



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.03 – Roadmap version 2 on System Architecture

Due date of deliverable: February 28, 2006

Actual submission date: May 2, 2006

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:

Grid, Peer-to-Peer, Scalability, Adaptability, Dependability, Resource Discovery, Checkpointing, Sabotage Tolerance, Component Model

Table of content

1. Executive Summary	3
2. Introduction	4
Context.....	4
Problem(s).....	4
Objectives	4
Tasks	4
Drivers.....	6
Summary of the performed work	6
3. Positioning.....	8
State of the art	8
Scalability	8
Adaptability.....	8
Dependability	9
4. Vision, Strategy and Roadmap	11
Vision and Scenarios.....	11
Application Scenarios	11
Strategy	12
Roadmap	13
Phases of the roadmap.....	13
Task 4.1: P2P based services and resource discovery	14
Task 4.2: Self-organizing Grid services using P2P technology	15
Task 4.3: Dependability mechanisms for computation and data Grids	16
Task 4.4: Fault-tolerance and robustness	17
Task 4.5: Adaptive management of systems and resources	19
Task 4.6: Integration of the proposed methods	21
Task 4.7: Testbed adaptation.....	21
Research Groups	23
Mechanisms	23
Future steps	24
5. Trust & security issues.....	25
6. Interaction with Industry	25
7. Links with other CoreGRID Institutes.....	26
8. References	28
9. Participants.....	31

1. Executive Summary

This document is the Roadmap Version 2 of the CoreGRID Institute on System Architecture, which is to be submitted as deliverable D.SA.03. It is a continuation of Roadmap Version 1, Deliverable D.SA.01 and describes the concrete research directions to be followed during the second phase of the project.

The Institute on System Architecture (CoreGRID WP4) focuses on the architectural principles of Grid applications and infrastructure that meet the mandatory properties on Next Generation Grids (NGG). 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 partners of the Institute conduct research in three main directions: scalability, adaptability, and dependability of Grid architectures and Grid services. In JPA1 (M1-M18), six research tasks had been identified along the following research directions:

1. Definition of capabilities and requirements of future Grids
2. A common experimental and benchmarking environment
3. Scalability
4. Adaptability
5. Dependability
6. Integration of the proposed methods

The main objective of the Institute on System Architecture is to exploit the synergy of research projects conducted by partners. 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 contribute their expertise and are committed to strengthen the collaboration beyond the NoE. Roadmap 1 identified individual research contributions and collaboration topics between partners. During the first phase of the NoE joint Workshops have been organized to promote the collaboration of partners.

In the development of JPA2 (M13-M30) for the second year of CoreGRID, it was decided to continue along the same axes of research as in JPA1 but at the same time to reconsider each of the initially proposed tasks based on the collaborative activities of the partners as they evolved thought their interaction during the first year of the project. On this basis JPA2 includes the following tasks:

1. P2P-based resource location and discovery
2. Self-organizing Grid services using P2P technology
3. Dependability mechanisms for computational and data Grids
4. Fault-tolerance and robustness
5. Adaptive management of systems and resources
6. Integration of the proposed methods
7. Testbed adaptation

The main change is in the concretization of the three relatively general research axes namely the scalability, adaptability, and dependability that were included in JPA1. The scalability task was renamed to “P2P-based resource location and discovery” to better reflect the research carried out in this task. The adaptability task was split into two discrete, yet collaborating, tasks, namely the “Self-organizing Grid services using P2P technology” task and the “Adaptive management of systems and resources” task in order to reflect more accurately the work performed under the general enough notion of adaptability. Similarly, the dependability task was split into two individual tasks, the dependability mechanisms for computational and data Grids” task and the “Fault-tolerance and robustness” task.

In addition to roadmap 1, many joint research groups have evolved from the interaction of the partners involved in this Institute which reflect precise research projects currently carried out. Each one of the above tasks includes two to three different research groups. A description of these research groups is included along with the task descriptions.

2. Introduction

Context

CoreGRID's principal goal is to strengthen and advance scientific and technological excellence in the area of Grid and Peer-to-Peer computing. The technical work of CoreGRID is structured around six complementary research areas that have been selected for their strategic and research importance and the existing European expertise towards the next generation Grid middleware. Those areas have been identified as follows:

- KDM Institute: Knowledge and Data Management
- PM Institute: Programming Models
- SA Institute: System Architecture
- IM Institute: Grid Information, Resource and Workflow Monitoring Services
- RMS Institute: Resource Management and Scheduling
- STE Institute: Grid Systems, Tools, and Environments

Research in each of the above areas is conducted by CoreGRID Research Institutes that all together form the CoreGRID's Research Laboratory. The present document describes the roadmap of the Institute on System Architecture (CoreGRID's WP4). This Institute coordinates the research activities and promotes exchange of scientific expertise between the participating organizations in this particular research area. 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 interaction mechanisms. Grid architecture has to meet certain demands that are novel for the computing science. Because of the central role of systems architecture in Grid computing research, this Institute has close connections with all other CoreGRID 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 that govern their composition and interoperability. 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.

Objectives

This Institute focuses on the architectural principles of Grid applications and infrastructure that meet the mandatory properties on Next Generation Grids. **The main objective of this Institute is to perform a significant improvement of architectural designs of future Grids** by focusing on specific issues of three particular key aspects: 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 the 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 their 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 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 the Next Generation Grid and its mandatory architectural properties. We believe this process, with the involvement of CoreGRID NoE, will collect experience, generalize it, and eventually distil generic architectural principles.

In order to advance towards novel architectural solutions for the Next Generation Grid platform, the following research tasks have been identified in JPA2 (M13-M30):

1. P2P-based resource location and discovery
2. Self-organizing Grid services using P2P technology
3. Dependability mechanisms for computational and data Grids
4. Fault-tolerance and robustness
5. Adaptive management of systems and resources
6. Integration of the proposed methods
7. Testbed adaptation

Compared to the research task identified for SA Institute in JPA1 (M1-M18), we should mention that the three initially proposed research axes, namely, scalability, adaptability, and dependability became concrete and focused research direction, as they evolved through the interaction of the partners during the first phase of the project. The “P2P-based resource location and discovery” task is the former scalability task to better reflect the research carried out in this direction. The “Self-organizing Grid services using P2P technology” and the “Adaptive management of systems and resources” tasks reflect more accurately the work performed under the general enough notion of adaptability. Finally, dependability mechanisms for computational and data Grids” and the “Fault-tolerance and robustness” tasks capture the ongoing research in the domain of dependability.

These research tasks have been selected to advance towards effective solutions for the architectural components and their interaction of the Next Generation Grid. It is expected that the different research tasks will provide results that will lead to architectural components that include the following mechanisms:

- Scalable resource location techniques taking advantage of the existing P2P computing paradigm.
- Existing component models with enhanced capabilities over structured overlay network resulting in self-managing applications. The target is to extend existing component models, used in order to achieve enhanced reconfiguration capabilities and dynamic adaptation.
- Dependability mechanisms for desktop Grid computing based on distributed checkpoint-and-recovery schemes in highly heterogeneous Grid applications.
- Sabotage tolerance techniques to cope with malicious failures that may undermine results of long running Grid computations.
- Trust management schemes that distinguish the trustable from the un-trustable participants in global computations.
- Fault-tolerance MPI adapted to Grid environments.
- Fault injection mechanisms for dependability benchmarking in Grid environments.
- Automatic configuration and management enabling system adaptation.

