Title: DEVICE AND METHOD FOR ALLOCATING JOBS IN A NETWORK

Abstract: A device and method for allocating jobs to host computers in a network are disclosed. The device and method groups the jobs into groups of jobs according to predetermined grouping criteria. One of the predetermined grouping criteria is whether or not the duration of the job is less than a predetermined duration. Jobs having similar characteristics are grouped together. One of the characteristics is the resources required by the job to be executed. The group of jobs is then matched and sent to one of the host computers. To facilitate matching, a representative job within each group of jobs is selected and matched to a host computer. The entire group of jobs is then sent to the matched host computer together. The jobs can be sent either sequentially, if the jobs require a resource of the matched host computer that cannot be shared, or in parallel, if the jobs can share the resources of the host computer. Once the matched host computer has completed running each job in the group of jobs, the finished group of jobs are returned from the matched host computer as a group, which further decreases scheduling overhead.
Without international search report and to be republished upon receipt of that report.

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.
DEVICE AND METHOD FOR ALLOCATING JOBS IN A NETWORK

FIELD OF THE INVENTION

This invention relates to networks of host computers, with each host computer having resources to process jobs. More particularly, the present invention relates to devices and methods that more efficiently allocate jobs to the host computers in a network.

BACKGROUND OF THE INVENTION

In the past, networks comprising at least two, and generally several, host computer systems have been used to process individual jobs. Generally, each host computer in a network has resources that can be used to perform the jobs. Furthermore, the resources of each host computer are transient and may vary over time depending on the availability, and, the general status of the host computer.

The term "jobs" generally refers to computer tasks that require various resources of a computer system to be processed. The resources a job may require include computational resources of the host computer system, database retrieval/storage resources, output resources and the availability of specific processing capabilities, such as software licenses or network bandwidth.
Jobs can generally be classified as (i) long running jobs, requiring relatively greater resources of the host computer system and running for a relatively long duration, and, (ii) short jobs, requiring relatively fewer resources and running for a relatively shorter duration. Short jobs and long jobs will both be processed by the same host computers.

It is known in the art to utilize devices that allocate jobs to corresponding hosts. However, in general, these prior art devices treat all jobs in the same way when allocating them to the host computers. In other words, the prior art devices evaluate the availability of host computers and the requirements of the jobs on an individual basis for each individual job. The prior art devices then map each individual job to the host that best suits the requirements of the job.

However, the prior art devices suffer from the disadvantage that by processing all of the jobs in the same way, short jobs tend to be allocated inefficiently. This inefficiency can result from a number of factors. For example, because there tends to be a large number of short jobs, a great deal of time and resources are spent allocating each individual short job to a corresponding host computer. In addition, the resources of the network may be underutilized because a particular host computer may complete a short job in a short
duration of time and remain idle until another job is allocated to it. It is clear that during this idle time, the host computer is being "starved" for jobs, which results in an overall decrease in the efficiency of the network, and, an imbalance in the utilization of the host computer.

Accordingly, there is a need in the art for a device and method to more efficiently allocate jobs to host computer systems in a network. In addition, there is a need in the art for a device and method to more efficiently allocate short jobs to host computers, thereby using fewer resources to allocate a large number of small jobs and reducing scheduling overhead. Furthermore, there is a need in the art for a device and a method that avoids host computer starvation, such as occurs while a host computer is idling between the time a host computer has completed one short job and the time another job is allocated to the host computer.

SUMMARY OF THE INVENTION

Accordingly, it is an object of this invention to at least partially overcome some of the disadvantages of the prior art. Also, it is an object of this invention to provide an improved device and method for allocating jobs to host computers in a network.
Accordingly, in one aspect, the present invention provides, in a network comprising at least two host computers, each host computer having resources for processing jobs, a device for allocating jobs to the host computers comprising: a grouping unit for grouping at least some of the jobs into groups of jobs according to predetermined grouping criteria; a mapping unit for mapping jobs to the host computers, said mapping unit mapping each group of jobs to one of the host computers.

