

Network-enabled Solvers: A Step Toward Grid-based Computing

December 13, 2001

Jack Dongarra

Two statements have been consistently true in the realm of computer science: (1) the need for computational power is always greater than what is available at any given point, and (2) to access our resources, we always want the simplest, yet most complete and easy to use interface possible. With these conditions in mind, researchers have directed considerable attention in recent years to the area of "grid computing." The ultimate goal is the ability to plug any and all of our resources into a computational grid, analogous to the electric power grid, drawing from these resources much as we plug our appliances into electric sockets today.

New Classes of Applications

The computational grid will be inherently more complex than existing computer systems, and programs designed for grid execution will reflect some of this complexity. Making grid resources useful and accessible to scientists and engineers will thus require new software tools that embody major advances in both the theory and the practice of building applications. Developers of grid middleware will target key challenges at the infrastructure level: security, resource discovery, resource management, and power management. The goal of this work is to simplify distributed heterogeneous computing much as the Web has simplified information-sharing via the Internet.

The grid will make it possible to implement dramatically new classes of applications. These applications range from new systems for scientific inquiry, through computing support for crisis management, to support for personal-life management.

Imagine remote, biodegradable sensors in the ocean, monitoring temperature, biological materials, and key chemical concentrations, and transmitting the measurements via wireless technology to digital libraries of oceanographic data. After mining and visualizing the data directly to derive new insights, scientists would use the refined data in large-scale predictive models. The sensors would then be redeployed to refine the system as a result of the predictions, and, in a final step, nanoactuators would be triggered to remove inappropriate concentrations of effluent or other non-native materials.

Imagine next an engineering system that integrates "teleobservation" and "teleoperation" to enable earthquake researchers to control experimental tools-seismographs, cameras, or robots at remote sites. Combining real-time, remote access to data generated by those tools, along with video and audio feeds, large-scale computing facilities for coupled simulation, data archives, high-performance networks, and structural models, researchers would be able to improve the seismic design of buildings, bridges, utilities, and other infrastructure around the world.

As a final example, imagine a personal digital assistant integrated into your eyeglasses, powered by body heat and capable of calling on ambient computing, information, and network resources.

As you entered a building, your personal information space would be available to you; local computing power would offload such tasks as face recognition, translation, and navigation, and you could be simultaneously monitoring your latest earthquake engineering experiment-or your stock portfolio.

These examples illustrate three themes that will be dominant in grid computing:

- Computing resources, no longer localized, will be distributed-and hence heterogeneous and dynamic.
- Computation will be increasingly sophisticated and multidisciplinary.
- Computation will be integrated into our daily lives, and hence subject to stricter time constraints than at present.

Grid concepts are being studied aggressively by many groups and are at the heart of major application projects and infrastructure deployment efforts, such as NASA's Information Power Grid, NSF's Partnerships for Advanced Computational Infrastructure (PACI) National Technology Grid and Distributed Terascale Facility, NSF's Grid Physics Network, and the European Union's European Data Grid and Eurogrid projects.

At the University of Tennessee and Oak Ridge National Laboratory, we have been working since 1995 on an approach to grid computing called NetSolve [1]. NetSolve gives users easy access to computational resources, both hardware and software, that are distributed across a network with respect to both geography and ownership. NetSolve searches for computational resources on the network, chooses the best one available, and (using retry for fault-tolerance) solves a problem and returns the answers to the user.

Network-enabled Solvers

The NetSolve project is one of several successful efforts under way to actualize the concept of computational grids. Our original motivation was to alleviate the difficulties usually encountered by domain scientists as they attempt to locate/install/use numerical software, especially on multiple platforms.

NetSolve has a client-agent-server design: Clients issue requests to agents, which allocate servers to service those requests; the servers then receive input for the problems, do the computations, and return the output parameters to the clients. A NetSolve user can gain access to "limitless" software resources without the tedium of installation and maintenance. NetSolve facilitates remote access to both software resources and hardware, possibly high-performance supercomputers, with complete opacity from the user's perspective. The NetSolve user needs no knowledge of computer networking and the like-does not, in fact, even have to be aware that remote resources are involved. Features like fault tolerance and load balancing further enhance the NetSolve system. As an example, consider a researcher who would like access to the PETSc collection of iterative solvers, without installing the software on his/her machine; by making a call to NetSolve, the researcher gains access to the PETSc software, as well as to hardware resources, to solve a specified problem.

The NetSolve agent---the gateway to the NetSolve system---maintains a database of servers, along with their capabilities (hardware performance and allocated software) and usage statistics. Using this information, the agent allocates server resources for client requests. In its resource-allocation mechanism, the agent balances load among its servers; it is also the primary component concerned with fault tolerance.

The NetSolve server, the computational backbone of the system, is a daemon process that awaits client requests. The server can be run on all popular strains of the UNIX operating system and has been ported to run on almost any architecture---it has run on single workstations, clusters of workstations, and shared-memory multiprocessors. It gives the client access to software resources and also provides mechanisms for integrating any software with NetSolve servers.

The NetSolve client user submits requests (possibly simultaneous multiple requests) and retrieves results from the system via the API provided for the language of implementation. NetSolve currently supports the C, FORTRAN, Matlab, and Mathematica programming interfaces. The functional interface completely hides all networking activity from the user. NetSolve version 1.4 can be downloaded from the project Web site at <http://icl.cs.utk.edu/netsolve/>.

As mentioned earlier, NetSolve is one of many research projects under way in the area of grid-based computing. A good resource on activity in the area is *The Grid: Blueprint for a New Computing Infrastructure*, edited by Globus originators Ian Foster and Carl Kesselman [2].

Conclusions

The scientific community has long used the Internet for e-mail, and for communication of software and papers. Until recently, though, there has been little use of the network for actual computations. This situation is changing rapidly, and developments will have an enormous impact on the future. NetSolve, as briefly described in this article, is an environment for networked computing whose goal is to deliver the power and/or software resources of computational grid environments to users who need these resources, but are not expert computer scientists. It achieves this goal with its three-part client-agent-server architecture.

References

- [1] H. Casanova and J. Dongarra, *Applying NetSolve's network enabled server*, IEEE Comput. Sci. & Eng., 5:3 (1998), 57-66.
- [2] I. Foster and C. Kesselman, eds., *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufman Publishers, San Francisco, 1998.

Jack Dongarra is a professor of computer science at the University of Tennessee, Knoxville. He was also an author of an article on the "Top-500" report, a twice-yearly list of the sites at which the 500 most powerful computer systems in the world are installed, which appeared in the [November issue of SIAM News](#).

The print version of this article (*SIAM News*, December 2001, page 4) was erroneously attributed to Jack Dongarra, Hans Meuer, Horst Simon, and Erich Strohmaier. Dongarra was the sole author of the article.

SIAM News regrets any inconvenience caused by the error.