Grids and clouds in scientific computing

F. Orellana [*][†]

on behalf of grid.dk

January 22, 2009

Abstract

This essay tries to narrow down possible merits and pitfalls of connecting and sharing computing resources in the particular case of Danish academia. A review is given of computing needs of various research groups as well as of established and emerging technology in the area. A discussion is given of what kind of resource sharing is desirable and possible with available technology.

1 Introduction: grids and clouds

Sharing computing resources is nothing new: in academic research, large-scale computing resources have always been shared by single or multiple teams. With the arrival of high-speed wide-area networking a natural idea was to connect individual computing facilities and sharing the connected ensemble of resources.

Consequently, in the years 1998-2003, grid computing emerged as a popular concept, driven by the need for data and computing intensive computation in various sciences, and in particular by the data processing required by the new generation of experiments of the Large Hadron Collider (LHC) at the European Centre for Nuclear Research (CERN). The concept of grid computing usually refers to the sharing of computing resources across organizational boundaries. Common arguments for grid computing are: better use of resources and more efficient collaboration. For an introduction to grid concepts, see e.g. ref. [1, 2]. Many, but not all practical implementations use the Globus Toolkit as foundation [3]. The popularity of these concepts led to the, primarily public, funding of a large number of grid computing projects. In Europe, notably the EU-funded EGEE project[4] promotes the vision of a common shared computing infrastructure for research and industry in Europe and across the globe. This infrastructure is to arise through the use of new as well as existing computing facilities in the participating countries, linked via the Internet, using open-source middleware adhering to certain open standards agreed upon in the Open Grid Forum. The EGEE was not alone with this vision - in fact, according to Wolfgang Gentzsch [1]: "*During the past 10 years, we have seen hundreds of grid projects come and go, passing away after government funding ran dry. Most of these projects did not have a realistic (nor pragmatic) sustainability strategy, let alone viable business and operational models for their infrastructures, tools, applications and services, or their intended users. Often, the only asset left after the project was the hands-on grid expertise of the project partners, which certainly is highly valuable but in and of itself does not justify all the effort and funding.*" [6]

Although the government-funded, academic grids of the past decade have largely failed to catch the interest of industry/business, it should be noted that there are actually a number of companies selling software under the label "grid computing"; in particular the "old" batch system vendors: Platform Computing (LSF), Altair (PBS) and SUN (GridEngine), but also newer players like Univa UD (formerly United Devices), Data Synapse, Gigaspaces, xkoto, 3tera (AppLogic), GridGain, Parabon and many more. These companies predominantly sell products providing functionality that is mostly *not* covered by said middleware; e.g:

- classical Linux/UNIX batch scheduling/processing on a local farm
- batch scheduling/processing on an ad-hoc grid of Windows or Linux PC's
- scaling of applications hosted on an application server
- scaling of database performance and accessibility

The core functionality of "traditional" grid middleware as exemplified by EGEE's gLite, is in fact to connect or rather aggregate the standard batch systems listed above. In the following we will keep referring to grid computing in this original sense.

The current situation is that the general popularity of the grid computing concept is fading, while the newer concept of "cloud computing" is gaining traction - at least judging from the number of searches on Google

[*] Niels Bohr Institute, University of Copenhagen

[†] E-mail: orellana@nbi.dk

[1] Gentzsch is the former program coordinator of the large German D-Grid project, active in OGF and one of the original initiators of what is now SUN Grid Engine.

(see figure 1). Cloud computing pretty much owes its existence as a concept to Amazon's successful opening up of its internal computing infrastructure to the outside world. Amazon actually seems to have a business case and several of the other big Internet companies as well as several of the big hardware vendors and hosting providers are now either providing, working on or considering similar offerings. E.g. both Google and Yahoo have started offering paid access to their internal computing infrastructures. [2]

Cloud computing has a lot in common with grid computing:

- both allot alleviating peaks in local computing demands, relying on external resources,
- both give access to a wider range of computing platforms than what is typically available locally,
- both involve managing large amounts of users and large computing infrastructures,

but there are some noticeable differences:

- the grid computing concept was defined in academic papers,
- *the cloud computing concept is not formally defined anywhere,*
- grid computing is almost non-existing outside academia,
- *cloud computing is almost non-existing in academia,*
- grid computing mostly provides job-oriented, high-latency, batch services,
- *cloud computing mostly provides simple, raw access to virtual machines,*
- grid computing is mostly used for batch processing of data files,
- *cloud computing is mostly used for scaling of web applications/sites, i.e. long-serving services,*
- grid computing used either legacy, command-line driven interfaces or heavy SOAP web services,
- *cloud computing typically uses light-weight REST web services.*

That said, the original grid vision was focused on collaboration and interaction between people across organisational boundaries. Such considerations are largely absent in cloud computing, which is driven by companies offering services to isolated individuals, companies or institutions. Moreover, what most cloud computing companies offer are exactly services, not the software they use to power these services, let alone the source code of this software. One notable exception here is the company Enomaly [8], which follows an open-source business model and also advocates a standardardisation of cloud computing interfaces. It should also be pointed out that cloud computing is more general than grid computing in the sense that the services offered are on a lower level [10]. That is, a grid can perfectly well be built on top of a set of clouds; in fact grids can be seen as a logical extension of clouds. For more detailed discussions of grids versus clouds, see [11], [12], [13], [14], [15]. Here we just note that academia is actually acting on the "threat" from cloud computing by developing open-source cloud provisioning systems. Notably the universities of California, Chicago and Madrid (Complutense) have recently launched the projects Eucalyptus [16], Nimbus [17] and OpenNebula [18] respectively. These projects all deliver the means of turning a Linux cluster into a virtual machine provisioning facility, i.e. a cloud and, interestingly, two of them, Eucalyptus and Nimbus support Amazon's EC2 interface.