In another aspect, the present invention provides, in a network comprising at least two host computers, each host computer having resources for processing jobs, a method for allocating jobs to the host computers comprising the steps of: (a) grouping at least some of the jobs into groups of jobs according to predetermined grouping criteria; and (b) mapping each group of jobs to one of the host computers.

One advantage of the present invention is that by grouping the jobs into groups of jobs prior to mapping the jobs, the groups of jobs can be mapped together. In this way, fewer resources are utilized mapping individual jobs. This reduces scheduling overhead and may decrease host computer idle time by decreasing the overall time required to allocate the jobs. Likewise, when execution of the group of jobs has been completed, the completed jobs can be
returned as a group, thereby decreasing scheduling overhead associated with the finished jobs.

A further advantage of the present invention is that the requirements of the jobs may be identified while the jobs are being grouped. In this way, jobs may be grouped together such that jobs having similar requirements will be grouped into the same group. This provides the result that a group of jobs can be matched to host computers having resources that best correspond to the requirements of the group of jobs, thereby efficiently allocating the groups of jobs.

A still further advantage of the present invention is that because the jobs having similar requirements are grouped together, only one job within a group need be matched to a corresponding host. In other words, if jobs requiring similar resources have been grouped together, a representative job within the group can be selected to represent the requirements of each of the jobs in the group. Mapping the single representative job within the group is sufficient to map the entire group. Accordingly, if a group of jobs contains ten or more jobs, the time and resources required to map that group of jobs could be about one tenth of the time and resources required to map each individual job within the group of jobs.
A still further advantage of the present invention is that the host computer to which the group of jobs is allocated will require additional time to process a group of jobs as opposed to a single job. This results in a decrease in the idle time between the time a host computer finishes processing a job or a group of jobs and the time when another job or group of jobs is allocated to the host computer. This also decreases host computer starvation and balances host computer utilization. Accordingly, by decreasing idle time, there is a corresponding increase in the time that the host computers are processing jobs, thereby increasing the efficiency of the overall network.

By increasing the efficiency of the overall network, a smaller number of computers may be needed in the network to process the same number of jobs. A decrease in the number of host computers the network must have in order to process the same number of jobs, may result in an overall cost savings.

In one embodiment, the device and method comprises a running unit which controls running or execution of the jobs. The running unit can run the group of jobs sequentially or in parallel based on the required resources of the jobs. For example, if the jobs in a group require a resource which can only be used by one job at a time, the job in the group can be run sequentially. However, if the job in the group can share the same resources, the job in the group can be run in parallel. The running
unit may be an independent unit, or, the running unit may form part of another unit, such as the dispatching unit.

Further aspects of the invention will become apparent upon reading the following detailed description and drawings which illustrate the invention and preferred embodiments of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings, which illustrate embodiments of the invention:

Figure 1 is a schematic diagram of a system incorporating a device according to one embodiment of the present invention;

Figure 2 is flowchart illustrating the method for allocating jobs according to one embodiment of the invention;

Figures 3A, 3B and 3C illustrate the grouping of jobs into a group and the mapping of the group to a host computer according to one embodiment of the present invention; and

Figure 4 represents a pending job list according to one embodiment of the present invention.
DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

5 Figure 1 shows a network comprising at least two, and preferably a plurality, of host computers H1, H2, H3...HN. For convenience, the plurality of host computers shall be referred to by reference numeral 2, while each individual host computer will be referred to as H1, H2, H3 to HN. The number of host computers H1, H2, H3 ...HN will be determined by the designer of the network, but the number will be at least two and often between 50 and 1000.

15 Each one of the plurality of host computers 2 will have certain resources for processing jobs. The plurality of host computers 2 may be homogenous, meaning that each of the plurality of host computers 2 has the same resources, or, heterogeneous, meaning that at least one of the plurality of host computers 2 has different resources. Furthermore, in a preferred embodiment, the invention will periodically assess the resources of each one of the plurality of host computers 2 to determine how the resources change over time depending on the loads applied to them and the status of each of the plurality of host computers 2. Therefore, even if the plurality of host computers 2 is homogenous in that they have similar resources, their status may be transient and vary with time based on the availability of the resources.
The network 8 further comprises a queue 4 which receives jobs, shown generally by reference numeral 6, but referred to individually as J1, J2, J3 … JN. The queue 4 may have 100 to 100,000 individual jobs 6 at any one time.

