INGENIO WORKING PAPER SERIES

INSTITUTO DE GESTIÓN DE LA INNOVACIÓN Y DEL CONOCIMIENTO

Grid Networks for Computing Innovation – Key Social Factors

Paul Haynes

Working Paper Nº 2010/03

INGENIO [CSIC-UPV] Universidad Politécnica de Valencia | Camino de Vera s/n | 46022 Valencia
tel. +34 963 877 048 | fax +34 963 877 991 | innco@ingenio.upv.es

Grid Networks For Computing Innovation – Key Social Factors

Paul Haynes[a]

[a] INGENIO (CSIC-UPV), Valencia (Spain)

> The Knowledge Grid has social characteristics. In the real world, people live and work in a *social grid* obeying social and economic rules and laws. The Knowledge Grid is a *virtual social grid*, where people enjoy and provide services through versatile flow cycles like cycle flows, material flows, energy flows, information flows and knowledge flows. An artificial interconnection environment can only be effective when it works harmoniously with social grids. (Zhuge 2004: 6-7)

The demand for computing power in new scientific research is insatiable, and while newer and more powerful computers are being designed, improved computing power could be harnessed by effective networking of existing machines. Voluntary computing (i.e. labs donate their idle computational resources in exchange for accessing other labs' idle resources) and more formalised grid computing seem, therefore, to be efficient ways of providing more computing power from existing resources. Formalising such sharing into an accessible computational resource able to deal with vast data sets, is the idea behind the grid. Indeed, in addition to exploiting underused computing power, grid systems promise parallel computing, more effective collaboration and increased reliability.

A number of projects exist that have Grid-like properties and some are being integrated into the worldwide Grid initiative[1]. The following is a small selection of such projects:

- Condor is a project which began in 1988 at the University of Wisconsin, with the aim of pooling the resources of all computers in a University department. Condor-G is a newer version, incorporating some tools from the Globus Toolkit, designed to additionally submit jobs to the Grid.
- CODINE, developed in the 1990s, provides cycle scavenging similar to Condor, and a simple graphical user interface to view all available resources. CODINE is an open source project supported by Sun Microsystems, and the next generation of CODINE has been renamed the Sun Grid Engine, and is free online.
- Legion was a project developed in 1993 at the University of Virginia, based on an object oriented approach. However, a lot of applications for metacomputing environments are not object oriented and a Legion spin-off, Avaki Corporation, provides commercial, closed source, Grid solutions.
- Nimrod was a project started at Monash University in Australia in 1994. This project in its newer incarnation – Nimrod/G – incorporates some tools from the Globus Toolkit, enabling it to send calculations out on the Grid.
- UNICOR was a German project, initiated in 1997, but it has since evolved. Its middleware has elements of a Grid Toolkit, as well as a Grid Portal. The integration of UNICORE with Globus is being carried out in the EU-funded project, known as GRIP.

Some research examining the way such grids were used, barriers to resource sharing, how participation evolved and the lessons which could be learnt from the experience of earlier Grid-like projects suggested that flexibility is the most important factor in reducing conflict or question if generic tools, such as the Grid can fit all knowledge-making practices. Douglas Thain, Toff Tannenbaum and Miron Livny, analysing such systems, identify four principles which enable such flexibility and enable such systems to engage potential users more effectively:

- Let communities grow naturally
- Leave the owner in control, whatever the cost
- Plan without being picky

---

[1] For a comparison between four examples: Condor, Nimrod, Legion and Globus, see Norman 2004: 13-42)

- Lend and borrow

Such principles emerge from a number of observations related to the way scientists and research groups contributed to Grid-like projects over twenty years. Such research shows that scientists and researchers generally wish to work together on common problems and with tools of appropriate power, researchers will organise the computing structures they need. Relationships are, though, complex and people invest their time and resources into many communities and networks with varying degrees of commitment and trust. Relationships and requirements evolve over time, often due to necessity, and structures that adapt effectively are typically those which facilitate cooperation but are not be wholly dependent upon cooperation as a principal.

To attract the maximum number of participants in a community, the barriers to participation must be low. The research shows that users will not donate their property to the common good unless they maintain some control over how it is used. Therefore, a system must provide tools for the owner of a resource to set policies and retract a resource for private use. One early problem with grid-like systems, that of the free rider problem, parasitic on the systems resource, was generally resolvable by retracting supply of computing power to uncooperative users.

