# Can We Connect Existing Production Grids into a World Wide Grid?

Peter Kacsuk[1], Attila Kertesz[1] and Tamas Kiss[2]

[1] MTA SZTAKI
1111 Budapest, Kende u. 13
{kacsuk, attila.kertesz}@sztaki.hu
[2] Centre for Parallel Computing, University of Westminster
115 New Cavendish Street, London, W1W 6UW
kisst@wmin.ac.uk

**Abstract.** The World Wide Web has become a phenomenon that now influences our everyday life in any possible areas and disciplines. This paper investigates how a grid equivalent of the WWW, the World Wide Grid can be created. We define requirements towards a workflow-oriented computational World Wide Grid and propose a solution how current production Grids can be connected in order to form the technical basis of this infrastructure. A meta-broker concept and its utilization to achieve the highest level of parallelism by the created architecture in a user transparent way are explained.

**Keywords:** Grid interoperability, meta-broker, parallelism, grid workflow, World Wide Grid.

## 1 Introduction

The goal of this paper is to assess where we are in the road of establishing a scientific, workflow-oriented, computational World Wide Grid (WWG) that is similar to the World Wide Web (WWW) in the sense that anyone can access and use its services according to his needs. If we look at the current trend of how and where grid develops we can see that isolated production grids have been created that are based on different grid technologies that are not interoperable or only in a limited way. One of the biggest challenges of the grid community is to solve the interoperability of these production grids in order to get closer to the idea of WWG.

This paper is intended to show that we are not far from creating a WWG if we make reasonable assumptions how the users would like to use a WWG and reasonable restrictions how such a WWG can be used and for what purposes. The basic assumptions are as follows:

1. We restrict ourselves to a computational grid type WWG
2. The goal of the user is to dynamically collect and use as many grid resources as possible to accelerate his grid application

3. The basic application type the user would run in the WWG is a complex workflow
4. The WWG will be used in the beginning by the scientific community

Assumption 1 says that we restrict the first version of the WWG to a computational grid where the size of files used in the WWG is not too large and hence they can efficiently be moved between computational grid resources. It does not exclude the usage of large files in the WWG but it significantly reduces the overall performance of the WWG if many users try to use large files. Obviously, in a longer term the efficient management of large files should also be solved in the WWG but since our goal is to show that a computational WWG is already feasible we neglect this problem in this paper. A follow-up paper will be written to outline the possible solutions for a data-oriented WWG.

Assumption 2 requires that all possible parallelisms of an application should be exploited in the WWG. Users go for the WWG to access and use many resources in parallel to speed up the execution of their complex applications. Section 2 will classify the types of parallelisms that can be achieved in the WWG and later it will be shown how such parallelisms can actually be utilized in the WWG.

Assumption 3 might require some more explanations then the first two assumptions. Why workflow applications are so interesting for the WWG? The workflow concept abstracts a collection of services (tasks) that can be executed in a partially ordered way and hence it is general enough to include as a special case any other types of applications. As a result, workflows could be considered as the most generic type of applications including any other types as special cases.

Assumption 4 is based on the current usage scenario of large e-science grids. In order to establish a WWG that is used by the general public or by businesses, it requires a significant improvement in the development of a grid market model. If only the scientific community will use the first WWG, then the grid market model could be much simpler than a real commercial one. Since our interest is to establish a WWG as soon as possible, it is better to start with a scientific WWG and later extend it to other directions.

At this point many readers could say that these assumptions are too restrictive and it is not worth defining and creating a WWG that can support only these kinds of applications and goals. We would argue that we cannot create at once the ultimate WWG. The WWW is much more complex today as it was in its initial stages and in the beginning it was used only for scientific purposes and only later it was extended towards the commercial world. Once we established an infrastructure that is useful and works, there are plenty of opportunities afterwards to improve and extend that system in the future. Even this restricted version of the WWG that is suggested in this paper can be used to support much more applications than we can dream today. The establishment of such a WWG could tremendously widen the user community of the grid and would significantly accelerate the take-up of grid technology world-wide and would lead later to a WWG usable for the whole population including commercial services as well.

