricted to the domain of homogeneous clusters or predictable execution environments and are not appropriate for Grid systems.

Contributions to adaptability and self-management of resources within Grids, services and systems have been made under several industrial research initiatives and within traditional university research. The best known example is the *Autonomic Computing* [1] initiative which comprised efforts to refine already known management techniques, as well as efforts to develop new methods. The initiative focuses mainly on the self-management of systems on hardware- and operating system level. Examples of research in this field include management of heterogeneous networks [2], prediction of resource demand and automatic adaptation [3], and automatic recognition of resource models [4].

Utility Computing refers to the idea of outsourcing the data centres of enterprises to external providers of computational, storage and service resources. Since the management cost is the decisive factor of profitability, a lot of effort in this area is made towards automation. The most prominent initiatives are driven by Hewlett-Packard in their projects "Always On Infrastructure" and "Utility Data Centre" [5]; also IBM is active in this area. The research here has been focused on automatic configuration of virtualized data centres, resource sharing under Service Level Agreements [6], scalable and distributed control infrastructures [7].

Another noteworthy research in the field of self-management originates from policy-based network management [2]. The paradigm is to control the behaviour of networks with a set of abstract (high level) directives (policies), which are first verified, evaluated and then translated into concrete actions for the lower network layers. As mentioned above, Grid computing does not currently feature truly adaptive or self-managing elements. However, there are some interesting projects which target these issues. For example, P-GRADE [8] is a workflow-based resource management system which can automatically react to changing conditions, and NorduGrid a production Grid which uses adaptive, decentralized brokering of resources.

Currently, no major Grid middleware utilizes tools for modelling and prediction of demand of applications, or complementary, of capacity of resources. This situation is partially due to the lack of implementations of suitable algorithms. While there exist frameworks for short-term prediction and scheduling support [9] for individual servers, they are not appropriate for long-term prediction, anomaly detection, or exploiting the correlations between the applications in a cluster. A further impediment comes from the fact that in current Grid architectures, tools for demand modelling and prediction have been not foreseen as a part of the middleware, and so other components such as schedulers etc. cannot take advantage of them. Finally, there is lack of a suitable standard for model description and exchange.

State of the art - Dependability

One of the important issues to be solved in Grid Computing infrastructures is the support for fault-tolerance. Due to the complexity and heterogeneity of Grid elements, there is a need to devise new fault-tolerance mechanisms that should be able to adapt to the scalable and dynamic environments of the Grid.

The field of dependability has gained notable advances in the past decades in the areas of distributed computing, parallel processing and clusters of computers. However, the fault-tolerant schemes that have been devised for those environments are mainly targeted to small-scale systems. The literature is full of papers about failure-detection and diagnosis, checkpoint-recovery, replication, group communication, reconfiguration, amongst other

techniques [10] that have been proved suitable for small and medium scale installations, mostly characterized by a homogenous and stable environment.

However, with the advent of Grid computing there is a clear need to adapt the fault-tolerance schemes to scalable, dynamic and wide-area environments that may comprise heterogeneous modules and different Grid middleware. This represents some interesting research challenges that should require extensive work from the European research community in this field.

Grid middleware itself is not reliable. For instance, the Globus Grid service container does not provide any means to achieve that reliability. This has to be dealt with by each particular service. The middleware that runs inside a cluster of the Grid may also lack support for fault-tolerance. As an example, MPI-based implementations are usually unreliable. In order to end successfully an MPI application all the involved units must run smoothly and any unexpected individual failure results in an application breakdown. Some fault-tolerance schemes for MPI have been proposed in the literature and there is still some work to do in this field. Similar problems can also be considered for job-batching middleware: if there is a breakdown, some mechanisms should exist to keep the partial results either to restart the job from an advanced point or to allow the analysis that eventually would lead to identification of failure causes. All these modules of middleware should provide different mechanisms for fault-tolerance. The most difficult goal is to make these mechanisms more tightly integrated in order to provide full-dependability at the application level.

The survey study presented in [11] stated that the two most frequent causes of failures in Grids are due to configuration problems and middleware failures, followed by application errors and hardware outages. Another curious fact was that solutions for failure-handling are mostly application-dependent, which have been requiring a large effort from application programmers to diagnose and provide error-recovery code able to resume the application after the occurrence of a failure. The first fact reveals the immaturity of Grid middleware that should be made more robust. The second fact is a more important one: nowadays the diagnosis and treatment of failures is too much application-programmer dependent. In next-generation of Grid middleware error detection, error recovery and failure-treatment should be assured in a more automatic and transparent way. To achieve this goal there is a clear need for extra research work in this area of dependability to improve the current state of Grid computing.

Extended Context

Beyond the state of the art in research on GRID architectures stated above, there is a variety of aspects in system architecture related to other research domains or Virtual Institutes. Many of them cannot be clearly classified as pure system architecture issues due to the all-pertaining nature of architecture questions. Examples are scheduling problems, a domain closely related to architectural design, which are the focus of WP6, or component-based programming models which are handled in WP3. While the researchers in WP4 will keep track of these aspects within their individual projects, we prefer to retain our focus on the aspects of scalability, adaptability, and dependability.

4. Vision, Strategy and Roadmap

Vision and Scenarios (end-users, technologies, computer science)

This Virtual Institute is devoted to the System Architecture of future Grid systems, with a particular focus on the domains of scalability, adaptability and dependability. A foundation of our vision is an examination and prognosis of requirements, capacities and challenges of the future Grid systems in the context of current and future application scenarios. The sources of such an analysis are the NGG1 and NGG2 Expert Group Reports and the contributions of the partners concerning particular domains. It is important to note that the vision attempts to cover the whole area of Grid architecture; it is not limited to goals that can be achieved by partners of the Virtual Institute, and therefore should not be understood as a research programme of the Virtual Institute. However, it facilitates for each partner and for the Virtual Institute as a whole the selection of the research directions, prioritizing of competing research problems, and focusing on the most pressing issues. The vision

also helps to recognize fields not covered by any of the partners, providing guidance in proposing new research projects and in applications for funding.

Application Scenarios

Some possible application scenarios envisioned by the NGG reports include a Crisis Management Scenario, where a natural or human-caused disaster has to be handled by mobile workers (police, fire fighters, environmental monitors, military, etc.). Workers have to collaborate in real-time, and also have real-time access to information, knowledge in order to improve their decision-making process. The proactive PDA (Personal Digital Assistant) Scenario addresses the issues of efficient provisioning of correct, appropriate and timely information to humans. A PDA can act proactively to locate and retrieve information depending on its current user preferences, location, schedule, and aspects of security and trust. Both scenarios put significant demands on information processing capabilities and computational infrastructure, such as utilization to local communication infrastructure; access to remote databases; synthesis of information from different sources, pre-emptive and on-demand; modelling and prediction capabilities. Other scenarios include industrial applications such as CFD simulations, scientific data analysis (DataGrid), collaborative environments.

The NGG1 and NGG2 Expert Group Reports have identified a set of properties of Grid systems required by the future application scenarios, such as described above. With respect to the scalability (S), adaptability (A) and dependability (D) issues of Grid system architecture, the relevant properties are the following ones:

- **[S]** *pervasive, with mobility as the cornerstone enhanced with more advanced pervasive computing facilities*: pervasive computing, such as in scenarios above, will necessarily be large-scale and thus will have to face the scalability challenges;
- **[A, S]** *self-managing with the ability to handle highly dynamic and unpredictable configuration of demanders and suppliers*: in the scenarios above the Grid services are created and connected on demand, and have to automatically adapt to the environment as there is little if any possibility for managing them by humans, yet they have to be dependable in order to be generally useful;
- **[D, A]** *resilient with the ability to handle highly dynamic and unpredictable configuration of the network connecting the computing nodes*: similarly to self-management demands above, and in particular in the case of mobile clients and service providers, Grid services in the outlined application scenarios have to be flexible and adaptable to the underlying networking infrastructure;
- **[A, D]** *flexible to handle various types of computing nodes and highly dynamic distribution of computation tasks among involved resources*: dynamically composed Grid services in the application scenarios above will necessarily be hosted on different types of computing hardware not known in advance, for which the Grid architecture has to be adaptable yet provide a required level of dependability; **[D, A]** *resilient with the ability to handle intermittent connectivity and associated synchronisation of information sources*: along with the resilience in handling potentially dynamic network configurations in the application scenarios outlined above, individual higher-level information services and connectivity to them can be temporarily or permanently unavailable, yet the Grid applications as a whole must be adaptable to this failures and provide dependable services.

For Grid systems exemplified by the application scenarios above, additional and/or more specific properties concerning the OS-related layer (*Grid Foundations*) have been identified in NGG2 as the following ones (only architecture-related are stated):

- **[A, D]** self-adaptive, self-healing , self-managing and self-reconfiguring;
- **[S]** scale-independent;
- **[A]** open for interoperation – cooperating operating systems or components;
- extended with the concept that the OS should be modular so that minimal configurations can be used without sacrificing interoperability;
- a clear and open interface for Grids Foundations Middleware (OS-related layer) to Grids Service Middleware;
- extended in the sense of context-aware geographically, temporally and role-based;
- re-use of standards in operating system components to encourage interoperability and to provide a consistent interface to Grids foundations;
- appropriate power consumption and code-size for the Grids entity (e.g. nano device).

While the above requirements cover the vision of future Grid systems from the perspective of application scenarios, functionality, and users, the domain experts within the Virtual Institute have contributed a

complementary yet more detailed vision, with special focus revolving around the aspects of scalability, adaptability, and dependability, as presented in the following sections.

General Aspects

For providing scalable implementation and deployment of future generation Grids it is necessary to adopt novel techniques such as Peer-to-Peer computing, agent-based software architectures and usage of semantic information and ontologies for representing state of Grid services and semantic-rich inter-service communication. Thus here we review some of the current and future technologies that will impact the architecture, the computational model and applications of future Grids. The architecture of the next-generation Grid will be derived considering both the technologies and methodologies that most probably will impact (and will integrate into) current Grid solutions, and the major requirements emerging in many computer science fields such as mobile and pervasive computing, ontology-based reasoning, Peer-to-Peer, and knowledge management, and agent-based software architectures. A key aspect that will characterize next-generation Grids more and more is the systematic adoption of metadata to describe resources, services, data sources over the Grid and the use and efficient transmission of such metadata to enhance, and possibly automate, processes such as service discovery and negotiation, application composition, information extraction, and knowledge discovery.

In spite of current practices and thoughts, the Grid and P2P models share several features and have more in common than we perhaps generally recognize. An integration between the two computing models could bring many benefits in designing future scalable Grids. As Grids used for complex applications include a large number of nodes, we should decentralize their functionalities to avoid bottlenecks. The P2P model could thus help to ensure grid scalability: designers could use the P2P philosophy and techniques to implement non-hierarchical decentralized Grid systems.

From the Grid point of view the main interesting aspects of P2P are scalability, self-configuration, autonomic management, dynamic resource discovery, and fault tolerance. On the other hand, current P2P systems are lacking in some fields that are crucial to the deployment of production-quality services, such as QoS negotiation, persistent and multi-purpose service infrastructure, complex services (beyond the simple file-sharing), robustness, performance and security. P2P and Grid communities can benefit each other sharing their key technologies.

The scale and decentralization of the Grid implies that Grid clients and services can possess only partial information about the Grid. Moreover, access to some information can be restricted due to security. Finally, the Grid is volatile, thus any information accumulated about it is inherently imprecise. The agent-based approach (e.g. [55, 56]) is generally considered to facilitate system development for such environments [53, 54], and its virtue for the Grid is well recognized [52, 51]. Agent-based software engineering can be used for building decentralized intelligent applications [54] on the Grid. It can also be used for enhancing the Grid infrastructure itself, providing e.g. knowledge-based information services, semantic service description etc. [51] that will utilize and complement the present day Grid infrastructure [52]. The integration in Grid systems of large-scale mobility and pervasive computing functionality poses new challenges and requirements to the underlying architecture:

- Ontology-based semantic modelling of user preferences, devices characteristics, and context enables reasoning about user's needs and the required adaptation of services.
- Adaptable and composable software infrastructure, able to find, adapt, and deliver appropriate applications (services) to the user's computing environment (devices) on the basis of context. To execute a user task the computing platform should dynamically find and compose the appropriate components and services, and, once instantiated, the application may need to move between devices and environments.

