

Operating System of the VX/VPP300/ VPP700 Series of Vector-Parallel Supercomputer Systems

● Yuji Koeda

(Manuscript received April 29, 1997)

The computers used in science and technology are generally either low-end, high cost-performance machines owned by individual research and development sections, or high-end ultra high-speed machines.

This paper describes the features of the UXP/V operating system of the VX/VPP300/VPP700 series of vector parallel supercomputers, which were developed to flexibly cover the requirements for science and technology computing. It also looks at the following important topics regarding vector parallel supercomputers :

- The method of allocating resources such as the CPU and memory
- The scheduling technique for batch processing
- The technology used to achieve high-speed I/O processing and network processing

This paper also describes a function for easy installation and administration and a strengthened operational management function for the computer center.

1. Introduction

The cost effectiveness of science and technology calculation environments has rapidly improved due to increases in the processing speeds of high-end EWSs and the falling prices of supercomputers and minicomputers. These changes have encouraged more research departments and researchers to use science and technology calculation machines. The VX, VPP300, and VPP700 series vector-parallel supercomputers were developed to meet the demands for very high processing speeds to perform complicated, high-level calculations.

This paper looks at problems connected with the operating system of vector-parallel supercomputers, and explains the development of UXP/V, which is the operating system of the VX, VPP300, and VPP700 series.

2. Development history

The history of UXP/V began with UTS/M, which is a UNIX^{Note1)} system that operates on a mainframe. UTS/M was based on the UTS, which was developed between 1982 and 1985 by Amdahl Corporation in the USA as a UNIX system on Amdahl's mainframe. At first, UTS/M was based on SVR2^{Note2)}, and was supplied as a business-oriented UNIX system for mainframes; then a UNIX system for supercomputers was requested. In 1990, VPO^{Note3)} was supplied as an additional function of vector processing so that the UNIX system could be used on a supercomputer.

In 1989, the System V UNIX and the BSD UNIX were unified into SVR4^{Note4)}. Taking this opportunity, we developed UXP/M by introducing the base of UTS/M into SVR4. We released UXP/M in 1991.

Note1) UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Note2) SVR2 is an abbreviation of UNIX System V Release 2.

Note3) VPO is an abbreviation of Vector Processor Option.

Note4) SVR4 is an abbreviation of UNIX System V Release 4.

Then, calculation capabilities about 20 times better than that of a conventional single processor or TCMP-type^{Note5)} vector computer was required in aircraft and weather service fields. In response, in 1993, we developed and supplied the VPP500 series of distributed-memory-type vector-parallel supercomputers. This series has the characteristics of both a conventional vector computer and a new parallel processing machine. The operating system for this series consists of two parts: UXP/M, which supports the standard UNIX function in a front-end processor, and a parallel operating system in a back-end processor for batch processing only.

Then, we developed UXP/V for the VX, VPP300, and VPP700 series to provide flexible support of various applications in research and development departments and superior cost effectiveness. UXP/V is a standalone UNIX operating system with extended parallel and vector functions. It can be stored in each processor. We began sales of UXP/V in 1995.

3. Aim of development

The VX, VPP300, and VPP700 series are distributed-memory-type vector-parallel supercomputers appropriate for high-speed processing of large amounts of data in science and technology calculations. Because these systems can be extended scalably according to the application, they can be used under various conditions. That is, these systems can be used as departmental systems exclusively used in research and development departments or as center systems shared by various research and development departments. The operating system for the VX, VPP300, and VPP700 series must have a higher performance, superior operation management functions, and wider application fields; also it must

be easier to operate, install, and manage.

1) Open architecture

This operating system is based on SVR4, which is the standard UNIX operating system. A standard GUI and language environment, such as X11^{Note6)}, Motif^{Note7)}, Fortran90, ANSI C, and Message Passing Library are used. Moreover, the function for vector-parallel supercomputer processing is extended by maintaining an open architecture. This improves the connectivity and operation of the system as a calculation server under a distributed system environment.