The queue 4 receives the jobs 6 from users. It is understood that the users may be any type of users with jobs 6 that must be performed by some of the plurality of host computers 2. Moreover, it is understood that the users need not be located proximate the network 8, but can be located remotely therefrom. Moreover, it is understood that the user may send the jobs 6 to the network 8 through other networks. The other networks may include intranets and internets, including the World Wide Web.

Figure 1 also shows a device, identified generally by reference numeral 10, for allocating the jobs 6 to the plurality of host computers 2 within the network 8. The device 10 receives jobs 6 from the queue 4 and allocates the jobs 6 to the plurality of host computers 2.

The jobs 6 may be any type of jobs 6 that can be performed by the host computers 2. In general, each job will have a header portion that uniquely identifies the job 6. The job 6 will have characteristics such as an indication of the user who submitted the job 6, an indication of the queue 4 from which the job 6 emanated, and the commands
and data that must be processed. A further characteristic of the job 6 is the resource requirements required to process the job 6.

5 In addition, the jobs 6 may have different running times. For example, the jobs 6 may be classified as (i) long running jobs that require relatively greater resources of the host computer system and run for a relatively long duration and

10 (ii) short jobs that require relatively fewer resources of the host computer system and run for a relatively shorter duration. Whether a job 6 will be classified as a long job or a short job will depend on several factors, such as the average type and duration of the job 6 which the network 8 generally receives from users. Another factor will be the capability of the plurality of host computers 2 to execute the jobs 6. For example, a job may be classified as a short job if its estimated duration is less than a predetermined duration, such as one minute, two minutes, or another predetermined duration depending on the network environment. The predetermined duration may vary between different networks and environments and will depend on a number of factors, including the resources of the network, and, the type of jobs generally being processed on the network. In addition, the predetermined duration may also vary over time so that the predetermined duration may be higher at different times of the day or when there are higher overall load requirements on the network.
The device 10 comprises a grouping unit 12 that receives the jobs 6 from the queue 4. The grouping unit 12 is designed to group at least some of the jobs 6 into groups of jobs 60. The grouping unit 12 groups the jobs 6 into groups of jobs 60 according to predetermined grouping criteria.

The predetermined grouping criteria can be any type of criteria which can increase the efficiency of the network 8. In particular, the predetermined grouping criteria can include the type of job 6, the resources of the host computer 6 required to process the job 6, the user who submitted the job 6 and other factors.

In a preferred embodiment, the predetermined grouping criteria comprises a predetermined duration of time required for the job 6 to be processed. In other words, one of the predetermined grouping criteria preferably comprises a predetermined duration to process the job such that jobs 6 will be grouped into a group of jobs 60 if the estimated duration to process a job 6 is less than the predetermined duration. In this way, the grouping unit 12 will group together short jobs 6 that require a relatively shorter duration of time to be processed, and, the grouping unit 12 will not group together long running jobs 6 requiring a relatively longer period of time to be processed. The duration of the predetermined duration can be any amount
selected by the system administrator. In one embodiment, the predetermined duration is between one minute and two minutes. The predetermined duration may also vary over time and may be a function of the number of jobs 6 and the availability of the plurality of the host computers 2.

The device 10 further comprises a mapping unit 14. The mapping unit 14 maps the jobs 6 and the group of jobs 60 to the plurality of host computers 2. For jobs 6 which have been grouped by the grouping unit 12 into groups of jobs 60, the mapping unit 14 maps the group of jobs 60 to one of the plurality of host computers 2. For jobs 6 which have not been grouped by the grouping unit 12, the mapping unit 14 will map the jobs 6 individually. This is illustrated in Figure 1, for example, by one of the groups of jobs, shown by reference numeral 60 and comprising jobs J2 and J4, being sent to host computer H1 and an individual job 6, in this example identified as job J3, being sent to host computer H2 for processing.