The above research direction evolved through the interaction of the participants in the meetings and Workshops organized within this Institute. The initial fragmentation of research that was encountered in the beginning of the project disappeared to a large extent. That resulted to a mutual benefit since the research horizons of many groups were broadened. However, one of the main goals of the Institute is to continue to identify gaps in the research issues that continuously evolve and are not yet addressed in the present roadmap.

Below we elaborate on each one of the aforementioned research tasks.

Drivers

Architectural issues pertain nearly all aspects of Grid systems. The diversity of research that naturally existed in the beginning constituted the driving force that brought participants in this Institute together and led them to the development of novel common research direction. This resulted to a mutual benefit, since the research horizons of the participants were broadened through their interaction and at the same time this gave rise to research developments that are currently underway. The driving technical aspects tackled in this Virtual Institute will come mainly from the research domains of scalability, adaptability and dependability as they provided the common grounds to move towards concrete research directions in this second phase of the project.

The most influential factors of the work in the SA Institute 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 are currently analysed, linked and 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.

Summary of the performed work

In the first 18 months of the work of this CoreGRID Institute, a lot of integration work within the SA Institute and between the SA and other Institutes took place. Moreover, the structure and partially the goals of the SA Institute have been modified as a result of the integration experiences or difficulties, gained research knowledge, new research threads contributed by the partners, and external evaluation. As this is not the place to list all activities and achievements of the SA Institute, we list only those which are relevant to the development of the SA Institute and this version of the roadmap.

Concerning the integration activities, we performed the following work:

- The milestone M.SA.01 has been completed successfully hand-in-hand with the completion of Roadmap V1 (D.SA.01). The work and discussions of the partners in respect to these tasks resulted in a good understanding of the problem domain (current state and challenges), and helped to learn the mutual expertise and research interests of the partners involved in the SA Institute in depth.
- The meetings of the SA Institute and other mobility-based integration activities contributed to further structuring and update the goals of the SA Institute, and initiated closer and new cooperations within the SA Institute. The meetings took place in: January 2005: Heraklion, Greece; July 2005: Barcelona, Spain; September 2005: Lissabon, Portugal (informal); January 2006: Paris, France. The partners performed 16 short visits, initiated one REP (between UCY and KTH), and contributed to intra-Institute collaborations via two accepted fellowship programmes.
- The SA Institute workshops allowed to transfer scientific results and knowledge between partners within the SA Institute, other CoreGRID institutes, and also to disseminate this knowledge to the external institutions. In particular, these workshops were:
 - *1st CoreGRID Workshop on Grid and P2P Systems Architecture* in Heraklion in January 2005 with three invited speakers and 10 talks by CoreGRID partners,
 - *2nd CoreGRID Workshop on Grid and P2P Systems Architecture* in Paris in January 2006 with 13 presented papers by CoreGRID partners, and a panel discussion. The papers submitted to this workshop constituted D.SA.02, which shows the research results targeting the research areas described in the SA Roadmap V1. As a contribution to the dissemination, selected papers of this workshop will be published as a special issue of the International Journal of Future Generation of Computer Systems (FGCS).
- The scientific collaborations between partners yielded more than 30 publications, among them 5 CoreGRID technical reports (with few other submitted), and other common publications targeting the goals set in the Roadmap Version 1. Most of the work has been done in the area of dependability, followed by the P2P/scalability research. A new topic of desktop Grids emerged through the recent research activities of the partners involved in the SA Institute.

Concerning the changes of structure and research goals of the SA Institute, most significant work has been done through establishment of the Research Groups and the creation of JPA2. Specifically, through taking a “bottom-up” approach to creating JPA2, we have achieved the following:

- increase in the share of activities towards scientific outcomes,
- a more focused research work programme with a strong correspondence to the other (funding) projects of the participants,
- creation of small groups of scientifically well-aligned partners instead of big "bags of partners",
- establishing more concrete goals and duties of individual partners.

In concrete terms, we have substituted three “scientific” tasks from JPA1 by five more focused “scientific” tasks in JPA2. Each of these tasks contains two or more research groups of partners with highly aligned interests backed by other funding projects. While these research groups are dynamic, currently there are 12 of them, each targeting at publishing at least one common paper per year. As a result of this restructuring, the number of common publications and the share of activities towards scientific outcomes have been visibly increased.

3. Positioning

State of the art

In what follows we present a snapshot of the current state of technology in the Grid architecture research with specific focus on the three main key aspects, namely, scalability, adaptability, and dependability.

Scalability

One of the main challenges regarding the Grid is scalability. Classical approaches to Grid resource location are either centralized or hierarchical and will prove inefficient as the scale of Grid systems rapidly increases. On the other hand, the P2P paradigm emerged as a successful model that achieves scalability in distributed systems [12,24]. 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, special attention should be exercised so that individual services will also become scalable and fault-tolerant, meaning that they are not vulnerable to attacks. Therefore, it is important for services to become scalable, decentralized, and, most importantly, self-organizing.

Grid and P2P are both resource sharing systems having as their ultimate goal the harnessing of resources across multiple administrative domains. They have many common characteristics such as dynamic behaviour and heterogeneity of the involved components. Apart from their similarities, Grid and P2P systems exhibit essential differences reflected mostly by the behaviour of the involved users, the dynamic nature of Grid resources (i.e., CPU load, available memory, network bandwidth, software versions) as opposed to pure file sharing which is by far the most common service in P2P systems. Another essential difference results from the demanding nature of sensitive Grid applications that are time and data critical and have strict fault tolerance and security requirements as opposed to P2P applications which use commodity hardware and exhibit best effort behaviour.

Although Grid and P2P systems emerged from different communities in order to serve different needs and to provide different functionalities, they both constitute successful resource sharing paradigms. It has been argued in the literature that Grid and P2P systems will eventually converge. The techniques used in each of these two different types of systems will result to a mutual benefit. As the scale of Grid systems rapidly increases, centralized management will prove inefficient and other methods will have to be considered. The QoS constraints that currently govern most Grid applications will loosen up as Grids will move towards more popular and diverse application scenarios. Strict resource participation rules will be relaxed as participating organizations may need to have their infrastructure for own use at certain periods and for Grid jobs at other times and the use of commodity hardware will be allowed. On the other hand, P2P systems will open up to more sophisticated applications and they will have to support more complex queries and different QoS levels.

Recently several Grid systems have been proposed that incorporate techniques from the P2P paradigm in their resource discovery techniques. Many of these systems are based on the unstructured P2P approach while others borrow techniques from the large gamma of existing structured P2P systems (MAAN, NodeWiz, SWORD, Mercury, etc.).

Adaptability

Adaptability is understood as the ability of automatic adaptation of Grid systems to the changes in internal and the external system state. Also this keyword is frequently connected to the notion of *self-management* which is understood as a set of system abilities to perform configuration, performance tuning, recovery and other management tasks automatically [19,49]. Current research covers several aspects, including systems modelling and prediction, control-theoretical approaches, P2P-based self-organisation of distributed components, and different techniques to enable self-configuration in the software layers. Recently there is a trend to link the research within adaptability/self-management with the domain of dependability, as for example self-recovery from faults is a particular case of self-management.

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* [1,3,4,5,14] 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 [15], prediction of resource demand and automatic adaptation [16], and automatic recognition of resource models [17].

Another noteworthy research in the field of self-management originates from policy-based network management [15]. 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 [20] 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.

A special field of adaptability/self-management is devoted to modelling and prediction of system behaviour, as exposed by metrics like workload, performance change, resource usage and others. These approaches play an important part in scheduling (long-term: capacity planning, short-term: resource sharing [60] and in fault-resilience (anomaly detection, proactive software rejuvenation). Currently, no major Grid middleware utilizes tools for modelling and prediction of system characteristics. This situation is partially due to the lack of implementations of suitable algorithms. While frameworks for short-term prediction and scheduling support [20] for individual servers do exist, 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.