In this paper we identify the main steps of creating a WWG system according to the assumptions mentioned above, and describe in detail the first of these steps that provides the basic interconnection and access mechanism of such a WWG system.

Section 2 analyses the reasons of the success of the WWW and compares the WWW concept with a potential WWG concept. Section 3 classifies the achievable types of parallelisms in a WWG system. Section 4 introduces the concept of meta-broker and shows how the various types of parallelisms can be exploited in the WWG by using the meta-broker concept. Section 5 gives a short overview of related research.

## 2 Comparison of WWW and WWG

Before starting to explain the technical details of creating such a WWG, it is important to compare the WWW concept and the proposed WWG concept. First of all, let's see why the WWW concept is so popular and successful. There are five main aspects that make WWW so attractive:

1. Services
2. User interface
3. Web search engines
4. Security
5. Interest to use

The original WWW concept was based on the web page services. The idea is that anyone can create and publish web pages and anyone from any client machine can access these published web pages. Another important concept here is that web pages can be linked together and this significantly facilitates the creation and usage of web pages.

The second appealing feature of the WWW is that its user interface is extremely simple and easy-to-use. A simple browser is sufficient to browse web pages no matter where these web pages were created. More than that these browsers are provided as part of the basic services of the client machines' operating systems and hence the user does not have to install any complicated WWW access software, it comes together with the client machine. Web portals help the users to access structured information over a set of web pages.

Web search engines [1][2][3] help the users to discover information in the Web. In fact the Web search engines provide the web information system by discovering relevant web page contents.

The HTTP and HTTPS protocols provide the necessary security mechanism. They require only a single port to be opened on the server machine and hence they can be securely managed. The security concept of the WWW is so reliable that even large banks trust this system and provide financial services through their web portals.

The final aspect of the WWW is the motivation of people to use it or to provide services by it. The WWW is an excellent way of creating communities, accessing information and special (e.g. financial) services and hence people are interested in using the WWW services. At the beginning when commercial exploitation of the WWW was not so apparent, people were interested in creating web pages because in this way they could increase their or their company's visibility. Later when it became clear that there are several business models by which companies can make profit by providing WWW services, the usage of the WWW became even more popular.

After the overview of the main aspects of the WWW let's see how these aspects can also be used to promote the WWG as a success story. In case of the WWG the services are grid resources (computing services, data services, etc.) or higher level grid services (for example, a workflow execution service). It is important that anyone should be able to provide grid services and anyone should be able to access these grid services from any client machine through the WWG. The same way as web pages can be linked together grid services should be linked together, too.

The user interface should be extremely simple and user-friendly, exactly the same way as in case of the WWW. Simple tools like a web browser should be available on every client machine in order to access the WWG services without installing any grid software on the users' machines. However, in the WWG the main objective is to run applications and hence instead of a simple browser, rather a simple application builder is the right GUI. Grid portals should help users to access the WWG services in a coordinated way releasing the user from the actual organization of resource collection and orchestration via the WWG.

Grid search engines similar to the Web search engines should help both users and grid services to discover relevant services of the WWG.

The current grid security mechanism is built on the concept of grid certificates and VOs. Although scientific papers always emphasize the importance of creating dynamic VOs, in practice VOs are quite static and their creation is a long procedure. This static nature of VOs is one of the reasons why the grid has been developed towards the isolated grids direction and not towards the WWG direction. The current certificate and VO scheme make the usage of the grid much more complicated than the usage of the Web. As a consequence in this respect some revision is necessary if we want to make the WWG really easy-to-use and popular.

Finally, the motivation of the usage of WWG should be made tempting for large user and grid service provider communities. Once the usage of WWG is simple enough it will be really attractive for a large user community to access grid resources and services in an "unlimited" way. This will raise a new problem. If there is no limit of accessing resources and services the whole WWG will collapse due to the huge demand of resources and services. A WWG market model is therefore unavoidable and obligatory from the very beginning in order to attract resource and service providers and to restrict the eagerness of WWG consumers in acquiring resources and services for their applications. A kind of WWG credit system must be introduced where resource and service providers can earn WWG credits and then their community can use these credits to acquire WWG resources and services.

Table 1 summarizes and compares the five main features of the existing WWW and a potential WWG.

