Invited Talks

High-Performance Distributed Computing: The I-WAY Experiment and Beyond

Ian Foster[1]

Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439
foster@mcs.anl.gov
http://www.mcs.anl.gov/globus/

Recent developments in networking are enabling innovative applications that integrate geographically distributed high-performance computing, database, display, and networking resources. However, there is as yet little understanding of the higher-level services needed to support these applications, or of the techniques required to implement these services in a scalable, secure manner. In this brief paper, I describe the I-WAY networking experiment, a large-scale wide-area computing testbed that has been used to investigate these issues. I also introduce the Globus project, a multi-institutional effort that is developing key technologies for I-WAY–like systems, including mechanisms for resource location, scheduling, authentication, and automatic configuration of high-performance distributed computations.

1 High-Performance Distributed Computing

High-performance distributing computing, or *metacomputing* as it is sometimes called, refers to the use of high-speed networks to connect supercomputers, databases, scientific instruments, and advanced display devices located at geographically remote sites [3]. In principle, metacomputing can both increase accessibility to supercomputing capabilities and enable the assembly of unique capabilities that could not otherwise be created in a cost-effective manner.

Experience with the I-WAY and other high-speed networking testbeds has provided convincing demonstrations that there are indeed applications of considerable scientific and economic importance that can benefit from access to high-performance distributed computing capabilities. Many of these applications fall into the following four general classes.

1. *Desktop supercomputing.* These applications couple high-end graphics capabilities with remote supercomputers and/or databases. This coupling connects users more tightly with computing capabilities, while at the same time achieving distance independence between resources, developers, and users.
2. *Smart instruments.* These applications connect users to instruments such as microscopes, telescopes, or satellite downlinks [17] that are themselves coupled with remote supercomputers. By allowing both quasi-realtime processing of instrument output and interactive steering, the utility of the instrument can be increased significantly.

3. *Distributed supercomputing.* More traditional supercomputing applications couple multiple, geographically distributed supercomputers in order to tackle problems that are too large for a single supercomputer or that can benefit from executing different problem components on different computer architectures [22, 19, 23].
4. *Collaborative environments.* A fourth set of applications couple multiple virtual environments so that users at different locations can interact with each other and with supercomputer simulations [8, 7].

Applications in the first and second classes are prototypes for future "network-enabled tools" that enhance local computational environments with remote compute and information resources; applications in the fourth class are prototypes of future collaborative environments.

2 The Globus Project

High-performance, geographically distributed computing requires tools for requesting, locating, scheduling, and programming diverse computational and network resources; for authenticating users, authorizing access to resources, and protecting the security of user computations; and for accessing both shared data and user file systems from geographically remote locations. These tools must scale to meet the requirements of computations that link tens or hundreds of resources located in multiple administrative domains and connected using networks of widely varying capabilities.

The Globus project is a multi-institutional research and development activity that is addressing these problems. Its goal is to provide software technologies that support the dynamic identification and composition of resources available on national-scale internets, and that provide mechanisms for authentication, authorization, and delegation of trust within environments of this scale.

In order to support the dynamic composition of computational and information resources, we are investigating the following topics.

- *Resource location*: uniform and scalable mechanisms for naming and locating computational and communication resources on remote systems, and for incorporating these resources into parallel and distributed computations.
- *Protocol and resource management*: scalable techniques for locating available network connections, making choices between alternatives according to their service type and security level, integrating chosen networks into running computations.
- *Resource-aware programming tools*: versions of high-level libraries and languages, such as MPI [16, 12], CC++ [4], Fortran M [9], and HPF, that allow programmers to specify high-performance distributed computations in a portable manner, while also providing access to low-level information when this is required for performance.

In the security area, we are focusing on two problems.

- *Data Access*: techniques for providing computations with uniform, efficient, and secure access to files. In particular, new protocols and algorithms that allow secure distributed file systems [5] to function efficiently in large-scale internetworked environments.
- *Authentication and Authorization*: authorization and access control mechanisms that provide fine-grain control over access to communication, computational, and information resources. Also, techniques based on delegation of trust for managing trust relationships and access control in large and dynamically-changing user communities across multiple administrative domains.

These basic techniques are being incorporated in a prototype software system for constructing high-performance distributed computations in national-scale internetworked environments. Preliminary versions of this software were used in the I-WAY networking experiment to support extensive experiments in wide area supercomputing.

