

Mass Data Processing System and NQS II

As many large-scale distributed parallel programs are executed on the Earth Simulator (ES), I/O processing for user files and work files has become the bottleneck of numerical simulation. So, in 2003, we have introduced the Network Queuing System II (NQS II)* as Job Manager and Mass Data Processing System (MDPS) to resolve this problem or to improve the system utilization and maintainability.

Mass Data Processing System

Mass Data Processing System (MDPS) was installed as a new data storage system, which renews the archive system (Figure 1). It consists of four File Service Processors (FSPs), 240TB magnetic disk system, and a 1.5PB cartridge tape library (CTL). MDPS is adopted aiming to improve manageability for a data transmission performance and accessibility.

Major improvements are as follows.

- The theoretical transfer speed of data between the work disk of ES and the storage becomes about 6 times faster. This improvement has been realized by expanding the capability of transfer cable and replacing a tape archive with the disk system of MDPS.
- Data I/O procedure becomes easy.
- MDPS enables users to access to the results computed by ES from remote locations, because it can transfer the data to a dedicated server out of Internal LAN.

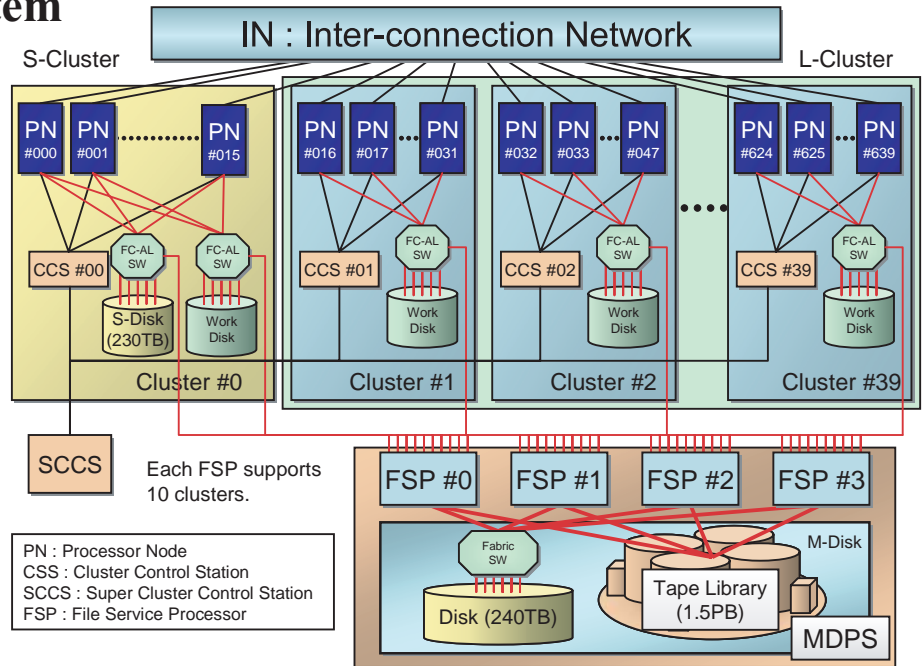


Figure 1: Configuration of the Earth Simulator. MDPS has 240TB HD and 1.5PB CTL.

Network Queuing System II*

In order to utilize MDPS efficiently, the Network Queuing System II (NQS II) was introduced as a job manager. And we have developed a scheduler for the job manager with following strategies.

- The nodes allocated to a job are used exclusively for that job.
- The job is scheduled by using elapse time instead of CPU time.

These strategies enable to estimate the job termination time and make it easy to allocate nodes for the next job in advance (Figure 2). Although ES has adopted a single batch queue environment, they also enable large-scale batch jobs to be executed in a short waiting time.

In ES, Processor Node (PN) are prohibited from access to user disk. Therefore, user files are copied from user disk to work disk before the job execution. This process is called "Stage-IN." In job scheduling, it is important to hide this Staging time. To hide Staging time, Staging process of the next scheduled job is executed by FSP in the background of the request execution in PN. This process is executed without using resources of PN. After the job execution, results of the simulation are also copied from work disk to user disk. This is called "Stage-OUT."

Main steps of Job Scheduling are summarized as follows:

1. Node allocation
2. Stage-IN (copy files from user disk to work disk)
3. Job escalation (rescheduling for earlier start time if possible)
4. Job execution
5. Stage-OUT (copy files from work disk to user disk)

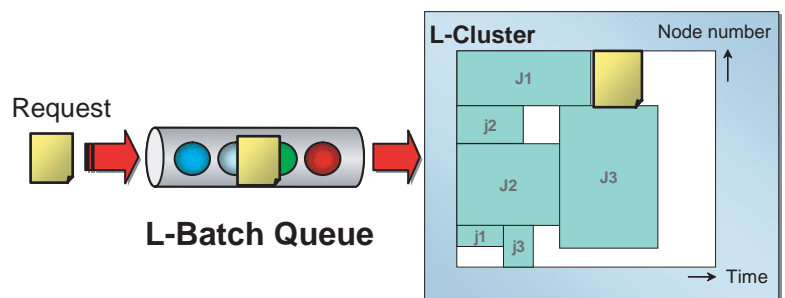


Figure 2: Job scheduling in L-Cluster. ES adopts elapse time based scheduling algorithm. The nodes allocated to a job are used exclusively for the job.

