GFD-I.70 Shantenu Jha, University College London
SAGA-RG Andre Merzky, Vrije Universiteit, Amsterdam
May 09 2006

A Collection of Use Cases for a Simple API for Grid Applications

Status of This Document

This document provides information to the GGF Applications Area in the Grid Standards council. It does not define any standards or technical recommendations. Distribution is unlimited.

Copyright Notice

Abstract

This document collects the use cases received by the Simple API for Grid Applications research group (SAGA-RG) and presents them for the purpose of reference in future documents.

Contents

Introduction

The SAGA Research Group of the GGF aims to define a high level, application oriented Grid API - a Simple API for Grid Applications (SAGA). This document collects and describes the use cases received in response to an elicitation for use cases by the SAGA Research Group. It aims to ensure, that the API requirements (which are described in the next documen SAGA RG) meet what the target community expects from such an API.

In response to the first call for use cases, the SAGA-RG received 5 use cases by GGF12 (September, 2004), and a commitment for at least 4 more at GGF12. However, it was agreed that more use cases were required and that more use cases could possibly be gathered without an unreasonable amount of additional effort. Hence the SAGA-RG issued a renewed call for Use Cases, with the request to submit responses by the 15th October, 2004. By the close of the

second call the SAGA-RG had received 23 use cases. Of these, 9 use cases were forwarded by the GridRPC-WG of the GGF thanks to the involvement and efforts by Craig Lee.

The first two sections of this document present the use case template as it was sent to various mailing lists and individuals who offered to liase with potentially interested parties, along with the acompanying Notes to facilitate drafting responses by providing sample answers to the use case template. Subsequent sections contain the use cases, presented as returned from the various communities, with only minor editorial (e.g. formatting) corrections.

Seeking and collating use cases represented just the initial efforts in the overall design and implementation of a SAGA API. Significant effort was expended in analysing the use cases received. For example, the three functional areas most requested by the use cases were, job management, resource management and data management. Similarly the three most important non-functional requirements that seem to emanate from the analysis were error handling, security and auditing. This helped define the functional areas, scope and focus of the design team for an initial version of the SAGA API. Detailed results of the analysis are available as a GGF document, "A Requirements Analysis for a Simple API for Grid Applications" (GFD.71).

We anticipate that this document may get updated as new use cases arrive during the SAGA API specification phase.

For more detailed and updated information on the SAGA RG, its scope, and for related documents, visit the group's home page at

http://forge.gridforum.org/projects/saga-rg/

Use Case Template -- Guidance Notes

This is a template for providing use cases to the GGF "Simple API for Grid Applications" research group (SAGA-RG); these use cases will be used to determine required functionality which must be supported by the resulting SAGA-API specification, and

to provide guidance to implementors of the API. We have kept these two features in mind while creating this template.

All sections apart from use case name and the description of the use case itself (3) may be considered optional.

If the use case is complex, we would appreciate efforts to split into sub-scenarios where possible, for example in the use case description (3) and in the descriptions of customers (4) and resources (5).

In each section we have tried to provide suggestions of questions to try to answer.

Please send the use cases to <saga-rg@ggf.org>, or if you want to submit anonymously for any reason, send directly to one of the group chairs or secretaries (see http://www.gridforge.org/).

Use Case Template

Name of use case: .....
Contact (name and address): .....
Authors (if different form contact) .....

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[ ]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[ ]
Please specify:	.....

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[ ]
Visualization	[ ]
Security	[ ]
Resource discovery	[ ]
Resource scheduling	[ ]
Workflow	[ ]
Data movement	[ ]
High Throughput Computing	[ ]

```
    High Performance Computing   [ ]
    Other                        [ ]
        Please specify:          ..............................
```

1.4 Are you an:

```
    Application user             [ ]
    Application developer        [ ]
    System administrator         [ ]
    Service developer            [ ]
    Computer science researcher  [ ]
    Other                        [ ]
        Please specify:          ..............................
```

2. Introduction:

2.1 Provide a paragraph introduction to your use case.
Background to the project is another alternative.
(E.g. 100 words).

...

2.2 Is there a URL with more information about the project ?

...

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs.
E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality,
please provide separate descriptions for each component,
bullet points of 50 words per functionality are acceptable.

...

4. Customers:

Describe customers of this use case and their needs. In particular, where and how the use case occurs "in nature" and for whom it occurs. E.g. max 40 words

...

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

...

5.2 Are these resources geographically distributed?

...

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

...

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

...

5.5 What information sources do you require, e.g. certificate authorities, or registries.

...

5.6 Do you use any resources other than traditional compute
or data resources, e.g. telescopes, microscopes, medical
imaging instruments.

...

5.7 How often is your application used on the grid or grid-like
systems?

[ ] Exclusively
[ ] Often (say 50-50)
[ ] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in,
for example the languages used, the tool-sets used, and the
user environments (e.g. shell, scripting language, or portal).

...

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent
in the application, and which are chosen by system
administrators, or by other means? E.g. who is specifying
the architecture and memory to run the remote tasks?

...

7.2 How are the resources selected? E.g. by OS, by CPU power,
by memory, don't care, by cost, frequency of availability

of information, size of datasets?

...

7.3 Are the resource requirements dynamic or static?

...

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

...

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

...

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

...

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

...

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

...

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

...

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

...

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

...

13. References:

List references for further reading.

...

UC 1: CoreGRID integrated toolkit

Name of use case: CoreGRID integrated toolkit

Contact: Rosa M. Badia <rosab@ac.upc.es>

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[X]

Please specify: Programming environment for Grid

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[ ]
Visualization	[ ]
Security	[4]
Resource discovery	[5]
Resource scheduling	[5]
Workflow	[4]
Data movement	[3]
High Throughput Computing	[ ]

High Performance Computing	[3]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[ ]
Application developer	[ ]
System administrator	[ ]
Service developer	[ ]
Computer science researcher	[X]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case. Background to the project is another alternative. (E.g. 100 words).

In CoreGRID WP7, task 7.3 an Integrated Toolkit will be designed. The objectives of this Integrated Toolkit is to provide a tools framework that allows application developpers to easily develop grid applications over an invisible grid. Project CoreGRID is just at its beginning, and therefore this use case is a description of the initial goals we have in this WP/task.

2.2 Is there a URL with more information about the project ?

http://www.coregrid.net

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs. E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

Based on a desciption of the application on an imperative language, the integrated toolkit will be able to run it in the grid, performing automatic parallelization, resource descovery, scheduling, ...

The following functionality will be required:
- Resource descovery
- Job submission
- Job status notification
- File transfer

4. Customers:

Describe customers of this use case and their needs. In particular, where and how the use case occurs "in nature" and for whom it occurs. E.g. max 40 words

If the customers you are asking for are the customers that will develop the integrated toolkit, then computer scientists with background in high-performance computing.

The final users of the integrated toolkit can be any scientist aiming to use the grid in an efficient way with low development effort.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

At current stage it is difficult to list this, however a pan-european testbed with heterogeneous resources it is planned insider CoreGRID.

The fundations of the integrated toolkit will be component oriented.

5.2 Are these resources geographically distributed?

Yes, at European level probably.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

See 5.1

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

N/A

5.5 What information sources do you require, e.g. certificate authorities, or registries.

N/A

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

Initially not.

5.7 How often is your application used on the grid or grid-like systems?

[X] Exclusively
[ ] Often (say 50-50)
[ ] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

The development languages will range between C and Java probably. The idea is to provide the final user with a wide range of input laguages, for example: C, C++, Perl, Java, shell script...

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

An application-level metadata repository will be also designed inside the same project. The information stored by this system can be used by the integrated toolkit to automatically perform resource discovery and allocation.

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

Not decided yet. Probably we can think of not only taking into account static information (CPU power, memory) but also dynamic (load level, network congestion...)

7.3 Are the resource requirements dynamic or static?

Probably dynamic.

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

Initially any resource is specially sensitive.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

Some partners have experience with Unicore.

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

Authentication, authorisation, data protection.

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

Integration with commodity technologies.

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

Compute resources and networks.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

The objective of the integrated toolkit is to increase the performance of the (sequential) application. Therefore, it is important that the undelying middleware does not reduce this perfomance.

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

Differents partners are using different solutions: GT2.4, GT3, Unicore,...

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

```
job_submit             (job_description, resource)
job_state              (job_id)
job_cancel             (job_id)
copy_remote_file       (source, dest)
erase_remote_files     (file1, file2, ..., resource)
wait_for_notifications ()
resource_status        (resource)
```

13. References:

List references for further reading.

UC 2: Cyber Infrastructure

Name of use case: Cyber Infrastructure
for Coastal Modeling

Contact: Chirag Dekate <cdekate@cct.lsu.edu>
Dr.Gabrielle Allen <gallen@cct.lsu.edu>
Dr.Greg Stone <gagreg@lsu.edu>

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organization:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[X]
Image analysis	[ ]
Other	[ ]
Please specify:	Oceanography (Coastal modeling - Storm Surge Models, Wave Models etc.)

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[3]
Remote steering	[3]
Visualization	[5]
Security	[2]
Resource discovery	[5]

Resource scheduling	[5]
Workflow	[4]
Data movement	[4]
High Throughput Computing	[5]
High Performance Computing	[5]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[ ]
Application developer	[ ]
System administrator	[ ]
Service developer	[ ]
Computer science researcher	[X]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case.
Background to the project is another alternative.
(E.g. 100 words).

WAVCIS program at LSU has various sensors off Louisiana coast which relay real time measurements (wind speed, wind direction, salinity, turbidity ...etc.). The measurements are displayed in real time in the program web page [1] and the post-processed results are archived in a SQL server. The datasets obtained from the sensors are then used to run coastal wave models in near real time, thus allowing researchers to validate models by comparing the model outputs with real-time offshore observations. Current research activity includes coupling of models where a model running over a larger area feeds output to another model running over a smaller region with higher resolution. For instance, we ran WAM [2] for the entire Gulf of Mexico region and fed the results into the SWAN [3] which was running for the Louisiana Coastal regions. Future plans include nesting of the SWAN model with multiple instances of SWAN models running for extremely high resolution and well defined smaller regions within the current SWAN run area. An illustration of this can be seen at [4]. The SCOOP project intends to integrate data

from disparate sources and different model outputs relevant to each regional area displayed using interactive means such as GIS based web-services.

2.2 Is there a URL with more information about the project ?

http://cs.cct.lsu.edu/wiki/space/SCOOP
http://wavcis.csi.lsu.edu/ http://e117-1.csi.lsu.edu/swan/

3. Use Case to Motivate Functionality Within a Simple API:

Scenario 1: Simple sensor-data based Modeling.

- Retrieve data from sensor database (RDBMS)
- Generate parameter files for model based on the sensor data.
- Stage swan model on computation resource
- Copy output files from computation resource to visualization resource
- Visualize using GIS enabled client

Scenario 2: Coupled Simulations

- Obtain Wind data files from storage repository
- Stage large area WAM model
- Copy the output results to visualization resource
- Copy one set of output results to SWAN modeling resource
- Stage SWAN model on the smaller region using the output obtained in the previous step as boundary condition
- Copy SWAN output to visualization resource
- Visualize both outputs using a GIS enabled visualization client.

Scenario 3: Coupled & Nested Simulations

- Obtain Wind data files from storage repository
- Stage large area WAM model
- Copy the output results to visualization resource
- Copy one set of output results to SWAN modeling resource
- Stage SWAN model on the smaller region using the output obtained in the previous step as boundary condition

- Copy SWAN output to visualization resource
- Copy SWAN output generated for selected regions over to the Task Farming resource
- Stage multiple instances of SWAN using the forcing conditions from the previous step over small regions with very high resolution.
- Copy output from all the runs to the visualization resource
- Visualize all the outputs using GIS enabled visualization clients.

All of the above scenarios will be carried out on different resources across various universities and research centers. Under such a diverse set of users it is very difficult to ensure a homogeneous environment for running grid application. For instance all the computational and data movement at LSU might be using grid technologies such as Globus, and other partners might be using traditional technologies such as SSH, SFTP, SCP, fork/exec etc. Hence a simple API is needed which allows user/modeler to transcend such limitations by adapting to the underlying infrastructure dynamically, with little interference from the user, thus creating a middleware transparency for the end user.

4. Customers:

Model Developers

The results obtained above will be compared with real-time observations obtained from sensors. Using comparative tools such as a time series graph comparing the model runs for specific parameter can be generated. Such comparisons, known as model validation, can result in modification to the physics of the model for that specific region.

Emergency Responders

Model predictions (with a healthy dose of skepticism) can be used for forecasting rough wave and current conditions during events such as hurricanes and tropical storms. This information could help public safety officials give alerts based on variety of information including model results.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

Hardware Resources:

- Large storage device for archiving input and output data of models (10 TB)
- HPC resources such as multiprocessor clusters as computation resources (Super Mike, Super Helix, etc...)

Software Resources:

- GIS software for Visualization and data analysis
- RDBMS for storage of Sensor data
- Grid Middleware (Globus, MPICH-G2, Cactus Task Farming, Condor)

Model Resources:

- Atmospheric Model results from variety of models : MM5, WRF, COAMPS
- Ensemble of Wave Models : WAM, SWAN, STWAVE, WaveWatch III, etc.
- Circulation and Surge Models : ADCIRC, ELCIRC etc.
- Other Models : Sediment Transport Models, CH3D, CH1D etc.

5.2 Are these resources geographically distributed?

The computational resources will be geographically distributed across institutions and supercomputing centers across the globe.

The input data stores will also be distributed across US. For instance the wind forcing predictions from atmospheric models are currently obtained from Jackson State University, Mississipi. University of Miami and UNC are other major sources for critical input data.

Visualization resources will also be distributed across the US.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

Some of our initial test cases use (4-processor systems, 8-node clusters, 2-processor systems and single processor systems). Even in these test cases our models take multiple hours to run (13 hours in some cases). Nesting and coupling are computationally intensive and dependency based systems, hence are time consuming. Using Cactus TFM we have reduced the time it takes to run the nested SWAN runs to acceptable limits. Models are being developed to be able to use the larger resources they need. Currently we have been able to take advantage of compute resources provided by large local clusters.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

MODEL

SWAN
  Language : Fortran with MPI extensions
  License : Opensource
  Limitations : Will not compile using GNU compilers
  Need PGI, Intel etc.

WAM
  Language : Fortran
  License : Closed source, can obtain code for
  research purposes
  Limitations : No Compiler limitations.

STWAVE
  Language : Fortran
  License : Closed source, can obtain code for
  research purposes

Limitations : No Compiler limitations

ADCIRC
Language : Fortran
License : Closed source, only binaries available
Limitations : Unknown, only binary executable available.

5.5 What information sources do you require, e.g. certificate authorities, or registries.

CA's, job managers, Database Systems

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

Our offshore stations have a total of 25 sensors including 2 ADCP (Acoustic Doppler Current Profiler). All the data obtained from the sensors is download using a satellite link to the WAVCIS databases.

5.7 How often is your application used on the grid or grid-like systems?

[ ] Exclusively
[X] Often (say 50-50)
[ ] Occasionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

We use a combination of Scripts (bash, csh), C programs and Gridsphere portal for staging the models.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

Users should be able to specify their own resources. In case of uncertainty or complex constraints, the user selects resource broker which negotiates the best possible resources for the user.

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

The resource is usually selected on the basis of the compatibility of the code with the underlying resources (compiler and OS)

7.3 Are the resource requirements dynamic or static?

The resource requirements are usually static , except in the case of special events such as Hurricanes and Tropical Storms when on-demand modeling iterated over short periods of time would be needed.

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

Results obtained from the ensemble models will form a critical component in making executive decisions during times of emergency. For instance in case of extreme events such as hurricanes etc, the storm surge prediction obtained from the models may be one of the factors to trigger evacuations of low lying regions. Hence security of the results obtained and role based availability (model developers have full access to the model code and the results, only a section of results available to the general public, all possible results

available to decision makers presented in relevant GIS based clients) of final products.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

GSI, SSH, Globus Simple CA Planned use of Gridsphere to provide role based access to results and data sets.

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

Authentication, authorization to ensure that only users with select privileges can run the stage the models and change parameter files. Audit trails should be required to ensure that the model has been executed on trusted resources that have not been compromised. Due to the sensitivity of the results and the nature in which they will be used (triggering evacuations and making critical road closures etc.) It is highly imperative that the output data be secured in the most efficient way possible.

8.4 What are the most important issues which would simplify your security solution?

The Security API that is developed should be easy to deploy and integrate within the existing framework with little modification while providing maximum functionality.

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

Under certain circumstances the models might require additional resources to be provided dynamically. Usually these are in the form of compute resources and storage resources depending on the Model needs

10. Performance Considerations:

In standard case all the simulations will be executed in a timely manner under a rigid schedule the large area WAM model coupled with medium area SWAN model will be run once a day. The nested SWAN model will be run every hour using sensor data and other model results generated in the previous step.

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

Current : GLOBUS, GAT, GridSphere
Planned : Condor

12. What Would You Like an API to Look Like?

The API should be very simple to use and program for end users. In a diverse community such as SCOOP, it is safe to assume that the underlying infrastructure is heterogenous. Hence any API should be able to deal with different architectures based on varied middleware.

For instance API_FileCopy(Source, Destination) would move file from source to destination irrespective of the underlying framework or middleware (scp, gridFTP..) etc. There by the user can focus on the task at hand (running models) rather than worrying about the underlying intricacies of the Grid.

13. References:

[1] WAVCIS Website, http://wavcis.csi.lsu.edu/

[2] Komen, G. J., Cavaleri, L., Donelan, M., Hasselmann, K., Hasselmann, S. and P. A. E. M. Janssen: "Dynamics and Modelling of Ocean Waves", 1994, Cambridge University Press, 532 p.

[3] Booij, N., R. C. Ris and L. H. Holthuijsen: "A third-generation wave model for coastal regions, Part I, Model description and validation". 1999, J. Geoph. Research, C4, 104, 7649-7666.

[4] http://e117-1.csi.lsu.edu/swan/images/modelareas.jpg

UC 3: DiVA

Name of use case: DiVA: Distributed Visualization Architecture

Contact: Wes Bethel <ewbethel@lbl.gov>
John Shalf <jshalf@lbl.gov>
Lawrence Berkeley National Laboratory
Mailstop 50F, 1 Cyclotron Rd.
Berkeley CA, 94611, USA.

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[X]
Please specify:	Distributed Data Analysis Environments that cut across all of the above fields

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[1]
Remote steering	[3]
Visualization	[5]
Security	[4]
Resource discovery	[5]
Resource scheduling	[4]

```
Workflow                    [5]
Data movement               [4]
High Throughput Computing   [1]
High Performance Computing  [3]
Other                       [3]
    Please specify:  remote/distributed data analysis
                     infrastructure, interactive data
                     analysis environments
```

1.4 Are you an:

```
Application user            [X]
Application developer       [X]
System administrator        [ ]
Service developer           [X]
Computer science researcher [X]
Other                       [ ]
    Please specify:         ..............................
```

2. Introduction:

2.1 Provide a paragraph introduction to your use case . Background to the project is another alternative. (E.g. 100 words).

Our vision is for scientific researchers to be able to perform visualization anytime, anywhere, using any of a large collection of resources, and without the need to be master of many different disciplines. Furthermore, we envision a stable environment for visualization and data analysis that can be used effectively by all scientific researchers as well as for visualization research by the scientific visualization community. We envision high performance visualization tools having the same ease of use and prevalence as common office productivity software. We strive for the ability to bring to bear the combined resources of many remotely distributed computational resources upon a single scientific visualization task that exceeds the capabilities of any one platform.

In the context of GGF/SAGA, the goal of DiVA is to create a common infrastructure that makes it it simpler to develop

and deploy Remote/Distributed Visualization applications. Visualization application developers are spending 80% of their time on wrangling transport, job launch, and security issues. This is particularly difficult because most the issues involved in transport, job-launching, and security are outside of the visualization application developer's core expertise, so these tasks take a long time and the developers do a poor job of it. This leaves a very small amount of time for visualization algorithms and tool functionality. We believe that such functionality requires simplified API's that address these needs (security, job-launching, data transport, and resource brokering) in a more domain-specific manner in order to present the application developer with a simpler, more compact, and easier-to-understand API to serve their needs.

2.2 Is there a URL with more information about the project ?

http://vis.lbl.gov/Research/DiVA/index.html

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs. E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

Use case for Graphical Workflow Tools (tools with interfaces like AVS, SciRun, and Amira). This use case could also apply to script-based interfaces that define a workflow (eg. VTK examples).

Definition "component" : A self-contained program unit with well-defined interfaces, but opaque internal operation. In the context of visualization workflows, a components are connected together to create applications dynamically. (interchangable with the term "modules" in the context of dataflow/workflow-based visualization applications). In the

context of a "Grid," persistent services that are part of the workflow would also appear transparently as "components" to the user even though they may only exist at one specific location. This kind of merging of the concepts of "servcies" and "components" in this manner violates the definitions of component and service. However, from the application user's point of view, it is important to minimize this distinction when possible.

The DiVA infrastructure should be able to

1: make the data source (whether it is a file or an instrument) available to the distributed workflow via a secure socket.

2. use performance prediction and resource availability information to assign the components of the "abstract workflow" to a concrete set of resources -- thereby creating a concrete workflow that connects the remote user to the remote data source in the most efficient/responsive pipeline possible.