Figure 1 shows a group path 17 for the groups of jobs 6 to be sent to the mapping unit 14 and a job path 15 for the individual jobs 6 to be sent to the mapping unit 14. It is understood that the individual jobs 6 and the groups of jobs 60 need not travel separate paths 15, 17 to the mapping unit 14, but rather could be otherwise identified, such as by
different headers. For example, Figure 4 shows a symbolic representation of a pending job list 40 with the job identification or header for each of the jobs 6 in the pending job list 40. The job headers 42 for individual jobs 6 are numbered by positive numbers, namely 1, 3, 5, 6 and 10. The group of jobs 60 is represented by a group of jobs header 46. As also illustrated in Figure 4, the group of jobs 60 identified by the group of jobs header 46 may be a dummy job and may have a negative value which in this embodiment is the value -1. The group of jobs header 46 may represent a group of jobs 60, including several grouped jobs 44. For example, the first group of jobs 60 comprises grouped jobs 2 and 4 and the second group of jobs 60 includes grouped jobs 7, 8 and 9. Accordingly, as illustrated in Figure 4, the group of jobs header 46 identifies a group of jobs 60 in the pending job list 40, and, may also identify the individual grouped jobs 44 in the group of jobs 60. Thus, it is not necessary for the jobs 6 and group of jobs 60 to have two separate paths 15, 17, but rather the groups of jobs 60 could otherwise be distinguished from the jobs 6, such as headers 42, 46 as illustrated in Figure 4. In this way, use of headers 42, 46, as illustrated in Figure 4, would permit the pending job list 40 to emanate in this fashion along a single path from the grouping unit 12 to the mapping unit 14.
In order to map the jobs 6 and the groups of jobs 60 to one of the plurality of host computers 2, the mapping unit 14 comprises a scheduling unit 20 and a dispatching unit 22 as shown in Figure 1. The scheduling unit 20 matches each group of jobs 60 to one of the plurality of host computers 2. Preferably, the scheduling unit 20 will determine the characteristics, such as the resources required by the jobs 6 in each group of jobs 60, and match the group of jobs 60 with one of the plurality of host computers 2 with resources which correspond to the requirements of the group of jobs 60. More preferably, the scheduling unit 20 periodically assesses the resources of each of the plurality of host computers 2 to match each group of jobs 60 having similar characteristics with one of the plurality of host computers 2 that has corresponding resources at that time. In this way, the scheduling unit 20 will account for transient effects in the resources of each of the plurality of host computers 2. The periodic assessment can occur at any periodic rate as determined by the system administrator or designer and could depend on the workload being sent to the network 8 at a particular time. The scheduling unit 20 will also match individual jobs 6 which have not been grouped by the grouping unit 12, to one of the plurality of host computers 2 depending on the resources of the plurality of host computers 2 and the requirements of the individual job 6.
Once the scheduling unit 20 has matched the individual job 6 or the group of jobs 60 to one of the plurality of host computers 2, the dispatching unit 22 sends the job 6 or the group of jobs 60 to the matched host computer HM. (For convenience, the matched host computer will be referred to by symbol HM, but it is understood that the matched host computer HM will be any one of the plurality of host computers 2 to which the scheduling unit 20 has matched a job 6 or a group of jobs 60). To accomplish this, the dispatching unit 22 generally assigns a group header to each group of jobs 60 before sending the group of jobs 60 to the matched host computer HM. The group header may correspond to the group of jobs header 46 shown in Figure 4, but also identify the matched host computer HM of the plurality of host computers 2 to which the group of jobs 60 is being sent. Likewise, the dispatching unit 22 may also assign a header to each job 6 before sending the job 6 to the matched host computer HM. The header for each job 6 may correspond to the header 42 shown in Figure 4, but also identify the matched host computer HM to which the individual job 6 is being sent.

In a preferred embodiment, the dispatching unit 22 has the ability to send the group of jobs 60 to the matched computer HM either sequentially or in parallel. For example, if the grouped jobs 44 in a group of jobs 60 can share the resources of a plurality of host computers 2, the dispatching unit
22 will send the group of jobs 60 in parallel. In this way, the matched host computer HM will execute the grouped jobs 44 in parallel, which means they will be executed substantially simultaneously. If, however, the dispatching unit 22 determines that the grouped jobs 44 in a group of jobs 60 cannot share the resources of the matched host computer HM, the dispatching unit 22 will send the grouped jobs 44 to the matched host computer HM sequentially. It is also understood that this function of the dispatching unit 22 could be performed by a separate unit, such as a running unit (not shown), that manages the execution of the grouped jobs 44 in the group of jobs 60.