In the type of scientific community which includes 700 researchers, there will always be idle resources available to do work, though some of these resources will be slow, misconfigured, disconnected, or broken. An over-dependence on the correct operation of any remote device will undermine such a system. Software designers must therefore spend more time contemplating the consequences of failure than the potential benefits of success. The research findings show that when failures occur, the initiative must be able and willing to retry or reassign work as the situation allows.

Such Grid-like projects tend to develop a large body of expertise in distributed resource management. Other experts in related fields such as networking, databases, programming languages, and security are often also available. Thain et al (2005) show that when the research benefits of such expertise is incorporated as much as possible within the system, while, at the same time accepting and integrating knowledge and software from other sources, such Grid-like projects are much more effective. The practices and methods that have contributed to the system as it currently exists are a reflection of the learning experience that has enabled the emergence of Grid computing, and such knowledge and practices should be a learning resource. (see Thain, Tannenbaum and Livny 2005: 325)

Other research, which assesses the barriers and enablers of Grid-like systems, has identified the need for fast, simple, scalable, and secure systems (Cirne et al 2006). In-depth interviews with those involved with eScience projects in the UK outline a number of problems, though these are generally technical issues, or problems that are perceived to have a technical solution (see Schopf and Newhouse 2004):

1) Security is seen as extremely challenging, and system administrators, developers, and users all want more information about common practices and current approaches.

2) Many current Grid tools need to be able to perform delegation, and the lack of an industry standard or even a well-understood set of tools for Web services is of great concern.

3) Having conquered the initial challenge of job submission using Grid tools, users are now concerned with understanding where a job is in its lifetime, where it is failing, why, and what to do next.

4) Software that builds non-deterministically, is hard to install, or doesn’t include verification test suite is seen as unacceptable by users.

5) The desire to have tools perform individual functions has been supplemented by the need to be able to compose these functions together in order to achieve a chain of services to solve application specific problems.

6) Most users want a layer between them and the tool in order to bring the functionality into their own comfort zone. These wrappers do not add functionality per se but significantly increase the usability and usefulness of a service.

7) With the overall time to failure for Grid components decreasing as their number increases, there is a strong need for better verification and instability analysis to discover and resolve problems before a user happens upon them.

6) Most diagnostic tools solve problems other than those seen at the user-level. Tools that look like normal user applications and can help an average user diagnose failures are an urgent requirement.

The findings drawn from the responses of Grid users included the need for on-going conversations between tool developers and users. The research concluded that grid tool developers must continue to talk and interact with application scientists because without such interaction, the tools will not be used (Schopf and Newhouse 2004: 9).

The issue of use is the most crucial issue for those developing the Grid because irrespective of how effective the technology operates, the system requires a critical mass of computer resource commitment from many projects to function. Research by Tony Hey and Anne Trefethen examines the motivation that scientists have for sharing resources. For science to make best use of its limited funds, sharing of paramount importance, though the motivation for any individual scientist is not so clear. Hey and Trefethen argue that funding agencies need to develop additional incentives to encourage such a community-minded approach to research, even when cooperation is likely to benefit all members, as in the case of the Grid. (Hey and Trefethen 2002: 1030).

One of the reasons why potential beneficiaries might be discouraged from using the Grid’s resources is that the Grid relies on trust and security issues that are inherently difficult to solve, because they involve social relationships, not just computer networking. The people in such virtual associations may find themselves in competition, or involved in a conflict with other members or factions, or engaged in sabotage. Often such problems are accentuated by the lack of a human face in such communities, a lack of assurance concerning commitment or opportunities that the system affords, or knowing who to contact when a problem occurred with a component or function of the Grid (see Schopf and Newhouse 2004: 6).

In addition, even those developing the Grid at CERN accept that there is a lot of hype in this field, and lots of promises being made that may never be fulfilled by the technology, recognising too that the novelty and uniqueness of sharing computer power could be confused as merely a web service. There are some precedents for computing infrastructure being overhyped, such as VLANS (Virtual Local Area Networks) and the OSI standard for internet communication, both in the 1990s. The idea of VLANS was basically very similar to a virtual organization, making a LAN for people working on a common project, but who were not necessarily sitting at the same desk in their building every day. The problem was that it proved practically impossible to manage the information about who was in and who was out, and though VLANS still exist, they serve less ambitious functions than initially envisaged.

