# MULTI-PROCESSOR SYSTEM, DISK CONTROLLER USING THE SAME AND NONDISRUPTIVE MAINTENANCE METHOD THEREOF

## BACKGROUND OF THE INVENTION

The present invention relates to a multi-processor system and a program version-up method therefor. More particularly, it relates to a disk controller comprising a multi-processor system including a plurality of processors and a control program version-up method therefor.

In the multi-processor system, processing of a job may be efficiently conducted by dividing the job into a plurality of processors for processing. As one form of application of the multi-processor system, a disk controller of a multi-processor configuration has been put into practice. Such a disk controller is disclosed in JP-A-2-62623. In the disk controller disclosed in JP-A-2-62623, a job such as reading or writing of data is distributedly executed by a processor which controls data transfer between a disk controller and a host computer, and a processor which controls data transfer between the disk controller and a disk drive. Specifically, for example, data read from one disk drive by one processor is transferred to the host computer by another processor. Thus, the data transfer process between the host computer and the disk controller and the data transfer process between the disk drive and the disk controller may be executed in parallel and

efficient data transfer mat be attained.

SUMMARY OF THE INVENTION

In the disk controller of the multi-processor configuration as described above, it is possible to stop the processing by some processors and continue the data transfer control by the other processors to maintain fault processors. However, in such a maintenance method, it is a premise that the processors which distributedly process a job execute programs of the same version. When the programs to be executed by the respective processors are to be versioned up and the maintenance is conducted in this manner, there occurs a period in which the programs of different versions co-exist and one job may be shared by the processors which execute the programs of different versions. In such a case, there may be no assurance of consistent processing of the job. As a result, when the versions of the programs executed by the processors are to be versioned up, the system has to be wholly stopped.

It is an object of the present invention to solve the problem encountered in the prior art system and enable the maintenance processing such as the version-up of the programs while continuing the processing by the system.

In order to achieve the above object, in the multi-processor system of the present invention, each processor belongs to one of a plurality of logical

clusters. The system includes a cluster non-restricted mode in which the distributed processing of one job among processors belonging to any clusters is permitted, and a cluster restricted mode in which the distributed processing of one job among processors belonging to different clusters is restricted.

In a maintenance method of the present invention, in the multi-processor system described above, the cluster mode is shifted from the cluster non-restricted mode to the cluster restricted mode and a cluster is selected from the plurality of clusters. The acceptance of a new job in the selected cluster is inhibited and the maintenance processing such as the version-up of the program of the whole processors in the selected cluster is conducted and the acceptance and processing of a new job among processors in the selected clusters is permitted again. The selection of the cluster and the maintenance processing are effected sequentially for each of the clusters, and after the maintenance processing has been completed for the processors of all clusters, the cluster mode is returned to the cluster non-restricted mode. During the maintenance of the processors in the selected cluster, the processors in the non-selected clusters are restricted to execute the processing of the job in the cluster restricted mode. In this manner, even if the processing of the job is continued by the processors which are not to be maintained during the maintenance of

the system such as the version-up of the program, the distributed processing of the job by the programs of different versions may be prevented and the maintenance process may be conducted without wholly stopping the system.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a configuration of a disk controller in accordance with the present invention,

Fig. 2 shows a format of cluster allocation information stored in a control memory,

Fig. 3 shows a format of execution cluster indication information stored in the control memory,

Fig. 4 shows a format of I/O job execution control information stored in the control memory,

Fig. 5 shows a format of cluster operation mode indication information stored in the control memory,

Fig. 6 shows a main flow chart of a version-up process of a disk control program,

Fig. 7 shows a detailed flow chart of the program version-up process for a cluster A,

Fig. 8 shows a detailed flow chart of the program version-up process for a cluster B,

Fig. 9 shows a configuration of a disk controller in accordance with another embodiment of the present invention,

Fig. 10 shows a format of distributedly

processable cluster combination information,

Fig. 11 shows another format of the distributedly processable cluster combination information, and

