Chapter 1

TRADING GRID SERVICES WITHIN THE UK E-SCIENCE GRID

Steven Newhouse,[1] Jon MacLaren,[2] and Katarzyna Keahey[3]

[1] *London e-Science Centre, Imperial College London*
[2] *Manchester Computing, The University of Manchester*
[3] *Mathematics and Computer Science Division, Argonne National Laboratory*

Abstract

The Open Grid Services Architecture (OGSA) presents the Grid community with an opportunity to define standard service interfaces to enable the construction of an interoperable Grid infrastructure. The provision of this infrastructure has, to date, come from the donation of time and effort from the research community primarily for their own use. The growing involvement of industry and commerce in Grid activity is accelerating the need to find business models that support their participation. It is therefore essential that an economic infrastructure be incorporated into the OGSA to support economic transactions between service providers and their clients. This chapter describes current standardization efforts taking place with the Global Grid Forum and the implementation of such an architecture within the UK e-Science Programme through the Computational Markets project.

1. INTRODUCTION

The term computational Grid is an intended analogy to electrical power grids: a vision of computational power available on tap, without the user needing to really care about precisely where and how the power was generated. For this vision to become a reality, Grid users, or consumers, must be able to access appropriate computational power; similarly, resource providers must be able to receive payment for the use of their resources.

Resource brokering is the process of discovering suitable resources for the consumer's purpose. By definition, resource brokering is the act of an intermediary responding to the immediate needs of its consumers, while collating information from the resources it represents. The provision of a brokering service is predicated on the existence of an interoperable standards-driven infrastructure for representing resources and their corresponding services, as well as on standard payment protocols. Without these capabilities there is no economic incentive to provide a resource brokering service, since different resource infrastructures have to be abstracted within the broker and, without standard payment mechanisms, there is no generated revenue for the organization providing that service.

The recent moves within the Grid community through the Open Grid Services Architecture (OGSA) [FKNT02] to standardize on a framework specification as opposed to a service implementation has provided a generic mechanism for resource virtualization that will enable resource brokering. Due to the steady increase in Internet trading, or e-commerce, a number of reputable organizations already provide secure on-line payment services (e.g., WorldPay [Wor]).

With standardized schemes to describe electronic money and to virtualize the underlying resource as services through OGSA, the outstanding requirement is to provide standardized mechanisms to describe the protocols needed to set the cost of using the service. Currently, this requirement is the focus of the Grid Economic Services Architecture Working Group (GESA-WG) within the Global Grid Forum [GES] (of which we are the chairs).

This chapter outlines a set of motivating use cases for the provisioning of services either through direct invocation or through a resource broker. We then examine how the demands of such an infrastructure could be met by the emerging Open Grid Services Architecture by extending its standard Grid Services with interfaces to support economic activity. We also describe activity taking place within the U.K.'s e-Science Programme to build such an infrastructure using the OGSA.

2. ECONOMY-BASED GRIDS

The marketing of computational services for economic reward has been the subject of much research activity over the past decade as the availability and power of distributed computing resources have evolved. One example of early work in exploiting distributed computing infrastructures was Spawn, which demonstrated how different funding ratios could be used to guide resource allocation and usage [WHH+92]. The growth of Grid infrastructures, such as the Globus Toolkit® [FK97, GLO], UNICORE [UNI], and Condor [LLM88], has promoted further discussion as to how economic paradigms may be used not only as an approach to resource allocation but as a means for making money. For instance, Nimord/G has shown how historical execution times and heterogeneous resource costs can be used for the deadline scheduling of multiple tasks within a fixed budget [ASGH95].

The key to trading in the real world is a medium of exchange that is deemed to have value and goods whose value can be assessed for exchange. Bringing an economic model into Grid computing presents two opportunities: using an economic paradigm to drive effective resource utilization, and motivating service provisioning for real economic gain by third-party service providers.

3. MOTIVATING USE CASES

The availability of flexible charging mechanisms that are fully integrated into the Grid infrastructure presents many commercial opportunities for independent service suppliers. One of the many architectural possibilities offered by OGSA is that of service provisioning through hierarchical encapsulation of service workflow and offering the encapsulated service as a single service to the user. The infrastructure provided by OGSA, when coupled with an economic mechanism, offers considerable scope for new service-oriented markets. These have recently been explored in a series of use cases being developed within the Global Grid Forum's GESA-WG [GES].

