

An Overview of GRID Computing

By
Rüdiger Berlich
and
Dr. Marcel Kunze

Introduction

In the computing world there are many areas where technical development can't keep up with the demand for computational resources. Sometimes, workarounds used to overcome such deficiencies gain a life on their own and become the basis for new developments.

As an example, modern particle physics experiments, such as the upcoming LHC experiments [2] at CERN/Switzerland or the BaBar experiment at SLAC (Stanford, USA) [3] will, over the years, produce more data than can be realistically processed and stored in one location, even when using sophisticated cluster architectures. Predictions for the data production of the four LHC experiments are in the range of one Petabyte per experiment per year, or altogether a data rate of 40 GBit/s. What's more, as the experiments evolve and particle accelerators become more sophisticated, the predicted growth in data production over the years far exceeds the predicted growth of computing power. The latter is described by Moore's Law, according to which the processing power doubles every 18 months. So a local cluster of a given size won't be able to keep up with processing this data, even if it is constantly being upgraded to the newest technology.

In such a situation, one has but two choices: One can try to find additional monetary resources to frequently increase the computing and storage power in the location where the data originates. Or, one can try to use distributed computing- and storage- resources already available in participating institutions - particle physics experiments are international by design. Single countries today cannot afford the huge cost involved in building and maintaining particle detectors and accelerators any more.

While, from a technical perspective, maintaining and administering local computing resources is of course preferable over distributed approaches, it becomes

immediately clear that, in times of tight budgets, using available distributed resources is the only possible solution to the challenges imposed by modern particle physics.

It should be pointed out that the need to examine more data than can be realistically stored and processed in a single location is not particular to particle physics. Thus, an additional cost saving effect stems from the possibility to share distributed computing resources not only among physicists, but also with other research disciplines and business ventures.

The Vision

The lack of suitable, standardised solutions for distributed, large-scale computation has sparked a new research discipline, called GRID computing. The vision behind this new approach was first put forward by Ian Foster and Carl Kesselman in their book „The GRID Blueprint for a new computing infrastructure“[4]. In short, it could be described as „Computing Power from a Plug in the wall“. In the end, one shouldn't need to care for the location where data is being processed. Really important are speed, safety, and reproducibility of results. It is this obvious analogy with the electrical power grid that has also given the name to GRID computing: You do not need to know where electrical power is being produced. All that is important to you is that it is delivered to your home in a reliable way.

Distributed Computing

Distributed Computing is not a new paradigm. But up until a few years ago networks were too slow to allow efficient use of remote resources. But as the bandwidth of high-speed WAN's today even exceeds the bandwidth found in the internal links of commodity computers, it becomes inevitable that distributed computing is taken to a new level. It now becomes feasible to actually think of a set of computers coupled through a high-speed network as one large computational device. Or in other words: „The world is your computer“. Of course there are still limiting factors. The „speed“ of a network is a complex variable. It consists of the bandwidth (the number of bits you receive per second on one end of the network) and the latency (the amount of time it has taken these bits to travel from the source to the recipient. While you

can today scale the bandwidth of a network GRID computing is that you do not need to connection to virtually any level - provided know where your program is actually you can pay for it - there are physical limits to executed. So instead of choosing a machine its latency. Obviously, data cannot travel from a list, you need to describe the faster than the speed of light. So there is a requirements of your program in a way that lower limit to the amount of time needed to can be understood by some GRID component transfer the data, no matter how sophisticated responsible for choosing the target machine. your network hardware is. But since this data If you are handling sensitive data, you need to will have to pass repeaters and routers along know that no unauthorised party can gain the way, the actual latency will be much access to it. Likewise, the owner of the higher than the physical limit. E.g., the machine used to do your computation needs to latency across the USA is in the range of 50 know that you are using his hardware only in msec. Still, this is not a very high value. As a the way it is intended to. In short, there must comparison, a modern IDE hard drive with be a trust relationship between the person who 7200 RPM has mean access times of 8.5 submits the job and the owner of the target msec. So while latency does form a limiting machine. The complicated part is that these factor, and will continue to do so in the two people do not know each other and indeed foreseeable future, network latency is already should not need to have to interact in any way in the range of, say, the mean access times of in order to allow the job submission. old MFM hard drives.

