TITLE OF THE INVENTION

INFORMATION PROCESSOR AND REAL-TIME DISTRIBUTED PROCESSING SYSTEM

BACKGROUND OF THE INVENTION

[0001]

[Field of the Invention]

The present invention relates to a constitution of a distributed processing system having a plurality of calculators connected to a network for executing a plurality of programs and more particularly to a processing system requiring real-time control represented by a protection controller for a power system.

[0002]

[Prior Art]

In a conventional general distributed processing system, various task allocation methods have been devised and put into practical use to improve the processing capacity and reliability. For example, in Japanese Application Patent Laid-Open Publication No. Hei 05-257908 and Japanese Application Patent Laid-Open Publication No. Hei 07-282013, a task allocation method is proposed depending on the characteristics and conditions of each calculator. By referring to the capacity of each calculator such as the processing capacity and reliability or the conditions such as the calculation load, task allocation and priority are decided. Tasks different in importance and urgency are

processed by a calculator having proper reliability, thus a distributed process which is reliable as a whole is performed.

[0003]

In addition to the aforementioned constitution, a mechanism of cooperation process may be used as a mutual relation of tasks. For example, in Japanese Application Patent Laid-Open Publication No. Hei 09-231183, a control mechanism for realizing time restrictions for the task cooperation process such as simultaneous start of tasks is provided. However, the mutual relation of tasks is not solved from the viewpoint of reliability.

[0004]

On the other hand, in a real-time processor requiring reliability, a countermeasure by multiplexing has been taken conventionally. For example, in many controllers such as power control for power generation, power transmission, and power distribution and traffic control such as aviation and railroad, malfunction prevention is severely restricted though the real time is necessary. Therefore, in such a real-time process, high reliability is realized by multiplexing of control objects, multiplexing of calculations, and multiplexing of algorithms.

[0005]

As a result, when the aforementioned object is mounted as a highly reliable real-time distributed process, a mutual relation is generated between

respective tasks to be distributed from the viewpoint of reliability. For example, when reliability is realized by mounting a plurality of multiplexed tasks for the same object, it is desirable from the viewpoint of reliability that the tasks are not processed by the same calculation process. Therefore, in the highly reliable real-time distributed process, it is necessary to consider not only the mutual relation of tasks from the viewpoint of calculation amount but also the mutual relation of tasks from the viewpoint of reliability.

[0006]

Conventionally, when a real-time processor requiring reliability is to be mounted in a distributed process, such a relative restriction must be just satisfied in the previous design. For example, in Japanese Application Patent Laid-Open Publication No. Sho 57-3516, a method for mounting a protection controller for a power system in a distributed processing system is proposed. However, adjustment of calculation amount and adjustment of reliability are manually performed at the time of design. Therefore, it is necessary to design a most suitable arrangement for each processor constitution. Further, since a predetermined task arrangement is only used, the effect is lost when the processor constitution is changed.

[0007]

[Problems to be Solved by the Invention]

As described in "Description of the Prior Art", a conventional general distributed processing system can be

applied to a controller requiring reliability.

[0008]

The previous distributed processing arts cannot specify the reliability and urgency of each of calculators or tasks and realize high performance and high reliability.

[0009]

There is no means provided for specifying the mutual relation of tasks for reliability.

[0010]

Therefore, to realize a highly reliable distributed processing system, in addition to the mutual relation of tasks for calculation amount as conventional, a mechanism for reflecting the mutual relation for reliability from the viewpoint of multiplexing is necessary.

SUMMARY OF THE INVENTION

[0011]

An object of the present invention is to specify the reliability and urgency of each of calculators or tasks and realize high performance and high reliability.

[0012]

Another object of the present invention is to allocate tasks so as to suppress the possibility of stopping the protective relay function and realize the protective relay function by very few calculators.

[0013]

[Means of Solving the Problems]

In a real-time processor requiring reliability such as power control and traffic control such as aviation and railroad, to keep the normal operation for an error in calculators, it is necessary to realize fail safe backup by multiplexing. When this processing object is to be realized by a distributed processing system, only by making the system itself fault tolerant, malfunctions cannot be prevented. It is necessary to realize fail safe backup by mounting a plurality of tasks on the basis of the concept of multiplexing of calculations and multiplexing of algorithms.

[0014]

In this case, it is necessary to consider the relation of the plurality of tasks. As a mutual relation of tasks, calculation by the same calculator may be restricted or it may not be suitable. Therefore, in the highly reliable real-time distributed processing system of the present invention, an execution calculator control mechanism for controlling this mutual relation is installed.

[0015]

The execution calculator control mechanism, to judge the calculator by which each task is to be executed, controls the urgency and importance of the execution time of each task and the degree of exclusiveness of reliability which is a mutual relation of tasks in the data base.

[0016]

The execution calculator control mechanism can judge a most suitable task arrangement for the condition of the distributed processing system by referring to the regulations in the data base.

[0017]

In consideration of the urgency and importance of the execution time of each task, a process of maximizing the use efficiency of CPU, giving priority to a most important task, and selecting a task to be executed without impairing the degree of exclusiveness of reliability which is a mutual relation of tasks and a process of selecting a calculator to which the concerned task is to be allocated can be executed.

[0018]

Further, when task scheduling is executed on the aforementioned distributed processing system, high reliability can be always realized deliberately. At the time of failure of a calculator or maintenance, function maintenance for realizing consistency of the CPU use efficiency and the reliability is obtained.

[0019]

Further, the present invention is an information processor for processing a plurality of tasks for realizing the protective relay function by a plurality of calculators connected by a network, which has a means for storing combinations of tasks to be processed by a specific calculator among the plurality of tasks and a means for selecting any one of the combinations of tasks

to be processed by the specific calculator and executing the tasks of the selected combination by the specific calculator.

[0020]

Further, the present invention is an information processor for processing a plurality of tasks for realizing the protective relay function by a plurality of calculators connected by a network, which has a means for storing combinations of tasks not to be processed by a specific calculator among the plurality of tasks and a means for executing the tasks of the combinations excluding the combinations of tasks not to be processed by the specific calculator by the specific calculator.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a constitution of a highly reliable real-time distributed processing system to which the present invention is applied.

Figure 2 shows another configuration No. 1 of the highly reliable real-time distributed processing system.

Figure 3 shows another configuration No. 2 of the highly reliable real-time distributed processing system.

Figure 4 shows operation No. 1 when the device fails.

Figure 5 shows operation No. 2 when the device fails.

Figure 6 shows an example of task scheduling by the highly reliable real-time distributed processing system.

Figure 7 shows an example of a schematic view of a data base of an execution calculator control mechanism.

Figure 8 shows time section No. 1 of task scheduling.

Figure 9 shows time section No. 2 of task scheduling.

Figure 10 shows time section No. 3 of task scheduling.

Figure 11 shows a design tool from an external communication network.

Figure 12 shows a constitution of a distributed processing system having communication channel monitoring tasks.

Figure 13 shows a constitution of a digital protective relay device to which the present invention is applied.

Figure 14 shows an example of a schematic view of a data base of an execution calculator control mechanism of the digital protective relay.

Figure 15 shows an example of a schematic view of a data base of an input-output control mechanism of the digital protective relay.

Figure 16 shows an example of a text form of the data base of the execution calculator control mechanism.

Figure 17 shows an example of a text form of the data base of the input-output control mechanism.

Figure 18 shows an example (system configuration) of the digital protective relay device to which the present invention is applied.

Figure 19 is a conceptual view of tasks in an example of the digital protective relay device.

Figure 20 shows operation flow at the start time of an execution calculator control mechanism in an example

of the digital protective relay device.

Figure 21 shows initial task arrangement at the start time in an example of the digital protective relay device.

Figure 22 shows normal operation flow of the execution calculator control mechanism in an example of the digital protective relay device.

Figure 23 shows an assumption of a calculator error in an example of the digital protective relay device.

Figure 24 shows a fault tolerant operation (1) in an example of the digital protective relay device.

Figure 25 shows a fault tolerant operation (2) in an example of the digital protective relay device.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0021]

The embodiments of the present invention will be explained in detail hereunder with reference to the accompanying drawings. The first embodiment is shown in Fig. 1.

