Project no. 034567

Grid4All

Specific Targeted Research Project (STREP)

Thematic Priority 2: Information Society Technologies

Deliverable D5.2 – First integrated intermediate prototype and evaluation report

Due date of deliverable: 1 December 2008

Actual submission date: 20 March 2009

Start date of project: 1 June 2006 Duration: 36 months

Organisation name of lead contractor for this deliverable: FT

Contributors: Jean-Michel Busca, Alicia Bou Bayona, Leandro Navarro, Joan Manuel Marqués Puig, Georgios Tsoukalas, George Vuoros, Vladimir Vlassov, Ahmad Al-Shishtawy

Editors: Ruby Krishnaswamy, Daniel Stern

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	X
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Contents

1. Executive Summary

This document is part of a research project partially funded by the IST programme of the European Commission as project number IST-FP6-034567.

The WP5 is responsible for integration, test, and evaluation of software modules produced within the project.

D5.4 has described the planned use scenarios and the integrations that need to be completed to illustrate these scenarios. The present document reports the integrations that have been performed and a first level of feedback. Qualitative evaluations have started, either on integrated software or on specific results. A first hand feedback is also provided within this document. Further feedback and analysis on the feedback will provide more precise indications on improvements.

Besides the "integrated" and demonstrated results, quantitative evaluations have been performed on the most important modules. Criteria (mostly performance and correctness) are valued that show generally an adequation with the Grid4all objectives.

2. Introduction

The object of this document is to summarize the 1st evaluation of the Grid4all results.

The evaluation has been performed in several ways but, as far as possible, we consider evaluation of integrated results, rather than evaluation of isolated modules. The term integration means here that several modules have been appropriately combined so as to constitute a demonstrator. Besides these qualitative evaluations performed on integrated results, quantitative evaluations have been performed on modules.

We have grouped at chapter 3 the results of qualitative evaluations and at chapter 4 the results of quantitative evaluations.

D5.4b had described the G4all modules and their principal "elementary" use cases. As a reference we recall in table below the list and a short description of the modules.

Domain	Modules & Services	Functionality
VO Management: administrate a VO, its members, its resources	Membership Manager	Authenticate members Maintain member info
	Resource Manager	Discover resources Allocate resources Donate resources Monitor resources
	Deployment Manager	Deploy application
	Reservation Manager	Reserve resources
Inter-VO services: allocate or offer leases for computational resources by brokering at resource markets; maintain and advertise information on resources	Market Information Service	Query information Subscribe
	Market factory	Select market Deploy market
	Currency and payment manager	Transfer currency
	Agreement Manager	Settle agreement Distribute agreement
	Negotiator	Interact with markets
	Auction service	Register Treat asks/bids
Security: manage security	Identity manager (VOMS)	Sign-on and create a proxy certificate with attributes
	Policy Administration Point Policy Repository Policy Editor	Create/edit/store VO policies by VO admin
	Policy Enforcement Point Policy Decision Point Policy Information Point	Access to a VO resource(s) protected with a PEP(s)
	PEP, PDP, PR, PIP, PAP	Set/view ACL for a VO resource. e.g. VOFS directory, by a application
Collaborative & Federative services: manage VO-wide file systems; provide support for collaborative editing of shared documents	Telex	Create/read/write/delete collaborative documents
	VOFS Manager	Maintain a VO-wide workspace
	VOFS User Agent	Maintain VOFS membership
	DFS File Server	Serve user files
	VBS Storage Server	Serve User storage
Execution Management	Scheduling Service	Computes schedules
	XtremWeb Server	Manages distributed execution
	XtremWeb Worker	Performs local computation
Overlay Services – DCMS	DCMS	Executing Self-* Component-Based Applications
Information Services	Matching Service Selection Service	Register service request/offers,
Application: support users to perform tasks within a VO	Application manager	Collaborative File Sharing Participate in a forum Collab. Netw Simulation Shared Calendar eMeeting GMovie

3. Qualitative Evaluation Results

3.1 Introduction

In order to test and prove the integration of the modules, several demonstrators have been designed and implemented. As demonstrators they do not comply to quantitative constraints but they must illustrate how fundamental Grid4all objectives have been attained:

- an easy setup and use,
- a reduced management and administration complexity,
- an access to collaborative tools and applications,
- the execution of applications drawing on resources available on the Internet.