3. Launch the pieces of the concrete workflow and monitor them so that the components can be relaunched in the event of component failure, contract violation, or new user requirements.

More Details

DiVA infrastructure must;

a) Discover available components on distributed resources. The list of available components must be searchable by different attributes. This overlaps the needs of RealtiyGrid.

b) User specifies the "abstract workflow". This is the set of operations that must be performed for the visualization application and the dependencies/connections between them. This is done using a graphical editor, but could also be done using scripting languages (eg. like VTK).

c) Select a file on some remote resource. The low-tech method is to make remote directory listings available on a per-host basis. The desired methodology would be to use a replica catalog service to provide a global virtual file

space for the purpose of identifying data files to read from. A similar "virtual space" would be useful for selecting running simulation programs that act as the source for data. This overlaps the needs of RealityGrid.

d) Once the data source has been identified, the "abstract" pipeline needs to be mapped onto real resources -- converted to a concrete pipeline. It is desirable that the mapping provide the best possible performance (minimizing response time in order to maximize interactivity... even at the expense of throughput). This requires a resource broker that understands the execution/flow dependencies between the components in a workflow and can select an optimal mapping of the pipeline onto the resources. It is presumed the resource selector takes into account the availability of components as per item "a)".

e) Once the resources have been identified, the components must be launched/executed on the selected resources in order to instantiate the pipeline.

f) The components will need to establish secure socket links where they are not located in the same memory space. In some contexts, this may only be authenticated socket connections, but other contexts require the sockets be encrypted as well.

g) Interactive visualization applications are very sensitive to QoS. The establishment of socket connections between components may require negotiation for QoS that should be hidden behind a simplified/common API.

h) The performance of a distributed workflow must be constantly monitored and components in the pipeline may need to be restarted or migrated to new hosts in the event of a failure or change in requirements. It is possible that APIs developed for user-space CPR for simulation applications can be leverage to support application fault-tolerance for distributed visualization workflows.

a) On startup, the application must gather a list of available "components". Typically this is done by consulting a local configuration file to find the locations of the binaries (or bytecode files) associated with each component as well as their names and interface definitions. For DiVA, we would like to support the discovery of remote modules as

well by contacting information services on other machines or a broker that locates components on all machines in a given Virtual Organization. From the application programmer's point of view, they want to be presented with a searchable database of components (regardless of location) that can be queried and sorted based on criteria such as "name," "location," interface definition, etc... Organization as an Relational Database or LDAP directory or even a flat-file is unimportant. The API should be able to hide these details as a query for components that satisfy the search criteria is presented.

b) The user will then select components to construct an abstract workflow that implements the visualization pipeline they desire. By abstract workflow, I mean that the components simply identify the algorithm to be performed without regard to the location of an implementation for that algorithm. So the workflow is defined without an explicit mapping to a particular set of resources. In this case, we will define an arbitrary workflow involving a
- Data "Reader" that is connected to ..
- an "IsoSurfacer" that is connected to...
- a "Renderer" that is connected to...
- a display window.

Please note that the "Reader" could easily be replaced by a connection to a running simulation code (eg. a remote-steering scenario), which would make component of the workflow "concrete". A mixture of "concrete" and "abstract" components should be supported, but the most general case is a completely abstract pipeline.

c) File Selection: The user browses for files to read using the GUI for the "Reader" component. The "reader" has not yet been instantiated, so the browser is actually doing remote reads of directories or is consulting a replica catalog. In the first scenario, the file browser widget associated with the abstract "reader" would present the user with a list of available machines where that "abstract reader" can be instantiated (the location of this is determined by the service described in step A). Selecting a machine will provide the user with a remote directory listing for that machine. When the user selects a file, the remote reader is instantiated on the machine where the file has been selected. This is the method employed by the LLNL VisIT remote visualization tool.

An alternative and more elegant approach is to use a replica catalog. In this case, the user is presented with different "catalogs" to choose from. The catalogs are abstract directory structures that can span many machines and can manage files that are replicated across different filesystems as a unified directory. In this scenario, when the user selects a file to be read, the application can choose from number of resources rather than just a single machine when it comes time to instantiate the reader.

The important point here is that we need an API that will make it easier to browse and select files in a distributed environment. Currently this task can constitute a rather complicated application on its own. It should be made nearly as simple in practice as the API's we use to get file listings for file-selection-widgets that browse local directories.

d) Mapping: In this step, the "abstract workflow" is converted into a "concrete workflow" through performance prediction followed by resource selection. This step is entirely hidden from the user, but is absolutely critical for proper operation. Once the reader has been instantiated (or the set of resources on which the reader *can* be instantiated is narrowed via the replica catalog), the system has enough information to begin instantiating the rest of the pipeline. However, it must do so in a manner that maximizes the pipeline performance depending on the user's current performance objective. It would otherwise be impractical for the user to manually select resources to meet these objectives. Here are some sample objectives (eventually the system should be able to automatically switch between which objective function it is optimizing)

- fastest response time when selecting a new dataset (eg. shuttling through timesteps)
- fastest response time when changing a parameter on a filter (eg. changing the isosurface level)
- fastest response time when redrawing the geometric model (eg. local OpenGL vs. parallel offscreen rendering at a remote host)

Optimizing this objective function requires a means to predict the performance of each component on a given machine with current operating conditions. The input parameters for the model can be "machine characteristics" and "data sizes"

(there are more potential parameters, but this is a start). The output is the prediction for the performance of that component on a given machine. In addition, the Network Weather Service or other performance modeling methods can be employed to predict the time taken to transfer data between each of the distributed hosts involved in the workflow.

Once you have a performance model that enables you to predict the performance of individual components and network links (even if the model contains considerable uncertainties), you can employ a Dijkstra weighted shortest path algorithm to find the set of resources that provide fastest response time in Order(N logN) complexity (where N is the number of resources available). That is to say, the mapping can be done in a fraction of a second on most machines and is highly scalable.

We need the API to provide a simplified interface to this decision making system for mapping the abstract workflow to a fixed/concrete set of resources. Existing brokers do not understand the dependencies between the different components that comprise a distributed workflow, and therefore will not be able to map the workflow in a manner that will offer reasonable performance.

e) Launching the components: After the mapping of the abstract pipeline to concrete resources occurs, the application manager (or app controller) needs to be able to launch all of the components of the pipeline using the best available means at its disposal. While RSLs offer a great degree of flexibility in specifying the job launching criteria, the application programmer *really* would like to call "Launch("host",<the same string you would type on the commandline to manually launch the component>); It should be able to hide the underlying mechanics of the secure launching mechanism.

f) Connecting the Components Together: Once the components are "launched," the components that reside on different machines need to be able to communicate with one another via network sockets. This leads to both security and performance considerations (discussed below). Secure Sockets Abstraction: The state-of-the-art use of sockets for remote visualization depends on security practices that are largely debunked as "unsafe" by our security professionals. Most distributed visualization application developers presume

that if components were launched securely, the sockets that connect these components together are also secure (who could guess the port number and startup time... its security through obscurity). This is clearly not safe, but the level of effort necessary to move to simple authenticated sockets is considerable (it is not practical to expect vis people to do this). At minimum, a new TCP socket must exchange an "shared secret" in order to make sure the connection is "authentic". If there was a single call to "OpenAuthenticatedSocket (host,port)", then that would be very useful. Likewise, an "OpenEncryptedSocket ()" would be useful in other circumstances. For instance, SGI's VizServer does not encrypt the socket used to convey your login & passwd to connect to the server (its as bad as telnet). File directory listings probably should be encrypted. This should just be targeted at a single TCP socket. We can use GridFTP for performance, but we really need to get the control sockets protected. As you know, GridFTP is terrible for metadata and control socket applications.

g) Communications Performance: The components in a vis pipeline are extremely sensitive to QOS. This requires much stronger QoS guarantees than can be delivered by the best-effort packet-switched networks. Recently, there have been a number of major scientific network backbones who support application-negotiated circuits (either PVCs or actual end-to-end lambdas). The current protocols for this negotiation are exceedingly complex.

The visualization application needs to communicate its resource requirements to the network infrastructure. When possible, the network infrastructure can provide a "lease" for a dedicated circuit to a component so that it can switch to fixed-datarate protocols or otherwise provide stronger guarantees of interaction. In the case that the vis application cannot obtain a dedicated circuit, it can still rely on TCP/GridFTP over best-effort networks with some degradation in performance. This negotiation can be simplified through an API that allows the application designer to declare the resource requirements and to install "callbacks" that inform the application of resource availability changes.

h) Runtime Application management: Once we have a running workflow application, it is important to preserve the illusion that the entire application is running in one place

(eg. make it appear to function just the same as your old Amira or AVS pipeline running on your desktop machine). This requires that the components be monitored to detect transient failures or "contract violations" where a resource fails to meet its performance prediction. It also requires interfaces that manage the restart of components with the same parameters on different resources if a migration is required. Unlike typical simulation applications where a full checkpoint-restart is required, the visualization workflows can have most of their state restored from the upstream component in the pipeline. This makes migration less data intensive, but no less complicated in terms of managing the state information.

Monitoring and Fault Tolerance: There must be a simple API that can report wallclock performance for each invokation of a component. The API can report available timers as a list of names and then respond to requests for the performance metrics associated with each of the named timers that are offered by the application. This is necessary as a heartbeat that determines whether a component is still "alive" and for detection of partial failure (sick machines) or outright contract violations (a failure of the performance model or machine).

State Migration/Redeployment: If it is determined that the component has died or needs to be moved in order to meet a different set of performance objectives, there must be a means to report the part of the component's state that must be preserved for it to be restarted in the same state and what state data must be acquired from upstream components in order to complete the components restart in the same state. This kind of state registration is common to many user-space checkpoint-restart APIs that are being discussed in the GGF CPR working group. If we could settle on one API (one that has a special notion of data that can be restored from upstream objects rather than being explicitly stored in the checkpoint), then we can reuse the same API's to enable fault-resiliance for distributed workflows and remote vis applications.

4. Customers:

Describe customers of this use case and their needs. In

particular, where and how the use case occurs "in nature" and for whom it occurs. E.g. max 40 words

The target audience for DiVA are visualization tool developers who wish to provide more sophisticated services for scientists at remote/distributed sites. The playing field for production-quality remote visualization tools is very small and in the case of each tool, restricted to a very limited set of remote-vis paradigms (eg. The typical client-server model selects only a single way to partition the vis workflow -- Ensight only sends remote geometry whereas SGI VizServer sends only compressed images). Currently the effort required to develop robust distributed data analysis applications is so overwhelming, that many such systems fail to graduate to mainstream/production-quality tools. DiVA aims to reduce the amount of effort required to implement such tools by offering an infrastructure that handles job placement, resource selection, and transport requirements.

The ultimate beneficiary of this effort are the domain scientists who need to analyze remotely located datasets that are simply impractical to FTP to their own workstation due to size, or computational requirements.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

Hardware: High end computers for both simulations and visualization Data: A common data model will be extremely useful to ensure interoperable components.

Software: The DiVA needs revolve around supplying software toolkits (tinker-toys) that enable programmers to assemble remote/distributed vis applications much faster than otherwise possible.

Services: Some of the functionality required must be packaged as services. The interfaces to these services can be hidden behind the API, but they are clearly outside of the scope of a fully stand-alone application. DiVA aims to deliver as much functionality as possible without depending entirely on the wide-deployment of these services

(there is always a back-door where services do not yet exist), but there are many benefits when such services are available.

5.2 Are these resources geographically distributed?

Yes.

The majority of users for supercomputing centers like NERSC and the NSF-PACI centers are located far away from the centers. Likewise, sensor data archives are typically located close to the device that generated the data -- resulting in widely distributed data sources.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

Potentially many. It depends on the dataset being analyzed and the algorithms employed. One would hope that the resource selection algorithm in the resource broker will tell you how many resources are appropriate. The focus of the system is to make the launching of the pipeline components easier in the context of RDV applications. Therefore, the number of resources employed is tangent to DiVA's design targets.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

DiVA will use a modified BSD-style OpenSource license. This allows any DiVA components to be used in commercial applications without royalties. The derivative software must acknowledge the source, but need not be OpenSource.

There are no restrictions on what language is employed to write pipeline components. Visualization developers typically write in C, C++ and apply parallelism using MPI or shmem. Many DiVA services will be implemented in C/C++ with multi-language bindings automatically generated using

wrapper generators like BABEL and SWIG.

The first-generation components can be stand-alone programs that communicate via shared memory handles and/or sockets. The DiVA effort is also investigating CCA component technologies that enable dynamic loading and binding between application components using direct pointer hand-off between application components.

5.5 What information sources do you require, e.g. certificate authorities, or registries.

Digital certificates are useful, but not required. The primary CA will be the DOE Science Grid Certificate Authority. We would like the system to be able to fall back to ssh or kerberos when a CA cert is not available.

The system should be able to make use of 3 different levels of services.
* Standalone/Local Service Advertisement: Uses local
* configuration
files to locate local, user-installed components. This is a single XML file that expresses this configuration information.
* Remote Service Advertisement: A machine can advertise the
* locations of components installed to remote users via a
* network information
service. It *only* advertises services that are available on the particular host it represents (eg. the head-node of a cluster can advertise services available on that cluster). This can be an OGSI service or an LDAP/MDS directory.
* Grid: A Virtual Organization should be able to advertise the locations of all Remote Service Advertisements that are available in that VO.

Here is a breakdown of some services identified as important to the DiVA architecture
- Certificate Authority: Of course we all need this... (grid don't work without it).
- Component Locator: A a searchable index for components that may be used in the distributed visualization pipeline. A number of technologies already exist that implement this service, but a common API to interface to this service is desired. The API must be consistent with searches on local indices of components (eg. the list of components that are

identified via local configuration files must be searched/accessed in the same way as for more sophisticated remote network services or even brokers that serve the entire VO.).
- Resource Mapper: This is an optional resource. The mapping can probably be done by the application, but it may take considerable time to gather the necessary performance data to make the mapping decisions. Mapping the abstract pipeline to a concrete set of resources may be done more efficiently with a distinct workflow-oriented brokering service. This distinction (whether it is a remote service or local subroutine call) can be hidden behind an API.
- QoS Negotiator: Abilene and ESNet are designing the protocol supporting application-controlled PVCs. The protocol directs an MPLS control-plane to set up dedicated circuits on behalf of applications at their request. Similar services are already deployed in CanarieNet and the NSF StarLight facility. It is desirable to have a uniform API for requesting PVCs (whether they are Layer-1 light paths or PVC's carved out of a packet-switched backbone).
- Global File Space: This can be provided by a replica catalog or by remote directory listers, or even by a global filesystem. Regardless, we need a single API to perform the directory listings regardless of the implementation of the actual service. So there are a number of currently available service implementations that can satisfy this need. However, we wish to hide our underlying objective to get remote directory listings in a simple manner.

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

The initial target is to focus on supercomputers (files and running simulations) as the primary data sources for DiVA components. However, at an abstract level all data sources are peers. That is to say, a data source that is a telescope or microscope is regarded in the same way as a file that resides at a fixed location. The DiVA infrastructure should make it easy to select the intermediate compute resources and computational components to rapidly supply an RDV pipeline for remote users of that fixed-location-data-source.

5.7 Please link all the above back to the functionalities described in the use case section where possible.

a) On startup, the application must gather a list of available "components". Typically this is done by consulting a local configuration file to find the locations of the binaries (or bytecode files) associated with each component as well as their names and interface definitions. For DiVA, we would like to support the discovery of remote modules as well by contacting information services on other machines or a broker that locates components on all machines in a given Virtual Organization. From the application programmer's point of view, they want to be presented with a searchable database of components (regardless of location) that can be queried and sorted based on criteria such as "name," "location," interface definition, etc... Organization as an Relational Database or LDAP directory or even a flat-file is unimportant. The API should be able to hide these details as a query for components that satisfy the search criteria is presented.

b) The user will then select components to construct an abstract workflow that implements the visualization pipeline they desire. By abstract workflow, I mean that the components simply identify the algorithm to be performed without regard to the location of an implementation for that algorithm. So the workflow is defined without an explicit mapping to a particular set of resources. In this case, we will define an arbitrary workflow involving a
- Data "Reader" that is connected to ..
- an "IsoSurfacer" that is connected to...
- a "Renderer" that is connected to...
- a display window. Please note that the "Reader" could easily be replaced by a connection to a running simulation code (eg. a remote-steering scenario), which would make component of the workflow "concrete". A mixture of "concrete" and "abstract" components should be supported, but the most general case is a completely abstract pipeline.

c) File Selection: The user browses for files to read using the GUI for the "Reader" component. The "reader" has not yet been instantiated, so the browser is actually doing remote reads of directories or is consulting a replica catalog. In the first scenario, the file browser widget associated with the abstract "reader" would present the user with a list of available machines where that "abstract reader" can be instantiated (the location of this is

determined by the service described in step A). Selecting a machine will provide the user with a remote directory listing for that machine. When the user selects a file, the remote reader is instantiated on the machine where the file has been selected. This is the method employed by the LLNL VisIT remote visualization tool.

An alternative and more elegant approach is to use a replica catalog. In this case, the user is presented with different "catalogs" to choose from. The catalogs are abstract directory structures that can span many machines and can manage files that are replicated across different filesystems as a unified directory. In this scenario, when the user selects a file to be read, the application can choose from number of resources rather than just a single machine when it comes time to instantiate the reader.

The important point here is that we need an API that will make it easier to browse and select files in a distributed environment. Currently this task can constitute a rather complicated application on its own. It should be made nearly as simple in practice as the API's we use to get file listings for file-selection-widgets that browse local directories.

* In addition, there needs to be metadata services associated with each remotely located view of a file. That is, we need to know more than simply the size of the file and its name on remote resources. We need to be able to expose information about the contents of the file (number of datasets stored in the file, the simulation that was used to create it, the size of each dataset or organization of the datasets in the file). Self-describing file formats like HDF and NetCDF allow one to quickly retrieve such metadata for open files, but such data needs to be stored in more of a database-like organization so that it can be searched and retrieved more rapidly. It is simply impractical to open all of the files stored in a tape archive in order to gather the necessary metadata to perform a search that locates a particular set of data files created by a Cactus simulation on a particular date.
* Very little attention has been paid to managing file
* permissions in
  distributed environments. While I have a uniform means to read data files, I do not have a uniform way to manage the permissions of these files to allow individual or group access. There needs to be a common API for setting and

checking these permissions. It would be desirable if this API support ACL syntax even though ACLs are not supported uniformly across all filesystems.

d) Mapping: In this step, the "abstract workflow" is converted into a "concrete workflow" through performance prediction followed by resource selection. This step is entirely hidden from the user, but is absolutely critical for proper operation. Once the reader has been instantiated (or the set of resources on which the reader *can* be instantiated is narrowed via the replica catalog), the system has enough information to begin instantiating the rest of the pipeline. However, it must do so in a manner that maximizes the pipeline performance depending on the user's current performance objective. It would otherwise be impractical for the user to manually select resources to meet these objectives. Here are some sample objectives (eventually the system should be able to automatically switch between which objective function it is optimizing)

- fastest response time when selecting a new dataset (eg. shuttling through timesteps)
- fastest response time when changing a parameter on a filter (eg. changing the isosurface level)
- fastest response time when redrawing the geometric model (eg. local OpenGL vs. parallel offscreen rendering at a remote host) Optimizing this objective function requires a means to predict the performance of each component on a given machine with current operating conditions. The input parameters for the model can be "machine characteristics" and "data sizes" (there are more potential parameters, but this is a start). The output is the prediction for the performance of that component on a given machine. In addition, the Network Weather Service or other performance modeling methods can be employed to predict the time taken to transfer data between each of the distributed hosts involved in the workflow.

Once you have a performance model that enables you to predict the performance of individual components and network links (even if the model contains considerable uncertainties), you can employ a Dijkstra weighted shortest path algorithm to find the set of resources that provide fastest response time in Order(N logN) complexity (where N is the number of resources available). That is to say, the mapping can be done in a fraction of a second on most machines and is highly

scalable.

We need the API to provide a simplified interface to this decision making system for mapping the abstract workflow to a fixed/concrete set of resources. Existing brokers do not understand the dependencies between the different components that comprise a distributed workflow, and therefore will not be able to map the workflow in a manner that will offer reasonable performance.

e) Launching the components: After the mapping of the abstract pipeline to concrete resources occurs, the application manager (or app controller) needs to be able to launch all of the components of the pipeline using the best available means at its disposal. While RSLs offer a great degree of flexibility in specifying the job launching criteria, the application programmer *really* would like to call "Launch("host",<the same string you would type on the commandline to manually launch the component>); It should be able to hide the underlying mechanics of the secure launching mechanism.

f) Connecting the Components Together: Once the components are "launched," the components that reside on different machines need to be able to communicate with one another via network sockets. This leads to both security and performance considerations (discussed below).

Secure Sockets Abstraction: The state-of-the-art use of sockets for remote visualization depends on security practices that are largely debunked as "unsafe" by our security professionals. Most distributed visualization application developers presume that if components were launched securely, the sockets that connect these components together are also secure (who could guess the port number and startup time... its security through obscurity). This is clearly not safe, but the level of effort necessary to move to simple authenticated sockets is considerable (it is not practical to expect vis people to do this). At minimum, a new TCP socket must exchange an "shared secret" in order to make sure the connection is "authentic". If there was a single call to "OpenAuthenticatedSocket(host,port)", then that would be very useful.

