

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
31 August 2006 (31.08.2006)

PCT

(10) International Publication Number
WO 2006/090407 A1

(51) International Patent Classification:
G06F 12/00 (2006.01)

(21) International Application Number:
PCT/IN2005/000062

(22) International Filing Date:
23 February 2005 (23.02.2005)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant (for all designated States except US):
HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. [—/US]; 20555 S.H. 249, Houston, Texas 77070 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **KUMAR, Mupirala Kishore** [IN/IN]; S-1, Alakananda Apartment, Coconut Garden Layout, Nagarabhavi, Bangalore 560 072 (IN). **PRAKASH, Bhanu Gollapudi Venkat** [IN/IN]; #248, 3rd Cross, 2nd Block, 1st Stage, HBR Layout, Bangalore, Karnataka 560 043 (IN). **LAKSHMIKANTHAIAH, Phalachandra H** [IN/IN]; No 674, 6th Main, 2nd Cross, Vijayanagar, Bangalore 560 040 (IN).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

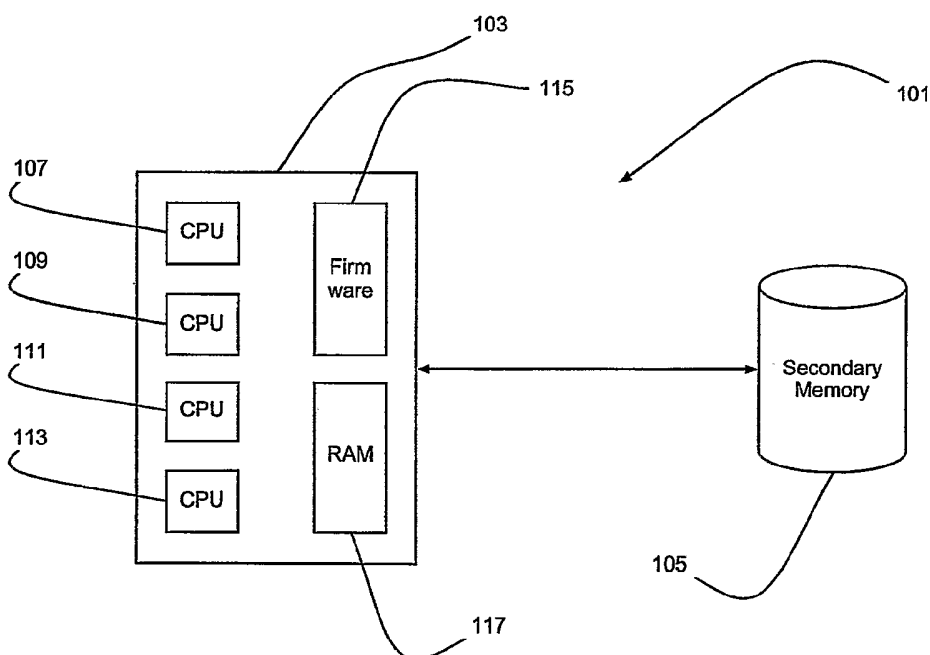
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search 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.

(54) Title: A METHOD OR APPARATUS FOR STORING DATA IN A COMPUTER SYSTEM



(57) Abstract: A method and apparatus is disclosed for storing data in a computer system in a memory dump in which when the operating system fails a portion of the system memory is freed by dumping the data from it. The computer system is then allowed to restart the OS working with the freed memory while a separate thread or processor, working either as a part of OS or firmware continues the dumping of the remaining system memory. When the remaining memory is freed it is made available for the OS.

WO 2006/090407 A1

A Method or Apparatus for Storing Data in a Computer System

Field of Invention

The present invention relates to a method or apparatus for storing data in a
5 computer system.

Background of the Invention

In computer systems, recovery from application software failures can be
achieved relatively easily by restarting the application. If the application is large,
10 application check pointing can help reduce the application recovery times.
Recovery from operating system (OS) failures takes longer compared to
application software failure, as an OS failure requires a reboot operation. Prior
to restarting an OS after a failure, a copy of the state or kernel image of the
failed OS along with its associated data is dumped or saved-off from system
15 memory to a pre-designated area of secondary storage associated with the
computer system. The secondary storage may be a single disk or a group of
disks or a partition within a disk. The dumped data is used in the diagnosis of
the OS failure. The computer system cannot run application software until the
new OS completes its reboot process.

20 Some computer systems include a mechanism which speeds up the dumping
process by only dumping the elements from the old OS that are relevant for
subsequent dump analysis. Other techniques to reduce the amount of time
taken to dump the failed OS use additional memory to first save specific parts of
25 relevant memory. One drawback of known systems is that the dumping process
delays the rebooting of the OS.

It is an object of the invention to reduce the time taken to reboot an OS after a
failure while still enabling the dumping of relevant data from the system
30 memory.

It is an object of the present invention to provide a method or apparatus for
storing data in a computer system, which avoids some of the above
disadvantages or at least provides a useful alternative.