The example of the OSI standard for internet communication is perhaps more serious. The OSI standard was a top down communication standard supported by governments and industry. It did not gain popularity, possibly because too much time was spent defining the standard but not enough time implementing it, whereas TCP/IP, a less sophisticated standard for the Internet, worked relatively well, was used by scientists, and by being perceived as more practical, became the predominant model for internet communication. As suggested above, a bottom-up approach to Grid development is favoured by some researchers (Thain et al 2005); however, the Linux-like emergence of a new standard is a possibility that poses as many threats to the Grid community as it solves, particularly as a more systematic approach to collaboration seems to be required for large-scale sharing and service oriented outputs, and seems to be afforded by general purpose protocols and a meta scheduler (see Norman 2004: 50). Gregor von Laszewski, Jonathan DiCarlo and Bill Allcock described the process of developing a more coordinated Grid as one that “requires us to think at higher levels of abstraction compared to traditional software development” (Von Laszewski, DiCarlo and Allcock 2007: 1691). This required dedicated systems, for example to monitor service use in order to develop next-generation scheduling systems, quality-of-service guarantees, adaptive systems, and optimisations.

A final set of people-centred issues focus on the barriers to effective use of grid-like systems. These include power inequalities, such as the prospect that in some grid models, smaller research labs have a disadvantage because Grid projects are likely to prioritise high visibility projects and/or high-qualified computer personnel. Such inequalities will continue to exist while there is a limitation on resource use, as Grid models indicate will be the case for the foreseeable future.

Additionally problems of support, particularly for specific data management techniques is also a probably limitation. Using Grid resources will still require sophisticated skills, which smaller initiatives and organisations will have some difficulty obtaining. Research examining 25 Grid-using projects in the UK found that that many of the vital features of the Grid, which were expected to be most effective for providing key data analysis solutions, such as resource discovery, scheduling techniques and data migration, were not used at all:

> What surprised us most about the tools and services in use by the groups we spoke with was not what they were using but what they weren’t…The software most of them were using was expected to be of production-release quality; they were not interested in prototypes or proof-of-concept software that was not resilient to failures. And in general, the groups were having enough of a challenge getting the basic functionality up and stable, so higher-level services were not considered an immediate priority within the next year. (Schopf and Newhouse 2004: 6)

In addition, many services were being used “off label,” i.e. services written for one function were being used in a setting not envisioned by the designer, which had implications for assessing the effects of upgrades. Without feedback from users, or addressing the possibilities of unintended consequences, which are often a feature of the way people engage with or adapt technology, even minor changes or improvements to the grid could cause problems with data sets or incompatibility with previously obtained results.

A final point to consider is that the structure of a grid can also change the nature of an organisation, including the roles relationships and dependencies, which are likely to have an impact on an organisation’s social configuration, which in turn can impact positively or negatively on the communities and institutions they influence. As Sean Norman observes:

> The configuration of a grid has a vast effect on the infrastructure of an organization when considering factors such as performance, reliability and security. All of these factors, including the direction grid computing will take in the future, must be taken into account when deciding whether or not it is beneficial to set up a grid or not (Norman 2004: 51)

References

Cirne, W., Brasileiro, F., Andrade, N., Costa, L. B., Andrade, A., Novaes, R. and Mowbray, M. (2006) Labs of the World, Unite!!! *Journal of Grid Computing* 4: 225–246

Hey, T. and Trefethen, A. E. (2002) The UK e-Science Core Programme and the Grid *Future Generation Computer Systems* 18 1017–1031

Norman, S. (2004) Grid Computing: A survey in Technologies, University of Western Ontario Computer Science Department Working Paper

Schopf, J. M. and Newhouse, S. J. (2004) State of Grid Users: 25 Conversations with UK eScience Groups, Argonne National Laboratory Technical Memorandum No. 278

Thain, D., Tannenbaum, T. and Livny, M. (2005) Distributed Computing in Practice: The Condor Experience, *Concurrency and Computation: Practice and Experience* 17(2-4) 323-356

Tuire, P. and Erno, L. (2001) Exploring Invisible Scientific Communities: Studying Networking Relationships within an Educational Research Community. A Finnish Case, *Higher Education*: 42: 493-513

Von Laszewski, G., DiCarlo, J. and Bill Allcock (2007) A Portal for Visualising Grid Usage *Concurrency and Computation: Practice and Experience* 19: 1683-1692

Zhuge, H. (2004) *The Knowledge Grid* New Jersey: World Scientific

Zhuge, H., Ding, L. and Li, X. (2007) Networking Scientific Resources in the Knowledge Grid Environment *Concurrency and Computation: Practice and Experience* 19:1087–1113