[0022]

A plurality of calculators, three calculators 102 to 104 in this example, are connected to a network via a communication network 101. Each calculator indicated here is a calculator for executing digital calculations. One of the calculators independently has a basic constitution such as a power source, a CPU, and a memory and it is a device which is not affected on the operation capacity by stop or failure of any other calculator. Further, each

calculator has the same program execution environment, so that the independent program module can be executed by every calculator. In the drawing, box-shaped numerals 105 to 109 described inside the calculators 102 to 104 abstractly express the program module to be executed inside each calculator. Each of the calculators can execute a plurality of independent program modules at the same time.

[0023]

It is requested to execute a plurality of tasks using the plurality of calculators mentioned above. For each of the tasks, the execution time is restricted. In the example shown in Fig. 1, it is requested to execute all the five tasks 105 to 109 of tasks 1 to 5.

[0024]

With respect to the plurality of task mentioned above, as a mutual relation of tasks, calculation by the same calculator may be restricted or it may not be suitable. For example, when one function is mounted so as to compute the AND of calculation results of multiplexed tasks so as to prevent malfunctions, it is not desirable to calculate these tasks by the same calculator for reason of reliability. Hereinafter, it is called an inter-task restriction condition for reliability.

[0025]

Execution calculator control mechanisms 110 to 112 mounted in one or a plurality of processors perform a process of executing allocation of tasks in consideration

of the rule of the inter-task restriction condition for reliability. As a realization means therefor, there are some methods available.

[0026]

Firstly, as shown in Fig. 1, it can be realized by a constitution that the execution calculator control mechanisms 110 to 112 hold a data base. The execution calculator control mechanisms obtain inter-task restriction conditions for reliability of the respective tasks beforehand and store them as data base contents 113.

[0027]

The data base contents describe the execution time restrictions and inter-task restriction conditions for reliability of the respective tasks. In Fig. 1, a description 114 concerning the task 1 represents an execution time evaluation index 115 concerning the execution time restriction for deciding the real-time capacity and an execution calculator evaluation index 116 concerning the relation between the task 1 and the task 4.

[0028]

According to this, in the task 1, the urgency and importance of the execution time are specified by the execution time evaluation index 115 and the degree of exclusiveness for reliability from the task 4 is specified by the execution calculator evaluation index 116. In the same way, the other respective tasks and mutual tasks are described. Namely, the processor is structured so as to obtain information for judging the

urgency, importance, and reliability of the processing contents composed of a plurality of tasks from the data base contents 113.

[0029]

The execution calculator control mechanisms 110 to 112 obtain the information of the data base contents 113 and judge the calculators for executing the respective tasks. In consideration of the execution time restrictions of the respective tasks, the execution calculator control mechanisms 110 to 112 execute a process (task selection process) of maximizing the use efficiency of CPU, giving priority to a most important task, and selecting a task to be executed without impairing the inter-task control condition for reliability and a process (calculator selection process) of selecting a calculator to which the concerned task is to be allocated.

[0030]

In the example shown in Fig. 1, severe inter-task restriction conditions for reliability are imposed on the tasks 1 and 4, the tasks 1 and 2, the tasks 2 and 3, and the tasks 3 and 5. Therefore, a task mounting pattern is prepared not so as to mount these combinations on the same calculator. One of the solutions is the task allocation pattern shown in Fig. 1. The tasks 1 and 3 are allocated to the calculator X, the tasks 2 and 4 to the calculator Y, and the task 5 to the calculator Z. The aforementioned inter-task restriction conditions for

reliability are all satisfied.

[0031]

Another means by which the execution calculator control mechanisms consider the rules of inter-task restriction conditions for reliability will be explained next. The execution calculator control mechanisms can obtain the aforementioned effect without having the data base.

[0032]

In the example shown in Fig. 2, each task has an execution time evaluation index for the task and an execution calculator evaluation index for each of the mutual relations between the other tasks in the program module. The communication network 101 and the plurality of calculators 102 to 104 are the same in the constitution as those explained in Fig. 1. A difference from the embodiment shown in Fig. 1 is that tasks 201 to 205 contain the aforementioned indexes. To execution calculator control mechanisms 206 to 208 shown in Fig. 2, a mechanism for obtaining indexes contained in the tasks 201 to 205 via communication is added. Therefore, the execution calculator control mechanisms communicate from the tasks when necessary and refer to transmitted indexes, thereby can execute the aforementioned task selection process and calculator selection process.

[0033]

As a means by which the execution calculator control mechanisms consider the rules of inter-task restriction

conditions for reliability, another method will be explained. Even if the indexes for all the other tasks are not contained in the program module beforehand, the aforementioned effect can be obtained. In the example shown in Fig. 3, a method that each task has a task name on the basis of a predetermined syntax is shown. The communication network 101 and the plurality of calculators 102 to 104 are the same in the constitution as those explained in Fig. 1. A difference from the embodiment shown in Fig. 1 is that tasks 301 to 305 have preset task names. Execution calculator control mechanisms 306 to 308 shown in Fig. 3 refer to the task names of the tasks 301 to 305 and calculate execution calculator evaluation indexes. To perform the operation, the execution calculator control mechanisms 306 to 308 hold a predetermined syntax rule 309. Syntaxes 310 to 312 describe exclusive restriction conditions 313 by the same calculator as a relation of two task name patterns.

[0034]

The execution calculator control mechanisms 306 to 308 recognize the task names at the stage of installation of the program module. The execution calculator control mechanisms 306 to 308 refer to the syntax rule 309, extract the syntax that the task name matches from the syntaxes 310 to 312, and can obtain the inter-task restriction condition for reliability. In the same way as with the embodiments shown in Figs. 1 and 2, the execution calculator control mechanisms 306 to 308 can

execute the task selection process and calculator selection process.

[0035]

As shown in Figs. 1 to 3, there is no need to execute the execution calculator control mechanisms by one calculator. As a function of the OS, the mechanisms can be mounted in all the calculators. A master-slave constitution may be used. Furthermore, they may be mounted in a computer intermittently connected via an external communication channel instead of the communication network 101, for example, a remote control console.

[0036]

By introducing the execution calculator control mechanisms explained above, consistency of the restriction conditions of the real-time capacity and reliability can be realized according to the constitution and capacity of calculators.

[0037]

Next, a constitution for realizing a utilization function of a task maintaining the high reliability will be explained using the present invention. In addition to the functions of the task selection process and calculator selection process of the present invention, a distributed processing system having the functions of task regeneration and calculator movement is supposed.

[0038]

The constitution will be explained by referring to

Fig. 4. In the example shown in Fig. 4, a case that a fault occurs in the calculator X is supposed on the basis of the example shown in Fig. 1. Except the occurrence of fault in the calculator X, a communication channel 401, a plurality of calculators 402 to 404, tasks 405 to 409, and execution calculator control mechanisms 410 to 412 are the same as those shown in Fig. 1.

[0039]

In the example shown in this drawing, the tasks 1 to 5 are tasks requiring the same calculation amount. However, with respect to the urgency and importance of the tasks, it is supposed that the task 5 is in a state of the highest priority. Further, in the calculators Y and Z, only four among the tasks 1 to 5 can be executed.

[0040]

When a fault of the calculator X is detected in this state by regular observation of the execution calculator control mechanism 410 mounted in the calculator X or the execution calculator control mechanisms 411 and 412 mounted in the calculators Y and Z, the execution calculator control mechanisms 411 and 412 execute the task selection process and calculator selection process again in correspondence with the fault of the calculator X.

[0041]

The condition of the distributed processing system after re-execution is shown in Fig. 5. To a communication channel 501, a faulty calculator 502 and normal

calculators 503 and 504 are connected. The task selection process and calculator selection process are performed by execution calculator control mechanisms 505 and 506. The allocation of three tasks 507 to 509 (tasks 2, 4, 5) is not changed from that before occurrence of the fault. A difference from that before occurrence of the fault is that a task 510 is newly allocated to the calculator 504 in place of the task 1 calculated by the calculator 502.

