IPO's copy!
9800464-1

APPARATUS AND METHOD FOR SCHEDULING JOBS IN A DISTRIBUTED PROCESSING SYSTEM

Portions of this patent application contain materials that are subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records but otherwise reserves all copyright rights whatsoever.

FIELD OF THE INVENTION

The present invention relates generally to distributed processing systems and, more particularly, to a system and method for scheduling jobs in a distributed processing system.

BACKGROUND OF THE INVENTION

A job scheduler is used to schedule jobs for service on a CPU of a computer system. A job scheduler could also be used to schedule other resources within the computer system, such as access to secondary storage devices, communication links to servers and other computer systems, etc. A job scheduler might also be used to schedule critical stations of an assembly line within a factory, the jobs in this case being work required on assemblies in progress. As used herein, the term "job" should therefore be understood to broadly encompass any form of work to be performed. The term "job" is used here

because that is the terminology commonly used in the context of computer operating systems. However, in other environments, it may be referred to as a task, procedure, assembly, work order, etc. The term "job" when used in the specification and claims encompasses all of these terms.

Examples of job schedulers can be found in U.S. Patents 5,414,845 and 5,442,730 of common assignee herewith.

Current job schedulers lack the ability to schedule a job at any periodicity, skip the job scheduled to be processed by a CPU or to schedule jobs to take place on remote machines in a network.

The Windows NT Operating System does permit scheduling jobs on remote machines, but such scheduling is very restrictive. For example, such scheduling can only be accomplished by command line entries, i.e., graphical user interface scheduling is not used. Also, both the source and target machines must be utilizing the Windows NT Operating System before a job originating from the source machine can be scheduled on the target machine. Finally, Windows NT has very limited periodicity choices.

Current job schedulers also lack the ability to exclude days from the job schedule and do not permit scheduling of or by diverse job types, such as DOS applications, Win32/Win95 applications, OS/2 applications, etc. Current job schedulers

are not integrated with network user/groups or access control of operating system.

SUMMARY OF THE INVENTION

Accordingly, the present invention is directed to a distributed processing system comprising a plurality of processors, a communications network for interconnecting the processors, and a job queue coupled to the network for storing jobs to be processed by the system. The distributed processing system of the present invention further includes a job scheduler coupled to the network for scheduling jobs in the job queue for processing by the system wherein the scheduler schedules a job to be processed by the system repeatedly at a selected periodicity.

The present invention also includes a method for scheduling jobs for processing in a distributed processing system including a plurality of processors interconnected by a communications network. The method comprises the steps of storing jobs to be processed by the system in a job queue; scheduling jobs in the job queue for processing by the system; and scheduling a job to be processed by the system repeatedly at a selected periodicity.

The present invention also includes a method for scheduling jobs comprising the steps of providing a local workstation having a first operating system, providing a target workstation having a second operating system, interconnecting the local and target workstations by a communications network, and providing a software program for

scheduling jobs to integrating the software program with the first and second operating systems such that the local workstation can schedule a job for processing in the target workstation.

The primary advantage of the present invention is scheduling a job for processing in a distributed processing system at any periodicity.

Another advantage of the present invention is skipping a job scheduled to be processed by the system.

Another advantage of the present invention is a job scheduler which schedule jobs to take place on remote machines in the distributed processing system.

Another advantage of the present invention is providing a job scheduler which integrates two or more different operating systems so that jobs may be scheduled between workstations having different operating systems.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1A illustrates a typical distributed processing system for which the present invention is designed;

Figure 1B illustrates an information processing system for implementing the present invention;

Figure 2 illustrates a block diagram of an embodiment of the present invention;

Figure 3 illustrates a graphical user interface for implementing the present invention; and

Figure 4 illustrates another graphical user interface for implementing the present invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT OF THE INVENTION

The present invention is practiced in a distributed processing computer environment. This environment consists of a number of information handling systems linked together by a communications network. Each information handling system includes one or more processors.