*NQS II is a registered trademark of NEC Corporation.

Programming Environment

Programming Model

Earth Simulator (ES) provides the three-level hierarchy of parallel processing: vector processing on an Arithmetic Processor (AP), parallel processing with shared memory in a Processor Node (PN), and parallel processing among PNs via the Interconnection Network (IN).

To bring out the highest performance of ES fully, you must develop well-parallelized programs that make the most use of such parallelism. The programming environment available on ES, as shown in Table 1, would help you develop good programs for ES.

Table 1: Programming Model of ES

	hybrid	flat
inter-PN	HPF / MPI	HPF / MPI
intra-PN	microtasking / OpenMP	
AP	automatic vectorization	

You can see in the table there are two different usages of the three-level parallelism.

The first is on the *hybrid* model, where you must distinguish the intra- and inter-PN parallelism and specify each of the two explicitly in your programs. The inter-PN parallelism can be specified with HPF or MPI, and the intra-PN with microtasking or OpenMP, in this model of parallelization.

The second is on the *flat* model, where ES as a whole is viewed as a flat system composed of 5120 (= 640x8) APs and you do not have to distinguish the intra- and inter-PN parallelism. Parallelization can be done with HPF or MPI, in this model of parallelization.

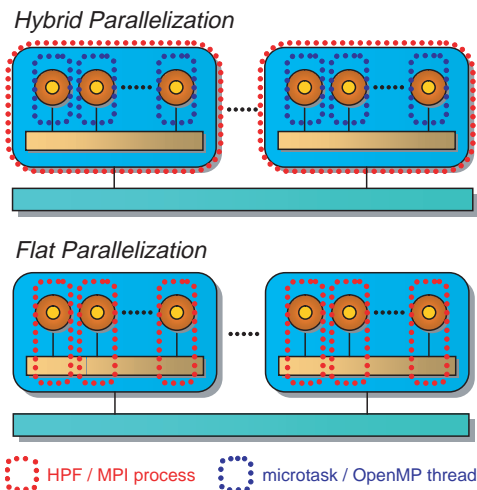


Figure 1: Two Models of Parallelization

Parallelization and Compilation

HPF/ES is an HPF compiler that allows you both easier and more efficient parallel programming on ES (to be described in the following section). MPI/ES, a message passing library based on the MPI-1 and MPI-2 standards, provides the capability of high-speed communication that fully exploits the features of IN and the shared memory in a PN. OpenMP is also available for the intra-PN parallelization.

Compilers for Fortran, C, and C++ (called FORTRAN90/ES, C/ES, and C++/ES, respectively) are available. All of them have an advanced capability of automatic vectorization and microtasking.

High Performance Fortran (HPF)

Features

HPF/ES supports features from the specifications of HPF2.0, the approved extensions, and HPF/JA, as well as some unique extensions for ES, as illustrated in Figure 2.

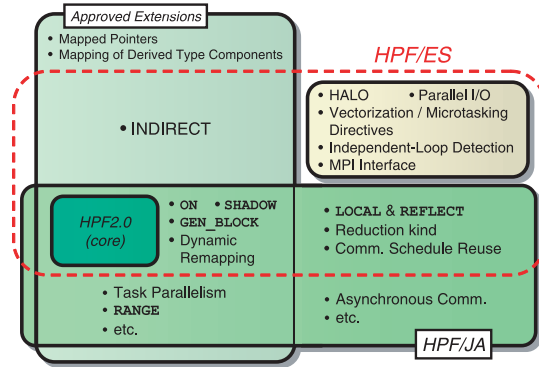


Figure 2: Supported Features of HPF/ES

The ES-specific extensions include:

- *HALO*, a feature for irregular problems such as finite element method;
- *parallel I/O*;
- *vectorization / microtasking directives*;
- *automatic detection of independent loops*;
- *interface to MPI subroutines*;
- *etc.*

Applications

We parallelized a plasma simulation code IMPACT-3D with HPF/ES and obtained the performance of 14.9 Tflops in the 512-nodes execution on ES. For this achievement we were given the Gordon Bell Award for language in SC2002.

HPF/ES is used for the simulations in various fields, for example, plasma, solid state physics, turbulence, etc.

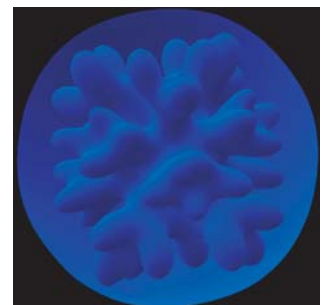


Figure 3: Result of IMPACT-3D: the pusher-fuel contact surface of stagnating targets in the laser fusion

HPF/ES for PC Cluster

HPF/ES is now available on the platform of PC Cluster. *HPF/ES for PC Cluster* provides you with the whole of the features of the original HPF/ES, except those dedicated to the ES-specific hardware (e.g. vectorization / microtasking directives). It can be distributed for free on a license agreement. Please contact us for more detail.