

(12) UK Patent Application (19) GB (11) 2 370 140 (13) A

(43) Date of A Publication 19.06.2002

(21) Application No 0120564.0

(22) Date of Filing 23.08.2001

(30) Priority Data

(31) 09652510

(32) 31.08.2000

(33) US

(71) Applicant(s)

Hewlett-Packard Company
(Incorporated in USA - Delaware)
3000 Hanover Street, Palo Alto, California 94304,
United States of America

(72) Inventor(s)

Thomas Edwin Turicchi Jr
Doug Grumann

(74) Agent and/or Address for Service

Carpmaels & Ransford
43 Bloomsbury Square, LONDON, WC1A 2RA,
United Kingdom

(51) INT CL⁷

G06F 11/34

(52) UK CL (Edition T)

G4A AFMT

(56) Documents Cited

WO 99/32970 A1

WO 98/52122 A1

Varki E. and Dowdy L., "Proceedings of MASCOT 96", 1996, pages 291-295, "Response time analysis of two server fork-join systems"

IBM Technical Disclosure Bulletin "Optimization of computer configurations" published 1975, Vol 18 No. 7, Angiulli et al, pages 2235 - 2237

(58) Field of Search

UK CL (Edition T) G4A AFMT

INT CL⁷ G06F 11/34

Online: WPI, EPODOC, PAJ, INSPEC, ELSEVIER, IBM
TDB, INTERNET

(54) Abstract Title

A delay accounting method for computer system response time improvement

(57) A method for determining the system resource parameters [220] to change for specified users [215] or groups of users [215] in order to improve system performance as defined by response times [115]. Methods are disclosed identifying system resources [205] whose parameters [220] are in need of adjustment in order to improve response time [115] for service by those resources [220]. In a representative embodiment, measured results of response time [115] for service provided by various system resources [205] is aggregated for a selected user [215] or group of users [215]. Averages of such aggregations indicate which resources [220] consume excessive amounts of time for processes [210] used by the user [215] or group of users [215]. Beneficial changes in values of system resources [205] related parameters [220] are then implemented.

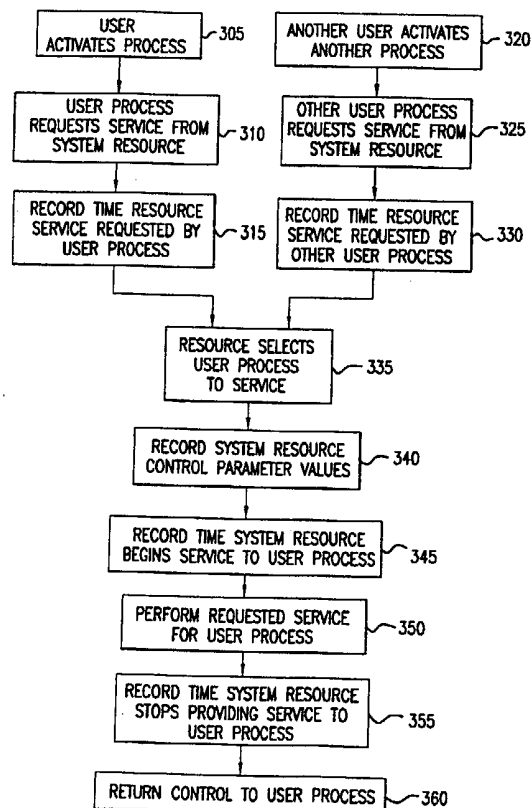


FIG.3

GB 2 370 140 A

$$\begin{array}{c}
 115 \quad \swarrow \\
 \text{TIME DELAY FOR} \\
 \text{PROCESS} \\
 \text{REQUESTING} \\
 \text{SERVICE BY A} \\
 \text{RESOURCE}
 \end{array}
 =
 \begin{array}{c}
 110 \quad \swarrow \\
 \text{DELAY WAITING} \\
 \text{FOR RESOURCE}
 \end{array}
 +
 \begin{array}{c}
 105 \quad \swarrow \\
 \text{TIME REQUIRED TO PERFORM} \\
 \text{REQUESTED SERVICE}
 \end{array}$$