Fig. 12 shows a configuration of the disk controller in accordance with a further embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Fig. 1 shows a configuration of a disk controller in accordance with a first embodiment of the present invention. A disk controller 200 comprises a plurality of processors 211, 1212, 213, 214, 221, 222, 223 and 224 each having a local memory LM, service processors 215 and 225 having disk drives 216 and 226, respectively, each storing a control program, and a shared memory 230 having a cache memory 231 and a control memory 232 and commonly accessible by the processors 211-214 and 221-224 and the service processors 215 and 216. Of the processors belonging to the disk controller 200, the processors 211, 212, 221 and 222 are connected to a host computer 10 through a channel controller 30. The processors 213, 214, 223 and 224 are connected to disk drives 40 and 41. Hereinafter, in order to discriminate those processors, the former are refereed to as channel control processors and the latter are referred to as drive control processors. The local memory of each of the processors 211-214 and

221-224 stores a data transfer program for controlling the data transfer between the disk drives 40 and 41 and the channel controller 30 and a maintenance support program for supporting the system maintenance. The service processors 215 and 225 are connected to a maintenance panel 240 which is provided as an interface to an operator.

In the present embodiment, the processors 211-214 and 221-224 are of a multi-processor configuration. Those processors are divided into a cluster A 210 which comprises the processors 211-214 and a cluster B 220 which comprises the processors 221-224. The service processor 215 belongs to the cluster A 210 and the service processor 225 belongs to the cluster B 220.

Fig. 2 shows a logical format of cluster allocation information stored in the control memory 232. The cluster allocation information 500 has three areas, a processor number area 510, an allocated cluster number area 520 and a processor attribute area 530. The processor number area stores the processor numbers allocated to the processors 211-214 and 221-224. Fig. 2 shows that, in the present embodiment, 211, 212, 213, 214, 221, 222, 223 and 224 are allocated to the processors as the processor numbers. (In the present embodiment, the processor numbers allocated to the processors correspond to the reference numerals in Fig. 1 for the sake of convenience.) The allocated cluster number area 520 stores the cluster number of the cluster

to which each cluster belongs. Fig. 2 shows that the processors 211, 212, 213 and 214 belongs to the cluster of the cluster number '1' (the cluster A 210 in Fig. 1) and the processors 221, 222, 223 and 224 belong to the cluster of the cluster number '2' (the cluster B 220 in Fig. 1). The processor attribute area 530 stores information indicating the attributes of the processors, that is, information indication which one of the channel control processor (channel control CPU) and the drive control processor (drive control CPU) each processor belongs to. Fig. 2 shows that the processors 211, 212, 221 and 222 are channel control processors and the processors 213, 214, 223 and 224 are drive control processors.

Fig. 3 shows a format of execution cluster indication information stored in the control memory 232. The execution cluster indication information 600 comprises an execution cluster indication area 610 which stores information indicating a cluster mode when the data is transferred. In the present embodiment, the cluster mode includes a cluster non-restricted mode in which any one of the clusters 210 and 220 may be used for the data transfer and a cluster restricted mode in which only one of the clusters may be used. When the data transfer is to be conducted in the cluster non-restricted mode, '1' is stored in the execution cluster indication area 610, and when the data transfer is to be conducted in the cluster restricted mode, '2' is stored.

Fig. 4 shows a format of I/O job execution control information stored in the control memory 232. The I/O job execution control information 800 is provided for each of the disk drives 40 and 41 connected to the disk controller 200. The I/O job execution control information 800 comprises access command request information relating to an access command issued from the host computer 10, executable processor designation information for indicating a processor which may execute the access command issued to the corresponding disk drive and I/O request execution state information for indicating an execution state of the access command to the corresponding disk drive. The access command request information comprises a command type area 811, a drive number area 812, a cylinder number area 813, a tack number area 814 and a record number area 815. The command type area 811 stores a command type of the access command issued to the disk drive corresponding to the I/O job execution control information. The access command may include a read command for reading data from the disk drive and a write command for writing data to the disk drive. The drive number area 812 stores the drive number of the disk drive to be accessed in accordance with the I/O job execution control information. The cylinder area 813, the track number area 814 and the record number area 815 store the cylinder number, the track number and the record number indicating the area to be accessed in the disk drive. The