If there are so many similarities in the concept of WWW and WWG, then why are we still missing the WWG as a working infrastructure and service? Unfortunately, there are some problems concerning all the five required features of the WWG.

**Table 1.** Comparison of WWW and WWG

| | WWW | WWG |
|---|---|---|
| Services<br>• Anyone can create and publish<br>• anyone can access | Web pages, web services | Grid resources, computing services, data services, etc. |
| User interface | Web browsers<br>Web portals | Grid application builder<br>Grid portals |
| Information discovery mechanism | Web search engines | Grid search engines |
| Security | HTTP and HTTPS protocols | Revised dynamic VO concept |
| Interest to join | User: access information and services<br>Provider: Increase own visibility, make money | User: use grid resources (by grid credit)<br>Provider: collect grid credits (later make money) |

**Services and resources**
There are many different production grids based on different grid technologies and middleware and they can not interoperate. As a result the services are not accessible by anyone from anywhere as it would be needed by the WWG concept proposed above. If a user is registered for the VOx of GridA then he cannot use the resources and services of VOy of GridB. Section 4 of this paper will show that based on assumptions 1-3 we can easily solve this problem and create a WWG that satisfies the required criteria.
**User interface**
The grid user interface is currently too complicated. In most cases production grids neglect the problem of user interface. They provide only a special command line interface and programming API. It means that there is no user interface standard like the web browser in case of the Web. Different grid middleware requires the usage of different command line interfaces and programming APIs. It means that a user who wants to use several grids (an obvious assumption of the WWG concept) has to learn several grid user interfaces. If the user wants to port a grid application from GridA to GridB he has to re-engineer the application according to the programming API of GridB.
**Information discovery mechanism**
The grid information system and discovery mechanism are not mature enough and the concept of a grid search engine is also missing from current grids. The usage of the information system is very limited in current production grids.

**Security**
The grid security mechanism is suitable for the rather static VO concept of current production grids but not for the WWG where VOs should be formulated dynamically on demand. As a consequence the concept of VOs should be revised in the framework of the WWG. However, the subject is so big and significant that an independent paper should deal with this issue. One solution could be that only pre-registered and verified applications can be used in the WWG that are taken from certified application registries. Another solution could be the usage of virtualization where the application is distributed inside a virtual machine that can not cause any harm to the executor resource.
**Interest to join**
Since the current usage of grid lacks the market concept (everyone can get what he needs without payment) a WWG would lead to the tremendous overload of resources. At least, the introduction of a simplified grid market concept would be necessary to establish a scientific WWG.

The following sections of this paper describe in detail the first of these features and show possible solutions that can be used in order to establish a scientific computational WWG. The analysis and solution of the other listed aspects will follow in future publications. Obviously, when the goal is to create a WWG as soon as possible any proposed solution should be built according to the current situation. It should be accepted as a fact that production grids already exist and they are not willing to give up their freedom of developing their own roadmap and direction. Therefore a WWG concept should be built on the autonomicity and collaboration of these existing production grids.

## 3 Parallelism in the WWG

As we have seen in the Introduction, assumption 2 says that the goal of the user is to

a. dynamically collect and use as many grid resources as possible
b. in order to accelerate his grid application.

Condition "a" means that the user needs a WWG where resources can be accessed from any production grid that are connected to the WWG no matter what type of grid middleware they are built on. As a result current production grids should be used in an interoperable way even if they are based on different grid middlewares. When this problem is solved any user from any grid can access all the grid resources and services that are connected to the WWG.

Condition "b" requires that both the application and the WWG should be able to support the largest possible classes of parallel execution.

In order to fulfil our requirements, first we examine the impact of parallelism on the grid middleware, and then investigate the problem of grid interoperability. We shall see that both conditions can be fulfilled by the introduction of interoperable grid brokers or by the introduction of a meta-broker.