FIG.1

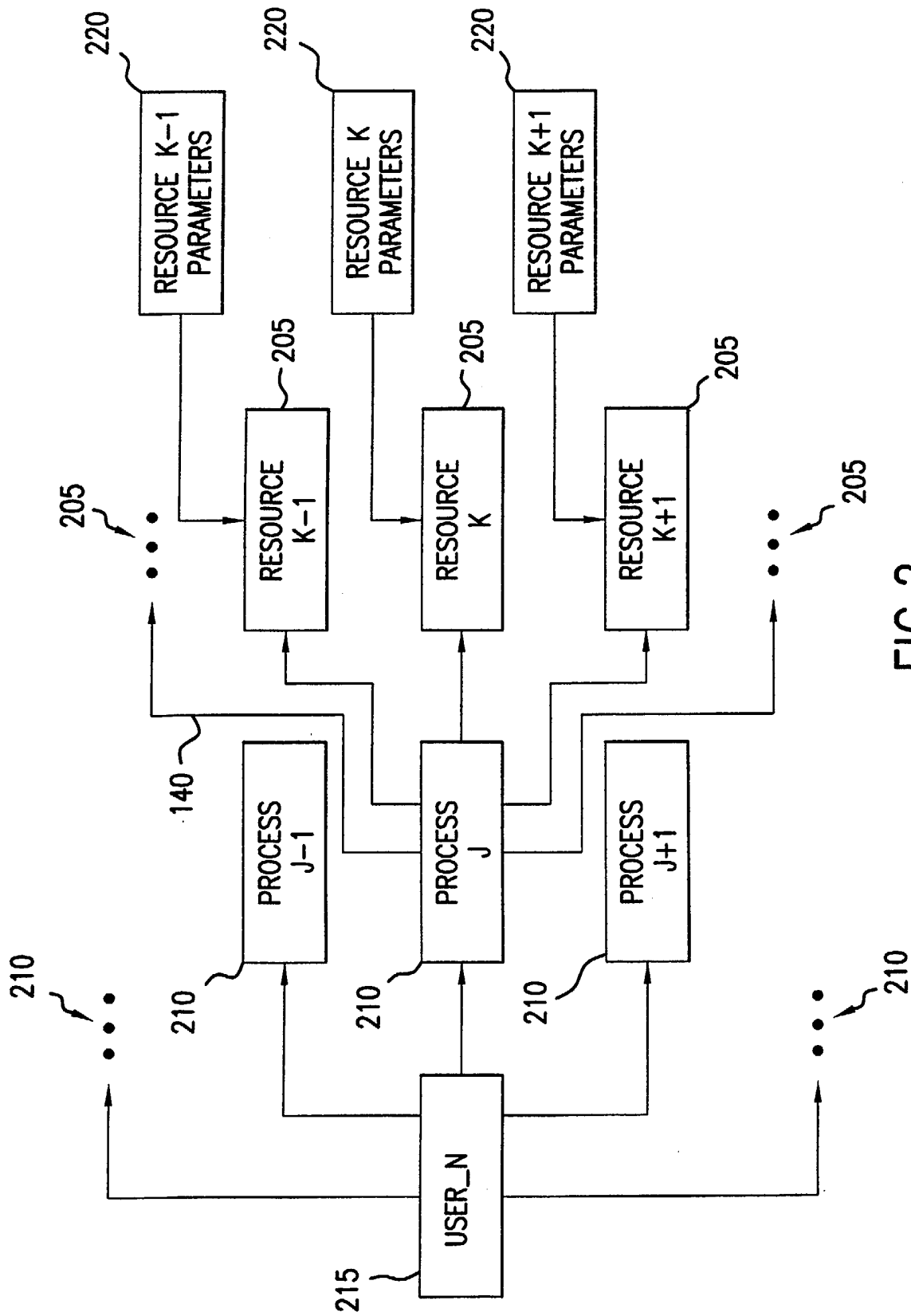


FIG.2

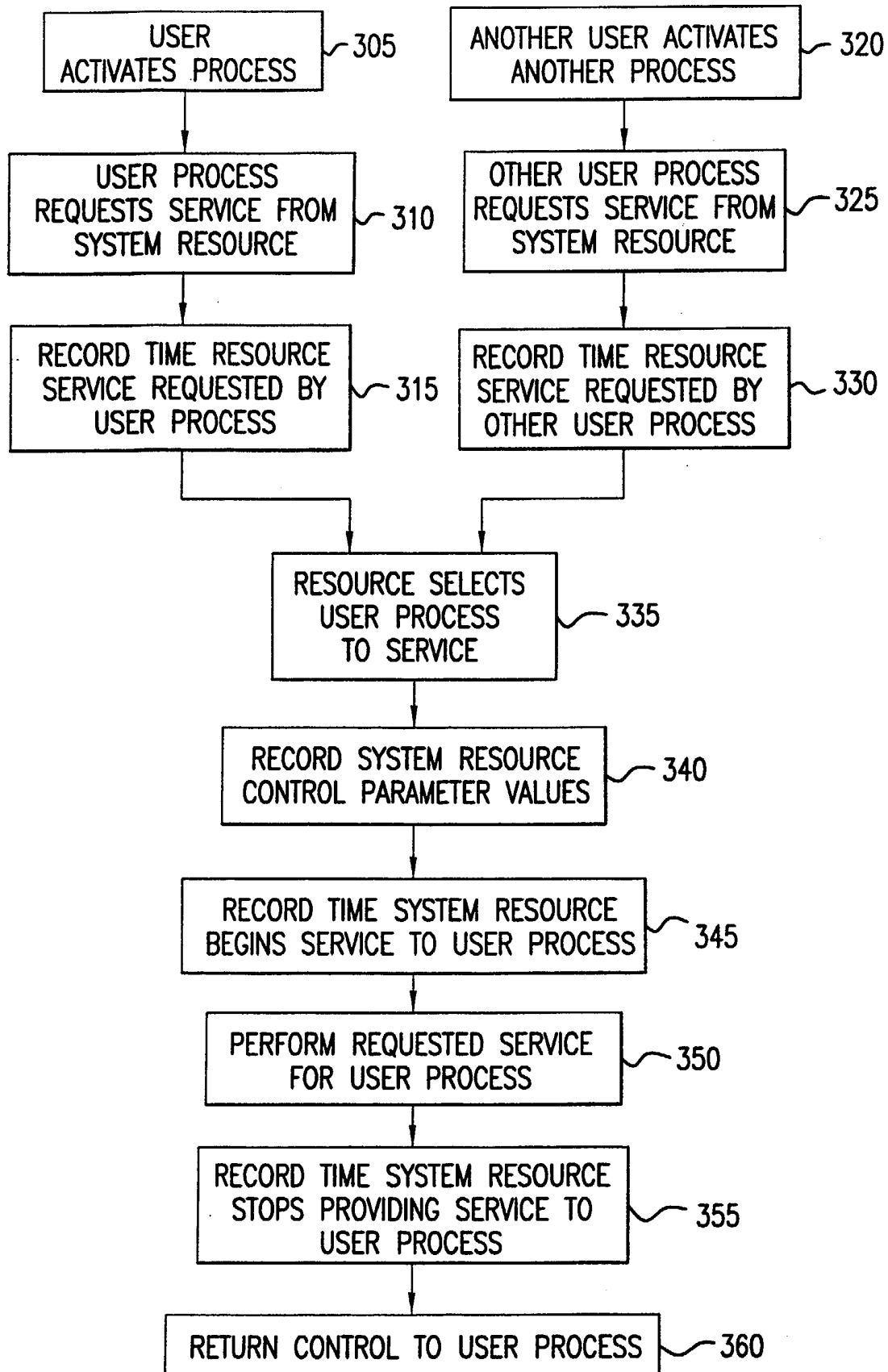


FIG.3

A DELAY ACCOUNTING METHOD FOR COMPUTER SYSTEM RESPONSE TIME IMPROVEMENT

5

FIELD OF THE INVENTION

The present invention relates generally to computer systems and, more particularly, to the adjustment of parameters to values that result in improved system performance as reflected in the service level provided to various users.

10

BACKGROUND OF THE INVENTION

Modern computing systems generally have numerous processes, often belonging to different users, running on the computer system at the same time. In order to improve system performance, it is often desirable to provide a different level of service for different processes. Administrators of modern computing systems generally have various parameters available that can be adjusted or configured to values that result in improved performance of the system. Size of swap space, memory dedicated to buffer-cache, maximum number of processes permitted, and maximum application data size are examples of such parameters that could be so adjusted.