Referring to Figure 1A, there is illustrated a distributed processing system 100 according to the present invention. Distributed processing system 100 includes information handling systems 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, and 128 connected via communications network 106. Network 106 can be any type of network including LAN, WAN, ATM, or other. Physical network protocols such as Ethernet or Token Ring can be used, and communications protocols such as TCP/IP or Netbios or Novell Netware can control the network. Network file system management can be provided by a network operating system such as a program based on the Sun Microsystems NFS network operating systems technology or CMU AFS technology. Other network file systems may include Open Group's DFS, SMB (Server Message Block), CIFS (Common Internet File System) or HTTP (Hyper Text Transfer Protocol). Each of these file system programs allows distributed processors to access and manage data residing on remote systems. These systems create a single logical file

system for each processor regardless of the physical location of the individual files.

Referring next to Figure 1B, there is illustrated a representative hardware environment which illustrates a typical hardware configuration of an information handling system 108 (see Figure 1A) in accordance with the present invention having one or more central processing units 10 such as a conventional microprocessor and a number of other units interconnected via system bus 12. System 108, shown in Figure 1B, includes random access memory (RAM) 14, read-only memory (ROM) 16, I/O adapter 18 for connecting peripheral devices such as disk units 20 and tape drives 40 to bus 12, user interface adapter 22 for connecting keyboard 24, mouse 26 having right mouse button 17 and left mouse button 19, speaker 28, microphone 32 and/or other interface devices such as touch screen device (not shown) to bus 12, communications adapter 34 for connecting system 108 to network 106 and display adapter 36 for connecting bus 12 to monitor or display device 38. It is to be noted that information handling systems 110, 112, 114, 116, 118, 120, 122, 124, 126, and 128 have a similar hardware configuration of that of information handling system 108 just described.

Referring now to Figure 2, a block diagram of the present invention is illustrated. Reference numerals in Figure 2 which are identical to the reference numerals used in Figures 1A and 1B represent identical parts. Figure 2 shows a portion

of distributed processing system 100 illustrated in Figure 1A including information handling systems 110, 116, and 122 interconnected via communications network 106. In this embodiment, information handling system 122 includes a job queue for storing jobs to be processed by distributed processing system 100. Job queue 130 is coupled to network 106 via information handling system 122. Information handling system 122 also includes a job scheduler 132 for scheduling jobs in job queue 130 for processing by distributed processing system 100. Job scheduler 132 schedules a job to be processed by distributed processing system 100 repeatedly at a selected periodicity which will be described in further detail in connection with Figures 3 and 4. As will also be described in connection with Figures 3 and 4, job scheduler 132 has the ability to skip a job scheduled to be processed by distributed processing system 100. Job scheduler 132 may be used to schedule jobs to be processed by information handling system 122 or any of the other information handling systems 108, 110, 112, 114, 116, 118, 120, 124, 126, or 128. It will be appreciated that the present invention may include one or more job schedulers and one or more job queues. It will also be appreciated that the present invention may include a job scheduler and a job queue in each of the information handling systems or any combination of information handling systems. In this manner, a job scheduler within an information handling system may be used to schedule a job in a remote information handling system. For example, job scheduler 132 in information handling system 122 may be used to schedule a job

from job queue 130 to be processed by information handling system 110. Job scheduler 132 is preferably implemented by a software program stored in memory in information handling systems 108, 110, 112, 114, 116, 118, 120, 124, 126, or 128. According to the invention, information handling systems 108, 110, 112, 114, 116, 118, 120, 124, 126, or 128 preferably have different or heterogeneous operating systems. The present invention provides for integration of job scheduler 132 with the various operating systems and network operating system so that any job originated in one information handling system can be scheduled in and processed by another information handling system.

Referring now to Figures 3 and 4, a display screen and graphical user interface for implementing the present invention is shown displayed on monitor 38. In this manner, a graphical user interface including a monitor 38 may be used to display information related to a job stored in job queue 130, and a user interface device such as mouse 26 may be used to control the information displayed on monitor 38 which, in turn, is interfaced with job queue 130 and job scheduler 132 to thereby control job scheduling.