35

Summary of the Invention

Some embodiments of the invention provide a method for storing data in a computer system, the method comprising the steps of:

- a) storing data from a first portion of a system memory in a secondary memory;
- 5 b) allocating the first portion of the system memory for subsequent use by an operating system (OS);
- c) rebooting and running the OS using the allocated memory; and
- d) storing data from a remaining portion of the system memory in the secondary memory.

10

The running of the OS and step d) may be carried out concurrently. Step a) may be carried out by firmware prior to steps b) to d). Step d) may be carried out by one or more processing threads of the OS. Each thread may be allocated a part of the remaining portion of system memory and when the data from the part is
15 stored in the secondary storage, the thread quits. Alternatively, step d) may be carried out under the control of firmware.

The computer system may comprise a plurality of CPUs and the processing of step d) may be allocated between the plurality of CPUs. Step d) may be carried
20 out by a subset of the plurality of CPUs. Each CPU may be allocated a part of the remaining portion of system memory and when the data from the part is stored in the secondary storage, each of the CPUs reverts to providing the OS. The method may further comprise the step of: e) allocating memory freed in step d) for use by the OS. In step d) the storing may be carried out in
25 predetermined blocks of the memory and in step e) as each the block is freed it may be allocated for use by the OS.

In step a) the data may be kernel data which is swapped out of the dump image. In step a) the first portion of the system memory may be the minimum amount
30 required to run the OS. In step a) the first portion may comprise up to 1% of the system memory.

Step a) may be carried out as part of a reboot of the computer system. The reboot may be carried out in response to an OS failure. The reboot operation
35 may be arranged to operate automatically in response to the OS failure.

Alternatively, step a) may be carried out as part of a back up operation for the system memory.

Other embodiments of the invention provide apparatus for storing data in a computer system; the apparatus comprising:
5 processing means operable to store data from a first portion of a system memory in a secondary memory;
a memory management system operable to allocate the first portion of the system memory for subsequent use by an operating system (OS); and
10 the processing means being further operable to reboot the OS using the allocated memory and to store data from a remaining portion of the system memory in the secondary storage.

Further embodiments of the invention provide a method of dumping data from a multiprocessor computer system memory during an OS reboot operation
15 comprising the steps of:
a) freeing a first portion of a computer system memory by saving data to a secondary storage area;
b) restarting the OS using the freed system memory and a first of the computer
20 system CPUs;
c) instructing a second of the CPUs to save the remaining data from the system memory to the secondary storage area; and
d) reallocating the second CPU for use by the OS when the saving of the remaining data is complete.

25 Further embodiments of the invention provide a method of dumping data from a computer system memory during an OS reboot operation comprising the steps of:
a) freeing a first portion of a computer system memory by saving data to a
30 secondary storage area;
b) restarting the OS using the freed system memory;
c) initiating a processing thread for saving the remaining data from the system memory to the secondary storage area; and
d) terminating the thread when the storage of the remaining data is complete.

35

Further embodiments of the invention provide apparatus for rebooting a computer system after an OS failure, the apparatus comprising:
means for storing data from a first portion of a system memory in a secondary memory area;

- 5 means for allocating the first portion of the system memory for subsequent use by the operating system (OS);
means for rebooting the OS using the allocated memory; and
means for concurrently storing data from a remaining portion of the system memory in the secondary storage area and providing the OS.

10

Further embodiments of the invention provide a computer processor for a computer system operable in response to an operating system (OS) failure to:
a) store data from a first portion of a system memory in a secondary memory area;

- 15 b) allocate the first portion of the system memory for subsequent use in rebooting the operating system (OS);
c) reboot the OS using the allocated memory; and
d) store data from a remaining portion of the system memory in the secondary storage area while also running the OS.

20

Further embodiments of the invention provide a computer program or group of programs arranged to enable a computer or group of computers to carry out a method for storing data in a computer system, the method comprising the steps of:

- 25 a) storing data from a first portion of a system memory in a secondary memory;
b) allocating the first portion of the system memory for subsequent use by an operating system (OS);
c) rebooting and running the OS using the allocated memory; and
d) storing data from a remaining portion of the system memory in the secondary
30 storage.

Further embodiments of the invention provide a computer program or group of programs arranged to enable a computer or group of computers to provide apparatus for storing data in a computer system, the apparatus comprising:

- 35 processing means operable to store data from a first portion of a system memory in a secondary memory;

a memory management system operable to allocate the first portion of the system memory for subsequent use by an operating system (OS); and the processing means being further operable to reboot and run the OS using the allocated memory and to store data from a remaining portion of the system memory in the secondary memory.

Brief Description of the Drawings

Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

- 10 Figure 1 is a schematic illustration of a computer system;
- Figures 2a & 2b are schematic illustrations of the reboot process of the computer system of Figure 1;
- Figure 3 is a flow chart illustrating the reboot process according to an embodiment of the invention; and
- 15 Figure 4 is a flow chart illustrating the reboot process according to another embodiment of the invention.