Dependability

One of the important issues to be solved in Grid Computing infrastructures is the support for fault-tolerance [42,46]. 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 [30,31,32,33], replication [9,11], group communication, reconfiguration, amongst other techniques [2,6,7,8,10,22] that have been proved suitable for small and medium scale installations, mostly characterized by a homogenous and stable environment.

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 [25,26,27,28,29,42,43,44]. Existing Grid middleware systems are 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 [35,36,37,38,39]. 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 a checkpoint 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 two most frequent causes of failures in Grids are due to configuration problems and middleware failures, followed by application errors and hardware outages [23]. 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.

Internet-based Grid computing has been receiving a wide importance, mostly in the quest for solutions for some grand-challenge scientific problems. Good examples are SETI@Home, LHC@Home, ClimatePrediction.net, Distributed.net, Einstein@Home, Predictor@Home, among several others. Some middleware tools like BOINC

and XtremWeb have been developed and are reaching maturity and acceptance among the community. However, these environments face some real problems that need to be solved by the research community, namely:

- a very low MTBF (Mean Time Between Faults) of the computing nodes which requires the mandatory use of fault-tolerance techniques;
- the scheduling techniques should take into account the high volatility of those computing nodes;
- the clear lack of autonomic capabilities from the Grid middleware tools;
- and the fact that open environments (like the Internet) are not trustable and very prone to attacks of malicious failures.

In order to reach full maturity it is necessary to devise and experiment some new techniques to increase the dependability of the desktop Grid applications, the robustness against malicious failures and the necessary mechanisms to provide a level of trust in desktop Grid environments. If the user does not trust in open desktop Grids they will not be used to run production codes. Also, if the Grid middleware is not dependable it will not be widely accepted.

4. Vision, Strategy and Roadmap

Vision and Scenarios

This 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, NGG2, and NGG3 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 Institute, and therefore should not be understood as a research programme of the Institute. However, it facilitates for each partner and for the Institute as a whole the selection of the research directions, prioritizing of competing research problems, and focusing on the most urging 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 in 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.). The 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, NGG2, and NGG3 Expert Group Reports have identified a set of properties of Grid systems required by the future application scenarios, such as described above. Within, the properties relevant to Grid system architecture are the following ones:

- ***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;
- ***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;
- ***resilient** with the ability to handle highly dynamic and unpredictable configuration of the network connecting the computing nodes:* similarly to the 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;
- ***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;
- ***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):

- self-adaptive, self-healing , self-managing and self-reconfiguring;
- scale-independent;
- 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 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.

Strategy

The principal procedures for bringing the research on architectures of Grid systems within the 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 Institute. The main tool here is bringing 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; Institute level meetings; cooperation within a common testbed.
- Collaboration on the SA Institute level, and between 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; Institute level meetings; “all-together-now” General Assembly meetings.

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

Task 4.1: P2P based service and resource discovery

Task 4.2: Self-organizing Grid services using P2P technology

Task 4.3: Dependability mechanisms for computational and data Grids

Task 4.4: Fault-tolerance and robustness

Task 4.5: Adaptive management of systems and resources

Task 4.6: Integration of the proposed methods

Task 4.7: Testbed Adaptation.

The above research tasks evolved from the three main research axes of this Institute, namely, scalability, adaptability, and dependability, as they evolved through the interaction of the participants and the coordination of their activities under the umbrella of the Institute on System Architecture during the first year of the project.

Roadmap

Phases of the roadmap

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

Phase 1: This part focused on gaining a common understanding of System Architecture problems, future requirements and capacities. The partners shared their views on the current state of Grid architectures and the important problems. During the organized meetings and Workshops presentations of individual and common research contributions were performed. Surveys on hot research topics were conducted. Particular focus lied in the domains of scalability, adaptability, and dependability.

Phase 2: This phase involves the enabling of the synergy effects between the partners research activities. 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 phase was performed during the first one and a half years of the project.

The **first two phases were completed** during the first year of the project and led to the second version of the roadmap on System Architecture. The three initially proposed research axes, namely, scalability, adaptability, and dependability, gave the opportunity to define focused research directions and to form specific research groups.

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 Institute;
- short visits between researchers, common tele-conferences, 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 is guided by the scientific vision of the 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. The Institute is currently undergoing this phase.

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 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 Institutes (by means of meetings, technical reports, and cross-collaboration between Institutes 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 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.

Task 4.1: P2P based services and resource discovery

Several P2P systems for resource discovery in Grid environments have been recently proposed. Such systems adopt different models and solutions, including structured or unstructured overlay networks, fully decentralized or super-peer architectures, and diverse strategies for improving routing performance and search precision. Moreover, they provide very different search capabilities, ranging from single-attribute search to multi-attribute and range queries. An important aspect that distinguishes Grid from P2P systems is the organization of resources. As opposed to P2P systems, large-scale Grids are generally built as federations of smaller Grids managed by diverse organizations. This organization-based architecture applies to most of the systems, in which typically one node per organization participates in the P2P network. Another important element in current Grids is the emergence of the OGSA and Web services as standard technologies. The OGSA model provides an opportunity to integrate P2P models in Grid environments since it offers an open cooperation model that allows Grid entities to be composed in a decentralized way. The general requirements for resource discovery are scalability, reliability, and support for dynamicity. Supporting very dynamic environments is fundamental, since the availability and status of resources within each node change dynamically over time. Another fundamental requirement in Grid systems is the ability to perform multi-attribute and range queries. With respect to scalability (both in time and traffic), structured systems perform better than unstructured systems, since Distributed Hash Tables (DHTs) are more scalable, self-organizing and load balanced than pure-P2P overlay networks. Another important advantage of DHTs is their ability to efficiently support range queries inherited from their data locality property. On the other hand, structured systems can be more difficult to maintain in very dynamic Grid environments, where the availability and status of resources vary significantly over time. For each resource in the system, the peer with the appropriate ID must be notified periodically, resulting to either increased traffic (if the period is too small), or stale information (if the period is too large). Unstructured systems, on the other hand, adopt diverse strategies to provide up-to-date results with limited network traffic, including experience-based query forwarding, message buffering and merging, routing indexes, and super-peer architectures. In particular, it has been demonstrated that the super-peer model is naturally appropriate to the organization-based nature of current Grids, ensuring limited network load and reduced response time with respect to pure-decentralized P2P systems. Both unstructured and structured systems show advantages and disadvantages. Hybrid approaches can be adopted to combine the efficiency of structured systems and the dynamicity of unstructured system, while overcoming their inherent drawbacks. For instance, structured protocols could be adopted for relatively static information, whereas unstructured approaches could be employed for more dynamic information. Moreover, the organization-based nature of Grids suggests the use of a super-peer architecture, in which different strategies (e.g. structured or unstructured protocols) may be adopted for intra-organization and inter-organization resource discovery. This task comprises two research groups as described below:

SA-1a: P2P techniques for resource discovery in Grids

This research group will perform a comparative study of various approaches that currently exist in P2P technology (structured vs. unstructured) and how these can benefit Grid research. Resource location or discovery is a key issue for Grid systems in which applications are composed of hardware and software resources that need to be located. Classical approaches to Grid resource location are either centralized or hierarchical and will prove inefficient as the scale of Grid systems rapidly increases. On the other hand, the P2P paradigm emerged as a successful model that achieves scalability in distributed systems. One possibility would be to borrow existing methods from the P2P paradigm and to adopt them to Grid systems taking into consideration the existing differences. Several such attempts have been made during the last couple of years.