In the following description of the operational features of and monitor displays in a graphical user interface in accordance with the present invention, the term "workstation" refers to information handling systems 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, and 128. The term "local

workstation" refers to an information handling system having a first operating system and including a job queue and a job scheduler. The term "target workstation" refers to an information handling system having a second operating system for processing jobs assigned thereto by a local workstation. All "icons" are viewed on monitor 38 and pointed to by mouse 26. Left mouse button 19 is pressed twice, or "double-clicked," to "open" an icon to display other viewing windows, or right mouse button 17 may be used to display a context menu or implement a drag-dropped scheduler on an object, as is well known in the art.

First, a Net Sched folder icon is double-clicked using mouse 26 and left mouse button 19, to open two icons, a local workstation icon, and a folder of domains icon. When the local workstation icon is double-clicked, an icon view opens that contains scheduled job icons as well as a job template icon. This may be a filtered view and the title bar of the view indicates when the view is filtered. The filtering criteria may be based on user, integrated access control of the person viewing the data, or job type. When a scheduled job icon is double-clicked, a Job-Settings notebook as shown in Figure 3 opens that will show all of the entry fields containing the current values for a selected job. When the job template icon is double-clicked, "dragged" or initiated in other ways, such as with right mouse button 17 "context menu" activated, a Job Template-Settings notebook opens to show all of the entry fields (that have defaults) containing standard

defaults. Non-defaulted entry fields are left empty (also shown in Figure 3).

When the local workstation icon settings option is selected from a right-mouse-button-pop-up menu or context menu, a Workstation-Settings notebook, shown in Figure 4, opens to the current view filter settings for the local workstation.

When the folder of domains icon is double-clicked, an icon view opens that contains folders for each of the available target domains. When a domain folder is double-clicked, an icon view opens that contains the target workstation icons available in the selected target domain. When a target workstation icon is double-clicked, an icon view opens that contains scheduled job icons as well as a job template icon. This is a potentially filtered view and the title bar of the view indicates when the view is filtered. When a scheduled job icon is double-clicked, a Job-Settings notebook opens that shows all of the entry fields containing the current values for the selected job. Here, executable objects or tabs may be inherited from the local operating system. For example, a tab on job object for "DOS settings" or "Associations" could be available, since this object inherits fields from the workplace shell object. When the job template is double-clicked, a Job Template-Settings notebook opens that shows all of the entry fields (that have defaults) containing the standard defaults. Non-defaulted entry fields

will be empty. When a target workstation icon settings option is selected from the right-mouse-button-pop-up menu, the Workstation-Settings notebook opens showing all of the current view filter settings for the selected target workstation.

From the desktop (or via a system Drives Icon), a user also has an option to schedule an executable file from the right-mouse-button-pop-up menu.

What follows is a description of the fields displayed in Figure 3.

Job Name field 200

The Job Name field contains the name of the selected job for an existing scheduled job. This field is blank for a new job on the Job Template-Settings notebook.

Job ID field (read-only 1) 202

This field will contain a job ID for an existing scheduled job. This field will be blank for a new job on the Job Template-Settings notebook. This field cannot be updated by the user.

Command field 204

The Command field contains the command text (.exe file and parameters) of the selected job for an existing scheduled job. This field is blank for a new job on the Job Template-Settings notebobk.

read-only 2 206

This field contains text similar to the following for an existing scheduled job: Last submitted by USERID at MACHINEX with a start date of June 1, 1996. This field is left blank for a new job on the Job Template-Settings notebook. This field cannot be updated by the user.

"How often to run the job" section 208

For an existing scheduled job, if the selected job is recurring, the "Every" button 210 is selected and the entry fields in this section will have the current values of the selected job. If the selected job is non-recurring (will run only once), the "Job will only run once" button 212 will be selected and the "Every" entry fields in this section will be grayed-out (disabled) and they will display the new job defaults.

For a new job on the Job Template-Settings notebook, the default for this section is for the "Every" button to be selected.