These demonstrators are:

- Programming self-managing distributed applications with DCMS
- Shared calendar using Telex middleware that uses the multilog abstraction provided by VOFS
- Collaborative File Sharing with Telex
- Federative VO workspaces with VOFS
- Advertisement and discovery of services using SIS
- gMovie execution using Scheduler and core VO services
- Allocation of resources on-demand to execute applications

In section 6 we summarize each demonstrator and its qualitative evaluation.

3.1 Evaluation of integrated demonstrators

3.1.1 Programming self-managing distributed applications with DCMS

Name	Programming self-managing distributed applications with DCMS
What is achieved in this integration	DCMS is used in several applications being currently developed at SICS, KTH, and France Telecom R&D
What is the rationale of this integration	DCMS facilitates development of self-managing applications for volatile environments like Grid4All
Modules involved	DCMS
Scenario	Three self-* applications are being developed and evaluated: a skeleton storage service with replication that only mimics storing real data, a compute service, and also a full-fledged storage service with replication capable of storing real data. Services can self-repair and self-configure according to service load, and use an external resource management service.
Environment assumptions	DCMS infrastructure running on VO member computers with VO resource mgmt
Lessons learnt	Positive: services are fairly straightforward to develop with Fractal and DCMS once the steep section of the learning curve is behind Negative: DCMS platform stability and documentation should be improved
Comparative (innovation as compared with other solutions)	DCMS is a novel tool that provides facilities to build efficient distributed self-management control loops for component-based applications
Proposed improvements Will they be ready before end of project?	Stability and documentation will be improved. Reliability of self-management through replication of elements of control loops will be studied and prototyped.

3.1.2 Shared calendar using Telex middleware on top of VOFS

Name	Shared calendar using Telex middleware on top of VOFS
What is achieved in this integration	This integration demonstrates: (1) the detection and the resolution of elaborate meeting conflicts through Telex, (2) the persistent storage and replication of application data through VOFS
What is the rationale of this integration	This integration exercises the complete software stack of Grid4All's data services.
Modules involved	Shared calendar application, Telex middleware, VO File System
Scenario	Three users schedule conflicting meeting while being off-line. (Some meetings, e.g. training sessions, are grouped in an all-or-nothing fashion.) When users reconnect, the shared calendar application reports conflicts and proposes solutions to users.
Environment assumptions	Shared calendar (including the Telex library) and VOFS software installed on nodes
Lessons learnt	Positive: seamless integration between Telex and VOFS Negative: Telex code is not stable (although API is), documentation should be improved.
Comparative (innovation as compared with other solutions)	Unlike existing solutions (e.g. Google calendar, Lotus Notes) Grid4All's shared calendar provides disconnected operation, conflict detection and resolution, in a peer-to-peer environment
Proposed improvements Will they be ready before end of project?	The commitment of a given conflict solution across users is not implemented yet. This is mostly a functional issue (which user is allowed to decide on a given meeting), which will be addressd by the end of the project.

3.1.3 Collaborative File Sharing with Telex

Name	**Collaborative File Sharing with Telex**
What is achieved in this integration	This integration demonstrates the detection of conflicts between file move operations and the consistent resolution of the conflicts across all sites
What is the rationale of this integration	This integration exercises Telex's distributed semantic commitment protocol
Modules involved	Collaborative File Sharing application, Telex middleware
Scenario	3 end users create a shared workspace, create folders, share files and discuss on their content.
Environment assumptions	CFS application (including the Telex library) installed on nodes
Lessons learnt	Positive: correct integration between Telex and CFS Negative: Telex code is not stable (although API is), documentation should be improved.
Comparative (innovation as compared with other solutions)	Conflict detection and resolution is a key feature to support collaborative work using shared workspaces in a decentralized environment where students form ad-hoc groups and networks. Disconnected operation is another useful feature although it could not be integrated on CFS as it was not ready on time during the development..
Proposed improvements Will they be ready before end of project?	The development ended in October 2008 (the main developer left the group), so no further improvements will be implemented.