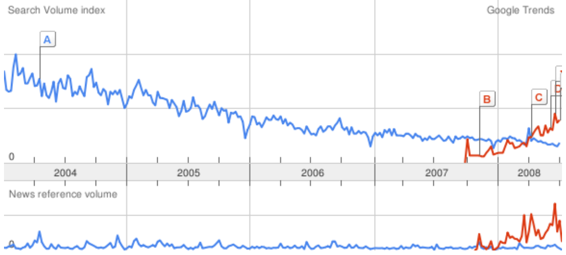

Figure 1: Popularity of grid computing versus cloud computing according to google.com/trends/. Blue: "grid computing". Red: "cloud computing". A: Oracle Joins Enterprise Grid Alliance to Drive Adoption of Grid Computing PR Newswire (press release) - Apr 20 2004. B: IBM Introduces 'Blue Cloud' Computing CIO Today - Nov 15 2007. C: Google and Salesforce.com in cloud computing deal Siliconrepublic.com - Apr 14 2008. D: Demystifying Cloud Computing Intelligent Enterprise - Jun 11 2008. E: Yahoo realigns to support cloud computing, 'core strategies' San Antonio Business Journal - Jun 27 2008. F: Merrill Lynch Estimates "Cloud Computing" To Be $100 Billion Market.. Source: [19]

What, then, can another national computing initiative do to avoid becoming one of Gentzsch' hundreds of grid projects passing away, leaving no assets? Again, Gentzsch offers some help in the form of 10 "Rules for Building a Sustainable Grid" [20]. Here we list the four that appear most sensible to us:

- Rule 1: Identify your specific benefits.

[2] An interesting aside is that for data processing, Google uses their (proprietary) implementation of the MapReduce [7] framework, whereas Yahoo is starting to use an open-source implementation of MapReduce, Hadoop [9]. MapReduce is a framework for parallel computations over large data sets on clusters of commodity computers. Sounds like grid computing? Well, since the service is used only by Google and in one administrative/organisational domain, it is more commonly referred to as cloud computing

- Rule 3: Don't re-invent wheels.
- Rule 5: Evolution, not revolution.
- Rule 9: Try not to grid-enable your applications in the first place.

Rule 1 in particular is in our opinion absolutely vital. In the remainder of this paper, we will try to flesh out this and to a some extent also the other 3 rules in the context of high performance computing (HPC) in Danish academia. For this, we first need to identify the users, their computing needs and the available resources. After that we will identify which benefits closer coupling or interoperation of computing resources has to offer Danish academia and estimate whether or not such benefits can actually be realised with the currently available technology.

It should be noted here that the identification of computing needs unavoidably has an element of subjectivity. The author is intimately familiar with computing in high energy physics, but less so with computing in other branches of science. This paper should therefore not be seen as a thorough and objective analysis, but more as an invitation to debate between all stakeholders.

2 Computing needs

In Danish academic research, the funding of computing hardware is done via the Danish Center for Scientific Computing (DCSC) [21]. As can be seen from the recent allocations of grants [21], the major Danish academic HPC usage comes from bioinformatics, computational materials science, computational astrophysics, computational and theoretical chemistry, high energy physics and biophysics. In this section we shall identify characteristics of the use of HPC in some of these areas of science.

2.1 High energy physics

In high energy physics (HEP), the main users of large-scale computing facilities come from the 4 new experiments, currently being put in place at the Large Hadron Collider at CERN. Denmark is participating in 2 of these 4 experiments, namely ATLAS and ALICE. The fact that the data produced by the experiments is stored and processed all around the world, implies that most of the computing running on the Danish HEP computing resources is in fact not necessarily controlled by danish physicists, but rather by the so-called production managers of each experiment. The data will of course be *analysed* by Danish physicists using these same computing resources. Compared to the centrally controlled production, such analysis activity is much less planned and is expected to involve unpredictable spikes in activity, depending to some extent on how "interesting" the data will be. The activity itself will typically amount to a user running ROOT [24], which in turn uses PROOF [25] to distribute compute jobs on a local cluster, where the data in question is assumed to be present.

For a detailed discussion of the production activity, see e.g. [22], [23], [26] and [27]. For a discussion of the analysis activity, see [26], [27], [28], [29] and [30].

Both kinds of activities typically involve compute jobs that only run on Linux systems, consume ∼2 GB of memory and require very large software (∼10 GB) packages to be installed on these systems. Moreover, these software packages are typically frequently updated (∼once per month).

All this means that, *grosso modo*, HEP has the following types of activity:

1. Manual installation - by the system administrator, of large software packages on the cluster.
2. Centrally (CERN/Oslo) coordinated batch processing of sequential data.
3. User controlled batch processing of sequential data.
4. User controlled analysis activity on a dedicated cluster.
5. Centrally coordinated storing of large amounts of data.
6. User controlled storing of data.
7. Creation of *ad hoc* groups for sharing data between geographically dispersed users (participation in global authentication/authorisation systems).
8. User download of stored data.

Concretely, the last 4 points imply that the computing resources used by HEP must at some level participate in or interoperate with the international grid infrastructure efforts centered around CERN - like NDGF [31] and EGEE [4].

Once a cluster is set up and running production, the only manual task a system administrator has to perform (besides normal system maintenance) is installing the large HEP software packages. For each of the 4 LHC experiments, these are bundled as a single large package that can be installed in a semi-automated fashion. The packages are typically developed and supported only on one platform - Scientific Linux, but usually run on other Linux distributions after some, smaller or larger, efforts. Because of the size of the software packages and the large number of files they contain, downloading and installating them typically takes several hours.

Although CERN-oriented computing is consuming the bulk of the CPU cycles on HEP resources in Denmark, it should be mentioned that there is also a fair amount

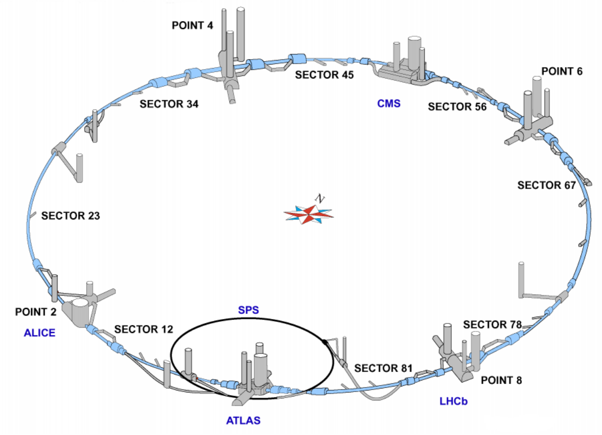

Figure 2: Schematic view of the LHC experiments at CERN. Source: [32].

of users of standard (commercial) software packages like Matlab and Mathematica.

