Cluster and Computational Grids for Scientific Computing

September 27-29, 2004
Le Château de Faverges de la Tour
FRANCE

Sponsored by

Operational Issues
and
Deployment Issues
for
Clusters and Grids

Philippe d'Anfray
CEA / DSI

Jack Dongarra
University of Tennessee and Oak Ridge National Laboratory

William Gropp
Argonne National Laboratory and University of Chicago

Bernard Tourancheau
Sun Microsystems laboratories

Summary

This report summarizes the activities and conclusions of the workshop "Cluster and Computational Grids for Scientific Computing," organized by Jack Dongarra and Bernard Tourancheau and held in Lyon in September 2004. The workshop included 31 invited presentations and a panel on deployment and operational issues for clusters and Grids. Several common themes emerged from the panel session and from the presentations.

Deployment issues

for clusters include power consumption and designing for scale as systems become ever larger. In addition, reducing the cost of operation through the use of well-designed, component-oriented system tools is becoming increasingly important as the number of nodes increases. Understanding the consequences of faults on cluster hardware and software is also critical. For Grids, the deployment issues include the issues faced by clusters and, in addition, issues of security and software interoperability. While several workable solutions have been suggested for providing security and a Grid software base, much work remains to be done. Current solutions are relatively low level and can be difficult to use. Further, when used in the context of virtual organizations (an organization made up of elements from multiple organizations formed to address a problem of shared interest), differences in policy and social interactions can dominate deployment issues. Technical solutions to these problems are not yet available.

Operational issues

for clusters and Grids focus on software. In both cases, applications are presented with relatively low-level software. In the case of clusters, however, the universal adoption of the MPI programming model and the support for higher-level software libraries provided by MPI has led to the availability of software that provides higher-level abstractions for the computational scientist. In the case of Grids, no such standards exist—and it is probably too early to develop them, although some valuable steps are being made, such as the SAGA interface. With Grids, moreover, even knowing which systems are available can be difficult. For example, basic TCP services may be provided on a resource but may not be available to the Grid software. Firewalls and other hardware and software security issues complicate the problem. Furthermore, the lack of quantitative data, rather than anecdotal evidence, makes it difficult to evaluate solutions.

The rest of this report summarizes the presentations of the workshop and comments on the contrasting state of the art in clusters and Grids. The presentations are available at http://www.cs.utk.edu/~dongarra/Lyon-2004-slides/program-sessions.htm .

Clusters

Clusters are, not unexpectedly, a much more successful and mature environment than are Grids. For example, over 50 percent of the Top500 systems are now clusters (in fact, they are running just one operating system, Linux). Issues include performance, power consumption, scale, I/O, programming abstractions, and fault tolerance. In this section, we summarize the presentations in these areas.

Performance

Jeff Hollingsworth, "Hardware Performance Monitors: Beyond Counting Events," described the current state of hardware and software support for accessing and exploiting detailed data about processor performance. For parallel programs on clusters, similar though less detailed data is available.

Scale

Craig Forrest, "Sun's HPCS Project and Challenges for Building Petascale Computers," spoke about how scalability is driving everything else (e.g., power, performance), especially with single address space systems.

Dan Reed, "The Challenge of Scale," discussed the emerging challenges for systems with 100K and more processors. Six areas were highlighted: power consumption, fault tolerance, reliability, availability, serviceability, and software complexity.

Matsuoka Satoshi, "The JST-CREST MegaScale Project," looked at ways in which petaflops computing can be achieved. While the focus of this talk was on the "megascale cluster," the extension into federated clusters was introduced. Also discussed were scalable checkpointing, techniques for low power, and a nearly all-commodity high-density cluster.

I/O

Stephen Wheat, "Cluster and Computational Grids for Scientific Computing: 2004," raised the issue of storage growth and the tighter integration of storage with the memory system. Also, even though this was primarily a hardware presentation, the key role of software was the concluding point.

Programming Abstractions

Guy Steele's paper "Design Themes for a High Productivity Programming Language for Scientific Computing, and Some Questions" (presented by Danny Cohen), focused on three key themes: "making stupid mistakes impossible," "design the language to be grown by users," and "emulating standard math notation." The presentation raised hard questions about the pros and cons of the global and local points of view (global is high abstraction but local appears necessary for performance in general).

Roldan Pozo, "The Role of Virtual Machine Technologies for Scientific Computing," focused on advances in the performance of Java and the .NET environment for numerical computing.

Jean-Yves Berthou, "Numerical Simulation at EDF," described some of the simulation needs of EDF, particularly multiscale physics and ways to couple applications and models.

Denis Caromel, "Beating Fortran MPI with Java ProActive," described advances in using distributed objects and futures with Java to provide a higher-level programming abstraction for distributed scientific computing. He presented experiments on clusters.