The "Every" button 210 and "Job will only run once" button 212 are mutually exclusive and when the "Job will only be run once" button 212 is selected, the "Every" entry fields are disabled. If the "Every" button 210 is selected, then the spin 1 and drop 1 fields are required. If the "Every" button 210 is selected, and drop 2 is enabled, then drop 2 is required.

a. spin 1 214
This spin button allows a user to select a range of 1-999. The default for this entry for a new job on the Job Template-Settings notebook is 7.

b. drop 1 216
This drop-down control drops down to the following list:
-minutes
-hours
-days
-months
The default for this entry for a new job on the Job Template-Settings notebook is days.

c. drop 2 218

drop 2 is disabled unless spin 1 specifies 7 and drop 1 specifies days, or unless drop 1 specifies months. When spin 1 specifies 7 and drop 1 specifies days, the drop 2 drop-down control will drop down to the following list:

-Mondays
-Tuesdays
-Wednesdays
-Thursdays
-Fridays
-Saturdays
-Sundays

The default for this entry for a new job on the Job Template-Settings notebook is the current weekday. When the drop 1 specifies months, the drop 2 drop-down control drops down to the following list:

-the start day specified below
-the third to the last day of the month
-the second to the last day of the month
-the last day of the month

The default for this entry for a new job on the Job Template-Settings notebook is "the start day specified below"

By using fields 214, 216, 218, any job can be scheduled to run at non-standard intervals such as every 36 hours and 15 minutes or every 34 days.

"When to start the job" section 220

For an existing scheduled job, if the selected job has a start date that is previous to the current date, then the "immediately" button 220 is selected and the entry fields for the following button will be disabled (the entry fields will contain the current date). If the selected job has a start date that is after the current date, then the button 224 in front of the entry fields is selected and the entry fields will be enabled and will contain the start date data for the selected job.

For a new job on the Job Template-Settings notebook, the default for this section is for the "Immediately" button 222 to be selected.

The "Immediately" button 222 and the start-date (lower) button 224 are mutually exclusive and when the "Immediately" button is selected, the start-date entry fields are disabled. If the "Immediately" button 222 is not selected, then the start-date entry fields are required.

a. Year field (spin 2) 226
This spin button allows a user to spin through a range of 1970 - logical maximum.

The default for this entry for a new job on the Job Template-Settings notebook is the current year.

b. Month field (spin 3) 228
This spin button allows a user to spin through a range of January-December. The default for this entry for a new job on the Job Template-Settings notebook is the current month.

c. Day field (spin 4) 230
This spin button allows a user to spin through a range of 1-31. The default for this entry for a new job on the Job Template-Settings notebook is the current day of the month.

If a user has specified in the "How often to run the job" section 208 that a weekly recurring (every 7 days) job run on a particular weekday (Mondays-Sundays), then the values originally specified in the Year, Month, and Day entry fields will be automatically adjusted (if necessary) to the date of the next occurrence of the specified weekday after the start date specified. The weekday specified for weekly recurring jobs takes precedence over the start date specified.

d. Hour field (drop 3) 232
This drop-down control drops down to the following list for programming languages that use a 12-hour clock:
-midnight
-1
-2
...
-12

For languages that use a 24-hour clock:
-0
-1
...
-23

The default for this entry for a new job on the Job Template-Settings notebook is the current hour.

e. Minute field (spin 5) 234
This spin button allows a user to spin through a range of 0-59. The default for this entry for a new job on the Job Template-Settings notebook is the current minute of the hour plus 10 minutes.

f. PM field (ck1) 236
This check box allows a user to indicate whether or not the specified time is AM (not checked) or PM

(checked). The default for this control is enabled if the current time is in the PM or disabled if the current time is in the AM. This control does not appear for languages that use a 24-hour clock.

"When to delete the job" section 238

For an existing scheduled job, if the selected job is a non-recurrent job ("Job will only run once" button 212 in the "How often to run the job" section 208 is selected), then this section will be disabled. If the selected job is a recurrent job and no deletion data was specified, then the "No deletion is scheduled" button 240 is selected (the deletion date entry fields will be disabled, but will contain the new job defaults of current month, day and current year +1). If the selected job is a recurrent job and a deletion date was specified, then the "Deletion date" button 242 is selected and the corresponding entry fields will be enabled and will contain the deletion date data for the selected job.