Likewise, an "OpenEncryptedSocket()" would be useful in other circumstances. For instance, SGI's VizServer does not

encrypt the socket used to convey your login & passwd to connect to the server (its as bad as telnet). File directory listings probably should be encrypted. This should just be targeted at a single TCP socket. We can use GridFTP for performance, but we really need to get the control sockets protected. As you know, GridFTP is terrible for metadata and control socket applications.

g) Communications Performance: The components in a vis pipeline are extremely sensitive to QOS. This requires much stronger QoS guarantees than can be delivered by the best-effort packet-switched networks. Recently, there have been a number of major scientific network backbones who support application-negotiated circuits (either PVCs or actual end-to-end lambdas). The current protocols for this negotiation are exceedingly complex.

The visualization application needs to communicate its resource requirements to the network infrastructure. When possible, the network infrastructure can provide a "lease" for a dedicated circuit to a component so that it can switch to fixed-datarate protocols or otherwise provide stronger guarantees of interaction. In the case that the vis application cannot obtain a dedicated circuit, it can still rely on TCP/GridFTP over best-effort networks with some degradation in performance. This negotiation can be simplified through an API that allows the application designer to declare the resource requirements and to install "callbacks" that inform the application of resource availability changes.

h) Runtime Application management: Once we have a running workflow application, it is important to preserve the illusion that the entire application is running in one place (eg. make it appear to function just the same as your old Amira or AVS pipeline running on your desktop machine). This requires that the components be monitored to detect transient failures or "contract violations" where a resource fails to meet its performance prediction. It also requires interfaces that manage the restart of components with the same parameters on different resources if a migration is required. Unlike typical simulation applications where a full checkpoint-restart is required, the visualization workflows can have most of their state restored from the upstream component in the pipeline. This makes migration less data intensive, but no less complicated in terms of

managing the state information.

Monitoring and Fault Tolerance: There must be a simple API that can report wallclock performance for each invokation of a component. The API can report available timers as a list of names and then respond to requests for the performance metrics associated with each of the named timers that are offered by the application. This is necessary as a heartbeat that determines whether a component is still "alive" and for detection of partial failure (sick machines) or outright contract violations (a failure of the performance model or machine).

State Migration/Redeployment: If it is determined that the component has died or needs to be moved in order to meet a different set of performance objectives, there must be a means to report the part of the component's state that must be preserved for it to be restarted in the same state and what state data must be acquired from upstream components in order to complete the components restart in the same state. This kind of state registration is common to many user-space checkpoint-restart APIs that are being discussed in the GGF CPR working group. If we could settle on one API (one that has a special notion of data that can be restored from upstream objects rather than being explicitly stored in the checkpoint), then we can reuse the same API's to enable fault-resiliance for distributed workflows and remote vis applications.

5.8 How often is your application used on the grid or grid-like systems?

[ ] Exclusively
[ ] Often (say 50-50)
[X] Occasionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

Our objective is to create a system that supports distributed visualization applications that work as well on the Grid as they do in a standalone/desktop context.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

Visualization application developers typically use C/C++ to implement applications. However, Python, TCL, and Java are increasingly popular for assembling applications that use components that are written in native C or C++ (for instance VTK supports wrapping in any of those bindings).

The services should be accessible using C and C++ bindings as a baseline. Those bindings can typically be wrapped so they are accessible in any of the above scripting languages.

The services/capabilities defined in DiVA do not require specification of a particular GUI toolkit (eg QT vs. FLTK) or other library/software dependencies. It is meant to be a toolkit that enables visualization applicationl developers to assemble distributed applications much easier without forcing them to abandon the algorithms, toolkits, or GUIs they employ to get their work done.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

The user specifies an "abstract workflow" that identifies the set of components that must be run and the dependencies between them. From a user perspective, the location of the components is immaterial -- there should be an illusion that everything is running locally.

When a user selects a datafile to read or a simulation to connect to, that will pin-down the data source to a particular location based on the user's selection. In a converse situation, a particular software component may be available on only one machine (eg. a software interface to a unique piece of hardware). However, the application does not need to expose the user to locality information unless absolutely necessary.

The mapping of the "abstract workflow" to resources is where the resources are finally selected. This involves constructing a performance model for the the components in the workflow and then finding the optimal assignment of the workflow to available resources based on this performance model (an O(n log(n)) complexity algorithm).

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

The resources used starts with all machines that are in the VO and is progressively reduced by the following objective criteria

- 1) availability: Is the machine available? Is the component specified in the abstract workflow available on that machine?
- 2) compatibility with requirements: Once the source data sizes are known, then the resource requirements for much of the pipeline can be computed. This criteria can also be used to cull the list of potential machines.
- 3) mapping to optimize performance: Using a simple performance model for each component and intervening network links, a weighted shortest-path algorithm can be used to select a set of resources that will provide the fastest response time for a given objective (eg. fastest rendering rate or fastest update rate when new data is read)

7.3 Are the resource requirements dynamic or static?

The resource requirements are absolutely dynamic. Even a small change to one parameter of one stage of the visualization pipeline can lead to dramtic changes in resource requirements. There are rare cases where the requirements stay relatively constant (provided the Grid offers resources with consistent performance... an unlikely event at that), but static resource requirements are probably the rarer case. Please refer to http://vis.lbl.gov/Publications/2002/VisViewpoints-May2003-IEEECameraReady.pdf

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

Certainly all of the above. Executable codes and computer hardware require protection via proper authorization. We do not assume that users require 'anonymous access' to any resource.

The ACE-RG has written a document that treats this issue in extreme detail.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

SSH is the primary method used for launching distributed pipeline components. We would like to continue to use it as a fallback where GSI is not available.

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

Please refer to everything that Stephen Pickles wrote to answer this question in his RealityGrid use case. They are completely relevant to DiVA.

The primary problem we faced is that most visualization programmers do not have any understanding of security issues. If we can, at minimum, simplify the API calls required to establish an authenticated socket and even make it simple to establish an encrypted socket (even for low-bandwidth), that would fullfill that requirement. Distributed visualization applications need to meet the bare minimum security standards (something that most currently do *not* do).

In most of the scenarios we have constructed, the security needs primarily involve supporting proper authentication in order to connect components together. Authorization is primarily required if a component is actually a persistent

service that is not running as the same UID as the user who constructed the workflow. We try to set up the bulk of DiVA valid use cases so that the components are running as the user's UID in order to sidestep such cases as much as possible.

The socket traffic for GUI events and component execution does not typically need to be encrypted, except in rare cases such as passing privileged information (encryption keys or passwords).

Global support for group authorization is almost a fantasy at this point.

Again, please refer to the ACE-RG security document.

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

The current GSI API is enormously complex because of all of the use-cases it supports. In our case, we need authentication for very restricted purposes (establishing a simple TCP socket for GUI events for instance). We do not require as much functionality. Less functionality should lead to a much more compact API.

It is important that the security API interact nicely with SSH. When GSI/PKI certs are not available, ssh works quite well. While the single-sign-on abilities of GSI are extremely important, many have found that ssh combined with ssh-auth can do a reasonably good job of emulating the same ability if used as a last resort. This should *not* be ignored (this consideration should be taken into account when designing these APIs).

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

Distributed workflows for visualization and data analysis applications are typically I/O intensive. In particular, visualization applications require consistent performance in

order to deliver effective interactivity. (so they are both I/O intensive, and they require consistency as well in some cases). So, support for dynamic provisioning of network resources (QoS and the like) is essential.

Performance prediction is also an essential feature for overall application scalability. This requires a very scalable means of collecting the parameters for a given component's performance model. The ability to search the database for component attributes may be an important performance factor in supporting this application.

I quote this from Stephen Pickles: "Arguably our most important scalability concern relates to the human effort involved in porting and deploying applications (and the middleware stacks they depend on) to an increasing variety of Grid resources."

This is the *biggest* and most important scalability objective for the DiVA project.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

a) Accurate performance prediction for components is essential for placing the components on grid resources so as to provide good responsiveness for the overall application. This includes both models of the components on a given machine/resource and the network infrastructure that connects them together (this can be alleviated somewhat if the app can negotiate a dedicated circuit). It is entirely possible that the performance prediction will map everything onto the user's desktop machine to get the best performance -- that is perfectly acceptable. The point is that the user will be hard pressed to solve this performance problem themselves every time they launch an application.

Performance prediction requires only marginally accurate performance models to get good predictions, but it is unclear how to serve up those models in a timely manner.

b) Data transfer rates between distributed resources are

essential. API's that simplify QoS negotiations are important

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

- gsissh, globus-url-copy, globus-job-run, GSI, XML, Chromium, and potentially gSOAP.
- We intend to use the GridLab GAT for job launching, info services, and file/replica management.
- We also find SSL/TLS essential even though it is not Grid software per-se.
- A long term goal is to interface to SRB for better integration with tertiary storage management systems.

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

This is completely off-the-cuff...
(do not take this too seriously)

```
// support for simple authenticated TCP sockets
SAGA_OpenAuthentTCPServer (securityhandle, portnumber);
SAGA_OpenAuthentTCPSocket (securityhandle, host, port,
                           passwdcallback);
     // where passwdcallback gets a passwd
     // if needed in order to  support
     // SSL/TLS security models.

// support for simple encrypted TCP sockets
SAGA_OpenEncryptedTCPServer (securityhandle, portnumber);
SAGA_OpenEncryptedTCPClient (securityhandle, portnumber,
                             passwdcallback);

// simple job launching
SAGA_Run (securityhandle,
```

```
            "string of the same damned thing I'd type on
             the commandline to launch my job if I logged into
             the system to do it");

    // Remote Directories: select a machine or a  replica catalog
    // to act as your root directory
    rmt_dir_context = SAGA_SetRoot (securityhandle, "URL, path,
    or replica catalog name to act as root directory");

    // get the number of file names so you can preallocate a list
    // to hold them
    int SAGA_GetNumFileNames (rmt_dir_context);

    // gets list of all names in the directory...  truncates
    // filenames that are longer than allocated storage
    SAGA_GetFileNames (rmt_dir_context, <preallocated array>,
                       maxnamelen);

    // get all  of the metadata about the file like filesize, and
    // filetype etc...
    SAGA_GetFileStat (rmt_dir_context, <stat array>);

    SAGA_ChangeDir (rmt_dir_context, <some rel/abs path>);

    This could go on for hours...
```

13. References:

List references for further reading.

DiVA Homepage http://vis.lbl.gov/Research/DiVA/index.html

J. Shalf and W. Bethel: "How the Grid Will Affect the Architecture of Future Visualization Systems," IEEE Computer Graphics and Applications, Volume 23, Number 2, March/April 2003, pp 6-9.
http://vis.lbl.gov/Publications/2002/ \
VisViewpoints-May2003-IEEECameraReady.pdf

M.J. Bennett, G. Bell, J. Shalf: "Transport Requirements for High Performance Network Applications that are NOT FTP" Submitted to the Second International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2004), Chicago Il. 2004.

http://vis.lbl.gov/Publications/2003/PFLDGridAppReqs.pdf

G. Allen, E. Siedel, and J. Shalf: "Scientific Computing on the Grid," Byte Magazine, Spring 2002.
http://vis.lbl.gov/Publications/2002/LBNL-51039-Byte2002.pdf

UC 4: Bulk job submission

Name of use case: Bulk job submission

Contact: Hrabri Rajic <hrabri.rajic@intel.com>

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[x]
Academic	[ ]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[ ]
Please specify:	Not tied to any area

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[2]
Remote steering	[ ]
Visualization	[ ]
Security	[ ]
Resource discovery	[ ]
Resource scheduling	[ ]
Workflow	[ ]
Data movement	[ ]
High Throughput Computing	[ ]

High Performance Computing	[5]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[ ]
Application developer	[ ]
System administrator	[ ]
Service developer	[ ]
Computer science researcher	[x]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case.
Background to the project is another alternative.
(E.g. 100 words).

Very often in certain industries, like in financial, bioinformatics, computational chemistry, or optimization there is a need to submit a set of parametric jobs which differ in just few parameters. Few DRM systems have a direct support for this kind of calculations via an array job mechanism. Abstracted layers on top of a DRM or Grid systems have additional ways of optimizing the execution of these parametric jobs.

2.2 Is there a URL with more information about the project ?

...

3. Use Case to Motivate Functionality Within a Simple API:

Few years ago we were involved in a large database calculations of different molecular characteristics for drug design. It is basically a parametric submission of a large number of jobs doing the same set of calculations.

4. Customers:
---

chemists, pharmacologists

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

There were 120 remote computational nodes and few client nodes with good graphics support.

5.2 Are these resources geographically distributed?

Yes, absolutely. One of the runs involved several sites across USA.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

130 tasks at one time any there was a need for doing more.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

The study was doen in collaboration with the software maker, so we had no licence needs. The application was done in Fortran and C. Part of the distributed solution was implemented in Java. There was also a high troughput database requirement. Typical use site would need licences on the remote nodes and graphical viewer licences on the client side.

5.5 What information sources do you require, e.g. certificate authorities, or registries.

...

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

No.

5.7 Please link all the above back to the functionalities described in the use case section where possible.

...

5.8 How often is your application used on the grid or grid-like systems?

[ ] Exclusively
[x] Often (say 50-50)
[ ] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

...

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent

in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

The runs were performed by the developers. The administrators had to set up the Grid as specified by the devlopers.

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

It was mostly "don't care" selection of the compute nodes. There was a requiremnt on the scheduler node to support large number of tasks at the time, necessitating fine tuning of the OS.

7.3 Are the resource requirements dynamic or static?

Doesn't matter.

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

In a non-demo like situation there is a concern about the nature of the calculations and the results.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

No.

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

Authentication mostly.

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

Simple API and deployment.

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

The biggest if not the only bottleneck was the scheduler and then number of processes OS limitations.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

Access to a database was closely monitored. Large number of independent tasks was an issue on the scheduler node.

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

...

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

There is a need for efficient submisison of bulk jobs.

13. References:

List references for further reading.

...

UC 5: Application Migration

Name of use case: Application Migration

Contact: Andre Merzky <merzky@cs.vu.nl>

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[x]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[ ]

Please specify: astrophisics, but the use case is generic

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[3]
Visualization	[1]
Security	[1]
Resource discovery	[5]
Resource scheduling	[5]
Workflow	[3]
Data movement	[5]
High Throughput Computing	[ ]

```
    High Performance Computing   [1]
    Other                        [ ]
        Please specify:          ..............................
```

1.4 Are you an:

```
    Application user             [ ]
    Application developer        [ ]
    System administrator         [ ]
    Service developer            [ ]
    Computer science researcher  [ ]
    Other                        [ ]
        Please specify:   Middleware Developer (higher levels)
```

2. Introduction:

2.1 Provide a paragraph introduction to your use case.
Background to the project is another alternative.
(E.g. 100 words).

One of the major scenarios targeted by the GridLab project is the ability to migrate a running application in a VO. The migration process may get triggered by various means:

- running out of time on the original resource
- a more powerful resource comes available
- a resource with more memory or local disk space is needed
- user prefers a different resource and triggers migration
- migration as part of a larger work flow scenario

The migrations includes following well defined steps:

- trigger migration
- discover new resource
- perform application level checkpointing
- move checkpoint data to new resource
- schedule application on new resource
- continue computation (and discontinue old job)

Several of these operations need to be done on application level - the use cases specifically describes those operations in respect to an Grid API.

2.2 Is there a URL with more information about the project ?

http://www.gridlab.org/

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs.
E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality,
please provide separate descriptions for each component,
bullet points of 50 words per functionality are acceptable.

Following the list from 2.1:

- trigger migration

  If the application triggers the migration process itself,
  it needs means to communicate with the resource management
  system it got started with, or with any other one which
  knows about its execution environment requirements (exe,
  input files, output files... -> job description). The
  request basically is:

```
rms  = Grid.getResourceManagementSystem ();
self = rms.getMyJobDescription ();
// perform checkpoint
// save state
```

  If the application migration gets triggered from outside
  the application, the application needs to have means to
  getified about this - it needs to know when to perform
  checkpointing and to shut down. There are many ways to do
  that - application steering like mechanisms seem the most
  convenient ones:

```
sub mycallback (userdata) {
  // perform checkpoint
  // save state
```

```
}
result = Grid.announceCheckpointCallback (mycallback,
                                          userdata);
```

- discover new resource

  If that operation is not performed by the resource management system itself, the application needs to discover new resources where itself can run on. It needs to provide its own job description.

```
host = GriResourceManager.discoverNewHost (self);
```

- perform application level checkpointing

  The checkpointing process itslef does not need Grid support per se, but the application needs to be able to announce the location of it's checkpoint files. These could be put into a replica catalog, or onto a global file system - but the resource manager needs to know about them, in order to make them available on the new resource:

```
app.checkpoint (filename);
grid.replicaCatalog.addFile (replicaname, filename);
rms.announceCheckpointFile  (replicaname);
```

- move checkpoint data to new resource

  If that operation is not performed by the resource management system itself, the application needs to be able to mograte its checkpoint files to the new resource:

```
grid.copyFiles (filename,    host);
  or
grid.replicate (replicaname, host);
```

- schedule application on new resource

  If that operation is not performed by the resource management system itself, the application needs to be able to start a copy of itself on the remore resource:

```
        copy = GriResourceManager.runJobOnHost (self, host);
```

- continue computation (and discontinue old job)

  Both are straight forward.

4. Customers:

Describe customers of this use case and their needs. In particular, where and how the use case occurs "in nature" and for whom it occurs. E.g. max 40 words

The cusomers of the use case are scientific communities with jobs
a) running for a very long time (~weeks)
b) with varying comuting demands (peeks requiring more powerful resource, or more disk space)
c) which are part of larger dynamic systems

Grand Challenge Simulations are specific target applications for that use case.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

- compute resources
- data storage systems
- resoure management systems
- data replication/movement systems
- remote steering or monitoring systems

5.2 Are these resources geographically distributed?

potentially yes.

5.3 How many resources are involved in the use case? E.g. how

many remote tasks are executing at the same time?

minimum: 2, maximum: unlimited, only one compute resource at the same time.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

Application:	C/Fortran code, open source http://www.cactuscode.org
API:	C api binding to Grid Services, open source http://www.gridlab.org/gat/
Services:	C and Java Services, open source, mostly basing on globus http://www.gridlab.org/

5.5 What information sources do you require, e.g. certificate authorities, or registries.

Resource Discovery and state preservation (repolica systems or similar) are the main requirements to information management.

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

No.

5.7 Please link all the above back to the functionalities described in the use case section where possible.

...

5.8 How often is your application used on the grid or grid-like systems?

[ ] Exclusively
[ ] Often (say 50-50)
[x] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

The application is actually used in Grids, but does not make full use of Grid capabilities (as the one described here).

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

Users work mostly on shells, portals are uder development. Programmers work on open source solutions, unix only, C, C++, Fortran.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

Compute Resources are selected manually or automatically (job description by users).

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

OS, Architecture, Memory, disk space, runtime (when, how long)

7.3 Are the resource requirements dynamic or static?

Vary from run to run, but mostly static, sometimes dynamic.
In the future more dynamic.

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

Data should get only accessed by owner or group. Resources are not to be compromised of course. --> standard academic security requirements.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

GSI for all communication and resource access.

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

authentication, authorisation, basic data protection

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

simple deployment

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

The scenario is not bound by scalability (the application of course is).

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

Full time to migrate to a better must result in a benefit if compared to having the computation simply continue on the old resource. However, on ocasions where simply continuation is not possible, performance penalties are acceptable.

In general: performance requirements depend on specific application/simulation.

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

- globus based services from the GridLab project
- Grid Application Toolkit from the GridLab project

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

An example of a migtration in GAT is included in the GAT release.

13. References:

List references for further reading.

http://www.gridlab.org/gat/

UC 6: DIRAC

Name of use case: DIRAC: Distibuted infrastructure with
remote agent Control

Contact: Garonne vincent <garonne@cppm.in2p3.fr>
CPPM, 163 avenue de Luminy, case 902,
13288 Marseille cedex 09.

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[X]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[ ]
Please specify:	.....

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[2]
Remote steering	[?]
Visualization	[?]
Security	[3]
Resource discovery	[5]
Resource scheduling	[5]

Workflow	[2]
Data movement	[3]
High Throughput Computing	[5]
High Performance Computing	[1]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[ ]
Application developer	[ ]
System administrator	[ ]
Service developer	[X]
Computer science researcher	[X]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case . Background to the project is another alternative. (E.g. 100 words).

The LHCb experiment is one of four particle physics experiments currently in development at CERN, the European Particle Physics Laboratory. Once operational, the LHCb detector will produce data at a rate of 4 Gb/s, representing observations of collisions of sub-atomic particles. This massive quantity of data (3 peta per year) then needs to be distributed around the world for the 500 physicists at 100 sites to be able to carry out analysis. Even before this analysis of real physics data can begin a large number of monte- carlo parameter sweep simulations are required to verify aspects of the detector design, algorithms, and theory. While the four Large Hadron Collider (LHC) experiments have coordinated with CERN to produce the LHC Computing Grid (LCG) \cite{LCG}, there is still a requirement within LHCb to manage and track computing tasks, provide a set of common services specific to LHCb, and to be able to make use of hardware and software resources not incorporated into LCG. DIRAC architecture (Distributed Infrastructure with Remote Agent Control) which has been developed to meet

these requirements and provide a generic, robust grid computing environment.