There are two classes of parallelisms achievable in the WWG:

1. Grid architecture parallelism
2. Grid application parallelism

The grid architecture parallelism can be further divided into two classes according to the usage of various grid resources:

– **Inter-Resource (IrR) parallelism** (parallel usage of several resources) within which we can distinguish:
  – **Inter-Grid (IrG) parallelism** (parallel usage of several resources within several Grids)
  – **Intra-Grid (IaG) parallelism** (parallel usage of several resources in the same grid)
– **Intra-Resource (IaR) parallelism** (usage of parallelism in a single resource having parallel architecture, e.g. cluster)

The current grids typically enable the exploitation of the Intra-Grid and Intra-Resource parallelism but they do not support Inter-Grid parallelism. Even worse, within the same grid, users are restricted to use the resources of a certain VO only where they are accepted as members. The current concept of VOs is strongly against the nature of WWG.

We can distinguish four types of grid application parallelism according to the granularity of tasks to be solved in the grid:

– Single job/service level (SJ)
– Parameter Sweep at job/service level (PSJ)
– Workflow level (WF)
– Parameter Sweep at workflow level (PSWF)

Intra-Resource parallelism can be applied to any types of grid application parallelism if the resource is a multi-processor one (either shared or distributed memory system or even multi-core processor) and the local job manager is able to distribute the parallel components of the application among the processors of the resource.

**Single job level parallelism** (SJ) can be exploited if the application consists of one job and this job is a parallel (e.g. MPI) one. In this case we can explore **process parallelism** that comes from the parallel execution of processes of the parallel application. Although there are some research projects aiming at the exploitation of Inter-Resource parallelism, SJ parallelism still best fits to the Intra-Resource parallelism where processes of a parallel application can be distributed among the nodes, processors or cores of a parallel resource. If there are N processes inside a parallel job, then the achievable parallelism is O(N).

**PS job level parallelism** (PSJ) can be exploited if the application consists of one job and this job should be executed with many different parameter sets (this is called **job instance parallelism**). PSJ parallelism fits both to Intra-Resource and Inter-Resource parallelism no matter whether the job is a sequential or parallel one. In the case of a sequential job Intra-Resource parallelism can be exploited by allocating many instances of the same job to a multiprocessor resource and the different job instances are simultaneously executed by different processors of the resource. If a sequential job is to be executed with M parameters, then the achievable parallelism is O(M). If the job is a parallel application with N processes and this should be executed with M parameters, then both process parallelism and job instance parallelism can be exploited and the achievable parallelism is O(MxN). Since both N and M can be in the range of hundreds or thousands, PSJ parallelism can require several thousands of resources and hence it really needs the large number of resources available in the WWG.

**Workflow level parallelism** (WF) can be exploited if there are parallel branches in the workflow. This is called **workflow branch parallelism** and it fits both to Intra-Resource and Inter-Resource parallelism. However, the amount of parallelism is typically not as big as in case of the PSJ parallelism so in many cases Intra-Grid parallelism is enough to handle it. If some of the jobs in the workflow are parallel (e.g. MPI) jobs, then two levels of parallelism can be exploited simultaneously: process parallelism and workflow branch parallelism. Different MPI jobs of the different branches of the workflow can be simultaneously executed on different resources (Inter-Resource parallelism) and on each such resource Intra-Resource parallelism can be applied for the parallel execution of the processes of the MPI jobs. If the maximum number of parallel branches in the workflow is B and in every branch there is a parallel application with N processes, then the achievable parallelism is O(BxN).

**PS workflow level parallelism** (PSWF) can be exploited if the application consists of a workflow and this workflow should be executed with many different parameter sets (this is called **workflow instance parallelism**). In such case three levels of application parallelism can be exploited:

- Workflow instance parallelism (among the several instances of the workflow)
- Workflow branch parallelism (among the branches of every workflow instance)
- Process parallelism (among the processes of a parallel node of a workflow instance)

Notice that job instance parallelism is a special case of workflow instance parallelism when the workflow consists of a single job. From another point of view workflow instance parallelism is a sum of achievable job instance parallelism when the component jobs of a workflow are executed with many parameters sets.

In order to exploit these three levels of parallelism, PSWF parallelism can require thousands or even millions of resources and hence it really needs the large number of resources available in the WWG. PSWF parallelism fits to the Inter-Resource parallelism (both IaG and IrG) and as a result can advantageously be used in the WWG. If the maximum number of parallel branches in the workflow is B and in every branch there is a parallel application with N processes, and the workflow should be executed with M different parameter sets, then the achievable parallelism is O(MxBxN). This is clearly the most demanding type of parallel application concerning the number of required grid resources.