SA-1b: Scalable resource location in P2P systems

The focus of this research group is to design a method for the reduction of the duplicate messages that are produced during the flooding process in unstructured P2P systems in order for flooding to become scalable. Lookup in unstructured P2P systems is mainly performed by having each node forward each incoming query message to all of its neighbours, a process called flooding. Although this algorithm has excellent response time and is very simple to implement, it creates a large volume of unnecessary traffic in today's Internet because each node may receive the same query several times through different paths.

Recommended Focus Areas

There are several issues that need to be addressed to achieve the ultimate goal of scalable Grid Computing. Within this task, we will be more concentrated in the following five topics:

1. Explore search and discovery techniques based on structured and unstructured Peer-to-Peer models.
2. Perform a qualitative comparison of the scalability of the resource discovery techniques used in existing Grid systems. Identify their weaknesses and propose concrete solutions
3. Adapt the P2P resource discovery techniques to Grid computing systems taking into consideration their basic differences.
4. Develop novel techniques that take into consideration the particularities of Grid architecture towards truly scalable resource and application independent discovery mechanisms.

Task 4.2: Self-organizing Grid services using P2P technology

Grid systems aggregate heterogeneous resources for specific tasks. However, Grid's existing centralized architectural solutions limit the scalability of Grid systems, and have to be abandoned for scalable, self-organizing designs that would allow these systems to scale. Grid systems can benefit from the P2P paradigm. Peer-to-Peer (P2P) systems have emerged as self-organizing, adaptive, autonomous, and scale-free distributed resource-sharing environments. Structured P2P overlay networks have sufficiently matured in the last few years that they can be considered as a basic part of an infrastructure for scalable distributed applications. What is lacking is that they do not provide any support for managing applications, e.g., deployment, versioning, and other kinds of configuration management. We will investigate how to build a component model over a structured overlay network, such that self-managing applications can be built on top of the resulting infrastructure. This work continues the work on the P2PKit infrastructure built in the PEPITO project and now being developed in the EVERGROW project.

More specifically, the basic objectives of this task are the following:

1. Proposition of self-organizing Grid services based on the P2P paradigm.
2. Investigating how to extend the abilities of existing component models over a structured overlay network resulting in self-managing applications.

The purpose of the proposed research task is to explore the experience gained from P2P distributed systems (which are self-organizing, adaptive, and scale-free) for the design of Grid services in general and also how to build a component model over a structured overlay network. This task includes the following research groups:

SA-2a: Building scalable self-organizing Services using P2P Technologies

This research group will contact research on mechanisms for self-organization and self-management of reconfigurable self-managing services (components) built on top of P2P networks that show a potential for building large-scale systems with autonomic management. Furthermore, research will be conducted on service models (service architecture) with self-* properties on top of P2P overlay networks. The model should enable development of large-scale autonomous distributed Grid services on top of P2P overlays. The outcome of this research group is expected to be a survey of existing approaches to building Grid services with self-* properties on top of P2P networks.

SA-2b: Self management for applications built on structured overlay networks

The main focus of this research group will be to perform an analysis of the self-organization abilities of structured overlay networks for low-level self-management and to provide higher level self-management functions such as self-configuration, self-update, etc. Furthermore, research will be conducted on a programming component-based model for composition of distributed services with self-* properties. The contribution of this research group is expected to be a mechanism for self-organization of reconfigurable self-managing services (components) build on top of P2P networks.

SA-2c: Scalability for desktop Grids

The primary purpose of this research group is to deal with the architectural and programming issues of desktop Grids and to develop an architectural framework that interconnects desktop Grids both in a hierarchical and symmetric way. This will enable large number of users to use the resources available on the desktop Grid system as well as, to build a large scalable desktop Grid from smaller organisational level desktop Grids as building blocks. This new architectural concept will significantly extend the current usability of desktop Grid systems.

The outcome of this research group is expected to be a decentralized BOINC architecture along with cluster and legacy services support in BOINC based desktop Grids.

Recommended Focus Areas

Several issues rest to be addressed in order to achieve the ultimate goal of a self managing Grid architecture. Some of them form the object of the research in this task and are the following:

1. Explore the experience gained from P2P distributed systems, which are self-organizing, adaptive, and scale-free, for the design of Grid services.
2. Contact a survey of existing approaches to building Grid services with self-* properties on top of P2P networks.
3. Analyse the self-organization abilities of structured overlay networks for low-level self-management.
4. Investigate how to build a component model over a structured overlay network, such that self-managing applications can be built on top of the resulting infrastructure.
5. Propose self-organizing Grid services based on the P2P paradigm.

Task 4.3: Dependability mechanisms for computation and data Grids

The main focus of this task is in the following three research topics:

- Dependability mechanisms for desktop Grid computing
- Sabotage tolerance in desktop Grid computing
- Self-healing SOA and Grid architectures

These three research topics will be directly mapped into three research groups that will involve several partners of the SA Institute and some other research groups from other workpackages.

SA-3a: Dependability mechanisms for Desktop Grid Computing

One of the major problems of desktop Grids is the volatility of computing nodes. A common solution to cope with the limitation imposed by resource volatility is checkpointing. It consists in periodically saving the application state into stable storage. When a failure interrupts a running task, the application can be resumed from the last available checkpoint reducing the recovery time. In other words, checkpointing is used in this case as a service to increase system's reliability.

An important issue regarding checkpoint lies in the place where it will be stored. An actual limitation of a middleware like BOINC is the fact that checkpoints are saved in the local disk of each computing node and they can only be used to resume the application in that same machine. This limitation has to be solved and we will study some techniques to improve the scalability and survivability of checkpoint data in large-scale systems, how to deal with the heterogeneity of systems and middleware, and how to adapt the checkpoint-and-recovery schemes to the paradigms of Grid applications.

An avenue of research that will be exploited will be the distribution of checkpoint images among the network for the sake of availability and scalability. In this case it seems promising to exploit the use of DHT-based and Peer-to-Peer techniques and to compare them with traditional hierarchical schemes for distributed storage.

Other related techniques should also be devised to complement the checkpointing services, namely: techniques for scalable failure-detection, reconfiguration schemes and new scheduling algorithms that should be taken into better account the mechanisms that will be introduced for the fault-tolerant execution of the applications.

SA-3b: Sabotage Tolerance in Desktop Grid Computing

Internet-based computing and desktop Grid must deal with sabotage and malicious failures that may undermine completely the results of a long-running computation. Some participants may behave in a malicious way to mislead a public computation or simply receive credits for computation they did not perform. Thereby it is of paramount important to devise techniques and strategies to cope with malicious participants and provide “*sabotage tolerance*” to Desktop Grid Middleware.

Together with the techniques for sabotage-tolerance it is mandatory to devise some protocols for trust management that should be adapted to these environments. If the computations that are performed in open environments are not trustable then Grid Computing will never be performed in those environments; only in closed clusters inside strict security domains.

The techniques to detect sabotage will provide some valuable information for the distributed maintenance of reputation lists among the federated services of a Grid environment. With this information there should be some high-level protocols that share cooperatively the information about trust and maintain an updated view about the reputation of the several participants.

The main goals of this work will be to identify the problems and to devise novel strategies for sabotage tolerance and trust-management in volunteer-based desktop Grid computing.

SA-3c: Self-Healing SOA and Grid Architectures

Our first step in studying the dependability of SOA-based architectures and Grid services is to see if existing SOAP implementations are prone to the problem of software aging, a phenomena that is observed in long-running applications where the execution of the software degrades over time leading to expensive hangs and/or crash failures. We do believe that SOA-based tools are highly prone to this problem due to the immaturity of the software and the inherent complexity.

