

Grid Computing

Linking Data Networks

Grid computing was designed as a democratic, collective computer networking paradigm, but the current crop of software is all the rave with research scientists, computer scientists, and the IT industry in general. We find out just what it means and explore some of the advantages.

BY RÜDIGER BERLICH



In 1998, Ian Foster and Carl Kesselman published their book, “The Grid – Blueprint for a New Computing Infrastructure”. This event is viewed by many as the start of a new era in distributed computing (see Figures 1 and 2). Foster and Kesselman’s vision was that of providing distributed resources for transparent public use, based on a standardized interface. The idea was that people could access computational power, content, and other computer services in an easy way, just like using electricity by plugging a device into a wall socket. This dream helped the founders derive a name for their project.

Tim Berners-Lee had a similar vision, way back in 1991 when he dreamed up the World Wide Web at CERN (Centre Européenne pour la Recherche Nucléaire) in Geneva. Just like it was for the WWW back in the 90s, CERN is again one of the driving forces behind Grid computing. This has unfortunately led some people to poke fun at the field of Grid technologies, referring to them as the Web on steroids.

What is a Grid?

Grid computing, rightly, has the reputation of being an important future-oriented technology. This is one reason why research sponsorship is forthcoming, whenever someone drops the name. Research groups with only a vague connection to distributed computing tend to add the magic word to their project portfolios. Of course, this kind of environment makes the task of precisely defining Grid computing all the more difficult.

The definition is practice-oriented by necessity. There is a vague distinction between Foster and Kesselman’s original vision, and those researchers who use distributed resources to tackle a genuine problem. The second group includes particle physicists. Starting in 2007, thousands of scientists scattered around the globe will be tackling the multiple PetaBytes of data from experiments with CERN’s Large Hadron Collider (LHC). The computational resources that the individual scientists have at their disposal locally are totally inadequate to the

task. Also, it makes more sense to take the programs to the data, rather than vice-versa, due to the sheer masses of numbers that need crunching.

Viewed in this light, an infrastructure that links the enormous memory capacity and ten thousands of CPUs in a virtual way begins to make sense. Traditional cluster technologies cannot cope with the scale, or the heterogeneous hardware zoo – a new approach is the only way to go: Grid computing. Grids are so well suited to particle physics because scientists often compute separate datasets with parallel, identical instances of a program. Although particle physics may not be the original motivation for Grid computing, it is definitely one of its driving forces.

This form of Grid computing is concerned with the global distribution of identical program instances, just like traditional batch systems working in local clusters. If you like, you could compare such Grid applications with clusters of clusters. The “Principles of Distributed Computing” box elaborates on the clus-

tering aspects. In contrast, genuine parallel applications that exchange masses of data between individual computational nodes are unlikely to play a major role in Grid computing.

Distributed databases are a different story, however. Grids could be put to extremely effective use in health services (to provide access to medical records), or to merge the data generated by an enterprise with global activities, or for search engine technologies.

Networks, Filesystems and Middleware

To achieve the goal of globally distributed computing, Grid computing relies on the new developments and enhancements in many technology based fields. High-speed public networks are one obvious prerequisite. The high-performance national networks in many industrialized nations will need to merge to form a "World Wide Grid".

The network technology is available, and the nominal bandwidth continues to grow at an acceptable rate. The speed at which networks are expanded is more a question of national budgets than a technological issue. In Europe, GÉANT [1], a cooperation between 26 national research networks plays a dominant role.

At present, a lot of research is going into the development of distributed filesystems which are indispensable for running data-centers – the computational nodes of the Grid. In some cases, these filesystems are globally accessible, although the high latency in Wide Area Networks does impact this potential (see box "Principles of Distributed Computing").

To run a program on a Grid, you need some experience with middleware. Its role can be compared to the network layer of an operating system. An application that wants to transfer data across a network does not need any knowledge of the underlying network hardware. In a similar way, a



Figure 1: Carl Kesselman at the Cern School of Computing 2002 in Vico Equense, near Naples, Italy. The co-author of the first book on grid computing is one of the founding fathers of grid computing.

Grid application should not need to worry about things like authentication between machines and authorization, or even billing – all of these tasks are the responsibility of the middleware.