2) Higher performance

Files can be accessed at high speed using the disk array unit, large-capacity high-speed file system, and software striping function.

This system can be connected to a LAN conforming to ANSI standard HIPPI (800 megabits per second)^{Note8)} or ATM-LAN (155.2 megabits per second)^{Note9)}. Such a connection increases the network transmission speed, which is important in a distributed system environment. Moreover, the performance of application programs is enhanced by using the compiler, which has superior optimization techniques, and by using the tuning tools for vectorization and parallel processing creation.

3) Easier installation and management

The operating system and basic software products are provided on disk. The user can start using this product by performing the minimum environment setup at installation. This system can be started and stopped using the power ON/OFF button, which enables the system to be easily used as a departmental system. Online manuals are enhanced so that required information can be effectively retrieved from a workstation.

4) Superior center operation management function

Note5) TCMP is an abbreviation of Tightly Coupled Multi-Processing.

Note6) X11 is an abbreviation of X Window System V11. X Window System is a trademark of X Consortium, Inc.

Note7) Motif is trademark of Open Software Foundation, Inc.

Note8) HIPPI is an abbreviation of High Performance Parallel Interface.

Note9) ATM is an abbreviation of Asynchronous Transfer Mode.

Interference between batch processing and interactive processing is prevented. The job^{Note10)} management function, which is the nucleus of center operation, is enhanced for flexible handling of various types of operating conditions. The center operation management functions such as the security function (AUDIT and ACL) are enhanced. Also, the automatic operation function is enhanced (system-start by power-on and system-stop by power-off can be automatically performed at a specific time using the calendar function of the hardware).

5) Continuity of resources

For inheritance of user resources at system changeover, binary compatibility with the VPP500 vector-parallel supercomputers and source code compatibility with the VP and VPX series vector supercomputers are assured.

4. Characteristics of the operating system

4.1 Standalone parallel operating system

The VX, VPP300, and VPP700 series are supercomputers having two or more processing elements (PEs) connected through a high-speed crossbar network. Figure 1 shows the system configuration. Each PE consists of a scalar unit, vector unit, main memory, data transfer unit, and other units. Standard UNIX with the extended parallel processing function and extended vector processing function operates on all PEs. Therefore, every PE can perform vector processing, parallel processing, and interactive processing. A PE to which input/output devices such as disk drives, tape units, and network units can be connected is also called an IOPE^{Note11)}. If there are a large num-