System Issues (Fault Tolerance, Interconnects, Software)

Rusty Lusk, "An Interoperability Approach to Systems Software, Tools, and Libraries," described a component-based approach to building scalable cluster systems software as part of the DOE Scalable Systems Software SciDAC project. A (relatively) strict definition of component is followed, which includes interchangeability. He noted that one strength of clusters has been the use of commodity components, which has kept prices low through competition and has accelerated development of tools as groups (both research and commercial) compete on performance and robustness in implementing standards such as C and MPI.

Al Geist, "A New Paradigm for Large-Scale Science: Computational End Stations," described a new deployment model that integrates application and software expertise and matches it to a computational resource.

Rich Graham, "OpenMPI: First Experiences with Mixed Network Communications," described the use of multiple network information connections, or NICs.

Franck Cappello, "MPICH-V: A Multi-Protocols Fault Tolerant MPI," provided a detailed and quantitative discussion of techniques for fault tolerance in a cluster setting. A standard set of benchmarks that NAS parallel benchmarks is using. He noted that there is no similar set of Grid benchmarks for data-centric computing, and most existing Grid benchmarks is too sensitive to the performance of the "end-points"—the compute resources contributed to the Grid, rather than the Grid services themselves.

Patrick Geoffray, “Hardware Folks Are from Mars, Software Guys Are from Venus: Design Choices for Myrinet/MX,” described the mismatch between the hardware and software models for moving data between systems. He mentioned, in particular, the challenges in using remote memory operations for programming models such as MPI.

Pete Beckman, “Attack of the Killer Operating Systems,” discussed the opportunities for exploiting commodity operating systems such as Linux as the foundation for highly scalable clusters and what sort of additions or changes will be necessary to support extreme-scale systems. The DOE FastOS project will be looking at scalable operating system issues, including testbeds for studying scale and fault-tolerance issues and novel approaches to scalability, such as collective and coalesced system calls.

Grids

Significant successes have occurred in Grid applications, especially where a specific goal has been combined with careful design and realistic expectations. Grids that unite data sources and data users have been particularly successful. Issues include the low level of programming abstraction, the mismatch between user needs and Grid capabilities, the immature software environment, the lack of testbeds, and the lack of a quantitative character to much of Grid research. The presentations in this area, reflecting the state of the art, tended to cover multiple issues and could not be divided up among these topics.

Franck Cappello, "MPICH-V: A Multi-Protocols Fault Tolerant MPI," He noted that there is no set of Grid benchmarks, like the standard set of benchmarks : NAS parallel benchmarks, for data-centric computing, and most existing Grid benchmarks is too sensitive to the performance of the "end-points"—the compute resources contributed to the Grid, rather than the Grid services themselves.

Dan Fay, "HPC, Web Services, Grids: eScience@Microsoft," described the importance of an "applications ecology," with tools to enhance developer productivity. Fay asked, "Are the right architectural pieces in place?" (and "How will we know?").

Andrew Grimshaw, "The Global Bio Grid," presented an example of a carefully designed Grid for a particular data-centric application, exploiting existing code. I/O optimizations matched to the database operations (rather than a simpler, general solution) contribute to the success of this Grid.

Fran Berman, "Grid Computing, a Midterm Evaluation," surveyed one part of the community and found both good and bad results. *Bad:* The Grid concept was oversold; a solid scientific discipline is lacking; the technical difficulties were underestimated, and the models and architecture are inadequate, with the result that Grids are still too hard to use. Grids are seen as a "solution in search of a problem." *Good:* The vision of Grids is commendable, a dedicated community has emerged, and functional software has been developed. Two significant issues remain unresolved: authentication and security

Jeff Hollingsworth, "Hardware Performance Monitors: Beyond Counting Events," described the current state of hardware and software support for accessing and exploiting detailed data about processor performance. For Grids, the level of available detail is much lower, and there is no clear agreement on what is the appropriate data (though this is now being discussed by the Grid community through workshops such as the Grid Performance Workshop, held earlier in 2004).

Ed Seidel, "Tools for Developing Grid Applications," looked at higher-level abstractions for Grid users, including a good example of the low level of current abstractions: remote file copy. He described SAGA, a simple, application-user-oriented API for Grids.

Craig Lee, "Operational and Deployment Issues Associated with a NASA Grid Project and Thoughts Thereon" (panel presentation), emphasized security, trust, and cross-domain issues and policies.

Bill Gropp, “Grids and Clusters: Lessons for Deployment and Operation” (panel presentation), discussed some of the reasons for the success of clusters, including engineered solutions (matching resources to need), scientific studies with reproducible measurements, and software solutions that scale to users (that is, users can acquire and use the software without help from the software developers). Clusters have benefited from extensive testing and evaluation; Grids need the same challenges and support.

Tony Hey, “Issues with Production Grids” (panel introduction), described some successes and challenges, along with an approach based on Web service Grids and workflows.