## 4 Resource selection to achieve the highest possible parallelism

After seeing that many resources should be used in a PSJ or PSWF application the next question is how to select the required resources in order to achieve the highest possible parallelism. In a WWG system four models can be distinguished:

- User selected Grid User selected Resource (UGUR)
- User selected Grid Broker selected Resource (UGBR)
- Broker selected Grid User selected Resource (BGUR)
- Broker selected Grid Broker selected Resource (BGBR)

More and more production grids use brokers nowadays inside their own boundaries in order to select the most appropriate resources for execution. However, even if the user is allowed to submit jobs to several grids, it is his responsibility to select between these grids. Brokering at Grid level is not supported in today's production environments. As a result, the BGBR model, that provides the largest freedom of accessing grid resources in the WWG, is hard to realize. The BGBR model means that the users can access several grids simultaneously and these grids have got brokers. However, the user is not connected directly to a grid broker rather to a new level of brokers called as meta-broker. It is the task of the meta-broker to select the right grid according to the users' requirements as described in the Broker Property Description Language (BPDL) [4]. BPDL is similar to the JSDL [6], but it provides metadata about brokers (grids) and not about resources. Once the grid is selected the meta-broker passes the job and the job description language (RSL [7], JDL [8] or JSDL depending on the actual grid) to the selected grid broker and it will be the task of this broker to select the best resource according to the requirements specified in the job description language (or sometimes called resource specification language). The architecture of the BGBR model can be seen in Fig. 1.

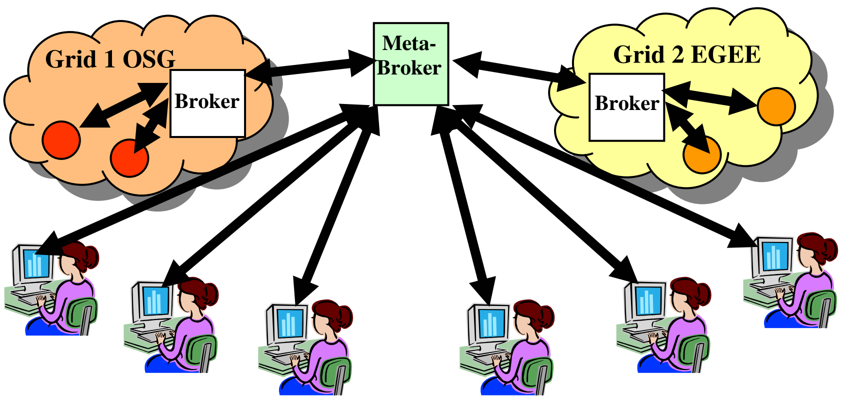


**Fig. 1.** Grids connected by a meta-broker.

The proposed Meta-Broker architecture is described in detail in [5]. In order to make the concept of meta-broker clear a short overview of its architecture concept is given here. Fig. 2 introduces the proposed architecture of the Grid Meta-Broker that enables the users to access resources of different grids through their own brokers. The Translator components are responsible for translating the resource specification language of the user (JSDL) to the language of the selected resource broker. Once a broker will be capable of supporting the JSDL standard [6], the corresponding Translator can be removed from the Meta-Broker. The Invokers are broker-specific components. They communicate with the interconnected brokers, invoking them with job requests and collecting the results. Data handling is also an important task of this component. After the user uploaded the job, proxy and input files to the Meta-Broker, the Matchmaker component tries to find a proper broker for the request. If no good broker was found, the request is rejected, otherwise the JSDL is translated to the language of the selected broker. The responsible Invoker takes care of transferring the necessary files to the selected grid environment. After job submission it stages the output files back, and upgrades the historical data stored in the Information Collector with the logging data of the utilized broker. The core component of the Meta-Broker is responsible for managing the communication (information and data exchange) among the other components. The communication to the outer world is also done by this part through its web-service interface.