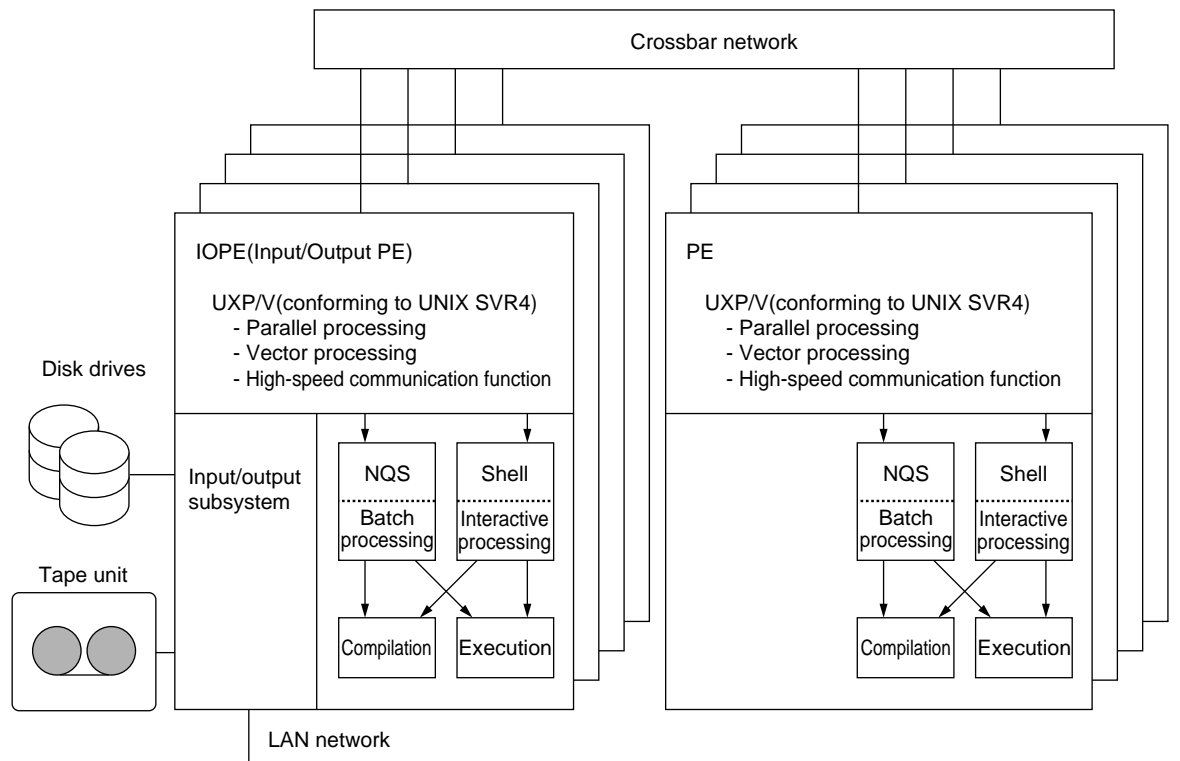


Fig. 1— Standalone parallel operating system.

Note10) A job is a series of tasks to be executed for a user.

Note11) IOPE is an abbreviation of Input/Output PE.

ber of PEs, a single IOPE may not be able to handle all of their I/O requests, and the many accesses to system disks via the IOPE will lengthen the system start-up time. To solve these problems, two or more IOPEs are prepared to distribute the disk and network load. Also, system disks are connected to two or more IOPEs so that the PEs using the system disks are grouped. The start-up time for a large-scale system can be shortened by starting the system in parallel in group units.

4.2 Interactive processing and batch processing

Interactive processing and batch processing can be executed on each PE. Each PE can be handled as a host on the network, and can accept a remote log-in from an arbitrary host. In batch processing, CPU distribution can be freely controlled without affecting interactive processing. UXP/V can select the appropriate PEs (PEs for interactive processing only or PEs for batch processing only can be selected). In a PE for batch processing only, a log-in from an external network is suppressed. In a PE for interactive processing only, execution of a batch job is suppressed. Memory and CPU resources are carefully allocated so that batch processing and interactive processing do not interfere with each other, even if batch and interactive processing are executed on the same PE. Memory is divided according to the purpose of use so that interactive processing does not conflict with batch processing. The distribution of CPU use between interactive processing and batch processing is set to assure a quick response in interactive processing.

4.3 Cooperation with workstations

Vector-parallel supercomputers should be used for calculations, and other operations suit-

able for workstations should be executed on workstations. For example, the development environment and GUI software products should be run on workstations as much as possible. To facilitate this, software products that must cooperate with workstations must be prepared. Some examples of these software products are the VPP Workbench (for supplying a program development environment with a GUI equivalent to that for workstations), NQS^{Note12)} (for submitting jobs), and OLIAS^{Note13)} browser (viewer for online manuals).

4.4 Job management and job scheduling

1) Job management

UXP/V jobs are managed by the partition manager (PM). The PM allocates and releases resources to jobs, and controls the priority. A job is entered as an NQS batch request. NQS requests the PM to allocate resources. A batch request to which the PM allocates resources is executed as a job. When the job execution terminates, the resources are released by the PM and the batch request terminates. Then, NQS reports the execution results to the person who entered the batch request.