2.2 Is there a URL with more information about the project ?

DIRAC: http://dirac.cern.ch LHCb experiment: http://lhcb.web.cern.ch/lhcb/

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs. E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

MC simulation/Production: User: a production manager Job submission: bunch of jobs Mode of submission: Organized jobs are planned in advance and perform a homogenous set of tasks. Goal: produce a big amount of data on a period, CPU bound , parameter sweep application Requirements: High Troughput computing (Group and end user) Analysis:

<>User: Many users Requirements: response time (as opposed to total throughput) Mode of submission: Chaotic jobs are submitted by many users acting more or less independently, and encompass a wide variety of tasks. The input is typically a selection/analysis algorithm to be applied to a very large dataset(i/o bound). Users can submit jobs of this kind at any time, simultaneously asking for the common input datasets.

4. Customers:

Describe customers of this use case and their needs. In particular, where and how the use case occurs "in nature" and for whom it occurs. E.g. max 40 words

...

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

Resources: Standart cluster accessible trough a batch system, grid system (e.g LCG) Software : linux and python interpreter

5.2 Are these resources geographically distributed?

Yes. spread over the world

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

60 computing sites representating something like 5000 nodes

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

Dirac use no license and it's a free project. DIRAC can be decomposed into four sections: Services, Agents, Resources, and User Interface. The core of the system is a set of independent, stateless, distributed services. The services are meant to be administered centrally and deployed on a set of high availability machines. Resources refer to the distributed storage and computing resources available at remote sites, beyond the control of central administration. Access to these resources is abstracted via a common interface. Each computing resource is managed autonomously by an Agent, which is configured with details of the site and site usage policy by a local administrator. The Agent runs on the remote site, and manages the resources there, job monitoring, and job submission. An Agent can be deployed on a gatekkeeper of a large cluster or just on a single worker

node of the LCG grid. PBS, LSF, BQS, Condor, LCG, Globus can be used as the DIRAC computing resources.

The User Interface allows access to the Central Services, for control, retrieval, and monitoring of jobs and files. It consists of a small set of distributed stateless Core Services, which are centrally managed, and Agents which are managed by each computing site.

The current implementation has been written largely in Python, due to the rich set of library modules available, its object oriented nature, and the ability. The workload management system (WMS) components use XML-RPC and instant messaging Jabber protocols for communication which increases the overall reliability of the system. The jobs handled by the WMS are described using Classad library which facilitates the interoperability with other grids. to rapidly prototype design ideas.

For all Service and Job state persistence, a MySQL database is used.

5.5 What information sources do you require, e.g. certificate authorities, or registries.

...

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

Usually some mass storage (HPSS) are used in several sites called "Tiers 0".

5.7 How often is your application used on the grid or grid-like systems?

[X] Exclusively
[ ] Often (say 50-50)
[ ] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

keeping the implementation in Python, Clients and Agents only require a Python interpreter for installation. These components which are distributed to the users and computing centers are also very small --- less than one megabyte compressed --- which further facilitates a rapid installation or update of the software. Software required for jobs is installed in a paratrooper approach, which is to say that each job installs all software it requires, if it is not already available. This software is cached and made available for future jobs run by the same Agent.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

When the user submit their jobs they specify their requirement (data need, CPU required, and so on), then the system tries to find the resource which match well theses requirements.

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

DIRAC uses a 'pull' paradigm where Agents request tasks whenever they detect their local resources are available. The collaborating central Services allow new components to be plugged in easily. These Services can perform functions such as scheduling optimization, task prioritization, job splitting and merging, to name a few.

7.3 Are the resource requirements dynamic or static?

The resource requirement are both of them static , e.g. for the cpu requirement and dynamic for the data requirement in regard with where the data are avalaible.

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

acces to the ressources must be secure and the tracability must be guarantee. The system adminitrator woudl like to know who has done something in the grid. This information is also to do the accounting of the system

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

No.

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

, authorisation,

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

are looking now the integration with web services technologies based on apache3, mod python, grid site which should provides the gsi security infrastructure.

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

system needed to be quickly and easily deployed across fourty or more sites, with little effort from local site administrators, either for installation or maintenance.

Similarly, users needed to be able to access the system from anywhere, with a minimal set of tools. The need to support intense computational load also required the system to be responsive and scalable, supporting over 10,000 simultaneous executing jobs, and 100,000 queued jobs.The final system must be able to deal with 30000 cpus and to manipulate store and analyse something like 2 peta per year.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

All Client/Service and Agent/Service communication is done via XML-RPC calls, which are lightweight and fast. Furthermore, instantiating and exposing the API of a Service as a multi-threaded XML-RPC server in Python is extremely straight forward and robust. We are lookink now to use apache+mod python with a security infrastructure that have a real web services and oriented services architecture.

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

?

13. References:

Garonne, V. and Stokes-Rees, I. and Tsaregorodsev, A.: "DIRAC: A Scalable Lightweight Architecture for High Throughput Computing" in Grid 2004, 5th IEEE/ACM

International Workshop on Grid Computing

UC 7: GriPPS - Grid Protein Pattern

Name of use case: GriPPS - Grid Protein Pattern Scanning

Contact: <Christophe.Blanchet@ibcp.fr>

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[X]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[ ]
Please specify:	.....

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[5]
Remote steering	[1]
Visualization	[0]
Security	[3]
Resource discovery	[1]
Resource scheduling	[4]
Workflow	[3]
Data movement	[5]

High Throughput Computing	[3]
High Performance Computing	[4]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[ ]
Application developer	[X]
System administrator	[ ]
Service developer	[ ]
Computer science researcher	[ ]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case . Background to the project is another alternative. (E.g. 100 words).

Genomics acquiring programs such as full genomes sequencing projects are producing greater amounts of data. The analysis of these raw biological data require very large computing resources. Functional sites and signatures of protein are very useful for analyzing these data or for correlating different kind of existing biological data. These methods are applied, for example for identification and characterization of the potential functions of new sequenced proteins, clusterization in protein family of the sequences contained in international databanks, and so on.

The Grid Protein Pattern Scanning-GriPPS project (granted by the french programACI GRID 2002) aims to develop and adapt these bioinformatic algorithms so that they can exploit the underlying grid infrastructure. Models of those algorithms will be devised to be able to foresee their behavior on a grid platform and proposals will be written to adapt other bioinformatic algorithms to the grid. Within this context, we propose to study such algorithms to identify the constraints related to the biological applications and to determine their granularity and the possible parallelization schemes that can

be applied to them.

2.2 Is there a URL with more information about the project ?

http://gripps.ibcp.fr

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs.
E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

Get fast access to sequence databank repository from worker node Start scanning of one protein pattern Download result file: patterns identified Download sub-base of sequences Repeat the job with an other protein pattern on the obtained subset of sequences

4. Customers:

Describe customers of this use case and their needs. In particular, where and how the use case occurs "in nature" and for whom it occurs. E.g. max 40 words

Used by biologist for identifying proteins with particular functionnality. Need to be able to put their own protein patterns against world-wide reference databases .

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

Storage:
- sequence database (SWISSPROT, TrEMBL, ..., user ones)

- pattern database (PROSITE, pFAM, ..., user ones) Software: PattInProt

5.2 Are these resources geographically distributed?

Yes, each database are maintained by different laboratories.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

They could be several depending on the pattern bank and the proetin sequence bank to analyze.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

Databank: open for acadmemic community
Software: PattInProt, closed source, C/C++

5.5 What information sources do you require, e.g. certificate authorities, or registries.

Certificate authorities to satisfy data privacy or access for some of them.

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

No

5.7 How often is your application used on the grid or grid-like systems?

[ ] Exclusively
[ ] Often (say 50-50)
[X] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

Available through web portal to the bioinformatic community.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

...

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

...

7.3 Are the resource requirements dynamic or static?

...

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

Data could be sensible if they are from patient or pharmaceutical/agronomic researches.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

no

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

authentication, authz, data protection, monitoring and log (transfer and access)

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

...

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

could be short: need a low payload time. these sort jobs may need to be ran on databank of several gigabytes

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which

grid technologies are you using or considering?

middleware

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

local access IO (on worker node) to remote file/databank (on storage element) write subset of the databank, selected by the pattern scan

13. References:

List references for further reading.

...

UC 8: HSEP

Name of use case: HSEP

Contact: Emmanuel Jeannot
LORIA
Campus Scientifique - BP 239
54506 VANDOEUVRE-les-NANCY CEDEX
France

Authors: Gerald Monard
Emmanuel Jeannot

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[X]
Please specify:	(Quantum) Chemistry

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[3]
Remote steering	[ ]

Visualization	[ ]
Security	[ ]
Resource discovery	[ ]
Resource scheduling	[ ]
Workflow	[1]
Data movement	[2]
High Throughput Computing	[4]
High Performance Computing	[5]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[ ]
Application developer	[ ]
System administrator	[ ]
Service developer	[ ]
Computer science researcher	[X]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case . Background to the project is another alternative. (E.g. 100 words).

New developments in the field of theoretical chemistry require the computation of numerous Molecular Potential Energy Surfaces (PESs) to generate adequate quantum force field parameters. Because workstations alone cannot fulfill the requirements of these modern chemical advancesy, we have tackled this problem using several up-to-date computer science technology such as a GridRPC middleware (DIET), molecular databases, script interfacing...

2.2 Is there a URL with more information about the project ?

...

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs.
E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

We have decided to use the DIET\footnote{\url{http://graal.ens-lyon.fr/DIET}} (D\istributed Interactive Engineering Toolbox) middleware to port our application on our grid. The DIET environment works under the gridRPC.

The application needs to store information about the computations that have to be performed and the results.It is required to access to the information very often and very efficiently. Therefore, we have decided to use MySQL databases for storing the information.

Since the DIET API is in the C/C++ language, some python scripts are used to interface DIET with the databases. These scripts are used to extract requests and to filter and store results in the databases. Requests and results are transferred in XML files.

On the computing side, python scripts are used to parse XML requests files and call the quantum chemistry (QC) solver with the right arguments. Several solvers can be used to compute \abinitio or semiempirical energies (\ie, quantum chemical energies, or QCEs). Each time a compute node register to the DIET environment, it tells DIET which kind of operation it is able to perform.

4. Customers:

Describe customers of this use case and their needs. In particular, where and how the use case occurs "in nature" and

for whom it occurs. E.g. max 40 words

The customer give a set iof molecule conformation and the system computes teh Potential Energy Surface of all the system. Then the system is able to fit structural chemistry parameters.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

A PC farm and a PC server linked by a network.

5.2 Are these resources geographically distributed?

Not yet, but it could as soon as we will be able to bypass firewall.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

So far we have 8 PC, but we could have many more. There is at most one task running per resources at the same time.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

We use the DIET middleware (opensource) a mysql Database. The code is in C sole script are in Python.

5.5 What information sources do you require, e.g. certificate authorities, or registries.

None

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

None

5.7 How often is your application used on the grid or grid-like systems?

[X] Exclusively
[ ] Often (say 50-50)
[ ] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

...

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

No ressources are selected by the user. Resources are Server that register to the environement.

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability

of information, size of datasets?

We use the GridRPC model but run the application under the pull mode (desktop grid model).

7.3 Are the resource requirements dynamic or static?

static

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

Resukts stire inthe data base are the added value of teh application. Only the administrator can perform critical things on it.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

none

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

none

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

none

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

Since the grain is relatively small, we need sufficient data in order to use all the ressources.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

...

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

GridRPC for a desktop grid app.

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

It should enable data management (persistence and redistribution).

13. References:

List references for further reading.

...

UC 9: Circuit Simulation

Name of use case: Circuit Simulation

Contact: Sommet R,
IRCOM CNRS UMR 6615
IUT GEII 7
Rue Jules Valles
19100 Brive la Gaillarde

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[X]
Please specify:	ELECRONICS

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[ ]
Visualization	[4]
Security	[ ]
Resource discovery	[ ]
Resource scheduling	[ ]

Workflow	[ ]
Data movement	[ ]
High Throughput Computing	[ ]
High Performance Computing	[5]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[X]
Application developer	[X]
System administrator	[ ]
Service developer	[ ]
Computer science researcher	[ ]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case.
Background to the project is another alternative.
(E.g. 100 words).

We have developed our application in SCILAB because it is a free and an open environment with numerous primitives (linear algebra, graphical subroutines...), and because it is a very interesting environment for prototyping rapidely new applications in our domain. Our needs relies on the use of the ASP possibilities from SCILAB.

2.2 Is there a URL with more information about the project ?

No

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs.
E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

Analysis of the topology of the circuit made of transistors, linear elements, generators.

For I=1 to number of transistors Call "The Physics based simulator" in order to fill in the Jacobian matrix of the problem. Assembly the global matrix Add the contribution of linear elements and generators Solve the sparse problem.

4. Customers:

Describe customers of this use case and their needs. In particular, where and how the use case occurs "in nature" and for whom it occurs. E.g. max 40 words

The use of tools in a friendly environment like SCILAB

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

PC under Linux, HP workstation

5.2 Are these resources geographically distributed?

No, but may be in the future

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

8 or 16 is a good value, but the same program is called (Physics based Simulators)

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

C, C++, SCILAB, Librairies : TAUCS, UMFPACKS, ARPACK, LAPACK, BLAS, SCALAPACK

5.5 What information sources do you require, e.g. certificate authorities, or registries.

I don't understand this question

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

No

5.7 How often is your application used on the grid or grid-like systems?

[ ] Exclusively
[ ] Often (say 50-50)
[x] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

SCILAB

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

It must be transparent for the users or the developers

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

By CPU power and memory available, depending of the size of the problem

7.3 Are the resource requirements dynamic or static?

dynamic

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

Maybe executable code

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

No

8.3 What are your security needs: authentication, authorisation, message protection, data protection,

anonymisation, audit trail, or others?

I don't know for the moment but maybe authorization

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

No response

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

Compute ressources

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

Access to a very fast sparse solver for very large system

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

ASP (client/agent/server) with DIET

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

For sparse Solver : possibility to select a preconditioner (for symmetric or unsymmetric matrix), the matrix format(matrix market or Harwell Boeing)

For the Physics based server : input is a text file which allow to fill the matrix of the problem

13. References:

List references for further reading.

Not yet

UC 10: KMC - Kinetic Monte carlo Model

Name of use case: KMC - Kinetic Monte carlo Model

Contact: <philippe@lifc.univ-fcomte.fr>

Authors: Christophe Ramseyer
Universit Besancon - France

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[X]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[ ]
Please specify:	.....

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[ ]
Visualization	[3]
Security	[ ]
Resource discovery	[ ]
Resource scheduling	[ ]

Workflow	[ ]
Data movement	[ ]
High Throughput Computing	[ ]
High Performance Computing	[5]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[ ] Application developer
[ ] System administrator	[ ] Service developer
[ ] Computer science researcher	[X] Other
[ ] Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case . Background to the project is another alternative. (E.g. 100 words).

This application simulate growth of atomic species on surface. This physical experiment is mainly governed by for random processes : deposition, diffusion, aggregation and desorption. Its main application is wires and gratings for electronic transport and quantun switches. This experience is simulated by a Kinetic Monte Carlo Model which simulated the comportement of the atoms depending on the properties of the surface. Input and output rather small sized. Computing may be very long (from one day to one month). Interaction is mandatory to verify the simulation evolution.

2.2 Is there a URL with more information about the project ?

not for the moment

3. Use Case to Motivate Functionality Within a Simple API:

- Source code of KMC is not distributed for customer installation.

- job submission may be synchronous : the customer send a set of parameters and check if the result is relevant.

- job submission may be asynchronous : the customer send several sets of parameters and compare a posteriori the results at the end of the jobs executions.

4. Customers:

Customers are physicians without computer science experience. They access the KMC application to test a surface, differents parameters of presure, temperature or grating.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

- hardware : parallel machine / cluster
- software : Diet, Corba, KMCsolver
- visualisation tools : standard image tools or video players

5.2 Are these resources geographically distributed?

no

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

5.4 Describe your codes and tools:

KMC code is not distributed, just installed on three servers. It is written in fortran

5.5 What information sources do you require, e.g. certificate authorities, or registries.

no

5.6 Do you use any resources other than traditional compute or

data resources, e.g. telescopes, microscopes, medical imaging instruments.

no

5.7 How often is your application used on the grid or grid-like systems?

[ ] Exclusively
[ ] Often (say 50-50)
[x] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

access to KMC is available through a servlet in a portal.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

no resources are selected by users, everything is done by the execution plateform

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

by performances evaluation, CPU and network load, Diet features.

7.3 Are the resource requirements dynamic or static?

dynamic

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

executable code

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

no

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

data protection + authentification

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

simple API

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

the application is not parallelised, such we just need several nodes in case of multiple submissions.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

as the application is not parallel, and the parameters are small sized, just the performance of the executing computer is to considerate for performances.

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

Diet

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

It could be interesting for the server to declare functions (or a table of functions) for interactivity between client and server. For instance, the client could be able to invoc a function during the application execution.

13. References:

List references for further reading.

...

UC 11: LAMMPS

Name of use case: LAMMPS

Contact: Jean-Louis Barrat
<barrat@lpmcn.univ-lyon1.fr>

Authors: Steve Plimpton

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[x]
Other	[ ]
Please specify:	University of Lyon

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[x]
Please specify:	Materials science

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[ ]
Visualization	[ ]
Security	[ ]
Resource discovery	[ ]
Resource scheduling	[ ]

Workflow	[ ]
Data movement	[ ]
High Throughput Computing	[ ]
High Performance Computing	[x]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[x]
Application developer	[ ]
System administrator	[ ]
Service developer	[ ]
Computer science researcher	[ ]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case . Background to the project is another alternative. (E.g. 100 words).

LAMMPS is a classical molecular dynamics code that models an ensemble of particles in a liquid, solid, or gaseous state. It can model atomic, polymeric, biological, metallic, or granular systems using a variety of force fields and boundary conditions.

LAMMPS runs efficiently on single-processor desktop or laptop machines, but is designed for parallel computers. It will run on any parallel machine that compiles C++ and supports the MPI message-passing library. This includes distributed- or shared-memory parallel machines and Beowulf-style clusters.

LAMMPS can model systems with only a few particles up to millions or billions. The current version of LAMMPS is written in C++. In the most general sense, LAMMPS integrates Newton's equations of motion for collections of atoms, molecules, or macroscopic particles that interact via short- or long-range forces with a variety of initial and/or boundary conditions. For computational efficiency LAMMPS uses

neighbor lists to keep track of nearby particles. The lists are optimized for systems with particles that are repulsive at short distances, so that the local density of particles never becomes too large. On parallel machines, LAMMPS uses spatial-decomposition techniques to partition the simulation domain into small 3d sub-domains, one of which is assigned to each processor. Processors communicate and store "ghost" atom information for atoms that border their sub-domain. LAMMPS is most efficient (in a parallel sense) for systems whose particles fill a 3d rectangular box with roughly uniform density.

2.2 Is there a URL with more information about the project ?

http://www.cs.sandia.gov/~sjplimp/lammps

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs. E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

Customer needs to create a trajectory starting from an input configuration and an input script, and to have access later to the created trajectory for analysis. the trajectory may represent an important amount of data and should pre An output configuration ferably be kept on the server that was used for the calculation. An output configuration and runtime flow traces should be returned to the user.

4. Customers:

Describe customers of this use case and their needs. In particular, where and how the use case occurs "in nature" and for whom it occurs. E.g. max 40 words

...

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

Departmental servers at University of Lyon, ENS Lyon, and national resource centers ...

5.2 Are these resources geographically distributed?

Yes ...

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

1

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

Code distributed under GPL licence

5.5 What information sources do you require, e.g. certificate authorities, or registries.

...

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

...

5.7 How often is your application used on the grid or grid-like systems?

[ ] Exclusively
[ ] Often (say 50-50)
[ ] Ocassionally on the grid, but mostly stand-alone
[x] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

...

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

number of processors is selected by users memory needs are determined by the application but can be estimated by the user. ...

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

CPU power is the most important criterion

7.3 Are the resource requirements dynamic or static?

dynamic ...

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

no security required for academic use ...

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

...

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

...

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

...

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

...

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

...

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

...

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

...

13. References:

List references for further reading.

...

UC 12: MNT - Digital Terrain Model

Name of use case: MNT - Digital Terrain Model

Contact: <philippe@lifc.univ-fcomte.fr>

Authors: Gerard Vidal (ENS Lyon - France)

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[X]
Image analysis	[ ]
Other	[ ]
Please specify:	.....

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[ ]
Visualization	[3]
Security	[ ]
Resource discovery	[ ]
Resource scheduling	[ ]
Workflow	[ ]

Data movement	[3]
High Throughput Computing	[ ]
High Performance Computing	[5]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[ ]
Application developer	[ ]
System administrator	[ ]
Service developer	[ ]
Computer science researcher	[X]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case . Background to the project is another alternative. (E.g. 100 words).