Overcoming Latency

While latency is a limiting factor, there do machine. Usually this means copying some exist some possibilities to reduce its effects. data set over the network before transferring One example is a semi-commercial approach the program code. Alternatively, one could by Canada-based Canarie, Inc. [7], called the bring the program to the data rather than vice Wavelength Disk Drive (WDD). The idea is versa.

to use the network itself as a storage device. During and after the calculation you need to An optical network is used to form a loop, i.e., get access to the output created by your data stays in the network until it is removed program, so this information needs to be by some interested party. As the data doesn't transported back to you.

need to be transferred into the network Last, but not least, you need to pay for the anymore – the network *is* the storage device - computing time you've used. The cost can and since it is, with certain likelihood, already vary from very tiny amounts of money to close to its recipient, access times to data are huge sums, but in any case there must be an reduced. Unfortunately, the storage capacity accounting infrastructure across country and of such a device is limited to a few Gigabytes. currency boundaries.

But it is still sufficient to allow the usage of Most of these requirements of GRID Wavelength Disk Drives as cache-like computing could probably be satisfied using devices. One could think of other ways to existing tools. E.g., With a Virtual Private reduce latency, such as speculative copying of Network and batch submission systems such frequently used data, a technique often used in as PBS it would be possible to submit jobs on modern processors to overcome the speed remote machines in a secure and reliable way. difference between CPU-caches and main But while many tools for distributed memory.

Requirements of distributed computing

Imagine you want to submit a compute job to So the task at hand now is the creation of a the GRID. There are certain parameters you *standardised* software infrastructure suitable want to be sure of. First of all, you want to to the requirements of distributed computing. know that your job is submitted to a computer Collectively, efforts to create such an that fits the requirements of your program. infrastructure are today often referred to as Such requirements may include the processor „GRID computing“.

type, local storage capacity for temporary

files, amount of RAM and various other

hardware parameters. Still, the idea behind

Before program execution starts, any data needed by your program in order to do its job must be accessible to it from the target

machine. Usually this means copying some data set over the network before transferring the program code. Alternatively, one could bring the program to the data rather than vice versa.

During and after the calculation you need to get access to the output created by your program, so this information needs to be transported back to you.

Last, but not least, you need to pay for the computing time you've used. The cost can vary from very tiny amounts of money to huge sums, but in any case there must be an infrastructure across country and currency boundaries.

Most of these requirements of GRID computing could probably be satisfied using existing tools. E.g., With a Virtual Private Network and batch submission systems such as PBS it would be possible to submit jobs on remote machines in a secure and reliable way. While many tools for distributed computing are available, they do not form – and, more importantly, do not intend to form – a homogeneous approach.

So the task at hand now is the creation of a *standardised* software infrastructure suitable to the requirements of distributed computing. Collectively, efforts to create such an infrastructure are today often referred to as „GRID computing“.

The World Wide GRID

There is a striking resemblance of GRID

computing to the World Wide Web. The Web data between participating compute nodes was started at CERN in late 1990 by Tim during the computation, so the degree of Berners-Lee as a means of efficient parallelisation and thus the speedup depend on information exchange between physicists all the amount of data they need to exchange. over the world [5]. GRID computing aims at One could argue that, for this reason, GRID providing a means for efficient exchange of techniques are more suitable to Data GRIDs computing power and storage capacities than computational GRIDs, as the number of between physicists and, just like the Web, it possible applications tolerant to the owes many of its current features to work comparatively large latencies is limited. But done by computer scientists at CERN. we have also seen that the distinction between Just like it was the case in the beginning of the local machine and GRID becomes more and World Wide Web, there are currently many more blurred. Latency is today much less of special purpose GRIDs, which usually use the an issue than it was 5 years ago and there are Internet for data transportation. It'll take its interesting new developments in the field of time until these GRIDs grow together and latency tolerant algorithms. So while the form a World Wide GRID, but the ultimate majority of GRID applications can be goal is a global, standardised infrastructure for expected to be of the data GRID type, transparent execution of compute jobs across computing GRIDs will soon start to play an network boundaries. Mind you, we are not important role as well.