2) Job scheduling mechanism

Jobs must be scheduled to use the parallel processing system effectively. The job scheduling mechanism of UXP/V is explained below.

i) PE allocation according to job class

The maximum processor count, execution mode (SIMPLEX or SHARED mode [explained later]), PE to be allocated, and other information are specified in a job class. The job class to be used is specified in the NQS queue. When the user selects an NQS queue and submits a job, the PM allocates PEs according to the job class setup information. **Fig-**

Note12) NQS is an abbreviation of Network Queuing System. NQS was ported from a program that was developed in response to a request from NASA.

Note13) OLIAS is an abbreviation of Online Information Access System.

Figure 2 shows an example of PE assignment according to the job class. In this example, NQS batch queue A is assigned to job class 0. The PM allocates 2-parallel job A1 to PE1 and PE2 according to the definition of job class 0. NQS batch queue D is used to submit a 4-parallel job. Using job class 3, 4-parallel job D1 is allocated to PE3 to PE6. Thus, PE allocation to submitted jobs can be freely controlled according to the job class.

ii) CPU distribution control

The CPU distribution ratio can be set for the scheduling class for interactive processing and the scheduling class for batch processing. Moreover, the CPU distribution value can be specified in batched job units. Units of CPU time called tickets are allocated to each job and then taken away as each job spends them on CPU time. A job that has spent all of its tickets loses execution priority and is not executed while there are jobs that still have tickets. The scheduler issues to each

job at fixed intervals an additional ticket whose length is proportional to the CPU distribution value.

Figure 3 shows an example of the CPU distribution control. In this example, 20% of the CPUs are assigned to interactive processing and 80% are assigned to batch processing. For the job to be executed in batch processing, the CPU time is distributed according to the CPU distribution value defined in batch queues A and B. At execution, jobs a1, b1, and b2 are allocated 20%, 30%, and 30% of the CPU time, respectively. Even if interactive processing and batch processing are both executed or there are many jobs to be executed, the CPU time to be allocated to the jobs can be freely controlled by this control function.

iii) Memory priority

In resource allocation, jobs with a higher memory priority are the first jobs to acquire resources. If there are two or more jobs with the same memory priority, the job that first