Software aging happens due to the exhaustion of systems resources, like memory-leaks, unreleased locks, non-terminated threads, shared-memory pool latching, storage fragmentation, data corruption and accumulation of numerical errors. There are several commercial tools that help to identify some sources of memory-leaks in the software during the development phase. However, not all the faults can be avoided and those tools cannot work in third-party software modules when there is no access to the source-code. This means that existing production systems have to deal with the problem of software aging.

The natural procedure to combat software aging is to apply the well-known technique of software rejuvenation. Two basic rejuvenation policies have been proposed: time-based and prediction-based rejuvenation. It is understood that predictive rejuvenation provides better results, resulting in higher availability and lower costs.

Our final goal is to include some simple techniques inside a SOAP implementation that will be able to predict software aging problems and will take the correct actions (using selective software rejuvenation) to prevent failures to happen. This corresponds to the vision of self-healing systems, extensively described in the Autonomic Computing initiative, proposed some years ago by IBM. Our goal is definitely to contribute to more dependable implementations of SOAP-based middleware tools and Grid services providing means to predict failures due to software aging in order to trigger rejuvenation actions in an automatic way.

Recommended Focus Areas

The specific research issues that will be exploited towards a truly dependable computational and data Grid architecture are the following:

1. Study of techniques to improve the scalability and survivability of checkpoint data in large-scale systems, how to deal with the heterogeneity of systems and middleware, and how to adapt the checkpoint-and-recovery schemes to the paradigms of Grid applications.
2. Develop techniques for scalable failure-detection, reconfiguration schemes and new scheduling algorithms.
3. Identify the problems and to devise novel strategies for sabotage tolerance and trust-management in volunteer-based desktop Grid computing.
4. Combat software aging by applying well-known technique of software rejuvenation such as time-based and prediction-based rejuvenation.
5. Include simple techniques inside the SOAP implementations that will be able to predict software aging problems and will take the correct actions, using selective software rejuvenation, to prevent failures.

Task 4.4: Fault-tolerance and robustness

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. In this task, we focus on two main issues:

1. 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.
2. Proper *benchmarking* related to fault tolerance and stress testing should be provided, to accurately evaluate systematic fault-tolerant solutions provided in 1.

For both research issues, there are three main challenges that are to be taken into account:

1. *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, failure-correction and failure and stress benchmarking.
2. *Dynamicity*: 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. Fault-tolerant and dependability benchmarking should also to setup fault and stress patterns that emulate accurately very dynamic environments.
3. *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. 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.

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.

Currently, this task has three research groups that are detailed below:

SA-4a: Fault-injection and Robustness Assessment for Grid Services

One of the techniques to evaluate the effectiveness of those fault-tolerance mechanisms and the reliability level of the Grid middleware it to make use of some fault-injection tool and robustness tester to conduct some experimental assessment of the dependability metrics of the target system. We presented and reviewed several software fault-injection tools and workload generators for Grid Services that can be used for dependability benchmarking in Grid Computing. The basis for the envisioned dependability benchmarking and stress testing tool is based on two software currently being developed within CoreGRID: FAIL-FCI (for the fault injection part), and QUAKE (for stress testing). At this point, it is still unclear how those tools can be integrated with existing Grid middleware, as there exists no common API to various services. Also, scalability issues of the tools themselves are still to be resolved.

SA-4b: Fault-tolerant MPI

Job-batching systems like Condor 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. 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.

Recommended Focus Areas

There are several issues that need to be addressed to achieve the ultimate goal of Dependable Grid Computing. Within this task, we will be more concentrated in the following six topics:

1. *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. This failure model should be used afterwards by the fault-tolerance and dependability benchmarking tool to simulate realistic fault behavior in tested applications.

2. *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. Those dependability attributes will also constitute the common metrics that are to be output by the benchmarking scheme.
3. *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. Those mechanisms are to be evaluated by a common benchmarking tool.
4. *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.
5. *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 since the existing schemes still present some restrictions that should be solved.
6. *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.

Task 4.5: Adaptive management of systems and resources

SA-5a: Modelling and Prediction of Workloads and System Behaviour

Modeling and prediction of system characteristics such as resource demand of applications, workloads of servers or machine performance degradation has a multitude of applications within Grids. One of them is increased efficiency of resource sharing by allowing long-term capacity planning and by providing support for scheduling [60] (this kind of applications is investigated in Task 6.8 of the RMS Institute). Another exemplary application area is dependability, where prediction of performance degradation enables adaptive software rejuvenation, anomaly detection, or preventive migration of applications.

Among a variety of possible approaches for modeling and prediction, several approaches promise particular effectiveness and are targeted in the SA Institute. These approaches include:

- classical time series decomposition methods based on ARIMA approach [52] - used for short-term prediction horizon, especially for prediction benchmarking,
- classical data-mining classification approaches like decision trees or Support Vector Machines [59] - both for short-term and long-term predictions and exploitation of trace correlations,
- methods based on Fuzzy Logic in combination with Genetic Algorithms [61] - useful for dynamic environments with little training data.

Some of these methods need further development (especially [61]), and partially require “adjustment” for the particular application scenarios like predictive software rejuvenation. Furthermore, a comparative analysis in different application scenarios is required. In addition, supportive techniques are required, for example automatic classification of traces into predictable or “random” for early assessment of modeling benefits.

To allow efficient development, evaluation and deployment of these methods, a java-based framework is being developed at ZIB as a part of the SA Institute activities for adaptability. This framework will give opportunity to deploy the modeling and prediction techniques both in off-line (backtest) modus, as well as in real-time (production) modus. The modular architecture will allow for usage of different algorithms as components.

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. This work will happen in cooperation with other CoreGRID workpackages, particularly Task 6.8 of the RMS Institute.

This work will happen in close conjunction with the applications scenarios from the dependability domain, thus fostering the interconnection of the topics in the SA Institute.

SA-5b: Automated Configuration, Management and Fault-Recovery of Resources in Grids

Lack of automatic configuration and management of systems such as Grid infrastructures has negative impact on both the cost of the operations and the dependability. A partial remedy to this problem can be achieved by the approach of automated and adaptive generation of workflows composed of configuration, management and recovery activities.

The research issues involved in this problem should be studied (among others) on the example of automated construction and configuration of complex resources, such as multi-tier applications or virtual data centers. This example has been chosen due to an existing cooperation with industry, in detail HP Laboratories in Palo Alto [55].

These problems can be tackled by using declarative descriptions of current and target states, and also possible "actions" such as software installation, service composition, configuration or recovery tasks. To turn these descriptions into executable sequences of management steps, techniques such as automated planning should be used. This part will handle practical work on basic adaptability and dependability engines (e.g. simple rule-based self-healing engines, and/or automatic planners) for providing proofs-of-concept in real-world scenarios.

Within this research domain, we will also attempt to develop a user-friendly formalization "language" for describing

- system state (global "services" state and resources)
- possible configuration and management actions enabling system adaptation and fault resilience
- target system state, including both performance and dependability requirements.

Further advantage of such a language will consist in a unification of the models and approaches from the domains of system management and dependability. The concrete implementation of this specification language will be only possible if a common postdoctoral fellowship between two partners of the SA Institute (UCO and ZIB) will be available. Both measures target to strengthen the interconnections between the areas of dependability and adaptability with the SA Institute.