15

One method of determining which system parameters can be used to manipulate service-level is by educated guesswork and experimentation by the operator. The disadvantages of this method are (1) it is time intensive for the computer operator, (2) its effectiveness is limited by the competency of the operator, and (3) in order to be effective under changing system loads and circumstances, it must be done continuously.

20

An alternative and more common method is to only change a very limited number of parameters, usually those that the operator is most familiar with. This has the

disadvantage that it is often not as effective as one would like.

Thus, there is a need for a method by which a computer system can be more effectively and efficiently improved. In particular, there is a need for a method to automatically and continually adjust system parameters to obtain improved system performance, even as the workload on the system changes.

5

SUMMARY OF THE INVENTION

The present patent document relates to a novel method for improving computer system performance by the adjustment of computer system parameters. This method can make appropriate adjustments even as the workload on the system changes. Previous methods for improving system performance have relied upon speculation as to the optimal system configuration or by controlling a limited number of system parameters.

Modern computer systems typically service a number of users and each of these users will have at various times one or more processes operating on the computer. Each process may need a variety of system resources often including the same resources needed by another process or processes. For the very restricted case of only one process requesting service by a system resource, the time delay for servicing the request by the process is the time which the resource needs in order to perform that request. For multi-user computer systems, an additional increment of time, the time delay results from the process having to wait until the resource is available to the process. The sum of these two times is the total time which the process will experience in requesting service by the resource. For a single user having a single active process this sum is also referred to as the response time, i.e., the time in which the user receives the service which he requested from the process. More than one process by the user will require a summation over all the processes which the user has activated.

When running various processes by the same or different users multiple workloads on the same system, or when attempting to give different users of the same workload different service levels, one needs the capability to (a) group processes according to the users or workload that they are performing, and (b) use the controls (knobs) provided by the system to allocate resources to the different groups of processes in order to control the service level that the end-users of those workloads receives. Modern computing systems have a variety of controls available, but it is often very difficult for the system operator to determine how to set or change the "knob" settings in order to manipulate the service level that the end-user receives. Representative embodiments disclosed herein monitor the running system and advise the operator which

"knobs" can be used most effectively to manipulate the service-level of each workload.

In order to remove double counting of wait times, during those times that a user has more than one process waiting for service by a resource, the method counts that part of the wait time only once. The summations assume a single microprocessor. In cases
5 where there are multiple microprocessors, some of the wait times, as well as processing times, occur in parallel and the response times at various levels will be less than or equal to that indicated by the equations.

In a representative embodiment, the time delay experienced by all processes for efforts done on behalf of a particular user during a request for service is aggregated. The
10 time delay is further aggregated and averaged on behalf of that user over multiple requests for service. This aggregation could also be done for a group of related users.

Decisions as to changing or not changing system parameter values for selected system resources is then made typically based on the results of the delay times obtained. Those system resources that have the longer delays are easily identified and appropriate
15 changes made in order to improve the response time for the user or user groups of interest.

Representative embodiments, as described in the present patent document, are advantageous in that they take very little time or effort on the part of the system operator to arrive at an improved system configuration. Additionally, as the usage characteristics
20 and load on the system change, this method finds and recommends new improvements to the system, without requiring the operator to continuously monitor and redo previous adjustments that were performed by hand. Another benefit is that this mechanism can be used on a subset of the parameters available on a system. Thus, the operator may choose to use this method to look for improvements in obscure or less understood parameters,
25 and make the macroscopic adjustments himself.

The operator is provided much greater visibility into how to best manipulate the service levels experienced by a user or group of users. The method is time efficient and the actions taken by the operator can be effective in response time improvement.

Other aspects and advantages of the present invention will become apparent from
30 the following detailed description, taken in conjunction with the accompanying drawings,

illustrating by way of example the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings provide visual representations which will be used to more fully describe the invention and can be used by those skilled in the art to better understand it and its inherent advantages. In these drawings, like reference numerals identify corresponding elements and:

Figure 1 is a drawing of an equation for a time delay experienced by a process requesting service by a resource as described in various representative embodiments of the present patent document.

Figure 2 is a diagram of a user-process-resource interaction as described in various representative embodiments of the present patent document.