This parallel application is based on a stereo vision algorithm. We focus on the particular stereo vision problem of accurate Digital Elevation Models (DEMs) reconstruction from a pair of images of the SPOT satellite. The input data consists in two images of a particular region taken from different points of view. From these images, we extract the three-dimensional information by finding couples of corresponding points and computing 3D coordinates using camera information. The input data size may be very large especially from satellite imagery which produces 6000 x 6000-pixel images, involving important computation times as well as very large memory demand.

2.2 Is there a URL with more information about the project ?

http://graal.ens-lyon.fr/~gridasp/aci_appli2.html

3. Use Case to Motivate Functionality Within a Simple API:

- Source code of DEM is not distributed for customer installation. They just access a DEM server to submit their problem.

- Datas needed for a job submission are six files plus scalar and boolean values.

- job submission may be synchronous : the customer send a set of parameters and check if the result is relevant.

- job submission may be ssynchronous : the customer send several sets of parameters and compare a posteriori the results at the end of the jobs executions.

- result of a job submission is an image file.

4. Customers:

Customers are geologists without computer science experience. They mesure the geological evolution of an area, somethime in place with just a laptop and GSM connection. They need to remotly access the server.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

- hardware : parallel machine / cluster, nodes with large memory
- software : MPI, Diet, Corba, DEMsolver
- visualisation tools (raster files, mpeg player)

5.2 Are these resources geographically distributed?

yes, in case of multiple submissions

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

one per job submission. Usually less than ten if there are mutliple submissions.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

DEM code is not distributed, just installed on three servers. It is written in C / MPI

5.5 What information sources do you require, e.g. certificate authorities, or registries.

no

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

no

5.7 How often is your application used on the grid or grid-like systems?

[ ] Exclusively
[ ] Often (say 50-50)
[x] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

access to DEM is available through a servlet in a portal.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

no resources are selected by users, everything is done by the execution plateform

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

by performances evaluation, CPU and network load, Diet features.

7.3 Are the resource requirements dynamic or static?

dynamic

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

executable code

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

no

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

data protection + authentification

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

simple API

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

compute resource, already tested on a T3E with 256 nodes.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

Data must not be transfered several times

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

Diet

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would

like in an API which would support your scenario.

...

13. References:

List references for further reading.

...

UC 13: RobGrid

Name of use case: RobGrid

Contact: Stephane Vialle,
SUPELEC, 2 rue Edouard Belin,
57070 Metz, France

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[X]
Please specify:	Computing Science & Process control

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[ ]
Visualization	[ ]
Security	[1]
Resource discovery	[ ]
Resource scheduling	[ ]
Workflow	[ ]

```
Data movement                [ ]
High Throughput Computing    [ ]
High Performance Computing   [1]
Other                        [ ]
    Please specify:   Remote and redundant control of
                      process (with time constraints)
```

1.4 Are you an:

```
Application user             [ ]
Application developer        [ ]
System administrator         [ ]
Service developer            [ ]
Computer science researcher  [X]
Other                        [ ]
    Please specify:          ..............................
```

2. Introduction:

2.1 Provide a paragraph introduction to your use case . Background to the project is another alternative. (E.g. 100 words).

This research project is about robot control across a Grid, first step toward a Grid solution for generic process control. A Grid can provide a homogeneous environment for computers from different sites. It can choose the most suitable machine for every task and transparently run redundant computations for critical operations, adding fault tolerance to the remote control.

built a Grid between France and Italy and successfully controlled a navigating robot and a robotic arm. Our Grid is based on the GridRPC paradigm, the DIET environment and an IPSEC-based VPN. We turned some modules of robotic applications in Grid services. Finally we developed a high-level API for specializing the GridRPC paradigm for our purposes, and a semantics for easy adding new Grid services.

2.2 Is there a URL with more information about the project ?

incoming ...

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs.
E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

- Customer uses a laptop+wifi to control a physical process and to be able to walk around during complex experiment. He runs computations on some powerful computing servers in a Grid, and does not care of the choice of the servers.

- Customer runs a complex process control (physical process) composed of many modules. Each module is a service and a Grid middleware manage these services. The RobGrid API allows a smart programming of the application, running critical services several times in parallel (redundant computation) to ensure these operations will succeed.

- Customer controls several physical processes and use several computers, but avoid to devote computers to particular processes. Any machine can be used to control a process, and a Grid middleware manages the matching of the resources and the process currently under control.

- Customer needs to control a process in a hostil environment, with very few computing resources around. He needs to use remote computing resources. The API allows to easily write applications that overlap the "physical times" (time to move some part of the physical processes) with some communication times across Internet, and to access parallel computers when extra CPU power is needed.

4. Customers:

- Customers of this use case can be computing researchers that aim to develop Grid solutions for process control. They can use our Grid and our API to implement their distributed and remote control.

- Customers can be "end users": people that need to control some physical processes from a distant location, or to control a lot of processes spread in the world and a lot of computing resources at different locations.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

- 2 robots (one mobile robot and a robotic arm)
- 1 cluster of 8 machines
- 3 4-processor PC
- 4 monoprocessor PC
- 1 gateway PC for IPSEC VPN management on the main site
- 1 firewall PC for IPSEC VPN management on the main site
- 1 PC on each other site (3 other sites)
- 1 bi-processor PC : grid server for DNS, Corba Name server, ...

- DIET & OmniORB
- IPSEC VPN
- NONO & Koala libraries developed at Supelec (in collaboration with Loria) for robot management
- RobGrid API (developed at Supelec)
- Some Robotic applications developed at Supelc (in collaboration with ENSAM)

5.2 Are these resources geographically distributed?

Yes :
- University of Salerno : one grid computing server
- 1 PC on Metz network ("wanadoo" provider)
  - 1 PC on Metz network ("free" provider)
  - Supelec Metz campus : 2 robots and the rest of the computing resources

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

We have run until 18 services in the same benchmark (during the same application execution) We have run our application on 3 sites (simultaneously) among the 4 availables We have run our 2 robots simultaneously, and 18 services on 5 computing servers, more the firewall, the gateway and the computing server of the grid (DNS, ...), using 3 sites.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

Usually we use gcc and g++, we write C/C++ code, and all are open source codes. We use different tools to write UML documentation (including Rational Rose).

We write and use source code and some pre-compiled versions of DIET.

5.5 What information sources do you require, e.g. certificate authorities, or registries.

We have installed our VPN using certificates, and sometimes "secret strings" (when pb happen with certificates).

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

YES : 2 robots. One mobile robot: Koala robot of K-Team company, and one robotic arm from "Francaise d'instrumentation".

5.7 How often is your application used on the grid or grid-like

systems ?

[X] Exclusively .....
BUT: it was just a testbed for the development of
our RobGrid API
[ ] Often (say 50-50)
[ ] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

- System Libs: Linux, IPSEC VPN, OmniORB COrba Bus, DIET, Nono, Koala
- RobGrid API: management of robot on the Grid, allows to easily write Grid services for robot control
- Robotic application: navigation, landmark detection, self-localization, lightness measurement

We shut down some services to simulate failures and to test our fault-tolerance mechanisms based on redundant Grid computations,and we measure execution times.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

The robot(s) is(are) inerent to the application. For the moment the user choose the machines to use! But in the next version the user should choose only the sites to use for redundant computations, and the Grid middleware should choose the machines to use on these sites.

The choice depends on the availability of the machines. Usually we do not look for powerful machine,s but for available machines for redundant computations. Ex: during the day machines at Metz on Internet and on the network of private provider (wanadoo, free, ...) and with limited Bw (128Kb/s, 512Kb/s) are more interesting than machines at the University of Salerno. During the night it is the opposite.

We run many services (ex: 18 services, corresponding to 8 different services, and some services run redundantly). So we need many machines, but each machine can be a classical PC or a small multiprocessor machine.

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

Main criteria are the OS and the availability of the machines (have different availability at different moment of the day), and the speed of the network to reach these machines.

7.3 Are the resource requirements dynamic or static?

Dynamic: depends on the date (some machines aer needed during the day, and others during the night, for example), depends on the possible failures of the machines and of the network (we choose available" and "reachable" machines).

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

Main sensitive point: if program and/or sensor output values are modified, then the robot can be damaged (physical damage!). So we need security. We have deployed a secure VPN: IPSEC with gateway and firewall, with data encryption, authentication (certificate based), limited login on our Grid ... "Full IPSEC install".

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

IPSEC, ssh and certificates.

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

authentication, authorisation, message protection, data protection.

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

Simple deployment: IPSEC is integrated to Linux, but remains difficult to install on many sites. We succeeded on Salerno and Supelec. We failed on Potenza and Bucarest (UPB). We succeeded on wanadoo and free networks, but each time the machines were ebooted they changed their IP (dynamic IP on these private provider networks) and VPN config had to be updated ....

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

For scalability experiment we need more robots (more physical processes to control), located on different sites ... and we need robotician/automatician partners to improve robotic applications.

We need more sites that accept to deploy IPSEC (sometimes some change have to be done in the site security configuration).

Of course we need more PC, but seems not a pb.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

We have time constraints : when the local machines fail or are overloaded, we use remote machines (remote machines take the control automatically) and we have to deal with Internet communication times. We send camera image, infra red sensor output. When application slow down we can observe the robot slow down.

We use to overlap communication time with computation times and with mechanical times. Each grid service is a kind of pipeline across Internet.

At the end we measure execution time, global execution times on 24h (one benchamrk each 15 minutes, to not burn the robot! we have burnt 2 motors and one robot ...). We can observe that if services are on different machines, when machines fails, not all service fail, a,d so the global slowdown of the application is limited. Many services continue to run on local machines, and remote machines control only some parts of the application. Slow don is limited to these parts. Stategy seems OK.

We try to learn what site is a good candidate to host redundant services at 8h, 12h, 16h, 20h, 24h 4h ...

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

We used to use a VPN based middleware and a GridRPC programming model, then we build our RobGridAPI : kind of distributed and remote object framework (like Java RMI).

But in the future we would like to use a GridRPC formalism on a "Globus-like" middleware, to test another technology than

VPN. We would like to test a middleware that does not need to have accounts on each site.

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

GridRPC was Ok, but we would like it would be easier to initialize the system, to exchange data like camera images. We are experimenting ProCative (other project), the API seems interesting, bu we pefer C/C++ to deal with drivers of robot devices.

RobGrid is RMI-like, so I think a GridRMI, in place of GridRPC could be interesting for our application. Seems more adaped to our use case.

13. References:

List references for further reading.

For an introduction to the robotic application and the first parallelization we experimented :

A. Siadat and S. Vialle (alphabetic order). Robot Localization, Using P-Similar Landmarks, Optimized Triangulation and Parallel Programming. Accepted in ISSPIT 2002: 2nd IEEE International Symposium on Signal Processing and Information Technology. Marrakesh, Morocco, December 18-21, 2002.

For an introduction to the RobGrid API:

F. Sabatier and A. De Vivo and S. Vialle. Grid Programming for Disributed Remote Robot Control.. 13th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2004), Modena, Italy. June 2004.

A article about the entire project has been submited to EGC05

UC 14: GRID-TLSE

Name of use case: GRID-TLSE

Contact: Michel Dayd <dayde@enseeiht.fr>
ENSEEIHT-IRIT, 2 rue Camichel
31071 TOULOUSE CEDEX FRANCE

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[x]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[x]
Please specify:	Sparse linear algebra solvers

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[ ]
Visualization	[ ]
Security	[ ]
Resource discovery	[ ]
Resource scheduling	[ ]
Workflow	[ ]

Data movement [ ]
High Throughput Computing [ ]
High Performance Computing [ ]
Other [x]
Please specify: Expert site for sparse linear algebra

1.4 Are you an:

Application user [ ] Application developer
[x] System administrator [ ] Service developer
[ ] Computer science researcher [ ] Other
[ ] Please specify: .....

2. Introduction:

2.1 Provide a paragraph introduction to your use case .
Background to the project is another alternative.
(E.g. 100 words).

The goal of the Grid-TLSE project is to design an expert site for sparse matrices. This web site provides tools and software for sparse matrices. It also includes a bibliography on sparse matrix software and collections of sparse matrices.

2.2 Is there a URL with more information about the project ?

http://www.enseeiht.fr/lima/tlse

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs.
E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

```
    Example of scenario : determination of the best symmetric
    permutation

   - Identify all the symmetric permutations available
    on the TLSE grid
    for each symmetric permutation
       do
          select one serve offering the symmetric permutation
          send the matrix to the server
          start calculation
          recover the results
       done
    - Determine the symmetric permutation that provided
    the minimun fill-in.
  - Send back the answer to the user
```

4. Customers:

```
  Describe customers of this use case and their needs. In
  particular, where and how the use case occurs "in nature" and
  for whom it occurs.  E.g. max 40 words

   There are two classes of users :
   - expert users that are deploying sparse linear algebra
     sofware over our grid and proceed to experiments aiming at
     comparing, debugging, ... their sofwares.
   - non-expert users that are mainly interested by uploading
     their matrices and determine which solver or which machine
     is the most efficient in terms of memory, execution time,...
     for their problems.
```

5. Involved Resources:

```
  5.1 List all the resources needed: e.g. what hardware, data,
      software might be involved.

    Softwares : large number of sparse softwares . A large
    database that hold description of the solvers, the computers
    available, informations related to the users, results of
    experiments and matrix collections.  The TLSE gird will
```

involve a wide range of computers from PC up to large parallel computers so that users may proceed to a wide range of experiments.

5.2 Are these resources geographically distributed?

At national level and we may anticipate utilization of computers in foreign countries.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

Typically we anticipate tens up to hundreds.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

The sparse linear algebra software is either open source or available in libraries such as Harwell Subroutine Library. The code we deveoped is open source. DIET is also publically available.

5.5 What information sources do you require, e.g. certificate authorities, or registries.

The expert site only requires a on-line registration to identify users.

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

NO

5.7 How often is your application used on the grid or grid-like

systems?

[x] Exclusively
[ ] Often (say 50-50)
[ ] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

We use Java, DIET on top of CORBA (OmniORB), XML, HTTP and Java Server Pages for the web interface. We use PostGresQL for the database. The linear algebra solvers are mainly written in Fortran 77 or 90, C or C++.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

Users are choosing softwares (e.g. sparse solvers) and may additionally select an architecture (e.g. PC LINUX or SGI,...). Selection of the machine(s) - this is a multi-parametric application - running the remote tasks depends on the availability of the software, the load of the computers, considerations on the availability of data on the machines,....

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

Resources are selected partly at the expertise level (depending on the user requirements in terms of software,

size of problem, type of architecture he wants to experiments,...) and at the middleware level (CPU load, management of data persistency,...).

7.3 Are the resource requirements dynamic or static?

The resource requirements are dynamic. Furthermore, an expertise often consists of several steps (each step is launching a bunch of executions over the grid) and executions at a given step depends from the results of previous steps.

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

We rely on the mechanisms that are - or will be available - at the Corba level on which DIET stands.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

We currently use some SSH.

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

Authentification, data protection are certalinly the main issues for our project.

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

Keep it as simple and as hidden as possible.

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

For our project : compute resources and data.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

Since we are not providing computational resources GFlop rate is not the main issue. Probably throughput is the main thing of interest.

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

DIET middleware developed at LIP-ENSL (France) by F. Desprez and his team.

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

Something like

```
Grid_execute('name_of-procedure', parameters )
```

with both synchronous and asynchronous calls and way of testing end of execution is fine. Additionally something sending back the list of services that are capable of

executing a specific operation (giving the name of a procedure - i.e. who can execute GEMM ? - or at a higher level - who is capable of executing a matrix multiplication with symmetric matrices ? -) would be of interest.

13. References:

List references for further reading.

Please refer to our website http://www.enseeiht.fr/lima/tlse.

UC 15: Hybrid Monte Carlo Molecular

Name of use case: Hybrid Monte Carlo Molecular Dynamics

Contact: David E. Konerding <dekonerding@lbl.gov>

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[ ]
Please specify:	structural biology

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[4]
Remote steering	[3]
Visualization	[5]
Security	[2]
Resource discovery	[4]

Resource scheduling	[4]
Workflow	[3]
Data movement	[4]
High Throughput Computing	[5]
High Performance Computing	[1]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[X]
Application developer	[X]
System administrator	[ ]
Service developer	[X]
Computer science researcher	[X]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case.
Background to the project is another alternative.
(E.g. 100 words).

MDDB is a project to build an integrated system for executing and analyzing large-scale molecular dynamics simulations. The goal is to harness commodity MD codes, commodity database products, and commodity distributed computing toolkits into an integrated system which permits the efficient distributed calculation of molecular dynamics trajectories, storage of primary results in a stable archive, analysis of primary and derived data, and visualization of resulting data. There are currently several kinds of simulation algorithms we currently employ (or would like to implement) to generate ensembles of molecular dynamics trajectories for estimation of thermodynamic or kinetic observables:

1: replica-exchange: N iterations of M independent replicas are carried out with a Metropolis Monte Carlo-like exchange trial using potential energy between iterations.

2: Dynamic evolution from a prepared state: Often, the

statistical dynamics of evolution from a prepared state (such as an initial configuration or phase space cell) or number of states (when estimating transition matrices) is of interest [7, 8]. In this case, a number of trajectories of moderate length (10-100 ps) are generated from an initial ensemble of starting states and simulated independently.

3: Transition path sampling: A Monte Carlo simulation in path space is conducted.

4: Hybrid Monte Carlo: sampling from the canonical distribution is achieved by conducting a number of short MD simulations in sequence. Each simulation begins with selection of a new momentum from the Maxwell-Boltzmann distribution, integration of the molecular dynamics trajectory, and then a Metropolis acceptance step using the change in total energy from initial and final steps of the trajectory.

5: Wang-Landau sampling. In Wang-Landau sampling [4, 5], trajectories are generated with random initial velocities as in hybrid Monte Carlo, but the acceptance of the trajectory is based on statistics accumulated in the form of a transition matrix containing counts of the generated transitions. The acceptance criterion attempts to produce a random walk in the coordinates of interest (e.g. potential energy, an intramolecular distance, etc.).

The MMDB use case states requirements for a number of features which are not anticipated to be implemented by SAGA, including:

* Direct relational database access Implementation framework
* for higher level algorithms Application-domain specific
* data processing

2.2 Is there a URL with more information about the project ?

http://www.dillgroup.ucsf.edu/~jchodera/research/mddb/

3. Use Case to Motivate Functionality Within a Simple API:

The MDDB project proposed a task scheduling system which duplicated the functionality provided by typical distributed computing toolkits. The project assumes a heterogeneous architecture, independent storage computing environment and uses it to implement a master/worker workload distribution system. Individual "task units" (typically single MD simulations of 1-100ps) are farmed out to workers by the master which manages the high-level simulation algorithm details. Typical operations performed by the master application are

- Discover resource
- Submit job to resource
- Move input files from storage to resource
- Run job
- Move output files from resource to storage
- High-level simulation algorithm details such as Metropolis Monte Carlo exchange trials

Following the primary simulation runs managed by the master, post-processing tasks are carried out to produce derived data which is then visualized. Many postprocessing steps consume large amounts of CPU, memory and storage and can also be adapted to distributed computing frameworks. The post-processing steps themselves often require advanced data access methods (transactional, relational) in addition to direct file I/O.

4. Customers:

This use case is currently implemented in an ad-hoc fashion across many academic research labs using molecular dynamics for protein folding and other structural biology endeavors.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

Hardware:
- Cluster connected with 100BaseT or gigabit Ethernet
- Large RAID storage system

Software:
- Molecular dynamics engine such as AMBER, CHARMm or NAMD in
- Related data-specific pre- and post-processing applications tools -Generic plotting and data analysis tools
- Scripting languages to automate workflow
- Batch queuing system

Data:
- Starting structures either from experimental structural biology or theoretical models
- Chemical force field

5.2 Are these resources geographically distributed?

The MDDB document specific lists a collection of geographically distributed compute resources that are used in the existing implementation of this use case, including local academic clusters and nationally funded supercomputers.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

In the current implementation, limited by the distributed computing framework, only a single local academic cluster resource is utilized in a single run.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

Molecular dynamics engines (written in C, C++, or Fortran) are licensed from academic research groups either with a GPL, or BSD-like license, or for a fee with a custom academic license. In some cases the source code is used to compile a binary, in other cases, provided binaries are used. Custom workflow and data analysis solutions are normally implemented using Java or Python. A number of

general purpose data analysis tools available under the GPL or BSD licenses are used.

5.5 What information sources do you require, e.g. certificate authorities, or registries.

Currently, no information sources such as CAs or registries are used, although it would be desirable to use them if they were available.

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

Not directly.

5.7 Please link all the above back to the functionalities described in the use case section where possible.

...

5.8 How often is your application used on the grid or grid-like systems?

[ ] Exclusively
[ ] Often (say 50-50)
[ ] Ocassionally on the grid, but mostly stand-alone
[X] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

The application is launched from a shell environment, using scripting language tools to manage the distributed computations.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

Currently, all resources are explicitly selected at the beginning of run time by the user (direct submission to a batch queuing system), at the discretion of the system administrators. The batch queuing system routes jobs to hosts with appropriate CPU architecture and memory based on the user request.

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

The resource selection in 7.1 is also subject to practicability constraints, such as running compute intensive jobs on fast processors and running data-intensive jobs on machines with fast connections to the storage. Cost (exception in the form of service units on supercomputers) is not normally an issue because the resources have already been paid for and reserved. Individual jobs do not normally have large data requirements, but the final data storage requirements can be modest to large (100GB-1TB) which typically drives the archival storage system selection.