The main (new) requirements of future Grids are:

- knowledge discovery and knowledge management functionalities, for both users' needs (intelligent exploration of data, etc.) and system management;
- semantic modelling of users' tasks/needs, Grid services, data sources, computing devices (from ambient sensors to high-performance computers), to offer high level services and dynamic service location and composition;
- pervasive and ubiquitous computing, through environment/context awareness and adaptation;
- advanced forms of collaboration, through dynamic formation of virtual organizations;
- self-configuration, autonomic management, dynamic resource discovery and fault tolerance.

On the basis of those considerations, we envision a general architecture of a next-generation Grid as shown in Figure 1.

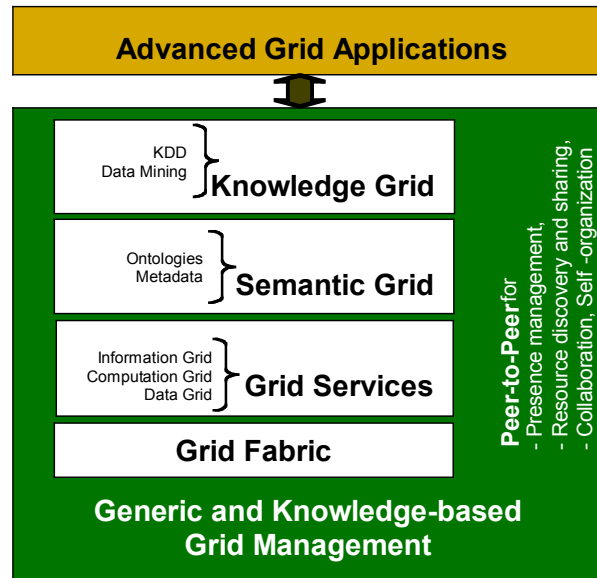


Figure 1: Main layers of the next-generation Grid

Network-centered OS for the Grid

It is becoming increasingly clear that existing Grid basic middleware such as e.g. the Globus Toolkit do not provide the necessary support for simple and reliable application construction and execution in Grid environments. The main problems with those are the client-server programming model which fails to scale in Grid environment, and the lack of support for self-healing and self-management. The underlying operating systems control a single computer and manage its resources exclusively, which is at odds with the concept of the Grid. One needs network-centric operating systems, or toolkits on top of existing operating systems where OS cannot be replaced, that provides any device operating within a GRID environment an “Grid surface” that is easy to use, consistent, performant and stable with features for automated assurance of availability and performance. Such an OS would facilitate the vision of “invisible Grid. An ideal solution for such an operating system is a modular operating system with dynamic loading of only those components required for the purpose and the device.

Scalability

In order to be able to efficiently manage Grids composed of thousands of nodes the Grid functionality needs to be decentralized to avoid bottlenecks. The Peer-to-Peer model could help achieve Grid scalability. Using the Peer-to-Peer philosophy and techniques, a Grid should be devised and designed as a non-hierarchical decentralized system. Until today several research models have been proposed, however no system, neither a viable prototype is available for large scale Grids. Several challenges have to be faced to implement scalable Grid systems and services, including the definition of fully decentralized models, the design and evaluation of scalable systems and applications, the implementation of lightweight Grid services for large scale pervasive Grids, and the development of naming systems for mobile Grids.

As Grid architectures keep on growing, middleware tools will have to adapt to the large number of resources. Tools providing resource access services such as monitoring tools and resource managers do not currently address reliability issues that occur on large scale architectures. As some parts (nodes, links, etc.) of the Grid fail, the system as a whole should not be compromised and mechanisms have to be provided to automatically react to these spurious changes. The services themselves will have to adapt to the larger scale: access to resources and data diffusion will have to be performed efficiently using work distribution and balancing to take advantage of distributed computing on the Grid. Service provision as well will be deeply affected by the larger scale. As users will not be able to manage by themselves the mass of information provided by the Grid, new interfaces will have to be developed that will provide automatic mechanisms for information synthesis and events handling.

Finally, future Grids will require a scalable, reliable and distributed catalog to store arbitrary metadata. In contrast to existing catalog solutions, the future solutions should allow to store arbitrary metadata and will be not restricted to a statically set-up database schema. This will make it easily usable and adaptable without the need to reconfigure installed systems, if new requirements on the data structures are raised. This often happens during

the project development or when new communities want to use the system. Furthermore, future Grid architecture needs a system for transparent access to files and replicas. The main drawbacks of traditional Grid replica systems is that they are not based on scalable software architectures and provide only the fundamental services of mapping logical file names to physical files names (replica locations).

Adaptability

Adaptability for resource management should provide mechanisms for automated adaptation and reconfiguration of the GRIDs on all hierarchy levels. Such mechanisms will include modelling of the state of the GRID and its analysis together with decision taking.

A basis for adaptive and “autonomic” management of resources within the Grid systems is an accurate modelling and analysis of the current state of the participating systems, the prevailing conditions, and the possible “Workflows”, i.e. sequences of actions. The research activities for this task should include modelling of demands/capabilities as well as modelling of workflows (including failure situations) typically occurring in the GRIDs. The results will lead to an appropriate modelling and description standard (e.g. based on CIM) for both aspects, and a set of methods for formal verification of the models, especially the workflow part. Moreover, the architectural components for management, dissemination and possibly semi-automatic creation of the models should be defined and implemented.

The core of adaptability for resource management contains mechanisms for decisions taken automatically by the system. This is a very challenging area, so that realistically only results in some selected application domains can be expected. These include for example design of a “reasoning engine” and “patterns” for adaptive behaviour, including optimization and actions as they would have been taken by human operators. The decision making will apply mainly to resource management and reactions to undesirable situations. A further topic to be covered is “Intelligent Execution”, i.e. execution mechanisms which ensure that an issued decision will be implemented despite of potential problems or failures in the lower layers. Also here “patterns” and “rules” for a variety of situations must be specified, and a feedback-based execution mechanisms.

A particular aspect of the adaptability is the *modelling and prediction of demands* (or more generally, dynamic characteristics such as free resource capacities). We envision demand characterization and prediction processes as a part of an architectural framework for automated management of applications and resources in Grid systems. Their output can be used by external components to tune and manage individual applications, e.g. in a demand optimizing module. In a distributed environment, demand prediction capabilities enable *proactive* resource management. In contrast to classical reactive resource management, proactive management schedules the tasks or applications to resources in anticipation of the expected demand surges or idle times. As a result, the number of hot-spots is reduced and the degree of resource utilization increases.

In addition to short-term scheduling, tools for demand characterization and prediction might also be used for automated “matching” of applications with complementary demand behaviour for server consolidation - several consumers can use the same resource as long as the respective phases of high and low demand balance each other out.

Further aspects of adaptability concern the *Grid service layer*. Here adaptability requires a software mechanism and an infrastructure to automatically reconfigure the compositions, given that necessary status information about the current status of Grid services/resources is provided by the underlying platform.

Dependability

Long-running applications that will execute in Grid Infrastructures are easily affected by the occurrence of partial failures in some components of the system, provided the increased complexity and the distribution of computing resources. Grid middleware should be instrumented with the support for fault-tolerance techniques to assure the resiliency of applications and the high-availability of crucial Grid Services. Within CoreGRID we will address the key fault-tolerant techniques and will try to solve the main challenges in dependable Grid computing.

Currently there are three main challenges in the study of dependability in Grid:

- *Scalability*: since Grid Computing assumes a much more scalable environment, there is a need to adapt existing mechanisms and techniques for failure-detection, failure-handling and failure-correction. So far, most of the existing fault-tolerance schemes have not been tested in scalable systems and may lack of performance of management capabilities when adapted to those systems.

- *Dynamic*: Grid environments are by nature much more dynamic than dedicated small and medium-size clusters. Fault-tolerance mechanisms should be re-structured in order to allow the adaptability of applications and middleware in the occurrence of partial failures.
- *Heterogeneity*: although there has been a strong effort in the standardization process of Grid middleware, it is clearly foreseeable that Grid Computing may happen in wide-area networks of computing elements that may run different pieces of software, like Globus, LCG, Condor, Nimrod, MPI, gLite, among others [12]. Each of these modules of middleware should provide their own support for fault-tolerance, and on top of that, there is some challenge to integrate the different fault-tolerance mechanisms in a consistent way, able to support the dependability and robustness of Grid applications.

Together with this list of challenges there is now some trend towards the integration of Grid computing and Peer-to-Peer. As was interestingly stated by Ian Foster in [13] the Grid addresses infrastructure but lacks support for fault-tolerance, while Peer-to-Peer addresses failure but lack support from a standard infrastructure. The integration of these two paradigms may yield a more fault-tolerant system taking into account issues like scalability, adaptability to failures and self-configuration at the level of protocols and middleware.

Similar integration should be done with other application paradigms and computing infrastructures. In this sense, it is important to keep a close attention to advances in dependability in somewhat separated fields like cluster computing, grid of clusters, global (Internet-based) computing, data dissemination Grids, P2P systems and Web-Services. In any of these fields there are several open issues and challenges to be solved, but the integration of fault-tolerance schemes from these different paradigms and systems would be the main challenge of all.

Strategy

The principal mechanics for bringing the research on architectures of Grid systems within the Virtual Institute closer to the above vision has the following components:

1. **exploiting the synergy effects within the portfolio of the research projects** conducted by CoreGRID partners;
2. recurring **analysis of gaps and priorities of partners' research activities in respect to the vision** as a means to offer a research guidance for every partner;
3. recurring **analysis of new requirements, opportunities and technical catalysts in the field of Grid system architecture**, in order to update our vision to the changes in this highly dynamic field;
4. **stipulating new joint research projects** to cover the research areas which are not represented in the current project portfolio, using FP6 instruments such as IPs (Integrated Projects) and STREPs (Specific Targeted Research Projects), as well as institutional and national funding.

Among the above strategy components, exploiting synergy effects has the highest impact on the results of the Virtual Institute. The main tool here is brining the partners together on different levels in order to enhance the research quality, enlarge the spectrum of opinions and last but not least reduce the redundancy, especially in the areas of testbeds and software development. These goals can be achieved by the following means:

- Collaboration between the partners on similar topics on the levels of "scientific articles" and partner projects.
 - o Instruments here are: co-authoring scientific articles and CoreGRID reports; short visits between institutions; longer-term scientific collaborations over the research project duration.
- Collaboration between the partners within each of the domain of scalability, adaptability, dependability.
 - o Instruments here are: internal "adjustment" of each partners research agenda according to the vision; agreements between partners to cover specific aspects within each domain; teleconferences; Virtual Institute level meetings; cooperation within a common testbed.
- Collaboration on the SA Virtual Institute level, and between Virtual Institutes.
 - o Instruments here are: collaboration on updating and feedback on the system architecture vision; reconciliation of the contradictive research approaches by the implementation within the common testbed; Fellowship Programs; Virtual Institute level meetings; "all-together-now" General Assembly meetings.

The above strategy gave rise to the following six tasks of this Virtual Institute:

Task 4.1: Definition of Capabilities and Architectural Components of Scalable, Adaptive and Dependable GRID

Task 4.2: Setup of a Common Experimental and Benchmarking Environment

Task 4.3: Scalable GRID Services

Task 4.4: Mechanisms for Adaptive GRID

Task 4.5: Dependability in GRIDS

Task 4.6: Integration of the Proposed Methods.

Roadmap

Phases of the roadmap

We have identified the following phases for this roadmap which should drive the activities of the partners involved within the Virtual Institute on System Architecture.