3.1 Coordination Between Services

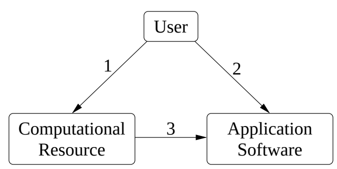

Figure 1.1. Coordinated use of application software on hardware.

Consider a simple scenario (shown in Figure 1.1) of a user wishing to use a commercial third-party application to analyze a self-generated dataset by using a computational resource. (We set aside for the moment the important factors that drive the selection of these services.) The user must obtain a quotation and reservation on the *computational resource provider* (1) before approaching the *application software provider*(2) to obtain a quotation for the use of that particular software on the computational resource. Once an acceptable quotation has been found from the compute and application providers—and this may be an iterative process because the cost of the software may depend on the class of computational resource and the time the data may take to process—the quotations and reservations are confirmed, and the computational resource may download and install the application software as required (3).

This process has already placed several requirements on the Grid infrastructure from both an economic and a general usage perspective. These requirements include a multiphase commitment to a resource reservation (one such approach using service-level agreements is described in Chapter **??**) and iterative negotiation to converge on an acceptable pricing for the resource reservation. Additional requirements such as authentication, authorization, and impersonation (of the user by the computational resource provider in order to retrieve the application software) should be met through the basic core middleware.

3.2 Service Aggregation

The process just described exposes the user to the potential complexity of negotiating and reserving resources between different service providers. Alternatively, an organization can provide this combined functionality directly to the user (see Figure 1.2). This form of resource broker can be described as an application service provider because it provides a complete service—running the user's data using application implementation on an arbitrary resource.

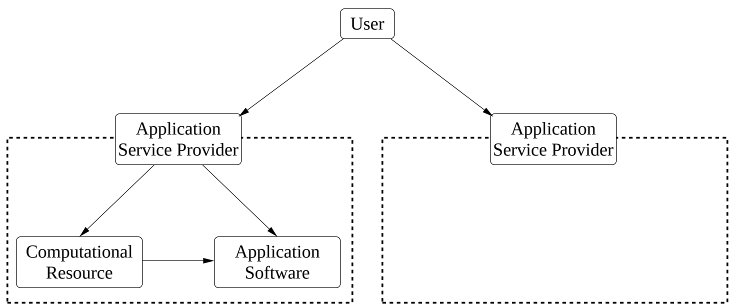

Figure 1.2. Service aggregation and virtualization.

Whereas previously the user was exposed to the full complexity of the underlying resource, in this scenario the application service provider had aggregated the services to supply a complete package. Two mechanisms can be used for providing this package. The application service provider can provide the computational infrastructure and application software through off-line purchases of the relevant equipment and software, as would normally be expected. In this case, the service provider has full control of the costs and can offer a service directly to the user. Alternatively, the application service provider can dynamically acquire these resources in much the same way as the user did in the earlier scenario.

A natural question is, "What are the economic benefits to the user?" The answer rests in part with the fact that the application service provider is able to derive potential economies of scale through the bulk purchase of computer resources and software licenses, by using the economic Grid infrastructure. These economies of scale can be passed on to the user as reduced costs, while the service provider still retains a profit margin for the service aggregation. Moreover, the service aggregator has the flexibility to switch suppliers as long it continues to deliver any contracted service levels. From the user's perspective, then, the service aggregator may be able to offer better pricing, faster discovery (since only a single aggregated service needs to be discovered, as opposed to several compatible services), and faster service delivery (as software may be pre-installed).

3.3 Service Brokering

In addition to these direct benefits, service aggregation can be viewed as a form of service (or resource) brokering that offers a convenience function—all the required services are grouped under one roof. But how does a user determine which of several application service providers should be selected for a particular application? The user could retain the right to select an application service provider service based on those that have been discovered from a registry service. Alternatively, this decision could be delegated to a service broker, which maintains an index of available application service providers.

The service broker is able to add value to its registry of application service providers by providing extra information about the services. This information may be as simple as cost, or it may include information about the reliability, trustworthiness, quality of service or service-level agreements, and possible compensation routes. Much like a financial consultant, the broker does not provide this added value service for free. Indeed, it may have a role in the financial transaction to provide an escrow account, acting as a trusted third party and holding the fee until the job is complete.

4. ARCHITECTURAL REQUIREMENTS

The preceding example of application service provision does not illustrate all of the features that may be required from an economic Grid services architecture. Indeed, many of the requirements from the scenario are a feature of a service-oriented architecture rather than that of an economic pricing mechanism. The emergence of the Open Grid Services Architecture from the Grid community is providing a service infrastructure upon which a variety of economic models may be developed and explored.