Detailed Description of Preferred Embodiments of the Invention

With reference to figure 1, a computer system 101 comprises a computer 103 connected to a secondary memory in the form of an external disk drive 105. The computer 103 is a multiple processor system having four central processing units (CPUs) 107, 109, 111, 113, firmware 115 and system memory in the form of random access memory (RAM) 117. The RAM 117 provides 100 gigabytes (GB) of system memory capacity. The firmware 115 is in the form of software written onto read-only memory (ROM). The firmware includes the basic input/output system (BIOS), which is software arranged to carry out the basic functions of the computer system such as controlling a keyboard, display screen, disk drives and serial communications.

- 30 The BIOS controls the initial start-up process of the computer system 101, during which the CPUs 107, 109, 111, 113 are initialized. Subsequently, a Unix™ operating system (OS) stored on the secondary memory 105 is loaded into the RAM 117 for operation. If a fault occurs during the operation of the OS then the OS must be restarted. The restart or reboot procedure is similar to the initial start-up of the computer system but with the addition step of carrying out a system memory dump before the OS itself can be restarted. The memory dump
- 35

involves saving data relating to the old, failed OS for later analysis. This saved data is commonly referred to as the old or dead system memory (OSM or DSM), a kernel image or an OS image.

- 5 The dumping process is initiated by software stored in the firmware and the process can be selective or non-selective. A selective dumping algorithm selects and saves only pages of system memory that contain kernel relevant data. A simpler non-selective dumping algorithm saves the whole system memory irrespective of relevance of the data for dump analysis. Non-selective
10 dumping is also referred to as a full dump. The present embodiment uses a non-selective dumping algorithm.

- The firmware 115 is arranged to carry out the reboot process in two phases. In the first phase, a first portion of the data from the OSM is saved to the disk drive
15 105. The amount of the first portion is arranged to free sufficient system memory for use by the new OS to enable it to boot-up and provide basic services to applications in the computer system 101. In addition, the new OS initially uses only two of the CPUs 107, 109. The size of this first portion of memory and the number of CPUs initially assigned to the new OS is determined
20 based on the number of CPUs required for the new OS to run effectively taking into account the anticipated application program load.

- In the second phase, while the new OS is booting-up and running using two of the four CPUs 107, 109, the remaining CPUs 111, 113 start to save the
25 remaining OSM to the disk drive 105. The system memory freed by the CPUs 111, 113 is progressively made available to the new OS thereby gradually increasing the size of the current system memory. Similarly, once a given CPU has completed the dumping of the data from its allocated part of the OSM, the CPU is joined to the resources of the new OS. Once all of the OSM has been
30 dumped, the new OS can use all of the CPUs 107, 109, 111, 113.

- This two phase approach reduces the downtime of the computer system compared to a conventional dumping system. Assuming that the OS boot time does not change significantly in either case, the savings accomplished with the
35 present approach are:

(dump time for OSM remainder)/ (dump time for total system memory)

Figure 2a is a view of the computer system 101 at the start of the second phase of a system memory dump as described above where 1 gigabyte (GB) (1%) of the old system memory has been dumped to the disk 105 (shown shaded). The first two CPUs 109, 111 (shown not shaded) are loading the OS from the disk 105 and restarting the OS using the 1GB of the RAM 117 (shown not shaded) freed during the first phase of the reboot process as described above. The third and fourth CPUs 111, 113 (shown shaded) are occupied in dumping the remainder of the old system data (shown shaded) from the RAM 117 to the disk 105. Each of the CPUs 111, 113 are allocated a 49.5GB portion of the data from the old system memory to dump to the disk 105.

When either of the third or fourth CPUs 111, 113 have completed the dumping of their allocated portions of the old system memory data they join the first and second CPUs 107, 109 in providing the OS. Figure 2b shows the computer system 101 when the third CPU 111 (now not shaded) has completed its dump allocation and is now providing the OS. The fourth CPU 113 is still in the process of dumping its allocation having completed 14.5GB. As a result, the system memory 117 still contains 35GB of OSM and the disk has 65GB of dumped data. Once the fourth CPU 113 has completed its allocated dump, all four processors will revert to providing the OS.

The processing carried out by the firmware 115 and the CPUs 107, 109, 111, 113 for restarting the computer system 101 after an OS fault will now be described with reference to the flow chart of figure 3. At step 301 a fatal OS fault is detected, the OS ceases operation and processing moves to step 303. At step 303, the first portion G bytes of the system memory for the failed OS is saved to disk and the remaining D bytes are designated as OSM and processing moves to step 305. At a step 305 the firmware resets the CPUs and other hardware and initiates the rebooting process. The reset is non-destructive in that the contents of the system memory are not reset or erased. At step 307 the OS is running using the G bytes of memory that were freed in step 303 and at step 309 the firmware allocates N CPUs to run the OS and the remaining M CPUs to complete the OSM dumping process. The processing is then split

between the CPUs running the OS which continue to step 311 and the CPUs completing the dump operation which continue to step 313.