Once the group of jobs 60 and the jobs 6 have been processed by the corresponding one of the plurality of host computers 2, the finished jobs 60F and 6F are sent to the job finished unit 30 as shown in Figure 1. The job finished unit 30 may also separate each finished job 6F in each group of finished jobs 60F. In other words, the plurality of host computers 2 will forward groups of finished jobs 60F to the finished job unit 30 if groups of jobs 60 were originally sent to the plurality of host computers 2, and, the plurality of host computers 2 will forward to the job finished unit 30 individual finished jobs 6F if individual jobs 6 were originally sent to the plurality of host computers 2. In either case, the job finished unit 30 will eventually send the finished jobs, shown
generally by reference numeral 16 and symbol FJ in Figure 1, to the user who originally sent the job 6 to the network 8.

5 Preferably, the job finished unit 30 will send a finished signal FS to the device 10 through the counter updater 32. The finished signal FS will indicate each job 6 or group of jobs 60 that have been finished to assist the device 10 in monitoring processing of the jobs 6 and the groups of jobs 60.

Figure 2 shows a flowchart 200 illustrating the method performed by the device 10 to group the jobs 6 into groups of jobs 60 according to a preferred embodiment of the invention. As illustrated in Figure 2, the first step 210 is for a job 6 to be submitted to the device 10. The job 6 is then sent to the grouping unit 12 and the grouping unit 12 executes step 212, namely to determine whether or not the job 6 can be grouped according to the predetermined grouping criteria. If the job 6 cannot be grouped, the method proceeds to step 214 where the job 6 is rejected from the grouping. The job 6 is then dispatched individually to one of the plurality of host computers 2 by the mapping unit 14 as illustrated at step 216 in flowchart 200.

If at step 212 it is determined that the job 6 can be grouped according to the predetermined grouping criteria, the flowchart 200 proceeds to step 220 where a determination is made as to whether
or not there is a group of jobs in existence having similar characteristics to the job 6 to be grouped. In other words, the grouping unit 12 will assess the characteristics of the job 6 to be grouped, including an identification of the resources required to process the job 6. If there exists a group of jobs 60 with grouped jobs 44 having similar characteristics, the job 6 to be grouped is placed into that group of jobs 60. It is understood that these characteristics can include other factors in addition to the identification of resources required to process the job 6, such as whether or not they were submitted by the same user, whether they have the same resource and host requirements, whether they came from the same queue 4, whether they have the same command and whether they have the same pre-execution.

If the result from the step 220 is "yes", the job 6 is added to the existing group of jobs 60, as illustrated at step 222. If the result from step 220 is "no", a new group of jobs 60 is created, as illustrated at step 224 and that job 6 becomes the first job in this new group of jobs 60.

After either step 222 or step 224, the method proceeds to step 226 where a determination is made as to whether or not the group of jobs 60 has reached a predetermined maximum group number. If the result of step 226 is "yes", then the method proceeds to step 228 where the group of jobs 60 is
sent to the mapping unit 14 to map the group of jobs 60 to one of the plurality of host computers 2 to run the group of jobs 60. Step 228 is generally performed by the scheduling unit 20 within the mapping unit 14 as described above. The scheduling unit 20 determines the requirements of the group of jobs 60 and matches the group of jobs 60 with a matched host computer HM, which is one of the plurality of host computers 2 having resources which correspond to the requirements of the group of jobs 60, as described above.

The next step 230 would be to dispatch the group of jobs 60 to the matched host computer HM. Step 230 is generally performed by the dispatching unit 22 as described above. Flowchart 200 then returns to the initial step whereby a next job 6 is submitted to the device 10. It is clear that the steps in the flowchart 200 will be executed on the next job 6. More particularly, the next job 6 may or may not be grouped, and if grouped, may be placed in a different group from the previous job 6.