Figure 3 is a flow chart of a process for accumulating data for a delay accounting method as described in various representative embodiments of the present patent document.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

1. Introduction:

As shown in the drawings for purposes of illustration, the present patent document relates to a novel method for improving computer system performance by the adjustment of computer system parameters. This method can make appropriate adjustments even as the workload on the system changes. Previous methods for improving system performance have relied upon speculation as to the optimal system configuration or by controlling a limited number of system parameters. In the following detailed description and in the several figures of the drawings, like elements are identified with like reference numerals.

When running various processes by the same or different users multiple workloads on the same system, or when attempting to give different users of the same workload different service levels, one requires the capability to (a) group processes according to the users or workload that they are performing, and (b) use the controls (knobs) provided by the system to allocate resources to the different groups of processes in order to control the service level that the end-users of those workloads receives. Modern computing systems have a variety of controls available, but it is often very difficult for the system operator to determine how to set or change the "knob" settings in order to manipulate the service level that the end-user receives. Representative embodiments disclosed herein monitor the running system and advise the operator which "knobs" can be used most effectively to manipulate the service-level of each workload.

2. The Computer System:

Figure 1 is a drawing of an equation for a time delay experienced by a process requesting service by a resource as described in various representative embodiments of the present patent document. Modern computer systems typically service a number of users and each of these users will have at various times one or more processes operating on the computer. Each process may need a variety of system resources often including the same resources needed by another process or processes. For

the very restricted case of only one process requesting service by a system resource, the time delay 115 for servicing the request by the process is the time 105 which the resource needs to perform that request. For multi-user computer systems, an additional time delay 110 results from the process having to wait until the resource is available to the process.

5 The sum of these two times is the total time 115 which the process will experience in requesting service by the resource. For a single user having a single active process this sum is also referred to as the response time 115, i.e., the time in which the user receives the service which he requested from the process. More than one process by the user will require a summation over all the processes which the user has activated.

10 Figure 2 is a diagram of a user-process-resource interaction as described in various representative embodiments of the present patent document. The computer system comprises a number of resources 205 only three of which, $k-1$, k , and $k+1$ are shown in figure 2. Other resources are indicated as rows of three dots above and below the three resources shown. Service may be requested by the various process 210 operable
15 on the computer system. The computer system further comprises a number of processes 210, only three of which, $j-1$, j , and $j+1$ are shown in figure 2. Other processes 210 are indicated as rows of three dots above and below the three processes shown. Various processes 210 are activated at various times by a computer system user 215 identified as the n th user or "user n " in figure 2. Typically the computer system further comprises a
20 number of users 215 not shown in figure 2. While figure 2 implies that any process 210 can access any resource 205, it will be recognized by one of ordinary skill in the art that some resources 205 may not be accessible by particular processes 210. Similarly figure 2 seems to imply that any user 215 can activate any process 210. However, it will be recognized by one of ordinary skill in the art that some processes 210 may not be
25 accessible by particular users 215.

3. Definitions & Equations:

The following definitions and equations describe time delays that will be experienced by the various users 215 of the computer system.

Service Level for a user for a given request = Response Time experienced for that request

The following symbols represent integers which are used as indices and which identify the item specified.

5

n = user n , the n th user of the computer system

I = identifying number associated with the i th request for service by a given user

10 j = identifying number associated with P_j , the j th process

k = identifying number associated with R_k , the k th resource

15 $T_A(n,i,P_j,R_k)$ = Time that the k th resource, R_k , spends in active work for the j th process, P_j , during the i th request for service by user n 215

$T_w(n,i,P_j,R_k)$ = Time that the j th process (P_j) spends waiting for the k th resource (R_k) during the i th request for service by user n 215

20 For a single process 210, the j th process 210, activated by the n th user 215, requesting service from a single resource 205, the k th resource 205, during the i th request by the n th user 215, the response time or time delay experienced is

$$T_s(n,i,P_j,R_k) = T_w(n,i,P_j,R_k) + T_A(n,i,P_j,R_k) \quad (\text{Eq. 1})$$

25

Total response time that user n 215 experiences for all processes P_j 210 using resources R_k 205 during the i th request by user n 215 for service.

$$T_{RK}(n,i,R_k) = \sum_j T_s(n,i,P_j,R_k) = \sum_j T_w(n,i,P_j,R_k) + \sum_j T_A(n,i,P_j,R_k) \quad (\text{Eq. 2})$$

30

In order to remove double counting of wait times, during those times that user n 215 has more than one process 210 waiting for service by resource R_k , $\sum_j T_w(n,i,P_j,R_k)$ counts that part of the wait time only once.