At step 313, the firmware initiates the dumping process by allocating blocks of the OSM to each of the M CPUs. For the first CPU, processing then moves to step 315 where the CPU dumps L bytes (where $L=D/M$) of OSM to the disk and once this is complete processing moves to step 317 where the firmware returns the CPU and the freed memory for use by the new OS. For the remaining M-1 CPUs, the processing of steps 315 and 317 is duplicated as shown by steps 321 and 323 for each remaining CPU (indicated by the dotted process step outlines).

At step 311 the N CPUs provide the OS using the G bytes of memory plus the memory freed as each of the M CPUs completes portions of its allocated dumping. In addition, as each of the M CPUs completes its allocated dumping, those CPUs join the N CPUs in running the OS. Thus at step 319, the number of CPUs running the new OS converges to M+N and the new system memory increases to G+D bytes.

In an alternative embodiment, the computer system is as described above in figure 1 but has a single CPU and an OS capable of multiple processing threads. The dumping process carried out by this system will now be described with reference to figure 4 in which at step 401 a fatal OS fault is detected, the OS ceases operation and processing moves to step 403. At step 403, the first portion G bytes of the system memory for the failed OS is saved to disk and the remaining D bytes are designated as read only OSM and processing moves to step 405. At a step 405 the firmware resets the CPU and other hardware and initiates the rebooting process. The reset is non-destructive in that the contents of the system memory are not reset or erased. At step 407 the OS is running using the G bytes of memory that were freed in step 403. The OS then initiates a number (N) of processing threads and moves to step 409 and provides the OS using the G bytes of memory. The N threads move to step 411 where each thread is allocated L bytes (where $L=D/N$) of the OSM for dumping. The number of threads to use for this purpose is computed based on the I/O bandwidth available to new OS and the amount of OSM to dump.

- At step 413, the first thread divides its allocated L bytes of OSM into K chunks and processing moves to step 415. At step 415 J bytes (where $J=L/K$) of the allocated OSM is dumped and the pages freed are marked as normal by the memory management subsystem of the OS making that memory available to the OS as well as applications. Processing then moves to step 417 where a check is carried out to determine if all K chunks have been dumped and if not processing returns to step 415 as described above. If, however, all K chunks have been dumped then at step 419, the thread is terminated.
- Each of the other N threads carry out the same processing steps for their allocated L bytes of OSM as shown in steps 421, 423, 425 and 427 (the steps shown in dotted lines each representing the remaining processing threads). As each thread progressively returns freed memory to the OS in step 409, the memory available to the OS steadily increases to full capacity at step 429 when all threads carrying out the dumping process have terminated.

In a further embodiment the computer system has a single CPU and a multithreading OS and instead of incrementally adding memory in chunks as described in step 415 above, the OS waits for the dumping threads to complete the whole dump and then accepts the complete OSM as normal useable memory.

In a yet further embodiment, the computer system is a multiple CPU system as described above for figure 1 which, in addition, has an OS which is capable of multithreaded processing.

In another embodiment, the system administrator can configure the number of threads or number of CPUs allocated to the dumping process depending on the anticipated application load on the system. This step can be carried out as a manual step in the boot-up procedure.

As described in the embodiments above, only the first portion or pages of the OSM need to be examined initially. These first pages are saved to the dump device such as a disk and the computer system is immediately allowed to transfer control back to firmware and the process of booting the new OS can start. In a computer system with a large system memory, only a small portion of

the entire system memory is needed for those first pages. The larger the system memory, the shorter the system down time and the greater the system availability. In other words, the restart can begin before the dumping process is complete and the OS can run effectively in parallel with the remainder of the boot operation. Furthermore, the arrangements described above can be used with either selective or non-selective dumping algorithms in either the first and/or the second phase of the process.

In the embodiments above, in the first phase, the first portion of the data from the OSM comprises 1% of the system memory. This is a typical arrangement for a computer system with a large system memory of tens of gigabytes. The first portion is determined as the amount of memory required for the kernel/OS to load and create the necessary data structures to be able to detect and configure all I/O devices and to execute multiple kernel threads that perform the dumping task. In the above embodiments, where the firmware performs the dumping task, the kernel will not require memory to execute dumping threads, however, it may have to provide additional memory that will be incrementally added into the system as each of the firmware owned CPUs perform their dumping task and make the memory available for the OS's use. For a given system configuration, based on the data available with the OS before it failed, the size of the first portion of OSM can be calculated at set-up and may include a safety margin to ensure that the next OS does not fail to boot-up owing to any minor oversight in the calculation by the current instance of the OS. In computer systems with smaller system memories, the size of the first portion will be a correspondingly larger percentage of the total memory.