SA-5c: Performance-aware Grid Resource Selection

One of the key architectural characteristics of open large-scale Grid infrastructures is the heterogeneity of resources, which is due to the distributed ownership of Grid resources and the incremental approach that most institutions take when purchasing or upgrading their clusters. Nevertheless, the architectural paradigms used so far by the Grid community to develop mechanisms for resource selection, job scheduling etc., ignore this fact and follow the more traditional approach of massive-scale parallel system research, where algorithms and tools assume a model comprising many homogeneous processors that have identical characteristics in terms of performance, reliability, maintenance, etc. In this task, we plan to investigate the effects that different aspects of heterogeneity have on the quality of service (functionality and performance) provided by Grids to end-users. We will explore the development of high-level metrics of heterogeneity and apply our findings for the development of new algorithms for resource selection and job scheduling in heterogeneous Grids. The goal of our work will be to investigate quantitatively the potential impact that heterogeneity of Grids has on the performance and reliability provided to Grid end-users; furthermore, to come up with algorithms for the matching of Grid jobs to heterogeneous resources.

Grid heterogeneity can be seen as an abstract model with wide manifestations. Diverse heterogeneity models based on certain resource characteristics have been adopted by researchers for developing resource selection algorithms. We need to categorize these heterogeneity models and quantitatively characterize heterogeneity based on computational capacity, communication bandwidth, and usage policies. So that, to have better understanding for the design of optimal and performance-aware resource selection and scheduling algorithms. So far, we have completed the preliminary study of the concerned areas. Much of the efforts have gone into the identification and study of the related work which provided us the basis for our own work. Getting exposure of the Production Grid System such as, EGEE by running jobs and benchmarking experiments was necessary to be able to run experimental jobs at later stage. We are also evaluating different simulation tools to choose best for our experiments and to study the effects of resources heterogeneity on performance.

We are in the process of modelling heterogeneity and defining performance metrics to quantifying and analysis of the simulated results. Characterization of heterogeneity and defining performance metrics is essential part of our work. In detail, we categorize and characterise resource heterogeneity and then depending on type and extent of heterogeneity, the effects on the machine utilization and the average response time for the user or application

is analyzed based on user and system centric performance metrics. In the next phase, we will extend our measures of resource heterogeneity sensitive to Grid and application performance, to develop scheduling algorithms for the performance aware resource selection by taking into account the heterogeneity effects of involved resources.

Recommended Focus Areas

To enable the benefits of modeling and prediction of workloads in Grids, we recommend the following steps

1. evaluation of existing and extension of some methods for modeling and prediction, complemented with a comparative evaluation in a variety of application scenarios,
2. adaptation of the algorithms to the dependability-related scenarios such as anomaly detection, or proactive software rejuvenation in order to foster interconnection between adaptability and dependability domains in the SA Institute,
3. further development of the software evaluation framework and deployment in Grid middleware in cooperation with other CoreGRID tasks, particularly Task 6.8 of the RMS Institute.

The Automated Configuration, Management and Fault-Recovery of Resources in Grids will focus on the following aspects:

4. development of simple “execution engines” based on rules or automated planning for automated generation of actions for systems configuration, management, and recovery
5. focus on the scenario of complex resource construction in order to foster links with industry,
6. development of a description language for specifying dependability and adaptability requirements, available actions, and system models (only if a CoreGRID fellowship will be granted) .

Regarding the Performance-aware Grid Resource Selection we will focus on the following aspects:

7. modelling and characterization of heterogeneity based on computational power, communication bandwidth and usage policies,
8. development of user and system centric performance metric (average response time, system utilization) to quantify heterogeneity impact on performance,
9. development of performance aware resource selection algorithms by taking into account the impact of heterogeneity.

Task 4.6: Integration of the proposed methods

Testing, evaluation and benchmarking of the approaches in each of the areas covered in Tasks 4.1-4.5 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, the common experimental environment described in Task 4.7 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, reconciliation with the current state of the technology will be considered.

Task 4.7: Testbed adaptation

In order to reinforce the cooperation within the SA Institute and to obtain quantitative results of the proposed approaches and facilitate the integration of the prototypic components, Task 4.7 will collect, analyse and disseminate information describing testing and benchmarking activities, present and potential, of individual partners of the SA Institute. Using this information, SA Institute partners can discover other partners that conduct or intend to conduct similar testing and benchmarking work, and thus may share research interests, software solutions know-how, and might consider sharing of hardware resources for mutual interests. The partners that participate in this Task already have a research program involving testing and benchmarking, and are committed to actively exploit possibilities for joint testbed work. This Task will also seek close cooperation

with CoreGRID integrated activities, in particular on the CoreGRID integrated testbed. The task will gather experience made when operating testbeds of individual partners and publish this information through reports on CoreGRID BSCW server.

Research Groups

Research Activity	Task	Partners
SA-1a: P2P techniques for resource discovery in Grids	4.1	FORTH-ICS, INFN, KTH, SICS, UNICAL, VTT
SA-1b: Scalable resource location in P2P systems	4.1	FORTH-ICS, KTH, SICS, UCY
SA-2a: Building scalable self-organizing services using P2P technologies	4.2	FORTH-ICS, INRIA-GL, KTH, SICS
SA-2b: Self-management for applications built on structured overlay networks	4.2	KTH, SICS, UCL, VTT
SA-2c: Scalability for desktop Grids	4.2	INRIA-GL, MTA SZTAKI, UCO, UoW
SA-3a: Dependability mechanisms for desktop Grid computing	4.3	INRIA-GL, INRIA-Oasis, MTA SZTAKI, UCO
SA-3b: Sabotage tolerance in desktop Grids	4.3	SICS, UCO
SA-3c: Self-healing SOA and Grid architectures	4.3	INRIA-GL, PSNC, MTA SZTAKI, UCO, UoW
SA-4a: Fault-injection and robustness assessment for Grid services	4.4	INRIA-GL, UCO
SA-4b: Fault-tolerant MPI	4.4	INRIA-GL, UCO
SA-5a: Performance-aware Grid resource selection	4.5	KTH, UCY
SA-5b: Modelling and Prediction of Workloads and System Behaviour	4.5	INRIA-GL, UCO, ZIB

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.

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 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. Furthermore, proposals will be sent for the REP program (Researcher Exchange Program).

Inviting external people

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

Common experimental and benchmarking activities

Within the SA Institute, task 4.7 is devoted to the common experimental and benchmarking activities. 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 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, which includes the website and the BSCW server, 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 Institute for uncomplicated sharing and dissemination of documents such as meeting presentations, reports, roadmaps, scientific papers and software.

Future steps

The activities of this 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 Institute attempts to achieve this goal by the following mechanisms:

- creation of "unofficial" 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 Institute are thus intentionally directed at proposing new joint research projects.

5. Trust & security issues

With the advent of large-scale system architectures, such as Peer-to-Peer Systems and Desktop Grids that are devised in this Institute, the likeliness of faults and attacks (that can be seen as coordinated faults in some cases) can not be neglected anymore. Traditional mechanisms like authentication, access-control, encryption and non-repudiation (a service that provides proof of the integrity and origin of data) have to be applied to solve the traditional problems of security, but they are not effective to solve the problem of trust management in open-systems. Some remaining problems have to be investigated, namely:

1. **Sabotage tolerance:** Internet-based computing and desktop Grid must deal with sabotage and malicious failures that may undermine completely the results of a long-running computation. Some participants may behave in a malicious way to mislead a public computation or simply receive credits for computation they did not perform. Thereby it is of paramount important to devise techniques and strategies to cope with malicious participants and provide “*sabotage tolerance*” to Desktop Grid Middleware.

Together with the techniques for sabotage-tolerance it is mandatory to devise some protocols for trust management that should be adapted to these environments. If the computations that are performed in open environments are not trustable then Grid Computing will never be performed in those environments; only in closed clusters inside strict security domains.