executable processor designation information comprises executable processor designation bit map 821. The executable processor designation bit map 821 has bits for the processors 211-214 and 221-224. The bit corresponding to the processor which can execute the access command registered in the access command request information is set to '1' (set) ,and the bit corresponding to the processor which cannot execute the access command is set to '0' (reset). The I/O request execution state information has an I/O request execution state area 831 which stores information indicating the execution state of the access command. The I/O request execution state area 831 stores the information indicating the execution state of the access command, that is, '0' when there is no execution request for the access command, '1' when waiting for the process by the drive control processor, '2' when waiting for the process by the channel control processor, and '3' when any process is being executed.

Fig. 5 shows a format of cluster operation mode indication information stored in the control memory 232. The cluster operation mode indication information 700 comprises a cluster number area 710 and an operation mode area 720. The cluster number area 710 stores the cluster number of the cluster. The operation mode area stores the information indicating in which operation mode the corresponding cluster is operating. In the present embodiment, the operation mode may include a

normal operation mode for conducting a normal operation, a suspended mode in which the access command is not accepted and a maintenance operation mode when a processor belonging to the cluster 210 and a processor belonging to the cluster 220 are executing data transfer programs of different versions. In the normal operation mode, '1' is stored in the operation mode area 720, in the suspended mode, '2' is stored, and in the maintenance operation mode, '3' is stored. The cluster operation mode indication information 700 is referred by the processors at a predetermined timing, for example, in synchronism with the search of the executable job.

In the present embodiment, the data transfer control process (the data read process herein) conducted by the processors of the disk controller 200 is executed as follows.

When the host computer 10 is to access to the data stored in the disk drives 40 and 41, it delivers an access command indicating the access to the channel controller 30. The channel controller 30 delivers the access command received from the host computer 10 to one of the channel control processors 211, 212, 221 and 222 of the disk controller 200. It is now assumed that the access command is the read request to the disk drive 40 and the access command is delivered to the channel control processor 211 of the cluster 210.

When the channel control processor 211 receives the access command from the channel controller

30, it refers the I/O request execution state area 831 of the I/O job execution control information corresponding to the disk drive 40, and if it is '0' (no execution request), it updates the information in the I/O request execution state area 831 to '3' (executing). Then, the channel control processor 211 stores the information designated by the access command to the areas 811-815 of the access command request information. When the information stored in the I/O request execution state area 831 is other than '0', the channel control processor 211 returns a busy response to the channel controller 30. The channel control processor 211 refers the execution cluster indication information 600 to check the information stored in the execution cluster indication area 610. When the information stored in the execution cluster indication area 610 is '1' (cluster non-restricted mode), the field (executability indication field) corresponding to the drive control processors 213, 214, 223 and 224, of the executable processor designation bit map area 821 of the I/O job execution control information 800 is set to '1', that is, a bit indicating the executability. On the other hand, when the execution cluster indication area 610 stores '2' (cluster restricted mode), the executability indication fields corresponding to the drive control processors belonging to the same cluster as that to which its own processor belongs, that is, the drive control processors 213 and 214 of the cluster A 210, of

the executable processor designation bit map area 821 are set to '1'. Then, the channel control processor 211 sets '1' indicating the waiting for the execution by the drive control processor in the I/O request execution state area 831.

Any one of the drive control processors (the drive control processor 214) refers the I/O request execution state area 831 of the I/O job execution control information to check whether '1' indicating the waiting for the drive control processor has been set or not. If the I/O request execution state area 831 is set to '1', it further checks whether the executability indication field corresponding to its own processor of the executable processor designation bit map area of the I/O execution control information 800 is set to '1' indicating the executability. If the corresponding executability indication field is set to '1', the I/O request execution area 831 is set to '3' and the data is read from the target disk drive in accordance with the information stored in the areas (the command type 811, the drive number 812, the cylinder number 813, the track number 814 and the record number 815) of the access command request information and it is stored in the cache memory 232 provided in the shared memory 230. Then, the I/O request execution state area of the I/O job execution control information 800 is set to '2' indicating the waiting for the channel control processor. The execution cluster indication information

600 is referred and if the execution cluster indication area 610 is set to '1' (cluster non-restricted mode), the executability indication fields corresponding to the channel control processors 211, 212, 221 and 222, of the executable processor designation bit map area 821 are set to '1'. On the other hand, if the information stored in the execution cluster indication area 610 is '2' indicating the cluster restricted mode, the drive control processor 223 sets '1' only in the executability indication fields corresponding to the channel control processors 211 and 212 belonging to the same cluster as that to which its own processor belongs, that is, the cluster A 210.

