(12) STANDARD PATENT (11) Application No. **AU 2006331830 B2**
(19) AUSTRALIAN PATENT OFFICE

(54) Title
Optimizing write and wear performance for a memory

(51) International Patent Classification(s)
G06F 12/02 (2006.01) ***G06F 12/08*** (2006.01)

(21) Application No: **2006331830** (22) Date of Filing: **2006.12.18**

(87) WIPO No: **WO07/075668**

(30) Priority Data

(31) Number	(32) Date	(33) Country
11/303,382	**2005.12.16**	**US**

(43) Publication Date: **2007.07.05**
(44) Accepted Journal Date: **2011.06.02**

(71) Applicant(s)
Microsoft Corporation

(72) Inventor(s)
Reinauer, Robert. L.;Fortin, Michael. R.

(74) Agent / Attorney
Davies Collison Cave, 1 Nicholson Street, Melbourne, VIC, 3000

(56) Related Art
US 6,275,436 B1 (TOBITA et al.) 14 August 2001
US 2003/0202377 A1 (HARARI et al.) 30 October 2003
US 6,026,027 A (TERRELL, II et al) 15 February 2000

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 July 2007 (05.07.2007)

PCT

(10) International Publication Number
WO 2007/075668 A1

(51) International Patent Classification:
G06F 12/02 (2006.01) *G06F 12/08* (2006.01)

(21) International Application Number:
PCT/US2006/048402

(22) International Filing Date:
18 December 2006 (18.12.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/303,382 16 December 2005 (16.12.2005) US

(71) Applicant *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **FORTIN, Michael., R.**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **REINAUER, Robert., L.**; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(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, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

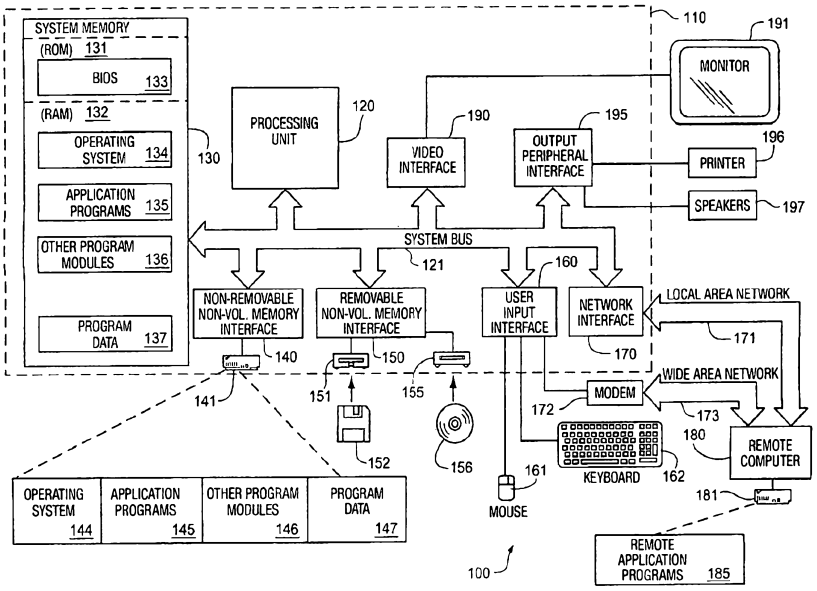
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, 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).

Declarations under Rule 4.17:
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

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: OPTIMIZING WRITE AND WEAR PERFORMANCE FOR A MEMORY



(57) Abstract: Determining and using the ideal size of memory to be transferred from high speed memory to a low speed memory may result in speedier saves to the low speed memory and a longer life for the low speed memory.

WO 2007/075668 A1

Background

[0001] Memory device have many trade offs. Memory which is fast may be expensive. Memory which is slow may be inexpensive. As the memory demands of modern applications exceed the physical capacity of customer systems, paging operations to the hard drive quickly make the system and applications appear slow and unresponsive, particularly when switching between tasks or users. In an ideal computer, all the memory would be as fast as possible, but such a computer would be too expensive to be realistic. In addition, there are some applications where cheap, long term storage is all that is needed. As a result, computers have a variety of memory from ultra fast memory close to the processor to slower memory such as disk based memory. In addition, computers have the ability to add more and different types of memory such as portable hard drives, floppy disks and flash memory devices. Each of these devices have performance and wear characteristics which may be maximized. It is desired to address one or more shortcomings of the prior art, or at least provide a useful alternative.