The techniques to detect sabotage tolerance will provide some valuable information for the distributed maintenance of reputation lists among the federated services of a Grid environment. With this information there should be some high-level protocols that share cooperatively the information about trust and maintain an updated view about the reputation of the several participants.

Within the SA Institute, the topic of sabotage tolerance overlaps with the research area of dependability and to some part with adaptability. In particular, the research group SA-3b: “Sabotage Tolerance in Desktop Grid Computing” within the Task 4.3: “Dependability mechanisms for computation and data Grids” is wholly devoted to this topic. The goal of this group is to identify the problems and to devise novel strategies for sabotage tolerance and trust-management in volunteer-based desktop Grid computing, with focus on protocols for trust management and techniques to detect sabotage. Also the research group SA-5a: “Modelling and Prediction of Workloads and System Behavior” from Task 4.5: “Adaptive management of systems and resources” contributes to this topic by providing methods for anomaly detection.

2. **Sandboxing:** The security system must protect the participant computers from virus-like attack, including hardware alteration, configuration modification, personal files spying and worms introduction. A general approach to protect a computer running a program is to confine the code execution inside an unbreakable envelope. Sandboxing is a well known technique implementing this principle by filtering the system calls which appear to be the main security holes of recent operating systems. According to a security policy, sandboxing neutralizes hostile behaviours, enforces resource usage and prevents any attempt to exploit security holes on the host. Because it offer a complementary security mechanism providing a runtime control of the execution, sandboxing may appear as the cornerstone technology to allow a wide and safe use of Large Scale Distributed Systems (LSDS). Currently, there is no research activity within the SA Institute devoted to this topic.
3. **Coordinated attacks:** the large publicity around Distributed Denial of Service (DDoS) attacks made it well known that centralized servers (or set of servers, for what matters) are prone to such coordinated attacks. In principle, totally distributed solutions, as Peer to Peer systems, should be less harmed by distributed attacks on a single point of failure, as they are able to handle many incoming and outgoing participants. However, it is still unclear whether some distributed attack patterns could permanently turn down the system by targeting possibly moving different crucial components over time. Currently, there is no research activity within the SA Institute devoted to this topic.

6. Interaction with Industry

Following the CoreGRID Industrial Advisory Board meeting in June 2005, the interaction of this Institute with industry will be given an increased importance. While this activity does not form a dedicated task, it should be a part of regular work of the institute and taken into account while organizing meetings, searching for external

cooperations, attending integrated events, or seeing use cases. Also the members of the institute dedicated themselves to follow the IAB recommendations stated below.

With the CoreGRID Industrial Advisory Board being the most important channel of this Institute to initiate contacts with industry, we also seek and encourage direct collaboration of the research groups with industrial partners. The examples for such cooperations are:

- Several partners (ZIB, UCL, SICS) of the SA Institute proposed an EU- project “SelfMan” (which has been de-facto accepted) related to the areas of P2P (Tasks 4.1 and 4.2) and of adaptability/self-management, i.e. primarily Task 4.5 “Adaptive management of systems and resources”. Through the participation of the industrial partners France Telecom and ePlus we expected further cooperations with the Telecom industry.
- Within the research group SA-5a: “Modelling and Prediction of Workloads and System Behavior” from Task 4.5 we have cooperation with a software manufacturer for highly dependable software for mobile phone operators. This collaboration will provide the Institute with real-world use cases especially for dependability applications.
- The topic “Automated Configuration, Management and Fault-Recovery of Resources in Grids” has already an existing cooperation with HP Laboratories, Palo Alto which provides application scenarios and testing environments related to Utility Computing products and research of HP.

Another opportunity for a closer cooperation with industry is the proposed collaboration with the NESSI initiative following the recent CoreGRID-NESSI Workshop, 27 January 2006 in Brussels.

In addition, the members of the SA Institute will actively seek to follow several of the recommendations of the IAB on a continuous basis while deciding on common activities, research agenda, and use cases. These recommendations are:

- o to estimate of the economic value of the research done in the project
- o to identify major areas of interest and current market trends
- o to ensure reusability of the research results
- o to establish more explicit interactions and connections amongst the 6 scientific workpackages
- o to request industrial members for business cases.

7. Links with other CoreGRID Institutes

The role of the Institute on System Architecture in respect to other CoreGRID institutes is to provide state-of-the-art approaches for architecture related problems. Since architecture pertains many aspects of the Grid development issues, our role is understood as a provider of “building blocks” within the areas of dependability, P2P / scalability, and adaptability/ self-management.

On the other hand, the SA Institute seeks input from other Institutes in order to create a higher quality of results. These inputs include, but are not limited to:

- **from the Intergated Activities (WP1):** Grid testbed "sandbox" and interfaces for inclusion of mechanisms for scalability, adaptability and dependability,
- **from the KDM Institute:** Services for processing Grid state information and Data Mining tools for discovering activity/failure patterns. Furthermore, possibility to work on the problem of storing the checkpoints within the Grid infrastructure,
- **from the PM Institute:** Interfaces (possibly component-based) between programming model and architectural mechanisms,
- **from the IM Institute:** Services for Grid monitoring and interfaces for controlling resources,
- **from the RMS Institute:** Fault-tolerant scheduling methods; application scenarios for prediction and modelling in scheduling (Task 6.8)
- **from the STE Institute:** Platform-independent system-level interface model.

This institute seeks to receive and transfer the knowledge to other Institutes via overlapping membership of partners in different Institutes, co-location of workshops and meetings, common fellowships / REPs and via centrally organized integration activities (mainly integration workshop). A further direct way to work with other Institutes is a direct collaborations between the SA Institute research groups and partners outside the SA

Institute, yet which participate strongly in other CoreGRID institutes. Currently, such cooperations between research groups and “associated” non-SA Institute partners include:

- cooperation KDM – SA Institutes via the institute INFN in the research group SA-1a: "P2P Techniques for Resource Discovery in Grids"
- cooperation RMS – SA Institutes via PSNC in the research group SA-3c: “Self-Healing SOA and Grid Architectures"
- cooperation IM – SA Institutes in terms of checkpointing in desktop Grid environments.

Further examples of cooperation activities with other Institutes are:

- cooperation KDM – SA Institutes via a common fellow on topic "Peer-to-Peer Models and Services for Scalable Grid Computing",
- cooperation RMS – SA Institutes on resource workload prediction (in the SA Institute: Task 4.5, in the RMS Institute: Task 6.8).

For the remaining duration of the CoreGRID project, closer ties with the PM Institute (WP3) especially regarding the component model in relation to the SOA are planned. Also more intense work with the WP7 is intended, with the purpose of incorporation of SA Institute results into Grid tools, and for getting closer feedback on realistic scenarios and problems from the users. The first step for this cooperation are co-located Institute meetings of WP3, WP4 and WP7 in June 2006.

