Final Project Report

GridCOMP: an Advanced Component platform for an Effective Invisible Grid

GridCOMP
Effective Components for the Grids

Featuring GCM ETSI Standards,
available in Open Source within

http://gridcomp.ercim.org

Designed and printed by:
azur communication
2323 Chemin de St Bernard - Space 6 - BP 255
Parc de Sophia Antipolis - 06227 Vallauris cedex (France)

Photos: IS/S. Prokhrov, S. Crinks, D. Turner, C. Schmidt, A. Benedek, M. Boncina, FOT/C. Georghiou
October 2008

GridCOMP is a Specific Targeted REsearch Project (STREP)
Co-funded by the European Commission

Grid Programming with COMPonents: an Advanced Component platform for an Effective Invisible Grid

GridCOMP is a Specific Targeted Research Project on Grid programming with components, partially funded by the European Commission in the sixth Framework Programme from June 2006 to February 2009.

The main objective of GridCOMP is the design and implementation of a component based framework suitable to support the development of efficient grid applications: **the Grid Component Model (GCM)**. The framework implements a kind of "invisible grid" concept as it properly abstracts all those specific grid related implementation details that usually require high programming efforts to handle.

This brochure presents the partners of the GridCOMP Consortium as well as its activities and achievements over the thirty three months of its EC-funded life time.

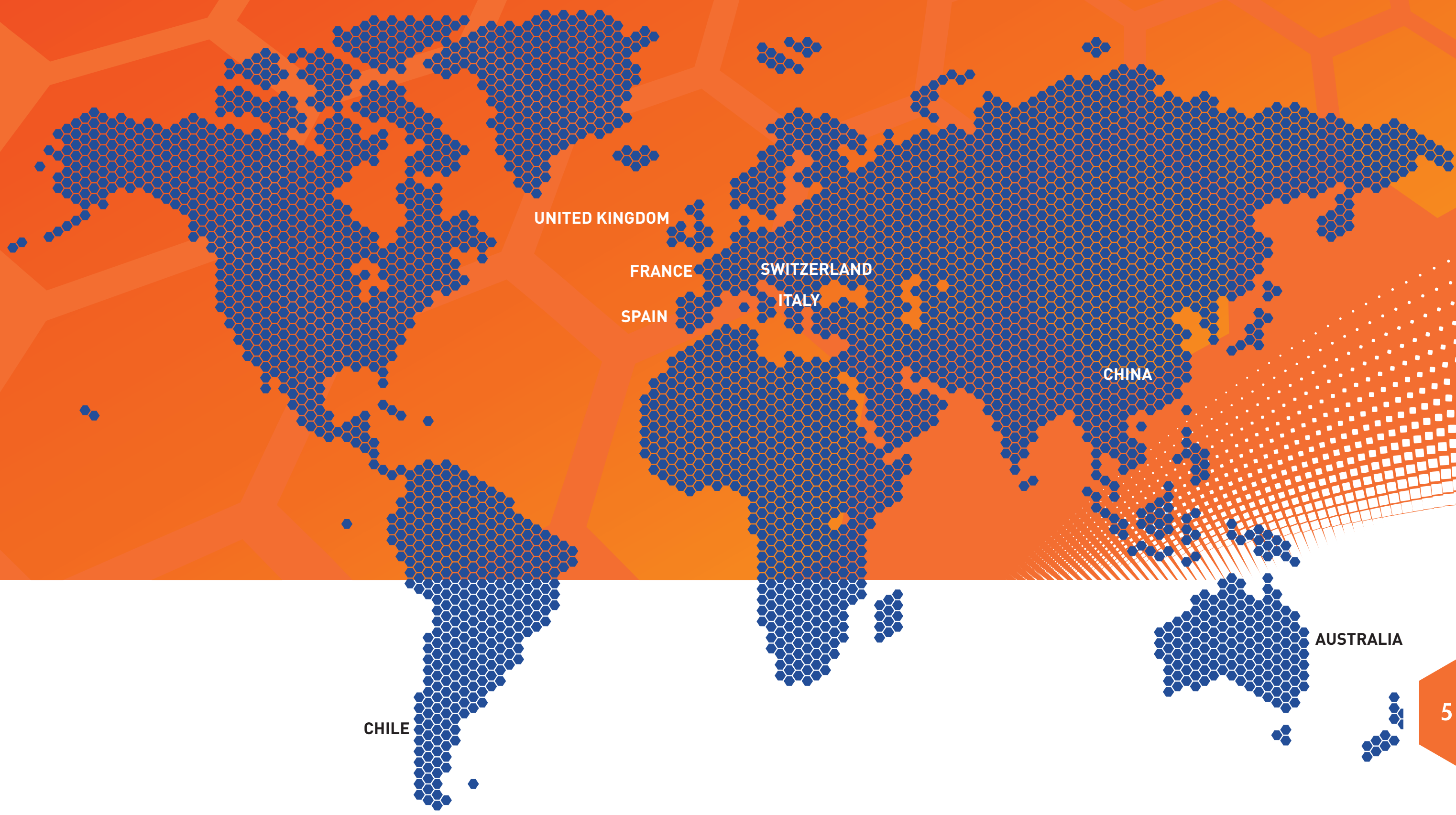
Project N°: FP6-034442
Duration: 33 months
Start Date: 1 June 2006
End Date: 28 February 2009
Overall cost: € 2 935 481
European Commission Funding: € 1 750 000

GridCOMP home page:
http://gridcomp.ercim.org/

Funding Agency:
European Commission FP6
Unit D3 : Software and Service Architectures and Infrastructures

Contents

GridCOMP Consortium



GEIE ERCIM	France
INRIA	France
University of Pisa – Dept. Computer Science	Italy
University of Westminster – Harrow School of Computer Science	UK
ISTI-CNR Pisa	Italy
GridSystems	Spain
IBM Research Zurich	Switzerland
ATOS Origin	Spain
Tsinghua University	China
University of Melbourne	Australia
University of Chile	Chile

Executive Summary

Vision

By implementing the Grid Component Model (GCM), which was originally defined in the CoreGRID Network of Excellence, **GridCOMP aims at providing a Grid programming framework ensuring efficiency as well as interoperability**. The GCM interoperability is being standardised within the ETSI Technical Committee on Grid Computing. The full integration of GCM components into Service Oriented Architecture (SOA), as well as Service Level Agreements (SLA) and Quality of Services (QoS) enforcement is the next goal.

In the domain of distributed software and middleware there are two different views nowadays:

- Architecture-based view which mainly relies on component model.
- Service-oriented view.

In the architecture-based view, software is described at design time with an Architecture Description Language (ADL). Some models also allow changes in the initial architecture by featuring reconfiguration at runtime. The Corba Component Model (CCM) and Service Component Architecture (SCA), as well as the Fractal and GCM models, support an architecture-based organisation. In the SOA view, software is seen as a set of independent services; location of these services may not be known at design time, they are only described. These services, accessible through Web Services, are lightly coupled in opposition to the architecture-based view where components could be strongly coupled. However GCM does not impose tightly coupled component assembly.

These two points of view each have their advantages but also similarities, such as the notion of software block (component or service). Thus, we believe there is a need for a programming framework, bridging the gap between component technologies and SOA in order to achieve code and service composition for flexibility and agility. The SCA initiative has already started to address this issue at design and deployment time; but the purpose is to integrate component ideas into SOA in order to feature the power of fully-fledged components with, for instance, the capacity to impose a reconfiguration or an interoperable and flexible deployment of components upon execution when needed.
Academia and industry are still lacking such mechanisms allowing dynamic reaction on events occuring in real-time, for instance machine failure or overload. Moreover, we still miss the mechanisms which take into account application level requirements, such as SLA, in order to improve contracted QoS. Another issue, tackled in the GCM, is the challenge to identify the basic

requirements for a given component in order to select dynamically the right set of host machines. This is a strong limitation that prevents the use of components in SOA. Such functionality will require a unified model, valid for most of the Grid application, describing application requirements. This leads to another issue that has to be treated at the same time: **standardisation**. There is a need to have a standardised component model, application description, and application level requirements in order to really promote a block of software at the level of reusable components.

Overall, we believe we have to achieve in a standardised way the full integration of components into SOA, with the capacity to monitor and dynamically maintain the QoS at execution.