[0042]

This allocation is derived from the combination for realizing consistency of the property of each task with the property of the mutual relation of tasks. The task selection process maximizes the importance and urgency of the whole tasks and the calculator selection process maximizes the reliability of a combination of tasks to be mounted in the same calculator. In the example shown in Fig. 5, the calculation amount of each task is the same, so that in two calculators, a plurality of mounting patterns for execution of the four tasks may be considered. However, it can be obtained uniquely that the mounting pattern most satisfying the execution time evaluation and execution calculator evaluation shown in Fig. 1 among the mounting patterns is:

calculator Y (task 2, task 4), calculator Z (task 1, task 5)

When the execution calculator control mechanisms process such task selection and calculator selection, as shown in Fig. 5, with the fault detection as a trigger, a

function for automatically allocating the alternative task 510 of the task 1 calculated by the calculator 502 to the calculator 504 can be realized.

[0043]

Next, a constitution for realizing a function for executing task scheduling with the high reliability kept will be explained using the present invention. The execution calculator control mechanisms for executing task scheduling, in addition to the functions previously explained, confirm existence of tasks to be generated in the future from the current calculation section. The execution calculator control mechanisms execute the aforementioned task selection process and calculator selection process for each time section that tasks to be processed are generated, obtain a most suitable solution for the plurality of time sections, and realize task scheduling.

[0044]

The embodiment will be explained by referring to Figs. 6 to 10. Fig. 6 conceptually shows schedules of generation of the tasks confirmed by the execution calculator control mechanisms and results of task scheduling for allocation of these tasks to the calculators. In the drawing, a transverse axis 601 indicates a time axis for indicating generation of tasks concerning the task 1 and processing therefor. An arrow 602 under the transverse axis 601 indicates an occurrence of an execution request of the task 1. A rectangle 603

above the transverse axis 601 indicates execution of the task 1. The calculator name inside the rectangle 603 indicates a calculator taking charge of execution. This case indicates that the task-1 execution request 602 is allocated to the execution 603 by the calculator X. Further, when execution corresponding to a new execution request 604 is to be allocated during the execution 603, it is indicated like a rectangle 605.

[0045]

In a first time section 606, the process can be realized by the task allocation by the execution calculator control mechanisms which is previously explained. The task request occurring in the time section 604 has the same content as that of the first embodiment. Namely, the task allocation constitution shown in Fig. 1 is used. Namely, the allocation of calculators of the tasks 1 to 5 shown in Fig. 6 coincides with the content of Fig. 1.

[0046]

In a second time section 607, the task-1 request 604 occurs newly. In correspondence with this request, the task scheduling by the execution calculator control mechanisms newly plans task allocation in the second time section by keeping the inter-task restriction condition for reliability shown in Fig. 7. In addition to the task allocation condition shown in Fig. 1, the task 1 is newly allocated to the calculator Z like the execution 605. The result is the condition shown in Fig. 8. In Fig. 8, in a

calculator Z 801, there is no inter-task restriction condition for reliability between a newly allocated task 802 (task indicated by a thick line in the drawing) and a task 803 in execution. As a whole, the inter-task restriction condition for reliability shown in Fig. 7 is satisfied.

[0047]

At the point of time of a third time section 608, the task-1 execution process 603 executed by the calculator X and the task-2 execution process 610 executed by the calculator Y are finished and inversely, a task-5 execution request 611 is received. The task scheduling by the execution calculator control mechanisms executes the task allocation shown in Fig. 9. From the viewpoint of the processing capacity of the calculators, the calculator capable of executing the task 5 is a calculator X 901 or a calculator Y 902. However, when the task 5 is allocated to the calculator X, in consideration of a task 903 (task 3) in execution, the inter-task restriction condition for reliability is not followed. As an allocation result for obtaining highest evaluation, execution of a new task 904 (task 5) by the calculator Y is selected. The reason is that there is no problem imposed in the reliability with a task 905 (task 4) in execution.

[0048]

At the point of time of a fourth time section 609, the execution processes of the tasks other than a task-5

execution process 612 executed by the calculator Y are all finished. Here, an execution request 613 of the tasks 1 to 5 is received. By the same process as that mentioned above, the task scheduling by the execution calculator control mechanisms executes the task allocation shown in Fig. 10. Other than a task 1002 (task 5) in execution by a calculator Y 1001, five tasks 1003 to 1007 are allocated newly. There is no violation in the inter-task restriction condition for reliability.

[0049]

As mentioned above, in each time section where an execution request occurs, when the task selection and calculator selection are processed together with the task in execution at present, task scheduling can be realized.

[0050]

With respect to the method for executing task scheduling by allocation of tasks in a plurality of time sections, a plurality of methods may be considered depending on a difference in the mounting method of the optimization means. In the aforementioned example, the task scheduling is executed successively along the flow of the time section. Separately from this, by solving a problem of a combination for optimizing the task allocation in all the time sections simultaneously and minimizing a difference in the task allocation condition between adjacent time sections, the task scheduling can be executed.

[0051]

By this task scheduling function, the distributed processing system always can realize high reliability deliberately. For example, even when a start request for a new task other than the tasks confirmed by the execution calculator control mechanisms occurs in the system or a start command for a new task is issued externally, the task scheduling is re-executed with the change reflected, thus the high reliability after changing can be kept. Further, for a fault of the calculators or maintenance, even when a command is issued for the utilization condition of the calculators via an external calculator, the utilization condition of the calculators is supposed according to the command content and then the task scheduling is re-executed, thus the function for realizing consistency of the use efficiency of CPU with the reliability can be maintained.

[0052]

Next, as a second embodiment, a development tool to which the present invention is applied is shown in Fig. 11.

[0053]

A processing system to be developed is the same as that of the embodiment shown in Fig. 1. Via a communication channel 1101 used for control, a plurality of calculators, that is, three calculators 1102 to 1104 in this example are connected to a network. In these calculators, it is requested to execute a plurality of tasks 1105 to 1109 that the execution time is restricted.

[0054]

To design this processing system, an external development calculator is used. A development calculator 1111 is connected via an external communication network 1110 not to be used for control. By use of this development calculator, the basic design of the aforementioned tasks and setting of various set values are executed.

[0055]

The development calculator 1111 has an execution calculator control mechanism 1112 having the same function as that explained in the embodiment shown in Fig. 1. Therefore, by instructing the equipment constitution of the processing systems 1101 to 1104 to be developed, the initial task allocation for realizing consistency of the CPU use efficiency with the reliability can be executed.

[0056]

By use of such a development tool, an initial design for executing the selection process of tasks to be executed and the selection process of calculators for calculating the selected tasks can be performed and a highly reliable real-time distributed processing system can be designed. Further, tasks are added, deleted, or corrected via an external communication network, thus most of the functions held by the highly reliable real-time distributed processing system can be realized without adding execution calculator control mechanisms to

the calculators constituting the processing system.

[0057]

As a third embodiment, a distributed processing system having a function for monitoring the communication channel by utilizing the present invention will be explained hereunder.

[0058]

To constitute the processing system, a plurality of calculators, that is, three calculators 1202 to 1204 in this example are connected to a network via a communication channel 1201.

[0059]

Here, a communication channel monitoring task for verifying the communication channel is added and the processing system is made more highly reliable. As a communication channel monitoring method, a plurality of tasks for executing communication via a communication channel to be monitored is formed as a set and the validity of mutual communication contents is verified, thus the normalcy of the communication channel is monitored.

[0060]

In this case, allocation of these monitoring tasks and task scheduling arise a complicated problem. Firstly, to realize a monitor function, it is preferable to execute not only monitoring tasks necessary at its minimum but also as many monitoring tasks to be executed as possible. However, the task execution which is an

original processing object must not be disturbed strictly. Secondly, the individual tasks constituting a set must be executed by a calculator different from that of the tasks of the same set. Allocation and scheduling of such monitoring tasks having a plurality of restrictions can be realized using the present invention.

[0061]

In the same way as with the example explained in the embodiment shown in Fig. 1, the plurality of calculators in the processing system have execution calculator control mechanisms 1205 to 1207. These execution calculator control mechanisms control the meanings of the tasks to be executed as data base contents 1208. The data base contents describe the execution time restriction of each task and the inter-task restriction condition for reliability. In Fig. 12, a description 1209 concerning the task 1 represents an execution time evaluation index 1210 concerning the execution time restriction for deciding the real-time capacity and an execution calculator evaluation index 1211 concerning the relation of the tasks 1 and 4.