3.1.4 Federative VO workspaces with VOFS

Name	**Federative VO workspaces with VOFS**
What is achieved in this integration	VOFS users can enable access control to their workspaces according to G4A security policies
What is the rationale of this integration	User identities, resources and access policies are prepared by the VO authorities to help VO members co-operate. This framework should also apply to VOFS for VO workspaces instead of requiring a separate security and access policy system.
Modules involved	VOFS, the Grid4All Security
Scenario	A user contributes some files to the workspace and instructs VOFS to grant access to all VO members. The user doesn't have to care about VO members because membership and authentication is performed by the Grid4All security and is administrated by VO authorities.
Environment assumptions	Grid4All Security and VOFS software
Lessons learnt	Positive: Supporting security checks via a generic filtering module is powerful and flexible Negative: Security via generic filtering makes editing security settings difficult as different modules are set up differently and a unified security configuration interface cannot be provided.
Comparative (innovation as compared with other solutions)	Contrary to most peer-to-peer and distributed systems, users retain strong control over their resources even though they can choose to use external modules for security. Modules may be enabled and disabled at will and only for subsets of the user's file hierarchy.
Proposed improvements Will they be ready before end of project?	It will be convenient and useful to simplicity and security to specify a minimal set of security settings that every module must implement. This allows users to configure basic security settings (e.g. allow everyone, read-only, strictly private) with a uniform interface through VOFS. This might not be available before the end of the project.

3.1.5 Advertisement and discovery of services using SIS

Name	**Advertisement and discovery of resources in SIS**
What is achieved in this integration	Demonstration of Semantic information Service functionality on advertisement and querying for resources.
What is the rationale of this integration	To assess the acceptance the SIS API.
Modules involved	Semantic Information Service
Scenario	For service advertisement: User registers offers in OWL-S format as follows: the WSDL service specification must be annotated (either automatically or manually) and be translated to OWL-S profile specification. This OWL-S profile specification is being registered in SIS. For service query: 1. User submits a query based on requested service I/O specifications. 2. System returns an ordered set of services. 3. User selects a service from the set of results.
Environment assumptions	None
Lessons learnt	Positive: –Automatic semantic annotation of services –Seamless registration of services using their data semantics –Market-oriented discovery of resources based on their semantics –Strong Data Consistency –Queries of “high” complexity can be answered –Multi –Attribute and Range Queries (for resources, markets and orders) –Support for Continuous Queries –Ranking of resources based on agents preferences. Negative: –Poor scalability due to being centralized –Mild performance due to the overhead imposed by the reasoning required –Dependence of API from the Grid4All resources' ontology (the API provides abstractions of the existing ontology classes)
Comparative (innovation as compared with other solutions)	Exploitation of an ontology which describes market-oriented and order-oriented attributes of resources. Integrated automatic semantic annotation of services Integrated ranking of resources based on agents preferences.
Proposed improvements **Will they be ready before end of project?**	Distributed design in order to improve scalability Yes

3.1.1 gMovie execution using Scheduler and core VO services

Name	**gMovie execution using Scheduler and core VO services**
What is achieved in this integration	How to plan execution of bag-of-tasks applications (gMovie) and manage their execution.
What is the rationale of this integration	Demonstrate API for provisioning of computational resources (allocation, deployment and life-cycle management). Innovative usage of offline scheduling service to provide estimates of completion time and consequently refine quantity of needed resources.
Modules involved	Scheduling Service, Reservation Management, Jade deployment tool
Scenario	User wants to transcode a movie; Scheduler computes different time-budget pairs; User chooses the transcoding with the pair satisfying his objectives; Reserve compute nodes on which the transcoding application can be executed. Deploy XtremWeb using Jade. Movie is transcoded. Transcoded movie is displayed
Environment assumptions	Grid infrastructure (XtremWeb OR Hadoop) installed on the nodes
Lessons learnt	Positive: Jade can be used to deploy legacy applications. Negative: Jade technology is not stable. Wrapping non-java applications is not supported.
Comparative (innovation as compared with other solutions)	Many systems such as XtremWeb, Hadoop offer support for execution of bag-of-task applications. Hadoop does not offer specific support to allocate new resources and add it to the worker pool serving the application. Hadoop does not offer specific deployment support. Applications (used in Map and reduce functions) are expected to be pre-installed on work nodes.
Proposed improvements Will they be ready before end of project?	Integrate with the real implementation of Reservation Management that will lease resources at resource market places (will be done before end of project) Develop an self-managing execution management service using DCMS (will not be done before end of project) Support for deployment of non-java applications (will not be done).