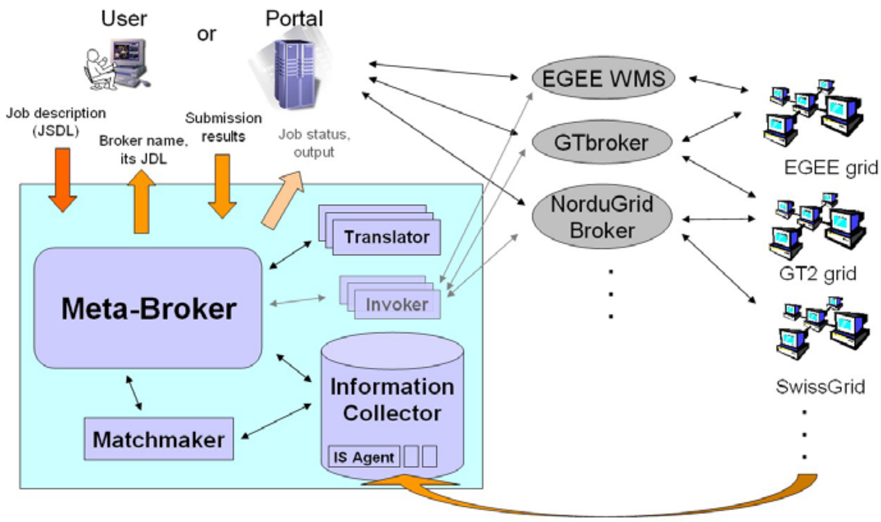


**Fig. 2.** The Grid Meta-Broker Architecture.

In the following we examine how SJ, PSJ, WF and PSWF parallelism can be exploited by the BGBR model and by a meta-broker.

**SJ level parallelism:**

In the BGBR case both the grids and the resources are selected by brokers and hence load-balancing can be achieved between grids and inside grids.

**PSJ level parallelism:**
Both grid and resource selection is done by a broker and hence both Inter-Grid (IrG) and Intra-Grid (IaG) parallelism are automatically provided by brokers. In principle all resources of all connected grids can be used in parallel and in a balanced way.
**WF level parallelism:**
Grid selection is the task of the broker and hence it is the broker's responsibility to explore the Inter-Grid (IrG) parallelism through the workflow branch-parallelism. Resource selection is also the task of the broker and hence it is the broker's responsibility to explore Intra-Grid (IaG) parallelism through workflow branch-parallelism and to explore Intra-Resource (IaR) parallelism by assigning parallel jobs of the workflow to parallel grid resources. Since both the grid and the resources are selected by brokers, load balancing can be achieved between grids and inside grids. However, workflow scheduler support is needed for the broker to exploit workflow branch-parallelism at IrG and IaG level.
**PSWF level parallelism:**
Both grid and resource selections are the tasks of the broker and hence both Inter-Grid (IrG) and Intra-Grid (IaG) parallelism can be exploited both through the workflow branch-parallelism and workflow instance-parallelism by the broker. Intra-Resource (IaR) parallelism can also be achieved as described in Section 3.

Advantages of the BGBR model are:

- Workflow instance-parallelism can be exploited both at IrG and IaG level.
- Workflow level scheduling support is not needed for the broker either at IrG or IaG level to achieve workflow-instance parallelism.
- Balanced usage of grids and resources are provided by the broker.

As a summary one can say that the most advantageous model to exploit every possible parallelism in a well balanced way is the BGBR model. So if we want to establish an efficient and powerful WWG it should be based on the concept of the BGBR model. Unfortunately, current grid middleware do not support the BGBR model. There is an on-going effort in the GIN VO of OGF to solve grid interoperability, but they concentrate at the moment on the SJ level. Within the framework of the CoreGrid project SZTAKI and UPC work on to design a meta-broker that can efficiently support the BGBR model [9].

Naturally, a single meta-broker would be a bottleneck in the WWG used by many thousands of scientists. Therefore, many uniform meta-brokers should be applied and these should be interconnected in order to realize load-balancing among them. In such a WWG every client can be connected in a standard way to one of the uniform meta-brokers. As a result if we want to build the WWG, the only thing we have to do is to define and implement:

1. The functionality of the meta-brokers
2. The intercommunication protocol of meta-brokers
3. The communication protocol of clients and meta-brokers
4. The standard intercommunication protocol of meta-brokers and grid brokers

The solution for requirements 1 and 3 are already designed and described in [5]. Work is needed for requirements 2 and 4. Notice that the BGBR model can be used even if requirement 4 is not fulfilled.