For a new job on the Job Template-Settings notebook, the default for this section will be for the "No deletion is scheduled" button 240 to be selected.

The "No deletion is scheduled button" 240 and the "Deletion date" button 242 are mutually exclusive and when the "No deletion is scheduled" button 240 is selected, the "Deletion date" entry fields are disabled. If the "No

deletion is scheduled" button 240 is not selected, then the "Deletion date" entry fields are required.

a. Year field 244 - Same as Year field (spin 2) already described, except the default for this entry for a new job on the Job Template-Settings notebook is the next year (current year +1).

b. Month field 246 - Same as Month field (spin 3 already described.

c. Day field 248 - Same as Day field (spin 4) already described.

Workstation-Setting Notebook

What follows is a description of the fields displayed in Figure 4. This may be a filtered view, and the title bar of the view indicates when the view is filtered. The filtering criteria may be based on user, integrated access control of the person viewing the data, or job type.

Job Name field 300

This entry field allows a user to display only jobs with the specified name text in the icon view. This entry field is blank by default.

Command field 301

This entry field allows a user to display only jobs with the specified command text in the icon view. This entry field is blank by default, e.g., filtering can also be based on command type, such as DOS, Win16, Win32, OS/2 16-bit, OS/2 32-bit, etc.

Submitter 302

This entry field allows a user to display only jobs submitted by the specified user ID in the icon view. This entry field is prefilled with the current logon user ID with that text selected for deletion if the user types some other user ID.

"Submitted from" field 304

This entry field allows a user to display only jobs submitted from a specified workstation in the icon view. This entry field is prefilled with the current workstation name with that text selected for deletion if the user types some other workstation name.

"How often the job is run" section 306

All of the checkbox controls in this section are mutually exclusive with each other.

a. "Job runs more than once" (ck1) checkbox control 308
This allows a user to display all jobs that will run more than once. This control will be unchecked (job to run only once) by default.

b. "every" (ck2) checkbox control 310
This allows a user to display all jobs that will run at a specified frequency. This control will be unchecked by default. If this control is checked, then the following corresponding fields are required:

i. drop 1 312
This drop-down control is prefilled with a default of "Exactly". It will drop down to the following list:
-Exactly
-More than
-Less than

ii. spin 1 314
This spin button allows a user to spin through a range of 1-999. This control is prefilled with a default of 7.

iii. drop 2 316
This drop-down control is prefilled with a default of "days". It will drop down to the following list:
-minutes
-hours
-days
-months

c. "Job runs only once" (ck3) checkbox control 318
This allows a user to display all jobs that will run only once. This control will be unchecked by default.

"When the job will run" section 320

a. "Filter enabled" (ck7) checkbox control 322
If this control is checked, all of the other entry fields in the section are enabled and required. If this control is not checked, all of the other entry fields in this section are disabled. This control will be unchecked by default.

b. drop 3 324
This drop-down control is prefilled with a default of "After". It will drop down to the following list:
-After

-On
-Before

c. Year field (spin 2) 326
This spin button will allow a user to spin through a range of 1970 -logical maximum.

The prefilled default for this entry is the current year.

d. Month field (spin 3) 328
This spin button allows a user to spin through a range of January- December. The prefilled default for this entry is the current month.

e. Day field (spin 4) 330
This spin button allows a user to spin through a range of 1-31. The prefilled default for this entry is the current day of the month.

"Time of day the job will run" section 332

a. Filter Enabled field (ck7) checkbox control 334 - Same as a. in the "when the job will run" section.

b. drop 3 336
This drop-down control is prefilled with a default of "After". It will drop down to the following list:

-After
-At
-Before

c. Hour field (drop 5) 338
This drop-down control will drop down to the following list for programming languages that use a 12-hour clock:
-midnight
-1
-2
...