3 I-WAY and I-Soft

The I-WAY, or Information Wide Area Year [6], was a wide-area computing experiment conducted throughout 1995 with the goal of providing a large-scale testbed in which innovative high-performance and geographically-distributed applications could be deployed. The I-WAY linked 11 existing national testbeds based on ATM (asynchronous transfer mode) technology to interconnect supercomputer centers, virtual reality research locations, and applications development sites across North America (Figure 1). When demonstrated at the Supercomputing conference in San Diego in December 1995, it connected supercomputers, mass storage systems, and advanced visualization devices at 17 different sites. This distributed supercomputing environment was used by over 60 application groups for experiments in high-performance computing (e.g., [22, 23]), collaborative design [7], and the coupling of remote supercomputers and databases into local environments (e.g., [17]). A primary thrust was applications that use multiple supercomputers and virtual reality devices to explore collaborative technologies in which shared virtual spaces are used to perform computational science. For the purposes of this experiment, all communication was performed by using standard IP protocols running over ATM Adaptation Layer 5 (AAL5).

The I-WAY experiment was intended not only as an opportunity for large-scale application experiments, but also as a testbed within which solutions to various software infrastructure problems could be deployed and studied in a somewhat controlled environment. Because the number of users (few hundred) and sites (around 20) were moderate, issues of scalability could, to a large extent, be ignored. However, issues of security, usability, and generality were of critical concern. Important secondary requirements were to minimize development and maintenance effort, both for the I-WAY development team and the participating sites and users.

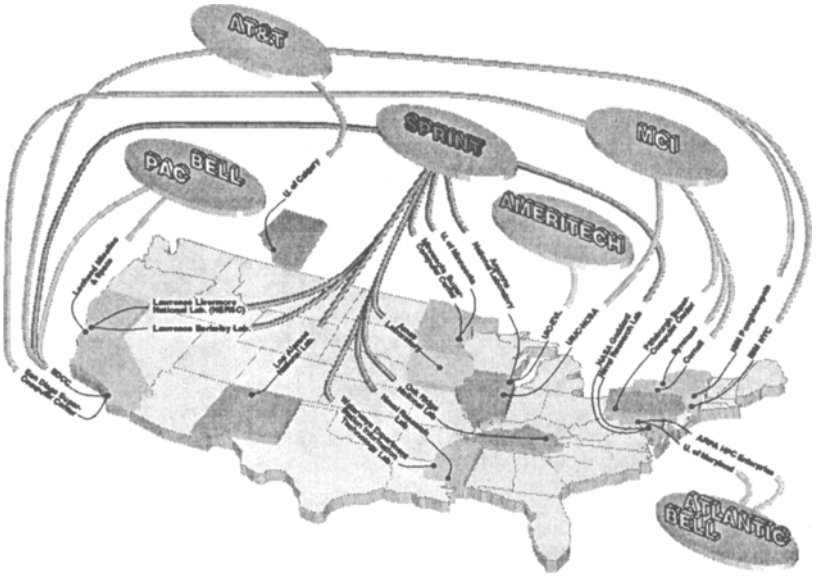

Fig. 1. The I-WAY network. Figure produced by Linda Winkler and Richard Foster.

To this end, members of the Globus project worked in collaboration with researchers and programmers at various I-WAY sites to develop a system management and application programming environment called I-Soft [11] that provided uniform authentication, resource reservation, process creation, and communication functions across I-WAY resources. A novel aspect of our approach was the deployment of a dedicated I-WAY Point of Presence, or I-POP, machine at each participating site (Figure 2). These machines provided a uniform environment for deployment of management software, and also simplified validation of system management and security solutions by serving as a "neutral" zone under the joint control of I-WAY developers and local authorities.

The task of developing software systems for environments such as the I-WAY is complicated by the fact that resources and users exist at different sites and in different administrative domains. Different sites have different access mechanisms for their resources, and cannot be expected to relinquish control to an external authority. Hence, the problem of developing management systems is in large part one of defining protocols and interfaces that support a negotiation process between users (or brokers acting on their behalf) and the sites that control the resources that users want to access. I-Soft addressed this issue by providing a simple computational resource broker that used scheduler proxies to provide a uniform scheduling environment integrating diverse local schedulers, and by