Summary

[0002] The present invention provides a computer system comprising:

a processor;

an input output device

a first memory device;

a second memory device having a first data storage speed, the second memory device being a rotatable disk based storage device;

a third memory device having a second data storage speed that is relatively slow when compared to the first data storage speed, the third memory device comprising a peripheral device that is external to and adapted to be plugged into the computer system;

a set of instructions stored in the first memory device and adapted to be executed by the processor when data is received by the computer system that is to be stored on the third memory device, the set of instructions, including instructions for:

storing the data in the second memory device,

testing the third memory device to determine preferred erase and preferred write size for the third memory device,

collecting additional data in the second memory device until a preferred amount of data is stored that is destined for the third memory device,

storing the data in the second memory device until it is of a size that is appropriate write size for the third memory device,

analyzing the data in the second memory device to determine if any of the data has been replaced with newer data before the data is written to the low speed third memory device, and

once an appropriate write size of data for the third memory device is stored in the higher speed second memory device, waiting until a period of low system activity to write the data to the lower speed third memory device.

Drawings

[0003] Preferred embodiments of the present invention are hereinafter described, by way of example only, with reference to the accompanying drawings, wherein:

[0004] Fig. 1 is a block diagram of a computing system that may operate in accordance with an embodiment of the invention; and

[0005] Fig. 2 is a diagram of a method in accordance with an embodiment of the invention.

Description

[0006] Although the following text sets forth a detailed description of numerous different embodiments, it should be understood that the legal scope of the monopoly is defined by the words of the claims set forth at the end of this patent. The detailed description is to be construed as exemplary only and does not describe every possible embodiment since describing every possible embodiment would be impractical, if not impossible. Numerous alternative embodiments could be implemented, using either current technology or technology developed after the filing date of this patent, which would still fall within the scope of the claims.

[0007] It should also be understood that, unless a term is expressly defined in this patent using the sentence "As used herein, the term '_____' is hereby defined to mean..." or a similar sentence, there is no intent to limit the meaning of that term, either expressly or by implication, beyond its plain or ordinary meaning, and such term should not be interpreted to be limited in scope based on any statement made in any section of this patent (other than the language of the claims). To the extent that any term recited in the claims at the end of this patent is referred to in this patent in a manner consistent with a single meaning, that is done for sake of clarity only so as to not confuse the reader, and it is not intended that such claim term by limited, by implication or otherwise, to that single meaning. Finally, unless a claim element is defined by reciting the word "means" and a function without the recital of any structure, it is not intended that the scope of any claim element be interpreted based on the application of 35 U.S.C. § 112, sixth paragraph.

[0008] Determining and using the ideal size of memory to be transferred from high speed memory to a low speed memory may result in speedier saves to the low speed memory and a longer life for the low speed memory. For example, some flash memory devices have memory write sizes that minimize the number of erases and writes to the flash memory. Other low speed memory devices may have ideal write sizes that enable large blocks on the low speed memory to store contiguous data. In addition, data may be accessed in fast memory even thought the data has been indicated to be sent to lower speed memory.

[0009] Fig. 1 illustrates an example of a suitable computing system environment 100 on which a system for the steps of the claimed method and apparatus may be implemented. The computing system environment 100 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the method of apparatus of the claims. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 100.

[0009a] The steps of the claimed method and apparatus are operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the methods or apparatus of the claims include, but are not limited to, personal computers, server computers, hand-held or laptop

devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

[0009b] The steps of the claimed method and apparatus may be described in the general context of computer-executable instructions, such as program modules, being executed by a

computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The methods and apparatus may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

[0010] With reference to Fig. 1, an exemplary system for implementing the steps of the claimed method and apparatus includes a general purpose computing device in the form of a computer 110. Components of computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

[0011] Computer 110 typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer 110 and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can accessed by computer 110. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode

information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer readable media.