Phase 1: This part focuses on gaining a common understanding of System Architecture problems, future requirements and capacities. This is done by sharing partners view on the current state of the Grid architectures, important problems, and the future requirements. Also contributions from external documents such as NGG1 and NGG2 reports are considered as the input. Particular focus lies in the domains of scalability, adaptability, and dependability. This roadmap already contains the initial results of these efforts.

Phase 2: The next phase targets the enabling of the synergy effects between the research activities of the partners. This includes collection of information about partners' current projects, starting or likely projects in the near future, specific expertise of the participating researchers, and research interests. On the basis of this information, cooperation links are identified and possible redundancies (e.g. in development of testbeds and software) are tackled. This roadmap (especially the detailed treatment of the research aspects below) includes the collected data about partner activities at the time of writing, in order to serve as a reference for finding matching scientific collaborators.

Phase 3: At this stage, the actual collaboration is performed on the basis of information gained in the previous phases. This collaboration will take different forms, including

- scientific collaborations on the levels of "article/paper", project, research domain and Virtual Institute;
- short visits between researchers, common tele-conferences, Virtual Institute meetings, organisation of common workshops;
- sharing of software and a common system architecture testbed;
- stipulation of new joint research projects in form of IPs or STREPs.

This cooperation will be guided by the scientific vision of the Virtual Institute, helping the partners to focus on the high-impact aspects of Grid system architecture research in a cooperative way. A detailed description of these aspects is provided below.

Phase 4: This phase comprises the reconciliation and assessment of the scientific methods and results achieved by the partners in respect to the vision of the Virtual Institute. Here scientific approaches from the domains of scalability, adaptability and dependability must be conceptually compared and adjusted to be made compatible with each other, and also with the current established solutions. In this way, a reconciled set of obtained methods and techniques can be assessed in respect to the vision.

During each of these phases, the obtained results will be promoted across all CoreGRID Virtual Institutes (by means of meetings, technical reports, and WP-overlapping membership of partners), and also within the scientific community (through conference and paper publications).

The following sections present the specific problems which are parts of the vision of the Virtual Institute and already are or are about to be addressed by the research of one or more partners. These descriptions thus provide research guidance for each partner, helping with focus and prioritization of the specific problems within system architecture research. Furthermore, the descriptions show which problems are already covered by existing projects, and identify recommended focus areas for future research. This provides a basis for on-going gap analysis and proposing of new projects targeting the research areas beyond the current activities. At the end of each of the domain descriptions, a detailed list of current or starting research projects of the partners is listed in order to identify specific research collaborators. The descriptions are structured according to the domains of scalability, adaptability and dependability (major parts), and the three supporting tasks which concern definition of capabilities and requirements of future GRIDs (Task 4.1), setup of a Common Experimental and

Benchmarking Environment (Task 4.2), and the consolidation of the technical approaches proposed by the partners (Task 4.6).

Definition of capabilities and requirements of future GRIDs (Task 4.1)

The partners involved in WP4 will analyse each of the regarded aspects of the future GRIDs (scalability, adaptability and dependability) in respect to the expected demands, application scenarios and technical possibilities in the future. This periodically repeated process should ensure that the above stated vision is still up-to-date, and especially targets the most pressing problems in the area of architecture. Over the course of the project, this process will facilitate a common understanding of the requirements, behaviours and capabilities of future GRIDs in a degree which could be hardly achieved by each individual partner or even by cooperating groups of them.

A Common Experimental and Benchmarking Environment (Task 4.2)

In order to reinforce the cooperation within WP4 and obtain quantitative results of the proposed approaches and facilitate the integration of the prototypic components, a WP4-specific part of the GRID testbed environment described in Section 6.1.6 of the DoW will be adjusted for simulation and benchmarking of architectural concepts. This testbed will be then adapted to new changes and requirements of the partners, if necessary.

The environment should provide realistic conditions both in the requirements and execution. The existence of the testbed will help to get a fast feedback on the appropriateness of the investigated methods and their interoperability. Furthermore, this testbed will reinforce the integration of results of WP4 with work of other workpackages by acting as an “interface” (due to the technical necessity of adjusting the GRID architectural design to the needs/results of other groups, and vice versa).

Scalability (Task 4.3)

There are several issues that need to be addressed before we can say that Grid systems are truly scalable. The most important of these scalability issues are described in more detail in what follows:

Scalable Grid Services

Individual Grid services are themselves not very scalable. Future Grid services should use existing research from the Peer-to-Peer community to provide scalable efficient Grid services. These services should build on existing research on structured Peer-to-Peer systems, such as Distributed Hash Tables (DHT), to provide, for example, efficient resource discovery, distributed file service, and location independent communication. While research on structured Peer-to-Peer systems provides a solid foundation with guarantees, it often assumes homogeneity of nodes and uniform usage patterns. Hence, concepts developed in unstructured Peer-to-Peer systems could also be used, using a grassroots approach where randomized exchange algorithms are deployed to provide services that automatically load balance in presence of non-uniformity.

Scalable Storage and Publishing Systems

Numerous decentralized and scalable storage systems have been proposed in the field of Peer-to-Peer. Most of them, however, either provide a read-only storage system or give no guarantees in terms of retrieval and stability of inserted files. Grids need a scalable storage system which self-organizes as nodes are added, removed, or when nodes fail. Such a system needs to give guarantees as files should not be stored with best-effort. Rather, a stored file should be retrievable, even in the presence of failures. Furthermore, the storage system needs to provide a way to read and write files in a secure manner.

Scalable storage and publishing systems of mutable data accessed and modified by possibly large amount of users should not rely on centralized resource access and modification control in order to avoid bottleneck effect. These systems must provide users with a distributed access control on data: who can do what (e.g. capability of creation, deletion, read and write); a distributed search mechanism based on data content and/or on associated meta-data; and distributed primitives to create, delete, read and write data. These primitives rely mainly on two concepts: Distributed Certificate Management and Data Multi-Consistency.

Distributed Certificate Management has already been addressed by SDSI/SPKI which are public-key infrastructures similar to X.509-based schemes but differ in the fact that the former does not rely on global name spaces. This key difference simplifies the distributed management of certificate. SPKI/SDSI requested primitives

are close to DHT (Distributed Hash Table) functionalities and some research projects have already studied the possibility to provide SPKI/SDSI mechanisms on top of DHTs. Investigating the integration of such security aware DHTs based on which Grid infrastructure could solve the problem of scalability in terms of servers and users.

Multi-Consistency Support on Mutable Data allows concurrent modification on data. These operations lead to multiple inconsistent versions of the same data. The system should provide convergent mechanism to unify these different versions of the same data.

Resource Discovery and Searching

In the near future, the way of using the Internet is changing. The Internet and resources are available for millions of small mobile terminal devices. Resource can be anything that can be reached or accessed via a network (including - but not limited to - files, documents, and services). Additionally, different network technologies make things a bit more complicated. The convergence of networks, terminal devices, and their user interface raise new demands on resources. One of the key questions with resources is the way of naming them. A major problem facing tools for information resource discovery on the Web is the lack of a mechanism for resource description within the Web's architecture. Usually, names should be unambiguous to be useful. For resources to be searchable or even accessible one must have some kind of a name. For example, there is only one IP address space for each domain name in the Internet. However, there may be (and usually are) many names pointing the same address space. The metadata or attributes in several discovery solutions e.g. VTT's BORIS are presented as keyword=value. Different objects require different kind of keywords. However it is not feasible or even possible to include all keywords that describe an object. An adequate subset of all keywords has to be defined in order to achieve queries that are at the same time fast and specific.

In general the important factors influencing the searching and retrieval of information are the data model, query language and overall system architecture. These are not independent factors and therefore, must be defined taking into consideration their interactions. The overall system architecture fundamentally determines the effectiveness of both search and data access. The query language must match the data model which also determines the possible queries and greatly influences the effectiveness of data access. (i.e. processing a generic self-describing data model might require a lot of resources, on the other hand it might not be possible to support detailed queries using an unstructured data model.)

An important property of the query language is the search types it can support. The search types are defined by the part of data that are significant in a search and the constraints given for that part. The significant part of the data is characterised by the scope of the search, while the possible constraints define the expressiveness. Regarding scope searches can be subject-based or content-based. In the first case constraints can be given only for one distinguished property of the data. In contrast content-based search allows one to select relevant data by any property. A widely used, simple case is name-based search, which is a special case of the subject-based search, where the distinguished property is called name and the only constraint allowed is equality. Typical usage scenarios for resource discovery however imply content-based searches which are more difficult to support effectively in a distributed and scalable way.

Resource Access and Service Provision

Solutions for resource discovery and searching e.g. BORIS do not always provide the necessary functionality for accessing resources. It is purely out of scope of a service discovery mechanism. The basic functionality for a service discovery protocol is to find and locate the resources the user needs. It is up to the user (application) to establish a connection to the actual service. This way the resources (CPU, memory, bandwidth) of a service discovery entity are freed for the mail functionality. It is not feasible to define a resource access method which would work for all imaginable services. This leads to a fact that there is really no reason to limit access methods by defining them on a service discovery level.

Current middleware for resources access and service provision have to be adapted to large volumes of resources and information. Ways to synthesize the information, to constitute automatic the formation of groups among similar tasks, and to handle the events have to be found so that users will be able to fully benefit from the potential offered by the Grid.

The Open Grid Service Architecture (OGSA) model provides an opportunity to integrate Peer-to-Peer techniques in a Grid environment since it offers an open cooperation model that allows Grid entities to be composed in a decentralized way. Although Grid Services are appropriate for implementing loosely coupled Peer-to-Peer

applications, they appear to be inefficient to support an intensive exchange of messages among tightly-coupled peers. In fact, Grid Services operations, as other RPC-like mechanisms, are subject to an invocation overhead that can be significant both in terms of activation time, memory consumption, and bandwidth needs. The number of Grid Service operations that a peer can efficiently manage in a given time interval depends strongly on that overhead. Therefore, one of the main challenges in this field is to design mechanisms for the integration of Peer-to-Peer techniques and Grid services in order to obtain a scalable and reliable global services for resource access.

Scalability vs. Reliability

Node heterogeneity and volatility are major issues in Grid architectures. As such, monitoring tools, as well as, resource managers should not depend on a fixed architecture and should not expect reactivity from all parts of the Grid. In this context, dynamic management of the Grid architecture and error handling policies have to be proposed as new and important functionalities for Grid middleware.

Scalability vs. Consistency

The information in the Grid is not constant. The data describing a specific resource usually changes in time. Some types of information can change quite often (e.g. processor load), while others change rarely (e.g. processor type). We can thus classify the information as static or dynamic based on how often they change. The higher the rate of change (the more dynamic is the information) the more difficult is to handle it in a scalable way. Since frequently changing information becomes stale quickly the tree of servers forming a cache chain cannot be too long, otherwise it either leads to inconsistency or limits the achievable scalability and hinders the efficiency of the information system.

Scalability in Insecure World

The issue of scalability needs to be considered at the level of Grid services. Much of the Grid research has been concerned with providing security by building a wall around the Grid, and by trying to keep malicious nodes outside. But as the Grid scales, this will be insufficient by itself. We need to have locks on our doors, as it is not safe to assume that nodes already within the walls will never be compromised. One way to proceed is to deploy gossiping techniques which disseminate information in a decentralized and scalable fashion whenever a malicious action is detected.

Multicast Services

Efficient data diffusion in Grids and efficient access to a large set of resources will be of importance for interactive Grid tasks such as node administration and management. The mechanisms for distant execution as well as data diffusion will have to make proper use of work distribution, parallelization and bandwidth sharing to scale to thousands of nodes and to an arbitrary Grid topology. One way to proceed should be to develop special tools for low level Grid management.

Advance Resource Location Mechanisms

In a multi-resource server Grid, each site can be a combination of multiple types of resources (CPUs, memory, disks, software, etc.); similarly, the applications submitted by the users are described by multiple resource requirements. In this scenario general models for resource description, based on metadata and ontologies, are a key element. At the same time, the adoption of search and discovery techniques defined in Peer-to-Peer networks can be used to implement scalable mechanisms for complex resource location. For instance, multi-attribute Peer-to-Peer search mechanisms should be investigated.