It will be understood by those skilled in the art that the apparatus that embodies a part or all of the present invention may be a general purpose device having software arranged to provide a part or all of an embodiment of the invention. The device could be single device or a group of devices and the software could be a single program or a set of programs. Furthermore, any or all of the software used to implement the invention can be communicated via various transmission or storage means such as computer network, floppy disc, CD-ROM or magnetic tape so that the software can be loaded onto one or more devices.

While the present invention has been illustrated by the description of the embodiments thereof, and while the embodiments have been described in considerable detail, it is not the intention of the applicant to restrict or in any way limit the scope of the appended claims to such detail. Additional advantages
5 and modifications will readily appear to those skilled in the art. Therefore, the invention in its broader aspects is not limited to the specific details representative apparatus and method, and illustrative examples shown and described. Accordingly, departures may be made from such details without departure from the spirit or scope of applicant's general inventive concept.

Claims

1. A method for storing data in a computer system, the method comprising the steps of:
 - a) storing data from a first portion of a system memory in a secondary memory;
 - 5 b) allocating said first portion of said system memory for subsequent use by an operating system (OS);
 - c) rebooting and running said OS using said allocated memory; and
 - d) storing data from a remaining portion of said system memory in said secondary memory.
- 10 2. A method according to claim 1 in which said running of said OS and step d) are carried out concurrently.
3. A method according to any preceding claim in which step a) is carried out by
15 firmware prior to steps b) to d).
4. A method according to any preceding claim in which step d) is carried out under the control of firmware.
- 20 5. A method according to claim 3 in which step d) is carried out by one or more processing threads of said OS.
6. A method according to claim 5 in which each thread is allocated a part of said remaining portion of system memory and when said data from said part is
25 stored in said secondary storage, said thread quits.
7. A method according to any preceding claim in which the computer system comprises a plurality of CPUs and the processing of step d) is allocated between said plurality of CPUs.
- 30 8. A method according to claim 7 in which step d) is carried out by a subset of said plurality of CPUs.
9. A method according to claim 7 or 8 in which each CPU is allocated a part of
35 said remaining portion of system memory and when said data from said part is stored in said secondary storage, each said CPU reverts to providing said OS.

10. A method according to any preceding claim further comprising the step of:
e) allocating memory freed in step d) for use by said OS.
- 5 11. A method according to claim 10 in which in step d) said storing is carried out in predetermined blocks of said memory and in step e) as each said block is freed it is allocated for use by said OS.
12. A method according to any preceding claim in which in step a) the data is
10 kernel data which is swapped out of the dump image.
13. A method according to any preceding claim in which in step a) said first portion of said system memory is the minimum amount required to run the OS.
- 15 14. A method according to any preceding claim in which in step a) said first portion comprises up to 1% of said system memory.
15. A method according to any preceding claim in which step a) is carried out as part of a reboot of the computer system.
- 20 16. A method according to claim 15 in which said reboot is carried out in response to an OS failure.
17. A method according to claim 16 in which said reboot operation is arranged
25 to operate automatically in response to said OS failure.
18. A method according to any preceding claim in which step a) is carried out as part of a back up operation for said system memory.
- 30 19. Apparatus for storing data in a computer system, the apparatus comprising:
processing means operable to store data from a first portion of a system memory in a secondary memory;
a memory management system operable to allocate said first portion of said system memory for subsequent use by an operating system (OS); and

said processing means being further operable to reboot and run said OS using said allocated memory and to store data from a remaining portion of said system memory in said secondary memory.

- 5 20. Apparatus according to claim 19 in which said processing means is further operable to run said OS concurrently with said storage of said data from said remaining portion of said system memory.
- 10 21. Apparatus according to any of claims 19 or 20 in which storage of data from the first portion of said system memory is carried out by firmware prior to said rebooting of said OS.
- 15 22. Apparatus according to any of claims 19 to 21 in which the storage of said data from said remaining portion of said system memory is carried out under the control of firmware.
- 20 23. Apparatus according to claim 21 in which the storage of said data from said remaining portion of said system memory is carried out by one or more processing threads of the operating system (OS).
- 25 24. Apparatus according to claim 23 in which each thread is allocated a part of said remaining portion of system memory and when said data from said part is stored in said secondary storage, said thread quits.
- 30 25. Apparatus according to any of claims 19 to 24 in which the computer system comprises a plurality of CPUs and the storage of said data from said remaining portion of said system memory is allocated between said plurality of CPUs.
- 35 26. Apparatus according to claim 25 in which the storage of said data from said remaining portion of said system memory is allocated to a subset of said plurality of CPUs.
27. Apparatus according to claim 25 or 26 in which each CPU is allocated a part of said remaining portion of system memory and when said data from said

part is stored in said secondary storage, each said CPU reverts to providing said OS.

28. Apparatus according to any of claims 19 to 27 in which the memory
5 management system is further operable to allocate said memory freed by said storage of said data from said remaining portion of said system memory is allocated for use by said OS.

29. Apparatus according to claim 28 in which said storage of said data from
10 said remaining portion of said system memory is carried out in predetermined blocks of said memory and as each said block is freed it is allocated for use by said OS.