[0012] The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 110, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, Fig. 1 illustrates operating system 134, application programs 135, other program modules 136, and program data 137.

[0013] The computer 110 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, Fig. 1 illustrates a hard disk drive 140 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156 such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is typically connected to the system bus 121 through a non-removable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

[0014] The drives and their associated computer storage media discussed above and illustrated in Fig. 1, provide storage of computer readable instructions, data structures, program modules and other data for the computer 110. In Fig. 1, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146, and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers here to illustrate that, at a

minimum, they are different copies. A user may enter commands and information into the computer 20 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 via an interface, such as a video interface 190. In addition to the monitor, computers may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through an output peripheral interface 190.

[0015] The computer 110 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 110, although only a memory storage device 181 has been illustrated in Fig. 1. The logical connections depicted in Fig. 1 include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

[0016] When used in a LAN networking environment, the computer 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, Fig. 1 illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0017] Fig. 2 may illustrate a method of efficiently storing data where there is a plurality of data storage devices with a hierarchy of data storage speeds. Memory device have many

trade offs. Memory which is fast may be expensive. Memory which is slow may be inexpensive. In an ideal computer, all the memory would be as fast as possible, but such a computer would be too expensive for a majority of uses. In addition, there are some applications where cheap, long term storage is all that is needed. As a result, computers have a variety of memory from ultra fast memory close to the processor to slower memory such as disk based memory. In addition, computers have the ability to add more and different types of memory such as portable hard drives 140 and flash memory devices 152. Each of these devices have performance and wear characteristics which may be maximized.

[0018] For example, flash memory 152 is made in two common forms: NOR flash and NAND flash. Other forms of flash memory 152 are also contemplated. Both types of flash memory 152 and EEPROM wear out after many erase operations, due to wear on the insulating oxide layer around the charge storage mechanism used to store data. A typical NOR flash memory 152 unit may wear out after 10,000-100,000 erase/write operations, a typical NAND flash 152 memory may wear out after 1,000,000. One limitation of flash memory 152 is that, although it can be read or programmed a byte or a word at a time in a random access fashion, it must be erased a "block" at a time. Starting with a freshly erased block, any byte within that block can be programmed. However, once a byte has been programmed, it cannot be changed again until the entire block is erased. When compared to a hard disk drive 140, flash memory 152 is limited in that it has a finite number of erase-write cycles, so that care has to be taken to minimize the number of erase/write cycles.

[0019] In other devices, the slowness of the write cycle may be a factor in determining a preferred write size. For example, if there is significant overhead time required to set up a write function, time could be saved by minimizing the number of writes by maximizing the size of the data to be written. Similarly, in some storage devices, repeated small saves can result in data being spread out across the storage device, which may result in longer read times than if the data was stored contiguously as may be the result of fewer, larger saves. In addition, the write cycle may be taxing on the processor and during times of high processor usage, it may make sense to save less frequently with larger storage sizes at less frequent intervals.

[0020] Referring again to hard drives 140, if there were no high speed memory, every write to the hard disk 140 may involve a performance hit while the system waits for the hard disk 140 to access the correct location on the hard disk 140 and write the data. This may take

at least 10 milliseconds on most drives, which is a long time in the computer world and may really slow down performance as the system waits for the hard disk.

[0021] When high speed memory is available such as RAM 132 (also referred to as write caching enabled), when the system sends a write to the hard disk 140, the logic circuit records the write in its much faster cache or high speed memory 132, and then immediately sends back an acknowledgement to the operating system saying, in essence, "finished." The rest of the system can then proceed on without having to wait for the actuator to position and the disk to spin, and so on which improves performance. However, the drive 140 sends back saying "finished" when it really isn't finished--the data isn't on the disk at all, it's only in the cache 132. The hard disk's 140 logic circuits begin to write the data to the disk, but of course this takes some time.

[0022] Typical PC 110 use involves loading programs, and loading and saving data frequently. All of these operations may require access to the hard disk 140. And therefore, hard disk 140 performance becomes an issue. Making the CPU 120 fast enough to process two million instructions while it waits doesn't really gain a user much unless it has something to do with that time. The applications where hard disk 140 performance issues are most important are those that do a lot of reading and writing to the hard disk 140, instead of doing a lot of processing, such as multimedia editing applications, especially those dealing with large audio and video files.