Achievements

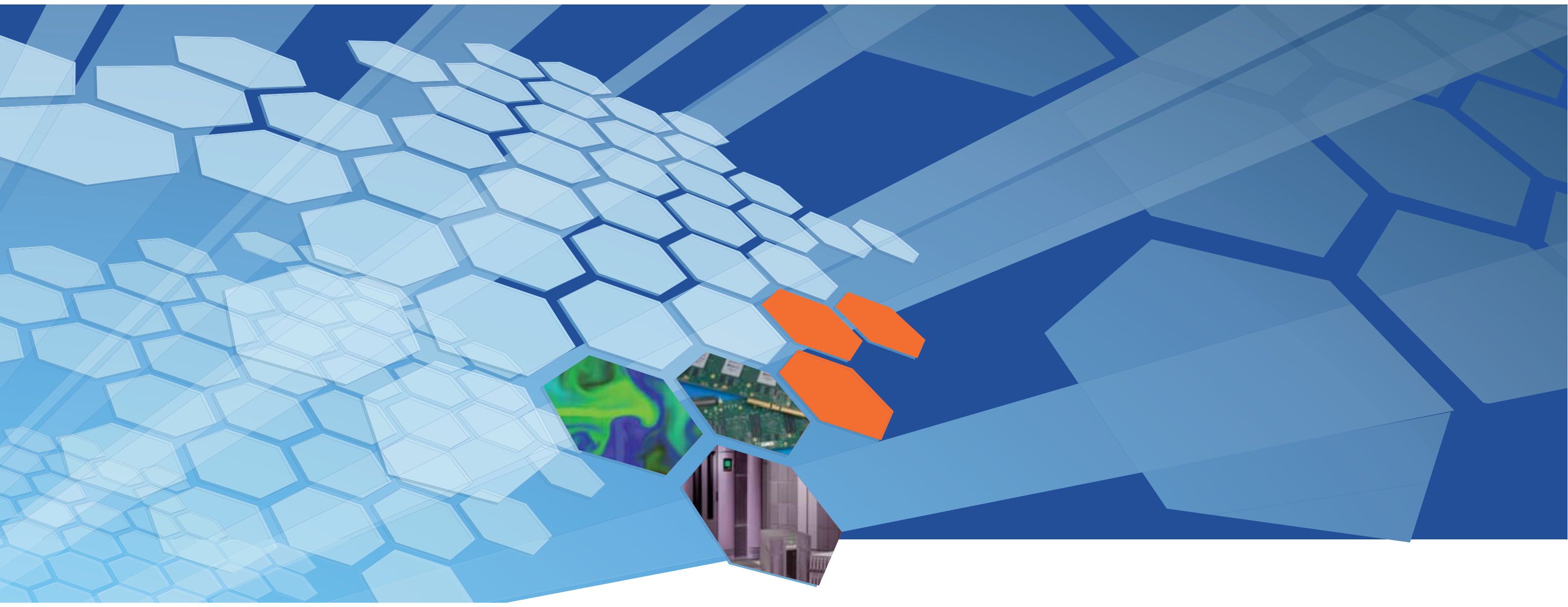
GridCOMP provides the reference implementation of the GCM. This prototype features a component framework allowing creation of remote components and their remote access in a transparent manner supporting collective communications. This prototype also includes a deployment framework providing interoperability with several grid schedulers and middlewares.

The **Non Functional Component Features** provide a prototype of *behavioural skeletons* modelling *common parallelism exploitation patterns* and implementing an *autonomic manager* which takes care of ensuring *user-supplied performance contracts (SLAs)*. The behavioural skeletons themselves are provided as composite GCM components. Users can instantiate such composites by providing other components to specify the functional part of the code. Behavioural skeletons will enable users to develop and deploy efficient grid parallel applications in a sensibly shorter time with respect to the time required to develop and deploy applications with similar efficiency built from scratch by the application programmers. Actually, GCM composite components implementing behavioural skeletons automatically deploy on target architecture while optimising usage of the existing resources. The initial set of behavioural skeletons developed within GridCOMP, implement parameter sweeping, master/worker as well as several types of common data parallel patterns.

A new component-oriented development methodology has been created. This methodology enables users to build large-scale Grid systems by integrating independent and possibly distributed software components, via well-defined interfaces, into higher level composite components. The main benefit from such an approach is improved productivity. As an implementation of this methodology, the **Grid integrated development environment (GIDE)** has been built. It is tightly integrated with Eclipse software framework and was designed to empower the end user with all the tools necessary to *compose, deploy, monitor, and steer Grid applications*.

Four use cases have been developed and promote the capabilities of the GridCOMP framework. They represent test cases as well as *demo applications with respect to the GridCOMP component framework*. They also point out the advanced features provided by the framework and, through their complete documentation, illustrate how *they can be exploited in real world scenarios*. The use cases represent a jump start for people new to GCM and are thus vital for the success of the framework. Industrial partners make use of their respective use cases to highlight the benefits of GridCOMP, both internally and to their customers.

GridCOMP also disseminated the GCM through many channels. Training materials introducing applications and frameworks produced within GridCOMP have been published online. Additionally, GridCOMP held several successful **workshops and tutorials in Europe and China** in order to get users familiar with the GCM platform and to ensure the impact of the results to a worldwide audience *via* our international partners.
In the meantime, **standardisation of the GCM** definition in four different standards made headway. Two items have been officially accepted as ETSI standards "GCM Interoperability Deployment" and "GCM Interoperability Application Description". The two remaining parts, namely, "GCM Fractal ADL" and "GCM Management API" are in the process of being standardised.



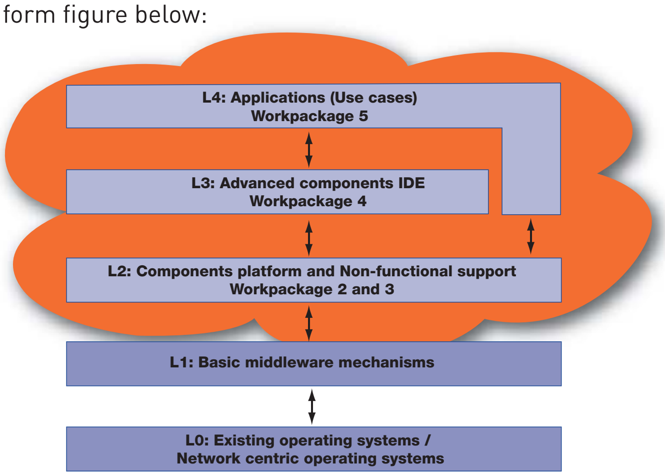
GridCOMP Structure

GridCOMP aims at designing and implementing a component based framework suitable to support the development of efficient grid applications, in order to reduce the complexity of Grid-based systems, thus empowering individuals and organisations to create, provide access to, and use a variety of services, anywhere, anytime, in a transparent and cost-effective way. The framework prototype is built according to a layered implementation approach.

To supervise the implementation of the GridCOMP full range of activities, a specific management architecture has been designed and the project is structured into seven work packages organising a wide array of tasks and representing all participants:

1. **Management**
2. **Component Framework Implementation**
3. **Non Functional Component Feature**
4. **Grid Interactive Development Environment**
5. **Use Cases**
6. **Dissemination**
7. **Collaboration**

In addition, the technical activities of the consortium carried out by the well-known research teams are aimed to implement by the layered platform figure below:



GridCOMP Structure

1. Management

The Project management is designed to ensure a coherent scientific multi-disciplinary, administrative and financial coordination of GridCOMP, while providing the participants with the support and tools required for the achievement of the project objectives.

The Coordination team members:

- Establish a democratic yet reliable overall organisation supporting the completion of the Implementation Plan activities;
- Support the integration of both research teams and research activities, and ensure in particular the interaction among the different GridCOMP work packages and activities;
- Assess the quality of work achieved and take appropriate measures as necessary, Supervise and review, in particular, the completion of the milestones and deliverables;
- Promote the STREP visibility: international dissemination, industrial liaison...;
- Handle all the administrative and financial tasks connected with the activities of the consortium;
- Ensure communication with the European Commission.

Scientific coordination

Denis Caromel – INRIA-CNRS-UNSA

Denis.Caromel@inria.fr

Administrative and Financial coordination

Patricia Ho-Hune – ERCIM

Patricia.Ho-Hune@ercim.org

The main decision-making body of the consortium is represented by the Steering Committee, which is composed of:

Denis Caromel, INRIA

Patricia Ho-Hune, ERCIM

Marco Danelutto, UNIPI

Vladimir Getov, UoW

Gaston Freire, GridSystems

Domenico Laforenza, ISTI-CNR

GridCOMP Structure

2. Component Framework implementation

Objectives

The main objective is to provide the reference implementation of the Grid Component Model (GCM) defined by the CoreGRID NoE and further developed in the GridCOMP project. The GCM is an extension of the Fractal component model for the Grid. The developed prototype takes the ProActive Parallel Suite as the starting point to provide the functional features of GCM components such as a deployment framework, primitive and composite components. GCM components turn standard code, potentially parallel and distributed, or legacy code, into components able to be deployed and composed hierarchically. This implementation is used in work package 3 to implement non-functional GCM features and is illustrated in the use cases.