If at step 226 the result is “no” and the predetermined maximum group number has not been reached, the method proceeds to the time out step 232 to determine if the time limit has been reached. The time out step 232 is used to ensure that a particular group of jobs 60 does not remain in the device 10 for an inordinate period of time without being processed. The time limit or time out may
occur if one of the grouped jobs 44 in a group of jobs 60 has been present within the device 10 for more than a predetermined maximum amount of time without being sent to a matched host computer HM. If this predetermined maximum time limit has been reached, a time out occurs and no further time is permitted for the group of jobs 60 to obtain additional jobs. Rather, the group of jobs 60 is sent to step 228 to be matched and dispatched to a matched host computer HM. If the result from step 232 is "no", indicating that the time out has not occurred, the method returns to the initial step 210 whereby another job 6 is submitted to the device 10.

The time out at step 232 may also occur based on factors other than the amount of time one of the grouped jobs 44 in a group of jobs 60 has been present within the device 10. For example, a time out may occur if resources, such as one of the plurality of host computers 2, becomes available so that the device 10 will dispatch a group of jobs 60 to efficiently utilize available resources. A time out may also occur if there are changes on the load of the plurality of host computers 2. In addition, if a grouped job 44 in a group of jobs 60 has a high priority and must be executed immediately, a time out for that particular group of jobs 60 will occur.

It is understood that the steps of flowchart 200 could be executed by the device 10 as described and illustrated above, or by another device (not
shown). Furthermore, it is understood that the steps of the flowchart 200 could be executed by a computer processor, executing pre-programmed computer software instructions. It is also understood that while the steps in flowchart 200 were described with respect to one job 6, the flowchart 200 is not limited to processing a job 6 at any one time. Rather, different jobs 6 could be executed at each step of the flowchart 200.

Furthermore, the device 10 may incorporate several processors, each one executing the method illustrated by flowchart 200 simultaneously. It is also understood that the jobs 6 and the groups of jobs 60 may be stored at any data storage location, provided the grouping unit 12 has access to each group of jobs 60 so as to be able to add jobs 6 to existing groups of jobs 60, create new groups of jobs 60 when necessary and dispatch the groups of jobs 60.

Figures 3A, 3B and 3C further illustrate the grouping method of the device 10. Figure 3A illustrates a number of jobs 6 present on the queue 4 for submission to the device 10. Figure 3B illustrates some of the jobs 6, namely jobs 1, 2, 3, 22, 23, 24, 45, 49, 56, 57, 99 and 106 from Figure 3A, having been grouped by the grouping unit 12. Accordingly, the grouped jobs 44 within the group of jobs 60 shown in Figure 3B would have satisfied the predetermined grouping criteria of the grouping unit 12, as described above, and have been grouped by the
grouping unit 12. Further, if the preferred embodiment illustrated by flowchart 200, and in particular steps 220, 222 and 224, has been followed, the grouped jobs 44 within the group of jobs 60 shown in Figure 3B would have similar characteristics, such as requiring similar resources to be processed. Therefore, the grouped jobs 44 within the group of jobs 60 shown in Figure 3B would satisfy the predetermined grouping criteria, namely they are short jobs and the estimated time for processing the grouped jobs 44 is less than the predetermined duration, and, they would have similar characteristics, such as they would have similar requirements for processing.

To facilitate matching the group of jobs 60 to one of the plurality of host computers 2, in a preferred embodiment, a representative job 300, which in the example illustrated in Figure 3B is job 56, would be selected from the group of jobs 60. As the jobs 6 have been grouped in the group of jobs 60 because they have similar characteristics, it is likely that the representative job 300 would fairly accurately represent the requirements of all of the jobs 6 in the group of jobs 60. In this way, the scheduling unit 20 would match the group of jobs 60 with one of the plurality of host computers 2 by matching the representative job 300 with the plurality of host computers 2. Thus, the scheduling unit 20 must match only the representative job 300, rather than each grouped job 44, in order to match
the entire group of jobs 60 to one of the plurality of host computers 2. This requires the scheduling unit 20 to perform much less processing, thereby decreasing the scheduling overhead, to match the group of jobs 60 to one of the plurality of host computers 2, in large part because the grouping unit 12 has already grouped the jobs 6 into a group of jobs 60 having similar characteristics.