30. Apparatus according to any of claims 19 to 29 in which said first portion of
15 said system memory is the minimum amount required to run the OS.

31. Apparatus according to any of claims 19 to 30 in which said first portion comprises up to 1% of said system memory.

20 32. A computer program or group of programs arranged to enable a computer or group of computers to carry out the method of claim 1.

33. A computer program or group of programs arranged to enable a computer or group of computers to provide the apparatus of claim 19.

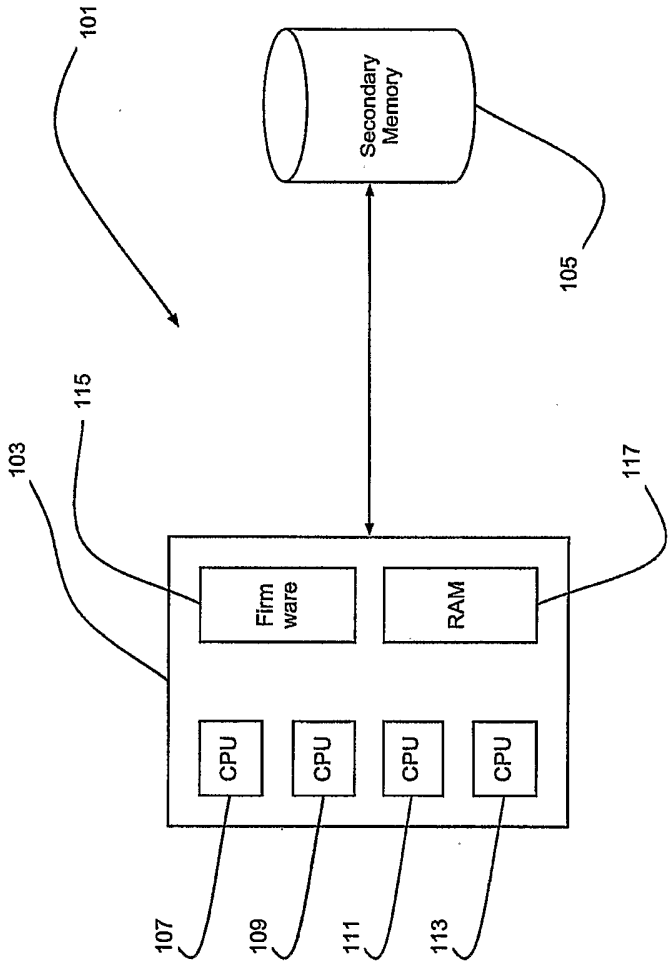


Figure 1

2/4

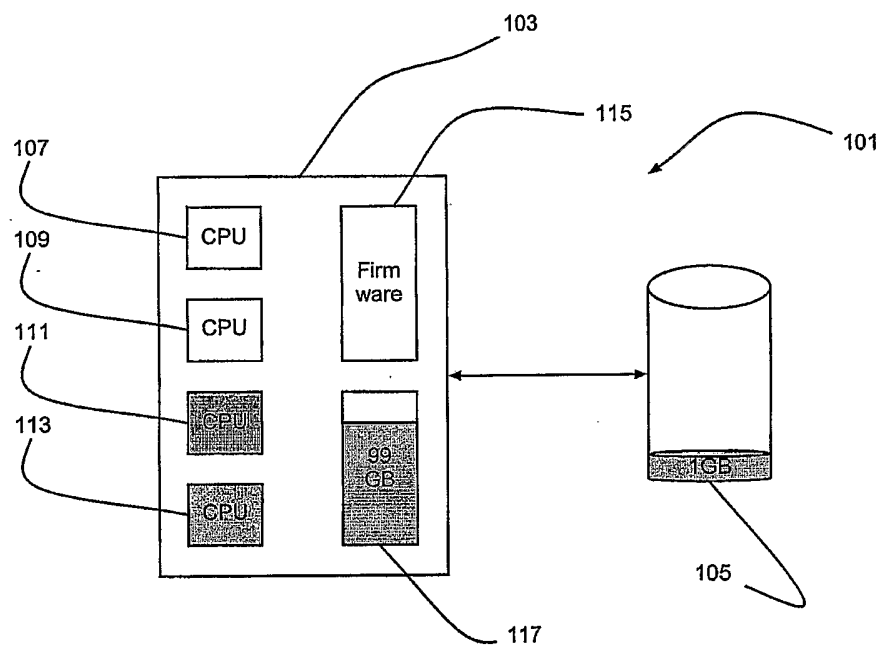


Figure 2a

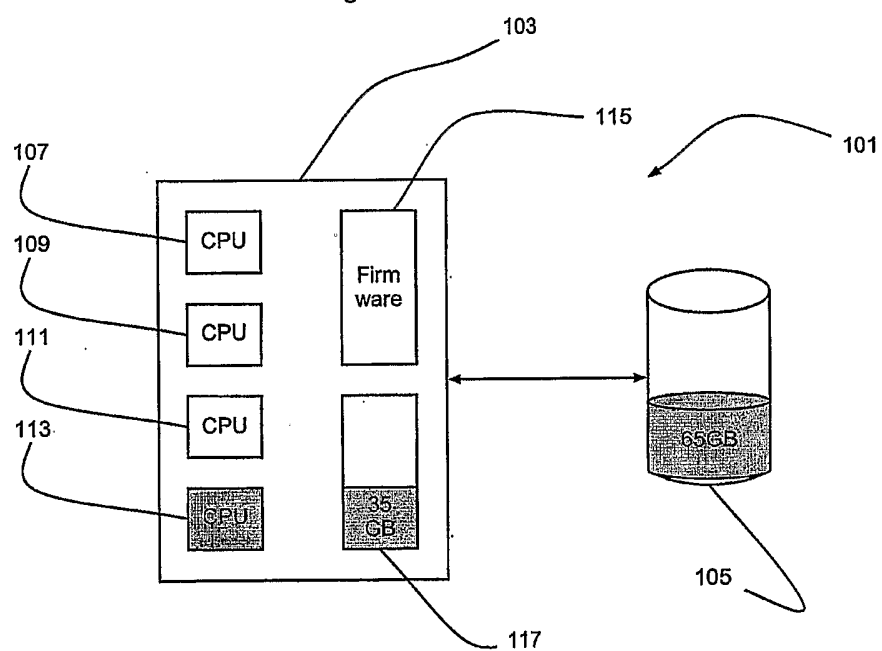