Then, any one of the channel control processors (the channel control processor 212) refers the I/O request execution state area 831 of the I/O job execution control information 800 and checks if it is set to '2' (waiting for the channel control processor) or not. If the I/O request execution state area 831 is set to '2', it further refers the executable processor designation bit map area 821 to check whether the executability indication field corresponding to its own processor is set to '1' or not. If the corresponding bit of the executability indication field is set, the channel control processor 212 sets the I/O request execution state area 831 to '3' and transfers the corresponding data stored in the cache memory 232 to the channel controller 30. Then, it sets the I/O request

execution state area 831 to '0' indication no execution request.

In the data read process by the disk controller of the present embodiment, the process is conducted in the same manner as the data read process in the disk controller of a conventional multi-processor configuration except that each processor refers the execution cluster indication information 600 and the cluster operation mode indication information 700, and refers the executable processor designation information (the executable processor designation bit map 821) and continues the processing while updating. Accordingly, the explanation thereof is omitted.

Fig. 6 shows a flow chart of a maintenance process (program version-up process) of the disk controller 200 of the present embodiment.

In the maintenance process, an operator first stores a new version data transfer control program in the control program storing disk drives 216 and 226 of the service processors 215 and 216 through the maintenance panel 240 (step 110). The operator then commands the start of the control program version-up process to the service processor 215 through the maintenance panel 240 (step 120).

When the service processor 215 receives the command to start the control program version-up process, it sets '2' indicating the cluster restricted mode to the execution cluster indication information 600 stored

in the control memory 232 (step 125). Then, when the channel control processors 211, 212, 213 and 214 receive a new access command from the channel controller 30, they set only the bits corresponding to the processors of the cluster to which its own processor belongs, in the executable processor designation bit map area 821 of the I/O job execution control information 800. Thus, the cluster non-restricted job soon disappears. Thereafter, the service processor 215 executes version-up process of the control programs stored in the local memories of the processors 211, 212, 213 and 214 belonging to the cluster A 210 (step 130). While the version-up process for the data transfer control programs of the processors of the cluster A 210 is executed, the processors 221, 222, 223 and 224 of the cluster B 220 execute the data transfer control process in the cluster restricted mode by the data transfer control programs of old version.

When the version-up process for the data transfer control programs of the processors of the cluster A 210 is completed, the service processor 215 requests the execution of the program version-up process of the processors belonging to the cluster B 220 to the service processor 225 (step 140). When the service processor receives the request, it executes the version-up process of the data transfer control programs stored in the local memories of the processors 221, 222, 223 and 224 of the cluster B 220 (step 145). When the

version-up process is completed, the processors 211, 212, 213 and 214 belonging to the cluster A 210 starts the execution of the new version programs and execute the data transfer control process in the cluster restricted mode while the version-up process of the programs of the processors of the cluster B 220 is executed.

When the service processor 215 confirms the completion of the version-up process of the data transfer control program of the cluster B 220 (step 150), it sets the operation mode area 720 of the cluster operation mode indication information 700 to '1' indicating the normal operation mode and sets the execution cluster indication information 600 to '1' indicating the cluster non-restricted mode (step 155). The maintenance panel 240 displays a message indicating the completion of the version-up process of the data transfer control programs (step 160).

Fig. 7 shows a flow chart of the version-up process of the data transfer control program for the cluster A 210 conducted in the step 130.