Middleware has only restricted potential for hiding the underlying computing infrastructure. As long as Grid users send programs that the target systems can handle interpretatively, things should be fine (the same thing applies to interpreters with just-in-time compilers). What happens if they start using pre-compiled binaries? This is a question that needs looking into – a large-scale Grid network can contain any kind of (potentially incompatible) hardware, be it a 64-bit RISC architecture, a 32-bit Intel box, or something completely different.



Figure 2: Ian Foster – just like Carl Kesselman, Ian is one of the grid founding fathers – seen at the Sun booth at the Supercomputing 2001 fair in Denver, CO. The slogan "Sun Powers The Grid" shows that the industry is genuinely interested in the new technology.

In this kind of environment, Java and C# would seem to gain many points by hiding the underlying architecture. On the other hand, modern Grid middleware can also support the restriction of a Grid application to a pre-defined architectural type.

Creativity Versus Standards

You can put the Babylonian confusion in the middleware area down as one of the characteristics of free, unrestricted research if you like. On the one hand, it is a good and quite normal thing to see

numerous creative approaches competing in a creative environment – just think about the mail clients or the various desktop environments for Linux. On the other hand, industry and research users are, for good reasons, looking for standards that will allow them to start writing programs.

Evolution should provide a solution. It is typical for Open Source that the best candidate asserts itself in the end. At present, and judged by the number of current installations, the Middleware Globus Toolkit is winning, along with software packages like the European Data Grid Toolkit, which provides additional functionality. Ian Foster and Carl Kesselman, the two authors of the Grid computing bible are members of Globus research groups and lend Globus some weight due to their activities.

The Globus Paradigm

However, the Globus project, with its variety of programs and versions, is responsible for some of the confusion in the Grid universe, no matter how honorable its intentions may be. The monolithic structure of version 2 of the Globus Toolkit (GT 2) has notched quite a few installations. Version 3 (GT 3) introduces the new Open Grid Service Architecture (OGSA), based on so-called Grid services. This paradigm shift caused some

degree of uncertainty among those responsible for Grid project management.

People had started to commit to GT 3 when a new version change was rung in at the beginning of this year. GT 4 attempts to retain compatibility to traditional Web services.

One important aspect of Grid computing is security. Networking thousands of computers scattered around the globe across the Internet is like throwing down the gauntlet to the tiresome but adventurous Script Kiddie fraction. Here the Globus Grid Security Infrastructure (GSI) component can be used for authentication and authorization.

Eurovision

There is some common Grid functionality that Globus does not cover – on purpose. Based on an underlying Globus framework, however, most of this functionality is implemented by the European Data Grid (EDG). The services

include the so-called Resource Broker, which distributes Grid applications transparently among appropriate computational resources according to their requirements. EDG was a genuine European Community Project equipped with sufficient funding and manpower. However, the project was recently replaced by EGEE (Enabling Grids for E-Science in Europe) in the near future. EGEE will build on the experience gained by the European Data Grid.

AliEn (Alice Environment, [2]), a product of the Alice Experiment at the

Large Hadron Collider, is a prime example of the power of Open Source. In contrast to EDG, AliEn is not a completely new development; instead it uses existing Perl modules, wherever it can. With a team of just a few developers, Alice’s “Extreme Programming” techniques have been successful in authoring a working Grid environment with functionality on a par with EDG.

Supercomputing applications in particular tend to access the results of the Unicore (Uniform Interfaces to Computing Resources, [3]) project, for example

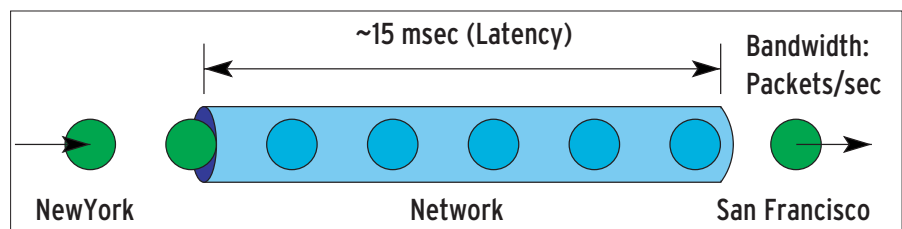


Figure 4: Network latency and bandwidth restrict the usefulness of Grid computing. The latency, the time a data packet takes to cross a network, in particular is something that cannot be controlled arbitrarily.