7.3 Are the resource requirements dynamic or static?

As designed, the compute resources can be dynamic while the storage system needs to be static.

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

This is an academic endeavor and does not involve proprietary data. Normal Unix security measures protect the data, although as the use case expands to include Grid resources, many of these protections will have to be relaxed. No executable code or hardware security is required.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

ssh

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

Authentication is the primary need. The others are not normally required.

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

Single API, simple deployment, integration with commodity technologies (such as SSH and standard Unix security) are all desirable.

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

Horizontal scaling of compute resources (more nodes), vertical scaling of data (larger single storage), horizontal scaling of nodes with fast data access, and vertical scaling of networks (fast, scalable communication between a master node and remote clusters)

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

Depending on the particular algorithm used, specific instances of this use case will experience varying amounts of performance enhancement relative to single processor jobs, depending on the communication requirements of the job and the heterogeneity of the compute resources available. Further, post processing is data intensive and will require fast, large memory CPUs attached to the storage system.

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

We have been considering Globus, using the Commodity kits (Java and Python)

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

We want the following simple calls: Discover resource, submit job, move file. Possibly some support for computational steering.

13. References:

List references for further reading.

Molecular dynamics engines:

D.A. Case, T.A. Darden, T.E. Cheatham, III, C.L. Simmerling, J. Wang, R.E. Duke, R. Luo, K.M. Merz, B. Wang, D.A. Pearlman, M. Crowley, S. Brozell, V. Tsui, H.

Gohlke, J. Mongan, V. Hornak, G. Cui, P. Beroza, C. Schafmeister, J.W. Caldwell, W.S. Ross, and P.A. Kollman (2004), AMBER 8, University of California, San Francisco.

Laxmikant Kal, Robert Skeel, Milind Bhandarkar, Robert Brunner, Attila Gursoy, Neal Krawetz, James Phillips, Aritomo Shinozaki, Krishnan Varadarajan, and Klaus Schulten. NAMD2: Greater scalability for parallel molecular dynamics.
Journal of Computational Physics, 151:283-312, 1999.

CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations, /J. Comp. Chem./ *4*, 187-217 (1983), by B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus.

Advanced simulation methods:

A. Brass, B. J. Pendleton, Y. Chen, and B. Robson. Hybrid Monte Carlo simulations theory and initial comparison with molecular dynamics. /Biopolymers/, 335:1307-1315, 1993.

Christoph Dellago, Peter G. Bolhuis, and Phillip L. Geissler. Transition path sampling. /Adv. Chem. Phys./, 123:1-78, October 2002.\

M. Scott Shell, Pablo G. Debenedetti, and Athanassios Z. Panagiotopoulos. Generalization of the Wang-Landau method for off-lattice simulations. /Phys. Rev. E/, 66:056703, 2002.

M. Scott Shell, Pablo G. Debenedetti, and Athanassios Z. Panagiotopoulos. An improved monte carlo method for direct calculation of the density of states. /J. Chem. Phys./, 119(18):9406-9411, 2003.

Yugi Sugita and Yuko Okamoto. Replica-exchange molecular dynamics method for protein folding. /Chem. Phys. Lett./, 314:141-151, November 1999.

UC 16: Collaborative Visualization

Name of use case: Collaborative Visualization
of Atmospherical Data

Contact: Herwig Zilken <h.zilken@fz-juelich.de>

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[X]
Image analysis	[ ]
Other	[ ]
Please specify:	.....

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[3]
Visualization	[5]
Security	[ ]
Resource discovery	[ ]
Resource scheduling	[ ]
Workflow	[ ]
Data movement	[ ]

High Throughput Computing	[ ]
High Performance Computing	[3]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[ ]
Application developer	[ ]
System administrator	[ ]
Service developer	[ ]
Computer science researcher	[X]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case.
Background to the project is another alternative.
(E.g. 100 words).

In a joint project, the institute of chemistry and dynamics of the geosphere (ICG2) of the Research Centre Juelich together with the Max-Planck-Institute for Meteorology (Hamburg) are using the transport model MOZART to calculate the distribution of chemical tracers and other physical properties (temperature, pressure) of the troposphere of the earth. Because the resulting dataset is very big, (about 1GB) it cannot be stored locally but only at one or more central servers. There is a demand to visualize this data to analyse it. To do this, the visualization systems must be coupled to the data servers by a fast network to get online access to the data. If the central data servers have enough computing power, also the post- processing of the data could be done there, e.g. to generate graphical primitives (iso-surfaces, streamlines) to speed up the process of visualization. Because the analysis of the data should be done by a distributed team of scientists located in Juelich and in Hamburg, also the visualization- systems should be coupled and synchronized to allow collaborative visualization sessions. It is intended to use a video conference system to support the direct communication of participating scientist.

2.2 Is there a URL with more information about the project ?

...

3. Use Case to Motivate Functionality Within a Simple API:

Scientists from Juelich and from Hamburg want to do a collaborative analysis of their atmospherical data. At each site, visualization systems are used to explore the data. The visualization systems will load an initial scene, connect to the data- servers and request a portion of the data, e.g. a time interval of interest or data for a selection of chemical tracers. The visualization systems also connect to an interaction server, which is responsible for the synchronization of the visualization. Furthermore a video conference system is started. While the visualization session is running, the scientists interact with the data, e.g. they rotate and translate the scene, decide to load additional data or to use several visualization techniques like cutting planes, iso-surfaces, streamlines, trajectories and so on. All changes in the scene are synchronized online by the interaction server which guarantees a consistent visualization to allow a well coordinated collaborative work session. It also could happen that an additional scientist will enter a running session. In this case the actual scene has to be distributed to this new client.

4. Customers:

The users (ICG2, Max-Planck-Institute for Meteorology) are 'topical' scientists (chemist, physicists).

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

- visualization-systems hardware: a broad range of systems from high-end (special graphics workstations, Virtual-Reality-Systems) to low-end (e.g. laptop) should be supported
- visualization-systems software: as the involved community of scientists is very heterogeneous, a wide variety of graphics software should be used, e.g. AVS/Express, VTK, VR-software.
- data-servers: need to have big storage capacity. If the data-server is also used as a preprocessor, it should be a high performance parallel computer.
- interaction-server: stores information of running visualization sessions. No big storage capacity needed here.
- fast network

5.2 Are these resources geographically distributed?

Yes, absolutely. Alls parts of the system (several visualization facilities, data-servers, interaction server) are distributed geographically.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

3 main tasks are running simultaneously:
- data-server activities
- interaction server
- visualization stations, in a collaborative session one at each site

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

Visualization systems: AVS/Express, VTK, VR-Software, e.g. Vista (WorldToolkit based VR-Software developed at University

Aachen) The Visit-Library
(http://www.fz-juelich.de/zam/visit) should be used for
communication. Own developments (implementation of data- and
interaction-server) are done in C++.

5.5 What information sources do you require, e.g. certificate
authorities, or registries.

...

5.6 Do you use any resources other than traditional compute or
data resources, e.g. telescopes, microscopes, medical
imaging instruments.

No.

5.7 Please link all the above back to the functionalities
described in the use case section where possible.

...

5.8 How often is your application used on the grid or grid-like
systems?

[ ] Exclusively
[ ] Often (say 50-50)
[ ] Ocassionally on the grid, but mostly stand-alone
[X] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in,
for example the languages used, the tool-sets used, and the
user environments (e.g. shell, scripting language, or portal).

...

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

The user can select all resources on his own. He can choose the servers and visualization systems.

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

...

7.3 Are the resource requirements dynamic or static?

The network load is highly dynamic because it depends on user-interaction and the (dynamic) number of visualization systems in a collaborative session.

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

The data which is send over the network is sensitive. There should be access restrictions and authentication.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

No.

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

Authentication.

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

An api which provides a reliable certification mechanism to authenticate a user who connects to the system.

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

Network is very important to scale up the number of possible participants of a collaborative visualization session. Also compute and storage resources are critical.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

Because the visualized data is very large, the necessary postprocessing (surface extraction, ...) could be a bottleneck. Furthermore heavy network traffic is concentrated at the central severs, especially at the data servers.

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

...

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

No details can be given now because the project is in an early stage of development. The API should have the following functionality:
- login mechanism for connecting to data- and interaction-server
- data exchange (send, receive) for the scientific data
- data exchange to synchronize the collaborative, distributed session
- naming mechanism to identify objects which are shared between several visualization systems

13. References:

List references for further reading.

...

UC 17: UCoMS Project

Name of use case: UCoMS Project

Contact: Zhou Lei <zlei@cct.lsu.edu>
Andrei Hutanu <ahutanu@cct.lsu.edu>

Authors: Zhou Lei <zlei@cct.lsu.edu>

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organization:

Industry	[ ]
Academic	[x]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[x]
Please specify:	Petroleum engineering applications

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[2]
Visualization	[4]
Security	[4]
Resource discovery	[5]

Resource scheduling	[5]
Workflow	[5]
Data movement	[5]
High Throughput Computing	[ ]
High Performance Computing	[5]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[ ]
Application developer	[ ]
System administrator	[ ]
Service developer	[ ]
Computer science researcher	[x]
Other	[x]
Please specify:	Middleware developer

2. Introduction:

2.1 Provide a paragraph introduction to your use case.
Background to the project is another alternative.
(E.g. 100 words).

The UCoMS research aims are to develop and deploy a Ubiquitous Computing and Monitoring System (UCoMS) for oil/gas exploitation and management. Being a nationally unique research cluster in IT for energy, UCoMS addresses key research issues in the areas of wireless network systems, grid computing, and application software. It includes three cohesive and interrelated areas of projects which together enable the construction of a useful UCoMS prototype for discovery and management of energy resources. The UCoMS proof-of-concept prototype will be developed and deployed utilizing several existing (possibly decommissioned) well platforms in the Gulf of Mexico. The technical solutions will be generally applicable to sensor networks, wireless communications, and grid computing. These solutions will effectively facilitate drilling and operational data logging and processing, on-platform information distribution and displaying, infrastructure monitoring/intrusion detection, seismic processing and inversion, and management of complex surface facilities and pipelines. Decommissioned well

platforms can be monitored and safeguarded using UCoMS, with a potential of fostering new industries as well in the future.

2.2 Is there a URL with more information about the project?

http://www.ucoms.org

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs. E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

A general application in our use case includes:
- Computational resource brokering
- Data collection from largely-distributed data sources
- Task migration and farming
- Result attainment
- Visualization

We need simple API for the following operations: stage executable, find data location, describe resources, visualization. A simple API will be the interface to resource brokering, data collection, task migration, and visualization on devices. All these processes are executed automatically.

Also, we may need a simple API to manage workflow operations.

4. Customers:

Describe customers of this use case and their needs. In particular, where and how the use case occurs "in nature" and for whom it occurs. E.g. max 40 words

Energy companies, energy analysts, government agency

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

Hardware involved in:
- Large-scale compute facilities
- Data storage system
- Visualization devices for good graphics support
- Wireless network and sensors network

Software:
- GAT
- Globus toolkit
- Storage management, such as CCT Archive tool based on GAT or SRB.
- Petroleum engineering simulation software, such as UTChem, IPARS, VIP, and data processing, such as Seismic Unix and Delivery.
- Task farming using CACTUS (or Condor)
- Triana for workflow

5.2 Are these resources geographically distributed?

Absolutely. All parts of the system, such as storage system, compute facilities, and visualization devices, are distributed.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

Usually, there are three kinds of resources involved in: compute facilities, sensor network, data storage system, and visualization devices. A large number of tasks could be executed simultaneously.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availability; do you regularly work from source code, or use pre-compiled applications; what languages

are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

In the project, C/C++/Fortran are involved in compute-intensive data analysis. XML is used for data description. Java/Perl are used for system management and monitoring. OpenGL and VRML will be used for visualization and rendering.

Some of the simulation software packages are open source, such as UTChem, while others require licenses, such as VIP. All middleware that will be used is open source.

5.5 What information sources do you require, e.g. certificate authorities, or registries.

Because of spatial variability, GIS based information management is needed. Seismic data and well log are critical. GSI-based simple CA is used. Since security of data and results is crucial, certificates are important.

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

Yes, because we include IBW, BGM for sensor/wireless devices.

5.7 How often is your application used on the grid or grid-like systems?

[x] Exclusively
[ ] Often (say 50-50)
[ ] Occasionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the

user environments (e.g. shell, scripting language, or portal).

- C/C++/Fortran for reservoir and drilling simulation, for well logging and processing, and for seismic processing and inversion.
- Java/shell/perl/xhtml/portal for information display by browser. It should be web-service based.
- VRML and OpenGL for visualization
- All kinds of computational Grid middleware, such as GAT, Condor-G, Globus toolkit, Triana, and so on, are used.
- Development IDE may be eclipse for Java and C/C++.
- Commercial petroleum engineering simulation software.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

Some resources, such as large-scale compute facilities and data storage system, are inherent in our project. Others, such as for system management, depend on applications.

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

CPU power, memory, and size of datasets are critical.

7.3 Are the resource requirements dynamic or static?

Basically, resource requirements are static. It is decided in advance what resources a particular task will require.

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable

code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

Commercial level security.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

GSI and SSH

8.3 What are your security needs: authentication, authorization, message protection, data protection, anonymisation, audit trail, or others?

Mostly, authentication is our first concern. We also need authorization, and data protection.

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

Simple API, deployment, and integration with commodity technologies.

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

Compute resources, network, and data.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

High performance for data processing, such as seismic processing & inversion, reservoir simulation, real-time

drilling control, is very important, which requires high performance compute facilities and high-speed access to data storage.

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

Firstly, we use GAT over Globus toolkit, Condor, and other data management services.

Secondly, some grid middleware, such as VDT, SBR, DRMAA, IBP, NWS/Ganglia, would like to use with GAT.

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

Here is a scenario with a pseudo code:

```
// for data collection
char **sourceData = {protocol://hostname:000/data1,
                     protocol://hostname2:000/data2,
                     protocol://hostname3:000/data};

DataLocation   dl  = provisionData     (sourceData);
JobDescription job = createJobInstance (hardwareRequirement,
                                        softwareRequirement,
                                        executable_Location,
                                        dl,
                                        output_location,
                                        errorno);

// automatically resource brokering and data moving.
SubmitJob (job);

// Maybe there is job checkpointing and migration in
// background.
```

```
    // optional for visualization
    PostProcessForRender (output_location);

    Also, we possibly need workflow control using simple API.
```

13. References:

List references for further reading.

```
    http://www.ucoms.org
    http://www.gridlab.org
    http://www.lgc.com
```

UC 18: Interactive Visualization Services

Name of use case: Interactive Visualization Services

Contact: Andrei Hutanu <ahutanu@cct.lsu.edu>

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organization:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[ ]
Please specify:	Distributed visualization, Grid computing

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[4]
Visualization	[5]
Security	[1]
Resource discovery	[4]
Resource scheduling	[4]
Workflow	[ ]
Data movement	[3]

High Throughput Computing	[ ]
High Performance Computing	[ ]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[ ]
Application developer	[ ]
System administrator	[ ]
Service developer	[ ]
Computer science researcher	[X]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case.
Background to the project is another alternative.
(E.g. 100 words).

Interactive visualization of large datasets needs resources. The memory, CPU and GPU cycles, software licenses, display and interaction resources needed to manage the ever-growing data sizes are not available on every researcher's desktop.

The advances in Grid computing technologies through projects like Globus, Condor, Unicore and Awaki to name just a few have made the sharing and aggregation of computational resources possible. Compute clusters connected by fast optical networks can be used together to solve larger and more complicated problems than the ones that can be solved by the largest supercomputer in the world.

Unfortunately, the visualization component needed to analyse the generated output is lagging behind. Part of the problem is that there are many very complex grid tools that can be used to distribute a visualization application. Both the complexity and diversity of these tools are barriers in the way of getting them adopted by developers of visualization tools.

Collaborative visualization involving any type of sharing between two or more users adds new requirements to the visualization of large data sets.

2.2 Is there a URL with more information about the project ?

Currently only :
http://www.cct.lsu.edu/Visualization/VizServiceDemo/

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs. E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

- A simulation has produced a large dataset (> 500 Gb), and stored it on a high-performance data archive on a remote machine A. We have two users, B and C using mid-range visualization hardware (dual-processor, high-end graphics card, etc..) large displays and a fast network connections (e.g. 10Gig). The users and the data are in three distinct locations, and a high-end visualization machines D is accessible for both of them. The users want to start and share a visualization session of the above mentioned data. A, B, C and D and are interconnected by optical networks.

  The proposed scenario looks as following : One of the users decides to start the visualization software on D The visualization software will need to spawn off a process on a machine E "near" the data archive. This will be a data service used to load select data of interest for the visualization session as requested by the users. On D (the machine where the visualization software is running), data flowing from A through E is transformed in images and video signal and using specialized hardware and software (to avoid read-back from the graphics card) the video signal is transformed in HD digital video streams. The users consume the video streams and share the interaction with the

visualization service. The networks between A, B, C, D, and E have to be provisioned before or at the time the visualization job is started. The networks, compute and visualization resources are part of a complex resource description that is used to submit the visualization job.

Simple APIs are required for the following operations:
- Scheduling and submitting the visualization job on machines D and E and the networks interconnecting all the machines
- Accessing the data from the data source A on the data service E
- Interprocess communication between D and E and between D and B and C respectively. (asynchronous)

- A very similar scenario is valid for one user only. User B would like to use the high-end visualization machine D to be able to interactively render more data than he's able to render on his local machine. Data service running on machine E is performing compute or memory extensive operations that cannot be performed on the local machine.

4. Customers:

Describe customers of this use case and their needs. In particular, where and how the use case occurs "in nature" and for whom it occurs. E.g. max 40 words

Users that want to collaboratively or individually visualize data so large that it cannot be interactively visualized on their local machines.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

Data source is a large capacity storage system, the data service runs potentially on multiple machines. In the simplest case, the data service serves as a cache for the data, in more complex scenarios uses CPU power to compute

visualization data on-the-fly. Visualization service runs on a visualization machine with at least one graphics pipe (potentially multiple cards).

5.2 Are these resources geographically distributed?

Yes, all the components are on separate machines interconnected by high-speed networks.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

At least two (data service and visualization service) but these can use multiple processors.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availability; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

C/C++ visualization applications. Looking for general framework that can be applied to a wide range of applications. The framework will be open source, working from source code. Using a SOAP engine for inter-task communication.

5.5 What information sources do you require, e.g. certificate authorities, or registries.

Need information about the resources.

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

Not currently.

5.7 How often is your application used on the grid or grid-like systems?

[X] Exclusively
[ ] Often (say 50-50)
[ ] Occasionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

Shell. Portal interaction possible in limited.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

The task of this service framework is to automatically select a set of resources that makes the solving of the task possible. If there are many choices, user selection is possible.

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

Network connectivity, memory and CPU.

7.3 Are the resource requirements dynamic or static?

Once the resources are selected they shouldn't need to be changed.

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

Data might be sensitive, as well as software resources (licenses). Hardware resources are always sensitive.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

Using GSI in various components.

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

Need, authentication and authorization and audit trail for all the components. Data protection possibly needed.

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

User-friendliness. User should only input a password requested by the application when needed. An operation should not fail until it explicitly asks the user for the required credential information.

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

Data size can scale indefinitely. Network capacity up to the current optical network capacities -- tens of Gbit/s. Compute resources -- up to 5 components in the entire scenario.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

Asynchronous transfer is required for efficient bandwidth usage (need ability to pipeline multiple transport operations).

Network transfer mechanism to scale up to the current fastest capacity networks.

Should be able to choose more security vs. more performance.

API for remote file access should not limit the performance of the storage system.

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

Web (Grid) services, grid job submission.

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

For data transport:

```
// starts send operation. callback is called at
// the end of the operation
AsyncResult beginSend (void* buf, int bufsize,
                       AsyncResultCallback cb,
                       void* userData);

// releases resources, if called before callback blocks
endSend (AsyncResult handle);

AsyncResult beginReceive (AsyncResultCallback cb,
                          void* userData);
```

```
Block* endReceive (AsyncResult handle);

void releaseBlock (Block* block);

AsyncResult beginConnect (AsyncResultCallback cb,
                          void* userData);

void endConnect (AsyncResult handle);

class AsyncResult {
 // cancel the operation
 void cancel ();
}

typedef struct Block {
        const void* const buffer;
        size_t            size;
} Block;


For job submission:
-------------------

No detailed API but should be able to specify
something like:

// Unspecified resource -- only requirements
ResourceDescription A = ResourceDescription
                        (10 CPU, 40 Gbyte RAM, X Gflops);
ResourceDescription B = ResourceDescription
                        (4 GraphicPipes, 4 Gbyte Video Ram)
                        + A;

// semi-specified resource : One of the machines that
// holds a copy of my data file
ResourceDescription C = ResourceDescription
                        (LogicalFile ("my_data_file"));

// specified resource : the two users
ResourceDescription D = ResourceDescription ("pc_1.lsu.edu");
ResourceDescription E = ResourceDescription ("pc_2.lsu.edu");

// hybrid resource .. some specified, some unspecified
// resources that need to be connected by a 10Gbit/s net
ResourceDescription F = ResourceDescription
```

```
                          (Network (A, B, C, D, E, F,
                                    10 Gbit/s));

    JobSubmit (myJob, F);
```

13. References:

List references for further reading.

```
   http://wiki.cct.lsu.edu/wiki/space/\
           Andrei+Hutanu/Prohaska_et_al_MardiGras2005.pdf
    http://www.cct.lsu.edu/Visualization/VizServiceDemo/
    http://www.zib.de/visual/projects/gridlab/hdf5/
    more to come..
```

UC 19: Iterative Image Reconstruction

Name of use case: Iterative Image Reconstruction

Contact: Rainer Schmidt <rainer@par.univie.ac.at>

Authors: Siegfried Benkner,
Rainer Schmidt,
Gerhard Engelbrecht

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[X]
Other	[ ]
Please specify:	.....