For languages that use a 24-hour clock:
-0
-1
...
-23

This entry field is prefilled with the current hour as a default.

d. Minute field (spin 5) 340
This spin button allows a user to spin through a range of 0-59. This entry field is prefilled with the current minute as a default.

e. PM field (ck4) 342

This check box allows a user to indicate whether or not the specified time is AM (not checked) or PM (checked). The default for this control is enabled if the current time is in the PM or disabled if the current time is in the AM. This control does not appear for languages that use a 24-hour clock.

"When the job will be deleted" section 344

This entire section should be disabled if the "Job runs only once" (ck3) checkbox control 318 is checked in the "How often the job is run" section 306. The two checkbox controls 346, 348 (ck5 and ck6) are mutually exclusive with each other.

a. No deletion date specified (ck5) checkbox control 346

If this control is checked, recurring jobs with no deletion date specified will be displayed. This control is unchecked by default.

b. ck6 348

If this control is checked, recurring jobs with the specified deletion date will be displayed. If this control is checked, then all of the corresponding entry fields are required. This control is unchecked by default.

i. drop 3 350 - Same as b. in "when the job will run" section.

ii. Year field (spin 2) 352 - Same as c. in "when the job will run" section.

iii. Month field (spin 3) 354 - Same as d. in "when the job will run" section.

iv. Day field (spin 4) 356 - Same as e. in "when the job will run" section.

If a required entry field on either of the panels shown in Figures 3 or 4 is left blank, an error message is displayed to the user to indicate which required fields still need to be filled in. The panel is left open and the cursor is focused on the required entry field that was erroneously left blank.

Although the present invention and its advantages have been described in detail, it should be understood that various changes, substitutions and alterations can be made herein without departing from the spirit or scope of the invention as defined in the appended claims.

WHAT IS CLAIMED IS:

1. A distributed processing system comprising:

a plurality of processors;

a communications network for interconnecting said processors;

a job queue coupled to said network for storing jobs to be processed by said system;

a job scheduler coupled to said network for scheduling jobs in said job queue for processing by said system; and

wherein said scheduler includes means for scheduling a job to be processed by said system repeatedly at a selected periodicity.

2. The distributed processing system, as recited in claim 1, wherein said scheduler includes means for skipping a job scheduled to be processed by said system.

3. The distributed processing system, as recited in claim 1, wherein said scheduler includes means for scheduling a job to be processed by any one of said processors.

4. The distributed processing system, as recited in claim 1, further including a monitor coupled to said network for displaying information related to a job stored in said job queue.

* C L M S *

5. The distributed processing system, as recited in claim 4, further including a user interface device coupled to said network for controlling the information displayed on said monitor.

6. A method for scheduling jobs for processing in a distributed processing system including a plurality of processors interconnected by a communications network, said method comprising the steps of:

storing jobs to be processed by the system in a job queue;

scheduling jobs in the job queue for processing by the system; and

scheduling a job to be processed by the system repeatedly at a selected periodicity.

7. The method, as recited in claim 6, further including the step of skipping a job scheduled to be processed by the system.

8. The method, as recited in claim 6, further including the step of scheduling a job to be processed by any one of the processors.

9. The method, as recited in claim 6, further including the step of displaying information related to a job stored in the job queue.

10. The method, as recited in claim 9, further including the step of controlling the information displayed.

11. A method for scheduling jobs, said method comprising the steps of:

providing a local workstation having a first operating system;

providing a target workstation having a second operating system;

interconnecting said local and target workstations by a communications network; and

providing a software program for scheduling jobs to integrating said software program with said first and second operating systems such that said local workstation can schedule a job for processing in said target workstation.

12. The method, as recited in claim 11, wherein said first and second operating systems are heterogeneous.

13. The method, as recited in claim 11, wherein said local workstation includes a first processor and said target workstation includes a second processor, said first and second processors being heterogeneous.

14. The method, as recited in claim 11, further comprising the steps of providing a network operating system for said network and integrating said software program for scheduling jobs with said network operating system such that said local workstation can schedule a job for processing in said target workstation.

15. The method, as recited in claim 11, further comprising the steps of defining user information and storing said user information.

16. The method, as recited in claim 11, further comprising the step of providing access control to said software program for scheduling jobs.