In short, HEP computing is highly batch oriented and involves only trivially parallelizable applications. These applications, however, need a quite high throughput of data: typically they use data files of 0.5 - 2 GB which they read at a rate of up to 2 MB/s. Moreover, HEP computing typically needs to access data registered and stored on remote resources via the grid infrastructure provided by NDGF and/or EGEE.

2.2 Bioinformatics

In bioinformatics the community is more heterogeneous and computing tasks are more diverse: activities are focused around the creation and use of databases and web services. Commonly, batch processing is used to power the number crunching involved, but not necessarily using one of the standard batch systems on the market. Instead, custom-made solutions are frequently used. Non-trivial parallel jobs, requiring fast interconnect between computing nodes are common, but so are trivially parallelizable jobs.

The computational work of a researcher typically consists in executing workflows, involving various web services and databases. Sometimes, a graphical workflow tool like Taverna [33] is used for this.

Like HEP, bioinformatics also typically relies on large amounts of software to be preinstalled on the clusters running the compute jobs. Also here there are a few standard software packages that are in common use, e.g. BLAST [34], but the actual computations are done using a large variety of standard and custom software packages that typically run only on the machine where they were set up. That is, in contrast to HEP, the software used is not necessarily packaged nor easily installable.

In the context of bioinformatics, grid computing has already been quite explored; notably by EGEE and NDGF.

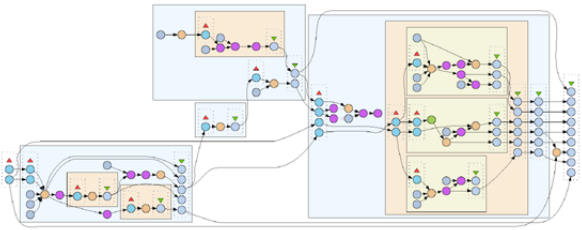

Figure 3: Example of a workflow. Source: [33].

2.3 Computational chemistry

Computational chemistry is a large consumer of CPU cycles in Demnark. Parallel jobs account for a good deal of this, but there are also a significant amount of serial/independent jobs. The software situation is much like that of bioinformatics and HEP with some large standard packages used throughout the community. Most of them are open source, but the most popular one, Gaussian, is commercial. In contrast to bioinformatics and HEP, it is commonplace that individual researchers modify the source code of such software packages on a regular basis. This is the case for e.g. the Scandinavian software package Dalton. In contrast to bioinformatics, the compute jobs are not database or web service oriented, but rather perform raw number crunching like in HEP.

In the context of computational chemistry, grid computing has already been quite explored; notably by NCSA and partners [37] and EGEE.

3 Existing computing resources

This section is a summary of the existing infrastructure. For more details, see appendix A.

3.1 Hardware

Not surprisingly, the bulk of the computing resources on Danish, academic facilities are made up of commodity computers ("pizza boxes" or blades) with Intel or AMD processors. Generally, each machine is equipped with a rather large amount of RAM - typically 2 GB per processor core. In fact, for a casual observer, a typical academic HPC server room looks very much like the typical server room of an Internet service provider or hosting company.

The main differences between the computing resources at each facility are:

- Some facilities have a subset of their computers interconnected with a faster network than standard gigabit ethernet - typically Infiniband. This reflects the fact that the computing jobs generally fall in two categories: parallel and non-parallel (or trivially parallelizable) jobs (parallel jobs require fast interconnect between computing nodes).

- The academic computing resources follow the general industry trend of moving to 64 bits and as a result, typically have both architectures present, but in varying proportions.

It should be mentioned that one site has a relatively large Power-6 installation. We do, however, not really see this as a big issue. The code running on this architecture is custom code and should compile on other architectures as well.

3.2 Software and services

The storage resources are all based on Linux file servers with attached RAID arrays. They also all export this storage to the computing resources and to the local users via a shared file system. The resources that already participate in NDGF, moreover allow access to storage via SRM and GridFTP.

For smaller clusters, the shared file systems used is NFS v3. For clusters with many worker nodes (more than about 20), the performance of NFS v3 becomes prohibitive and a high-performance system like GPFS (from IBM) is used.

All computing clusters run one or several flavours of Linux. The selection of installed software differs from site to site, depending on the profile of the users. Licensed software, like Matlab and Mathematica, consume a significant amount of CPU cycles, but the really heavy consumption comes from open source or custom software. A special case is the quamtum chemistry application Gaussian, which is licensed *and* a heavy CPU consumer.

4 Why interoperation?

For a new system, replacing an older system, to be successful, it is, as already stated, mandatory to very clearly identify the "specific benefits" that are ultimately the reason for putting in place such a system. In our opinion, from the end-users point of view, the relevant benefits of a networked computing infrastructure are:

1. increased effective amount of computing resources available to each researcher and in particular to researchers that do not have access to a local cluster[3]
2. new and more efficient ways of carrying out research, i. e. *extended* functionality of the computational and collaborative tools available to each researcher

For the resource owners, the most interesting benefit is probably reduction of the total cost per CPU-cycle used. This will be achieved if the first of the above two benefits is realised *and* the added administration costs are not too large.

In the following subsections the above two benefits will be discussed in turn.

4.1 Increasing the available amount of computing resources

Today's scientific discoveries are typically done in close competition with research groups from all over the planet. How fast simulations or data processing can be carried out and thus the amount of computing resources available to a given research group, can be a crucial parameter, determining whether or not years of research end in success or failure. Therefore, cluster administrators typically have high- and low-priority queues, with high priority assigned to jobs requiring to run only a short time and lower priority the longer jobs require to run. Also typically, scientific groups that find themselves in particularly close competition, e.g. up to an important conference are given high priority.

If assignment of priorities could be done on a national scale, all involved research groups stand to gain: urgent tasks needing massive amounts of computing power could benefit from underutilised resources elsewhere in the country.

4.2 Collaboration and mobility

As we have seen in section 2, some Danish researchers are already accessing scientific data across borders and participating in so-called virtual organisations (VO's).

In the original grid vision, this way of working was to be made ubiquitous: researchers should not only be able to access remote data, but also to share data and computing resources with colleagues, locally and remotely, by forming virtual organisations on the fly - and have collaborators join and leave these.

In an age of globalisation, digitalisation and ever increasing volumes of data, the ways collaboration in research is done are bound to change. Adapting to this can be done in many ways, but any path chosen will require more seamless interaction with and between the involved computing resources.