Figure 2b

3/4

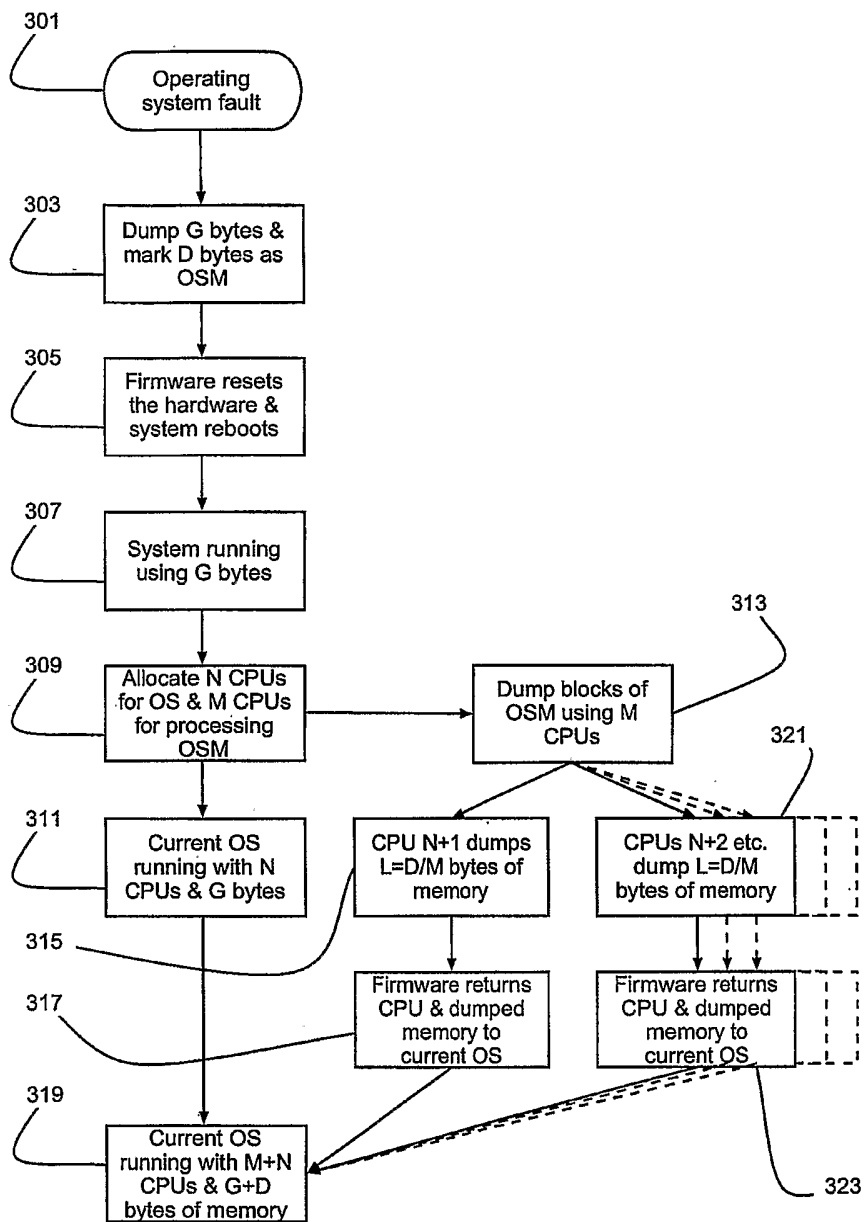


Figure 3

4/4

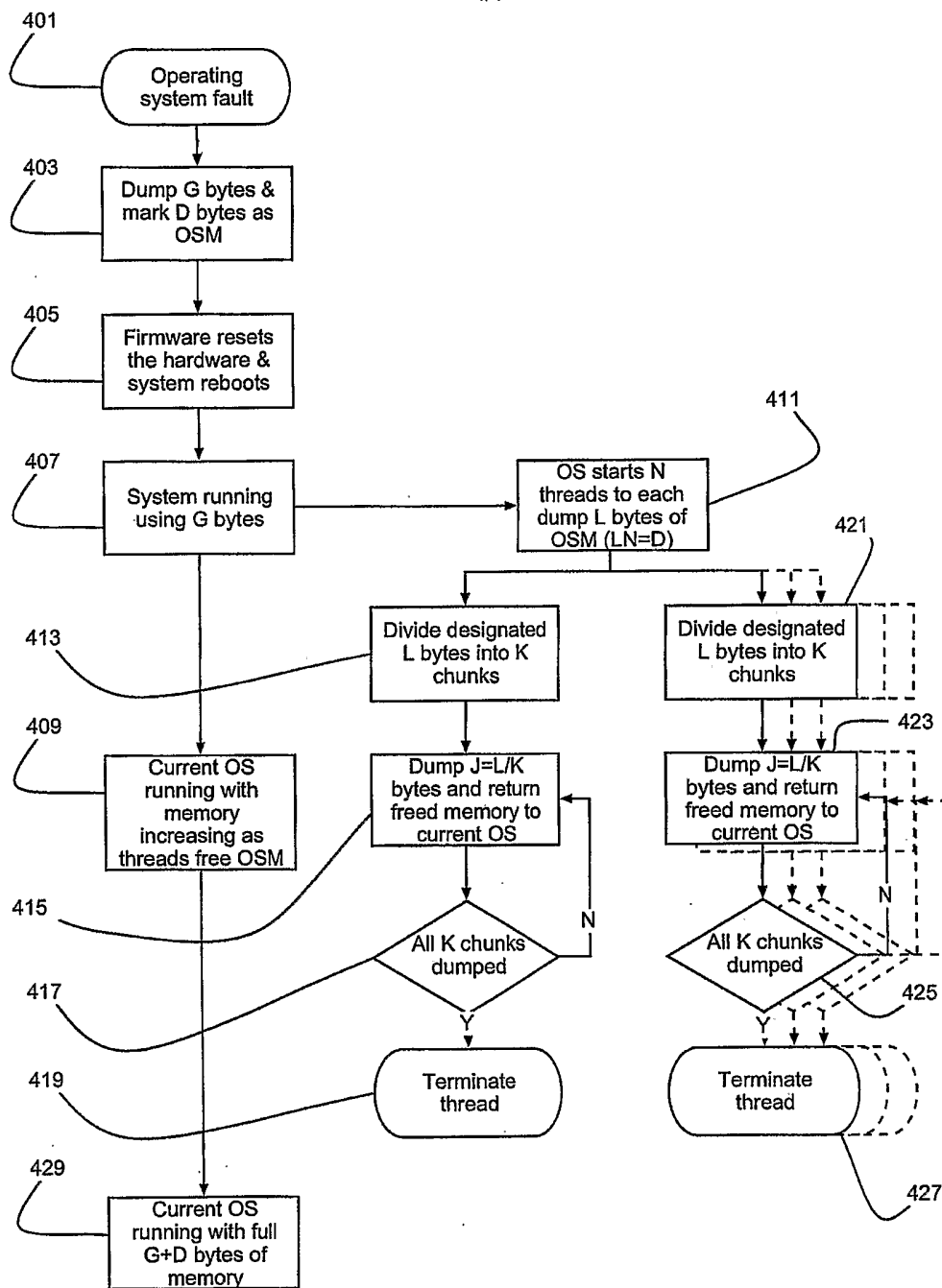


Figure 4

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IN05/00062

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : g06f 12/00

US CL : 711/002

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 711/002(text search only)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 6,209,088 (RENERIS) 27 March 2001, whole document	1-2,5-6,19-21,23-24,32-33
Y	US Pub 2004/0019891 (KOENEN) 29 January 2004, whole document	3-6,21
Y	US 6,681,348 (VACHON) 20 January 2004, whole document	1-3,5-6,1-21,23-24,32-33

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"B" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

19 August 2005 (19.08.2005)

Date of mailing of the international search report

29 SEP 2005

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Facsimile No. (571) 273-8300

Authorized officer

Mano Padmanabhan

Telephone No. 571-272-4171

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IN05/00062

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. ☒ Claims Nos.: 4,7-18,22,25-31
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐
☐

The additional search fees were accompanied by the applicant's protest.

No protest accompanied the payment of additional search fees.