The processors 211, 212, 213 and 214 belonging to the cluster A 210 refer the executable processor designation bit map area 821 of the I/O job execution control information 800 of all disk drives to confirm that there is no cluster non-restricted job (step 310). Specifically, they confirm whether the bit indicating the executability to the processors of both the clusters

210 and 220 is set in the executable processor designation bit map area of the I/O job execution control information 800 of each disk drive. The confirmation is effected while each processor recognizes that the cluster restricted mode is indicated by the execution cluster indication information 600 at the completion of execution of the job.

The service processor 215 sets '2' indicating the suspended mode in the operation mode area 720 corresponding to the cluster A 210 of the cluster operation mode indication information 700 (step 315). When the channel control processors 211 and 212 of the cluster A 210 confirm that the operation mode is set in the suspended mode, they inform to the channel controller 30 the non-acceptance of a new access command (step 320). By the process of the step 320, the job to be processed by the processors of the cluster A 210 is no longer registered.

The processors 211, 212, 213 and 214 of the cluster A 210 refer the executable processor designation bit map area 821 of the I/O job execution control information 800 to confirm that the bit corresponding to the processors of the cluster A 210 is not set to '1' indicating the executability (step 325). When the processors 211, 212, 213 and 214 confirm that the bit corresponding to the processors of the cluster A of the executable processor designation bit map area 821 is not set, they informs to the service processor 216 the

completion of the shift of its own processor to the suspended state (step 330).

When the service processor 215 receives the notice of completion of the shift of all processors of the cluster A to the suspended state in the step 330, it sets '3' indicating the maintenance operation mode in the operation mode area 720 corresponding to the cluster A of the cluster operation mode designation information 700 (step 335). When the processors 211, 212, 213 and 214 confirm that the operation mode has been set to the maintenance operation mode, they request the transfer of the data transfer program of a new version to the service processor 215 (step 340). In response to the request, the service processor 215 transfers the data transfer control program of the new version stored in the disk drive 216 in the step 110 (Fig. 6) to the processors 211, 212, 213 and 214 (step 345). The processors 211, 212, 213 and 214 store the received data transfer programs in the local memories of their own processors.

When the data transfer control programs of the new version have been stored in the processors of the cluster A, the service processor 215 sets the information indicating the completion of the storage of the program in the control memory 232 and informs it to the processors. In response to the notice, the processors start the execution of the new programs. When the channel control processors 211 and 212 confirm the

storage of the new data transfer control programs in the local memories of the processors, they inform to the channel controller 30 the resumption of the acceptance of the access command (step 350).

Fig. 8 shows a flow chart of the version-up process of the data transfer control program for the cluster B 220 in the step 145 of Fig. 6.

The service processor 225 sets '2' (suspended mode) in the operation mode area 720 corresponding to the cluster B of the cluster operation mode indication information 700 (step 405). Since the non-existence of the cluster non-restricted job has been confirmed in the step 310 of the version-up process of the data transfer control program for the cluster A, the confirmation is not necessary here.

The channel control processors 221 and 222 belonging to the cluster b 220 refer the cluster operation mode indication information 700, and when they confirm that the operation mode is set to the suspended mode, they inform to the channel controller 30 the non-acceptance of a new access command (step 415). Then, the processors 221, 222, 223 and 224 of the cluster B refer the executable processor designation bit map area 821 of the I/O job execution control information 800 to confirm that the bit corresponding to the processors of the cluster B is not set (step 420). When the processors 221, 222, 223 and 224 confirm that the bit corresponding to the processors of the cluster B of the

executable processor designation bit map area 821 is not set, they inform to the service processor 225 the completion of the shift of their own processors to the suspended state.

When the service processor 225 receives the notice of completion of the shift of all processors of the cluster B to the suspended state, it sets the operation mode area 720 corresponding to the cluster B of the cluster operation mode indication information 700 to '3' (maintenance operation mode) (step 426). When the processors 221, 222, 22 and 224 of the cluster B confirm that the operation modes have been set to the maintenance operation mode, they request the transfer of the data transfer control programs to the service processor 225 (step 427). In response to the request, the service processor 225 transfers the data transfer control program stored in the disk controller 226 to the processors of the cluster B (step 430). The processors 221, 222, 223 and 224 store the transferred data transfer program in the local memories of their own processors.