3.1.2 Allocation of resources on-demand to execute applications

Name	Allocation of resources on-demand to execute applications
What is achieved in this integration	Usage of Reservation Management API by applications to lease resources.
What is the rationale of this integration	In a self-managing application scenario, applications executing in a VO should have access to APIs that allow them to allocate computational resources on-demand in a manner similar to utility computing.
Modules involved	DCMS, Deployment Manager, Reservation Manager, Market Information Service, Auction Service implementing K-DA.
Scenario	Client (an application manager) requests the Reservation Manager to lease resources. It sets the resource properties, the quantities, the start, end time and the duration for which the resources are required. Reservation Manager deploys a buyer agent that finds a matching auction. Buyer agent formulates the bid and submits to the auction. Buyer agent receives Agreement. The Agreement specifies the Provider information such that the allocated remote resources can be joined to the host VO.
Environment assumptions	All nodes installed with minimal Jade container. One node (part of market place infrastructure) installed with market place components.
Lessons learnt	Positive: Reasonably smooth API level integration between the different components. Negative: Oscar (OSGi implementation) package dependency management is complex. A number of improvements are required (see below) before the result can be exploited.
Comparative (innovation as compared with other solutions)	The innovation lies in the overall integrated capability to show that self-managing applications executing within a VO may grow by leasing computational resources. Currently resource market-places (to broker between multiple providers) do not exist and hence cannot be compared.
Proposed improvements Will they be ready before end of project?	These proposed improvements will not be ready before the end of project. Intelligent budget management: within the demo budget is associated to each resource request. This is not realistic in a real VO deployments. Budget should be managed over the larger time-span, e.g., lifetime of an application, a set of applications, or for the VO itself. Reservation Manager should correctly support allocation requests satisfied by resources allocated at different times. Market factory to facilitate auction deployment and to eagerly set up auctions. Flexible trading (buyer and seller) agent frameworks to implement different consumer and provider policies.

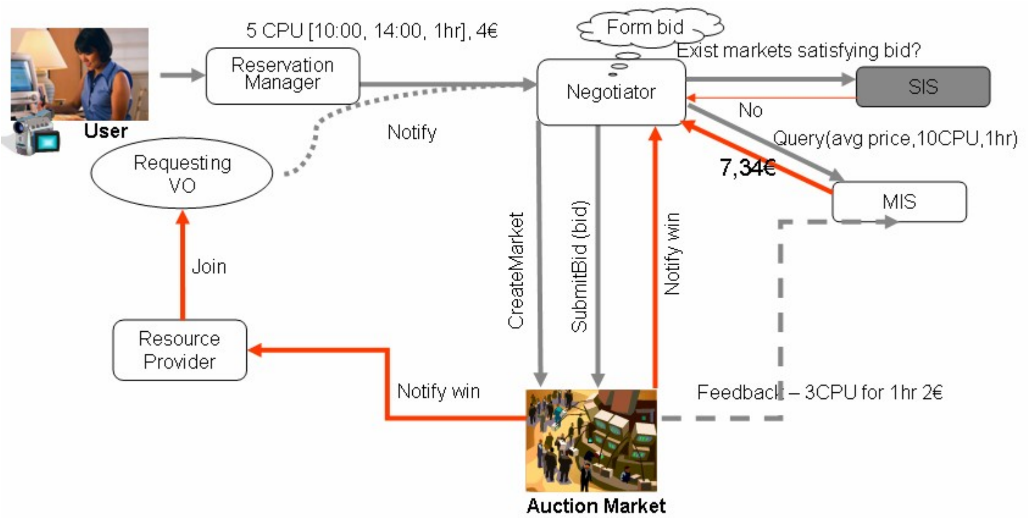

Figure 1:Allocation of resources on-demand demonstrator (grey modules are not utilized)

3.2 Early feedback from qualitative Evaluations

3.2.1 Telex

Giorgos Santimpantakis, master student of the University of the Aegean, developed a prototype of a Telex-based system that assists the collaborative development and evolution of ontologies. He reports on his experience in using Telex.