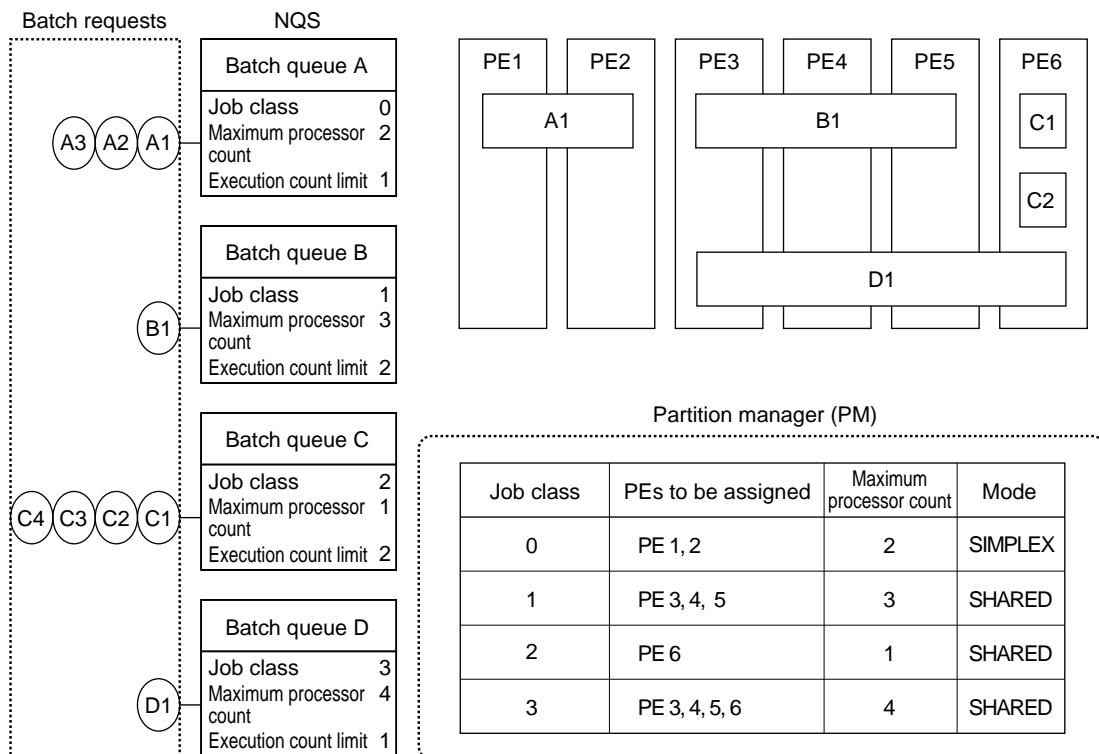


Fig. 2— PE allocation according to job class.

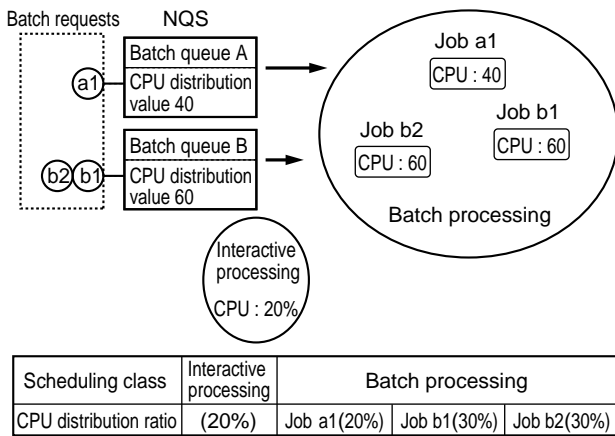


Fig. 3— CPU distribution control.

requested resource allocation is given priority. After jobs have been submitted from NQS, their memory priority can be changed as required. Therefore, a specific job can be executed earlier than the other jobs by changing its memory priority.

iv) SIMPLEX mode and SHARED mode

Jobs are executed in SIMPLEX mode or SHARED mode. SIMPLEX mode is used to execute jobs by exclusively using the PE. The SHARED mode is used to execute jobs with a high throughput. While a job is being executed in SIMPLEX mode, other jobs are not executed on the PE but wait until the PE becomes idle. If a job is being executed, another job in SIMPLEX mode waits until the job being executed terminates. On the other hand, jobs in SHARED mode can share the same PE. If there is an idle part of a PE, the submitted jobs are sequentially allocated to the PE and executed.

v) Fixed priority

The absolute priority for job execution can be specified (this priority is called the fixed priority). To execute an urgent job on a PE that is currently executing another job, the PE's CPU can be allocated to the urgent job by increasing the fixed priority.

vi) Harmonized scheduling

Some parallel jobs can be executed as concur-

rently as possible on PEs by increasing the priority of these jobs only for a fixed time at specific intervals. Therefore, several parallel jobs can operate as if they are exclusively and periodically using the system (this control is called harmonized scheduling). This minimizes the overhead of synchronization between PEs. Figure 4 shows an example of CPU assignment using the harmonized scheduling function. In this example, the CPU assignment period is set to 100 milliseconds. The CPU of every PE is assigned to parallel job A for 40 milliseconds and then to parallel job B for 40 milliseconds. The CPU is assigned to other processes for the remaining 20 milliseconds and during the input/output operations for the two jobs.