talking about every-day jobs here like, for

example, spell checking of text documents. **Hardware-Infrastructure**

The tasks likely to be executed over a GRID It shouldn't come as a surprise that some of will be huge analysis jobs like weather the main initiatives related to GRID forecasts or simulation of particle decays. As computing deal with the formation of high-described in the beginning, local clusters or speed networks and the provision of large workstations have become insufficient for clusters. Here are two of the more well-known such large-scale computation. efforts:

Data GRIDs vs. Computing GRIDs

So far we have offered the needs of particle physics as an explanation, why GRID Computing was started. Distributed data processing requires parallelisation. A typical particle physics analysis requires the analysis of millions of collisions of particles (or in short „events“). Usually, when processing one event, there is no need to have any information about the processing of another event. Thus, the analysis of a given set of events is an embarassingly parallel problem, as one only has to run the same analysis with a portion of the dataset on more than one compute node, and collect and assemble the results in the end. This is the typical situation found in data GRIDs, i.e. GRID environments tailored to the needs of the processing of huge data samples. In comparison, a computing GRID deals with the execution of parallel algorithms rather than the distributed processing of huge amounts of data. A

computing GRID could thus be described as a **Software-Infrastructure**

„Super-Cluster“, assembled from local In GRID computing, the link between clusters and single machines all over the applications and the physical GRID world. As we have described above, latency is infrastructure is provided by a „middleware“. the limiting factor for parallel computation. It is the middleware's task to addresses most Usually parallel algorithms need to exchange of the requirements of distributed computing

- Geant [8] is a four year project, set up by a consortium of 27 European national research and education networks, to form a fast, pan-European network infrastructure. It incorporates nine circuits operating at speeds of 10 Gbit/s plus eleven others running at 2.5 Gbit/s.

The TeraGrid [10] is an effort to build and deploy the world's largest and fastest distributed infrastructure for open scientific research. When completed, the TeraGrid will include 13.6 teraflops of Linux Cluster computing power distributed at the four TeraGrid sites, facilities capable of managing and storing more than 450 terabytes of data, high-resolution visualization environments, and toolkits for GRID computing. These components will be tightly integrated and connected through a network that will initially operate at 40 gigabits per second and later be upgraded to 50-80 gigabits/second.

mentioned above.

Due to the huge variety of projects, the following list can only be a subset of the middleware packages used in science:

- The most common software component in GRID computing today is the Globus toolkit [9].
- Cactus [11] is a higher-level middleware targeted more at computing GRID projects than data GRIDs.
- Legion [13] falls into the same category as Globus, but aims more at generating the illusion of a single, distributed computer.
- Unicore allows for seamless inter-operation of supercomputers over WAN
- The Sun Grid Engine is a commercial GRID middleware, incorporating many of the features of a load leveller

Traditionally, in local compute clusters the task of sending information from one node to another has been handled by the Message Passing Interface. There is an implementation of MPI, called MPICH-G [19] that uses the Globus toolkit for authentication. It doesn't fall into the category of middleware, but should nevertheless be mentioned here due to its importance in computing GRIDs. Basically it allows treating a GRID environment like a local cluster, albeit with a larger latency.

GRID Initiatives

Again this can only be an incomplete list.

- The **Global Grid Forum** [14] acts as a standardising body in GRID computing, similar to the Internet Engineering Task Force (IETF). Between two and three meetings per year aim at providing a forum for the discussion of new technical developments.
- The **European DataGrid project** [15] (EDG) was initially started with the needs of particle physics experiments in mind, but today incorporates many other research disciplines, including genome research. The project, which is funded by the European Union, aims at setting up a „computational and data-intensive grid of resources for the analysis of data coming from scientific exploration“. Like the name suggests, the EDG is purely targeted at data GRID applications.
- The **CrossGrid** [16] aims at providing an application framework for interactive computation on top of the infrastructure provided by the EDG.

- The Grid Physics Network, or in short **GriPhyN** [17], is the American counterpart to the EDG. It is mostly aimed at particle physics experiments.
- **LCG** aims at providing a virtual computing environment for the upcoming LHC experiments (CERN). It collaborates closely with Geant, EDG, GriPhyN and other GRID initiatives.
- **GridPP** [18] „is a collaboration of particle physicists and computer scientists from the UK and CERN, who are building a GRID for particle physics“.