One way to increase the productivity of researchers is support mobility: researchers should have their computational infrastructure available from anywhere with an Internet connection. In particular, they should be able to start some large computations at their working place and for example check on the progress from home or from some other institution.

[3] An implicit assumption underlying this is of course that the amount of funding available for computing is finite and independent on how efficiently it is used.

5 Lessons from other projects

National grid initiatives exist in practically all European and many non-european countries and Denmark has also seen a previous effort in the area. In this section an attempt will be made to extract some lessons from these.

5.1 Denmark

In Denmark, grid computing research and participation in international grid collaborations have been undertaken by groups located at:

- the high energy physics group at the Niels Bohr Institute, Copenhagen University (NorduGrid [47], KnowARC[48], NDGF, EGEE)
- the computer science institute at Aalborg University (NorduGrid, NDGF)
- the e-Science Center at the University of Copenhagen (NorduGrid, NDGF)
- NDGF main office in Kastrup (small administrative staff)
- Center for Biological Sequence Analysis, Danish Technical University (SOAP web services [44], EMBRACE [45])

A grid research project, Danish Center for Grid Computing (DCGC) [46], was funded by a national science foundation grant of 7.5 million DKK from 2003-2006 and had participation of the Niels Bohr Institute and Aalborg and Odense Universities. Although the project was a research project, it did operate a pilot grid.

Moreover, independent grid computing research has been and is being carried out at the computer science institutes in Copenhagen, Aalborg and Odense.

With respect to building a sustainable distributed computing infrastructure, the main outcome of these efforts is:

- hands-on grid expertise of a small handful of people
- the participation of two major clusters in NorduGrid (Copenhagen and Aalborg) - continuing to this day
- contributions to the development of the NorduGrid ARC middleware
- the development of an alternative grid solution - Minimum Intrusion Grid (MIG) [49]

Currently, to our knowledge, grid computing is routinely used by two Danish research groups:

- The high energy physics group at the Niels Bohr Institute is using the international grid infrastructure operated by NDGF.
- The gene sequencing group at the Bioinformatics group at Aarhus University is using a MIG infrastructure covering two clusters in Copenhagen and Aarhus, effectively using sandbox resources of DCSC.

Out of the 5 regional operating centres of DCSC, only 2 are effectively contributing resources to NorduGrid via ARC.

Thus, although a national batch system nominally exists, resource sharing on a national level is not taking place.

Lessons learned

For the case of NorduGrid, inspecting the monitoring web page [47] is instructive. It is seen that: 1) The vast majority of grid jobs are jobs simulating CERN data, controlled by two persons, located at Oslo University. 2) A good amount of computing resources are idle.

Generally, adoption over the last 5 years has extended only to a small handful of people. We believe this is due to the following:

- The functionality of the deployed grid middleware does not suffice, i. e. does not provide enough or not the right usability and/or functionality for users to adopt and use it.
- The effort a researcher has to invest in using grid resources is too big.
- Hype and power-grid analogies have alienated sysadmins.

We believe the following measures could help improve on this:

- integrate sites in a sustainable manner: preferably a driving force should be the local system administrator and installations should either be carefully maintained and supported or better not made at all
- prepare proper and realistic planning based on site metrics (CPU occupancy time, job characteristics etc.)
- avoid overselling the product but be very concrete about what it offers - now, not in some distant future

5.2 Other countries

In **Sweden**, **Norway** and **Finland**, like in Denmark, the ARC middleware has been used to establish national batch queues [52, 53, 54]. Norway is special in that it has a massive number of CPU cores (∼10’000) available via NorduGrid. In Finland and Sweden this number is comparable to that of Denmark (2-4000 CPU cores). Overall, the involved institutions in each of these 3 countries appear to be slightly more numerous than in Denmark and it is also our feeling that the same goes for the number of active grid users.

In **Switzerland**, the recently formed national grid association has published a paper on their experiences with various grid middleware products [55] with which they have solid experience: gLite, ARC, XtremWeb-CH [56] and Condor [57]. Their overall conclusion about the middleware reads:

"*Regarding the Grid middleware, our experience is similar to that of previous Grid projects: most of the middleware tools are still fairly demanding to install, maintain, and use, mainly due to their complexity and insufficient documentation. This applies to the system administrators' side, but even more to application developers and users.*"

For gLite it was found that it is still only supported on one platform (Scientific Linux), is intrusive and complex and that operating the gLite services require significant manpower.

For ARC it was found that although it is less intrusive and easier to install than gLite, it suffers from lack of data management, non-data-awareness of the job scheduling, lack of software management and general instabilities.

Along with the Swiss report, the few independent "user experience" or "lessons learned" accounts on grid computing we're aware of are all rather well in line with the above and thus confirm the summary by Gentzsch quoted in section 1:

In 2004, a report on the Swedish SweGrid [58] said:

"*In the survey, most users said porting to the Grid is cumbersome or even difficult. The overall impression of the majority of users was only fair or even poor. Also many users did not think all the HPC resources should be made available on the Grid.*"

Also in 2004, the **Korean** K*Grid, a $37.5 million Globus-based project, running from 2002-2006, reported [50]:

"*Our application scientists say:*
- Grid is hard to access, hard to use, and hard to get the benefit
- Network bandwidth/latency on the Grid is terrible
- We are hard to find application models suitable to the Grid
- We don't know why we should use the Grid"

On the other hand, lots of seemingly positive experience-reports can be found at the recurring EGEE (gLite) User Forum events [51], which also exhibit an impressive array of different applications.

A 2007 report, also by Gentzsch, "Grid Initiatives: Lessons Learned and Recommendations", has two lists of lessons learned and recommendations extracted from his own experience in various grid projects and from interviews with "major representatives" of several large grid initiatives. To our judgement, the lists are well substantiated. As they are rather long, we only reproduce part of them here:

"*- Most of the successful projects in the early days had a strong focus on just one topic.*
- Successful projects were mostly application and user driven.
- The user point-of-view is paramount; a 'build it and they will come approach' will not work.
- There was a high risk often with projects which focused on both applications and infrastructure.
- Missing software engineering methods and especially low usability resulted in low acceptance of project results.
- A lot of the grid middleware currently promoted is really intended for research and demonstrations but needs significant effort to be made suitable for large-scale production usage.
- Application communities shouldn't start developing a core infrastructure from scratch.
- The grid access portal has to be extremely user-friendly.
- Try to study and/or use an existing grid if possible.
- Focus on understanding your user community and their needs.
- Instrument your services so that you collect good data about who is using which services and how. Analyze this data and learn from watching what's really going on, in addition to what users report as happening."