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[4]
Visualization	[2]
Security	[3]
Resource discovery	[4]

Resource scheduling	[ ]
Workflow	[1]
Data movement	[4]
High Throughput Computing	[4]
High Performance Computing	[4]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[ ]
Application developer	[ ]
System administrator	[ ]
Service developer	[ ]
Computer science researcher	[X]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case . Background to the project is another alternative. (E.g. 100 words).

The European GEMSS Project is concerned with the creation of medical Grid service prototypes and their evaluation in a secure service-oriented infrastructure for distributed on demand/supercomputing. One of the applications considered in GEMSS is medical image reconstruction for single photon emission computed tomography (SPECT), which is an effective and robust method for diagnosis of cancer, heart diseases and functional pathologies of the human body.

2.2 Is there a URL with more information about the project ?

http://www.gemss.de/

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs.
E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

1) Service discovery: The registry is inquired by specifying service categories (e.g. SPECT) and arbitrary attributes (name-value pairs).

2) Service selection: Is done be a GEMSS specific QoS negotiation process. Clients and services basically exchange XML descriptors which then might lead to a QoS contract and further on to a resource reservation.

3) Upload of 2D projection data.

4) Upload of reconstruction meta-data (region of interest, accuracy, etc.).

5) Start reconstruction.

6) Querying status information (optional).

7) Download of results.

If the user is not satisfied with the results, he may initiate another reconstruction with different parameters (meta-data) by starting with step 4.

4. Customers:

Describe customers of this use case and their needs. In particular, where and how the use case occurs "in nature" and for whom it occurs. E.g. max 40 words

- Science: Application developers using the middleware to test and develop codes on different platforms and conditions.

- Industry: Medical staff using a graphical client for on

demand "real-time" image reconstruction.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

- Compute Resources: Parallel computers (clusters) on the service-side Any Java enabled platform on the client-side
- Advance reservation and scheduling system (e.g. COSY, Maui) on service hosts.
- Web/Grid Service hosting environment
- Security infrastructure (PKI infrastructure for x509v3 certificate handling)
- Visualization Software (e.g. Spect Client developed by Univ. Vienna)

5.2 Are these resources geographically distributed?

Yes, the the current GEMSS testbed comprises two PC clusters located at NEC St. Augustin and ISS Vienna, and various distributed client installations.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

One client negotiates with n services, the particular reconstruction is a one-to-one relationship between client and service. Several clients can execute the same use case in parallel on the Grid.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availability; do you regularly work from source code, or use pre-compiled applications; what languages are

your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

The GEMSS middleware will be distributed under LGPL licence.

There are various applications (mostly parallel MPI codes in C and Fortran) which are open and closed source and run under different licenses. In general no changes are required to integrate the applications with the GEMSS infrastructure.

GEMSS makes use of diverse open-source 3rd party tools and libraries (only stable releases) e.g. Apache ant, tomcat, axis which are not modified.

5.5 What information sources do you require, e.g. certificate authorities, or registries.

We use x509v3 certificates, a CA located at NEC and registration authorities (RA) at the individual institutions.

Registries in GEMSS provide a common Web service interface to be independent from the actual implementation (e.g. UDDI).

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

The SPECT application uses CT scanners to acquire 2D projection input data. The medical instruments are not directly utilized by the Grid infrastructure.

5.7 How often is your application used on the grid or grid-like systems?

[ ] Exclusively
[X] Often (say 50-50)
[ ] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

The Grid applications are exposed as Web/Grid services which implement application, QoS negotiation, and recovery interfaces. The services can be deployed into Java Web service hosting environments.

Client application make use of an API to contact the registry services, specify the QoS descriptors, negotiate with the services, and execute the application workflow.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

Resource selection can be done at different levels. Services can be selected directly (by endpoint), be found in a registry, and/or be negotiated based on several constraints. Resource selection, negotiation strategy, etc. can be delegated to a component provided by the GEMSS client API or can be individually implemented by the client application developer.

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

As the applications are pre-installed on the computing resources the user is not aware of the hosting OS. The GEMSS service environment incorporates performance and pricing models as well as an interface to the resource scheduling and reservation system. During the negotiation process clients specify the resource requirements in means of begin/end time and price.

7.3 Are the resource requirements dynamic or static?

Resource requirements and offers are negotiated dynamically, then fix for the execution.

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

Transmission of medical data is very sensitive and advanced security a major success criteria within GEMSS. We require end-to-end security which does not allow job delegation and remote data storing.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

Message layer: Own implementation of WS security (signing + encryption). Transport layer: SSL connection.

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

All of them (anonymisation of medical data is often only possible to a certain degree).

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

- Integration with commodity technologies (e.g. GSI).
- Simple deployment, integration, configuration (is currently rather complex).
- A simple API to select what security mechanisms should be used.

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

Data is shipped via encrypted SOAP messages, the size is depending on the application (1 - 10 MBs for SPECT data). Patient data must be stored remotely (exept the current session data) for other purposes than processing. This requires high network capacity.

The client requests which are made during a QoS negotiation process result in temporary resource reservations. If many clients request one resource within short time, the system could be blocked by too many temporary reservations.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

The application performance models is based on historical data and meta-data descriptions (input data description, number of iteration, number of nodes). We require dependable performance estimates (which is difficult for some applications) as the reservation system relies on them.

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

- GEMSS is Web service based, we use: WSDL, SOAP (attachments), WS-Security
- We have experimental results with the OGSI (lite) container.
- For Scheduling and advance reservation the COSY resource scheduling system and MAUI scheduler and resource reservation system are used.

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

We would need support for resource reservation and QoS negotiation.

13. References:

List references for further reading.

S. Benkner, G. Berti, G. Engelbrecht, J. Fingberg, G. Kohring, S..E. Middleton, and R. Schmidt. GEMSS: Grid-infrastructure for Medical Service Provision. In Proceedings of HealthGRID 2004, Clermont-Ferrand, France. January 2004.

D.M. Jones, J.W. Fenner, G. Berti, F. Kruggel, R.A. Mehrem, W. Backfrieder, R. Moore, A. Geltmeier. The GEMSS Grid: An Evolving HPC Environment for Medical Applications. In Proceedings of HealthGRID 2004, Clermont-Ferrand, France. January 2004.

Jean A.M. Herveg, Federico Crazzolara, Stuart E. Middleton, Darren Marvin, Y. Poullet. GEMSS: Privacy and security for a Medical Grid. In Proceedings of HealthGRID 2004, Clermont-Ferrand, France. January 2004.

UC 20: RealityGrid

Name of use case: RealityGrid: Real Scientific Computing on the Grid

Contact: Shantenu Jha <s.jha@ucl.ac.uk>
Centre for Computational Science
University College London

Stephen Pickles <stephen.pickles@man.ac.uk>
Manchester Computing
The University of Manchester

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[X]
Please specify:	Condensed-matter Physics - Complex fluid dynamics - Biomolecular systems - Polymer physics

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[5]
Visualization	[5]
Security	[4]
Resource discovery	[1]
Resource scheduling	[4]
Workflow	[ ]
Data movement	[4]
High Throughput Computing	[3]
High Performance Computing	[5]
Other	[ ]
Please specify:	Checkpoint, Migration

1.4 Are you an:

Application user	[X] Application developer
[X] System administrator	[ ] Service developer
[X] Computer science researcher	[ ] Other
[ ] Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case . Background to the project is another alternative. (E.g. 100 words).

The RealityGrid project is about enabling the use of high-end scientific codes on a computational grid. The project involves a wide range of application codes spanning a large number of areas of physics. Most of the applications are tightly-coupled parallel simulations written in MPI or OpenMP.

The use case common to all applications is the "computational steering" of the code. However "computational steering" is a broad term and incorporates many ideas.

At the simplest level, computational steering involves the ability to monitor the evolution of the system under study, and to manipulate parameters that affect the system's behaviour.

The usefulness of computational steering is enhanced by the ability to take a checkpoint of the simulation, spawn a new simulation (possibly on a different number of processors on a different computational resource) from an existing checkpoint, and sometimes to rewind a running simulation to the state defined in a previous checkpoint.

The ability to visualize aspects of the simulation as it evolves is a frequent requirement. In a subset of applications, it is more natural to influence the behaviour of the simulation by interacting with the visualization.

2.2 Is there a URL with more information about the project ?

http://www.realitygrid.org/
http://www.sve.man.ac.uk/Research/AtoZ/RealityGrid/
http://www.sve.man.ac.uk/Research/AtoZ/RealityGrid/ \
Steering/ReG_steering_api.pdf

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs. E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

a. Locate a resource to host a simulation or visualization code (currently done manually) b. Launch a simulation or visualization on an appropriate resource

c. Stage input files (including checkpoint files) and retrieve or archive output files

d. Monitor and steer the simulation as it runs through user-friendly GUIs. We regard pause and resume as steering operations.

e. Dynamically connect/disconnect the steering GUI to/from the simulation

f. Visualize one or more aspects (e.g. physical fields) of the running simulation - on-line visualization

g. Dynamically connect/disconnect the visualization to/from the simulation

h. Take a checkpoint of the status of the simulation i. Discover and manage checkpoint files and associated metadata

k. Rewind to a previously taken checkpoint and/or spawn a new simulation from an existing checkpoint (using functionality in i-iii)

l. Co-allocation and advance reservation of resources to host simulation and visualization components

m. Realtime analysis of live streams of data emitted from a simulation (this is a generalization of the concurrent visualization requirement in (f)).

4. Customers:

Describe customers of this use case and their needs. In particular, where and how the use case occurs "in nature" and for whom it occurs. E.g. max 40 words

The customers of this use case are computational scientists, with reasonably mature and sophisticated codes, and are looking towards the Grid for faster, more efficient and effective environments to support in silico scientific investigations.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

Hardware: High end computers for both simulations and visualization
Data: No specific data requirements
Software: Application codes instrumented with the

RealityGrid Steering library. Currently we
also depend on:
- OGSI/WSRF::Lite as a hosting environment for
  our services
- globus for file transfer and job submission

5.2 Are these resources geographically distributed?

Yes. Application scientist often use machines on the TeraGrid and NGS for simulations as part of the same calculation.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

Possibly many. They may be homogenous tasks (e.g. several replica of a similar simulation) or heteregenous tasks (visualization + simulation).

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

Within the RealityGrid project, there are several different application codes, which may be open-source, close-sourced or third-party codes under license/agreement.

Currently all tools used are publically available.

All applications involve compiling from source code, as we modify them to interface with the RealityGrid steering library.

Application codes are written in different languages- Fortran, C, C++, and are mostly parallel (MPI,OpenMP).

Tools and "glue layer" are written primarily in C, C++ and Perl.

5.5 What information sources do you require, e.g. certificate authorities, or registries.

We require digitial certificates. We get these from the UK e-Science CA. Some collaborators use the DOE CA.

We use our own registry (implemented using OGSI service group constructs) to discover running simulations. We do not currently use registries such as GIIS or UDDI for resource discovery - the set of resources which have sufficient capability for our purposes, and to which we have access, and on which our applications have been deployed is small enough that configuration files are adequate.

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

A "sister" project which will be launched soon, will involve instrumentation.

5.7 Please link all the above back to the functionalities described in the use case section where possible.

???

5.8 How often is your application used on the grid or grid-like systems?

[ ] Exclusively
[ ] Often (say 50-50)
[X] Occasionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

The primary reasons for the above:
- Overhead and "grid" stability reasons: If we had a persistent, stable and usable grid [24/7, 365], we would probably migrate to category 1 (i.e. exclusively) for certain applications.
- Not all problems addressed simulations require features which come as value-added on the grid

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

Simulation codes are written in Fortran90 and C, and instrumented using server-side functions of the RealityGrid steering library.

Steering clients exist in several flavours:
* The Qt/C++ GUI for workstations uses the client-side
* functions of
the RealityGrid Steering Library.
* A .NET client suitable for PDAs is tooled against the WSDL description of the RealityGrid Steering Grid Service and Service Registry.
* A Java client packaged as a GridSphere Portlet is tooled against WSDL.
* A Java client in the ICENI framework has been built against the Steering Library using JNI.

Job launching and migration is treated separately to computational steering. We use a graphical "wizard" for these purposes; it is written in Qt/C++ and shells out to Globus commands (using either commands in the Globus client distribution, or available with the Java CoG kit). The wizard uses gsoap to communicate with the RealityGrid services (registry, checkpoint tree, and steering grid services).

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

Hardware resources are chosen by the user, As well as which grid-service container to use (in turn partly determined by resources chosen). The location of the RealityGrid registry is pre-determined.

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

Resources used are determined by:
i. availability
ii. compatibility with requirements, e.g.
there are some problem sizes which can run only on the largest possible machines

The decision as to which resource to use is made by the application scientist based upon i + ii. Currently there are just a handful of resources typically accessible, but at some stage (e.g. as the number of choices increase) a resource broker may become critical.

7.3 Are the resource requirements dynamic or static?

ReG Applications fall in both categories i.e. dynamic and static.

A fundamental premise of computational steering on the grid, is that while interacting with a live simulation, if the analysis requires further investigation, the infrastructure exists to do so (this may involve spawning simulations, starting and connecting a visualization, or just simply rewinding). Thus the hardware resources are in general dynamic. The RealityGrid computational steering system provides the tools and software infrastructure required to use the hardware resources.

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

As our use-case is exclusively academic, security is a lesser priority than simple and efficient utilization. Theft or malicious tampering of data would be an inconvenience (and embarrassment) but wouldn't be the end of the world.

As typically hardware platforms used are externally maintained and owned, security concern is that of a "responsible user".

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

GSI (but this is not built into the application, but into the tools layer).

8.3 What are your security needs: authentication, authorisation, message protection, data protection,

anonymisation, audit trail, or others?

Authentication and authorisation on the end resource is adequately provided by GSI for the purposes of job launching and file transfer. Security of the middle-tier services used for computational steering is currently lacking, due largely to the absence of a standardised security model that supports delegation in a Web service world. Message and data protection matter, but are not urgent (however some industrial collaborators take a different view). Anonymisation is a non-requirement for us. Audit trail is important as a piece of the larger provenance puzzle.

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

All of these are important. Also important are the ease of management of private keys for mobile users.

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

Compute resources: the more the merrier. Both in terms of the number of resources as well as the performance (CPU and interconnect) of the resource.

Networks: Some applications can require signficant amounts of data being shipped around; larger the physical system, larger will be the requirement. Also, if real-time vizualization is used, the network capacity should scale while the reliabilty must remain high

Our approach for computational steering introduces an overhead that can diminish scalability due to the implied synchronisation of the parallel simulation code and serialisation of output data streams through a single processor. However, the impact on scalability is acceptably small in most cases of interest to us.

Scalability of visualization software and hardware as the size of the dataset increases is important - we need

visualizations that can keep up with the simulation.

Arguably our most important scalability concern relates to the human effort involved in porting and deploying applications (and the middleware stacks they depend on) to an increasing variety of Grid resources.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

Performance of
i. High-end computational/visualization resources is critical.
ii. Data transfer rates between distributed resources

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

Currently using: gsissh, globus-url-copy, globus-job-run, GSI, WSDL, Grid Services (OGSI::Lite), SOAP, XML, GridSphere, Access Grid, VizServer, Chromium, gsoap.

Currently considering WS-RF, WS-Notification, WS-Security, SRB, JSDL, UNICORE, Sakai.

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

When talking about APIs, it maybe helpful to differentiate between application level API and tool level API.

In developing tools for job management and data transfer, we would welcome simple APIs that provide:
- the ability to define and launch a job on a Grid resource
- copy a file from one remote machine to another
- sufficient abstraction to future proof our codes from changes in fashion.

We would also welcome convergence on a standard API for instrumenting a code for computational steering and on-line visualization. This would increase the motivation for computational scientists to adopt computational steering techniques.

13. References:

List references for further reading.

1. Shantenu Jha, Stephen Pickles and Andrew Porter
   A Computational Steering API for Scientific Grid Applications: Design, Implementation and Lessons
   GGF12 Workshop on Grid Application Programming Interfaces.

2. Stephen Pickles, Robin Pinning, Andrew Porter, Graham Riley, Rupert Ford, Ken Mayes, David Snelling, Jim Stanton, Steven Kenny, Shantenu Jha,
   The RealityGrid Computational Steering API
   Version 1.1, RealityGrid technical report, 2004,
   http://www.sve.man.ac.uk/Research/AtoZ/RealityGrid/ \
   Steering/RealityGrid_steering_api.pdf).

3. The Use of Recovery and Checkpoint in RealityGrid
   Stephen Pickles for GridCPR WG
   http://gridcpr.psc.edu/GGF/
   http://gridcpr.psc.edu/GGF/docs/ReG-GridCPR-use-cases.pdf

4. S.M. Pickles, R. Haines, R.L. Pinning and A.R.Porter,
   Practical Tools for Computational Steering,
   Proceedings of the UK e-Science All Hands Meeting, 2004.
   http://www.allhands.org.uk/proceedings/papers/201.pdf

UC 21: GRID superscalar

Name of use case: GRID superscalar

Contact: Rosa M. Badia <rosab@ac.upc.es>

Authors: Raul Sirvent
Rosa M. Badia

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[X]
Please specify:	Automatic gridification of user provided code.

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[ ]
Visualization	[ ]
Security	[ ]
Resource discovery	[ ]

Resource scheduling	[4]
Workflow	[5]
Data movement	[3]
High Throughput Computing	[ ]
High Performance Computing	[3]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[ ]
Application developer	[ ]
System administrator	[ ]
Service developer	[ ]
Computer science researcher	[X]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case . Background to the project is another alternative. (E.g. 100 words).

The aim of GRID superscalar is to reduce the development complexity of Grid applications to the minimum, in such a way that writing an application for a computational Grid may be as easy as writing a sequential application. Our assumption is that Grid applications would be in a lot of cases composed of tasks, in most cases repetitives. The granularity of those tasks will be as simulations or programs, and the data objects will be files. GRID superscalar provides an underlying run-time that is able exploit the parallelism existent at the task level, select the machine to send each task, send and receive the required files, etc.

To adapt a user application to use the GRID superscalar only some slight modifications have to be performed in the code, mainly, the programmer selects those functions or programs of the application that should be executed in the grid by writing its interface in an IDL file. Any sequential code may benefit from the GRID superscalar.

2.2 Is there a URL with more information about the project ?

http://people.ac.upc.es/rosab/index_gs.htm

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs.
E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

We suppose that a customer wants to gridify its sequential code and execute it into a grid. Moreover, we extract all possible parallelism between these tasks. These tasks are tipically simulators, that need input files (configuration files, trace files, data files, ...) to perform its computation. So, the main functionalities needed are:

- Task submission: Allow a remote invocation of a executable file. This file can receive some parameters as inputs, and some files, that must be transfered to the machine (the worker) with the executable file.
- State notification: We need notifications in order to know when the transfer of files to the worker machine ends, when a task is waiting into a queue, when is active, and when it has finished.
- Results collection: At some stages during the execution of the user's main program developed with GRID superscalar, the runtime needs to transfer some files in order to see if the task has executed correctly, and to finally retrieve results of the execution. The transfer will be sometimes synchronous when we are retrieving a file that is immediately needed in the master, and others will be asynchronous when the file needs to be transferred when a job ends.

4. Customers:

Describe customers of this use case and their needs. In particular, where and how the use case occurs "in nature" and for whom it occurs. E.g. max 40 words

A user has developed or is developing an algorithm (more or less complicated) and he does this step without thinking in parallelism or the grid (in a sequential programming way). Then, this user thinks in a way of executing its code in a grid, or to extract parallelism from its algorithm. Here GRID superscalar can help him.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

- Computers or clusters forming a grid to execute jobs.
- Globus Toolkit 2.4 in order to run jobs (this could be changed).
- gsiftp service in all the different machines in order to transfer files involved in the computation (this could be changed).

5.2 Are these resources geographically distributed?

They could be.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

Minimum one (master and worker running on the same machine) and maximum unlimited. In every machine more than one worker can be defined in the configuration files (all workers the user considers he needs).

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of

third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

GRID superscalar library is distributed upon request (without source files). We are using GT2.4 for running jobs, gsiftp for transferring files, and we are introducing the Condor ClassAds library for specifying restrictions.

User code can be written in C/C++ or Perl. We generate some stubs in both parts (master and worker), that must be compiled with the main code of the application in order to be run.

5.5 What information sources do you require, e.g. certificate authorities, or registries.

Certificate Autority to sign Globus certificates.

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

No.