Distributed Catalogue to Store Metadata

Future Grids will need to employ a catalog to store metadata. This metadata catalog cannot be made scalable by the traditional client-server approach. Instead, the architecture of this catalog will be based on further developed and enhanced structured Peer-to-Peer networks. The component catalog plays a central role for the other components in Grids. It allows storing reliably and querying meta-data for the whole system without the need to know which servers are online or offline, or which servers may fail next. It could build the base for many other components, for example the GDIS, the Dynamic World-Wide-Web, or the Dependable Global Storage component.

To build a scalable Grid, the reliability and availability of files, their placement and access strategy play a vital role. An optimal system will solve these issues transparently by itself. Therefore users may specify desired file availability and the system calculates according to the analytical model how many copies of the file have to be stored on the unreliable servers in the system and where these copies should be placed. If a file is accessed, the system decides by itself which parts of the file are read from which replica for an efficient data access. Having such facilities makes the need for backups obsolete. An envisioned Grid file system by itself ensures that files are available and enough copies exist in the overall system. In contrast to RAID systems or erasure codes, that also increase the availability of files, each replica is a stand-alone file in this approach. This makes the use of an individual replica independent from the whole system infrastructure, which may ease the transition to our system for the end-users.

Recommended focus areas within the domain of scalability

Some interesting and promising research areas are:

- Resource naming based on research conducted in Peer-to-Peer systems
- Resource discovery based on research conducted in Peer-to-Peer systems
- Fully decentralized models of Grid services based on Peer-to-Peer service invocation
- Peer-to-Peer models for the implementation of scalable Grid systems and applications
- Design and implementation of lightweight Grid services for large scale pervasive Grids
- Naming systems for mobile Grids
- Search and discovery techniques based on structured and unstructured Peer-to-Peer models
- Scalable user interfaces for Grids with information synthesis and automatization of Grid tasks
- Reliability and its tolerance to resource variation. Scalable dynamicity of Grid middleware
- Efficient distant execution and data diffusion in heterogeneous large scale architectures.

These recommendations are especially suited to a hierarchical vision of the Grid:

- Simple distributed security infrastructure in Peer-to-Peer systems
- Multi-consistency support on mutable resources in Peer-to-Peer systems
- Handling of dynamic information and its influence on scalability and consistency

Concerning the metadata catalog, further work must address the following issues:

- Enhancement of structured Peer-to-Peer systems to support range queries instead of only single direct lookup
- Efficiently load balancing of the catalog on heterogeneous servers with respect to catalog size, query load, and network traffic, without completely destroying the routing and lookup efficiency of structured Peer-to-Peer networks
- Distribution of the data across the servers in order to efficiently answer queries
- Introduction of redundancy in order to make the distributed catalog reliable

To develop a system for transparent access to files and replicas, the following aspects should become subject of research:

- Study existing [41] or (if necessary) develop new mathematical models to describe the availability of files in Grid and Peer-to-Peer systems
- Allow efficient partial file access to enable multi-source data access
- Study existing or (if necessary) develop new replica placement algorithms to decide where additional replicas should be stored, if replicas should be moved from one location to another, or whether replicas should be deleted.

Research projects related to scalability

The following table describes the research projects of the partners which are related to the domain of scalability. It serves as a reference for identification of potential cooperations and for the analysis of research gaps in respect to the vision.

Institute	Contributions
CETIC	ORAGE: A collaborative hierarchical file sharing environment based on peer-to-peer techniques (DHT, BitTorrent)
CNRS	<p>Ka-tools 2: Software and algorithms for</p> <ul style="list-style-type: none"> • deployment and diffusion of a parallel program • large data transfers to large number of nodes • reconfiguration of (many) distant environments <p>with the goal of building a complete software suite allowing each user to deploy his own environment with efficiency and robustness.</p>
ICS-FORTH	<p>Scalable Peer-to-Peer systems: Design and implementation of mechanisms to improve the scalability of Peer-to-Peer systems through tracing, caching and network reorganization.</p> <ul style="list-style-type: none"> • Providing short-term solutions to address the pressing needs of the Peer-to-Peer infrastructure. • Contributing to the design of the next generation Internet based service infrastructure
INRIA	<p>Grand Large: Design and evaluation of a firewall/NAT friendly P2P system allowing the transparent execution of some cluster software over a set of resources distributed on the Internet.</p> <ul style="list-style-type: none"> • Desktop Grids and Content sharing P2P systems rely on specific mechanisms to manage tasks and file accesses. In the cluster domain, A lot of software tools are available and optimized for these two purposes. The scalability problem of cluster tools is linked with their inability to handle different administration domains without modifying the configuration and policies of domains to be connected. • We propose to design, implement and test a Firewall/NAT friendly P2P layer allowing a transparent extension of cluster software over multiple administration domains. We will test the performance of the resulting system to discover the typical cluster tool features that make them applicable in this context.
KTH/SICS	<p>GES3: Grid Enabled Scalable Self-organizing Services</p> <p>The project works towards Grid-enabled scalable self-organizing services on three different levels:</p> <ul style="list-style-type: none"> • Development of a decentralized DHT-based (P2P) infrastructure providing basic services such as discovery, name-based communication, publish/subscribe mechanisms, and group membership and communication services. • Building a Grid-enabled (i.e. OGSA compliant) application service platform, with services like storage management, searching and indexing, group services, and data distribution. • Finally, building a number of application prototypes to test and validate the infrastructure. <p>The goal of this project is fourfold:</p> <ul style="list-style-type: none"> • Design and implement a practical decentralized DHT-based infrastructure offering a number of basic services. • Design and implement a package of application services on top of the basic infrastructure. • Develop a few pilot applications that make use of the application services. • Validate the applications and hence, indirectly, validate the application services and basic infrastructure, as regards robustness, load-balancing, performance, self-organization (i.e. adaptation as nodes join and leave the infrastructure).
MU	<p>Scalable Grid Monitoring Architecture</p> <p>This project will provide a monitoring architecture that will support:</p> <ul style="list-style-type: none"> * Gathering of data from a large number of different sources, using both active

	<p>and passive sensors.</p> <ul style="list-style-type: none"> * Providing of alternative transport mechanisms, adaptable to data requirements, removing thus a border between monitoring and information systems. * Combining and actively processing monitoring data within the infrastructure, opening a way to adding new processing elements and/or layers on demand. * Internal failure detection and (semi)automatic reconfiguration in presence of infrastructure failures. <p>This architecture must support Grids of very different sizes, so its scalability is of premium importance. In particular this project is working towards:</p> <ul style="list-style-type: none"> • A scalable architecture for Grid information and monitoring systems. • A Framework for combination of different transport protocols and active elements into one infrastructure. • A Framework for combination of data from passive and active monitoring sensors (including possible “ontologies”).
MTA SZTAKI	Flexible Monitoring Infrastructure for large scale Grids: Development of a flexible and extensible monitoring infrastructure for monitoring Grid entities (resources, services, applications) in a scalable and reliable way.
MTA SZTAKI	Information System for Resource Brokering in Large Scale Grids: Development of generic information system architecture to help resource location and brokering. Development and evaluation of different resource classification and selection algorithms and their effect on scalability and descriptiveness.
UCAM	GROWL - Building an existing prototype web service client library for accessing bioinformatics data
UCAM	Multicast Transport for Grid Computing
UCL	Decentralized service architecture: The project is experimenting with the design and construction of an architecture in which services and components are first-class entities. The architecture will be decentralized and allow for long-lived services. Security and fault tolerance are important. The main objective is finding answers to the following questions: <ul style="list-style-type: none"> • What are the fundamental principles underlying the operation of a decentralized architecture based on first-class services and components? • What are the effects on the programming language? • What are the fundamental services that must be provided?
UCL	Collaborative applications for highly dynamic environments: The project attempts to simplify the writing of applications in highly dynamic distributed environments, with PDAs connected through ad hoc networks, connected to Internet nodes with high volatility (they come and go quickly). Points of interest are: <ul style="list-style-type: none"> • How should the dynamicity of the underlying distributed system be made visible to the application programmer? • What are the fundamental abstractions for programming?
UCO	Fault-Tolerant Desktop Grid Computing
UCO	Reliable Data Dissemination in Data Grids
UNICAL	Peer-to-Peer Techniques for Grid computing: An architecture for a scalable Grid information system based on a decentralized P2P protocol and Grid services for a GIS that builds on the OGSA model.
UNICAL	GDIS: Peer-to-Peer techniques Data Integration on Grids: A service-based system for data integration on Grids that uses a P2P model for integrated data schemes and can run queries on distributed databases as contribution to the development of a decentralized p2p protocol and its use on a Grid-service based architecture for a GIS that builds on the OGSA model..
VTT	TIP2P - Open service architecture in peer-to-peer systems <ul style="list-style-type: none"> • Objectives: development of lightweight P2P platform (and features) for users of wireless devices (e.g. cell phones, PDAs) which could locate and communicate with another nodes. • Result: BORIS Object Request Infrastructure is a trader that introduces a new and simple solution to the problems of resource naming and discovery.
VTT	RIBS - IP-based Communications in Industrial, Automation and Building Systems: This project is focused on:

	<ul style="list-style-type: none"> • Design and implementation of a software platform (OS, TCP/IP, Middleware) for testing dynamic real time device systems capabilities for automation systems. • Acquiring competence for embedded communication devices • Examination of real time communication requirements for automation systems.
ZIB	<p>ZIB-DMS - A P2P-based replica management framework</p> <p>ZIB-DMS will be a framework for management of distributed file replica with rich and customisable attributes. The directory for the files is based on a DHT, and so scalable.</p> <p>The main objectives of the project are:</p> <ul style="list-style-type: none"> • Creation of a distributed and scalable replica catalogue for location-transparent data access • Integrated support for customized file attributes. • Scalability studies in general.

Table 1: Research projects related to scalability

Adaptability (Task 4.4)

There are several aspects that need to be addressed to make Grid systems adaptable to internal and external changes. The partners involved in CoreGRID will pursue research in the following topics:

1 - Modelling and Prediction of Demand/Capacity

Modeling and prediction of demand of applications (or of capacity of resources) can bring two benefits into Grids. Firstly, it increases efficiency of resource sharing by allowing long-term capacity planning and by providing support for scheduling. The second application concerns the automatic management of resources.

Sharing of resources in Grids and data centers is motivated by economical reasons.

Among a variety of possible approaches for demand/capacity modeling, several approaches should have particular effectiveness. Methods in the first category try to discover repetitive patterns in the data, such as daily peaks. Examples of the methods are the ARIMA-X-12 used by the U.S. Census Bureau, or a periodicity mining approach which automatically adapts to the changes over time [43].

Another class of modeling approaches uses methods borrowed from econometrics, such as ARIMA-models and Kalman filtering. A significant improvement here comes from using multivariate approaches, which allow for exploiting of correlations in the input data. Finally, data mining classification techniques offer another set of modeling methods. Classifiers such as Naive Bayes or J48 can be easily adapted to model (discretized) demand, and provide computationally efficient predictions.

However, these methods alone will not help to exploit the full benefits of modeling and prediction if they remain detached from Grid and cluster architectures. Instead of considering these tools as optional (and possibly proprietary) parts of Grids installations, they need to be integrated into Grid middleware and possibly even the underlying operating systems. Due to the fact that applications in Grids might be executed on different resources, particular demand is that the *demand models should be attached to the applications* (or their input data), and not to the resources. Finally, a standard for describing the demand models and prediction results is necessary.

2 - Adaptive Workflows for System Management

Lack of automatic management of systems such as Grid infrastructures has negative impact on both the cost of the operations and the dependability. The former issue comes from the rising share of the human operators in the Total Cost of Ownership, and the later is illustrated by the fact that more than 50% of system crashes comes from erroneous (manual) configuration.