6 Relevant technology

In the previous sections we have outlined possible benefits of and concerns with a networked computing infrastructure. To move on to more concrete considerations, we shall now jump-start such a discussion by giving a list of relevant software. This list is non-exhaustive.

6.1 SOAP web services

One way of sharing computing resources is to put a web service in front of some service one has to offer. This is precisely what e.g. gLite does for a traditional batch system.

In the bioinformatics community it is customary to offer SOAP web service interfaces to services more specialised than general-purpose batch systems - typically databases.

Other examples include the application services implemented on the NERC cluster grid in the UK via their G-Rex framework [60].

That said, it should also be mentioned that many other remote execution paradigms exist, including XML-RPC, CORBA, Java RMI and UNIX rexec.

6.2 Open-source/academic batch system aggregators

Batch system aggregation is the core functionality of what is commonly referred to as “grid middleware”. The following are examples of grid middleware stacks:

- ARC
- gLite
- Unicore [61]
- Nimrod [62]
- MIG

6.3 Commercial batch system aggregators

There are actually a few commercial or semi-commercial products that offer similar functionality to that of “academic” grid middleware. Examples include:

- MOAB Grid Suite [63]
- ProActive [64]
- GridBus [65]
- Fura [66]

6.4 Grid/batch systems

An interesting type of systems are batch systems that are born with wide-area capabilities, i.e. support remote job submission and control, including authentication and file handling. Examples include:

- Condor
- XtremWeb [67]
- Inferno [68]
- GridGain [69]

6.5 Cloud systems

As already mentioned, since the opening of the cloud wave by Amazon, a few open-source projects, allowing anyone to provide similar services to those of Amazon, have seen the light of day:

- OpenNebula
- Nimbus
- Eucalyptus
- Enomaly
- OpenQRM (cloud plugin) [70]

6.6 Open-source/academic batch job managing systems

Batch and grid systems, typically offer only command-line tools for running single jobs. Managing many jobs is typically done via home-brewn scripts. Using higher-level user interfaces can make life easier - also for users running only single jobs. Web portals in particular are popular; examples include GridSphere, LunARC, but also a few desktop applications exist, notably GridWay, Ganga, GridPilot.

7 Discussion

As we saw in section 4, there are compelling reasons for putting in place a networked computing infrastructure for academic research on a national level. In this section we shall try to be more concrete about the envisioned benefits and the feasibility of realizing them with the available technology listed in section 6.

One important point to take into account before engaging in such a discussion is whether or not building a networked computing infrastructure will ultimately be a more cost effective way of improving comptutationally intensive research.

So, ultimately our measure of success is very simple: the increase in the amount of CPU cycles used by the research community as a whole. This increase should at some point in time exceed the increase that would have been gained by simply buying more buying more hardware for each individual institution and researcher.

The following discussion is thus guided by a principle of optimising return on investment (ROI): the money spent on setting up and maintaining a networked computing infrastructure should be balanced by the extra computing time gained by researchers.

7.1 Collaboration

Enabling richer patterns of collaboration is an important motivation for coupling HPC resources. In the grid world, this is done through VO’s. On the EGEE grid, VO’s are administered via a central VOMS server. On NorduGrid a more distributed or ad-hoc approach is employed: VO’s can be defined on a central VOMS server, an LDAP server or simply through a text file on a web server. If choosing to use one of those two grid systems, using one of their central VOMS servers is an option that should be considered. We see the following advantages:

- researchers can participate in those international collaborations that use VO’s of the international grids (EGEE or NorduGrid)
- hosting and maintenance of the VOMS server is done elsewhere

We, however, also see the following disadvantages:

- creation of VO's is typically not possible for a standard user
- dependency on a central service beyond our control

Operating our own VOMS server is also an option, but given the significant installation and maintenance efforts required, it should be considered carefully in the light of our ROI principle. Moreover, it would dismount the first of the above advantages - which we view as important.

Another option would be to use the VO solution of MIG (see section 6). This would avoid the above disadvantages, but also dismount the first, important, advantage.

7.2 Software provisioning

An important assumption underlying the arguments of section 4 is that the majority of the computing jobs of a given researcher can in fact run on any of the involved computing centres and not just on the one with which the researcher is affiliated. From the hardware perspective, as we saw in section 3, apart from 32/64-bit and networking heterogeneity, this is the case. There is, however, another obstacle to running jobs anywhere: differences in installed libraries and software. The consequence of this is that jobs running custom software will most likely only run on a subset of the involved resources. Jobs running standard software packages should in principle be able to run everywhere these are installed.

A base-line requirement is thus that a common set of the most commonly used software packages in each scientific field should be installed at all sites. Ideally, sites should simply be able to subscribe to one or several software catalogue and software from the catalogue should be automatically downloaded and installed if a job requests it. None of the systems described in section 6 have such functionality - in our experience they all put a significant and continuous burden of software installation and updates on local system administrators. The ARC middleware has seen some development effort in this area, but they were not continued and are not part of the stable distribution.

Another desirable feature is that users should be able to add and remove software packages to/from such a software catalogue and make them public, private or accessible to selected VO's. This would cater for the Dalton use case mentioned in section 2.

One major issue with implementing such an automatic software installation system is that installing a software package frequently requires administrator privileges. Such privileges should obviously not be given to any job needing some sofware package. Another issue is that, some software (e.g. the CERN HEP software) packages run only on one or a few platforms. A third issue is that some software packages have a large number of dependencies - perhaps not even documented. All of these issues could be resolved by integrating virtualization with the software installation system.

A fourth issue is that, as we have seen, several of the most popular software packages are commercial, i.e. licensed. This means that an automatic installation system would have to integrate with a VO membership system.

7.3 User experience

A networked computing system is by definition more complex than each of the stand-alone systems it is made up of. It is of course desirable that the end-users of such a system do not feel this added complexity, but also from the point of view of ROI, it is imperative to minimise the extra time both system administrators and users spend on using the new system.