Principles of Distributed Computing

Grid applications are subject to the same laws as any distributed application. The network bandwidth and latency dictate the performance of most applications. Network bandwidth is a measurement of the amount of traffic a network can handle per unit of time. The latency measures the signal runtime between the sender and receiver. On Linux, you can use the *ping* command to test

this value. Ping measures the time for the round trip, which is twice the latency. The round trip time between Forschungszentrum Karlsruhe, Germany, and Ruhr University in Bochum is about 20 ms (see Figure 3). The latency on a local network should be an order of magnitude less. Latency times on multiprocessor systems are negligibly small in contrast.

know that multiple parallel instances of the program will be running. At the end of the program, either the user, or the software, simply needs to merge the results of the individual computations.

Exchanging Data Slows Down the Application

Traditional parallel cluster applications tend to exchange data during execution. This can be an issue, if one instance of a program has to wait for another, in order to carry on its task. In this case, high latency and low bandwidth on a WAN can aggravate the issue. Local networks are better suited (LANs or clusters), as are multiprocessor systems.

This makes it easy to recognize the kind of applications that lend themselves to Grid computing. As they typically run across a WAN, the quality of the communication between the computational nodes is the major factor in determining whether it makes sense to run an application on a Grid. Although the Grid theoretically provides almost unlimited computational resources, the performance suffers if nodes waste computational time waiting for other nodes to respond.

Particle physics in particular, and other branches of research in general, can leverage Grid computing environments to analyze large quantities of data in as short a time as possible. They typically use applications which are “nicely parallel”.

Some applications remain unaffected by high latency and low bandwidth. The same program can compute a dataset segment on multiple computers – which may be distributed around the globe – at the same time; the individual instances do not exchange any data, however. This kind of application is often referred to as being “embarrassingly parallel”, or in a more optimistic frame of mind, as being “nicely parallel”. It does not place any additional demands on the programming. A developer may not even need to

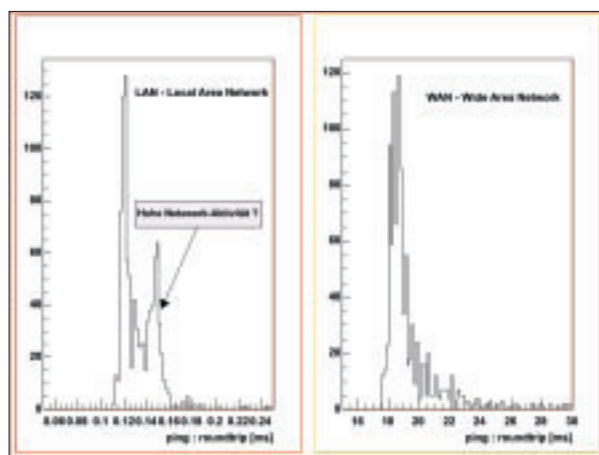


Figure 3: Histogram of round trip times between two network connections (double latency, measured with *ping*). The graph on the left shows a LAN, the one on the right the connection from the Research Center at Karlsruhe, to the University of Bochum, both in Germany. The values on the WAN are typically around 18 to 20 milliseconds, whereas the LAN achieves between 0.11 and 0.16 milliseconds.

the German Weather Service. Just like Globus and EDG, Unicore uses a kind of distributed batch system. In addition to the middleware discussed previously, names such as Cactus [4], Legion [5], and Condor [6] are commonly heard.

Standards, Grid Forum, and D-Grid

The variety and independence of the various projects make it quite clear that Grid computing desperately needs uniform standards and protocols. The Global Grid Forum (GGF) is well on its way to becoming a standards organization for Grid computing. It aims to play a role similar to that of the IETF (Internet Engineering Taskforce) for the Internet.

Grid experts meet at several events throughout the year, to exchange experiences. The 10th Global Grid Forum took place at the Humboldt University in Berlin, Germany, in March 2004. The

choice of venue may surprise some people, as you will be hard-pressed to find a German project among the ranks of the major Grid initiatives, from the Malaysia Grid to semi-military installations such as the US Department of Energy [7]. At GGF10, the German Federal Minister of Education and Research, Ms Edelgard Bulmahn, introduced a very large-scale, multi-organization German Grid initiative – D-Grid.