3) Job swap function

The job swap function temporarily moves a currently executed job to a job swap area or returns the job to the original area. Using this function, an urgent job can be executed and operation can be switched smoothly.

i) Execution of urgent job by NQS queue selection

To execute an urgent job, an NQS batch queue with a high priority must be prepared. In the job execution, an urgent job can be executed earlier than other jobs by submitting the urgent job to the queue. While another job is being executed on a PE, a job swap automatically occurs because of job resource contention. Figure 5 shows an example in which an urgent job is executed by selecting an NQS queue. In this example, job 1 in the batch queue with memory priority 10 is being executed on PE1 to PE4. When an urgent job is submitted to the batch queue with memory priority 20, job 1, which has a lower memory priority, is swapped out to the swap device. Then, the urgent job is allocated to PE1 to PE4 and executed. When the execution of the urgent job terminates, job 1 in the swap device is swapped in PE1 to PE4 and its execution is resumed.

ii) Forced job swap-out

A job management command for forced swap-out of a currently executed job is provided. Using this command, the operation status can be changed according to the time. Because the resources to be used after operation switching can be reserved, the next job

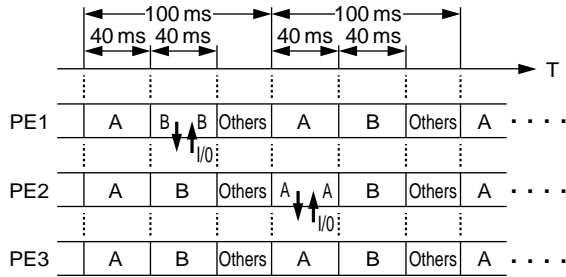


Fig. 4— Harmonized scheduler.

execution can be quickly started.

4) Job freeze function

If the system must be stopped for periodic maintenance when a job is still running, the job must be stopped temporarily and the frozen data stored in files. After the system is restarted, the job can be thawed using the frozen data and the job can be restarted.

4.5 High-speed input/output processing

To process jobs at high speed, the input/output processing speed must be increased in addition to the speed of the vector operation and parallel processing features. The following explains the high-speed input/output function developed for UXP/V.