Moreover, it is important that the system eventually is used by as many users as possible. If, after the system has been deployed and running for, say, a year, and a significant fraction of the total CPU cycles on the involved clusters is still consumed by jobs run by other means, benefit 1 of section 4 cannot be achieved.

What has been seen in other projects (see section 5) is that the user experience is a decisive parameter in this respect: most users will not accept a more complicated procedure to get their work done without getting significant benefits in return. The most relevant benefit here is an increased number of available CPU cycles and the users who really appreciate this are the few carrying out really large-scale computations and typically consuming most of the CPU cycles. These users are usually prominent researchers that have plenty of access to funding and are likely to prefer buying their own cluster over dealing with complicated procedures.

From this analysis it is clear that the user experience is absolutely crucial for the success of a networked computing system. As we have seen, this has proved problematic in the past: existing grid systems all offer a more complex command-line interface than their underlying batch systems and tend to alienate users. Web portals appear to fare better, but don't catch the needs of the really big CPU-cycle consumers.

7.4 Single sign-on

All the traditional grids use the X.509/RSA based GSI security framwork of Globus. Historically, these systems have had the problem that many users find the procedures involved in obtaining and using a so-called "grid certificate" too complicated. Moreover the procedures are by some considered to contain inherent security problems. For this and other reasons, a short-lived certificate service (SLCS) was developed [38] by the Swiss academic network provider, SWITCH, for

EGEE. What SWITCH did was to take advantage of the fact that Switzerland already had a country-wide Shibboleth[39]-based infrastructure for authentication and single sign-on to academic web sites.

Several other countries have a similar infrastructure and are planning a setup along the lines of SWITCH. In Norway for example, an SMS-based system has been proposed. For an account of this, together with an account of the mentioned grid-certificate procedures and security problems, see [40].

The idea of SLCS is that a user should still have an RSA key and an X.509 certificate at her disposal, but be faced with much simpler procedures for obtaining and using authentication credentials.

In Danish academia there is no common web authentication system. On the other hand there is a national certificate authority [41] which has issued close to a million digital certificates. This certificate authority is currently in the process of changing vendor and will move to a more mobile architecture. Also, some Danish institutions are participating in another single sign-on project [42].

7.5 Data management

The main purpose of a distributed infrastructure is to allow researchers to process large amounts of data. The process of making such data available to the compute nodes doing the actual processing should clearly be as straight forward as possible for the researcher. With traditional grid systems like gLite or ARC, input data files are uploaded to remote storage via GridFTP by the command-line job submission tool and then made available to the batch system by the server-side grid components.

In the Nordic countries, a large distributed data management system is operated by NDGF. This system can in principle be used by all Danish academic researchers and is accessible via the SRM protocol, supported by both gLite and ARC.

Recalling section 4, the researcher should be able to start compute jobs and access their output from any PC with an Internet connection. This can be achieved by deploying one of the batch job management systems of section 6.

7.6 Batch infrastructure

In the two grid middleware stacks, EGEE's gLite and NorduGrid's ARC, we don't see any major change as compared to 5 years ago, when both the Swedish "grid project", SweGrid and the last Danish "grid project", DCGC, were launched.

The lessons learned from past projects tell us that basing a batch computing infrastructure on gLite usually requires full participation in EGEE (meetings, mailing lists, regular software updates), with an associated time consumption of approximately 1/2 FTE per site. Moreover, a number of central services have to be deployed and operated, or the system has to be dependent on central services operated in other countries. The former option would introduce significant further deployment and maintenance expenses. The user interface consists of a suite of command-line tools for submitting and manipulating single compute jobs, copying files from and to storage resources etc. It is distributed as a 156 MB compressed archive and available only for Scientific Linux 4.

Basing the infrastructure on ARC can perfectly well be done without active NorduGrid participation (see [23] and [55]), with a somewhat lower time consumption (about 1/4 FTE per site), with only one central service requiring only small deployment and maintenance expenses. In Denmark, we moreover have the significant advantage of being able to profit from collaboration with NDGF, which runs an ARC-based grid on a Nordic level. Although ARC was constructed for the demanding requirements of high energy physics (large job volumes, large in/out), we have seen that ARC suffers from some bottleneck and performance problems, that have the potential of short-circuiting the CPU-utilization gains that networking resources should cause, i.e. prevent us from achieving benefit 1 of section 4. The user interface consists of a suite of command-line tools with similar functionality to those of EGEE. This suite is distributed as a 5 MB tarball that runs on all major Linux distributions and allows accessing data on both NorduGrid and EGEE resources, but only running jobs on NorduGrid resources.

Another option for the computing infrastructure is MIG. The small amount of documentation and usage reports makes it difficult to estimate how much effort it would take to deploy and operate a MIG based system or how reliable or performant such a system would be. It is, however, a Danish academic project and the necessary expertise is available. MIG uses either SSH or its own wide-area network file system to stage input and output files from and to a central server. Although the performance of this is *a priori* not expected to match the requirements of high energy physics (see section 2), the distributed file system of MIG could be an interesting way of improving the user experience. MIG offers a web portal, which, however, would need some polishing to meet the usability requirements outlined in section 7.3.

7.7 Metrics

In order to carry out planning, assign jobs to resources, realize benefit 1 of section 4 and have a measure of success, metrics are needed: as a minimum the occupancy/idle times for all involved resources, but ideally also statistics about usage patterns and the nature of jobs (running times, disk I/O, software packages used, number of jobs submitted per user, where jobs are submitted from, etc.). Even more ideally, resource owners should be able to bill individuals, virtual organisations or institutions for consumed computing power, network and disk storage.

The grid systems we are familiar with do not provide such statistics or billing capabilites and so most likely, bare occupancy/idle-time numbers will have to do.

8 Conclusions

The term "grid computing" is now 10 years old and a large number of national "grid projects" have come and gone. We believe that when launching a similar project, it is important to decide on specific goals and measures of success and to be realistic about possible achievements. In particular we find it necessary to heed the advice in section 5 and:

- document computing needs thoroughly
- document the amount of computing power available at the involved facilities
- document computing usage patterns
- produce computing resource usage statistics for all involved facilities

Regardles of the outcome of such a study, we have already seen that computing paradigms differ significantly across the various research communities. To cater for example for the standard high energy physics and bioinformatics researcher, without requiring significant changes in the working pattern of one or the other, does simply not appear possible with the currently available technology.