In this section we outline the basic mechanisms required to support such an infrastructure. We assume that economic models, dealing with issues such as price setting and Grid Services market creation, will be provided by other work in this area (see Chapter **??**). Our goal is to define an open infrastructure to enable the application of these pricing models to generic Grid Services.

4.1 Exploiting the Open Grid Services Architecture

The OGSA builds on the established Web Services infrastructure provided through the eXtensible Markup Language (XML) [XML], the Simple Object Access Protocol (SOAP) [BEK$^+$00], and the Web Services Description Language (WSDL) [CCMW01]. It provides an infrastructure to securely create, manage, interact with, and destroy transient Web Service instances within distributed hosting environments [FKNT02].

The Grid Service Specification defines the interface and the semantic behavior that must be supported by the Web Service for it to classed as a Grid Service [TCF+03]. This specification is under development and is being standardized within the Open Grid Services Infrastructure Working Group (OGSI-WG) of the Global Grid Forum [TCF+03].

A Grid Service has three features of interest in constructing an economic framework to trade resources:

- The *Grid Service Handle* (GSH) provides a unique identifier to a service instance running in a service environment.
- Each Grid Service has a *service data element*(SDE)—an XML document—that describes the internal state of the service. The Grid Service provides standard ports to support updating, searching, and so forth of the SDE by other entities.
- A Grid Service may support a *factory port* (or interface) that allows new service interfaces to be instantiated within the hosting environment.

The GESA-WG [GES] is analyzing the architectural requirements of an economic infrastructure within the context of the OGSA.

4.1.1 Grid Service Handle

The GSH is used by the client-side code to contact the specified service or factory instance. By assuming that the economic architecture is able to embed the cost of a transient Grid Service as one of the SDEs of a service factory (not an unreasonable assumption), the GSH effectively provides an identifier to a cost quotation for the use of the service. This price can also be advertised by other Grid advertising mechanisms; however, we assume here that the factory is a reliable source of such quotations. This service price quote may vary depending on factors such as the time the service will be performed, the time the quote is requested, the identity of the requestor, the level of Quality of Service (QoS) factors with which the service should be performed, and the guarantee with which those QoS representation can be delivered.

4.1.2 Service Data Elements

The application service provider scenario has illustrated that many of the issues relating to the selection of services within an economic architecture concern service rather than function:

- Does this service offer any bulk purchase discounts?
- Can I trust this service to deliver on its commitments?

- Is my data secure while it is residing on the remote server?
- Will I be compensated if anything goes wrong?

Such service metadata is encapsulated within the SDE structure provided by OGSA and may be collected from many service instances for presentation within an advertising service. This metadata may be static (extracted from the Grid Services Description Language document that defines the service interface) or dynamic (generated by the service or inserted from other authorized services). Standardization of the required and optional elements of this metadata that is one of the challenges now facing the community.

4.1.3 Factory Ports and Service Level Agreements

The factory model of service generation used within the OGSA provides a powerful abstraction to deal with pricing of Grid Services. We encapsulate the cost of using the service within the instance produced by the service factory (which can be referenced by the user through the GSH). This approach strengthens the link between the GSH acting as a quotation to the cost of invoking a service. Every quotation is created with an expiration time that puts time limits on its use.

The Chargeable Grid Service contains additional port types (to set prices, etc.), thereby extending the capability of the Grid Service being offered for sale. This approach enables existing client-side code to use the economic Grid Service without having to regenerate these interfaces.

In addition to providing quotes on service prices, the factory needs to support the negotiation for services with concrete QoS specifications as well as the creation of such services. Extending the concept of the OGSA factory to allow negotiation of service-level agreement (SLAs), as shown in [KM03], provides this capability. As a result of the negotiation process, concrete and well-defined SLAs are issued to concrete clients.

An SLA is a bilateral agreement [KKL+02] between the client and the service provider (represented by the factory) specifying the level of QoS with which the service will be provided (including the price or a pointer to price-generating mechanisms if the price should change during the lifetime of the contract), monitoring mechanisms that can determine whether the QoS requirements are met, and corrective actions to be taken if the requirements are not being met. Corrective actions may include adaptive scheduling, such as preemption of other executions, or QoS adjustments, such as price cuts or other kinds of compensation. Furthermore, the QoS conditions listed in the SLA should specify exhaustively the actions taken to provide overall QoS. For example, it is not enough to say that data will be secure; the composition of different security mechanisms used throughout the process should be specified. The SLA will usually be digitally signed by both parties agreeing to it.