A partial remedy to this problem can be achieved by the approach of automated and adaptive generation of workflows composed of system management actions. In the first step, a pool of atomic management actions is specified, including preconditions and effects of each action. Using a declarative (PROLOG-like) specification format allows for deployment of automated planning tools for composing such atomic actions into workflows. A created workflow handles a particular management procedure, such as migration of data between permanent stores, or setting up of a database. The feedback from the execution of each of the atomic steps is collected, and is used for re-planning of the workflow in case of failure.

The approach described above requires specifications of a variety of atomic actions together with their preconditions and effects in order to be usable. Also, specialized, efficient planners must be able to handle large problem instances.

3 - Composition of Web Services

Current work on adaptability is mainly focused from the web-services perspective. In the context of Web Services, available information about status of resources is restricted to describe a service or an application. However, adaptability is only possible if the current status of the services and resources in Grid is fully made available and utilized. When applications are viewed as compositions of components and services, automatically reconfiguring these compositions require careful manipulation of the overall status information.

Given that necessary status information about the current status of Grid services/resources is furnished by the underlying platform, we need a software mechanism and a infrastructure to automatically reconfigure the compositions.

4 - Adaptive Communication Layers in Grid Systems

First, since the Grid is made of different clusters, it is very frequent to have different networking environment, both at the low (myrinet, ethernet) or high (firewalls) levels. As a consequence, most middleware implementations have a conservative approach and rely on plain TCP/IP connections, sacrificing performance to remain portable. Some low level middleware like Ibis [48] provide different communication layer to maximize performance however, the adaptability at runtime is limited. ProActive [50] is a high level middleware which offers different layers for communication as shown in

Figure 2. The future work will focus on how one is being able to dynamically change the communication layer of a running application.

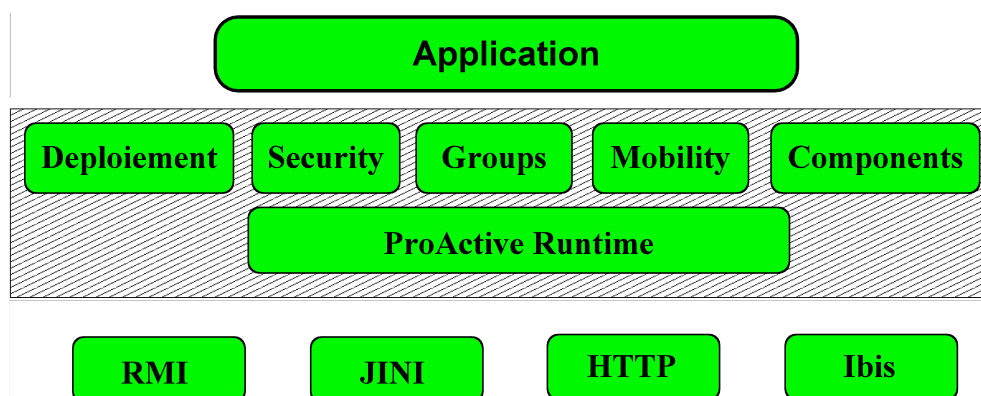


Figure 2: Communications layer in ProActive

Second, we consider a distributed Grid application which can have its network usage modified by changing the granularity of the tasks. When deploying on a Grid, inter-cluster links can become a bottleneck, effectively lowering its performance. The idea would be to have the application checks the environment (through the middleware) and decides how to adapt itself. As an example, all the parts of the application running on the same cluster could perform a high precision computation whereas a lower precision one could be performed by the parts at the frontier of clusters (see

Figure 3). We are currently working on a 3D electromagnetism application, jem3D [49], which could be a good candidate for this work.

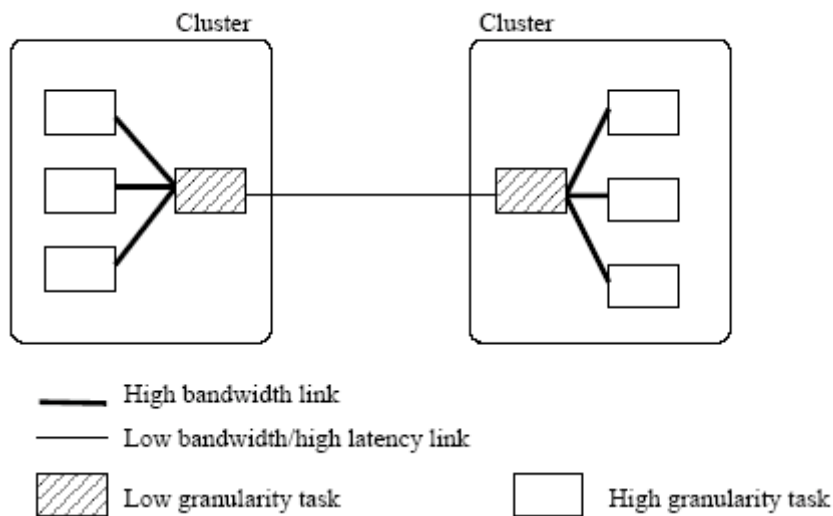


Figure 3 : Jem3D Cluster

Recommended Focus Areas

Modeling and Prediction of Demand/Capacity

To enable the benefits of demand/capacity modeling, we recommend to follow the following steps

- development and refinement of additional methods for demand modeling, and anomaly detection, especially such taking into account the correlations between applications
- standardization of a model/anomaly/prediction description format, e.g. as a part of the CIM framework [44]
- integration of the modeling and prediction algorithms into the Grid middleware as “obligatory” tools.

Adaptive Workflows for System Management

The automated composition of atomic management actions into workflows requires the focus on the following aspects:

- standardization of the atomic action descriptions, and of the resulting workflows
- easier specification of atomic actions with their preconditions and effects, by means of graphic editors, possibly automatic extraction of descriptions from management logs
- development of specialized, efficient planners.

Composition of Web Services

Preferably, more focus is required on maintaining the information pertaining to the overall status of both software and hardware resources of the Grid system of interest. Most specifically, we do need support in terms of collecting, indexing and maintaining information describing the resources, status of them and dependences. Such a collection of information is necessary for successfully assess and construct execution plans or reconfigure the system components.

Research projects related to adaptability

The following table describes the research projects of the partners which are related to the domain of adaptability. It serves as a reference for identification of potential cooperations and for the analysis of research gaps in respect to the vision.

Institute	Contributions
-----------	---------------

CNRS	<p>Cigri - Automatic treatment of tasks and information: The project is building a regional-scale middleware for high-performance multi-parametric applications. It handles submissions, deployment, monitoring and error treatment of a very large number of tasks. Users are provided with synthetic error reports for their application. One solution is already in place, with several restrictions. One of the goals is to provide a less restricted software.</p>
INRIA	<p>OASIS - Inclusion of adaptation mechanisms at various levels in a Grid middleware : This project is designing adaptable strategies for using the best transport protocol for communication between active objects. Several transport protocols for the proActive library to let active objects interoperate have already been realized. The study of adaptive strategies and implementation techniques to be able to dynamically use the most appropriate protocol will be dealt with in the future.</p>
UCL	<p>Decentralized service architecture: The project is experimenting with the design and construction of an architecture in which services and components are first-class entities. The architecture will be decentralized and allow for long-lived services. Security and fault tolerance are important. The main points of interest are :</p> <ul style="list-style-type: none"> • What are the fundamental principles underlying the operation of a decentralized architecture based on first-class services and components? • What are the effects on the programming language? • What are the fundamental services that must be provided?
UCL	<p>Collaborative applications for highly dynamic environments: The project attempts to simplify the writing of applications in highly dynamic distributed environments, with PDAs connected through ad hoc networks, connected to Internet nodes with high volatility (they come and go quickly). Points of interest are:</p> <ul style="list-style-type: none"> • How should the dynamicity of the underlying distributed system be made visible to the application programmer? <p>What are the fundamental abstractions for programming?</p>
UOW	<p>AUTOGRID: Temporal Modelling of Intelligent Grids : The project aims to develop the theoretical foundations for a multi-layer self-organizing generic Grid architecture (AUTOGRID), which will automatically manage reconfiguration of components in Grid systems in a safe and optimal way. Goals are :</p> <ul style="list-style-type: none"> ▪ Automatically generate requests on reconfigurations of components of a system. ▪ Develop an engine to automatically provide optimal solutions. ▪ Provide efficient information flow to enable passing the requests to this engine and the solutions back.
UOW	<p>Automatically Reconfigurable Component Model : The main objective of the project is to derive automatically reconfiguring execution plans for jobs, using the up-to-date state information. Goals are :</p> <ul style="list-style-type: none"> ▪ Develop approximate models for adaptive engine to monitor, assess and to provide a feedback for reconfiguration of execution plans. ▪ Develop approximate models for deriving execution plans. ▪ Use the state information and original constraints (of jobs) to prune the search space.
ZIB	<p>Time series framework for demand modelling : The project's aim is the construction of a framework for modelling, prediction and anomaly detection of demand (or other statistics) of applications or resources. Contains data structures, persistency layers, several algorithms, backtesting and a „real-time advisor“ parts. The main goals are:</p> <ul style="list-style-type: none"> • Development and evaluation of different modelling, prediction, and anomaly detection algorithms. • Creation of an architectural prototype for pluggable „advisers“ for self-management infrastructures in Grids.
ZIB	<p>Feedback-based planning and execution of system management tasks : The project is working on a system which generates sequences and workflows of system management actions from declarative descriptions of actions and targets, and provides a feedback-based execution. Optionally, the system includes a component for automatic deduction of action specifications from examples. The main goals are:</p>

	<ul style="list-style-type: none"> • Creation of a prototype for „feedback loop“ control. • Implementing of a specialized but efficient planner. • Automatic generation of action specification or “rules“ from examples. • Possibly, definition of a specification standard.
--	---

Table 2: Research projects related to adaptability

Dependability (Task 4.5)

There are several issues that need to be addressed to achieve the ultimate goal of Dependable Grid Computing. Within Core-Grid we will be more concentrated in the following five topics:

1- Failure Detection and Diagnosis

The first steps of fault-tolerance are: the detection of failures and the corresponding diagnosis to determine the reason and the nature of the error. These are the cornerstone mechanisms of any fault-tolerance scheme.

An important step in this topic is to establish a failure-model for Grids, that should take into account the diversity of computing and storage elements located on multiple sites. This methodology should also include the definition of dependability metrics and attributes for Grid applications. This will be important for the decision about the most adequate mechanisms and techniques that should be applied in each application scenario. Failure detection and diagnosis should be adapted to the failure model and taken in consideration the defined dependability attributes.

There are tens of papers in the literature about failure detection in small and medium scale distributed systems. The main challenge in the area of Grid is to devise enhanced mechanisms for failure detection in wide-area networks, comprised of heterogeneous elements and different middleware, able to detect network partitions, transient and permanent failures, detection of failures in wired but also wireless environments, detection of crash-failures but also more complex failure models. Although there is some work in this direction (e.g. [14]) there are some issues that should be addressed in a more effective way.

According to the survey presented in [11] the main difficulty of the Grid users has been the failure diagnosis that so far requires too much involvement from the system managers and application users. It is clear that this issue requires extensive work to be much more integrated within the Grid Middleware. So far, there have been some considerable advances in Grid Monitoring Systems (e.g. [15]) but most of the existing systems have been focused on performance issues. An interesting step forward is to enhance Grid Monitoring Systems with the ability to conduct failure diagnosis and determine the possible chain of error-propagation. Since Core-Grid includes some good team of researchers in this area of Monitoring we expect some good results in this particular issue: the enhancement of Monitoring Architectures to conduct the failure-diagnosis and to provide input to the middleware modules that will provide the next steps of fault-tolerance.

A related problem to diagnosis is the definition of a theory and framework for error propagation. Some published work [16] has presented some advances in this topic. The idea is to understand the nature of the error and the possible propagation chain throughout the interacting modules in the rest of the system. Once an error is diagnosed and its impact is understood, then the system should apply a different technique, like retry, replicate, resume from a previous state, reset or partial reboot. Without the previous assessment about the error and its damage it can be useless to apply any static recipe for failure-detection. This topic of error propagation needs to deserve some more attention in the area of Grid-Computing, due to its complex nature and the fact it assumes the integration of diverse elements.