That said, with the emerging cloud/virtualisation technology, establishing an infrastructure that covers all fields of research might certainly be possible in the not too far future.

Thus, for the moment, we find it advisable to leverage the existing hands-on grid expertise to establish a national batch system, integrating existing batch computing resources and any existing grid deployments.

For such an endeavour we propose the following:

- make a set of command-line tools seamlessly available to users, *optionally* replacing the command-line tools for the local batch system they are used to
- be clear about the fact that such batch functionality does not cover all use cases
- pay special attention to user-friendliness
- document the system clearly and thoroughly
- provide single sign-on, job control and file access from anywhere; initially via a web portal; later also via integration with desktop, file systems etc.
- establish a central catalogue with a selection of software packages covering a significant fraction of the user needs and ensure this software is installed on all sites - in the longer run, create an automated and user-driven software installation system
- implement VO and software package management from the web portal - i.e. make it easy for users to form VO's and share computers, storage and software
- to avoid the risk of focusing on both applications and infrastructure (see section 5), recruit a small selection of pilot researchers with strong computational needs, patience and a good understanding of computing

We have seen that the software to establish such a system does not appear to be readily available in a production state and certainly not as a single software package. Therefore, we expect the use of various software packages like those listed in section 6 as well as a good deal of customisation and development. The time needed to do such development, carry out alpha and beta testing and produce truly user friendly interfaces should not be underestimated.

Keeping in mind the available expertise, a possible selection of software packages to help put in place such a system is: ARC, MIG, Lunarc and GridPilot.

It should well be worth, however, to give other systems like those listed in section 6 a try and assess if one or several of these systems or their components can provide any missing functionality.

W.r.t. authentication, we see no way around using the X.509/RSA based GSI security framework of Globus for interacting with the international academic grids (NorduGrid and EGEE). To make life as easy as possible for users, certificates from both the Danish national certificate authority as well as those from the existing NorduGrid and MIG certificate autorities should be recognised by all sites. The implementation of virtual organizations is a subject that needs further study.

Moreover, we stress that in the longer run tighter desktop integration through graphical file management and job control is desirable.

Finally, we believe that allocating resources to keep up with cloud/virtualization developments is a good longer-term investment.

Appendices

A Danish academic HPC resources

The funding model in Danish academia requires each research group with large computing needs to apply to a central organisation - DCSC. If such an application is successful, the research group then chooses one of the 5 regional operating centres to procure and operate the hardware. Alternatively, the research group can be granted the right to use so-called sandbox resources. These are a a pool of computing resources to which each operating centre has commited to assigning 10 % of its total computing resources.

In this section the characteristics of the hardware installations at each operating centre together with the main use of each installation is summarised.

A.1 University of Copenhagen

The current installation at DCSC/KU includes the following data processing hardware:

- A 24-processor (R12K) SGI Origin 3400 with 22 GB of main memory
- A 22-processor (R12K) SGI Onyx 3400 with 12 GB of main memory
- A 64-processor (Itanium-2) SGI Altix 3000 with 128 GB of main memory
- A 526-core IBM e1350 cluster with InfiniBand interconnect with AMD Opteron processors with 1-2 GB of memory per core
- 20 IBM Bladecenter H with a total of 2200 cores from processors of various kinds - including Intel Quad-Core, AMD Dual-Core, IBM Power6 and IBM CELL processors and 2 GB of memory per core
- A DELL PowerEdge M1000e blade system equipped with Intel Xeon processors and a total of 40 cores with 2 GB of memory per core

The IBM Bladecenters have 70 TB of attached storage which is served to the outside via dCache [43], i. e. can be accessed via GridFTP (and internally via DCAP). An additional 26 TB of the storage is served to the data processing nodes via a network file system.

Additionally, the installation includes the following storage hardware:

- 2 SGI TP9100 raid arrays containing ∼5.2 TB of raw storage
- 28 IBM EXP3000 systems providing 240 TB of raw storage
- A IBM TS3500 (3584) tape library

The EXP3000 and TS3500 storage will be made available via dCache.

All the compute nodes run the Linux operating system CentOS (version 5).

The biggest “customers” of the cluster are astrophysics, bioinformatics and high energy physics. These areas of research have rather differing needs in terms of disk/data access, RAM requirements and connectivity requirments. This is reflected in the fact that a dCache installation is operated solely for the benefit of high energy physics, while a subset of the processing units are interlinked with infiniband - primarily for the benefit of astrophysics.

A.2 Danish Technical University

See [21].

A.3 University of Aarhus

See [21].

A.4 University of Aalborg

See [21].

References

[1] I. Foster and C. Kesselman: *The Grid: Blueprint for a New Computing Infrastructure* (1998), Morgan Kaufmann; 1st edition (November 1998), ISBN: 1558604758

[2] Ian Foster, Carl Kesselman, Steven Tuecke (2001), "The Anatomy of the Grid - Enabling Scalable Virtual Organizations", http://arxiv.org/abs/cs.AR/0103025

[3] I. Foster, et al.: The Globus Toolkit, http://www.globus.org/

[4] Enabling Grids for EsciencE, http://cern.ch/egee/

[5] Open Grid Forum, http://www.ogf.org/

[6] Wolfgang Gentzsch: "Grids are Dead! Or are they?" (June 2008), GridToday, http://www.gridtoday.com/grid/2381106.html

[7] Jeffrey Dean and Sanjay Ghemawat, Google Labs (2004): "MapReduce: Simplified Data Processing on Large Clusters", http://labs.google.com/papers/mapreduce.html

[8] Enomaly, http://www.enomaly.com/

[9] Hadoop, http://hadoop.apache.org/

[10] Bernard Lunn (July 2008): "Does Yahoo Independence Matter to The Rest Of Us? Yes!", www.readwriteweb.com

[11] EGEE (2008): "An EGEE comparative study: Clouds and grids: Evolution or Revolution?", https://edms.cern.ch/file/925013/4/EGEE-Grid-Cloud-v1_2.pdf

[12] Ian Foster (2008): "There's Grid in them thar Clouds", http://ianfoster.typepad.com/blog/2008/01/theres-grid-in.html

[13] Shantenu Jha, Andre Merzky and Geoffrey Fox (2008): "Using Clouds to Provide Grids Higher-Levels of Abstraction and Explicit Support for Usage Modes", OGF draft artf6243, http://www.ogf.org/Public_Comment_Docs/...