- 5 Average response time that user n 215 experiences for all processes P_j 210 using resources R_k 205 during multiple, for example a total of I requests, requests by user n 215 for service

$$T_{AVG_RK}(n) = \sum_i T_{RK}(n,i,R_k) / I \quad (\text{Eq. 3})$$

10

Total response time that user n experiences for all processes P_j 210 using all resources R_k 205 during the ith request by user n 215 for service

$$15 \quad T_R(n,i) = \sum_j \sum_k T_S(n,i,P_j,R_k) = \sum_j \sum_k T_w(n,i,P_j,R_k) + \sum_j \sum_k T_A(n,i,P_j,R_k) \quad (\text{Eq. 4})$$

For a total of I requests by user n 215, the average response time for user n is

$$20 \quad T_{AVG}(n) = [\sum_i T_R(n,i)] / I \quad (\text{Eq. 5})$$

For a total of N users 215, the average response time for each user is

$$25 \quad T_{ALL} = [\sum_n T_{AVG}(n)] / N \quad (\text{Eq. 6})$$

The above summations assume a single microprocessor. In cases where there are multiple microprocessors, some of the wait times, as well as processing times, occur in parallel and the response times at various levels will be less than or equal to that above.

30

4. Representative embodiments:

Figure 3 is a flow chart of a process for accumulating data for a delay accounting method as described in various representative embodiments of the present patent document. In block 305 the user 215, for example user n 215, requests service by process 5 210, for example process j 210. Block 305 then transfers control to block 310.

In block 310, process j 210 requests service of system resource 205, for example resource k 205. Block 310 then transfers control to block 315.

In block 315, resource k 205 records the time process j 210 requested its service, also referred to herein as a first time. Block 315 then transfers control to block 335.

10 In block 320 another user 215, for example user n-1 215, requests service by the same process j 210. Block 320 then transfers control to block 325.

In block 325, process j-1 210 requests service of the same system resource k 205, for example resource k 205. Block 325 then transfers control to block 330.

15 In block 330, resource k 205 records the time process j-1 210 requested its service. Block 330 then transfers control to block 335.

In block 335 resource k 205 selects which of the two competing requests for service that it will service next, for example the request from process j 210. Block 335 then transfers control to block 340.

20 In block 340 resource k 205 records salient system resource control parameter values for resource k 205. Block 340 then transfers control to block 345.

In block 345 resource k records the time that the system resource k 205 begins service to process j 210, also referred to herein as a second time. Block 345 then transfers control to block 350.

25 In block 350 system resource k 205 performs the service requested by process j 210. Block 350 then transfers control to block 355.

In block 355 resource k records the time that the system resource k 205 ends service to process j 210, also referred to herein as a third time. Block 355 then transfers control to block 360.

In block 360 resource k returns control to process j 210.

30 $T_{w(n,i,P_j,R_k)}$ which, as defined above, is the time that process j 210 spends

waiting for service by resource k 205 during the i th request for service by user n 215 is obtained as the subtraction of the clock time recorded in block 315 (time request for service made) from the clock time recorded in block 345 (time service begun). $T_A(n,i,P_j,R_k)$ which, as defined above, is the time that resource k 205 spends in active
5 work for the process j 210 is obtained as the subtraction of the clock time recorded in block 345 (time service begun) from the clock time recorded in block 355 (time service ended). For process j 210, activated by the user n 215, requesting service from resource k 205, during the i th request by user n 215, the response time or time delay experienced is then $T_S(n,i,P_j,R_k)$ as computed in equation 1.

10 In a representative embodiment, the time delay experienced by all processes for efforts done on behalf of a particular user 215, as for example user n 215, during the i th request for service is aggregated as given by equation 2, and further aggregated and averaged on behalf of that user 215, as for example user n 215, during multiple requests for service is aggregated as given by equation 3. This aggregation could also be done for
15 a group of related users 215.