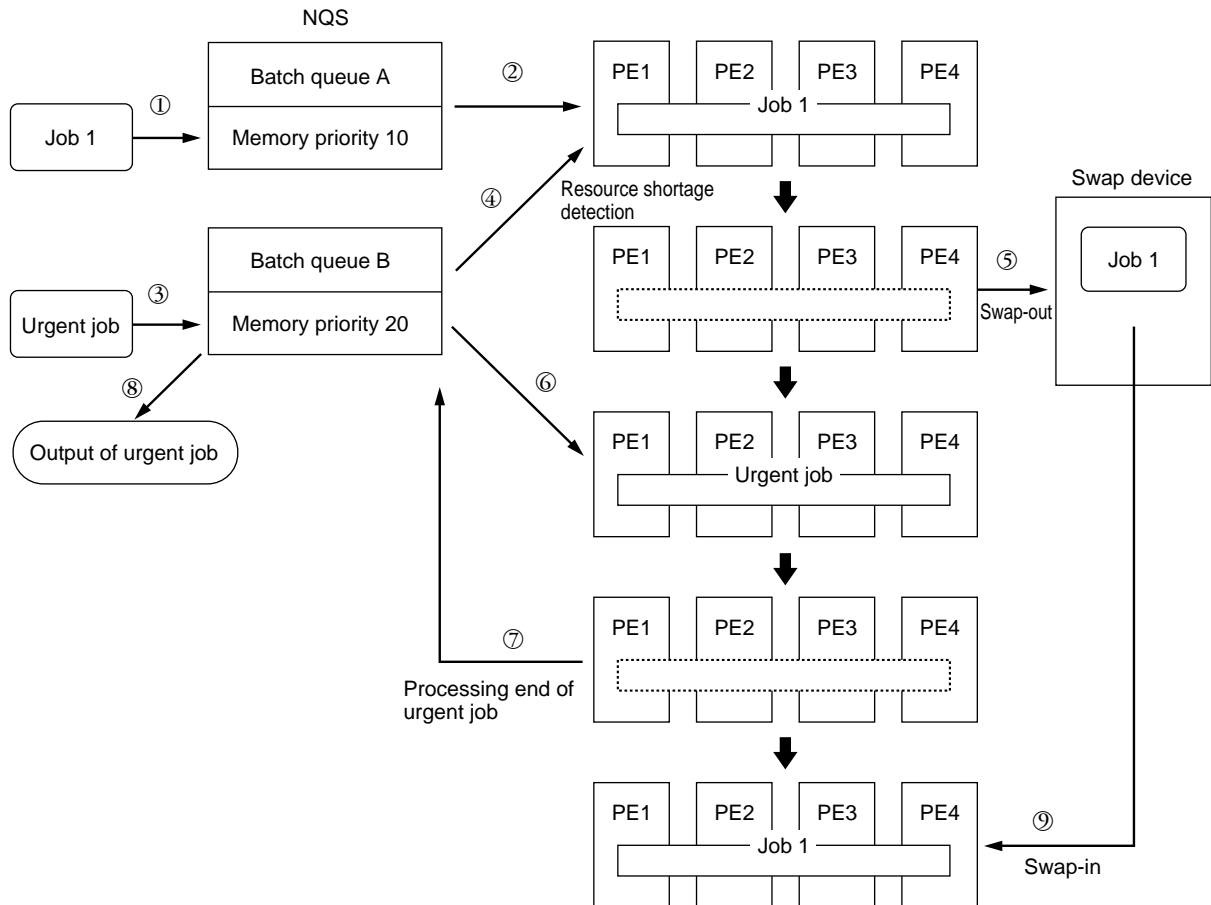


Fig. 5— Execution of urgent job by NQS queue selection.

1) Memory resident file system (mrfs)

The mrfs^{Note14)} is a file system constructed in the real memory of a PE. Because an access to a file on mrfs uses the vector function, data is transferred between memory areas at high speed. There are two types of mrfs: the shared mrfs (which can be shared between two or more jobs) and the job mrfs (which can be used only during job execution). A shared mrfs can be handled like an ordinary file system. Because the shared mrfs is a volatile file system, the files in mrfs are deleted when the file system is unmounted or the PE is shut down. The job mrfs is assigned to an area when a job starts. When the job terminates, the job mrfs is automatically released. **Figure 6** shows an example use of mrfs.

The compiler, linker, library, and other resources to be commonly used between two or more jobs are arranged in the shared mrfs. These resources are used by the jobs that execute a series of operations such as compilation, linkage, and execution. The intermediate creation results (e.g., objects, load modules, and temporary files) of the job use the job mrfs. Thus, the jobs are executed at high speed without input/output accesses to the disks.

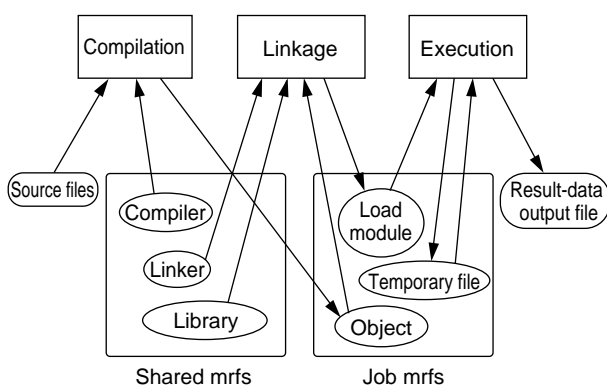


Fig. 6— Example of mrfs use.

2) Very Fast and Large File System (VFL-FS)

The VFL-FS^{Note15)} is a file system for high-speed input/output of large amounts of data. In this file system, data blocks are allocated to contiguous areas on a medium. Compared with the standard UNIX file system (in which data blocks are allocated to discrete areas on a medium), the disk-head seek time and rotational delay are reduced. Moreover, because a large amount of data from a job is transferred directly to a medium and not through the system buffer, the overhead of copying data to the system buffer is avoided. Thus, VFL-FS has a high input/output performance for large amounts of data.