A typical scenario might look like the following. Having obtained from the factory (or other advertising mechanisms) a quotation for the use of a transient Grid Service, a client negotiates with the factory for an SLA based on this quotation. If the conditions have not changed, the SLA requested by the client is issued. If the conditions have changed (the price of the service changed, or resources on which the QoS was predicated became unavailable), the SLA is renegotiated. At the time and manner specified in the SLA, the service is provided. If the QoS promised in the SLA is not provided according to the agreed-upon monitoring mechanisms, corrective actions are taken.

4.1.4 An Example: Application Service Provider

We continue with our motivating example of the coordinated use of a computational resource and an application software services. The user searches a community registry for service instances that support these capabilities. The user may specify additional nonfunctional requirements, such as a certain refund policy, or a particular architecture. The user's client contacts the factory port on each service and requests a particular level of resource use from both services (e.g., 16 processors with an interconnect greater than 100 Mbs running a Solaris 2.8 operating system and a compatible version of the application software) and a minimum termination time of the reservation.

The factory generates a new service instance for each requested service use and returns these to the user. By querying the SDEs of the newly created services, the user can obtain the agreed price for using the service and the agreed terms and conditions. The SDE of the newly created service may differ from that of the original because the latter may support multiple approaches to setting the price of the software while the created service describes only the agreed-upon protocol. If the user is unhappy with the offered reservation, the GSH may be discarded (or retained until it expires) and the process restarted from the original service. Alternatively, the price-setting protocol may allow the price to be adjusted through the newly created service, which will again generate a new GSH for use in further negotiation steps.

These transient service reservations will be destroyed when their lifetime expires. If the user takes up the reservation, by invoking part of the underlying Grid Service, then the reservation will be confirmed, any subsequent resource consumption will be monitored and recorded in a Resource Usage Service, and charging will then take place when the service invocation is complete.

4.2 The Grid Economic Services Architecture

The constructs provided by OGSA enable a Chargeable Grid Service to be built that can encapsulate an existing Grid Service with the mechanisms needed to set the cost of using a service and to offer it for sale. This approach exploits

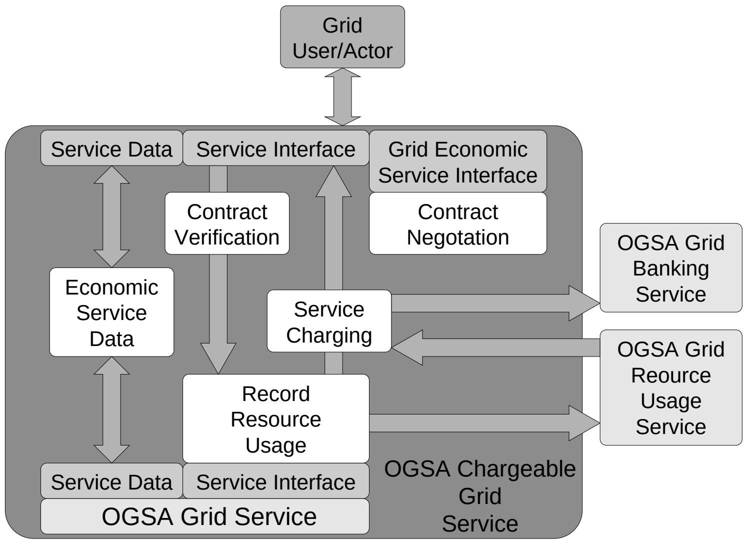

Figure 1.3. The current Grid economic economic architecture.

the basic infrastructure within OGSA for transient Grid Services while retaining considerable flexibility as to the eventual economic model that is used to set the cost of using the service.

Figure 1.3 shows the internal structure within a Chargeable Grid Service. The service data elements are composed from those contained by the underlying Grid Service and from the additional elements generated by the Chargeable Grid Service to describe the economic state of the service. This information is accessible through the standard Grid Service ports. An invocation by an authorized client on the service interface is verified and passed to the underlying service. On completion of the service invocation, the resources used by the service are recorded in an external service—the Resource Usage Service. The resources consumed during the service invocation (e.g., memory, disk space, CPU time) may be charged per unit of consumed resource rather than per service invocation. The cost of using the service is passed to an external service—the Grid Banking Service—for later reconciliation.