8. References

1. A.G. Ganek, T.A. Corbi. The dawning of the autonomic computing era. IBM Systems J., March 2003.
2. P. Koopman, H.Madeira. Dependability benchmarking & prediction: A grand challenge technology problem. Proc. 1st IEEE Int. Workshop on Real-Time Mission-Critical Systems: Grand Challenge Problems, Arizona, USA, Nov 1999.
3. A. Brown, J. Hellerstein, M. Hogstrom, T. Lau, S. Lightstone, P. Shum, M.P. Yost. Benchmarking autonomic capabilities: Promises and pitfalls. Proc. Int. Conf. on Autonomic Computing (ICAC'04), 2004.
4. S. Lightstone, J. Hellerstein, W. Tetzlaff, P. Janson, E. Lassetre, C. Norton, B. Rajaraman and L. Spainhower. Towards benchmarking autonomic computing maturity. Proc. 1st IEEE Conf. on Industrial Automatics (INDIN-2003), Canada, August 2003.
5. A. Brown, C. Redlin. Measuring the effectiveness of self-healing autonomic systems. Proc. 2nd Int. Conf. on Autonomic Computing (ICAC'05), 2005.
6. J. Durães, M. Vieira and H. Madeira. Dependability benchmarking of web-servers. Proc. 23rd Int. Conf. (SAFECOMP 2004), Potsdam, Germany, September 2004. Lecture Notes in Computer Science, Vol. 3219/2004.
7. K. Gross, W. Lu. Early detection of signal and process anomalies in enterprise computing systems. Proc. 2002 IEEE Int. Conf. on Machine Learning and Applications, (ICMLA'02), June 2002.
8. P. Sobe. Stable checkpointing in distributed systems without shared disks. Proc. Int. Parallel and Distributed Processing Symposium (IPDPS'03), 2003.
9. J.R. Douceur, R.P. Wattenhofer. Large-scale simulation of replica placement algorithms for a serverless distributed file system. Proc. 9th IEEE MASCOTS, 2001.
10. L. Sarmenta. Sabotage-tolerance mechanisms for volunteer computing systems. Proc. 1st Int. Symposium on Cluster Computing and the Grid, 2001.
11. D. Szajda, B. Lawson, and J. Owen. Toward an optimal redundancy strategy for distributed computations. Proc. IEEE Int. Conf. on Cluster Computing (Cluster 2005), Boston, MA, USA, 2005.
12. V. Lo, D. Zappala, D. Zhou, Y. Liu, and S. Zhao. Cluster computing on the fly: P2P scheduling of idle cycles in the Internet. Proc. 3rd Int. Workshop on Peer-to-Peer Systems (IPTPS 2004), 2004.
13. D. Stainforth, A. Martin, A. Simpson, C. Christensen, J. Kettleborough, T. Aina, and M. Allen. Security principles for public-resource modelling research. IEEE Computer Society, 2004.
14. IBM. Autonomic computing manifesto, October 2001.
15. R. Haas, P. Droz, and B. Stiller. Autonomic service deployment in networks. IBM Systems Journal, 42(1):150–164, 2003.
16. 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.
17. 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.
18. 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.
19. 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.
20. P. Kacsuk, G. Dzsa, 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.
21. 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.
22. F.Gartner. “*Fundamentals of Fault-Tolerance Distributed Computing in Asynchronous Environments*”, ACM Computing Surveys, 31 (1), March 1999
23. 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
24. 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
25. 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

26. M.Baker, G.Smith. "GridRM: A Resource Monitoring Architecture for the Grid", Technical Report University of Portsmouth UK, June 2002
27. 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
28. 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
29. 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
30. M.Livny, J.Pruyne. "Managing Checkpoints for Parallel Programs", Proc. IPPS Second Workshop on Job Scheduling Strategies for Parallel Processing, 1996
31. 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.
32. S. Krishnan, D. Gannon. "Checkpoint and Restart for Distributed Components in XCAT3", Proc. 5th IEEE/ACM International Workshop on Grid Computing, Nov 2004.
33. GGF Grid Checkpoint Recovery Working Group, <http://gridcpr.psc.edu/GGF/>
34. 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
35. 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.
36. G.Fagg, A.Bukovsky, J.Dongarra. "Harness and Fault-Tolerant MPI", Parallel Computing, Vol. 27, No 11, pp. 1479-1495, October 2001
37. 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.
38. 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
39. 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
40. 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
41. 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.
42. Weissman. "Fault-Tolerant Computing on the Grid: what are my options?", Proc. 8th IEEE International Symposium on High Performance Distributed Computing, August 1999
43. G. Kola, T. Kosar, M. Livny, "Phoenix: Making Data-intensive Grid Applications Fault-tolerant", In Grid 2004, Pittsburgh, PA, November 2004
44. GMA-WG: Grid Monitoring Architecture Working Group, <http://www-didc.lbl.gov/GGF-PERF/GMA-WG/>
45. 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
46. 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
47. 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
48. 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/>
49. 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.
50. F. Schintke, A. Reinefeld: Modeling Replica Availability in Large Data Grids, Journal of Grid Computing, 1(2):219-227, 2003.

51. Seif Haridi, Peter Van Roy, Per Brand, Christian Schulte, *Programming Languages for Distributed Applications*, New Generation Computing, 16(3):223-261, 1998.
52. 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.
53. Distributed Management Task Force (DMTF). *DMTF CIM Concepts White Paper*. http://www.dmtf.org/standards/published_documents.php
54. 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.
55. A. Andrzejak, U. Hermann, A. Sahai, "FEEDBACKFLOW - An Adaptive Workflow Generator for System Management", 2nd IEEE International Conference on Autonomic Computing (ICAC-05), Seattle, June 2005.
56. 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
57. 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
58. 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.
59. A. Andrzejak, P. Domingues, and L. Silva, "Classifier-based Capacity Prediction for Desktop Grids", Integrated research in Grid Computing - CoreGRID workshop, Pisa, Italy, 2005.
60. P. Domingues, A. Andrzejak, and L. Silva, "Using Checkpointing to Enhance Turnaround Time on Desktop Grids", submitted to Supercomputing 2006.
61. A. Andrzejak and S. Plantikow, "Defining Fuzzy Logic Controllers with Scarce Model Data", submitted to First International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks (FeBID'06), 2006.

9. Participants

Partner Name	No.	Researchers	T4.1	T4.2	T4.3	T4.4	T4.5	T4.6	T4.7
FORTH-ICS	11	Paraskevi Fragopoulou Evangelos Markatos Charis Papadakis Elias Athanasopoulos	x					x	x
INRIA	14	Franck Cappello Gilles Fédak Thomas Héroult William Hoarau Derrick Kondo Olivier Peres Benjamin Quetier Ala Rezmerita Sébastien Tixeuil J.B. Stefani N. de Palma S. Sicard C. Taton Denis Caromel Alexandre Di Costanzo Christian Delbe			x	x	x	x	x
KTH	15	Ali Ghosti Seif Haridi Vladimir Vlassov	x	x			x	x	x
MTA SZTAKI	20	Zoltán Balaton (T4.2) Gábor Gombás József Kovács		x				x	x
SICS	19	Konstantin Popov Ali Ghodsi Per Brand Seif Haridi	x	x				x	x
UCL	31	Peter Van Roy Raphael Collet Kevin Glynn Boris Mejias	x	x	x			x	x
UCO	27	Luis Silva Patricio Domingues Paulo Marques Joao Silva Filipe Araujo Bruno Cabral			x			x	x
UCY	28	Athina Stassopoulou Eleni Tsiakkouri George Tsouloupas Marios Dikaiakos Wei Xing Hassan Rasheed		x	x			x	x
UNICAL	23	Domenico Talia Paolo Trunfio Carmela Comito	x					x	x

UoW	37	Henrio Ludovic J. Thiyagalingam Stavros Isaiadis Vladimir Getov			x		x	x	x
VTT	40	Janne Väre Kimmo Ahola Mika Pennanen Mikko Alutoin Pertti Raatikainen Sami Lehtonen	x	x				x	x
ZIB	41	Alexander Reinefeld Artur Andrzejak Felix Hupfeld Florian Schinkte Thomas Röblitz Thomas Steinke	x				x	x	x

Table 1: Participants of the Institute on System Architecture (WP4)