It is understood that this process of matching a group of jobs 60 by matching a representative job 300 of the grouped jobs 44 can be incorporated into the method illustrated by flowchart 200. For example, step 228 of matching the group of jobs 60 to the one of the plurality of host computers 2 could, in this preferred embodiment, comprise the substeps of selecting a representative job 300 and matching the representative job 300 to one of the plurality of host computers 2.

Figure 3C illustrates the process of the scheduling unit 20 matching the representative job 300 to one of the host computers A to I. As illustrated in Figure 3C, the primary candidates would be host computers B, E and I, which determination may be made based on the factors described above. The host computers B, E and I may not necessarily be the “best” ones of the plurality of host computers 2 for processing the representative job 300, but they would likely be better than another of the plurality of host
computers 2. If the resources of host computers B, E and I are equally appropriate for processing representative job 300, the scheduling unit 20 will arbitrarily match the representative job 300 to one of the host computers B, E and I and send the group of jobs 60 to the matched host computer HM.

In a preferred embodiment, as described and illustrated herein, the jobs 6 being grouped into groups of jobs 60 will require relatively less time to process, and therefore would be "short jobs". As such, by placing the short jobs into groups of jobs 60, the matched host computer HM which will execute the group of jobs 60 will require additional time to execute the group of jobs 60 as compared to executing a single one of the grouped jobs 44 within the group of jobs 60. This increases the likelihood that the matched host computer HM will be processing the grouped jobs 44 while the device 10 is allocating another job 6 or group of jobs 60 to the matched host computer HM. In this way, the idle time of the matched host computer HM decreases, thereby increasing the overall efficiency of the entire network 8. Furthermore, as the device 10 will group several jobs 6 into groups of jobs 60, the overall resources and time required to map the grouped jobs 44 to one of the plurality of host computers 2 decreases greatly.

It is understood that while the present invention has been described with respect to jobs 6
having certain characteristics and an estimated duration to process which is below a predetermined duration, the present invention is not necessarily limited to these specific characteristics and values. Rather, the present invention would encompass jobs having various characteristics and predetermined durations that will be set by system administrators and persons skilled in the art for each specific situation.

It will be understood that, although various features of the invention have been described with respect to one or another of the embodiments of the invention, the various features and embodiments of the invention may be combined or used in conjunction with other features and embodiments of the invention as described and illustrated herein.

Although this disclosure has described and illustrated certain preferred embodiments of the invention, it is to be understood that the invention is not restricted to these particular embodiments. Rather, the invention includes all embodiments which are functional, electrical or mechanical equivalents of the specific embodiments and features that have been described and illustrated herein.
The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. In a network comprising at least two host computers, each host computer having resources for processing jobs, a device for allocating jobs to the host computers comprising:
   a grouping unit for grouping at least some of the jobs into groups of jobs according to predetermined grouping criteria;
   a mapping unit for mapping jobs to the host computers, said mapping unit mapping each group of jobs to one of the host computers.

2. The device as defined in claim 1 wherein the predetermined grouping criteria comprises a predetermined duration such that jobs will be grouped into a group of jobs if the estimated duration to process a job is less than the predetermined duration.

3. The device as defined in claim 2 wherein the mapping unit individually maps jobs having an estimated duration greater than the predetermined duration to a host computer.

4. The device as defined in claim 2 wherein the grouping unit identifies characteristics of each job; and
wherein the grouping unit groups together jobs having similar characteristics.

5. The device as defined in claim 4 wherein said characteristics of each job comprise an identification of resources required to process each job.

6. The device as defined in claim 5 wherein for each job to be grouped, the grouping unit adds the job to be grouped to an existing group of jobs having jobs with characteristics similar to the job to be grouped.

7. The device as defined in claim 6 wherein if the grouping unit cannot identify an existing group of jobs having jobs with characteristics similar to the job to be grouped, the grouping unit creates a new group and makes the job to be grouped a first job in the new group.