## 5 Related Work

The most notable work related to grid interoperability is carried out within the framework of the GIN initiative [10] of the OGF. As written there: "The purpose of this group is to organize and manage a set of interoperation efforts among production grid projects interested in interoperating in support of applications that require resources in multiple grids." The GIN related web page of the UK NGS [11] writes: "Grid Interoperation Now (GIN) is a Community Group of the Open Grid Forum (OGF). It aims to organize and manage a set of interoperation efforts among production grid projects to support applications that require resources in multiple grids." Obviously the goal of the GIN is very close to the objectives of establishing a WWG although their ambitions do not go so far. The phase 1 tasks of the GIN VO is "to plan and implement interoperation in specific areas, initially data location and movement, authentication/authorization and identity management, job description and execution, and information services."

The GIN has created the GIN VO with resources from the major production grids: TeraGrid [20], UK NGS [11], EGEE [21], OSG [22] and NorduGrid [23]. All these grids allocated some grid sites to do experiments on their interoperability. In the framework of the GIN VO activity a GIN Resource testing portal [12] has been set up based on the P-GRADE/GEMLCA portal technology [19]. This portal enables the job submission (even workflow submission) to all these grid sites and hence can be used to constantly monitor their availability and usability in the GIN VO. Although the goals of the GIN and the WWG described in this paper have many similarities the concept of the GIN is quite different from the implementation concept of the WWG.

A major European effort in providing grid interoperability between gLite, UNICORE and Globus is the OMII-Europe project [13] that tries to establish interoperability at the level of five services (JSDL, OGSA-DAI, VOMS, RUS and GridSphere).

The World Wide Grid testbed [14] initiated by Monash University has the same name as we used in this paper but their WWG has a quite different meaning than our WWG. Their WWG is not about connecting the existing production grids in order to establish an interoperable WWG, rather it is a volunteer grid test-bed specifically intended to test the grid economy concept developed in the Grid Economy project [15]. Their recent paper on InterGrid [16] shows many similarities with our concept of connecting existing grids into a WWG. They introduce InterGrid Gateways to connect the different grids while we propose the usage of meta-brokers. They do not emphasize the importance of advance grid portals and workflow interoperability but they put more emphasis on grid economy.

The meta-broker concept for providing grid interoperability is quite new and was first proposed by SZTAKI [17] at the CoreGrid workshop organized in conjunction with EuroPar'2006. Since that time another CoreGrid partner, the Technical

University of Catalonia (UPC) has started to work on this concept. The redefinition of the Broker Property Description Language (BPDL) [9] is an on-going joint work between SZTAKI and UPC. The detailed architecture plan of the meta-broker is described in [5].

## 6 Conclusions

We have seen that the highest level of parallelism can be exploited in the WWG if workflows are executed as parameter sweep applications. In order to exploit the largest possible parallelism the most advanced architecture concept of the WWG is based on the BGBR model. In the BGBR model every client can be connected in a standard way to one of the uniform meta-brokers. As a result to build the WWG, the only thing we have to do is to define and implement:

1. The functionality of the meta-brokers
2. The intercommunication protocol of meta-brokers
3. The communication protocol of clients and meta-brokers
4. The standard intercommunication protocol of meta-brokers and grid brokers

Once these requirements are defined and implemented, any existing grid can be connected to the WWG provided that the broker of the given grid realizes the standard intercommunication protocol of meta-brokers and grid brokers. Even if requirement 4 is not fulfilled, those grids for which the Translator and Invoker module of the meta-broker is already available can be connected to the WWG. The meta-broker should be implemented as an open source grid service in order that any grid could extend it with the necessary Translator and Invoker module by which the given grid could be accessed by the meta-broker.