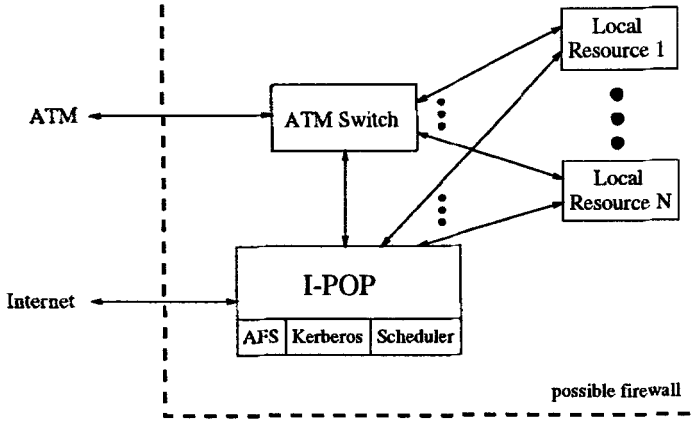

Fig. 2. An I-WAY Point of Presence (I-POP) machine

using authorization proxies to construct a uniform authentication environment and define trust relationships across multiple administrative domains [11]. I-Soft also addressed issues of system heterogeneity by providing resource-aware parallel programming tools based on the Nexus runtime system [13]. These tools used configuration information regarding topology, network interfaces, startup mechanisms, and node naming to provide a uniform view of heterogeneous systems and to optimize communication performance [10]. These various I-Soft services allowed a user to log on to any I-POP and then schedule resources on heterogeneous collections of resources, initiate computations, and communicate between computers and with graphics devices—all without being aware of where these resources were located or how they were connected.

The I-WAY experiment proved extremely useful as a means of identitying truly important issues in wide-area high-performance computing. In particular, we learned that system components that are typically developed in isolation must be more tightly integrated if performance, reliability, and usability goals are to be achieved. For example, resource location services in future I-WAY–like systems will need low-level information on network characteristics; schedulers will need to be able to schedule network bandwidth as well as computers; and parallel programming tools will need up-to-date information on network status.

4 Related Work

The Globus and I-WAY projects build on the results of considerable previous work in distributed computing, parallel computing, and high-speed networking. To name just a few examples, Condor [18], Nimrod [1], and Prospero [21] address the problem of locating and/or accessing distributed resources; AFS [20] and

DFS address problems of sharing distributed data; and MPI [16], PVM [14], and Isis [2] address problems of coupling distributed computational resources.

The Distributed Computing Environment (DCE) and Common Object Request Broker Architecture (CORBA) are two major industry-led attempts to provide a unifying framework for distributed computing. Both define (or will soon define) a standard directory service, remote procedure call (RPC), security service, and so forth; DCE also defines a Distributed File Service (DFS) derived from AFS. Some DCE mechanisms (RPC, DFS) may well prove to be appropriate for implementing I-POP services; CORBA directory services may be useful for resource location. However, both DCE and CORBA appear to have significant deficiencies as a basis for application programming in I-WAY–like systems. In particular, the remote procedure call is not well-suited to applications in which performance requirements demand asynchronous communication, multiple outstanding requests, and/or efficient collective operations.

Two other research projects are addressing similar themes and goals. The Legion project at the University of Virginia [15] and the Globe project at the Vrije Universiteit in Amsterdam [24] both seek to develop a universal, object-based software infrastructure for computing in wide area environments; research topics include scheduling, file systems, security, fault tolerance, and network protocols. The Globus and I-WAY efforts are distinguished by their focus on high-performance systems and applications, in which the efficient management and scheduling of high-speed networks are central concerns.

5 Future Directions

The Globus project is now addressing various issues identified in the course of I-Soft development. Particular focus areas include resource location, scheduling, automatic configuration, scalable trust management, and resource-aware tools and applications. In addition, we are working with colleagues to define and construct a "persistent I-WAY" that will provide further opportunities for application experiments and evaluation of wide area management and application programming tools. We hope to extend the scope of the I-WAY to include international connections in the near future.

Acknowledgments

The Globus project is led by the author and Carl Kesselman of the California Institute of Technology. The I-WAY project is centered at Argonne, the National Center for Supercomputer Applications, and the Electronic Visualization Laboratory. This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38. Future work on Globus will be supported by DARPA.