[0062]

In this example, it is supposed that there are two differences from the embodiment shown in Fig. 1. Firstly, it is supposed that an execution time evaluation index 1213 concerning a communication channel monitoring task A 1212, an execution time evaluation index 1215 concerning a communication channel monitoring task B 1214, and an

execution calculator evaluation index 1216 between the tasks are added.

[0063]

Secondly, an execution time evaluation index 1217 concerning the task 5 is set in a lower priority concerning execution than the execution time evaluation indexes of the tasks 1 to 4 and the execution time evaluation indexes of the monitoring tasks. Namely, it is set that the task 5 is not a necessary task.

[0064]

Here, two tasks are formed as a set and a function for monitoring the communication channel is added. The monitoring function is necessary at its minimum, so that a higher priority of execution than that of the task 5 is set.

[0065]

When such a setting is given to the execution calculator control mechanisms 1205 to 1207, task allocation is executed for the three calculators 1202 to 1204 like 1220 to 1225. The execution of the task 5 is omitted. Further, with respect to the other six tasks, the arrangement of the respective tasks is decided with the mutual exclusive relation satisfied.

[0066]

When the tasks for monitoring the communication channel are set as a part of control objects of the execution calculator control mechanisms like this, a highly reliable real-time distributed processing system

for seizing the normalcy of the communication channel can be realized.

[0067]

Furthermore, by use of the task scheduling using the execution calculator control mechanisms, the communication channel can be monitored periodically. By periodically requesting to process a set of monitoring tasks of lower execution priority, within the excessive calculation capacity of the processing system, as high reliability as possible can be realized.

[0068]

In the aforementioned embodiment, monitoring of the communication channel is described. By the same method, a monitoring function for a calculator can be mounted. Firstly, as calculator monitoring tasks, a test for checking a predetermined calculation formula is executed and a plurality of other calculators connected by the network and a plurality of tasks for verifying results via communication are prepared. Then, to the execution calculator control mechanisms, a restriction or condition that each of the calculator monitoring tasks is executed by a different calculator respectively is given. By doing this, the calculator monitoring tasks are automatically subjected to suitable task allocation and scheduling and identification of an abnormal calculator or detection of a fault of the network can be realized.

[0069]

As a fourth embodiment, using a processing system to

which the present invention is applied, an example that a digital protective relay device for protecting one or a plurality of power devices is realized will be considered. It is shown in Fig. 13.

[0070]

Firstly, as devices to be input, with respect to each of power devices 1301 to be protected and controlled such as a bus, a transformer, and a transmission line, voltage and current information 1302 used for the trouble detection function necessary for equipment protection and the observation function necessary for various types of control is transferred to input devices 1303 connected to them. In the same way, as devices to be output, with respect to power devices 1304 to be controlled such as a circuit breaker, a control signal 1305 such as a breaking command is transmitted to output devices 1306 connected to the respective devices.

[0071]

These input devices 1303 and output devices 1306 are connected to a plurality of calculators 1308 to 1311 via a communication channel 1307. In this case, the calculators are calculators for executing digital calculations. One of the calculators independently has a basic constitution such as a power source, a CPU, and a memory and it is a device which is not affected on the operation capacity by stop or failure of any other calculator. Further, each calculator has the same program execution environment, so that the independent program

module can be executed by every calculator. In the drawing, box-shaped numerals 1311 described inside the plurality of calculators 1308 to 1310 abstractly express the program module to be executed inside each calculator. Each of the calculators can execute one or a plurality of independent program modules at the same time.

[0072]

Allocation of execution of the respective program modules to the calculators is controlled by an execution calculator control mechanism 1312. This is the same function as that of the execution calculator control mechanisms of the aforementioned first embodiment shown in Fig. 1. Therefore, in the data base in the execution calculator control mechanism, the execution time restriction of each task and the inter-task restriction condition for reliability are described.

[0073]

In Fig. 13, the execution calculator control mechanism controls the data base contents shown in Fig. 14. The execution time restriction of each task and the inter-task restriction condition for reliability are described. In the description shown in Fig. 14, a description 1401 concerning each task includes an execution time evaluation index 1402 concerning the execution time restriction for deciding the real-time capacity and an execution calculator evaluation index 1403 concerning the other tasks.

[0074]

For example, among the calculators for protecting the device A, the program module "Device A, main, normal system" for always executing the main protection calculation and "Device A, main, standby system" standing by so as to immediately switch and back up when the program module becomes inexecutable must be executed by different calculators. Accordingly, an inter-task restriction for reliability is imposed to the relation of the task "Device A, main, normal system" and the task "Device A, main, standby system".

[0075]

In the same way, to prevent malfunctions, "Device A, main, normal system" and "Device A, fail safe, normal system" must be executed by different calculators. Therefore, an inter-task restriction for reliability is imposed to the relation of the two tasks. When such a calculator restriction is imposed, unless the two calculators in which the program module meaning the two tasks is executed enter an inoperable state in the same time zone, no possibility of malfunctions of the protection function for the device A is generated.

[0076]

Further, even if a single fault of the device occurs, task allocation for keeping the high reliability more can be imposed. For example, even in a situation that any of the tasks of the normal system is switched to the standby system due to a device fault, if the main tasks and fail safe tasks are designed so as to be executed by different

calculators, in the task relation of "Device A, main, normal system" and "Device A, fail safe, standby system", as an inter-task condition for reliability, a task execution calculator evaluation index can be set. In this case, an exclusive condition 1404 may be set in a lower priority than that of the inter-task restriction for reliability 1403.

[0077]

As mentioned above, the design shown in Fig. 14 is executed for all the tasks concerning the device A. Further, according to the same concept, the tasks concerning the device B are designed. By the execution calculator control mechanism having this data base, the calculator for executing each program module is properly decided. As a result, the task allocation shown in Fig. 13 is obtained. It cannot be realized from the number of calculators to satisfy all the inter-task restrictions for reliability and imposing of conditions. Arrangement for satisfying the higher-priority restriction 1403 and satisfying the exclusive condition 1404 as far as possible at the same time is set. Utilizing the processing capacity of the whole system at an optional point of time, a highly reliable system can be realized.

[0078]

Furthermore, in the aforementioned embodiment, a control mechanism is also installed so as to control the aforementioned input and output relation. As shown in Fig. 13, the input-output devices J to M are connected fixedly

to the devices A to D. The control mechanism is structured so as to transmit or receive input-output information of the devices via the multi-drop communication network. An input-output relation control mechanism 1313 controls the situation that the tasks in the calculator use the V and I information of any input device and transmit a control signal to any output device in the data base.

[0079]

As an example, the operation on the system shown in Fig. 13 in which the input device J 1314 and the input device J' 1315 are prepared as input devices connected to the device A will be considered. The input-output relation control mechanism receives a designation 1503 that an input device J 1501 and an input device J' 1502 are in an alternative relation by the data base contents schematically shown in Fig. 15. Therefore, upon detection of a fault of the input device J, the input-output relation control mechanism instructs each calculator so as to substitute input data to be transmitted by the input device J' for input data to be transmitted by the input device J. By doing this, an effect of the fault of the input device J can be avoided.

[0080]

In the above explanation, the contents of the data base as shown in Figs. 14 and 15 are mounted by designating on the rule base at the design stage of the processing system or at the time of maintenance thereof.

The method will be explained by referring to an example.

[0081]

Fig. 16 shows a data format to be given to the execution calculator control mechanism. The contents represented schematically in Fig. 14 can be given by text data according to a predetermined syntax.

[0082]

In Fig. 16, the first line 1601 indicates a comment statement. On the second line 1602 and the subsequent lines, the protection functions concerning the devices A and B are set.

[0083]

Firstly, the tasks concerning protection of the device A are declared. By the description on the third line 1603, the program module for processing the function of the device A, main, normal system is declared under the name of "module_A1" as a task for performing the normal process. The character string "Device A, main, normal system" behind the mark # is a comment statement. Furthermore, by the description on the fourth line 1604, the program module for processing the function of the device A, main, standby system is declared under the name of "module_A2" as a task for performing the standby process. In the same way, for the device A, four tasks in total are declared. Description concerning the real-time capacity of each task, for example, the task process start time, end time, and priority are separately set.

[0084]

Next, the relation between the normal system concerning protection of the device A and the standby system is declared. On the seventh line 1605, it is defined that module_A1 is backed up by module_A2. Hereinafter, in the same way, backup of the fail safe function will be described.