2- Checkpointing-and-Recovery

The technique of checkpoint-recovery has been extensively studied in the past two decades [17] and has been highly used in long-running scientific applications. Currently, there are several commercial systems and application-level libraries that support different implementations of checkpointing [18, 19]. However, most of these systems are focused on workstation clusters, dedicated parallel machines and medium-scale distributed systems. The main challenge in this topic is to extend the current checkpointing algorithms for scalable systems, comprised by thousand of nodes and heterogeneous elements. The work published in [20] gives some insights

about the future of this technique in peta-scale systems, and we feel this is an avenue with a high potential for research.

Checkpoint-recovery requires some stable storage infrastructure to keep the checkpoint data. An interesting issue is to build a highly scalable storage system, without compromising the performance and the scalability of the applications running on the Grid. A possible task of research will be the exploitation of Peer-to-Peer techniques to achieve this goal of scalability [21].

Apart from the scalability issues it is also mandatory to keep working on checkpointing mechanisms and protocols and adapt them to the Grid infrastructures. In this track it is necessary to consider three main issues:

- *User-Transparency*: checkpointing can be completely transparent to the application-programmer or alternatively it can be made available in the form of a system-library and the programmer has to instrument the code with some checkpointing calls. While the first approach has the advantages of transparency the second one provides lower complexity, performance overhead and can be more easily used in heterogeneous systems;
- *Checkpoint Level*: the method can be applied at the system-level or at the application level. In the first case the checkpoint data should include some state from the operating system, including the state of the communication channels and process context. In the second case, the checkpoint only contains the application data. This particular case is applied when the application-programmer makes use of some checkpointing primitives and is responsible for the definition of consistent global states of the application. In the former case, when the method is applied at the operating system level, it is a requirement for the checkpointing algorithm to achieve a consistent global state and deal with the sources of non-determinism and state of the communication channels;
- *Heterogeneity*: checkpoints can be stored in a binary format that only enables the rollback-recovery of the application in a homogeneous system or they can be stored in some neutral format that allows the recovery of the application in a possibly heterogeneous system.

The extension of checkpointing algorithms and protocols for Grid Computing, should take into account these three issues and it will be interesting to conduct some experimental work to evaluate different mechanisms according to a set of metrics like: portability, checkpoint size, interoperability, scalability, correctness and availability. Some practical work has been done in this direction [22] but this topic has still a lot of open issues to be addressed.

The community of researchers within Core-Grid should pay attention to some of the guidelines of the Grid Checkpoint Recovery Working Group from the GGF [23]. This working group will define a user-level API and associated layer of services that will permit the recoverability of jobs among heterogeneous Grid resources.

Checkpointing is not only a technique for fault-tolerance and recoverability of long-running applications. In the case of Grid Computing it is also seen a potential method for strategic job preemption and migration [24]. This allows job load balancing among available Grid resources and will facilitate the implementation of fault-tolerant scheduling algorithms. The usage of checkpointing for job-migration in scalable and heterogeneous clusters has some open issues that need to be studied in detail.

Another important issue is to compare the usage of checkpointing versus a primary-backup approach for replication to achieve high-available Grid Services. The main dependability metrics and the performance overhead of both schemes should be experimentally evaluated.

3- Fault-Tolerant MPI

Job-batching systems like Condor [24] already provide support for system-level checkpoints for job-migration and recoverability. However, they are not targeted to parallel programs but rather limited to embarrassingly parallel applications. The standard for executing message-passing parallel applications is MPI.

In both versions of the MPI standard the support for fault tolerance is only specified for the communication channels, which are guaranteed to be reliable, but not for process/machine faults. If a process or machine fails the default behavior is for all other nodes participating in the computation to abort. The user may change this by providing error handlers, but it is not assured they will be even called. When MPI was first devised the dominant systems were parallel machines and dedicated clusters. These systems were considered quite reliable. However, the MTBF that is expected for a Grid environment is considerably lower and it is mandatory to include some support of fault-tolerance for the next version of MPI.

Many research projects have been studying this issue, like MPICH-V [25], FT-MPI [26], MPI-FT [27], MPICH-GF [28], MPI/FT [29], among others.

The way to deal with faults in MPI programs is still an open issue of research. Basically there are two main options: (i) the MPI implementation provides some API for fault-tolerance that should be used by the application programmer; (ii) or the MPI implementation provides some logging and checkpointing protocol for automatic rollback-recovery. While this last approach has the potential advantage of transparency it still has some issues to be addressed like the higher performance overhead and the lack of portability. Although there is been several contributions in this topic there is still work to be done and the researchers of Core-Grid can play a very active role in this topic.

4- Dependability for Data Grids

While most of the previous fault-tolerance schemes are targeted for the Computational Grid, it is also important to address the issues of dependability in Data Grids.

Within this topic we foresee the need for extended research in issues like: data-replication protocols, resilient data catalogs, replica management and placement, fault-tolerant wide-area data dissemination, data consistency and high-availability schemes for critical data-Grid services. The existing work in the literature [30] will have to be enhanced, experimentally evaluated and implemented in production Grids. Some P2P techniques will certainly be useful to achieve some of the previous goals.

5- Fault-tolerant Global Computing

In some sense, the first steps towards the convergence between Grid Computing and Peer-to-Peer systems have already been done by exemplar systems like Boinc [31] and XtremWeb [32]. Boinc has called the attention of the community as the middleware basis for projects like SETI@Home, among others. These systems make use of CPU cycles and resources that are donated by Internet users. Since these open systems are naturally unstable the support for fault-tolerance is absolutely mandatory, and several schemes have been developed for failure-detection, task-replication, task-logging, fault-tolerant scheduling and sabotage tolerance to detect malicious results.

Core-Grid can contribute to the advances in this topic, since one of the partners (INRIA) is the promoter of XtremWeb.

This area of fault-tolerance for Global Computing has an interesting set of problems to be addressed, like the combination of checkpointing with task-replication schemes to increase the turnaround of jobs, more advances in fault-tolerant scheduling algorithms, more effective techniques for sabotage tolerance and trust in open environments, more scalable protocols for resilient task distribution.

An interesting step forward is to relax one of the strong assumptions in these systems: so far, these Global Computing systems have only been used to execute coarse-grain master-worker applications. An interesting, although difficult, challenge is to make these systems able to execute SPMD applications. Together with this step in the application paradigm it is also necessary to address the dependability issues for this type of applications, where the individual processes communicate with each other, making fault-tolerance more difficult to achieve [33].

Recommended focus areas within the domain of dependability

The following list presents some avenues of future research in the area of Dependability in Grid. Some of them will be surely addressed by the team of researchers within Core-Grid.

- *Definition of a failure model for Grid:* the first step in this topic should be a detailed analysis of failures that occur in real applications running on Grid environments. Some small steps have been done in [11] and [34] but this task still has some work to do. After this step, it is important to define a failure-model that should be appropriated to the several flavors of Grid Computing environments;

- *Definition of dependability attributes for Grid applications*: definition of attributes and metrics to evaluate and assess the dependability requirements of Grid applications and to adopt the best fault-tolerant method for each particular application or system;
- *Failure detection*: extend existing failure-detection algorithms for wide-area networks with support for more complex failure models, network partitions and scalability;
- *Failure diagnosis*: improved mechanisms for failure diagnosis, hopefully integrated with Grid Monitoring Systems [35];
- *Error propagation*: definition of models for error-propagation in complex systems like the Grid, and instrumentation of Grid middleware for error-assessment;
- *Robustness mechanisms at the middleware level*: address mechanisms like exception-handling, micro-rebooting, partial replication, error-recovery and software rejuvenation to make the Grid middleware more robust to failures;
- *Configuration management*: there is some potential to address the problem of configuration management in complex systems like Grid infrastructures, and the schemes that will be devised for configuration management should be able make some contribution to higher dependability of the applications.
- *Checkpointing-recovery*: this method will be highly used in Grid applications but is necessary to adapt some of the existing algorithms and mechanisms to the Grid infrastructure, taking into account metrics like: portability, checkpoint size, interoperability, scalability, correctness and checkpoint data availability. The integration of P2P techniques to implement a scalable storage system to keep the checkpoints seems to be a promising topic to be addressed;
- *Fault-tolerant MPI*: this topic will deserve the attention of the research community in the next coming years. The road to obtain a fault-tolerant implementation of MPI or an enhanced MPI implementation with fault-tolerance primitives will be clearly followed, and this NoE will certainly give some contributions to this issue;
- *Fault-tolerant scheduling*: the previous work on fault-tolerant scheduling of applications has been too much restricted to Master-Worker applications [36]. It is important to extend the support for fault-tolerant scheduling for other application paradigms and to consider more complex failure models.
- *Task-replication*: it will be interesting to conduct some experimental studies and modeling about usage of task-replication schemes, mainly in coarse-grain applications that execute in wide-area networks;
- *Combine checkpointing and task-replication*: study the integration of checkpoint-recovery together with task replication schemes in order to augment the application turnaround and obtain the better levels of performance when trying to achieve high resiliency in desktop Grid applications;
- *Grid-Services Replication*: some of the Grid Services have to be made highly-available and one possible technique is to replicate them on two or more hosts. The extensive work that has been done in replication schemes and group communication protocols in the past 20 years should be applied and adapted to the particularities of Grid Services. The study presented in [37] is an example of a first step towards this goal, but some more work is expected in this topic.
- *Reliable wide-area data movement*: it is important to continue the research work in mechanisms to achieve consistency and resiliency in wide-area data movement;
- *Protocols for dependable data Grids*: there are still open issues to be addressed in protocols for data replication, replication catalogs, replica location and movement, and the usage of P2P techniques seems to bring high potentialities to this topic;
- *Workflows for Dependable Grid*: a very appealing avenue of research is the definition of high-level workflows for failure-handling in Grid applications [38];
- *Naturally fault-tolerant and scalable algorithms*: the work presented in [39], although not being general-purpose, has proved that some algorithms can be made scalable and have the property of natural fault-tolerance.

This property can be achieved in a minor percentage of applications, but it represents a good topic of research to understand the behavior of natural self-healing applications and the adaptation of this attribute to new classes of algorithms.

- *Sabotage Tolerance*: an interesting topic of research that merges the fields of fault-tolerance and trust and security is the development of distributed protocols for sabotage tolerance. These protocols are particularly relevant in Global computing environments [31] since the existing schemes still present some restrictions that should be solved.

- *Dependability Benchmarking*: after developing some fault-tolerance schemes it is necessary to evaluate the dependability metrics that can be achieved. This may require the construction of tools for dependability benchmarking, mainly targeted to evaluate the robustness of Grid applications.

Research projects related to dependability

The following table describes the research projects of the partners which are related to the domain of dependability. It serves as a reference for identification of potential cooperations and for the analysis of research gaps in respect to the vision.