[0023] Access time is the metric that represents the composite of all the other specifications reflecting random performance positioning in the hard disk 140. The most common definition is:

[0024] Access Time = Command Overhead Time + Seek Time + Settle Time + Latency

[0025] Command overhead may refer to the time that elapses from when a command is given to the hard disk 140 until something actually starts happening to fulfill the command. The seek time of a hard disk 140 may measure the amount of time required for the read/write heads to move between tracks over the surfaces of the platters. Switching between tracks requires the head actuator to move the head arms physically, which being a mechanical process, takes a specific amount of time. The amount of time required to switch between two tracks depends on the distance between the tracks. However, there is a certain amount of "overhead" involved in track switching, so the relationship is not linear. It may not take double the time to switch from track 1 to track 3 that it does to switch from track 1 to track 2.

[0026] Seek time is normally expressed in milliseconds (commonly abbreviated "msec" or "ms"). Of course, in the modern PC 110, a millisecond is an enormous amount of time as high speed system memory such as RAM 132 may have speed measured in nanoseconds, for example (one million times smaller). A 1 GHz processor may (theoretically) execute over one million instructions in a millisecond. Even small reductions in seek times can result in improvements in overall system performance, because the rest of the system is often waiting for the hard disk 140.

[0027] The hard disk 140 platters are spinning around at high speed, and the spin speed may not be synchronized to the process that moves the read/write heads to the correct cylinder on a random access on the hard disk 140. Therefore, at the time that the heads arrive at the correct cylinder, the actual sector that is needed may be anywhere. After the actuator assembly has completed its seek to the correct track, the drive 140 may wait for the correct sector to come around to where the read/write heads are located. This time is called latency. Latency is directly related to the spindle speed of the drive and such is influenced solely by the drive's spindle characteristics. Accordingly, any actions that can reduce access time may have a large impact on system performance.

[0028] The amount of available RAM 132 may also have an effect on the size of data that is capable of being stored in high speed memory. For example, if a system only has 8k of high speed RAM 132, more frequent write operations will have to occur than if the system had 512kb of high speed RAM 132 simply because the RAM 132 cannot store much data.

[0029] At block 200, the method may receive data to be stored on a lower speed data storage device. For example, a word processing file may be designated to be stored on a hard drive 140. At block 205, the data may be stored in a high speed memory. For example, the word processing file may be stored in a cache in RAM 132. At block 210, the method may collect additional data in the high speed memory 132 until a preferred amount of data is stored that is destined for a particular lower speed data device such as the hard drive 140. At block 215, the method may store the data in high speed memory 132 until it is of a size that is appropriate write size for a particular storage device. As described previously, the preferred amount of data may vary by device. For example, the preferred write size may be a multiple of 2 raised to a power such as is one of 128 kb, 256 kb, and 512 kb. In the flash memory example, the size may be the size of a flash memory block such that erase/write operations will be minimize by only writing to the flash device when an entire flash block is ready to be stored (rather than multiple erase/write operations). At block 220, once an appropriate write

size of data for the particular device is stored in the high speed memory device 132, the data may be written to the lower speed device 140.

[0030] At block 225, the method may keep the data in high speed storage 132 until a determination is made that the data no longer is needed in high speed storage 132. For example, if a system has excess RAM 132, it may not tax the system to keep data that has already been written to low speed memory 140 also in high speed memory 132. As the data has already been stored in low speed memory 140, it may not be needed in high speed memory 132, but it may be kept there anyway as a convenience to the user.

[0031] At block 230, the method of claim 1, wherein the data to be written to the low speed storage device 140 is backed up on an additional low speed storage device 140. In some instances, a user or a program may instruct that data be stored on a low speed storage device 140. The method may wait for an optimal amount of data before writing to the low speed memory. In the mean time, the low speed memory such as a flash memory unit 152 may be removed from the system. In these cases, it may be beneficial to have a back-up of the data in an additional low speed storage device. The data may be moved from the high speed memory 132 to an additional memory that is lower speed that the high speed memory 132 but higher speed than the low speed memory. At block 235, the method may warn a user that the data has yet to be written to the low speed memory device such as the flash 152 when a low speed memory device has been removed from the system before the data has been written from the high speed memory 132 to the low speed memory device 152. At block 240, the user may be presented the option to write the data from the high speed memory 132 to the low speed memory 152 before the low speed memory device is removed. At block 245, the method may wait until a period of low system activity to write to the low speed storage device.