[0085]

At the end of the definition data of the device A, the exclusive relation between the tasks is declared. On the ninth line 1606, a restriction that the "Device A, main, normal system" function and the "Device A, fail safe, normal system" function are allocated to different calculators is defined in a highest priority. On the eleventh line 1607, it is defined as a condition in a lower priority that it is desirable to allocate the "Device A, main, normal system" function and the "Device A, fail safe, normal system" function to different calculators. Hereinafter, in the same way, five relations existing between the four tasks are defined.

[0086]

In the text data given to the execution calculator control mechanism shown in Fig. 16, in the same way as with protection concerning the device A, the setting concerning the device B is also described. In this way, the rule of the inter-task restriction condition for reliability schematically shown in Fig. 14 can be given.

[0087]

Next, the data format to be given to the input-output

relation control mechanism will be explained. The contents represented schematically in Fig. 15 can be given by the text data as shown in Fig. 17 according to a predetermined syntax.

[0088]

In Fig. 17, the first line 1701 indicates a comment statement. On the second line 1702 and the subsequent lines, setting concerning the input devices, setting concerning the output device, and setting concerning input and output are executed.

[0089]

Firstly, the input amount to be transmitted by the input device J is declared. By the description on the third line 1703, a name of "a_bus_v1" is set in the first input amount of the input device J. Hereinafter, all the input amounts transmitted by the input device J are declared and the names are set.

[0090]

In the same way, the output amount received by the output device is also declared. On the seventeenth line 1704 and the subsequent lines, the output device L is defined. In the same way as with the description concerning the input amount, on the eighteenth line 1705, the output amount is declared and the name is set.

[0091]

Finally, for each task, the input-output amount used by the task is set. On the twenty-second line 1706 and the subsequent lines, the input-output amount of the task

module_A1 is set. The arguments possessed by the program module for realizing the aforementioned task are sequentially set. On the twenty-third line 1707, it is defined to use input a_bus_v1 as a first argument. In the same way, the input-output amounts required by the aforementioned program module are all defined. Then, a one which can be substituted for the input-output amount used by the task is set. On the twenty-eighth line 1708, it is defined to use a2_bus_v1 instead when an error occurs in the input a_bus_v1 used as a first argument.

[0092]

By the text data given to the input-output relation control mechanism shown in Fig. 17, the relation of the input-output amount schematically shown in Fig. 15 and the rule concerning the substitution therefor can be given.

[0093]

When a representation method on the basis of the syntax as shown in Figs. 16 and 17 is introduced, a designer and an operator can write a rule directly in the text. Further, by a mechanism for outputting a text file in the aforementioned representation from a design and an operation tool having GUI, the designer and operator obtain an environment of easier design and operation.

[0094]

As mentioned above, when a highly reliable real-time distributed processing system having the execution calculator control mechanism indicated in Embodiment 1

controls tasks for realizing a digital protective relay for a power device, a highly reliable protective relay system can be realized.

[0095]

There are a plurality of mutual relations of tasks to be controlled. For example, as explained in Embodiment 3 mentioned above, as a mutual relation between the normal system tasks of the first calculation object and the standby system tasks, a restriction or condition of execution by different calculators may be cited. According to this, a digital protective relay for preventing the protective relay function from stopping when a calculator fails is realized.

[0096]

Further, in a protective relay having a main detection task for realizing a main detection function for the protective relay and a trouble detection task for realizing a trouble detection function as a fail safe element, as a mutual relation between the main detection task and the trouble detection task of the same protective relay, a restriction or condition of execution by different calculators may be cited. According to this, a digital protective relay for preventing malfunctions is realized.

[0097]

In addition to it, in a protective relay having a package bus protective task for realizing a package bus protection function of a bus protective relay and a split

bus protective task for realizing a split bus protection function, as a mutual relation between the package bus protective task and the split bus protective task for the same bus, a restriction or condition of execution by different calculators may be cited. According to this, a digital protective relay for preventing the bus protective relay from malfunctions is realized.

[0098]

Furthermore, with respect to each of transmission lines composed of parallel circuits, in a protective relay having transmission line tasks for realizing a transmission line protective relay function, a restriction or condition of execution by different calculators for a plurality of transmission line tasks for transmission line protection belonging to the same parallel circuit may be cited. According to this, a digital protective relay for minimizing the effect of a calculator fault for protection of the parallel transmission lines is realized.

[0099]

A highly reliable protective relay system controlled by the highly reliable real-time distributed processing system as mentioned above will be explained as a fifth embodiment by referring to the drawing.

[0100]

Firstly, the whole system image is shown in Fig. 18. Via a light communication channel 1801 for realizing communication of a light medium, a plurality of

calculators, for example, three calculators 1802 to 1804 in this example are connected to a network. Each calculator indicated here is a calculator for executing digital calculations. One of the calculators independently has a basic constitution such as a power source, a CPU, and a memory and it is a device which is not affected on the operation capacity by stop or failure of any other calculator. Further, each calculator has the same program execution environment, so that the independent program module can be executed by every calculator. Furthermore, each calculator obtains a signal synchronized with the commercial frequency via the light communication network 1801. Therefore, protective relay calculations synchronized with the electrical angle of the commercial frequency can be performed.

[0101]

To the light communication network 1801, not only the calculators but also input-output devices are connected. This configuration is shown in the drawing in consideration of one phase of the power line. On a power line 1805 for supplying the commercial power, a circuit breaker 1806 and a transformer 1807 are installed and the respective devices have input-output devices 1808 to 1811. The system has a configuration that these input-output devices are connected to the light communication network 1801.

[0102]

The input-output devices 1808 to 1811 have an analog-

digital conversion function, a communication function, and an electricity-light conversion function and are mounted to the circuit breaker 1806 and the transformer 1807. The input-output devices 1808 and 1809 realize a communication function for voltage-current information from the circuit breaker to the calculators and breaking control information from the calculators to the circuit breaker as a double communication terminal of the circuit breaker 1806. In the same way, the input-output devices 1810 and 1811 realize a communication function as a double communication terminal of the transformer 1807. The respective input-output devices and the calculators are independent of each other from the viewpoint of power source and from the viewpoint of insulation.

[0103]

In the same way as with the embodiments explained previously, in this system, the execution calculator control mechanism 1812 and the input-output relation control mechanism 1813 are mounted in one or a plurality of optional calculators.

[0104]

In a highly reliable protective relay system controlled by a highly reliable real-time distributed processing system having the aforementioned constitution, a process of a plurality of tasks having a real-time capacity is requested. Those tasks are represented abstractly in Fig. 19. Box-shaped numerals 1901 to 1906 shown in Fig. 19 indicate tasks processed by the

calculators 1802 to 1804.

The task indicated by the numeral 1901 will be called "Task1" or "TrPrMN" hereinafter. Task1 is a task for realizing the main transformer differential relay element 87T calculation and the sequence calculation for specifying an ignition condition of the element.

[0105]

The task indicated by the numeral 1902 will be called "Task2" or "TrPrFD" hereinafter. Task2 is a task for realizing the different algorithm trouble detection function for fulfilling the fail safe function of Task1 mentioned above and the sequence calculation for specifying an ignition condition of the element.

[0106]

The task indicated by the numeral 1903 will be called "Task3" or "TrPrMN-FT" hereinafter. Task3 is a task for realizing the standby function for fulfilling the fault tolerant function of Task1 mentioned above. When Task1 becomes inexecutable, to back up the function immediately, Task3 stands by continuously by performing calculations of the minimum part of the same algorithm as that of Task1. Task3 executes calculations for continuing a process necessary to hold the preceding value such as reading of the input amount or the digital filtering process.

[0107]

The task indicated by the numeral 1904 will be called "Task4" or "TrPrFD-FT" hereinafter. Task4, in the same

way as with Task3, is a task for realizing the standby function for fulfilling the fault tolerant function of Task2 mentioned above.

[0108]

The task indicated by the numeral 1905 will be called "Task5" or "SS-P/P'" hereinafter. Task5 fulfills the function for monitoring the normalcy of the input-output device P (1808 shown in Fig. 18) and the input-output device P' (1809 shown in Fig. 18) explained in Fig. 18. For example, to the input-output device, a hardware function for superimposing the 12th harmonic component for the commercial frequency on the quantity of electricity (analog value) input from the power line as an inspection signal is added. By doing this, when an optional calculator monitors the inspection harmonic component included in information transmitted or received from the input-output device, the normalcy of the input signal can be monitored.