5. BUILDING THE U.K.'S COMPUTATIONAL MARKETPLACE

The UK e-Science Programme started in April 2001 as an ambitious £120M three-year effort to change the dynamic of the way science is undertaken by exploiting the emerging Grid infrastructures to enable global collaborations in key areas of science [Tay02]. Within this multidisciplinary activity a core program focused on developing the key middleware expertise and components that would be needed by the U.K. science and business communities to encourage adoption.

As the global Grid infrastructure started to emerge and its commercial adoption started to become a reality, the lack of an economic infrastructure to motivate the provision of Grid Services started to become a barrier to adoption. This situation was recognized within the U.K.'s e-Science Core Programme, and in response the Computational Markets project [MAR] was formed to develop and explore the potential of such an infrastructure within the academic and commercial Grid communities. Project participants include the regional e-science centers in London (lead site), the North West, and Southampton; a variety of commercial partners including hardware vendors, application software vendors, and service providers; and end users within the engineering and physics communities. The U.K.'s Grid Support Centre will deploy the infrastructure developed through the project throughout the UK e-Science Grid.

The project has two main goals: to develop an OGSA infrastructure that supports the trading of Grid Services, and to explore a variety of economic models within this infrastructure through its deployment across a testbed between the e-science centers involved in the project. This will include the instantiation of the Chargeable Grid service, the Resource Usage Service, and the Grid Banking Service, as outlined previously.

One possible long-term outcome from the project is to change the model of resource provisioning within the U.K. for computational, and implicitly data, services. Currently, investigators requiring use of the U.K.'s high- performance computing resources (after passing a peer review) are awarded a budget for the use of the service. This budget is tied to a particular set of resources at a center and cannot be used to purchase general compute or data capability from other providers. Future models for resource provisioning could see this budget available for expenditure on the resources at university computing centers or through the provision of local compute clusters. The ability of researchers to flexibly acquire the most appropriate resources as they are needed would ensure transparent use of these resources and reduce the barriers to new entrants in the provisioning of these resources within the UK community. Key to any form of economic activity is a trustworthy medium of exchange. Within this project this capability is encapsulated in the Grid Banking Service, which records financial transactions and checks that the customer has the ability to pay. In reality we expect this service abstraction to be implemented by trusted third parties such as credit card companies, since we consider the development of an e-currency to be outside the scope of this project. The banking service will also be able to define a conversion mechanism between different currencies if required.

A key goal within the Computational Markets project, and within the wider U.K. e-Science Programme, is that the project's activity contribute to building international standards within the Grid community. It is envisaged that the project will produce a reference implementation of the economic architecture defined through various activities within the Global Grid Forum.

6. ACTIVITY WITH THE GLOBAL GRID FORUM

We focus here on three working groups within the Global Grid Forum's Scheduling and Resource Management area that are actively contributing to the definition of the economic architecture described earlier.

The *Grid Economic Services Architecture Working Group* (GESA-WG) is capturing a set of motivating use cases to identify the requirements for the underlying economic service architecture defined earlier in this chapter. A key element within the overall architecture is the consumption of resources. The Resource Usage Service within GESA exposes the consumption of resources within an organization by a user. Many of these resources (e.g., CPU and memory) might be used to determine the cost of having used the service.

The controlled sharing of resource usage information that has been captured by the underlying service infrastructure is becoming an increasing priority with virtual organizations around the world. A service interface is being defined by the *Resource Usage Service Working Group* (RUS-WG) [RUS] that will enable the secure uploading of consumed resource information and the extraction from the service by authorized clients.

An assumption with the RUS-WG activity is a standard mechanism to interchange data between different Grid entities. The resource information (its values, quantities, and structure) that may need to be exchanged between different centers is being defined within the *Usage Records Working Group* (UR-WG) [UR]. Several possible interchange formats (including XML) are envisaged for this information.

7. THE FUTURE

The past few years have seen the early adoption of Grid infrastructures within the academic and business community. While the use of Grid mechanisms is not yet widespread, their adoption will certainly be accelerated by the Open Grid Services Architecture and its use of Web Services as its service infrastructure. Future Grid environments may therefore comprise thousands of Grid Services exposing applications, software libraries, compute resources, disk storage, network links, instruments, and visualization devices for use by their communities. Nevertheless, while this vision of a pool of Grid Services available for general use is appealing, we emphasize that it is not realistic, as such a service infrastructure would have to be paid for by its users.