The theoretical background of Telex, the Action-Constraint Framework, is general enough to support the application's requirements. Several enhancements, however, are suggested to improve Telex's functionality: (1) Applications should be able to roll back to some *previously committed state* of a document. As a consequence, garbage collection of actions and constraints should not be automatic but rather driven by the application. (2) In the ontology editing system, the cause of a conflict may lay in previously-executed actions. As a help to users, Telex should provide some means to identify the sequence of actions that lead to a conflict. Finally, Telex documentation should be improved: a tutorial, together with a streamlined sample application are needed; configuration files should be commented.

3.2.2 VOFS

These items were collected from interaction with users during presentation, demonstrative use and discussion of VOFS. No items were collected after extended private use of VOFS by the users, therefore they only represent first-contact comments and concerns of users, not actual usage results.

On the positive side, the users considered the ability to easily access other people's files as basic and no-one was conceptually surprised. This made VOFS functionality welcome but somewhat expected. Users acknowledged the necessity of this functionality in their everyday computer interaction.

On the negative side, several issues emerged. First, some users were not familiar with VO and basic Security concepts (such as a certificate). Another issue is that users not familiar with distributed systems could not discern the boundaries of their own peer, resources and their own control.

Conventional file managers make different assumptions and display a behaviour that is unfamiliar.

A transfer queueing system may be required for some scenarios involving large files or slow connectivity. Concerns over no control over extended transfers (e.g. cancelling, prioritising) were voiced by users.

4. Quantitative Evaluation Results

4.1 VOFS

Name	VOFS
Test objective	Assess the performance of VOFS
Testing steps	Alternate simplified evaluation of performance with tuning cycles until performance is satisfactory and then evaluate and report it in detail
Comparative	Local disk file system
CURRENT STATUS	Evaluating performance in a storage back-end tuning cycle. VOFS is currently 3 to 10 times slower than a local file system on average and consumes more CPU time. (note: numbers are approximate and illustrative)
NEXT STEPS	Evaluate performance in a networking scalability tuning cycle, final evaluation.

4.2 DCMS + Jade

Name	DCMS infrastructure performance
Test objective	Assess the performance of underlying DCMS for application deployment
Testing steps	Communication components perform series of lookups for systems of varying size
CURRENT STATUS	The lookup time scales logarithmically with the system size, as desired. Actual lookup times vary greatly between different environments, ranging from less than 10 ms on Grid5000 where a direct send is less than 2 ms, to 400 ms in the much slower PlanetLab environment where a direct send takes up to 100 ms.
NEXT STEPS	Complete development of evaluation framework

4.3 Telex

Name	Telex – Scheduling correctness
Test objective	Check that schedule generation is correct
Testing steps	Several representative cases (i.e. action-constraint graphs) are manually generated. For each test case, the perfect list of schedule is manually computed. A tool compares the list of generated schedules against the correct list of schedules.
Comparative	Correct list of schedules manually computed
CURRENT STATUS	Completed. An additional tool generates random graphs and checks that (i) each output schedule is correct, (ii) schedules are output by decreasing order of quality (ii) the list of schedules is exhaustive.
NEXT STEPS	N.A.

Name	Telex – Reconciliation correctness
Test objective	Check that replica reconciliation is correct
Testing steps	Several sets of per-node actions and constraints are randomly generated. After execution, a tool compares the committed schedules at each node.
Comparative	Ideal committed state obtained when committing all actions and constraints at once after they have been propagated to all nodes
CURRENT STATUS	Completed.
NEXT STEPS	N.A.