[0109]

The task indicated by the numeral 1906 will be called "Task6" or "SS-Q/Q'" hereinafter. Task6, in the same way as with Task5, fulfills the function for monitoring the normalcy of the input-output device Q (1810 shown in Fig. 18) and the input-output device Q' (1811 shown in Fig. 18) explained in Fig. 18.

[0110]

For these plurality of tasks, the real-time capacity is requested. In the example shown in Fig. 19, it is

requested to execute the six tasks of Task1 to Task6 within a predetermined execution time using the three calculators 1802 to 1804 shown in Fig. 18.

[0111]

In Fig. 19, the execution time restrictions predetermined for each of the six tasks of Task1 to Task6 are shown in 1907 to 1912. In the same way as with the aforementioned embodiment, the data concerning these execution time restrictions is given to the data base controlled by the execution calculator control mechanism beforehand. The contents of the respective execution time restrictions will be explained hereunder.

[0112]

The content shown in the execution time restriction 1907 is that Task1 has an execution priority of Level 1, and a request for the task occurs at a time frequency of 30 deg, and the period from occurrence to processing is 30 deg. With respect to Level 1, a smaller value indicates a higher priority, so that Level 1 means a highest priority task. The processing period unit deg is expressed in units of an electrical angle for one cycle of the commercial frequency and as explained in Fig. 18, it designates only the time at the time of performing the protective relay calculation synchronized with the electrical angle of the commercial frequency.

[0113]

In the same way, the execution time restriction 1908 designates that Task2 has an execution priority of Level

1, and the occurrence frequency is 30 deg, and the processing period is 30 deg. Namely, it is specified that Task1 and Task2 for realizing the main protection function are highest priority tasks.

[0114]

Further, the execution time restrictions 1909 and 1910 specify that Task3 and Task4 for realizing the fault tolerance of the protection function are priority tasks next to the highest priority, and in the same way, the occurrence frequency is 30 deg, and the processing period is 30 deg.

[0115]

On the other hand, the execution time restrictions 1911 and 1912 have a lowest execution priority in design. It is specified that the occurrence frequency is 10 deg and the process may be completed within a period of 2 seconds. Further, for Task6, a non-periodical task generation rule such that the task is not generated for 1 minute after system start is also mentioned.

[0116]

In this embodiment, in addition to the execution time restrictions specified for the respective tasks explained above, in the same way as with the previous embodiments of the highly reliable real-time distributed processing system, inter-task restriction conditions for reliability for specifying the mutual relation of tasks are necessary as a highly reliable protective relay system.

[0117]

Exclusive restrictions 1913 to 1916 show inter-task restriction conditions depending on the relation of Task1 to Task4. In the same way as with the aforementioned embodiment, the data concerning these inter-task restriction conditions for reliability is given to the data base controlled by the execution calculator control mechanism beforehand. The contents of the respective inter-task restrictions for reliability will be explained hereunder.

[0118]

The content shown in the exclusive restriction 1913 indicates that Task1 and Task3 have an exclusive priority of Level 1. With respect to Level 1, a smaller value indicates a higher priority, so that Level 1 means that the exclusive arrangement of calculators is considered in top priority. The reason is that if Task3 for realizing the fault tolerant function of Task1 is not processed by a different calculator, it is quite meaningless.

[0119]

In the same way, the exclusive restriction 1914 specifies that for Task2 and Task4, the exclusive arrangement of calculators is considered in top priority.

[0120]

On the other hand, the exclusive restriction 1915 indicates that Task2 and Task4 are on the exclusive priority Level 2. The tasks for realizing the fail safe function of Task1 must be processed by a different calculator in principle. However, Task2 is composed of a

different algorithm as a trouble detection system, so that even if it is processed by the same calculator as that of the main detection Task1, it is not meaningless. Therefore, the inter-task restriction for reliability is specified so as to consider the exclusive arrangement of calculators as much as possible.

[0121]

In the same way, the exclusive restriction 1916 specifies the inter-task restriction for reliability for Task3 and Task4 so as to consider the exclusive arrangement of calculators as much as possible.

[0122]

On the basis of the specified contents of the execution time restrictions and inter-task restrictions for reliability explained above, the execution calculator control mechanism decides arrangement of the tasks and controls an execution command. The processing stages thereof will be explained sequentially. Firstly, the operation at the time of system start, that is, the initial operation in this embodiment will be explained concretely.

[0123]

Fig. 20 is a flow chart showing the operation of the execution calculator control mechanism at the time of system start. Firstly, in a process 2001, the execution calculator control mechanism initializes all the calculators. Then, it confirms the processing capacity condition of each calculator and executes a process 2002

for confirming the calculation capacity of each calculator. Next, in a process 2003, the execution calculator control mechanism calls the specified contents concerning all the tasks from the data base. Concretely, they are the execution time restrictions and inter-task restrictions for reliability explained above. It executes a process 2004 for selecting a task having a start request from the beginning of system start from them. In the example shown in Fig. 19, Task1 to Task5 are initial start request tasks, though Task6 is not a one.

[0124]

By the process 2004, the confirmation of the execution time restrictions and inter-task restrictions for reliability concerning the initial start request tasks is completed. On the basis of the data concerning the restriction conditions, in a process 2005, the execution calculator control mechanism minimizes the violation amount, that is, follows the execution time restriction of each task, maximizes the CPU use efficiency, gives priority to the important tasks, prevents the inter-task restriction conditions for reliability from impairment, thereby selects tasks to be executed and calculators to which the tasks are allocated. For that purpose, it performs a calculation process for solving a combination optimization problem and decides task arrangement from a result of this. When the arrangement position of each task is decided, the execution calculator control mechanism executes a process

2006 for allotting a program module (hereinafter, referred to as task module) of those tasks in an execution form and a process 2007 for outputting a command for executing the allotted task module.

[0125]

Furthermore, the execution calculator control mechanism not only starts the initial tasks but also calculates task arrangement in the next time section beforehand and allots the task module. The next time section is a time section where a task generation request is presented next. The next time section in this embodiment is 30 deg after the time of start in synchronization with the electrical angle because, as shown in Fig. 19, the minimum task generation frequency is 30 deg.

[0126]

In a process 2008, the execution calculator control mechanism confirms the task request in the next time section from the real-time capacity of tasks specified in the execution time restriction explained previously and performs a next task arrangement optimization process 2009 for it in the same way as with the aforementioned. It compares the arrangement of the allotted initial task module with the arrangement of the next task module decided now and performs a process 2010 for allotting the next task module to the target calculator as required. By doing this, in the next time section, no task is generated due to an external cause and if a task request

occurs as scheduled, the tasks can be executed at a most suitable arrangement.

[0127]

Fig. 21 shows the state of initial task arrangement allocated by the operation at start of the execution calculator control mechanism mentioned above. In a calculator X (2101 in the drawing), Task1 (2102 in the drawing) and Task5 (2103 in the drawing) are arranged, and in a calculator Y (2104 in the drawing), Task2 (2105 in the drawing) and Task3 (2106 in the drawing) are arranged, and in a calculator Z (2107 in the drawing), Task4 (2108 in the drawing) is arranged. In this arrangement, the inter-task restrictions for reliability (exclusive restrictions of two tasks here) indicated in 1913 to 1916 shown in Fig. 19 are all satisfied. The process for all the tasks requested at the current stage is executed and the whole system is under a sound condition. For example, even if time elapses and a Task6 request occurs, the execution calculator control mechanism can plan a process of automatically allocating Task6 to the arrangement part of Task5 and returning it to Task5 after end, so that the sound state of the whole system can be kept.

[0128]

When a condition other than the plan is generated as a whole of the system, for example, when any of the calculators becomes unusable, the execution calculator control mechanism must execute re-allocation of tasks.

The processing flow thereof is shown in Fig. 22.

[0129]

Firstly, the execution calculator control mechanism, when a supposed time (30 deg in this embodiment) elapses from the last start of the execution calculator control mechanism itself, starts the process. It is mounted by a process 2201 for setting a signal in synchronization with the electrical angle as an interruption trigger.