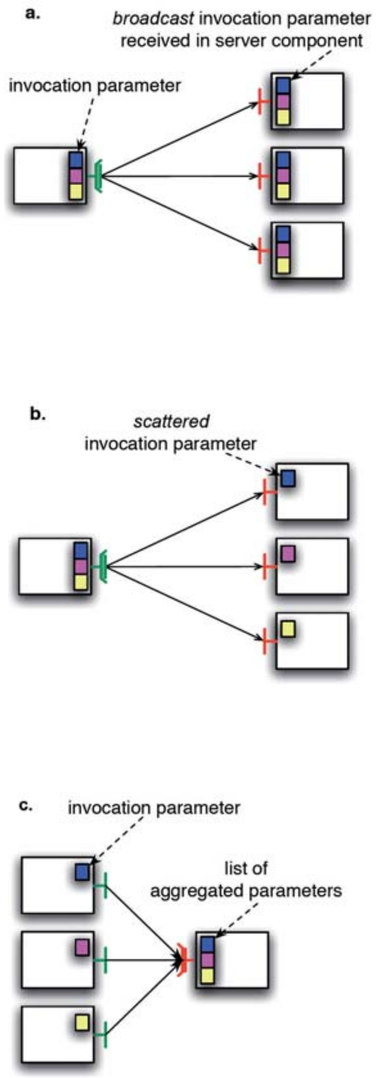
Leader: Denis Caromel
Name of the institute: INRIA
Email: Denis.Caromel@inria.fr

Main activities and achievements

Two major achievements have been realised: a deployment framework and a component framework.
The GCM deployment framework is the result of several years of tests and developments. Those activities in collaboration with ETSI lead to two standards "GCM Interoperability Deployment" and "GCM Interoperability Application Description". These standards define, respectively, XML schemas describing a Grid infrastructure and a Grid application. Implementation of these standards ensures interoperability with several grid schedulers and middleware such as Platform LSF, PBS, OAR, SSH, SGE, Globus, EGEE gLite, etc. Both standard and GCM component applications can leverage GCM deployment. Graphical tools allow users to monitor a deployed application and manage resources acquired with the GCM deployment.

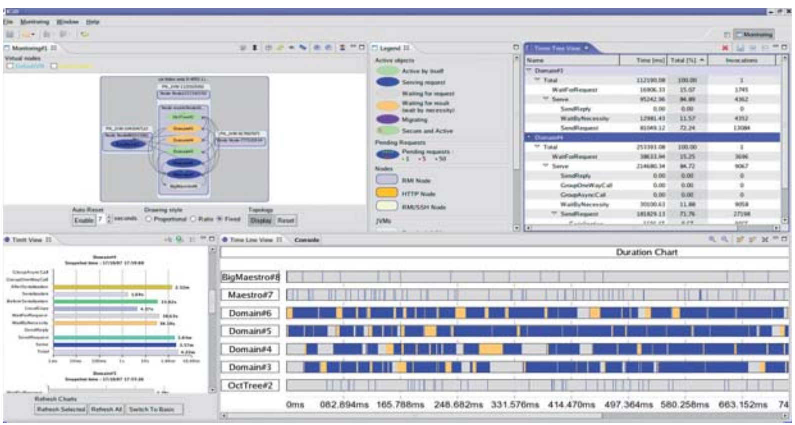
GridCOMP provides a Java-based implementation of the GCM functional features. This prototype features a component framework allowing creation of local and remote components. These components are able to communicate remotely in a transparent manner supporting collective communications. In addition to singleton interfaces, collective interfaces express the parallel behaviour of a Grid application at design time. GCM defines two kinds of collective interfaces, namely, multicast and gathercast. A multicast interface provides abstractions for one-to-many communications (Figures a and b); it transforms a single invocation into a list of invocations. A gathercast interface provides abstractions for many-to-one communications (Figure c); it transforms a set of invocations into a single invocation. These interfaces are customisable and typed. A component assembly is defined statically using an Architecture Description Language (ADL) and/or dynamically, using API.

The GCM reference implementation targets all software architects in need of a comprehensive framework to express at design time the parallelisms and the distribution of an application. Therefore, the architecture of the system itself captures the parallel/distributed aspects, acting as a powerful specification and documentation. Further, developers will not need to spend extensive time to learn distributed or multi-core programming or implement collective communication, but rather concentrate on the business code and leverage the GCM framework. Finally, applications will be seamlessly deployable on various Grid infrastructure or multi-core using the GCM deployment framework.



Future orientation and foreseen research challenge

The GCM deployment standard already provides interoperability at deployment time with the support of various middleware and schedulers. In addition, services offered by a GCM component can be accessed through Web Service (WS). Next steps will be support of WS bindings allowing a given GCM component to access another GCM component or application using WS. Such features will improve the interoperability at the component communication level with other middleware. A more general objective is to provide an SCA implementation with dynamicity at runtime thanks to the GCM features. Therefore, GCM components are the building blocks for integrated SOA towards SLA and QoS.



Graphical monitoring and visualisation of ProActive/GCM application with IC2D

3. Non Functional Component Feature

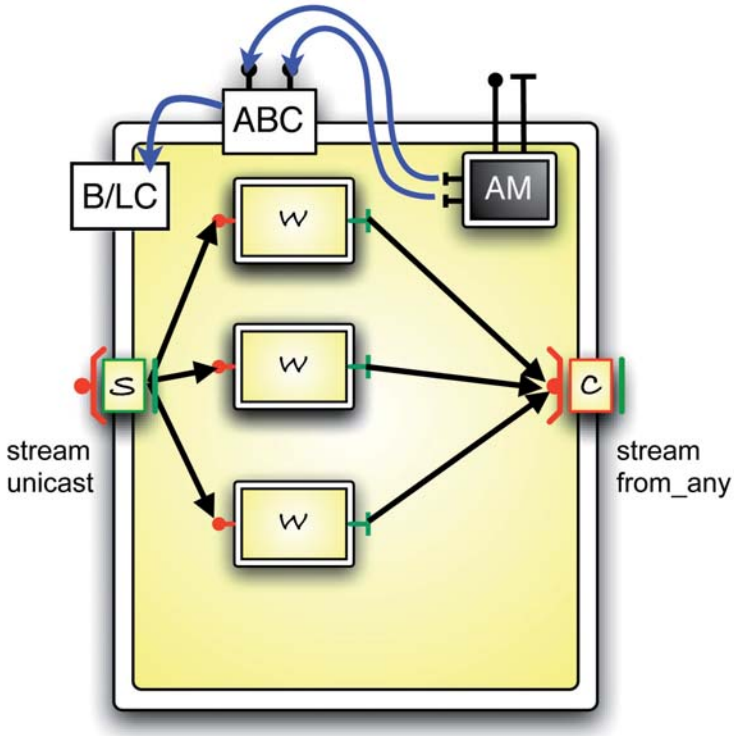
Objectives

The main objective is the design and implementation of non-functional features of GCM components. That is to say, the goal is to enable the handling of non-functional features of GCM components such as performance and fault tolerance by means of autonomic managers properly programmed within GCM components. The availability of such autonomic managers will raise the level of abstraction presented to GCM programmers by moving most (or possibly all) of the non-functional concerns from application programmers to the system programmers implementing the autonomic managers in GCM components. The implementation of autonomic managers in components should exploit the component framework as designed and implemented in work package 2. The Non Functional Component Features work package aims at providing autonomic managers, taking care of performance-related concerns in composite GCM components, which model well-known parallel patterns. These autonomic managers also support the use cases considered within GridCOMP.