GRID Computing and Linux

We have seen that there is a huge variety of GRID computing projects. Open research benefits from open platforms. As an example, it can be expected that, should GRID computing be successful, the middleware will become part of the Operating System. In an ideal GRID environment, users and authors of software packages should need to know as little as possible about GRID computing, just like today an ordinary user doesn't need to know anything about networking in order to send an email. This is nowadays often referred to by the term “Invisible Computing”. In order to develop an integrated, GRID-aware OS the free availability of the source code is mandatory. This eventually makes Linux (and other Open Source Operating Systems) the ideal platform for GRID computing. Linux already has a strong position in GRID computing, owing to the fact that GRID computing inherits many of its features and ideas from clustering. Linux is strong in this area.

While, thanks to the involvement of many commercial and non-commercial institutions, it is safe to conclude that GRID computing will play an important role in the future, it is not yet possible to exactly determine all areas in which GRID computing will have an impact on society. Competing (but lower-scale) efforts such as .Net, Mono, .Gnu or OneNet might fulfil some of the roles „traditionally“ assigned to GRID computing. Possibly even the naming conventions might change and „GRID computing“ might become a catch-all term for everything related to

Conclusion and Outlook

It should have become clear by now that GRID computing is still very much „work in progress“ and, in some ways, a buzz word. While, thanks to the involvement of many commercial and non-commercial institutions, it is safe to conclude that GRID computing will play an important role in the future, it is not yet possible to exactly determine all areas in which GRID computing will have an impact on society. Competing (but lower-scale) efforts such as .Net, Mono, .Gnu or OneNet might fulfil some of the roles „traditionally“ assigned to GRID computing. Possibly even the naming conventions might change and „GRID computing“ might become a catch-all term for everything related to

distributed computation. The important point is that a lot of work is being invested in the moment in order to take distributed computing to the next level. Similar to the World Wide Web, this might even be the next job-engine.

References

- [1] The “Transtec Compendium” www.transtec.co.uk
- [2] Information about the Large Hadron Collider at CERN lhc-new-homepage.web.cern.ch
- [3] Information about the BaBar experiment www.slac.stanford.edu/BFROOT/
- [4] The GRID, Blueprint for a new computing infrastructure; Ian Foster and Carl Kesselman, Morgan Kaufmann, ISBN 1-55860-475-8
- [5] The web : how it all began public.web.cern.ch/Public/ACHIEVEMENTS/web.html
- [6] OGSA Whitepaper - The Open GRID Services Architecture www.globus.org/research/papers.html
- [7] Information about the Wavelength Disk Drive www.ccc.on.ca/wdd
- [8] Geant – a pan-european network infrastructure www.dante.net/geant/aboutgeant.html
- [9] The Globus toolkit www.globus.org
- [10] An effort to create the world's largest and fastest distributed infrastructure for open scientific research www.teragrid.org
- [11] The Cactus middleware www.cactuscode.org
- [12] The Unicore middleware www.unicore.org
- [13] The Legion middleware legion.virginia.edu
- [14] The Global GRID Forum www.gridforum.org
- [15] Homepage of the EU DataGrid <http://eu-datagrid.web.cern.ch/eudatagrid/>
- [16] Information about the CrossGrid project www.crossgrid.org/
- [17] The GRID Physics Network www.griphyn.org/
- [18] The GRID for UK particle physics www.gridpp.ac.uk/
- [19] Homepage of MPICH-G2 www.unix.mcs.anl.gov/mpi/mpich/

The authors

From 1999 to 2001 Rüdiger Berlich was Managing Director of the UK office of SuSE Linux AG. He is now working in GRID research. (ruediger@berlich.de)

Dr. Marcel Kunze is the head of the department of GRID computing and eScience of Forschungszentrum Karlsruhe / Germany. (Marcel.Kunze@hik.fzk.de)

The authors would like to thank Forschungszentrum Karlsruhe University Bochum (www.ep1.ruhr-uni-bochum.de) and the German Federal Ministry of Education and Research (www.bmbf.de) for their support.