8. The device as defined in claim 5 wherein the mapping unit comprises:
   a scheduling unit for matching each group of jobs which has similar characteristics with one of the host computers which has corresponding resources; and
   a dispatching unit for sending each group of jobs to the one matched host computer which has corresponding resources.
9. The device as defined in claim 8 wherein the resources of the host computers are time sensitive; and

wherein the scheduling unit periodically
assesses the resources of each host computer to
match each group of jobs which has similar
characteristics with one of the host computers which
has corresponding resources at that time.

10. The device as defined in claim 8 wherein the scheduling unit selects a representative job in a
group of jobs; and

wherein the scheduling unit matches a group of
jobs having similar characteristics with one of the
host computers which has corresponding resources by
matching the representative job in the group of jobs
with one of the host computers which has resources
corresponding to the characteristics of the
representative job.

11. The device as defined in claim 8 wherein the dispatching unit assigns a group header to each
group of jobs before sending the group of jobs to
the one matched host computer; and

wherein the group header identifies the group
of jobs and the one matched host computer to which
the group of jobs is being sent.

12. The device as defined in claim 8 wherein the
dispatching unit sends each group of jobs to the one
matched host computer which has corresponding
resources once a number of jobs in each group reaches a predetermined maximum group number, or, a time out occurs.

13. The device as defined in claim 8 wherein the dispatching unit sends each group of jobs to the one matched host computer which has corresponding resources in parallel, if the jobs can share the resources, and, sequentially, if the jobs cannot share the resources.

14. In a network comprising at least two host computers, each host computer having resources for processing jobs, a method for allocating jobs to the host computers comprising the steps of:

   a) grouping at least some of the jobs into groups of jobs according to predetermined grouping criteria; and

   b) mapping each group of jobs to one of the host computers.

15. The method as defined in claim 14 wherein the predetermined grouping criteria comprises a predetermined duration such that jobs will be grouped into a group of jobs if the estimated duration to process a job is less than the predetermined duration.

16. The method as defined in claim 15 further comprising the step of:
c) for each job having an estimated duration greater than the predetermined duration, mapping the job individually to a host computer.

17. The method as defined in claim 15 further comprising the steps of:
   a1) prior to grouping the jobs, identifying characteristics of each job, said characteristics of each job comprising resources required to process the job; and
   a2) grouping together jobs which have similar characteristics by adding a job to be grouped to an existing group of jobs which has jobs with characteristics similar to the job to be grouped.

18. The method as defined in claim 17 wherein if an existing group of jobs which has jobs with characteristics similar to the job to be grouped cannot be identified, performing the substeps of:
   i) creating a new group of jobs; and
   ii) making the job to be grouped a first job in the new group of jobs.

19. The method as defined in claim 17 wherein step (b) of mapping each group of jobs to one of the host computers comprises the substeps of:
   b1) matching each group of jobs which has similar characteristics with one of the host computers which has corresponding resources; and
b2) sending each group of jobs to the one matched host computer which has corresponding resources.

20. The method as defined in claim 19 further comprising the steps of:
   assessing the resources of each host computer on a periodic basis; and
   matching each group of jobs which has similar characteristics with the one matched host computer which has corresponding resources at that time.

21. The method as defined in claim 19 further comprising the steps of:
   selecting a representative job in a group of jobs; and
   matching the group of jobs which have similar characteristics with one of the host computers which has corresponding resources by matching the representative job in the group of jobs with one of the host computers that has corresponding resources to the representative job.

22. The method as defined in claim 19 wherein the step of sending groups of jobs to the one matched host computer comprises the substeps of:
   determining if a number of jobs in the group of jobs has reached a predetermined maximum group number, and, determining if a time out has occurred for a group of jobs;
sending the group of jobs to the one matched host computer if either the number of jobs in the group of jobs has reached a predetermined maximum group number, or, a time out has occurred.

23. The method as defined in claim 19 wherein the step of sending groups of jobs to the one matched host computer comprises the substeps of:

determining whether the jobs in the group of jobs can share the resources of the one matched host computer; and

sending the jobs in the group of jobs in parallel if the resources can be shared, and, sending the jobs in the group of jobs sequentially if the resources cannot be shared.
Figure 2
Figure 3A
Figure 3B

Figure 3C
Figure 4