[14] Ian Foster (2008): "A critique of "Using Clouds to Provide Grids..."", http://ianfoster.typepad.com/blog/2008/09/a-critique-of-u.html

[15] Mayur Palankar, Ayodele Onibokun, Adriana Iamnitchi, Matei Ripeanu (2008): "Amazon S3 for Science Grids: a Viable Solution?", *submitted to DADC08*, www.csee.usf.edu/~anda/papers/AmazonS3_TR.pdf

[16] Eucalyptus, http://eucalyptus.cs.ucsb.edu/

[17] Nimbus, http://workspace.globus.org/

[18] OpenNebula, http://www.opennebula.org/

[19] Google Trends, http://www.google.com/trends/

[20] Wolfgang Gentzsch (February 2008): "Top 10 Rules for Building a Sustainable Grid", http://www.gridtoday.com/grid/2095469.html

[21] Danish Center for Scientific Computing, http://dcsc.dk/

[22] F. Carminati et al. (2004), "Common Use Cases for a HEP Common Application Layer (HEP-CAL')", HEP-CAL RTAG Report, LHC-SC2-20-2002, http://project-lcg-gag.web.cern.ch/

[23] M. Dosil, A. Farilla, M. Gallas, V. Giangiobbe and F. Orellana, "Massive data processing for the ATLAS combined test beam," IEEE Trans. Nucl. Sci. **53** (2006) 2887.

[24] "ROOT, An Object-Oriented Data Analysis Framework", http://root.cern.ch/

[25] "PROOF: the Parallel ROOT Facility", http://root.cern.ch/twiki/bin/view/ROOT/PROOF

[26] G. Duckeck et al. [ATLAS Collaboration] (2005), "ATLAS computing: Technical design report" , CERN-LHCC-2005-022, http://cdsweb.cern.ch/record/837738

[27] F. Carminati et al. [ALICE Collaboration] (2005), "ALICE computing: Technical design report" , CERN-LHCC-2005-018, http://cdsweb.cern.ch/record/832753

[28] Juerg Beringer (2003), "Physics analysis at LHC: A wish list for grid computing" (2003), CERN-LCGAPP-2003-09, http://lcgapp.cern.ch/project/mgmt/doc.html

[29] S. Gonzalez de la Hoz, L. M. Ruiz and D. Liko, "First experience and adaptation of existing tools to ATLAS distributed Eur. Phys. J. C **53** (2008) 467.

[30] S. Gonzalez de la Hoz *et al.*, "Analysis facility infrastructure (Tier-3) for ATLAS experiment," Eur. Phys. J. C **54** (2008) 691.

[31] Nordic Datagrid facility, http://ndgf.org/

[32] Imperial College HEP research group web site, http://www.hep.ph.ic.ac.uk/

[33] Taverna project,
http://taverna.sourceforge.net/

[34] BLAST: Basic Local Alignment Search Tool,
http://blast.ncbi.nlm.nih.gov/Blast.cgi

[35] Gaussian,
http://www.gaussian.com/

[36] DALTON,
http://www.kjemi.uio.no/software/dalton/dalton.html

[37] Computational Chemistry Grid,
https://www.gridchem.org/

[38] SWITCH: Short Lived Credential Service (SLCS),
http://www.switch.ch/grid/slcs/

[39] Shibboleth,
http://shibboleth.internet2.edu/

[40] Henrik Austad, "Creating, sending, signing and retrieving certificates for authentication with ARC and NorGrid",
https://ow.feide.no/metacenter:work-documents:2008.5

[41] Digital Signatur,
http://www.digitalsignatur.dk/

[42] Where Are You From (WAYF),
https://www.wayf.dk/

[43] The dCache storage manager,
http://www.dcache.org/

[44] SOAP web services offered by the Center for Biological Sequence Analysis, Danish Technical University,
http://www.cbs.dtu.dk/services/

[45] EMBRACE Network of Excellence - A European Model for Bioinformatics Research and Community Education,
http://www.embracegrid.info/

[46] Danish Center for Grid Computing,
http://www.dcgc.dk/

[47] NorduGrid,
http://www.nordugrid.org/

[48] KnowARC,
http://www.knowarc.eu/

[49] Minimum Intrusion Grid,
http://www.migrid.org/

[50] K*Grid: Lessons learned (2004),
http://pragma6.cnic.ac.cn/

[51] EGEE User Forum,
http://egee2.eu-egee.org/egee_events/userforum

[52] SweGrid,
http://www.swegrid.se/

[53] NorGrid,
http://www.norgrid.no/

[54] M-Grid,
http://www.csc.fi/

[55] Nabil Abdennadher, Peter Engel, Derek Feichtinger, Dean Flanders, Placi Flury, Sigve Haug, Pascal Jermini, Sergio Maffioletti, Cesare Pautasso, Heinz Stockinger, Wibke Sudholt, Michela Thiemard, Nadya Williams, Christoph Witzig, "Initializing a National Grid Infrastructure - Lessons Learned from the Swiss National Grid Association Seed Project, " Cluster Computing and the Grid, IEEE International Symposium on, vol. 0, no. 0, pp. 169-176, 2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID), 2008

[56] XtremWebCH,
http://www.xtremwebch.net/

[57] Condor,
http://www.cs.wisc.edu/condor/

[58] "SweGrid - what do users of a Grid really think?",
http://www.hoise.com/primeur/

[59] Wolfgang Gentzsch, "Grid Initiatives: Lessons Learned and Recommendations", Report,
http://www.renci.org/publications/reports.php

[60] Reading e-Science Centre Projects,
http://www.resc.rdg.ac.uk/projects.php

[61] UNICORE (Uniform Interface to Computing Resources),
http://www.unicore.eu/

[62] NIMROD,
http://messagelab.monash.edu.au/NimrodG

[63] MOAB Grid Suite,
http://www.clusterresources.com/

[64] Proactive Parallel Suite,
http://proactive.inria.fr/

[65] The Gridbus project,
http://www.gridbus.eu/

[66] Gridsystems (makers of Fura),
http://www.gridsystems.com/

[67] XtremWeb,
http://xtremweb.net/

[68] Inferno,
http://www.vitanuova.com/grid/

[69] GridGain,
http://www.gridgain.com/

[70] OpenQRM,
http://www.openqrm.com/