Institute	Contributions
INRIA	<p>OASIS - Elaboration of Grid fault tolerance protocols :</p> <p>The project is working on the design of an adaptable fault tolerance protocol mixing communication induced checkpointing and message logging. The main objectives are</p> <ul style="list-style-type: none"> ▪ A fault tolerance protocol for communication induced checkpointing ▪ A single protocol that mixes communication induced checkpointing and message logging. For example, message logging between clusters and communication induced checkpointing inside each cluster.
INRIA	<p>Grand Large - Design of vertical fault tolerance bus for MPI :</p> <p>This project is working on the design of generic methods to provide and publish fault tolerance at different levels of the software-stack of an MPI application. Fault Tolerance for MPI is now widely studied with many different points of view. Some researches tend to provide automatic and transparent fault tolerance, while others provide mechanisms to help the application tolerate the failures by itself. The idea of a fault-tolerance bus is to determine a generic API to register and publish the kind of faults the different levels may handle. For example, the classical goal of the network layer is to provide lossless, ordered, secure communications. This has a cost which may not be usefully paid for specific of applications or specific protocols which tolerate by themselves and more efficiently losses, or perturbations of the communication. Other protocols may require a stable communication layer to provide higher level fault-tolerance. A means to publish and require these ability is now necessary to unify the different approaches to fault-tolerance for MPI computing.</p>
INRIA	<p>Grand Large - Design and evaluation of MPI Fault-Tolerant protocols for the Grid :</p> <p>The project is designing and evaluating Message Passing Fault-Tolerant protocols which use efficiently the natural hierarchy of Grids. Composition of Fault-Tolerant Protocols is not straightforward: according to the active fault-tolerant protocol at a cluster level, the crash of a node in this cluster may change the whole output of the cluster to the other nodes of the Grid (e.g. Assume the cluster provides fault tolerance using a Chandy-Lamport algorithm, when a fault occurs, the whole cluster rollbacks and may provide an output different from the previous execution). The project is trying to define the necessary collaboration between the different layers of Fault-Tolerance, and to determine if the composition may be effective or if a single fault-tolerance protocol on the Grid, without possibility of hierarchy should be developed.</p>
INRIA	<p>Grand Large - FAult Injection Language (FAIL) :</p> <p>This project is designing and evaluating distributed fault-injection software. The</p>

	<p>Architecture permits to inject faults independently of the programming language used in the tested application, and is aimed at large-scale GRID environments.</p> <p>In a network consisting of several thousands computers, the occurrence of faults is unavoidable. Being able to test the behaviour of a distributed program in an environment where the faults (such as the crash of a process) can be controlled is an important feature that matters in the deployment of reliable programs.</p> <p>The project proposes to design, implement and test FAIL, a new tool for software fault injection in distributed applications. The solution would allow to elaborate complex faults scenario in a simple way, while preventing the user to write low level code.</p> <p>Besides, it should be possible to generate probabilistic scenarios (for average quantitative tests) or deterministic and reproducible scenarios (for studying the application's behaviour in particular cases). Finally, FAIL should work with applications written in numerous programming languages without requiring the modification of their source code.</p>
MU	Scalable Grid Monitoring Architecture : see Task 4.3
MTA SZTAKI	Flexible Monitoring Infrastructure for large scale Grids : see Task 4.3
MTA SZTAKI	Workflow Management in Grid Environments : see Task 4.4
UCAM	Multicast Transport for Grid Computing : see Task 4.3
UCAM	<p>XenoGrid: A dynamic, fault-resilient Grid :</p> <p>Grids themselves can be deployed as services on the XenoServer platform for global public computing (http://www.xenoserver.org). Using XenoServers as a substrate for the deployment of Grids will provide important security, fault-resilience, reconfigurability, and on-demand mobility benefits. Grid nodes can be easily monitored, restarted, replicated, or migrated whenever needed.</p> <p>The project is building an experimental Grid testbed running on the XenoServer platform and investigating the dynamicity and dependability advantages gained.</p>
UCL	Decentralized service architecture : see Task 4.3
UCL	<p>Security based on the principle of least authority :</p> <p>The project tries to apply thoroughly the principle of least authority to languages and systems, and explore to what degree this will allow the solution of problems such as viruses and too much ambient authority. The central questions this project is trying to answer are :</p> <ul style="list-style-type: none"> ▪ What are the fundamental principles of language-based security? ▪ What does such a language look like?
UCL	Collaborative applications for highly dynamic environments : see Task 4.3
UCO	Fault-Tolerant MPI
UCO	Fault-Tolerant Desktop Grid Computing : see Task 4.3
UCO	Reliable Data Dissemination in Data Grids : see Task 4.3
UOW	<p>Autonomy and high service availability in wireless Grid environments :</p> <p>The project is trying to achieve the integration of mobile devices into the Grid following a clustering approach where all devices in the same subnet are grouped and presented as a single virtual system to the Grid. A set of proxies resides between the wireless “cluster” and the Grid taking care of this virtualization using a number of middleware services. The ultimate goal is to provide failure prediction, detection and recovery along with high service availability and autonomy capabilities in this wireless Grid environment. Subgoals are :</p> <ul style="list-style-type: none"> ▪ Implement the necessary set of middleware services that will constitute the proxy engine responsible for the virtualization of available mobile devices. ▪ Identify limitations of current failure detection schemes when ported in unreliable and dynamic environments. ▪ Prototype and evaluate a lightweight failure detection and recovery scheme suitable for wireless environments. ▪ Develop high availability mechanisms based on existing and tested schemes (like heartbeats or intelligent agents) modified for wireless environments.

Table 3: Research projects related to dependability

Integration of the proposed methods (Task 4.6)

Testing, evaluation and benchmarking of the approaches in each of the areas covered in Tasks 4.2-4.4 will take place on a continuous basis. However, since elements of GRID architectures are very likely to be interdependent, the proposed individual mechanisms might not work efficiently together, be redundant, or allow better solutions in combination. Therefore it is necessary to perform integration work and studies on the interoperability of the methods in a more focused task.

In order to obtain quantitative results of the proposed approaches and facilitate the integration of the prototypic components, a testbed environment described in Task 4.2/Section 6.1.6 of Dow will be used for the integration of mechanisms and the interoperability benchmarking.

The process might require changes to each of the technical approaches as well as addition of new components or characteristics. Furthermore, based on the results of Task 4.1, reconciliation with the current state of the technology will be considered.

Mechanisms

Workshops

Workshops are held to enhance the collaboration among the partners. The workshops take place every four months. As it is crucial to develop a common understanding about the research activities, extended workshops featuring full day presentations of corresponding search results, especially joint activities, are planned.

Planned meetings:

- The first meeting of the workpackage 4 (System Architecture) has taken place at ICS-FORTH in Heraklion, Greece, on January 18th 2005. It was attended by over 30 researchers from the 16 institutes and universities participating in WP4. The main topics of the meeting were the coordination of the work around the scientific projects pursued by the partners, and the preparation of the WP4 Roadmap.
- The meeting was preceded by the "1st CoreGRID Workshop on GRID and P2P Systems Architecture" which took place at the same location one day before, and was organized by the WP4. Among the invited speaker were Mema Roussopoulos from Harvard University, USA and Ian Taylor from Cardiff University, UK.
- The next workshop is planned for 1st of June 2005. This one-day workshop will be used for discussing the joint activities within the tasks. Co-location of the workshop in Barcelona together with meetings of other Virtual Institutes (WP6, WP5) will provide opportunities for exchange of scientific ideas and creating further contacts.

Partner Meetings

Specific research activities are discussed in partner meetings. The partners inform each other about their research tasks and expected results prior to the actual meeting in order to allow for a focused discussion on the subject. The partner meetings are held in form of short visits between partners and workshops on the Virtual Institute level.

E-meetings and tele-conference meetings

To further enhance the communication between the partners tele-conferences and e-meetings are held to discuss pressing issues.

Researcher/Student Exchanges

Exchange of students is planned by several partners to improve the level of integration within the CoreGRID JPA.

Inviting external people

External people are invited to the workpackage activities to spread the knowledge about CoreGRID. Members of WP4 frequently take part in committees and forums to avoid redundancy and improve cooperation.

A common experimental testbed

Within WP4, task 4.2 is devoted to the development of a common experimental testbed. This activity provides an opportunity to hands-on scientific exchange and cooperation.

Proposing new projects

To cover the issues from the vision which are not subject to the research by any of the partners, or are not investigated in a sufficient degree, proposal of new projects (STREPs or IPs) is intended. The particular focus, timing and partners of such projects will be discussed during the forthcoming Virtual Institute meetings.

Dissemination of results

Publications on collaborative work will be submitted to the respective conferences and journals as well as to the CoreGRID website. Like this partners will be informed about the results. CoreGRID results will also be considered in the partners' individual research projects, thus enhancing the relevance and spreading the knowledge about CoreGRID within the scientific community.

CoreGRID portal

The CoreGRID portal is being used for exchanging information between the partners and for publishing information beyond the network. This is done via a public and several private sections in the network.

Exchange of documents

The members of CoreGRID have access to a WWW-based document/file sharing platform, a BSCW server, which facilitates exchange and sharing of documents. This tool is used by the Virtual Institute for uncomplicated sharing and dissemination of documents such as meeting presentations, reports, roadmaps, scientific papers and software.

Future steps

The activities of this Virtual Institute are aligned with and support the overall objectives of CoreGRID to ensure sustainable integration within the European Grid research community. This implies that the links and cooperations established during the duration of the project continue beyond its lifetime. This Virtual Institute attempts to achieve this goal by the following mechanisms:

- creation of "inofficial" research partnerships by bringing together researchers with similar interests and supporting joint research papers
- fostering of student exchanges, joint PhD supervision involving several partners
- joint workshops and conferences which create durable bindings between partners.

Furthermore, it is expected that the gap analysis and the consequential stipulation of new research projects on institutional, national or European level will have the strongest impact on the sustainable cooperation beyond the end of CoreGRID. The activities within the Virtual Institute are thus intentionally directed at proposing new joint research projects.

5. Links with other CoreGRID scientific workpackages

The systems architecture Virtual Institute is very important for the success of Grid computing since it defines the basic building blocks of the Grid system. The Grid architecture research in the context of CoreGRID will focus on three different issues, namely, scalability, adaptability, and dependability. Several of the institutes involved in WP4 already have active projects that explore more than one of the above issues simultaneously. This research is expected to be integrated in the context of CoreGRID.

The primary goal of the institutes involved in system architecture is to promote and encourage the close collaboration among them. We will first focus on the exploitation of cooperation within the WP4 workpackage with the known instruments. This has already been promoted and can be proven by the numerous short visits among institutes involved in WP4.

However, close collaboration with the other Virtual Institutes will be investigated. This is motivated by the fact that the research aspects related to system architecture are interwoven with a large number of other research topics in GRID systems. Therefore we expect an active and continuous cooperations and links to other Virtual Institutes during the project duration. Some of these links are determined by the technical input which the partners in WP4 are expecting from the other Virtual Institutes as a complement of their research work. There inputs include, but are not limited to:

- **from WP1:** GRID testbed "sandbox" and interfaces for inclusion of mechanisms for scalability, adaptability and dependability,
- **from WP2:** Services for processing GRID state information and Data Mining tools for discovering activity/failure patterns,
- **from WP3:** Interfaces (possibly component-based) between programming model and architectural mechanisms,
- **from WP5:** Services for Grid monitoring and interfaces for controlling resources,
- **from WP6:** Fault-tolerant scheduling methods and distributed, P2P-based scheduling approaches,
- **from WP7:** Platform-independent system-level interface model.