Leader: Marco Danelutto
Name of the institute: Dept. Computer Science, Univ. of Pisa
Email: Marco.Danelutto@di.unipi.it

Main activities and achievements

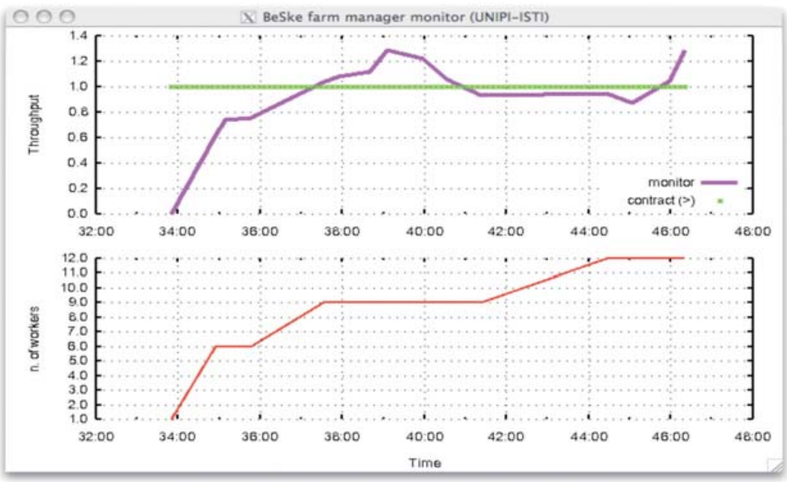
Partners involved in the work package implemented several distinct behavioural skeletons. A behavioural skeleton is a composite GCM component modelling a well-known parallelism exploitation pattern (such as task farm or data parallel patterns) that also provides the user with an autonomic manager taking care of ensuring a user supplied performance contract. GCM programmers can use the skeleton by simply providing the worker component (i.e., the component computing a single task in the task farm or a partition of the final result in a data parallel pattern) and the performance contract. Performance contracts include requiring throughput not smaller than a



given constant and/or not higher than another user-supplied constant and can be roughly intended as first order logic formulas over execution parameters. Autonomic managers within the GCM composites implement a classical control cycle: the current composite component behaviour is monitored and in case a violation of the user contract is detected, proper corrective actions are planned and executed. The control cycle executed within the autonomic manager allows component execution to adapt to varying features of both the Grid target architecture and the application requirements themselves. Experiments on Grid5000 with both synthetic applications and real use cases demonstrated both task farm and data parallel behavioural skeleton (and consequently composite autonomic manager) functionality and efficiency. Partners of the use case work package were able to use these behavioural skeletons while programming the use case prototypes.

Future orientation and foreseen research challenge

The behavioural skeleton approach adopted within the work package can be clearly extended in such a way that i) multiple non-functional features are concurrently managed, and ii) hierarchical composition of autonomic managers is exploited to achieve autonomic management of large Grid applications programmed using a nesting/composition of several distinct behavioural skeletons. While the latter point has already been partially investigated within the Non Functional Component Features work package and preliminary results have already been achieved, the former topic still needs consistent research activity. The availability of a GCM programming framework supporting both features will provide users with advanced environments where all the relevant, non functional features are handled within the environment itself, much in the perspective of a cloud architecture for component- based applications.



GridCOMP Structure

4. Grid Interactive Development Environment

Objectives

The main objective is to develop an Integrated Development Environment for Grid (Grid IDE – GIDE), supporting two different streams of operations: development and data centre management. The following are the specific objectives. The primary objective is to design and develop an integrated tool to support:

- Graphical composition of component based grid applications
- Deployment of the components
- Monitoring of the deployed components
- Steering of the deployed components

Secondary objectives include:

- Establishing and expanding collaborative work with industrial partners on testing and experimenting with their use cases
- Completing implementation of the composition development cycle for these use cases
- Platform/OS-independent implementation of resource monitoring
- Producing a standalone steering application for Data Centre operators

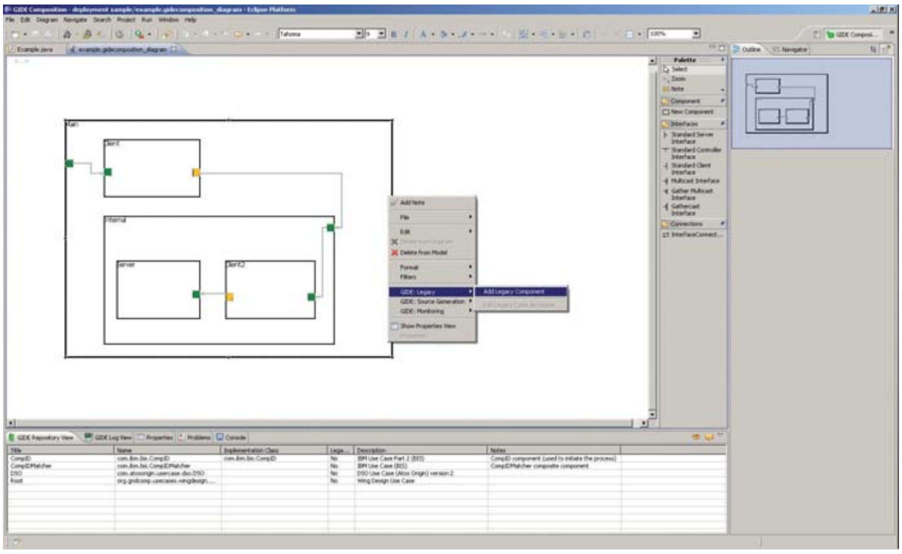
The Grid Integrated Development Environment (GIDE) provides extensive user support for composition, deployment, monitoring, and steering of Grid applications in a single framework based on the Grid component model (GCM). The main advantages of this methodology are: reduced software development cycle, increased portability, and support for dynamic properties in the generic component-based Grid system built on top of the ProActive Grid middleware.

UNIVERSITY OF WESTMINSTER

Leader: Vladimir Getov
Name of the institute: University of Westminster
Email: V.S.Getov@westminster.ac.uk

Main activities and achievements

After the delivery of the first partial prototype of Grid IDE at the beginning of July 2007, the internal GIDE mechanisms were re-engineered in August so that it relies on the Graphical Modelling Framework (GMF) as recently introduced in Eclipse. Thus, instead of directly relying on the Graphical Editing Framework (GEF), the version released at the beginning of September 2007 stemmed from EMF/GMF. Ver-



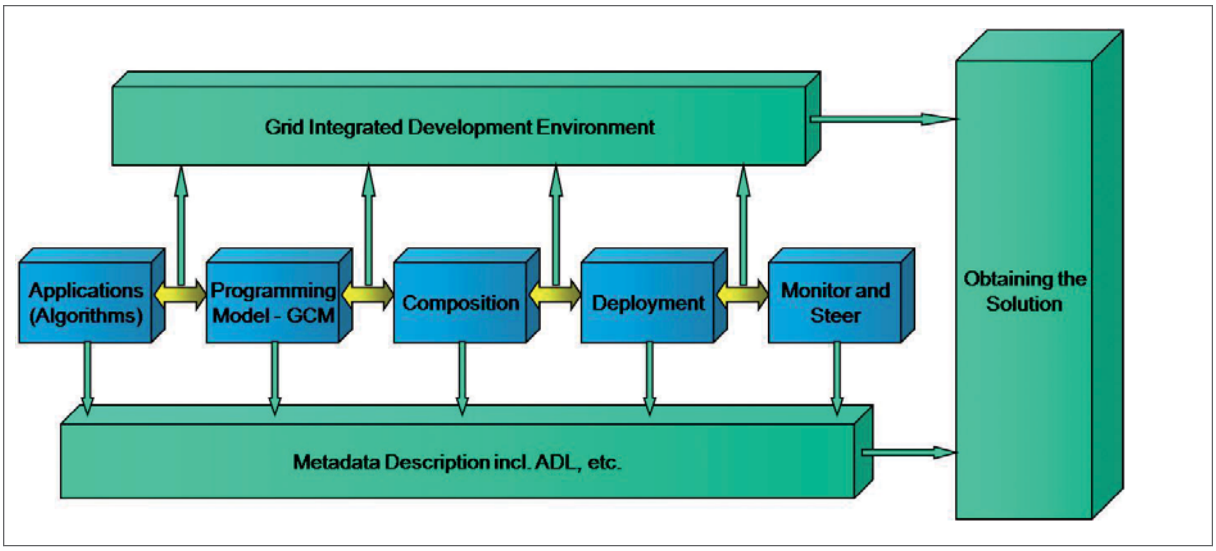
sion 2.1.0 was released in September. In addition, collaborative work with IBM on the BIS use case was also advanced by experimenting and demonstrating the GIDE support for component composition and development. The benefits of using such an approach are:

- shorter development cycle,
- higher portability, and
- support for dynamic properties.

