Operating Systems for Supercomputers

Birds of a Feather Session

Supercomputing ’89, Reno, NV.

Daniel J. Kopetzky[†]
John Riganati[†]
Dieter Fuss[‡]

[†] Supercomputing Research Center
16100 Science Drive
Bowie, MD. 20715

[‡] Lawrence Livermore National Laboratory
P. O. Box 808, L-669
Livermore, CA 94551

ABSTRACT

This paper summarizes the discussions held in a birds-of-a-feather session on the topic of future operating system support for supercomputers. The group collectively worked to identify current problem areas and identify potential future ones.

1. Background

Advanced systems software development for supercomputers in the 90’s may be viewed as falling primarily into two areas — distributed computing and parallel processing. The driving forces for distributed computing come from a desire to increase user productivity (for example, with a friendly workstation environment), maximize efficient use of resources, and enhance resource sharing; the driving forces for parallel processing come from a desire to increase performance, especially for studying physical phenomena and other large-scale calculations or to decrease cost at a constant performance.

Requirements for advanced systems software development for distributed computing and parallel processing may be structured into at least four areas: the operating system, the application run-time environment, tools and utilities, and command language interface. A "birds-of-a-feather" session held and Supercomputing ‘89 in Reno NV on 16 November, 1989 concentrated on only the operating system.

2. Topics Presented

A set of four topic areas were presented. End users require *support* for program creation, debugging, and tuning. *Communication Management* has become an important topic in a heterogeneous computing environment populated with machines that range from supercomputers to workstations. *Processor Management* encompasses the problems of choosing from multiple computers that could run a task. *Storage Management* covers how main memory is used, file system capabilities, and data archiving.

The group was asked to view them as a starting point to which they could amplify or add their own concerns.

3. "...illities" That We Want

Desired Capabilities
Ease of use
Performance Manageability
Expandability Flexibility Interoperability Maintainability
Portability Standardization
Reliability Recoverability
Security

The list to the left indicates some of the capabilities that are desired in a computing system. Making systems easy to use is a quest for reducing the people cost associated with running, and programming at a supercomputing center.

Jobs are assigned to supercomputers because of the machine's performance. Predicting and measuring the performance of tasks is necessary. System managers should be able to create different levels of service.

We must be able to plan the growth of a center. That may take the form of expanding an existing machine or adding new ones. A variety of machines must be supported.

Programs may have useful lifetimes that exceed that of a machine. Mechanisms and guidelines to increase portability are needed.

Long running computation will encounter machine failures. System support for minimizing lost work is needed. some tasks may require a dynamic allocation of computing resources from a distributed computing environment.

A security policy must be provided to insulate users from each other. These features need to be maintained while permitting global access to a computing center. Furthermore, some tasks may require a dynamic allocation of computing resources from a distributed computing environment.

4. User Support

Debugging Multi process Network Heterogeneous
Performance Prediction Measurement Production
Seamless Environment

Multiple process tasks adds a new dimension for bugs to inhabit. With the state of the computation spread across several processors cleanly stopping the task and repeatability are difficult to achieve.

Placing the component computations on processors separated by a network adds another layer of complexity and diminishes the amount of direct hardware control over a program.

Debugging in a heterogeneous system adds a further complication of trying to map different hardware capabilities to a common view.

The performance of a program should be planned from its initial design. Programmers need enough information to be able to predict the cost of the system services that they use.

The actual performance of a code needs to be measured and those results correlated with the predicted performance.

To insure that programs continue to meet their expected performance profile measurements must continue throughout production code runs.

5. Communication Management

Virtual Resources
    Naming
    Disk

Intertask
    On machine
    Cross machine
    To workstations

Gigabit links
    Protocols
    Global networks

Computing centers will have multiple supercomputers. High speed networks are needed to transfer information between those machines and from the machines to the front end and workstations of the users.

The peripheral devices can be shared across several machines. The NFS disk sharing protocol is one example of information sharing used in the mini-computer world. Special attention needs to be given to problems caused by high interconnecting high performance systems. Naming conventions are needed to identify these distributed resources without requiring each program to maintain explicit knowledge of the location of the resources.

Standards for building groups of communicating parallel processes will allow a single task to be spread across several processors. Different solutions to this problem may be needed for the special cases of groups of processors running in a shared memory machine and for communication between machines made by different manufacturers.

Network protocols designed for high speed networks may be needed to augment standard protocols that were developed primaryly for long haul, high error rate, communication. Wide area regional network access to supercomputer centers are required by some user communities.

6. Processor Management

Authentication
    Accounting
    Access Control

Usage Control
    Priority
    Researcher's iface
    Administrator's iface

Load Balancing
    CPU Selection
    Distributed syscalls
    Process migration
    Checkpoint/restart

Scheduling
    Long running jobs
    Batch
    Interactive

The high cost of supercomputing usually means that machines must be shared across cost accounting boundaries. Access control mechanisms are used to constrain users into using a subset of a center's resources. These mechanisms must be implemented such that they do not interfere with the goals of distributed computing.

Researchers need to be able to control their computing costs. Priority schemes allow a trade off between the cost and completion time. Whereas researchers probably want to conserve funds administrators want to minimize idle (unpaid for) resources. This may lead to significant differences between resource control mechanisms for those two groups.

In a heterogeneous computing environment it may be the case that some tasks can execute (with differing performance metrics) on any of a variety of machines. It was mentioned that a system which permits distribution of system calls can facilitate load sharing. Process migration allows tasks to adapt to a changing pattern of resource availability.

The scheduling of work needs to be considered with a view of the entire computing center. The presence of high-speed long haul networks may even permit load shifting from one center to another. A balance is needed between machine efficient batch jobs and interactive computation. Finally tasks that take a large amount of wall time, exceeding weeks of supercomputer time, will require special scheduling consideration.

7. Memory Management

Main Memory Coordination Swapping & Paging
I/O Scaling Cache control Intelligent disk
File System RAID disk Data compression Beyond gigabytes
Archiving

Supercomputers have enormous main memory systems. Supercomputer applications seem to want all of that memory for themselves. Algorithms for allocation of main memory and system calls to address memory as a negotiable resource are needed to grow from the minicomputer environment where a task can be moved to secondary storage in a few seconds.

Over time processor performance improvements have outstriped those in the storage area. Complex I/O systems that grow with increased system capabilities may be needed to preserve a balance between computation and I/O.

Newer technology will force change in the ways that storage is used. Data compression techniques have been used to increase the recording capacity of disk drives. Perhaps these same techniques can extend network and channel bandwidth by moving data in a compressed form. Individual data sets will grow as computing capacity increases

The retention of data for years will need standards that outlive the machines that originally generated the data.

8. Resource Control and Administration

Brokering CPU cycles Processor type Disk storage Disk bandwidth
System Updates Internal OS interfaces Customization guides Site configuration
Documentation Vendor User generated

The identification of available excess resources and their assignment to interested parties can be accomplished through the actions of a software system called a resource broker. This lessens the need for individual processors to get involved in strategic resource assignment.

Because supercomputing installations generally differ portions of the system need to be tailored to each site's requirements. Some times this includes inserting extensions into the base systems's service repertoire. Guidelines for this type of modification are needed to ease the burden of re-integration when the vendor updates system software.

The presence of high-quality online documentation helps educate the user community and should yield better utilization of existing software and hardware resources. Attention needs to be given to providing the same quality documentation delivery system to be used by local authors.