We foresee, instead, the emergence of resource brokers that add value to the basic service infrastructure by finding annotating services with information relating to their capability and trustworthiness. Users will be able to obtain their required services from these brokers, who may offer a guarantee as to their capability. Alternatively, users may seek out and discover their own services. These services need not be provided for free; indeed, for widespread acceptance of the Grid paradigm, organizations must have a mechanism for defining and connecting revenue from service provision.

The Internet has brought us ubiquitous access to data and simple services for little or no cost. The Grid offers the possibility of ubiquitous access to more complex services, but their appearance will be predicated on the service provider receiving an income for its provision. The proposed economic architecture is in its early stages of development but will build upon OGSA to be open and extensible across many deployment scenarios and economic models, thereby providing an infrastructure that will enable utility computing. Within this architecture we can see the speculative purchase of resources by services for later resale (a futures market), customer-dependent pricing policies (Grid miles), and other mechanisms to encourage the maximum utilization of resources by maximizing revenue generation.

Acknowledgments

This work is being supported in part by the Computational Markets project funded under the UK e-Science Core Programme by the Department of Trade and Industry and the Engineering and Physical Sciences Research Council (`http://www.lesc.ic.ac.uk/markets/`), and by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under contract W-31-109-Eng-38.

References

[ASGH95] D. Abramson, R. Sosic, J. Giddy, and B. Hall. Nimrod: A tool for performing parameterized simulations using distributed workstations. In *Proceedings of the Fourth IEEE International Symposium on High-Performance Distributed Computing (HPDC-4)*, August 1995.

[BEK+00] Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk, Satish Thatte, and Dave Winer. Simple Object Access Protocol (SOAP) 1.1. Technical Report Note 08, World Wide Web Consotium (W3C), May 2000. Available from `http://www.w3.org/TR/SOAP/`.

[CCMW01] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web Services Description Language (WSDL) 1.1. Technical report, W3C, 2001. Available from `http://www.w3.org/-TR/-wsdl/`.

[FK97] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications*, 11(2):115–129, 1997.

[FKNT02] I Foster, C. Kesselman, J. M. Nick, and S. Tuecke. Grid services for distributed systems integration. *IEEE Computer*, 35(6):37–46, 2002.

[GES] GGF Grid Economic Services Architecture Working Group (GESA-WG). `http://www.gridforum.org/3_SRM/gesa.htm`.

[GLO] Globus Project. `http://www.globus.org`.

[KKL+02] A. Keller, G. Kar, H. Ludwig, A. Dan, and J. L. Hellerstein. Managing dynamic services: A contract based approach to a conceptual architecture. In *Proceedings of 8th IEEE/IFIP Network Operations and Management Symposium (NOMS 2002)*, April 2002.

[KM03] K. Keahey and K. Motawi. Taming of the Grid: Virtual application services. Technical report, Argonne National Laboratory, Mathematics and Computer Science Division Technical Memorandum ANL/MCS-TM-262, 2003.

[LLM88] M. J. Litzkow, M. Livny, and M. W. Mutka. Condor - a hunter of idle workstations. In *Proceedings of the 8th International Conference on Distributed Computing Systems*, pages 104–111, 1988.

[MAR] DTI e-Science Core Technology Programme Computational Markets project. `http://www.lesc.ic.ac.uk/markets/`.

[RUS] GGF Resource Usage Service Working Group (RUS-WG). `http://www.gridforum.org/3_SRM/rus.htm`.

[Tay02] John Taylor. Plenary keynote, GGF 5. `http://www.gridforum.org/Meetings/ggf5/plenary/Mon/`, July 2002.

[TCF+03] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, D. Snelling, and P. Vanderbilt. Open Grid Services Infrastructure (OGSI) 1.0 draft. Technical report, Global Grid Forum (GGF), March 2003. Available from `http://www.gridforum.org/ogsi-wg/`.

[UNI] Unicore. `http://www.unicore.org/`.

[UR] GGF Usage Record Working Group (UR-WG). `http://www.gridforum.org/3_SRM/ur.htm`.

[WHH+92] C. A. Waldspurger, T. Hogg, B. Huberman, J. O. Kephart, and W. S. Stornetta. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering*, 18(2):103–117, 1992.

[Wor] WorldPay UK. `http://www.worldpay.co.uk/`.

[XML] W3 Consoritium XML: eXtensible Markup Language. `http://www.w3c.org/XML`.