Micah Beck, “Achieving Deployment Scalability” (panel presentation), described principles for deploying resources through a unified view of data transfer, storage, and processing called Internet Backplane Protocol (IBP).

Philip Papadopoulos, “OptiPuter: The Impact of Bandwidth on Distributed System Design,” looked at how the rapid growth in bandwidth enabled by optical networks might change the way applications are built and the type of applications that are possible. This talk may represent the longest-term view presented at this meeting.

Kenichi Miura, “National Research Grid Initiative (NAREGI) Project,” described Grid-related projects in Japan, including Grid software and middleware, a testbed for high-end Grid, and specific Grid-enabled applications.

Andrew Chien, “Realistic Online Simulation of Large-Scale Grids,” described work in simulating Grids at several levels of detail. He also presented results, including the effect of changing router buffer sizes on resisting a denial-of-service attack.

Denis Caromel, “Beating Fortran MPI with Java ProActive,” described advances in using distributed objects and futures with Java to provide a higher-level programming abstraction for distributed scientific computing. He presented experiments on a grid of clusters.

Thomas Lippert, “Unicore: From Project Results to Production Grids,” described experiences with UNICORE and its evolution to a uniform interface to Grid services.

Wolfgang Gentzsch, “Best Practice: MCNC Grid Computing and Networking Services (GCNS),” described the approach of the Microelectronics Center of North Carolina to building and deploying Grid applications, building on existing components. He recommended starting with specific application projects and building testbeds before production platforms.

Cherri Pancake, “Making CyberInfrastructure Accessible (and Appealing?) to Users: Case Studies from Engineering,” discussed the lessons from three projects and pointed out the mismatch between computer science providers (interested in research) and users (interested in rock-solid infrastructure). She emphasized that we need to listen to users and address their concerns, rather than consider them a threat to the vision (one of the reasons for the success of clusters has been a recognition that the customer is always right, even when wrong).

John Morrison, “WebCom-G on Grid-Ireland (A VM Approach to Hiding the Grid),” described a graph-based model for distributed computing where security via trust management is a major component.

Pascale Primet, “The French ACI GRID Initiative,” described a nationwide instrument for Grid researchers in France, targeting issues in programming, scalability, networking, quality of service, applications, and new approaches for IP based protocols.

Philippe d’Anfray, “Grids: Users’ Feedback from CEA,” described experiences with applications tests. He pointed out that experiments cannot be conducted entirely locally because of the unique features of the Grid, including availability and trust among the actors. Other issues included reliability and performance, deployment and maintenance of the middleware (including version management), fault-tolerance, programming, and resource accounting.

Joel Saltz, “Truth Telling about Large Scale Data,” described an example of a large-scale image database and techniques for providing efficient access (including the use of memory hierarchy concepts). He emphasized the important of end-to-end performance characterization and the need for Grid testbeds.

Henri Casanova and Yves Robert, “A Realistic Network/Application Model for Scheduling Divisible Loads on Large-Scale Platforms,” described recent research into scheduling of certain types of applications. They introduced a novel model that captures some of the basic parameters of the Grid networked platforms and heuristic solutions to achieve very good performance on the scheduling of tasks over a given platform. The corresponding optimization problem is NP-complete but the presented heuristics have reasonable computation time complexities.

Frederic Desprez, “Recent Advances in DIET,” describes a toolbox for implementing distributed computations and applies it to solving sparse linear systems. The advantage here is not in performance (the time to describe the problem over the Grid can exceed the solution time on a PC cluster) but in the ability to deliver access to sophisticated and powerful software to users.

Thierry Priol, “Objects, Components, Services for Grid Middleware: Pros and Cons,” discussed the Web service model in the context of the earlier distributed object model as used in CORBA. He pointed out the lessons learned in CORBA (particularly the negative consequences of partial implementations of the standard, breaking interoperability) and other Object -based or Component-based approaches which provides a mature view of the Grid middleware. While the low level of abstraction in most current Services oriented approach might be an issue for Grid middleware.

Clusters and Grids: Comparing and Contrasting

One feature that has benefited clusters is that a small cluster (in the limit, even a single machine running multiple processes) can serve as a testbed for cluster software. This approach is much more difficult to accomplish for Grids because of the additional issues faced in Grid computing. Many opportunities do exist, however, to exploit the success of clusters in advancing the state of Grid computing. Three such opportunities were represented at this meeting:

1 Clusters as Grid testbeds
2 Clusters as Grid resources
3 Clusters as a model for Grid evolution (e.g., as a model for software development, increasing levels of abstraction in the programming model, model for usage)

Clearly, the development path of clusters can provide a useful model for Grids. The successes of Grid applications have also underscored the fact that Grids bring unique challenges that will need their own solutions. As this workshop repeatedly indicated, much remains to be done.