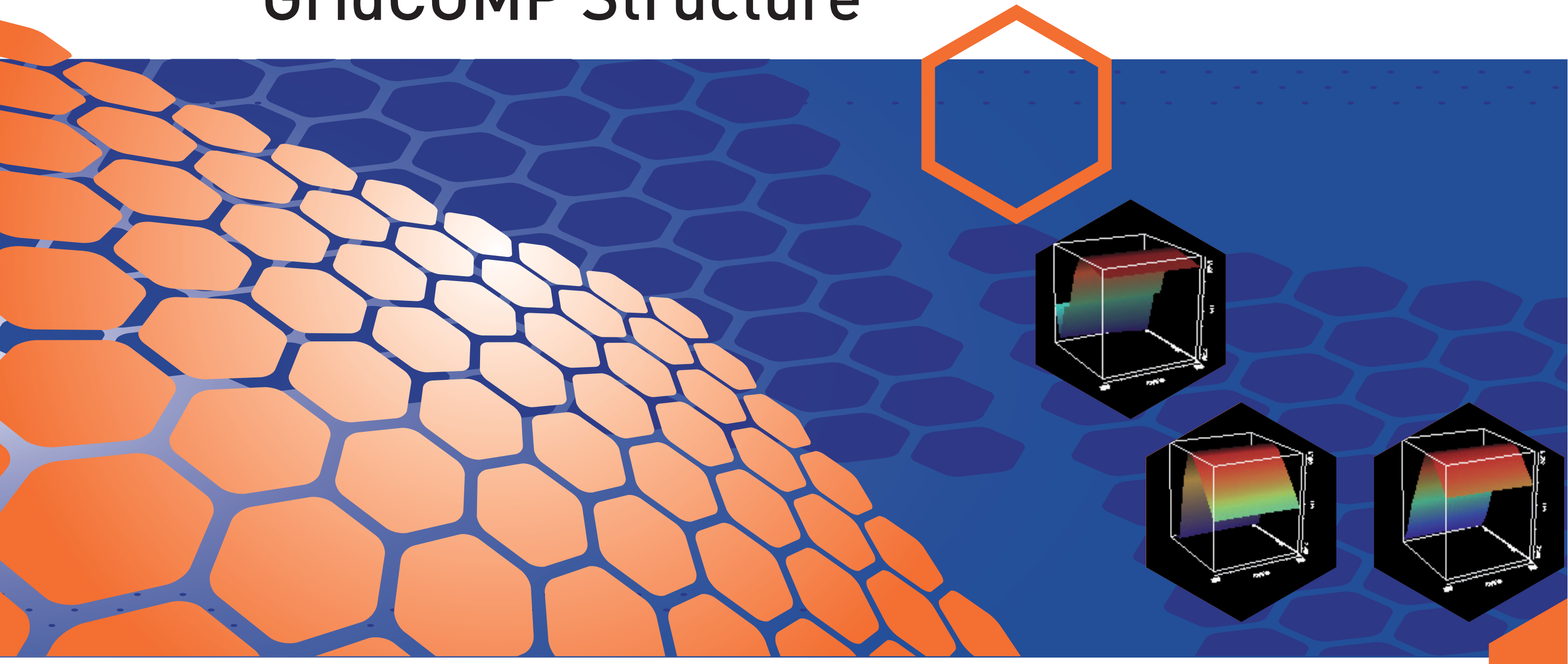
These benefits have been realised through the use of the development tools provided within the GIDE for graphical and hierarchical component based development, as well as support for deployment, monitoring and steering.

Future orientation and foreseen research challenge

We will continue extending the functionality of the GIDE, in particular in relation to the component monitoring and steering functionalities. In addition, future work will focus on applying the development methodology to a selection of applications, with a view to gathering development statistics in order to accurately gauge the impact of the GIDE development methodology on the development lifecycle.



GridCOMP Structure



5. Use Cases
GridSystems Use Case

EDR Processing (Extended Data Record)

Record processing is a common computing problem that enterprises have to deal with. Telco companies for instance, take the information from the calls their customers make, process the data by following some specific rules, and then draw useful conclusions and thus obtain valuable business knowledge. The transformation usually takes a lot of time and requires a considerable amount of computing resources.

Objectives

The objective in this use case is to provide a high performance computing solution, based on a Grid Component version EDR program, thus improving the quality of the processing by offering scalability, redundancy, load balancing, and reduced computing times.

Leader: Gaston Freire Amoedo
Name of the company: GridSystems S.A.
Email: gfreire@gridsystems.com

Before & After GridCOMP

Single platform, expensive specific hardware
Multi platform, low-cost hardware

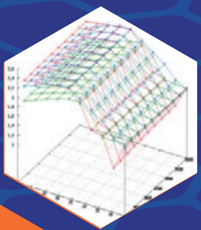
Changes in deployment need code changes
XML-based configurable deployment

Ad hoc design
Component based top-down design, applying composition

High management costs
Automatic management, providing redundancy and load balancing

Expensive and difficult to scale
Cheap and easy to scale

Commercial, closed source
Free, open source framework and libraries based



The GridCOMP Solution

The GridCOMP Solution offers to the EDR application:

- A grid-computing component-based model that is fully portable because it is solely based on Java
- The Composition of components and the ability to follow a top-down design of the application and an easy way to reuse code
- Collective interfaces that hide the complexity of Grid computing
- An autonomic management component that provides load balancing

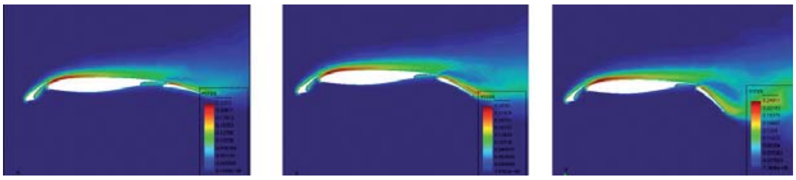
Benchmarks

Grid 5000 offers the following results:

- Performance scales close to linearly when increasing the number of nodes
- Given the data-intensive nature of the application, its performance is dependant on the load of the network and the size of the file fragments distributed and processed. This is the reason for the observed variability in some of the results.

Wing design application

In the aerospace sector, the software that computes the aerodynamic wing performance for a given configuration is used to test different configurations of the wing features and to eventually find an acceptable design. Merak is an application that permits analysis of parameter variation in incompressible turbulent flow around a triple element airfoil, to evaluate the stalling angle of different wing geometries. Turbulence is simulated using different k-eps models and finally the application can extract the desired information and create the graph.



Objective

Merak manages small amounts of information, but needs lots of computing power. Our objective is to use the GridCOMP solution to wrap and grid-enable this existing legacy code and also to prove the integration of data staging for the input files and output files into this sweeping. Proving both of these objectives is crucial for the adoption of the GCM by the industry.

The GridCOMP Solution

GridCOMP offers to this application:

- Legacy code wrapping support, enabling its re-usability in modern, component-based, grid applications.
- Composition of components to combine new and pre-existing components
- Collective interfaces that abstract and hide the complexity of distributed computing
- Autonomic component management that provides load balancing
- Tools to design the architecture of the grid application (GIDE)

Before & After GridCOMP

Legacy application, obsolete code, hardly reusable
Legacy code wrapped component, ready to be reused in new applications

Long response time, due to serial execution
Distributed execution renders reduced response time

Manual management of failed executions
Automatic management, providing redundancy and load balancing

Buying faster hardware as the only way to scale up (due to serial execution)
Scaling up means adding more low cost hardware

Benchmarks

Scalability experiments using Grid 5000 shows that performance scales linearly when increasing the number of nodes.

GridCOMP Structure

5. Use Cases
IBM Use Case

Biometric Identification

Presentation

In recent years biometric methods for verification and identification of people have become very popular. Applications span from governmental projects like border control or criminal identification to civil purposes such as e-commerce, network access, or transport. Frequently, biometric verification is used to authenticate people meaning that a 1:1 match operation of a claimed identity to the one stored in a reference system is carried out. In an identification system, however, the complexity is much higher. Here, a person's identity is to be determined solely on biometric information, which requires matching the live scan of biometrics against all enrolled (known) identities. Such a 1:N match operation can be quite time-consuming making it unsuitable for real-time applications.

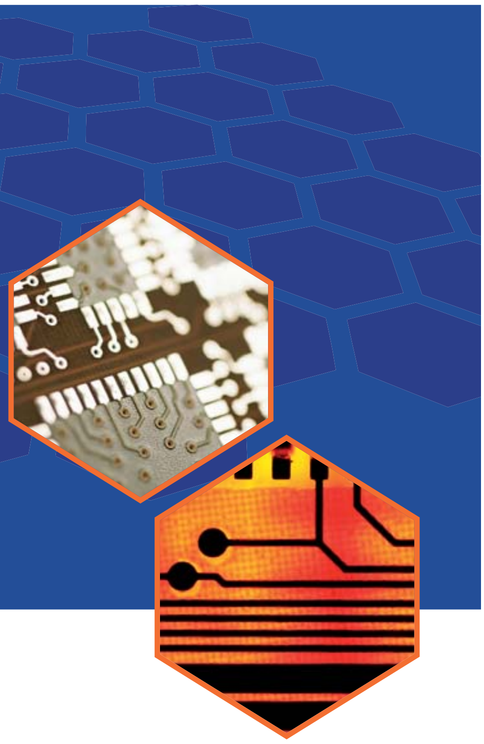
Leader: Thomas Weigold
Name of the company: IBM Zurich Research Laboratory
Email: twe@zurich.ibm.com

Objective

The objective of the use case is to build a biometric identification system (BIS), based on fingerprint biometrics, which can work on a large user population of up to millions of individuals. To achieve real-time identification within a few seconds, the BIS application takes advantage of the Grid via GCM components. The goal is to simplify the programming of the distributed identification process though the use of the GCM framework while being able to deploy the resulting application on arbitrary existing, potentially heterogeneous, hardware platforms.