3) Disk array units

RAID disk array units can be connected. A RAID disk array unit contains two or more disk drives that perform input/output operations in parallel to increase the disk access speed and improve reliability.

4) LVCF (logical volume manager)

LVCF^{Note16)} is used to construct logical volumes such as large-capacity volumes and stripe volumes by combining various real devices. A large-capacity volume is a continuous logical volume created by connecting two or more physically different real devices. By using a large-capacity volume, a large-capacity file system can be created and devices can be used effectively. A stripe volume is created by arranging two or more physically different real devices. These parallel devices have parallel input/output operations. The file access speed for these stripe volumes is increased by transferring data to these real devices simultaneously. When this stripe volume function is combined with the above disk array unit, parallel input/output operations are executed by both hardware and software, which increases the file access speed.

Note14) mrfs stands for Memory Resident File System.

Note15) VFL-FS stands for Very Fast and Large File System.

Note16) LVCF is an abbreviation of Logical Volume Control Facility, which is a logical volume manager developed by VERITAS Software Corporation of the United States.

4.6 High-speed network

When a vector-parallel supercomputer is connected as a calculation server to a network in a distributed system environment, the network processing speed must be increased. Increasing the network processing speed increases the processing speed of the total system, including the vector-parallel supercomputer. The TCP/IP communication protocol makes it possible to connect the communication mediums listed below and extend the system functions. This protocol is widely used in research and development fields.

1) HIPPI-LAN

The TCP/IP communication function conforms to RFC1374 and is created on an HIPPI network conforming to ANSI X3.218, X3.210, and X3.222. A file transfer function and NFS^{Note17)} that use a high-speed (800 megabits per second) communication path are supported.

2) ATM-LAN

The point-to-point TCP/IP communication function can be used by connecting an ATM switch (exchange) and using the RFC1483 capsule control method. High-speed (155.2 megabits per second) communication with each host can be performed using the star-type switch connection.

3) Inter-PE TCP/IP communication function

The TCP/IP communication function is supported in the crossbar network. The TCP window size, which determines the maximum amount of data that can be transferred continuously, and the transfer data length are extended. Interprocess communication programs that use standard interfaces such as the socket/TLI/RPC can be used on a high-speed (570 megabytes per second) communication path of the hardware.

4) Inter-PE NFS

File access between PEs is executed using NFS. The processing speed of the standard NFS is much lower than that of the crossbar network,

which is 570 megabytes per second. To solve this problem, the NFS protocol was modified so that the slower part of the NFS protocol was removed and the inter-PE communication function of the crossbar network was used directly. Direct data transfer between PE memory increases the remote file access speed.

4.7 High reliability

Maintenance is important in a large-scale system having many PEs. Even if some PEs become faulty, UXP/V can continue system operation by isolating the faulty PEs. Moreover, a PE can be maintained during system operation because its power can be turned off separately.

We are also attempting to enhance AUDIT, ACL, and other security functions that are important to central operation.

5. Conclusion

This paper looked at the resource assignment, scheduling, high-speed input/output, system installation, and operation functions of vector-parallel supercomputers. Also, this paper explained the operating system function support in UXP/V.

In the future, we will examine how to increase the processing speed (the most important factor of supercomputers), and study various operating environments in order to find ways to enhance the operation management function and simplify operation. Also, we will continue to develop better operating systems for supercomputers.



Yuji Koeda received the Master degree in Electrical Engineering from the University of Yamagata, Yamagata, Japan in 1985.

He joined Fujitsu Limited, Kawasaki in 1985, where he is currently engaged in development of network communication systems for UNIX mainframe computers and supercomputers.

E-mail : koeda@open.nm.fujitsu.co.jp

Note17) NFS is an abbreviation of Network File System, which is a trademark of Sun Microsystems, Inc. of the United States.