5.7 How often is your application used on the grid or grid-like systems?

[X] Exclusively
[ ] Often (say 50-50)
[ ] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

As explained above, user can write its code in C/C++ or Perl. So, user's environment is a unix shell where he can edit, compile, and launch its code. The tool used is GT2.4.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

User tells the runtime what resources it can use, but the runtime chooses the best appropriate resource.

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

The runtime estimates how many time a task needs in order to be executed in a machine (this is time of transferring input files plus time for executing the task there). The time of transferring input files can be avoided

7.3 Are the resource requirements dynamic or static?

Dynamic. If files are already in the machine, we are not going to transfer them, so the transfer time is not considered.

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

Security is determined by Globus (GSI).

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

GSI.

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

Authentication, authorisation, data protection.

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

Integration with commodity technologies.

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

The application tries to scale using all different available computing resources.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

Since performance improvement is pursued, degradation due to the underlying grid middleware will affect (i.e. with GT2.4 we have 30 seconds penalty per task).

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

GT2.4.

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

```
job_submit (job_description, machine_where_to_run)
job_state  (job_id)
job_cancel (job_id)

copy_remote_file   (source, dest)
erase_remote_files (file1, file2, ..., machine)

wait_for_notifications ()

/* to know more information about
the ending of the remote execution. The objective of this
call is to obtain an error code from the remote application
itself, not just the grid middleware error code */
get_job_error_code (job_id)
```

13. References:

List references for further reading.

- Rosa M. Badia, Jess Labarta, Ral Sirvent, Josep M. Prez, Jos M. Cela, Rogeli Grima: "Programming Grid Applications with GRID Superscalar", Journal of Grid Computing, Volume 1 (Number 2): 151-170 (2003).
- GRID superscalar web site: http://people.ac.upc.es/rosab/index_gs.htm

UC 22: Computational Steering of

Name of use case: Computational Steering of
Ground Water Pollution Simulation

Contact: Wolfgang Frings <w.frings@fz-juelich.de>
Research Centre Juelich
Central Institute for Applied Mathematics
D-52425 Juelich

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[ ]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[X]
Image analysis	[ ]
Other	[ ]
Please specify:	.....

1.3 Which of the following apply to or best describe this use case Multiple selections are possible, please prioritize with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[5]
Visualization	[5]
Security	[ ]
Resource discovery	[ ]

Resource scheduling	[ ]
Workflow	[ ]
Data movement	[ ]
High Throughput Computing	[ ]
High Performance Computing	[4]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[ ]
Application developer	[ ]
System administrator	[ ]
Service developer	[ ]
Computer science researcher	[X]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case . Background to the project is another alternative. (E.g. 100 words).

At the Institute of Petroleum and Organic Geochemistry (ICG-4) of the Research Centre Juelich two parallel codes (TRACE and PARTRACE) for the simulation of solute transport in heterogeneous soil-aquifer systems (e.g. pollutants in ground water) have been developed and are subject to continuous enhancements. The codes TRACE (ground water flow) and PARTRACE (solute transport) can run independently or as a coupled MPI-application. The progress of the running simulation can be monitored by ParView, an AVS/Express based application that allows an online-visualization of the coupled simulation. Among the steering capabilities of ParView are the ability to insert solutants into running simulations and the ability to select 3D-points in the simulated area for which more detailed data analysis and recording is required (so called break-through curves - BTC). ParView uses VISIT, a library for online visualization and computation steering for its connection to TRACE and PARTRACE.

2.2 Is there a URL with more information about the project ?

http://www.fz-juelich.de/zam/visit
(--> 'Projects using VISIT' --> ParView)

3. Use Case to Motivate Functionality Within a Simple API:

The parallel simulations TRACE (A) and PARTRACE (B) are submitted as a single batch-job to the batch-system of an HPC-system. Occasionally, a visualization/steering application (C) is attached to (A) and (B) with separate connections. (A) and (B) send data (large arrays and small parameter sets) to (C) and upon request receive steering parameters back from (C). Eventually, (C) detaches from (A) and (B).

4. Customers:

The users (ICG-4) are 'topical' scientists (chemist, physicists) that use TRACE/PARTRACE to treat their scientific problems, like calculating transport properties of the ground or compare their models and simulations with real-life experiments. The simulations are mostly performed on the IBM supercomputer in Juelich. They use the visualization/steering capabilities mainly to save computer (and their own) time by stopping simulations that go wrong and by adjusting the BTCs during the simulation according to intermediate results.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

- parallel computer,
- local computer with visualization capabilities (from Laptop to Cave)
- sufficient network bandwidth

5.2 Are these resources geographically distributed?

most of the time no, since supercomputer and user are located on the campus of the Research Centre Juelich, but occasionally yes, when a researcher is visiting remote collaboration partners or conferences.

In metacomputing experiments that are conducted within the project VIOLA, TRACE and PARTRACE are running on two or more parallel computers which are geographically distributed. In that case, the visualization is also running at a different location.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

typically a single parallel computer, on which TRACE and/or PARTRACE is running, plus the users visualization-device.

In metacomputing experiments (e.g. in the VIOLA project), the simulations may be distributed over two or more parallel computers.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

- TRACE (Fortran 90), PARTRACE (C++) are own developments of ICG-4 and undergo frequent enhancements
- VISIT (C) and ParView (AVS/Express) are being developed by ZAM
- no proprietary software except Fortran and C/C++ compilers and AVS/Express are needed

5.5 What information sources do you require, e.g. certificate authorities, or registries.

A registry is required that allows the components of the system to find each other. With VISIT, the visualization/steering application registers its service (being able to steer a specific application) and the simulation queries the registry for possible steerers.

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

no.

5.7 Please link all the above back to the functionalities described in the use case section where possible.

Section 3. lists the minimal functionally required to realize the above.

5.8 How often is your application used on the grid or grid-like systems?

[ ] Exclusively
[ ] Often (say 50-50)
[X] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

Simulations coded in FORTRAN90 and C++, running on UNIX/AIX-systems under control of a batch-system
Visualization coded in AVS/Express, running on Linux or Windows-systems

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying

the architecture and memory to run the remote tasks?

The user selects all resources:
- the HPC-system by submitting the simulation to a particular batch-system (which chooses when and on which CPUs to run the job)
- the visualization-system, which is typically his workstation

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

depending on the batch-system, typically by specifying number of CPUs and maximum runtime

7.3 Are the resource requirements dynamic or static?

static during a single run (simulation) visualization can be dynamic, attachment to a simulation from different systems with different capabilities (performance, network bandwidth)

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

The simulation establishes a network connection to an external application (the visualization/steering application) and is controlled by it. Therefore the steerer needs to authenticate properly. Besides that, there are no special security considerations.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

UNICORE

8.3 What are your security needs: authentication,

authorisation, message protection, data protection,
anonymisation, audit trail, or others?

authentication and authorisation of the steerer.

8.4 What are the most important issues which would simplify
your security solution? Simple API, simple deployment,
integration with commodity technologies.

a simple API for the simulation and the
visualization/steering application that is mostly independent
of the underlying Grid system (like specifying a certificate
or password to use/accept)

9. Scalability:

What are the things which are important to scalability and to
what scale - compute resources, data, networks ?

Large parallel simulations may produce huge amounts of data.
Therefore the system must scale with respect to:
- number of CPUs/tasks of the simulation
- volume of data to be transferred to visualization
- volume of data to be visualised

10. Performance Considerations:

Explain any relevant performance considerations of the use
case.

for visualization and steering, latency and data throughput
of the system may be an issue. Aspects that influence that,
are:
- simulation performance: interaction with the visualization
only takes place at certain times, typically once per
simulation time-step. To get some interactivity, such a
time-step should not take to long
- network bandwidth for large bulk transfers of simulation
data may be required
- visualization capability

In total, the simulation uses an expensive resource (HPC-time) and should be slowed down as little as possible by the interaction with the visualization/steering application.

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

We have build a version of VISIT, which uses UNICORE to access the parallel computer through a firewall and are currently working on an integration with the new Web-Services based UNICORE being developed in the UniGrids project.

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

Here an example of the simulation (client) API of VISIT:

```
/* attach to a visualization, using a service-name and a
 * password */
vcd = visit_connect_seap (servicename, password, timeout);

/* send some data */
visit_send (vcd, tag, timestanp, data, datatype, dimensions);

/* receive some data */
visit_recv (vcd, tag, &timestamp, data, &datatype,
            &dimensions);

/* detach from visualization */
visit_disconnect(vcd);
```

For higher-level functions that provide more options and parameters (e.g. complex data-structures to be exchanged, distributed data-arrays in a parallel application) a static API could end up in a large number of functions with complex parameter-lists. An alternative is a simple specification-language for the data to be exchanged and a code-generator that generate wrapper-functions with an

application-specific API. Here is an example code:

```
/* init connection for application 'trace' */
lvisit_trace_init();

while (SimTime) {

  /* test connection, open a new one if necessary  */
  lvisit_trace_check_connection();

  /* receive parameters from visualization in application
   * structure 'parm' */
  lvisit_trace_parm_recv(&parm)

  /* send 3d vector data 'velo' (which is distributed) to
   * visualization */
  lvisit_trace_velo_send(&velo,nx,ny,nz,3); }

/* close the connection */
lvisit_trace_close()
```

Here an excerpt of the corresponding definition in the specification-language:

```
...
dataset velo:
  datatype     = double
  direction    = sim -> vis
  dimension    = 3
  veclen       = 3
  compression  = wavelet
  distribution = domain
...
```

13. References:

List references for further reading.

http://www.fz-juelich.de/zam/visit

J.Brooke, T.Eickermann, and U.Woessner: Application Steering in a Collaborative Enviroment, Proceedings of the ACME/IEEE SC 2003 Conference, Phoenix, 2003.

UC 23: Visualization Service for the Grid

Name of use case: Visualization Service for the Grid

Contact: Pascal Kleijer <k-pasukaru@ap.jp.nec.com>

Eiichi Nakano <e-nakano@ax.jp.nec.com>
NEC, HPC Marketing Promotion Division

Arihiro Yoshida <yoshida@grid.nii.ac.jp>
NII, National Institute of Informatics

Authors: Pascal Kleijer

1. General Information:

This section consists of check-boxes to provide some context in which to evaluate the use case.

1.1 Which best describes your organisation:

Industry	[X]
Academic	[X]
Other	[ ]
Please specify:	.....

1.2 Application area:

Astronomy	[ ]
Particle physics	[ ]
Bio-informatics	[ ]
Environmental Sc.	[ ]
Image analysis	[ ]
Other	[X]
Please specify:	Massive Data Visualization: - Real-time - Post-Processing - Database Post Visualization Access

1.3 Which of the following apply to or best describe this use

case Multiple selections are possible, please prioritize
with numbers from 1 (low) to 5 (high):

Database	[ ]
Remote steering	[4]
Visualization	[5]
Security	[1]
Resource discovery	[ ]
Resource scheduling	[ ]
Workflow	[ ]
Data movement	[3]
High Throughput Computing	[1]
High Performance Computing	[5]
Other	[ ]
Please specify:	.....

1.4 Are you an:

Application user	[X]
Application developer	[X]
System administrator	[ ]
Service developer	[X]
Computer science researcher	[ ]
Other	[ ]
Please specify:	.....

2. Introduction:

2.1 Provide a paragraph introduction to your use case .
Background to the project is another alternative.
(E.g. 100 words).

This project lies within the NAREGI (National Research Grid Initiative) framework. Our goal is to provide a Grid Service Framework for Massive Data Visualization of various simulations' types, including coupled simulations.

Most simulations in any scientific fields can never accomplish their purpose without visualization (in concurrent or in post-processing), which enables researchers to observe and analyze their unrecognizable numerical results.

This project ambitions the realization of an API for the grid services infrastructure. This API serves to connect two remote systems client and server by the grid medium and offers the client a set of tools to perform visualization of large-scale simulations. The visualization is based on images transferred from the server to the client and visualization controls sent by the client to the server.

Additional features such as Tracking/Steering is also in the possible range of inclusion.

2.2 Is there a URL with more information about the project ?

http://www.naregi.org
http://www.cs.vu.nl/ggf/apps-rg/meetings/ggf12/kleijer-final.pdf
http://www.cs.vu.nl/ggf/apps-rg/meetings/ggf12/kleijer-slides.pdf

3. Use Case to Motivate Functionality Within a Simple API:

Provide a scenario description to explain customers' needs.
E.g. "move a file from A to B," "start a job."

Please include figures if possible.

If your use case requires multiple components of functionality, please provide separate descriptions for each component, bullet points of 50 words per functionality are acceptable.

To help understand the following points a small glossary is added:

Server: A visualization capable solver, data-mining program, third party application, etc. running on a grid resource.
Service provider: A grid service or any program that interface the client and server. The provider can run on a different resource as both server and client, it can also be part of the server application.
Client: The visualization application running on the user's local computer.

The functionalities can be applied to one or more of the following target categories: {common, post-processing, concurrent, database}.

- Locate a resource to host a simulation (with visualization capability). This should be done automatically based on the nature of the simulation the user is preparing. [common]
- Launch the simulation server, including visualization service, on the resource allocated. [common]
- Send visualization control commands and receive a sequence of images back (streaming capability). [common]
- Replay a sequence of image (eventually video) stored on a remote resource. No simulation is running, only post-processing output is read. [post-processing, database]
- Steer the simulation/application (pause, stop, resume, etc). [concurent]
- Plug-in or Plug-out from a running simulation without affecting it. [concurent]
- Real-time tracking of simulation parameters. [concurent]
- Database access to retrieve a stored result. [database]
- File transfer. [common]

4. Customers:

Describe customers of this use case and their needs. In particular, where and how the use case occurs "in nature" and for whom it occurs. E.g. max 40 words

We can categorize different customer types:

- Post-Processing: They want to visualize and manipulate massive amount of data on site to avoid transfer between systems.
- Concurrent: They want to visualize an on going massive data computation. No need for data output, the simulation results is displayed on the fly.
- Repository Access: They want to access images, videos, or processed simulation results in a repository (database, file system, etc.)
- Application Service: Some applications like PyMol can be made available on high performance systems to perform visualization jobs remotely.

In each case, they are looking toward the Grid for its potential without the hassle of knowing how to operate of the Grid.

5. Involved Resources:

5.1 List all the resources needed: e.g. what hardware, data, software might be involved.

Hardware: High-end computers for simulations and visualization The hardware does not need High-end rendering capability like a SGI machine. Any HPC system is acceptable. Client can run on any piece of hardware.
Data: No specific data requirements
Software: Now Unicore and/or Globus as middleware. Later, any OGSA or WSRF implementation will do.

5.2 Are these resources geographically distributed?

Yes. The resources can span from a local machine to a fully earth's spread system.

5.3 How many resources are involved in the use case? E.g. how many remote tasks are executing at the same time?

From 1 to N, where N is the maximum capacity the grid can offer. In the case of coupled simulations the interactions between the different tasks can be quite complex.

5.4 Describe your codes and tools: what sort of license is available, e.g. open or closed source license; what sort of third party tools and libraries do you use, and what is their availablility; do you regularly work from source code, or use pre-compiled applications; what languages are your applications developed in (if relevant), e.g. Fortran, C, C++, Java, Perl, or Python.

All code produced within the project will be open-sourced at the finalization of the project. All third party code used must have similar license agreements.

Languages: FORTRAN & C for the visualization side. Java for the client side.

5.5 What information sources do you require, e.g. certificate authorities, or registries.

A registry is required that allows the components of the system to find each other. We would like to use registries such as GIIS or UDDI for resource discovery, but own registry usage is also possible. This will depend on the deployment environment.

5.6 Do you use any resources other than traditional compute or data resources, e.g. telescopes, microscopes, medical imaging instruments.

In general case no, but that may depend on the application developed.

5.7 Please link all the above back to the functionalities described in the use case section where possible.

N/A

5.8 How often is your application used on the grid or grid-like systems?

[X] Exclusively
[ ] Often (say 50-50)
[ ] Ocassionally on the grid, but mostly stand-alone
[ ] Not at all yet, but the plan is to.

The goal of this project is to offer the Grid capability to other applications or projects. We are therefore completely committed to the Grid or Grid-like systems.

6. Environment:

Provide a description of the environment your scenario runs in, for example the languages used, the tool-sets used, and the user environments (e.g. shell, scripting language, or portal).

Simulation servers are generally written in FORTRAN or C. The visualization library has bindings in FORTRAN and C. This code runs, for example, on OSs like Super-UX, AIX, or *NIX systems.

The Grid Service wrappers can be written in Java, C/C++ depending on the application wrapped. Most likely target will be the same as the simulation/visualization servers.

To avoid the hassle of cross-platform compatibility the client sides are targeted to be all Java based.

7. How the resources are selected:

7.1 Which resources are selected by users, which are inherent in the application, and which are chosen by system administrators, or by other means? E.g. who is specifying the architecture and memory to run the remote tasks?

The resources are decided by the problem the user wants to solve. The application will request the registry. The user may have to choose between different available resources if they are available.

7.2 How are the resources selected? E.g. by OS, by CPU power, by memory, don't care, by cost, frequency of availability of information, size of datasets?

OS, Architecture, Memory, disk space, CPU power, software availability (if the service is pre-installed).

7.3 Are the resource requirements dynamic or static?

For a given request the resource is in general static, but this also depends on how the server side is setup.

8. Security Considerations:

8.1 What things are sensitive in this scenario: executable code, data, computer hardware? I.e. at what level are security measures used to determine access, if any?

The information transmitted on the grid by each peer is on most cases highly sensitive. The middleware (communication layer) must ensure tight security.

8.2 Do you have any existing security framework, e.g. Kerberos 5, Unicore, GSI, SSH, smartcards?

For the moment we exclusively relay on the UNICORE or Globus kits.

8.3 What are your security needs: authentication, authorisation, message protection, data protection, anonymisation, audit trail, or others?

Setup phase: authentication, authorization Runtime: message protection

8.4 What are the most important issues which would simplify your security solution? Simple API, simple deployment, integration with commodity technologies.

Middleware should take care of the security issues. The application should just specify that high security connection is necessary.

9. Scalability:

What are the things which are important to scalability and to what scale - compute resources, data, networks ?

Large-scale simulations are generally limited by the computation resource available. The scaling would take place

on the number of CPU allocated. Another scalable factor would be the image size to control the data size.

10. Performance Considerations:

Explain any relevant performance considerations of the use case.

The computational resource is the most critical issue. In visualization we must expect at least 1 frame/second for rendering (without the simulation computation time, if any)

The second important matter is the network latency, the faster the response the better. Network capacity is dependent on the resolution of the image the user wants, not on the size of the problem to simulate.

11. Grid Technologies currently used:

If you are currently using or developing this scenario, which grid technologies are you using or considering?

Using: WSRF. XML, Unicore, Globus.

Future: SAOP, GridFTP, MPICH-G

12. What Would You Like an API to Look Like?

Suggest some functions and their prototypes which you would like in an API which would support your scenario.

We are currently defining a general visualization framework API. From the client point of view the following code can be considered as an example:

Post-Processing Service:
****

```
// Generate an instance of Post visualization service.
VisualizationService pvs = VisServiceFactory.newInstance
                           (POST_PROCESSING_SERVICE, ...);

// Set data file which contents is visualized
pvs.setDataFile ("gsiftp://localhost:2811/tmp/file1");

// Setting parameter
pvs.setCamera (...);
pvs.setLight  (...);

// Generate a visualized image
byte[] imageData = pvs.getImage();


Concurrent Service:
*******************
// Generate an instance of Concurrent service.
VisualizationService cvs = VisServiceFactory.newInstance
                           (CONCURRENT_SERVICE, ...);

// Setting initial parameter
cvs.setCamera          (...);
cvs.setLight           (...);
cvs.setBackgroundColor (...);
cvs.setImageSize       (...);

// Get the visulization streeming. Images or image sections
// are streamed on an input stream.
InputStream vizInput = cvs.getInputStream ();

// Visualize
while (...)
{
    // Process the content of the stream.
    ...

    // Set new parameters.
    ...
}


In these examples we consider that authentication, job
launching, etc is done prior to get a visualization service.
```

13. References:

List references for further reading.

a) Pascal Kleijer, Eiichi Nakano, Toshifumi Takei, Hiroshi Takahara, Arihiro Yoshida: "API for Grid Based Visualization Systems" GGF 12 Workshop on Grid Application Programming Interfaces.

b) http://www.naregi.org

Summary

This document collects the use cases received by the Simple API for Grid Applications research group (SAGA-RG) and presents them for the purpose of reference in future documents.

Seeking and collating use cases represented just the initial efforts in the overall design and implementation of a SAGA API. Significant effort was expended in analysing the use cases received. For example, the three functional areas most requested by the use cases were, job management, resource management and data management. Similarly the three most important non-functional requirements that seem to emanate from the analysis were error handling, security and auditing. This helped define the functional areas, scope and focus of the design team for an initial version of the SAGA API. Detailed results of the analysis is available as a GGF document, "A Requirements Analysis for a Simple API for Grid Applications" (GFD.71).

Security Considerations

This document is informational, and contains a set of use cases. As such, it does not address security considerations directly. Security, however, is discussed in several use cases, and several security requirements to Grid APIs are explicitely listed.

Contributors

Most significant acknowledgments are to be given to the many use case contributors, without whom, this document would obviously not have been possible. We thank members of the SAGA Research Group and many others for providing useful feedback and discussion.

Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be

claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

Disclaimer

Full Copyright Notice