6. References

1. IBM. Autonomic computing manifesto, October 2001.
2. R. Haas, P. Droz, and B. Stiller. Autonomic service deployment in networks. *IBM Systems Journal*, 42(1):150–164, 2003.
3. L.W. Russel, S. P. Morgan, and E. G. Chron. Clockwork: A new movement in autonomic systems. *IBM Systems Journal*, 42(1):77–84, 2003.
4. G. Lanfranchi, P. D. Peruta, A. Perrone, and D. Cavanese. Toward a new landscape of systems management in an autonomic computing environment. *IBM Systems Journal*, 42(1):119–128, 2003.
5. Hewlett-Packard Company. Utility data center, November 2001.
6. J. Rolia, A. Andrzejak, and M. Arlitt. Automating enterprise application placement in resource utilities. In *Proceedings of 4th IFIP/IEEE Workshop on Distributed Systems: Operations and Management (DSOM 2003)*, Heidelberg, October 2003.
7. A. Andrzejak, S. Graupner, V. Kotov, and H. Trinks. Self-organizing control in planetary scale computing. In *Proceedings of CCGrid, 2nd Workshop on Agent-based Cluster and Grid Computing (ACGC)*, Berlin, 2002.
8. P. Kacsuk, G. Dzsá, J. Kovcs, R. Lovas, N. Podhorszki, Z. Balaton, and G. Gombs. Pgrade: a grid programming environment. *Journal of Grid Computing*, 1(2):171–197, January 2003.
9. P. Dinda, D. O'Hallaron, An Extensible Toolkit for Resource Prediction In Distributed Systems, technical report CMU-CS-99-138, School of Computer Science, Carnegie Mellon University, July, 1999.
10. F.Gartner. “*Fundamentals of Fault-Tolerance Distributed Computing in Asynchronous Environments*”, *ACM Computing Surveys*, 31 (1), March 1999
11. R.Medeiros, W.Cirne, F.Brasileiro, J.Sauvé. “*Faults in Grids: why are they so bad and what can be done about it?*”, *Proceedings of the Fourth International Workshop on Grid Computing*, 2003
12. I.Foster, A.Iamnitchi, “*On Death, Taxes and the Convergence of Peer-to-Peer and Grid Computing*”, *Proc. 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03)*, Feb. 2003
13. I.Foster, C.Kesselman, “*The Grid: Blueprint for a New Computing Infrastructure*”, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1999.
14. P.Stelling, I.Foster, C.Kesselman, C.Lee, G.Laszewski. “*A Fault Detection Service for Wide-Area Distributed Computations*”, *Proc. 7th IEEE Symposium on High-Performance Distributed Computing*, 1998, pp. 268-278
15. M.Baker, G.Smith. “*GridRM: A Resource Monitoring Architecture for the Grid*”, Technical Report University of Portsmouth UK, June 2002
16. D.Thain, M.Livny. “*Error Scope on a Computational Grid: Theory and Practice*”, *Proc. 11th IEEE International Symposium on High Performance Distributed Computing HPDC-11 20002 (HPDC'02)*, July 2002, Edinburgh, Scotland
17. E. N. Elnozahy, L. Alvisi, Y.M. Wang, D.B. Johnson. “*A Survey of Rollback-Recovery Protocols in Message-Passing Systems*”, Technical Report CMU-CS-99-148, Carnegie Mellon University, 1999

18. A. Beguelin, E. Seligman, P. Stephan. "Application-level Fault-Tolerance in Heterogeneous Networks of Workstations", Parallel and Distributed Computing on Workstation Clusters and Networked-based Computing, June 1997
19. M. Livny, J. Pruyne. "Managing Checkpoints for Parallel Programs", Proc. IPPS Second Workshop on Job Scheduling Strategies for Parallel Processing, 1996
20. E. Elnozahy, J. Plank. "Checkpointing for Peta-Scale Systems: A Look into the Future of Practical Rollback-Recovery", IEEE Transactions on Dependable and Secure Computing, 1(2), April-June, 2004, pp. 97-108.
21. C. Germain, G. Fedak, V. Néri, F. Cappello, "Global Computing Systems", Proceedings of the 3rd International Conference on Large-Scale Scientific Computing, 2001
22. S. Krishnan, D. Gannon. "Checkpoint and Restart for Distributed Components in XCAT3", Proc. 5th IEEE/ACM International Workshop on Grid Computing, Nov 2004.
23. GGF Grid Checkpoint Recovery Working Group, <http://gridcpr.psc.edu/GGF/>
24. D. Thain, T. Tannenbaum, M. Livny, "Condor and the Grid", in Fran Berman, Anthony J.G. Hey, Geoffrey Fox, editors, Grid Computing: Making The Global Infrastructure a Reality, John Wiley, 2003
25. A. Bouteiller, F. Cappello, T. Hérault, et al. "MPICH-V2: a fault tolerant MPI for volatile nodes based on pessimistic sender based message logging". In High Performance Networking and Computing (SC2003), 2003.
26. G. Fagg, A. Bukovsky, J. Dongarra. "Harness and Fault-Tolerant MPF", Parallel Computing, Vol. 27, No 11, pp. 1479-1495, October 2001
27. S. Louca, N. Neophytou, A. Lachanas, and P. Evripidou. "MPI-FT: Portable fault tolerance scheme for MPI". In Parallel Processing Letters (PPL), volume 10(4). World Scientific Publishing Company, 2000.
28. N. Woo, S. Choi, H. Jung, 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, Tokyo, Japan
29. R. Batchu, J. Neelamegam, et al. "MPI/FT: Architecture and taxonomies for fault-tolerant, message passing middleware for performance-portable parallel computing". Proceedings of the 1st International Symposium of Cluster Computing and the Grid (CCGRID2001, Melbourne, Australia, May 2001
30. X. Qin, H. Jiang, "Data Grid: Supporting Data-Intensive Applications in Wide-Area Networks", Technical Report TR03-05-01, Dep. of Computer Science and Engineering, University of Nebraska-Lincoln, May 2003
31. D. Anderson. "BOINC: Berkeley Open Infrastructure for Network Computing", Technical Report Univ. California Berkeley, 2002
32. F. Cappello, S. Djilali, G. Fedak, et al. "Computing on Large Scale Distributed Systems: XtremWeb Architecture, Programming Models, Security, Tests and Convergence with Grid", FGCS Future Generation Computer Science, 2004.
33. Weissman. "Fault-Tolerant Computing on the Grid: what are my options?", Proc. 8th IEEE International Symposium on High Performance Distributed Computing, August 1999
34. G. Kola, T. Kosar, M. Livny, "Phoenix: Making Data-intensive Grid Applications Fault-tolerant", In Grid 2004, Pittsburgh, PA, November 2004
35. GMA-WG: Grid Monitoring Architecture Working Group, <http://www-didc.lbl.gov/GGF-PERF/GMA-WG/>
36. E. Heymann, M. Senar, E. Luque, M. Livny, "Adaptive Scheduling for Master-Worker Applications on the Computational Grid". in Proceedings of the First IEEE/ACM International Workshop on Grid Computing (GRID 2000), Bangalore, India, December 17, 2000
37. X. Zhang, D. Zagorodnov, M. Hiltunen, K. Marzullo, R. Schlichting. "Fault-Tolerant Grid Services using Primary-Backup: Feasibility and Performance", Proc. IEEE. Intl. Conf. on Cluster Computing (CLUSTER), San Diego, California, USA, September 2004
38. S. Hwang, C. Kesselman. "Grid-Workflow: A Flexible Failure Handling Framework for the Grid", Proc. 13th IEEE Int. Symposium on High-Performance Distributed Computing (HPDC-13), June 2003
39. A. Geist, "Development of Naturally Fault-Tolerant Algorithms for Computing on 100,000 Processors", Journal of Parallel and Distributed Computing, <http://www.csm.ornl.gov/~geist/>
40. Reinefeld, F. Schintke, T. Schütt: Scalable and Self-Optimizing Data Grids, Annual Review of Scalable Computing, Vol. 6, edited by Yuen Chung Kwong, World Scientific, June 2004.
41. F. Schintke, A. Reinefeld: Modeling Replica Availability in Large Data Grids, Journal of Grid Computing, 1(2):219-227, 2003.
42. Seif Haridi, Peter Van Roy, Per Brand, Christian Schulte, Programming Languages for Distributed Applications, New Generation Computing, 16(3):223-261, 1998.

43. Artur Andrzejak, M. Ceyran: *Characterizing and Predicting Resource Demand by Periodicity Mining*. In: Journal of Network and System Management, special issue on Self-Managing Systems and Networks, Vol. 13, No. 1, Mar 2005.
44. Distributed Management Task Force (DMTF). *DMTF CIM Concepts White Paper*. http://www.dmtf.org/standards/published_documents.php
45. P. Dinda, D. O'Hallaron, *An Extensible Toolkit for Resource Prediction In Distributed Systems*, technical report CMU-CS-99-138, School of Computer Science, Carnegie Mellon University, July, 1999.
46. Artur Andrzejak, U. Hermann, A. Sahai. *FeedbackFlow - An Adaptive Workflow Generator for System Management*, ZIB-Report 05-12, 2005.
47. M. Ghallab, D. Nau, P. Traverso. *Automated Planning - Theory and Practice*. Elsevier 2004.
48. R. V. van Nieuwpoort and J. Maassen and G. Wrzesinska and R. Hofman and C. Jacobs and T. Kielmann and H. E. Bal, Ibis: a Flexible and Efficient Java-based Grid Programming Environment, Concurrency and Computation: Practice and Experience, to appear
49. Fabrice Huet and Denis Caromel and Henri E. Bal, A High Performance Java Middleware with a Real Application, Proceedings of the Supercomputing conference, nov 2004, Pittsburgh, Pennsylvania, USA
50. Laurent Baduel, Françoise Baude, Denis Caromel, Arnaud Contes, Fabrice Huet, Matthieu Morel, Romain Quilici, Jose C. Cunha and Omer F. Rana (editors), *Grid Computing: Software Environments and Tools, Programming, Deploying, Composing for the Grid*, Springer-Verlag, to appear.
51. D. de Roure, N. R. Jennings, and N. Shadbolt. The semantic Grid: Past, present and future. Proceedings of the IEEE, 93, 2005.
52. I. Foster, N.R. Jennings, and C. Kesselman. Brain meets brawn: Why grid and agents need each other. In Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04), New York, USA, July 2004. IEEE.
53. N.R. Jennings. An agent-based approach for building complex software systems. Communications of the ACM, 44(4):35–41, 2001.
54. Katia P. Sycara. Multiagent systems. AI Magazine, 19(2), summer 1998.
55. M. Wooldridge. Agent-based software engineering. IEEE Proceedings of Software Engineering, 144:26–37, 1997.
56. M. Wooldridge. Introduction to MultiAgent Systems. John Wiley & Sons, 2002.

6. Participants

Partner Name	No.	Researchers	T1	T2	T3	T4	T5	T6
CNRS	5	Denis Trystram	X		X			
		Estelle Gabaron			X			
		Feryal Moulai			X			
		Guillaume Huard	X	X	X			X
FORTH	11	Evangelios Markatos			X			
		Paraskevi Fragopoulou	X		X			
INRIA	14	Arnaud Contes						
		Benjamin Quetier		X				
		Christian Delbe					X	
		Christian Perez						
		Daniel Hagimont				X		
		Denis Caromel						
		Emmanuel Cecchet				X	X	
		Fabrice Huet				X		
		Franck Cappello	X	X	X			
		Francoise Baude						
		Gilles Fedak	X	X	X			
Jean-Bernard Stéfani					X	X		

		Sammy Haddad						
		Sebastien Tixeuil	X				X	
		Thomas Herault	X				X	
		William Hoarau					X	
KTH	15	Ali Ghosti		X	X			
		Seif Haridi			X			
		Vladimir Vlassov		X	X			
MTA SZTAKI	20	Zoltán Balaton	X	X	X	X		X
MU	16	Ludek Matyska		X	X			
		Miroslav Ruda		X				
		Petr Holub			X		X	
SICS	19	Ali Ghodsi			X			X
		Konstantin Popov	X	X	X		X	X
		Per Brand	X	X	X		X	X
UCAM	24	Andy Parker	X	X				
		Jon Crowcroft	X	X	X			
		Mark Calleja	X					
UCL	31	Peter Van Roy						
		Valentin Mesaros				X		
UCO	27	Joao Gabriel Silva					X	
		Luis Silva	X				X	
		Nuno Santos					X	
		Patricio Domingues					X	
UCY	28	Athina Stassopoulou						X
		Eleni Tsiakkouri		X				X
		George Tsouloupas						X
		Marios Dikaiakos		X				X
		Wei Xing		X				X
UNICAL	23	Domenico Talia	X	X	X			X
		Paolo Trunfio	X	X	X			X
UoW	37	Henrio Ludovic				X		
		J. Thiyagalingam		X		X		X
		Stavros Isaiadis				X		
		Vladimir Getov						
VTT	40	Janne Väre			X			
		Kimmo Ahola			X			
		Mika Pennanen			X			
		Mikko Alutoin				X	X	
		Pertti Raatikainen				X	X	
		Sami Lehtonen			X			
ZIB	41	Alexander Reinefeld	X					
		Artur Andrzejak	X			X	X	
		Felix Hupfeld			X	X		
		Florian Schinkte		X	X			
		Thomas Röblitz		X				X
		Thomas Steinke				X		

Table 4: Participants of the Virtual Institute on System Architecture (WP4)