Programmed Chaos

The Grid cannot be regarded as “World Wide” judged from today’s standpoint. It is as far away from being global as it is from being standardized. Conference attendees have thus sometimes referred to the Grid as the G-word (alluding to the four letter word). The problem is not only related to funds or staff. Too many clever people have come up with too many (equally clever) solutions – as

is typically the case in research. It will be interesting to see what kind of effect the growing interest among businesses will have on the new technology. Although insiders are not surprised or concerned about the chaotic landscape, commercial and scientific exploitation requires standards – so ending the chaos. ■

INFO

- [1] GÉANT project: <http://www.dante.net/server/show/nav.007>
- [2] AliEn project: <http://www.cerncourier.com/main/article/42/9/6>
- [3] Unicore project: <http://www.unicore.org/>
- [4] Cactus environment: <http://www.cactuscode.org/>
- [5] Legion project: <http://legion.virginia.edu/>
- [6] Condor project: <http://www.cs.wisc.edu/condor/>
- [7] US Department of Energy Science Grid: <http://doesciencegrid.org/>

Searching for the Indivisible

Elementary particle physics is concerned with searching for the basic building blocks of all material, and with describing the forces that act between them. As far as we know today, an elementary particle does not have an internal structure. When scientists think that they have discovered an elementary particle, this often proves to be untrue on closer inspection. By today’s standards, atoms are gigantic, complex objects that comprise electrons, protons and neutrons, the latter being comprised of quarks, and there is no end in sight.

Enormous Rings for Minuscule Particles

To help search for new particles and sub-structures in known particles, researchers use enormous accelerator systems, such as the PEP II Ring at the Stanford Linear Accelerator Center (SLAC) in California, or the accelerator rings at CERN in Geneva, Switzerland. These accelerators cause various particle types to collide.

The Large Electron Positron Collider (LEP), which was introduced at CERN in 1989, accelerated electrons, and their anti-particles, positrons, through a 27 kilometer ring causing them to collide at four points. The LEP tunnel runs below Geneva and the Swiss/French Jura mountain range. The ring is so huge that its calibration not only depends on tidal effects, but also on seasonal variations in the water level of Lake Geneva, which affect the surrounding countryside.

Just recently, LEP was closed down to allow an even bigger successor to be set up. The

new Large Hadron Collider (LHC) is currently being installed in the former LEP tunnel. Together with its experiments – including Atlas and Alice – the accelerator, which is due to go on-line in 2007, should allow scientists to generate previously unknown energy levels. Achieving increasingly tiny dimensions requires increasingly large amounts of energy. Some particle types are so heavy (and following Einstein’s $E=mc^2$ so full of energy) that they cannot be created using known accelerators. One example is the long sought-after Higgs-Boson.

Fills up an 80GB Hard Disk in 16 Seconds

Higher energy levels also mean more data for the scientists to store and process. For example, during the Alice experiment, scientists will be causing collisions between heavy ions. The tracks of thousands of loaded and neutral particles need to be reconstructed for each collision. The corresponding data stem from the signals of various sub-detectors, organized like the skin of an onion. Due to the higher collision rates, LHC experiments generate about 40 GBits of data per second. In other words, they would take just 16 seconds to fill a normal 80GB hard disk. However, these experiments are scheduled to go on for many years. The data rates have also brought about changes in the computational infrastructure in particle physics. Where Unix and VMS machines, and occasionally mainframes, used to crush the numbers just a few years ago, the focus has shifted to Linux-based machine farms today. The free Linux operat-

ing system’s road to victory began way back in 1995 at CERN, when the first group of physicists installed Linux on the hard disks of its workstations.

Unfortunately, moving to Linux led to a few unexpected problems. Physicists want their programs to produce the same results wherever they run. Different distributions, and different versions of the same distribution will typically use different libraries and kernels. This in turn can affect the computational accuracy; in particular, mathlib has proved to be a critical point. This problem was previously unknown on homogeneous Unix platforms.

The Main Motivations for Grid Computing

The masses of data that the LHC is expected to generate place an enormous strain on the computational infrastructure. The applications need fast networks, arbitrary access to individual records, and an enormous computational capacity.

Instead of storing and processing data centrally, the LHC scientists intend to use existing or newly created computational resources belonging to the participating countries. The aim is to distribute the computational and storage load – this is one reason why the project is a major motivating factor behind Grid computing.

Wherever particle physics meets Grid computing, Linux proves to be a blessing and a curse at the same time. It continues to provide a cheap and stable platform, but the variety of distributions leads to more or less Babylonian scenarios at datacenters.