Name	Telex – Performance and scalability
Test objective	Assess the performance and the scalability of Telex w.r.t. the number of nodes
Testing steps	The micro-benchmark is run with various action-constraint graph sizes (see above) The STMBench7 benchmark is run in various setting (see above)
CURRENT STATUS	Most of the local, low-level metrics have been measured, Based on the shared calendar application: –disk occupation: ~5KB/meeting –memory consumption: ~50KB/meeting –first schedule computation time: ~100ms for10,000 meetings in memory
NEXT STEPS	Measure global, high-level metrics: (first site) convergence latency, communication complexity) and assess scalability w.r.t. number of nodes.

4.1 SIS

Name	Query-Answering performance
Test objective	Test the performance of query execution within the SIS
Testing steps	Measure response times to queries against various collections and numbers of advertisements.
CURRENT STATUS	Performance is satisfactory for a registry with advertisements of mild complexity
NEXT STEPS	Improve performance though a distributed SIS design

4.2 MIS

Name	Retrieve market information
Test objective	Obtain average, minimum maximum price for a certain time period and product
Testing steps	Publishing of market data from the Auction Service Afterwards querying of the data as input for new bids
CURRENT STATUS	Isolated MIS tests and simulations using the interface provided to the CAS
NEXT STEPS	Integration and testing with GRIMP/CAS

4.3 Scheduling Service

Name	Deployment of XtremWeb workers
Test objective	Correctness and performance Jade based deployment
Testing steps	Deploy XtremWeb worker as a legacy application using Jade. This is used as part of gMovie demo.
CURRENT STATUS	Deployment and configuration of workers were done correctly (they joined Master and completed their assigned jobs). Different numbers (until 10) workers were deployed simultaneously on different nodes and no scalability problem was identified.
NEXT STEPS	No next steps

Name	Correctness of MinMin and XtremWeb forecast heuristics
Test objective	Compare the performance and quality of the two heuristics
Testing steps	Four compute nodes and task size varying between 100 to 500
CURRENT STATUS	XtremWeb estimation of completion time is comparable to that of MinMin. The time taken by XtremWeb forecast is linear whereas that taken by MinMin is exponential. This shows that that XtremWeb forecast method could be used to quickly compute estimates of job completion times. This heuristic may be used to refine allocation requests (quantities, durations and time ranges).
NEXT STEPS	No next steps

4.4 CAS – K-DA

Name	**Mechanism evaluation**
Test objective	Evaluation of k-pricing multi lane double auction
Testing steps	Evaluation through real simulation: Data generation, mechanism evaluation, and final results analysis.
Comparative	Comparative against K-DA from JASA framework. Performance, Efficiency, memory usage, scalability.
CURRENT STATUS	Evaluation and Comparative already done.
NEXT STEPS	N/A

4.5 CAS – CA

Name	**Comparison of heuristics developed to solve the Winner Determination Problem for Grid4ALL CA model**
Test objective	Compare the allocative efficiency (surplus) and time to compute WDP with that of an exact resolution using CPLEX solver.
Testing steps	Same set of input instances (bids representing jobs from buyers and bids representing bundle offer from sellers) are resolved using the following methods: ● CPLEX solver ● Greedy methods ● Gradient descent method
Comparative	Comparison with other combinatorial auction mechanisms described in literature have not been done. The GreedyX heuristic proposed by the SORMA project presents a 70% average efficiency
CURRENT STATUS	When instance sizes (bids) exceed 100, CPLEX either stops (out of memory) or takes many hours to achieve reasonable gaps from best estimated solution. The different greedy methods achieve different levels of efficiency. The best achieved allocative efficiency (closeness to optimal solution) reached 85% of optimality. The gradient descent method has shown to reach 82% of optimality over the compared input instances. It also improves the quality of solution by increasing the number of allocated jobs.
NEXT STEPS	Provide tools to decide which algorithm should be used in which situation. Compare the CA mechanism and the proposed heuristics to similar mechanisms proposed in literature.

1. Conclusion

This document has presented results of two categories of evaluation: qualitative evaluation of integrated and demonstrated results; and quantitative evaluation of the most important functional modules in the system. Qualitative evaluation is performed by persons external to the consortium. According to the type of result evaluated, this process consists of either developing an application using the result or using the result as an end user.

Results show a good level of integration as well as an acceptable unitary behaviour (performances, correctness). Improvement proposals have been noted. The final qualitative and quantitative results will be reported in the D5.3.