A Secure Distributed MPP Scheduling System for Tera/PetaFLOP Computing

March 16, 1995

Abstract

Over the past two years the Mathematics and Computer Science Division at Argonne National Laboratory has developed an extensible scheduling system for MPP machines such as the IBM SP architecture. One of this scheduling system's most valuable features is that it has isolated the job-scheduling aspect of a scheduling system from the popular message-passing systems, security systems, and various job-starting mechanisms that are currently available. Thus the Argonne Scheduler is not tied to any particular message-passing library but can provide scheduling services for any message-passing system, including PVM, p4, and MPI, the new message-passing standard. The design of the scheduler also allows the integration of various standard security packages without a complete rewrite. Thus this scheduling system can easily keep pace with emerging message-passing and security systems. We propose to convert the Argonne Scheduler to a distributed MPP scheduling system that uses Kerberos to provide multirealm authentication and security. Use of DCE DFS/AFS will also be investigated, to make data and programs available to the distributed systems to which the scheduler has access.

1 Principal Investigators

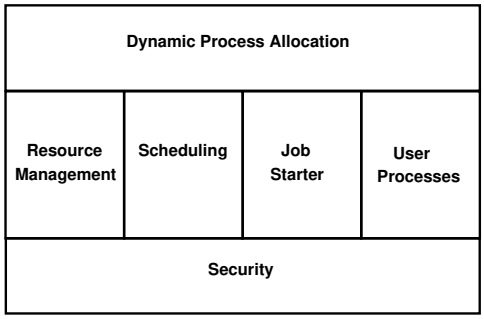
David Lifka Mathematics and Computer Science Division Argonne National Laboratory 9700 S. Cass Ave. Argonne IL 60439 Email: lifka@mcs.anl.gov Phone: 708-252-6175 Fax: 708-252-5986	Doug Engert Electronics and Computing Technologies Division Argonne National Laboratory 9700 S. Cass Ave. Argonne IL 60439 Email: engert@anl.gov Phone: 708-252-5444 Fax: 708-252-9689

2 Proposed Research

Many MPP machines are in operation at various DOE, commercial, and university sites. Existing or future codes could benefit from access to these massively parallel processors. Currently, however, network security and administrative difficulties restrict scheduling jobs across these distributed machines.

To overcome this restriction, we propose to enhance the Argonne Scheduler to securely schedule parallel and nonparallel jobs across distributed systems connected by

a wide area network. The Argonne Scheduler was developed at Argonne and has been running on Argonne's 128-node IBM SP system in production mode for nearly a year. The scheduler code is written in Perl and has a modular design so that various pieces can be modified without rewriting the entire code. Most significant, the Argonne Scheduler can make use of any message-passing system, security system, or job-starting systems that are available. The following diagram shows the various pieces of a complete scheduling system.



While some other distributed queuing systems address all of these pieces, the Argonne Scheduler addresses only scheduling. It is designed so that it can use other programs or packages to handle the other tasks. Such flexibility allows the Argonne Scheduler to keep pace with the most recent advances in these other areas. This is extremely important especially in the areas of security and current message-passing systems. To incorporate Kerberos or any other security system will not require a rewrite of the code. Because the Argonne Scheduler is not tied to any particular message-passing library, users can run any type of message-passing code including p4, PVM, and MPI, the new message-passing standard. This flexibility will greatly facilitate teraflop or even petaflop applications because developers will be able to produce their code in the message-passing system they are comfortable with, instead of having to invest the time in learning another message-passing system.

We will investigate the various ways the Argonne Scheduler must be modified in order to run on distributed platforms. In particular, a working prototype will be developed and demonstrated at the Supercomputing 1995 conference in San Diego, California. This prototype will use Kerberos for multirealm authentication and security. The area of security the Argonne Scheduler will have to address is in the authentication process of distributed systems on a wide area network. When a user is granted access to a distributed set of nodes, the Scheduler will have to make sure that a particular user is securely authenticated on the nodes on the distributed systems. The current version of the Argonne Scheduler uses NIS for user authentication. While this approach is suitable for a single machine, it will not work for distributed systems. We will design a system in which one central scheduler has access to resources controlled by remote schedulers and interacts with them in such a way that user's jobs at various sites have priority on their local machines but still will run other jobs requiring resources from several sites. We will incorporate the use of Kerberos to simplify the authentication process on the distributed systems. It will be implemented so that a user only has

to authenticate once and the scheduler will handle the authentication on any of the distributed systems a job may require. Once the nodes have been allocated, users requiring security for messages in their program's interprocess communication will be able to use any message-passing system that provides sufficient security.

As part of the development of the prototype system, we will use DCE DFS/AFS for secure access to data files for processors in the various distributed systems that are scheduled. This will involve setting up a special AFS cell that each of the participating distributed systems has access to. Simple Job setup will be facilitated because all of the distributed nodes will access their data files from the same file system path.

The development of the Scheduler will provide the capability to run many types of jobs that can benefit from the increased parallelism and access to different computer resources. Examples of such jobs are the following:

- A massively parallel computation run on 1024 or more nodes.
- A real-time simulation run across many compute nodes and rendered in one or more CAVEs.
- A meeting or simulation held in two or more distributed CAVES.
- A task-farm parallel job that needs many runs of the same program with different input data sets.

We will be able to leverage the use of the Argonne 128-node SP system and the CAVE virtual reality theater for the development of the distributed scheduler. We will develop a secure system that allows users of the CAVE to schedule time on the IBM SP to do real-time visual analysis of their SP computations. Once this system is working, we will extend it to distributed CAVE systems such as the one at University of Illinois at Chicago. As these systems are used, we will study the usability issues involved both from an administrative point of view and from a user perspective. The test beds will also provide data on the usability of current wide area networks for these very demanding applications. Our prototype system, which will be ready for Supercomputing 1995, will incorporate any fixes or useful utilities that are developed on these test beds.

3 Sharing Results

We are currently distributing the Argonne Scheduler to other sites that have requested it for use on their IBM SP systems. The University of Maryland and Cornell University have already begun to install it on their machines. University of New Mexico and the Maui Supercomputer Center have recently requested the code as well. The advantage of having more and more sites using the current Argonne Scheduler is that

they will be comfortable with its administration and user interface, making participation in a distributed system much easier. As part of Argonne CRADA agreement with IBM we are helping IBM add the necessary scheduler features to their scheduling product LoadLeveler. Eventually we may be able to tie other IBM customers into this distributed system who are using LoadLeveler.

4 Budget

We will be able to leverage the use of the Argonne SP, CAVE, and LabSpace facilities for this work. Funding will be used strictly for effort.

Effort	Cost ($)
Staff(0.9 FTE)	$117,900
Laboratory Overhead (21%)	$32,100
Total	$150,000