[0130]

The execution calculator control mechanism, firstly after start, executes a process 2202 of judging whether the calculation capacity table, that is, the condition of the calculators is changed by collecting various monitoring results. When a device error is found, it starts a process of performing the fault tolerant function. This will be described later. A process when there is no device error will be explained hereunder.

[0131]

Firstly, in a process 2203, the execution calculator control mechanism reads a current task request. Any of the tasks of this system, when it is necessary to start another task as a result of the operation of itself, must present a task request to the execution calculator control mechanism. For example, the main detection task of Task1 has a function for expelling an abnormal input amount and it is structured so as to present a request of precise error monitoring at that time. Namely, Task1 sends a Task6 request to the execution calculator control

mechanism under the condition of detection of abnormal input and output. Such a request is read by the process 2203.

[0132]

Whether there is a difference between this current task request and the scheduled task in the next section in the last execution of the execution calculator control mechanism is judged by a judgment process 2204 of existence of an unscheduled task request and the process is branched.

[0133]

When there is no unscheduled task, the task module executing in this time section is allotted already in the last execution of the execution calculator control mechanism. Therefore, a current task execution command is immediately transmitted by a process 2205.

[0134]

When there is an unscheduled task, a process 2206 re-optimizes the arrangement of an executable task, that is, the task module in the tasks allotted in the calculators. When the execution time restriction of the unscheduled task is of high priority, the execution calculator control mechanism may re-arrange the tasks. Whether they are actually re-arranged depends on the inter-task restriction for reliability with the other allotted tasks.

[0135]

In either case, the execution calculator control mechanism automatically selects most suitable arrangement

in the executable tasks. Then, a process 2207 transmits an execution command to the most suitable arrangement tasks in the aforementioned executable tasks. When there is a time allowance compared with the task module allotment, a process 2208 allots a part of tasks as supplementary tasks and a process 2209 immediately transmits an execution command to the supplementary tasks.

[0136]

Even when there is an unscheduled task or even when there is no unscheduled task, processes 2210 to 2212 perform an arrangement optimization process for a task request in the next time section and execute a process of allotting the next task module to the target calculators as required. This procedure is exactly the same as that of 2008 to 2010 explained in Fig. 20.

[0137]

Next, a case that a device error is recognized by the judgment process 2202 and the fault tolerant function is performed will be explained.

[0138]

Firstly, a process 2213 searches the tasks executed in the device where an error occurs for existence of a standby system task (hereinafter called a fault tolerant task) for fulfilling the fault tolerant function. In the example shown in Fig. 19, it is specified that [Task3] TrPrMN-FT is a fault tolerant task for [Task1] TrPrMN. As mentioned above, a fault tolerant task for the tasks in execution in the device at the time of occurrence of an

error is searched from the data base.

[0139]

In this case, when there is a fault tolerant task, it is necessary to immediately switch the corresponding task from the standby processing mode to the main processing mode. This is realized by transmitting a main-slave mode switching command to the fault tolerant task searched by the process 2113 by a process 2214. By switching the main-slave mode, the function is maintained immediately.

[0140]

After switching of the main-slave mode, the tasks are re-arranged. The concept is basically the same as that of the process for generation of an unscheduled task which is already explained. In a combination of the task modules arranged already, the mechanism selects a most suitable execution module. However, since a calculator error occurs already, firstly, a process 2215 re-prepares a calculation capacity table and updates the condition of calculators confirmed by the execution calculator control mechanism. Then, the processes 2206 to 2209 explained previously select most suitable tasks executable under the current condition and transmit an execution command to them. Further, task arrangement optimization in the next time section is also executed by the processes 2210 to 2212 explained previously.

[0141]

According to the process flow shown in Fig. 22 explained above, when a calculator error occurs, the

fault tolerant function is performed immediately, and then in the next time section, that is, after a step of 30 deg, most suitable task arrangement is autonomously obtained, and the execution is continued.

[0142]

Using an example of the task arrangement shown in Fig. 21, the operation at the time of occurrence of a calculator error will be explained hereunder by referring to Fig. 23. The task specification contents (execution time restrictions and inter-task restrictions for reliability) are the same as those shown in Fig. 19 and a case that a calculator error occurs in the time section that the task arrangement is the same as that shown in Fig. 21 will be considered.

[0143]

In Fig. 23, calculators 2301 to 2303 are the same as the calculators 2101, 2104, and 2107 shown in Fig. 21 and the task arrangement is also the same. When the calculator 2301 becomes unusable, [Task1] TrPrMN (2304 shown in the drawing) and [Task5] input-output P/P' monitoring (2305 in the drawing) become inexecutable. The operation of the execution calculator control mechanism at this time will be explained hereunder according to the flow chart shown in Fig. 22.

[0144]

Firstly, the execution calculator control mechanism interrupting and starting in the process 2201 judges an error in the calculator X (2301 shown in Fig. 23) in the

process 2202. In this case, the program immediately moves to the fault tolerant start process. In the process 2213, the mechanism confirms that the fault tolerant task [Task3] TrPrMN-FT is executed by the calculator Y (2302 shown in Fig. 23) for [Task1] TrPrMN. For [Task5] input-output P/P' monitoring, there is no fault tolerant task. Thereafter, immediately in the process 2214, the mechanism instructs [Task3] TrPrMN-FT to switch the main-slave mode. As shown in Fig. 24, [Task3] TrPrMN-FT 2401 shown in the drawing provides the function of Task1 continuously instead of [Task1] TrPrMN or equivalent.

[0145]

Next, in the process 2215, after reflecting by the condition that the calculator X is not usable, the mechanism executes the process 2216 for planning a better task execution configuration, if any, by the allotted task module. Here, concretely, the mechanism searches for a combination for canceling the inter-task restriction violation for reliability (exclusive Level 2) of [Task1] TrPrMN 2401 shown in the drawing and [Task2] TrPrFD 2402 shown in the drawing and the execution time restriction violation (execution priority Level 2) of [Task3] TrPrMN-FT not executed by any calculator, though there is no solution thereof in this example. Therefore, the task arrangement configuration shown in Fig. 24 is urgent arrangement for an error in the calculator X. In the process 2207, an execution command is transmitted to the tasks 2401 to 2403. Since there is no task which can be

supplemented and executed, the processes 2208 and 2209 do not execute task allotment and execution command issue.

[0146]

Next, the mechanism plans most suitable arrangement in the next time section, that is, in the future section after lapse of an electrical angle of 30 deg and allots the task modules. Firstly, in the process 2210, the mechanism checks for a task request. In this case, on the basis of the task specification contents shown in Fig. 19, the mechanism confirms a task request of Task1 to Task4. Then, in the process 2211, the mechanism executes next task arrangement optimization. As a result, the mechanism can obtain the solution (task arrangement) shown in Fig. 24.

[0147]

When the task arrangement shown in Fig. 25 is compared with that shown in Fig. 24, the arrangement of [Task1] TrPrMN 2501 in the drawing is not changed, though the arrangement of [Task4] TrPrFD-FT 2502 shown in the drawing and [Task2] TrPrFD 2503 shown in the drawing is interchanged. Accordingly, the inter-task restriction violation for reliability (exclusive Level 2) of [Task1] TrPrMN and [Task2] TrPrFD can be canceled. However, with respect to the execution time restriction violation (execution priority Level 2) of [Task3] TrPrFD-FT, no arrangement for canceling it is found. Therefore, the task arrangement shown in Fig. 25 is obtained.

[0148]

Finally, the mechanism allots the next task module in the process 2212. Concretely, the mechanism allots the task module of [Task4] TrPrFD-FT to the calculator Y and the task module of [Task2] TrPrFD to the calculator Z. By doing this, a task module necessary to realize the task arrangement shown in Fig. 25 is prepared in the next time section.

[0149]

If there is no change in the calculator condition and task request condition in the next time section, that is, in the future section after lapse of an electrical angle of 30 deg, the execution calculator control mechanism can continue a most suitable task process only by transmitting an execution command to tasks 2501 to 2503.