References

1. D. Abramson, R. Sosic, J. Giddy, and B. Hall. Nimrod: A tool for performing parameterised simulations using distributed workstations. In *Proc. 4th IEEE Symp. on High Performance Distributed Computing.* IEEE Press, 1995.
2. K. Birman. The process group approach to reliable distributed computing. *Communications of the ACM*, 36(12):37–53, 1993.
3. C. Catlett and L. Smarr. Metacomputing. *Communications of the ACM*, 35(6):44–52, 1992.
4. K. M. Chandy and C. Kesselman. CC++: A declarative concurrent object oriented programming notation. In *Research Directions in Object Oriented Programming.* The MIT Press, 1993.
5. J. Cook, S.D. Crocker, Jr. T. Page, G. Popek, and P. Reiher. Truffles: Secure file sharing with minimal system administratore intervention. In *Proc. SANS-II, The World Conference On Tools and Techniques For System Administration, Networking, and Security.* 1993.
6. T. DeFanti, I. Foster, M. Papka, R. Stevens, and T. Kuhfuss. Overview of the I-WAY: Wide area visual supercomputing. *International Journal of Supercomputer Applications*, 1996. in press.
7. D. Diachin, L. Freitag, D. Heath, J. Herzog, W. Michels, and P. Plassmann. Remote engineering tools for the design of pollution control systems for commercial boilers. *International Journal of Supercomputer Applications*, 1996. To appear.
8. T. L. Disz, M. E. Papka, M. Pellegrino, and R. Stevens. Sharing visualization experiences among remote virtual environments. In *International Workshop on High Performance Computing for Computer Graphics and Visualization*, pages 217–237. Springer-Verlag, 1995.
9. I. Foster and K. M. Chandy. Fortran M: A language for modular parallel programming. *Journal of Parallel and Distributed Computing*, 26(1):24–35, 1995.
10. I. Foster, J. Geisler, C. Kesselman, and S. Tuecke. Multimethod communication for high-performance metacomputing applications. Preprint, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., 1996.
11. I. Foster, J. Geisler, W. Nickless, W. Smith, and S. Tuecke. Software infrastructure for the I-WAY high-performance distributed computing experiment. In *Proc. 5th IEEE Symp. on High Performance Distributed Computing.* IEEE Computer Society Press, 1996.
12. I. Foster, J. Geisler, and S. Tuecke. MPI on the I-WAY: A wide-area, multimethod implementation of the Message Passing Interface. In *Proceedings of the 1996 MPI Developers Conference.* IEEE Computer Society Press, 1996.
13. I. Foster, C. Kesselman, and S. Tuecke. The Nexus approach to integrating multithreading and communication. *Journal of Parallel and Distributed Computing*, 1996. To appear.
14. A. Geist, A. Beguelin, J. Dongarra, W. Jiang, B. Manchek, and V. Sunderam. *PVM: Parallel Virtual Machine—A User's Guide and Tutorial for Network Parallel Computing.* MIT Press, 1994.
15. A. Grimshaw, W. Wulf, J. French, A. Weaver, and P. Reynolds, Jr. Legion: The next logical step toward a nationwide virtual computer. Technical Report CS-94-21, Department of Computer Science, University of Virginia, 1994.
16. W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message Passing Interface.* MIT Press, 1995.

17. C. Lee, C. Kesselman, and S. Schwab. Near-real-time satellite image processing: Metacomputing in CC++. *Computer Graphics and Applications*, 1996. to appear.
18. M. Litzkow, M. Livney, and M. Mutka. Condor - a hunter of idle workstations. In *Proc. 8th Intl Conf. on Distributed Computing Systems*, pages 104–111, 1988.
19. C. Mechoso et al. Distribution of a Coupled-ocean General Circulation Model across high-speed networks. In *Proceedings of the 4th International Symposium on Computational Fluid Dynamics*, 1991.
20. J.H. Morris et al. Andrew: A distributed personal computing environment. *CACM*, 29(3), 1986.
21. B. Clifford Neumann and Santosh Rao. The Prospero resource manager: A scalable framework for processor allocation in distributed systems. *Concurrency: Practice and Experience*, June 1994.
22. J. Nieplocha and R. Harrison. Shared memory NUMA programming on I-WAY. In *Proc. 5th IEEE Symp. on High Performance Distributed Computing.* IEEE Computer Society Press, 1996.
23. M. Norman et al. Galaxies collide on the I-WAY: An example of heterogeneous wide-area collaborative supercomputing. *International Journal of Supercomputer Applications*, 1996. in press.
24. M. van Steen, P. Homburg, L. van Doorn, A. Tanenbaum, and W. de Jonge. Towards object-based wide area distributed systems. In *Proc. International Workshop on Object Orientation in Operating Systems*, pages 224–227, 1995.