Decisions as to changing or not changing the system parameter value for system resource 205 is then made typically based on the results of equation 3. Those system resources that have the longer delays are easily identified and appropriate changes made in order to improve the response time for the user or user groups of interest. It should be
20 noted that other aggregations and averages are possible as specified in equations 4-6.

5. Concluding Remarks:

Representative embodiments, as described in the present patent document, are advantageous in that they take very little time or effort on the part of the system operator
25 to arrive at an unproved system configuration. Additionally, as the usage characteristics and load on the system change, this method finds and recommends new improvements to the system, without requiring the operator to continuously monitor and redo previous adjustments that were performed by hand. Another benefit is that this mechanism can be used on a subset of the parameters available on a system. Thus, the operator may choose
30 to use this method to look for improvements in obscure or less understood parameters,

and make the macroscopic adjustments himself.

The operator is provided much greater visibility into how to best manipulate the service levels experienced by a user or group of users. The method is time efficient and the actions taken by the operator can be effective in response time improvement.

5 While the present invention has been described in detail in relation to representative embodiments thereof, the described embodiments have been presented by way of example and not by way of limitation. It will be understood by those skilled in the art that various changes may be made in the form and details of the described embodiments resulting in equivalent embodiment that remains within the scope of the
10 appended claims.

CLAIMS

1. A program storage medium readable by a computer, tangibly embodying
2 a software program of instructions executable by the computer to perform
method steps for computer system performance improvement, the
4 method steps comprising:

6 recording a first time [315], wherein the first time is the computer clock
time that a user's [215] computer process [210] requests service from a
8 computer system resource [205];

10 recording value [340] of at least one preselected computer system
resource parameter [220];
12
14 recording a second time [345], wherein the second time is the computer
clock time that the resource [205] initiates service to the computer process
[210];
16
18 recording a third time [355], wherein the third time is the computer clock
time that the resource [205] completes the requested service;
20
22 subtracting the first time [315] from the second time [345] and assigning
the result to a wait time [110];

24 subtracting the second time [345] from the third time [355] and assigning
the result to a processing time [105];

26 summing the wait time [110] and the processing time [105];

28 assigning a response time [115] to the result of the method step summing
the wait time [110] and the processing time [105]; and

30
32 when a user service time exceeds a preselected value, wherein the user
service time is dependent upon the response time [115],

34 changing one of the recorded parameter values [220].

2. A program storage medium as recited in claim 1, providing the user
2 service time is equal to the response time [115].

3. A program storage medium as recited in claim 1, wherein the method
2 steps further comprise:

4 except the method step changing one of the recorded parameter values
[220] when the user service time exceeds a preselected value, performing
6 the method steps of claim 1 for at least one additional process [210]
requesting resource [205] service;

8
10 summing the response times [115] obtained for the processes [210],
wherein when more than one process [210] is waiting for service, the wait
time [110] is included in only one of the response times [115]; and

12
14 setting the user service time equal to the result of the method step
summing response times [115].

4. A program storage medium as recited in claim 1, wherein the method steps further comprise:

except the method step changing one of the recorded parameter values [220] when the user service time exceeds a preselected value, performing the method steps of claim 1 for at least one additional user [215] resource [205] service request, and for at least one additional process [210] requesting resource [205] service for at least one of the user [215] resource [205] service requests;

summing the response times [115] obtained for the processes [210], wherein when more than one process [210] is waiting for service, the wait time [110] is included in only one of the response times [115];

dividing the results of the method step summing the response times [115] by the number of requests activated by the user [215]; and

setting the user service time equal to the result of the method step dividing the results of the method step summing the response times [115].

5. A program storage medium as recited in claim 1, wherein the method steps further comprise:

except the method step changing one of the recorded parameter values [220] when the user service time exceeds a preselected value, performing the method steps of claim 1 for at least one additional process [210] requesting resource [205] service, and for at least one additional resource [205] from which at least one of the processes [210] requests resource [205] service;

10 summing the response times [115] obtained for the processes [210],
wherein when more than one process [210] is waiting for service, the wait
12 time [110] is included in only one of the response times [115]; and

14 setting the user service time equal to the result of the method step
summing response times [115].