Architectural design

The BIS use case can be considered a business-process or workflow-driven application. Figure 1 outlines the high-level architectural design of the BIS. It is built around a workflow execution engine acting as the central control unit of the system. A number of business processes are implemented as workflow scripts running within the engine. The processes comprise functionality accessible from the demo application (e.g.

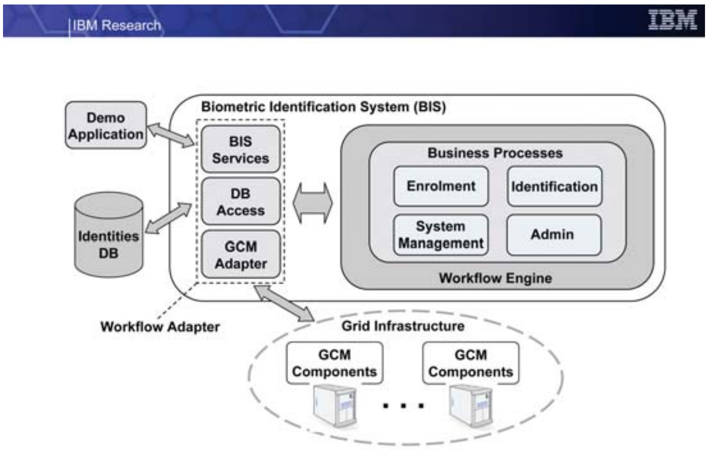


identification) as well as internal system management logic required to control the distributed biometric matching. Furthermore, the BIS provides a number of adapters to the workflow engine such that the business processes can interact with external entities, namely, the database (DB) storing information about enrolled identities, and the interface to the Grid infrastructure.

The GCM adapter (Figure 1) is triggered by the workflow scripts and provides distributed biometric matching functionality via GCM components. Here, the basic approach is to have one component encapsulating the biometric matching functionality, which is then deployed on all Grid nodes in a SPMD-style setting. Then the database of enrolled identities is distributed across the nodes and this way the 1:N matching operation is executed in parallel.

The GridCOMP Solution

The GridCOMP Solution offers to the BIS application:



- A platform independent, high-level component framework for programming distributed applications.
- Advanced built-in features such as hierarchical composition, collective interfaces, virtual nodes, deployment descriptors, and support for autonomic management hiding the complexity of Grid programming.
- The GIDE, a comprehensive tool set supporting the complete software development cycle from graphical composition to component deployment and monitoring.

The GridCOMP computation results in:

- An identification system that can work on a large user population in real-time.
- An identification system that can be easily deployed to arbitrary existing hardware and thus is cost-efficient.
- A system that is easily scalable without any software change.
- An efficient software architecture (e.g. hierarchical components, strict separation of concerns) leading to reduced development time and component reuse.

Why BIS as GridCOMP use case?

There are several reasons why the BIS has been chosen as a use case for GridCOMP. Firstly, because it is considered a business process application which is centrally driven by a workflow engine. This makes it somewhat different to traditional Grid applications often coming from the scientific computing domain. The idea was to attract people coming from other application domains (e.g. BPM/workflow, security etc.) to look into the demo and find out what advanced Grid middleware could do for them. Secondly, it is a good test case to evaluate how the framework integrates with workflow systems. Thirdly, the BIS is somewhat different to the other use cases. Not only because it is workflow driven but also because it represents a data parallel problem rather than a task-parallel problem. Finally, the idea actually goes back to a previous customer request asking if such a system, which could work on arbitrary hardware infrastructures, could be built. This makes it a real world problem.

Before & After GridCOMP

Biometric identification is slow
An identification system working on a large user population in real-time

System modifications are required if the QoS contract changes
The system scales independently thanks to autonomic reconfiguration functionality

Specific dedicated hardware is required
Deployment on arbitrary hardware infrastructure without code changes

Grid application development is complex and time consuming
Reduced development time due to advanced features such as behavioural skeletons and the GIDE

GridCOMP Structure

5. Use Cases
Atos Origin Use Case

Computing of DSO value

Presentation

The use case selected by Atos Origin uses PL/SQL-based source code, and the candidate application selected was the so called "Computing of DSO value". The DSO (Days Sales Outstanding) is the mean time that clients delay to pay an invoice to Atos. This information is needed by several internal departments updated as much as possible and the process takes about 4 hours to compute around 6.600 clients.

Leader: Elies Prunes
Name of the company: ATOS Origin Sae
Email: Elies.Prunes@atosorigin.com

Leader: Fabio Tumiatti
Name of the company: ATOS Origin Sae
Email: Fabio.Tumiatti@atosorigin.com

Objective

The objective of using GridCOMP is to reduce the execution time without upgrading the infrastructure. Some of the benefits implemented using GridCOMP will be the possibility to update the information more frequently and maintain or reduce infrastructure cost.

Architectural design

The DSO application is based on a client/server application and there are three main elements:

- A Graphical User Interface (GUI) used to enter some input data or parameters needed for the computation. This GUI runs on the client side and connects to the Database;
- Some PL/SQL processes which are called normally by this user interface in order to access the data stored in the database and process them to compute the results. This part runs on the server side, and it is executed by the database engine;
- The Database which stores the data.

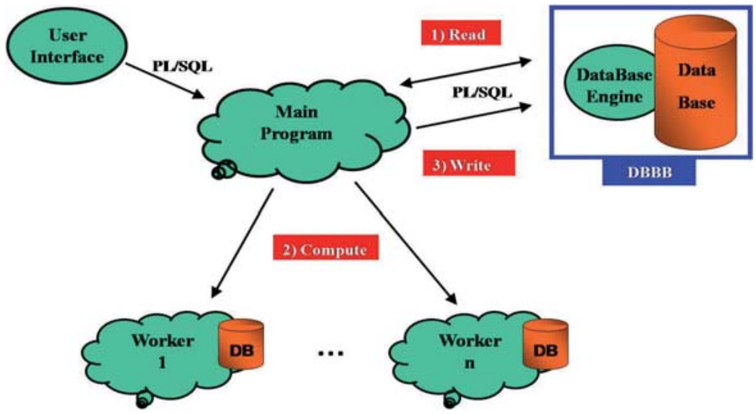
The architecture proposed for use with the Atos Origin use case is to put the main program (using ProActive) between the user interface and the database, as the orchestrator of the whole application. In this case, the user sends a request to the main pro-

gram, asking for the whole workflow to be executed. The main program will connect to the main database and read some data (tasks) to send to the remote nodes to be executed in parallel. The remote nodes will compute the information received and send the result. Then the main program will write the result in the main database finishing the process. The master database will store all data and only some part of the information will be sent to the remote node. Each node will contain a database engine to store part of the data sent by the scheduler and to start the PL/SQL code process.

The GridCOMP Solution

GridCOMP solution offers to the DSO application:

- A grid-computing component-based model that is fully portable because it is solely based on Java.
- Complexity of Grid programming hidden in features such as composition of components, deployment descriptors, virtual nodes, collective interfaces, autonomic management.
- An easy way to develop new applications using Grid components.



The GridCOMP computation results in:

- Reduced execution time without upgrading the infrastructure.
- More frequently updated information and maintained or reduced infrastructure cost.
- The ability to scale up easily by just adding more low-profile machines and without changing the application.

Before & After GridCOMP

Process takes about 4 hours to complete (over 6.000 clients)
Reduced execution time

Information not updated frequently
Information updated more frequently
Maintained/Reduced infrastructure costs

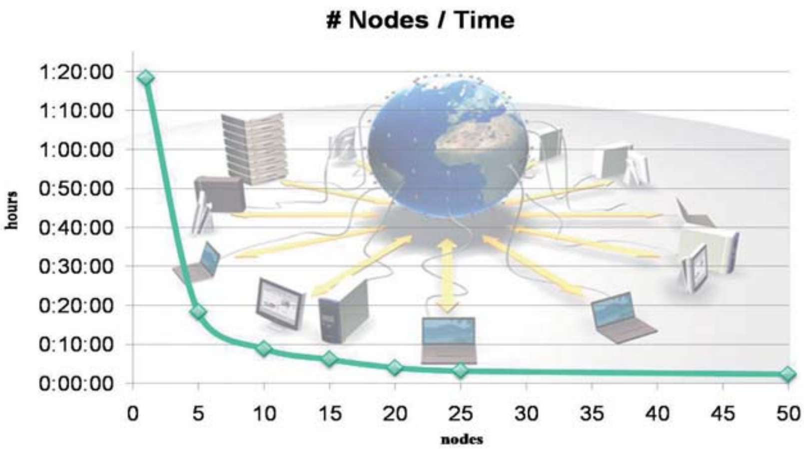
Benchmarks

Some tests were performed on top of Grid5000 platform to measure the benefits using a grid solution. As you can see in the chart below, the calculation time using 25 nodes is almost 96% less than using only one node.