[0150]

[Effects of the Invention]

By the distributed processing system of the present invention, task allocation for realizing consistency of the real-time capacity with the reliability is executed according to the constitution and capacity of calculators. Further, by executing task scheduling, the distributed processing system can always realize high reliability deliberately. Also for a process that a task start request occurs whenever necessary or also in a case of a calculator error or maintenance, function maintenance for realizing consistency of the CPU use efficiency with the reliability can obtained.

WHAT IS CLAIMED IS:

1. An information processor for processing a plurality of tasks for realizing a protective relay function by a plurality of calculators connected by a network, comprising:

means for storing combinations of tasks to be processed by a specific calculator among said plurality of tasks and

means for selecting any one of said combinations of tasks to be processed by said specific calculator and executing said tasks of said selected combination by said specific calculator.

2. An information processor according to Claim 1, wherein said plurality of tasks include normal system tasks for always performing processes included in a protective relay and standby system tasks for performing said processes included in said protective relay in an emergency case and

said combinations of said tasks which can be processed by said specific calculator do not include combinations of said normal system tasks and said standby system tasks.

3. An information processor according to Claim 1, wherein said plurality of tasks include main detection tasks for realizing a main detection function of a protective relay and trouble detection tasks for

* C L M S *

realizing a trouble detection function of a protective relay and

said combinations of said tasks which can be processed by said specific calculator do not include main detection tasks and trouble detection tasks of a same protective relay.

4. An information processor according to Claim 1, wherein said plurality of tasks include package bus protective tasks for realizing a package bus protection function of a bus protective relay and split bus protective tasks for realizing a split bus protection function of a protective relay and

said combinations of said tasks which can be processed by said specific calculator do not include package bus protective tasks for a same bus and split bus protective tasks for a same bus.

5. An information processor according to Claim 1, wherein said plurality of tasks have transmission line tasks for realizing a transmission line protective relay function of each of transmission lines composed of a parallel circuit and

said combinations of said tasks which can be processed by said specific calculator do not include a plurality of transmission line tasks for transmission line protection belonging to a same parallel circuit.

* C L M S *

6. An information processor for processing a plurality of tasks for realizing the protective relay function by a plurality of calculators connected by a network, comprising:

means for storing combinations of tasks not to be processed by a specific calculator among said plurality of tasks and

means for executing tasks of combinations excluding said combinations of said tasks not to be processed by said specific calculator by said specific calculator.

7. An information processor according to Claim 6, wherein said plurality of tasks include normal system tasks for always performing processes included in a protective relay and standby system tasks for performing said processes included in said protective relay in an emergency case and

said combinations of said tasks which are not processed by said specific calculator include combinations of said normal system tasks and said standby system tasks.

8. An information processor according to Claim 6, wherein said plurality of tasks include main detection tasks for realizing a main detection function of a protective relay and trouble detection tasks for realizing a trouble detection function of said protective relay and

* C L M S *

said combinations of said tasks which are not processed by said specific calculator include main detection tasks and trouble detection tasks of a same protective relay.

9. An information processor according to Claim 6, wherein said plurality of tasks include package bus protective tasks for realizing a package bus protection function of a bus protective relay and split bus protective tasks for realizing a split bus protection function of a protective relay and

said combinations of said tasks which are not processed by said specific calculator include package bus protective tasks for a same bus and split bus protective tasks for a same bus.

10. An information processor according to Claim 6, wherein said plurality of tasks have transmission line tasks for realizing a transmission line protective relay function of each of transmission lines composed of a parallel circuit and

said combinations of said tasks which are not processed by said specific calculator include a plurality of transmission line tasks for transmission line protection belonging to a same parallel circuit.

11. A highly reliable real-time distributed processing system for executing a plurality of tasks by a

plurality of calculators connected by one or a plurality of networks, wherein when a calculation process that to each of said plurality of tasks, a restriction or condition for completing execution of said task within a predetermined time is added is to be executed, a process of judging a degree of sufficiency of a restriction or condition for executing restrictions or mutual relations between a plurality of optional tasks or each of the same by different calculators is executed, thus within an executable range by all said calculators, maximum or sufficient reliability is obtained.

12. A highly reliable real-time distributed processing system according to Claim 11, wherein an execution time evaluation index for completion of said execution of said each of said plurality of tasks within said predetermined time is provided, and an execution calculator evaluation index for executing each of said mutual relations between said plurality of tasks by different calculators is provided, and

said system has an execution calculator control mechanism for executing a selection process of tasks to be executed and a selection process of calculators for calculating said selected tasks within said executable range by all said calculators so as to maximize said evaluation indexes.

13. A highly reliable real-time distributed

* C L M S *

processing system according to Claim 12, wherein said system has said execution calculator control mechanism for storing said execution time evaluation index for each of said tasks and said execution calculator evaluation index for each of said mutual relations between tasks other than said tasks in a data base beforehand and executing said task selection process and said calculator selection process by referring to said indexes.

14. A highly reliable real-time distributed processing system according to Claim 12, wherein said execution time evaluation index for each of said tasks and said execution calculator evaluation index for each of said mutual relations between tasks other than said tasks are set in a program module for realizing said tasks beforehand and said system has said execution calculator control mechanism for executing said task selection process and said calculator selection process by referring to said indexes transmitted from said tasks.

15. A highly reliable real-time distributed processing system according to Claim 12, wherein when a name for controlling each of said tasks is to be provided, in a symbol for indicating said task name, a symbol for describing said execution time evaluation index for said task and a symbol for describing said execution calculation evaluation index for each of said mutual relations between tasks other than said tasks are

included, thus said system has said execution calculator control mechanism for extracting said execution time evaluation index and said execution calculator evaluation index from said task name given beforehand.

16. A highly reliable real-time distributed processing system according to Claims 12 to 15, wherein a process of said execution calculator control mechanism is executed by a plurality of calculators on said network.

17. A highly reliable real-time distributed processing system according to Claims 12 to 15, wherein a process of said execution calculator control mechanism is executed by one or a plurality of calculators not connected to said network always.

18. A highly reliable real-time distributed processing system according to Claims 12 to 16, wherein said execution calculator control mechanism always monitors an operating condition of said calculators and upon detection of a change in a condition of any of said calculators on said network, re-executes said task selection process and said calculator selection process.

19. A highly reliable real-time distributed processing system according to Claim 18, wherein said execution calculator control mechanism controls existence of a task to be generated from a current calculation

* C L M S *

section in future and said evaluation indexes concerning said task, thereby obtains most suitable solutions for said task selection process and said calculator selection process via a plurality of time sections and executes task scheduling.

20. A highly reliable real-time distributed processing system according to Claim 18, wherein when a start request of a new task not included in contents controlled by said execution calculator control mechanism occurs or a start command of a new task is issued from outside, said execution calculator control mechanism re-executes said task selection process and said calculator selection process.

21. A highly reliable real-time distributed processing system according to Claim 18, wherein when a command is issued to a utilization condition of said calculators via an external calculator, said execution calculator control mechanism supposes a utilization condition of said calculators according to contents of said command, thereby re-executes said task selection process and said calculator selection process.

22. A highly reliable real-time distributed processing system according to Claim 11, wherein an execution time evaluation index for completion of said execution of said each of said plurality of tasks within

said predetermined time is provided, and an execution calculator evaluation index for executing each of said mutual relations between said plurality of tasks by different calculators is provided, and

said system has a development tool capable of being initially designed for executing a selection process of tasks to be executed and a selection process of calculators for calculating said selected tasks within said executable range by all said calculators so as to maximize said evaluation indexes.

23. A highly reliable real-time distributed processing system according to Claims 11 to 22, wherein said system, for each of a plurality of communication channels for connecting between a plurality of calculators, has a set of communication channel monitoring tasks for verifying normalcy of communication contents as said tasks and

said system executes a process of judging a degree of sufficiency of a restriction or a condition for executing each task composed of a set of communication channel monitoring tasks for verifying a same communication channel by different calculators, thereby confirms normalcy of each of said communication channels.

24. A highly reliable real-time distributed processing system according to Claims 11 to 22, wherein said system has a plurality of calculator monitoring

tasks for executing a test for checking a predetermined calculation formula as said tasks and verifying results via communication with a plurality of other calculators connected by a network and

said system executes a process of judging a degree of sufficiency of a restriction or a condition for executing each of said calculator monitoring tasks by different calculators, thereby can identify a faulty calculator or detect a fault of said network.

* C L M S *