When the processors of the cluster B confirm that the new data transfer control program has been store in the local memories of all processors of the cluster B, they start the execution of the new data transfer control program. The channel control processors 221 and 222 inform the resumption of the acceptance of the access command to the channel

controller 30 (step 440).

By the maintenance process described above, the data transfer program s executed by the respective processors may be versioned up without interfering the processors executing the data transfer control programs of different versions. Thus, the maintenance process can be conducted without wholly stopping the disk controller.

Fig. 9 shows a configuration of the disk controller in accordance with a second embodiment of the present invention. The present embodiment has a similar configuration to that of the embodiment described above except that the processors of the disk controller 20' are divided into four clusters (cluster A 210, cluster b 220, cluster C 270 and cluster D 280).

Where three or more clusters are included as in the present embodiment, a plurality of combinations of clusters which can distributedly process a job may be defined. In the present embodiment, in order to permit to each processor to recognize the combination of clusters, distributedly processable cluster combination information is provided in the control memory 232 in the shared memory 230.

Figs. 10 and 11 show formats of the distributedly processable cluster combination information. The distributedly processable cluster combination information 100 includes information indicating whether one job can be distributedly processed or not among

clusters, in a field at a crosspoint of cluster numbers indicated on an ordinate and cluster numbers indicated on an abscissa. For example, the distributedly processable cluster combination information shown in Fig. 10 includes information OK indicating the distributed process is accepted in all fields. It indicates that the processors of all clusters have the data transfer control programs of the same version in their local memories and a job my be distributedly processed among any processors of any clusters. The distributedly processable cluster combination information shown in Fig. 11 includes information NG indicating that the distributed process is not permitted in only the fields of combinations of the cluster A and other clusters. Such setting of the distributedly processable cluster combination information is made only when only the processors belonging to the cluster A execute the data transfer program which is different from those of the processors of other clusters.

In the present embodiment, in the data transfer control process, when the executable processor designation bit map 821 of the I/O job control information 800 is to be set, each processor sets the executable processor designation bit map 821 in accordance with the following criterion. Namely, it refers the execution cluster indication information 600, and when the execution cluster indication information area 610 is set to '2', it refers the distributedly

processable cluster combination information 100 and sets '1' in the executability indication field corresponding to the processors belonging to other cluster corresponding to the field having OK set therein. When '1' is set in the execution cluster indication information area 610, the bit '1' is set in the executability indication fields corresponding to all processors.

In accordance with the present embodiment, the maintenance process such as the version-up of data transfer program may be conducted in finer unit. In the present embodiment, when the number of clusters is further increased, the cluster group for which the version-up of the program has been completed is defined as a first set and cluster group for which the version-up of the program has not been completed is defined as a second set, and the maintenance process may be conducted while conducing a closed job within each set.

In the above embodiments, the elements of the disk controller are directly connected to each other although they may be connected through a bus. Fig. 12 shows a configuration of the disk controller when the elements are connected by a bus. In Fig. 12, the processors 211, 212, 213 and 214, the service processor 215 and the shared memory 230 are connected through a bus 201. In the disk controller of this configuration, the maintenance process may be conducted for each cluster in the same manner as that described above so that the modification such as version-up of the control

programs executed by the respective processors may be conducted without wholly stopping the data transfer.

In accordance with the present invention, the programs of different versions are executed by the processors of different clusters to avoid the interference of the programs of different versions. Thus, the system maintenance such as the version-up of the program may be conducted without wholly stopping the operation of the multi-processor system.




WHAT IS CLAIMED IS:

1. A maintenance method of a disk controller connected between a host computer and a disk drive and having a plurality of processors for controlling data transfer between said host computer and said disk drive, comprising the steps of:

dividing said processors into a plurality of clusters;

shifting a cluster mode from a cluster non-restricted mode in which distributed processing of a job among clusters belonging to any clusters is permitted to a cluster restricted mode in which a cluster to which processors permitted to distributedly process a job belong is restricted;

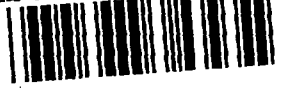
selecting at least one cluster of said plurality of processors as a cluster to be maintained;

versioning up programs to be executed by the processors belonging to said cluster to be maintained; and

executing the version-up programs by a new program by the processors belonging to said cluster to be maintained.