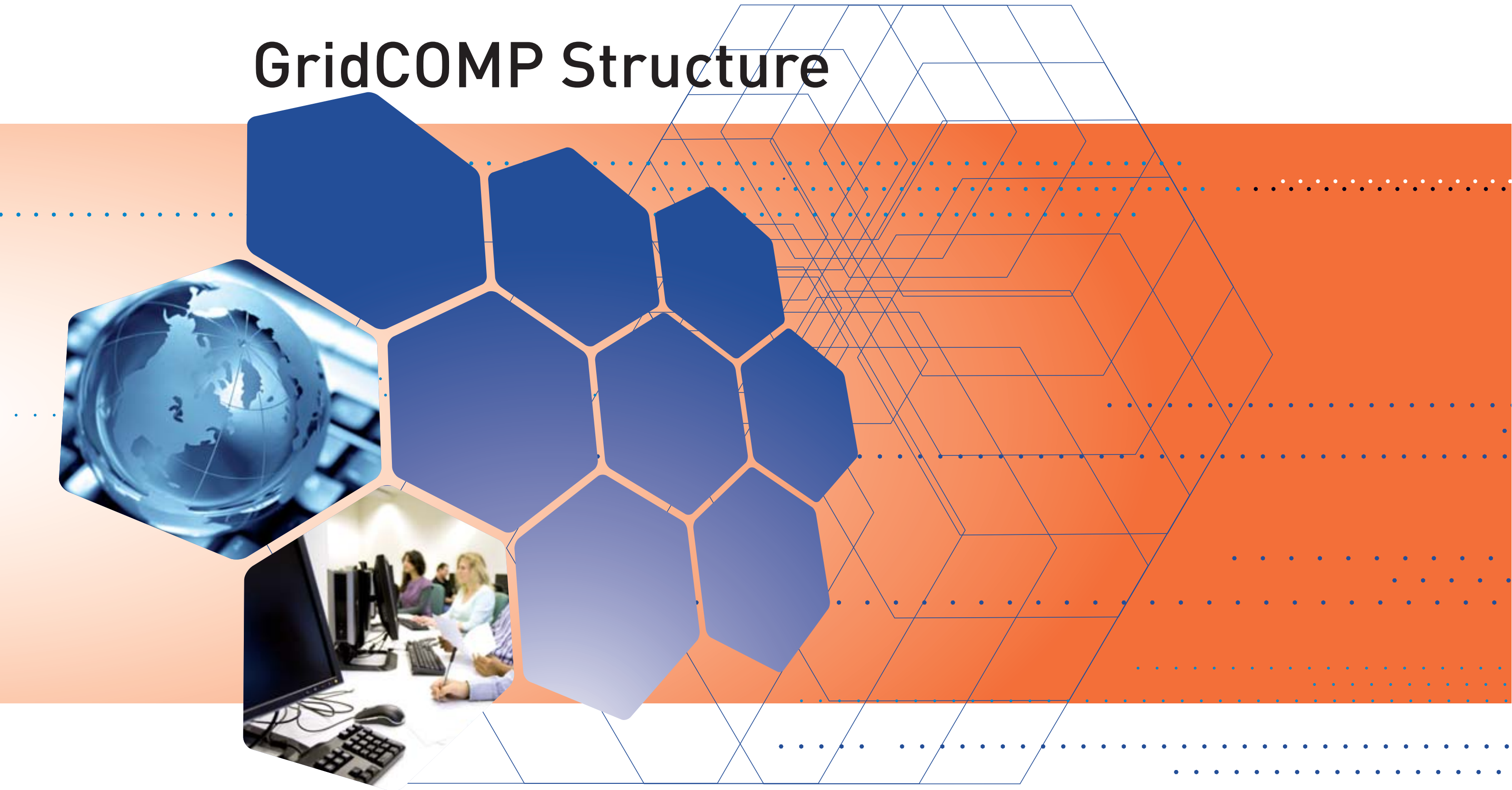
Selection strategy

Several criteria were defined in order to categorise the potential use cases and then select the most appropriate as a GridCOMP use case. These criteria cover not only the technical feasibility of the use case but also some commercial aspects, such as:

- Atos Origin interest in the results
- Technical support availability
- Source/object code constraints
- Hardware constraints
- Ability of the project middleware to support Application code wrapping
- Feasibility of the timeframe
- Complementary to the other GridCOMP use cases



GridCOMP Structure



6. Dissemination

Objectives

The objective of dissemination towards the overall Grid community is threefold:

- dissemination and promotion of the project and its results
- dissemination of knowledge and training material
- dissemination of GridCOMP standards

These three tasks not only ensure the international visibility of the project and its achievements, but also lay a reliable ground for the uptake and adoption of the GridCOMP standards worldwide. In that sense, training and standardisation activities are essential components of the project.

Main activities and achievements

To gain the highest visibility and raise public awareness and participation, GridCOMP implemented a large range of activities:

GridCOMP website http://gridcomp.ercim.org/ was launched in June 2006 and continuously improved. This website will be maintained after the end of the project and will include the update of the training materials, the advanced prototypes and source codes.

GridCOMP Flyer was designed and distributed at conferences and scientific events that partners attended. The partners acting as relays for the promotion of GridCOMP, the network has gained a large visibility.

GCM Training - Tutorials

Training activities were organised as soon as the component framework was designed in order to teach to interested industrial entities the possibilities provided by the GridCOMP environment. Such activity was also performed in a "virtual way", thus exploiting the Web to provide users with slides, tutorials, sample code, etc. In addition, tutorials were held in both Europe and China with a twofold goal: a) to demonstrate the results achieved, and b) to retrieve feedback from the user community concerning the choices performed and the implementation results achieved. Tutorials were also performed at the CoreGRID Summer school in July 2008 in Dortmund, Germany. These workshops and tutorials held in Europe and China supported the international dimension of the project and ensured efficient dissemination, targeting relevant communities and stakeholders through training activities aimed at promoting the various prototypes and products resulting from the project.

Dissemination events

- GridCOMP organised a first successful workshop in October 2007 in Beijing, China, during the Grids@Work week. 100 attendees from Europe and China were gathered, including representatives of other EU-funded projects such as EchoGRID and BRIDGE.
- Hands-on demonstrations were performed at OGF 23 in Barcelona, Spain, during the CoreGRID industrial days.
- GridCOMP also participated in EC concertation meetings of Unit D3 "Software and Service Architectures and Infrastructures" and demonstrated its prototypes in Brussels on 22-23 September 2008.
- GridCOMP organised its first conference open to the overall Grid community in Sophia Antipolis on 21 October 2008. This event was an opportunity to hear the feedback from several industrials on GCM performance.
- Another meeting entitled "From Components to Services to Utilities" took place on 22 October 2008 and gathered many EU-projects which may be interested in using GCM.
- GridCOMP will also showcase its demos at ICT 2008 organised by the European commission on 25-27 November 2008 in Lyon, France, where 4000 delegates are expected.

Publications: GridCOMP consortium published more than 40 scientific papers, and gave numerous presentations during scientific seminars and collaboration meetings.
Articles have also been produced on CORDIS and eStrategies Projects magazine.

Standardisation

GridCOMP partners participate in common standardisation body activities to enforce and promote GCM. INRIA is involved in the ETSI TC Grid. "GCM Interoperability Deployment" and "GCM Interoperability Application Description" have been accepted as ETSI standards, while two parts "GCM Fractal ADL" and "GCM Management API (Java, C, WSDL)" have officially been accepted as Work Items and are in the process of being standardised. The GCM standardisation ensures interoperability to the GCM user and guarantees support from industry.

ETSI World Class Standards

Several actions targeted towards the Chinese stakeholders have also been undertaken owing to Tsinghua University, partner in Beijing. Hence, Tsinghua University made ChinaGrid CGSP become one protocol compliant with GCM. Moreover, Tsinghua prepared the Chinese GridCOMP website, flyers, and GCM/ProActive was spread throughout the ChinaGrid community.

GridCOMP Structure

7. Collaboration

Objectives

The cooperation aims at exploiting synergy between the EU-projects and increasing the impact of the ICT initiative. The goal is also to exchange information and exploit synergies among the work developed by the projects relevant to the Grid activity.

Main Activities and achievements

GridCOMP consortium participated in many collaboration activities with several EU-funded projects.

Interactions with NESSI

NESSI-Grid has referenced the GCM in their Strategic Research Agenda (SRA) and GridCOMP partners contributed to the second and third version of the NESSI-Grid SRA.
GridCOMP also provided contributions to NESSI TG1 for the White Paper on Architectural Descriptions of Grid Projects.