Overall, we can state that there is no real technical obstacle to create a scientific computational World Wide Grid where complex parameter sweep workflow applications could run and exploit the largest possible parallelism. It means that technically the WWG can be established by simply introducing the meta-broker concept and using a network of uniform meta-brokers to connect the existing production grids. Obviously, the meta-broker itself does not help in managing workflows, only to submit nodes (jobs or service requests) of the workflow in a grid transparent way. To assist the workflow execution in the WWG workflow managers are needed. Moreover these workflow managers also should be interoperable. This issue has been investigated in a CoreGrid technical report [18].

## 7 Acknowledgements

The research described in this paper was carried out under the FP6 Network of Excellence CoreGRID funded by the European Commission (Contract IST-2002-004265) and particularly in the work of two institutes of the CoreGrid project:

Resource Management and Scheduling Institute and Grid Systems, Tools and Environments Institute.

## References


1. Woogle web service search engine. http://data.cs.washington.edu/webService/
2. SAL. http://www.salcentral.com/salnet/srchsal.htm
3. eSigma's web service search engine. http://www.esigma.com/telos/discover/browseservices.html?cid=0,telos:categories:general:utilities
4. A. Kertesz, I. Rodero and F. Guim, Data Model for Describing Grid Resource Broker Capabilities, CoreGRID Workshop on Grid Middleware in conjunction with ISC'07 conference, Dresden, Germany, June 25-26, 2007.
5. A. Kertesz, P. Kacsuk, Meta-Broker for Future Generation Grids: A new approach for a high-level interoperable resource management, CoreGRID Workshop on Grid Middleware in conjunction with ISC'07 conference, Dresden, Germany, June 25-26, 2007.
6. A. Anjomshoaa, et al, Job Submission Description Language (JSDL) Specification, Version 1.0: http://www.gridforum.org/documents/GFD.56.pdf
7. GT 2.4: The Globus Resource Specification Language RSL v1.0: http://www.globus.org/toolkit/docs/2.4/gram/rsl_spec1.html
8. JDL Job description language. http://www.grid.org.tr/servisler/dokumanlar/DataGrid-JDL-HowTo.pdf
9. A. Kertesz, I. Rodero and F. Guim, Meta-Brokering requirements and research directions in state-of-the-art Grid Resource Management, Technical report, TR-0116, Institute on Resource Management and Scheduling, CoreGRID - Network of Excellence, November 2007.
10. GIN Project. https://forge.gridforum.org/projects/gin
11. UK NGS web page on GIN. http://wiki.ngs.ac.uk/index.php?title=GIN
12. GIN Resource testing portal. https://gin-portal.cpc.wmin.ac.uk:8080/gridsphere/gridsphere
13. OMII-Europe Project. http://www.omii-europe.com/
14. World Wide Grid testbed. http://www.gridbus.org/ecogrid/wwg/
15. Grid Economy project. http://www.gridbus.org/ecogrid/
16. M. D. Assuncao, R. Buyya and S. Venugopal, InterGrid: A Case for Internetworking Islands of Grids, Concurrency and Computation: Practice and Experience (CCPE), Online ISSN: 1532-0634; Print ISSN: 1532-0626, Wiley Press, New York, USA, Jul 16 2007.
17. A. Kertesz, P. Kacsuk: Grid Meta-Broker Architecture: Towards an Interoperable Grid Resource Brokering Service, CoreGRID Workshop on Grid Middleware in conjunction with Euro-Par 2006, Dresden, Germany, August 28-29, 2006, pp. 112-116, Springer-Verlag Berlin Heidelberg
18. P. Kacsuk, T. Kiss: Towards a scientific workflow-oriented computational World Wide Grid, CoreGrid Technical Report, TR-0115, Institute on Grid Systems, Tools and Environments, CoreGRID - Network of Excellence, December 2007.
19. P. Kacsuk and G. Sipos, Multi-Grid, Multi-User Workflows in the P-GRADE Grid Portal, Journal of Grid Computing, Vol. 3, Nos. 3-4, 2005, pp. 221-238.
20. The TeraGrid website, http://www.teragrid.org/
21. The gLite website, http://glite.web.cern.ch/glite/
22. The OSG website, http://www.opensciencegrid.org/
23. The NorduGrid website, http://www.nordugrid.org/