6. A program storage medium as recited in claim 1, wherein the method
2 steps further comprise:

4 except the method step changing one of the recorded parameter values
[220] when the user service time exceeds a preselected value, performing
6 the method steps of claim 1 for at least one additional user [215] resource
[205] service request, for at least one additional process [210] requesting
8 resource [205] service for at least one of the user [215] resource [205]
service requests, and for at least one additional resource [205] from which
10 at least one of the processes [210] requests resource [205] service for at
least one of the user [215] resource service requests;

12 summing the response times [115] obtained for the processes [210],
14 wherein when more than one process [210] is waiting for service, the wait
time [110] is included in only one of the response times [115];

16 dividing the results of the method step summing the response times [115]
18 by the number of requests by the user [215]; and

20 setting the user service time equal to the result of the method step
dividing the results of the method step summing the response times [115].

7. A program storage medium as recited in claim 1, wherein the method

2 steps further comprise:

4 except the method step changing one of the recorded parameter values
 6 [220] when the user service time exceeds a preselected value, performing
 8 the method steps of claim 1 for at least one additional user [215], for at
 10 least one additional user [215] resource [205] service request by at least
 12 one of the users [215], for at least one additional process [210] requesting
 resource [205] service for at least one of the user [215] resource [205]
 service requests by at least one of the users [215], and for at least one
 additional resource [205] from which at least one of the processes [210]
 requests resource [205] service for at least one of the user [215] resource
 service requests by at least one of the users [215];

14 summing the response times [115] obtained for the processes [210],
 16 wherein when more than one process [210] is waiting for service, the wait
 time [110] is included in only one of the response times [115];

18 performing all above method steps for at least one additional user [215];

20 dividing the results of the method step summing the response times [115]
 22 by the number of requests by the users [215]; and

24 setting the user service time equal to the result of the method step
 dividing the results of the method step summing the response times [115].

8. A computer operable method for computer system performance
 2 improvement, comprising the steps of:

4 recording a first time [315], wherein the first time is the computer clock
 time that a user's [215] computer process [210] requests service from a

6 computer system resource [205];

8 recording value [340] of at least one preselected computer system
resource parameter [220];

10 recording a second time [345], wherein the second time is the computer
12 clock time that the resource [205] initiates service to the computer process
[210];

14 recording a third time [355], wherein the third time is the computer clock
16 time that the resource [205] completes the requested service;

18 subtracting the first time [315] from the second time [345] and assigning
the result to a wait time [110];

20 subtracting the second time [345] from the third time [355] and assigning
22 the result to a processing time [105];

24 summing the wait time [110] and the processing time [105];

26 assigning a response time [115] to the result of the method step summing
the wait time [110] and the processing time [105]; and

28 when a user service time exceeds a preselected value, wherein the user
30 service time is dependent upon the response time [115],

32 changing one of the recorded parameter values [220].

9. A computer operable method as recited in claim 8, providing the user
2 service time is equal to the response time [115].

10. A computer operable method as recited in claim 8, which further
2 comprises the method steps of:

4 except the method step changing one of the recorded parameter values
[220] when the user service time exceeds a preselected value, performing
6 the method steps of claim 15 for at least one additional process [210]
requesting resource [205] service;

8

10 summing the response times [115] obtained for the processes [210],
wherein when more than one process [210] is waiting for service, the wait
time [110] is included in only one of the response times [115]; and

12

14 setting the user service time equal to the result of the method step
summing response times [115].



INVESTOR IN PEOPLE

Application No: GB 0120564.0

Examiner: Michael Powell
Waters

Claims searched: 1 to 10

Date of search: 9 April 2002

Patents Act 1977 Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.T): G4A (AFMT)

Int Cl (Ed.7): G06F (11/34)

Other: Online: WPI, EPODOC, PAJ, INSPEC, ELSEVIER, IBM TDB, INTERNET

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	WO 99/32970 A1 (AB INITIO) see fig 4 and page 7 lines 3 to 30, page 10 lines 5 to 14 and "analysis over time" pages 12,13 and 14	1
A	WO 98/52122 A1 (COMPUWARE) see figure 13 and claim 1	
A	Varki E. and Dowdy L., "Proceedings of MASCOT 96", 1996, pages 291-295, "Response time analysis of two server fork-join systems" see section 4	
A	IBM Technical Disclosure Bulletin "Optimization of computer configurations" published 1975, Vol 18 No. 7, Angiulli et al, pages 2235 - 2237	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.