Interactions with EGEE

In 2007 and 2008, GridCOMP contributed to the EGEE Achievements booklet which was published in October 2007 and October 2008, thus presenting the project success story.

Interactions with EchoGRID and BRIDGE

GridCOMP participated to a joint meeting with the EchoGRID and BRIDGE projects during the Grids@Work week in Oct-Nov 2007. GridCOMP was presented to the audience and it was then followed by a discussion comparing Proactive/GCM and other middleware (EGEE gLite, GRIA, CNGrid GOS and CROWNE) to analyse their pros and cons of running e-science and industrial applications. This cooperation with EchoGRID and BRIDGE projects was very successful and paved the way for further collaboration activities with these projects. Additionally, GridCOMP submitted an application for a joint exhibition stand with EchoGRID and BRIDGE in order to participate in the ICT

2008 event in Lyon, France, on 25-27 November 2008. The joint exhibition stand showcasing project demos has been accepted and will be manned in the International Village.

Interactions with CoreGRID

GridCOMP has a close collaboration with the CoreGRID Network of Excellence and the partners participated in many CoreGRID events such as the Summer schools in 2007 and 2008. GridCOMP partners also attended OGF 23 in Barcelona and participated in the CoreGRID industrial days. On 4-5 June 2008, INRIA, University of Pisa, ISTI/CNR and University of Westminster presented demos, showing the implementation of some of the CoreGRID definitions like the GCM programming model within GridCOMP.

Interactions with XTreemOS

Following the recommendation of the Commission to develop cooperation with the XTreemOS Integrated Project, GCM and its reference implementation developed within GridCOMP was presented during the XTreemOS general meeting in Slovenia. This presentation was the occasion to plan support of the GCM within XtreemOS.

Reference in Gridipedia- interaction with BEinGRID

The ProActive/GCM middleware providing the reference implementation of the GCM has been referenced in the Gridipedia public repository managed by the BEinGRID EU-funded project.

Interactions with ChinaGrid

Through encapsulating ChinaGrid CGSP main component into GCM component, ChinaGrid users can use ChinaGrid resource through GridCOMP programming interface. At the same time, CGSP is also one deployment protocol of GCM, and GCM users can use ChinaGrid resource through GCM.

Finally, GridCOMP organised a technical concertation meeting with the participation of EC-funded projects from INFSO/D3 Unit (Software & Service Architectures and Infrastructures) and INFSO/F3 Unit (GÉANT & e-Infrastructure). About fifteen projects were invited for this meeting held on 23 October 2008 in Sophia Antipolis. The key objective of this meeting was to stimulate cooperation and knowledge exchange between all members of EU projects related to Components, Services, and Utilities for Grid and large scale IT systems. This meeting gives strong visibility to participating projects and helps to achieve dissemination and collaboration objectives. The agenda is available at http://gridcomp.ercim.org/content/view/37/16/.

International Cooperation

GridCOMP can reach out to a worldwide audience owing to its partners in China, Chile and Australia.

Tsinghua University is the most reputable University in China, especially in the Information Technology area. As a founding member, Tsinghua University attended the construction of all three Grids of China, ChinaGrid (China Education and Research Grid), CNGrid (China National Grid), and CROWN (China Research and development environment Over Wide-area Network). The key member of the GridCOMP group at Tsinghua University is Dr. Yongwei Wu, who works in the Department of Computer Science and Technology, and his current research interests include Grid computing, distributed and parallel computing technology, and symbolic computation.

To build an effective Grid component framework, Tsinghua wrapped the legacy code and developed specific techniques for turning the codes into components and deploying the components remotely. In addition, Tsinghua developed a Node Resource Monitor prototype which was integrated into the GIDE. Tsinghua also ensures that GridCOMP interoperates with ChinaGrid Support Platform (CGSP), a grid middleware developed for China Grid. CGSP integrates all sorts of heterogeneous resources, especially education and research resources distributed all over the China Education and Research Network (CERNET), and provides transparent and convenient grid services for science research and high education.

Yongwei Wu
Tsinghua University, Beijing, China
wuyw@tsinghua.edu.cn

The University of Chile is the oldest and main university in Chile. The Distributed and Parallel Research Group at the Computer Science Department (DCC) performs research on all aspects of distributed and parallel systems, in particular programming languages and environments. The group is getting stronger in research at an international level since a technology transfer program devoted to high-speed networking has been very successful in creating new companies and getting development contracts from Chile, Japan, Korea and the US.

Dr José Piquer is Associated Professor at the Computer Science Department of the University of Chile, where he has been working in Distributed Systems programming and Networks since 1986. He is currently Director of the NIC-Chile research labs, and leader of many Chilean research and development programs. Main themes of his current research are Distributed Garbage Collection (for Distributed Programming Languages), Multimedia Network Adaptive Applications and Advanced Networks.

The University of Chile developed a complete group framework refactoring for GridCOMP, and the software comprises about 10 new classes for the core implementation of Dynamic Multicast and Gathercast.

José Piquer
University of Chile, Santiago, Chile
jpiquer@nic.cl

The University of Melbourne is an international research and teaching university, with a huge experience in Grid environments exploiting Web Services. Dr. Rajkumar Buyya is the Director of Grid Computing and Distributed Systems Laboratory (GRIDS Lab) which is engaged in the research and innovation for realising Next Generation Cloud and Grid Computing. He supervises a research team of over 20 members, and led the creation of a new research group at Melbourne University on Parallel and Distributed Computing which involves many academic staff members. He currently serves as CEO of Manjrasoft, a company set up to commercialise technologies developed by the GRIDS laboratory.

UoM developed seamless integration of two complex systems, component-based distributed application framework, ProActive and Gridbus Resource Broker. The integration solution provides: (i) the potential ability for component-based distributed applications developed using ProActive framework, to leverage the economy-based and data intensive scheduling algorithms provided by the Gridbus Resource Broker, and (ii) the execution runtime environment from ProActive for the Gridbus Resource Broker over component-based distributed applications.

Rajkumar Buyya
University
of Melbourne,
Melbourne, Australia
raj@csse.unimelb.edu.au

Exploitation

Target audience

The GridCOMP results target all software architects in need of a comprehensive framework to express at design time the parallelisms and the distribution of an application. Therefore, the architecture of the system itself captures the parallel/distributed aspects, acting as a powerful specification and documentation. Developers will not need to spend extensive time to learn distribute programming or implement collective communication, but rather concentrate on the business code and leverage the GCM framework.
The INRIA spin-off company, ActiveEon provides support and services around GCM and ProActive, uses standard GCM deployment for all developments and uses a standard GCM programming model to develop components, and build component applications.
In addition, projects such as SOA4ALL (EU), INRIA ADT Galaxy, Pole de Compétitivité AGOS (with HP, Oracle), and QosCosGrid (EU) use ProActive the GCM reference implementation.
Regarding the **Non Functional** work package, the results target all those communities of programmers that need to develop efficient grid parallel applications exploiting parallelism according to well known parallel patterns (the ones modelled within the behavioural skeletons). In particular, application experts not particularly expert in parallelism exploitation on grids, will enjoy the ease of writing applications using behavioural skeletons. Therefore, the main target audience of WP3 results comprises all those programmers who need to develop parallel applications (for performance reasons) without being obliged to spend too much time fine tuning the parallelism exploitation related code aspects.
For the **GIDE**, the platform is aimed at supporting several different user groups: application developers, application users, and data centre operators. Being an open-source software package that conforms to the Eclipse license, GIDE is freely available for download and installation, which has been very important in maintaining very close collaborative links with CoreGRID and other EU projects via our collaboration activities.
The **use cases** are important to promote the capabilities of the GridCOMP framework. Therefore, the target audience is not only applications developers starting to use the framework but all potential users, including software architects and Quality Assurance managers.
For example, IBM, with the biometric identification use case intends to attract people coming from other application domains such as business process management or security by demonstrating what advanced Grid middleware could do for them.
In addition, Atos Origin plans to use the GridCOMP results to develop new applications/solutions for several internal departments and for Atos' clients with the same needs as the selected use case (most of the sectors use Oracle-PL/SQL).
GridSystems, INRIA, and the ActiveEon spin-off worked at integrating the results of GridCOMP within the GridSystems' middleware, Fura. From the INRIA point of view, this adds a powerful resource manager to the GridCOMP platform, which can be exploited in any market. From the point of view of GridSystems, GridCOMP adds on top of Fura a distributed programming paradigm that complements the offer of the company in the telco, engineering, and financial sectors. GridSystems plans to release a Fura version integrable with GridCOMP during 2009, and the use case applications will be used as demonstrators.

GridCOMP

Effective Components for the Grids