2. A maintenance method according to Claim 1 wherein said version-up step includes a step for stopping the acceptance of a new job by the processors belonging the said cluster to be maintained, a confirmation step for confirming the non-existence of a job to be processed by the processors belonging to said cluster

This Sheet Has Been Replaced



to be maintained, and a storing step for storing the new program in a local memory of each of the processors belonging to said clusters to be maintained.

3. A maintenance method according to Claim 1 further comprising the step of:

repeating said selection step, said version-up step and said execution step sequentially for all clusters.

4. A maintenance method according to Claim 3 further comprising the step of:

setting the cluster mode to said cluster non-restricted mode after the programs of the processors of all clusters have been versioned up.

5. A maintenance method according to Claim 1 further comprising the step of:

executing a job by the processors belonging to the clusters other than the selected clusters in said cluster restricted mode in parallel to said version-up step.

6. A disk controller connected between a host computer and a disk drive comprising:

a plurality of processors each belonging to one of a plurality of predetermined clusters for controlling data transfer between said host computer and said disk drive; and

execution cluster indication means for indication whether processing for an access request of said disk drive issued from said host computer is to be

This Sheet Has Been Replaced

executed in a cluster non-restricted mode in which distributed processing of the access request among all processors is permitted or in a cluster restricted mode in which distributed processing of the access request by processors belonging to different clusters is restricted;

said plurality of processors processing the access request in accordance with the mode designated by said execution cluster indication means.

7. A disk controller according to Claim 6 further comprising:

processor designation information provided for each access request to designate the processors which can execute the processing of the access request.

8. A disk controller according to Claim 7 further comprising:

a terminal device for accepting a program version-up command from an operator;

control means for controlling said execution cluster indication means to indicate the cluster restricted mode in accordance with the program version-up command accepted by said terminal device; and

version-up means for selecting one of said plurality of clusters, suspending the processing of the access request in the processors belonging to the selected cluster and versioning up programs to be executed by the processors belonging to the selected cluster.

This Sheet Has Been Replaced

9. A disk controller according to Claim 8 further comprising:

a memory for storing a copy of data stored in said disk drive;

said plurality of processors including a first plurality of processors for controlling the data transfer between said memory and said host computer and a second plurality of processors for controlling the data transfer between said memory and said disk drive.

10. A disk controller according to Claim 9 wherein each of said clusters includes at least one of said first processors and at least one of said second processors.

11. A disk controller according to Claim 10 further comprising:

cluster combination information for indicating a combination of clusters to which the processors which can distributedly process a job belong;

said processors each setting information for designating the processors which can distributedly process the job to be executed by its own processor in accordance with said processor designation information.

12. A program version-up method for a multi-processor system for distributedly processing a job among a plurality of processors each having a local memory for storing a program, comprising the steps of:

(a) dividing the processors into a plurality of clusters;

This Sheet Has Been Replaced

(b) shifting a cluster mode indicating a job distribution among the clusters from a cluster non-restricted mode in which the distributed processing of the job among processors of different clusters is permitted to a cluster restricted mode in which the distributed processing of the job among processors of different clusters is inhibited;

(c) selecting one cluster from the plurality of clusters as a selected cluster and stopping the acceptance of a new job by the processors in the selected cluster;

(d) versioning up programs stored in the local memories of the processors belonging to the selected cluster by new programs;

(e) resuming the acceptance of new job by the processors belonging to the selected cluster after the completion of the version-up of the programs;

(f) repeating said steps (c) to (e) sequentially for the plurality of clusters; and

(g) returning the cluster mode from the cluster restricted mode to the cluster non-restricted mode.

13. A program version-up method according to Claim 12 wherein said step (c) includes a step for confirming that there in no job which is being executed in the cluster non-restricted mode.

14. A program version-up method according to Claim 13 wherein said step (d) includes a step for confirming

This Sheet Has Been Replaced

* C L M S *

that there is no job to be processed by the processors belonging to the selected cluster.

This Sheet Has Been Replaced