[0032] As an example, the user or method may indicate that a word processing file is to be stored on a flash memory 152. The method may wait for enough data to match the optimal write size for flash memory 152 so the data may not be written to the flash memory 152 immediately if the size of the word processing file is less than the size of the optimal write size. If the flash is removed 152, the user may be warned that the data has yet to be written to the flash memory 152. In addition, the data may be already backed up in a "medium" speed storage device and this backup may be used to keep the data from being overwritten while in the high speed storage device 132.

[0033] At block 250, the method may organize the data in the high speed memory 132 such that the write to the low speed memory 140 is more efficient. The organization can be as simple as rearranging the data in the high speed memory 132 to be contiguous such that a transfer to the low speed memory 140 may proceed more efficiently.

[0034] At block 255, the method may analyze the data in the high speed memory 132 to determine if any of the data has been replaced with newer data before the data is written to the low speed memory 140. At block 260, the method may analyze the data in the high speed memory 132 to determine whether any of the data to be written to the low speed memory 140 is duplicate data wherein the older duplicate data is not written to the low speed memory 140 device. At block 265, the method may invalidate blocks in the high speed memory 132 when write operations impact those blocks before they have been committed to the low speed device 140.

[0035] At block 270, the method may keep an index of data in high speed memory 132 that is to be written to low speed memory 140 and if the data written to be written to the low speed device 140 is resident in the high speed memory 132, the method may serve read requests for the data from the high speed memory 132 without regard as to whether the data has been written to low speed memory 140.

[0036] At block 275, the method may test the low speed memory 140 to determine the preferred write size for the low speed memory 140. For example, the method may inquire of a flash memory 152 what is the ideal erase/write size. At block 280, the method may also use indication from the low speed storage device 140 to calculate the preferred write size. For example, in the flash memory 152 example, the method may determine that blocks are erased in 1024 byte increments so the method may determine that save operations should occur in 1024 byte increments. The preferred write size may be related to the preferred erase size of the low speed memory 152.

[0037] Although the forgoing text sets forth a detailed description of numerous different embodiments, it should be understood that the scope of the patent is defined by the words of the claims set forth at the end of this patent. The detailed description is to be construed as exemplary only and does not describe every possible embodiment because describing every possible embodiment would be impractical, if not impossible. Numerous alternative embodiments could be implemented, using either current technology or technology

developed after the filing date of this patent, which would still fall within the scope of the claims.

[0038] Thus, many modifications and variations may be made in the techniques and structures described and illustrated herein without departing from the scope of the present claims. Accordingly, it should be understood that the methods and apparatus described herein are illustrative only and are not limiting upon the scope of the claims.

[0039] Throughout this specification and the claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" and "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

[0040] The reference in this specification to any prior publication (or information derived from it), or to any matter which is known, is not, and should not be taken as an acknowledgment or admission or any form of suggestion that that prior publication (or information derived from it) or known matter forms part of the common general knowledge in the field of endeavour to which this specification relates.

THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

1. A computer system comprising:

a processor;

an input output device

a first memory device;

a second memory device having a first data storage speed, the second memory device being a rotatable disk based storage device;

a third memory device having a second data storage speed that is relatively slow when compared to the first data storage speed, the third memory device comprising a peripheral device that is external to and adapted to be plugged into the computer system;

a set of instructions stored in the first memory device and adapted to be executed by the processor when data is received by the computer system that is to be stored on the third memory device, the set of instructions, including instructions for:

storing the data in the second memory device,

testing the third memory device to determine preferred erase and preferred write size for the third memory device,

collecting additional data in the second memory device until a preferred amount of data is stored that is destined for the third memory device,

storing the data in the second memory device until it is of a size that is appropriate write size for the third memory device,

analyzing the data in the second memory device to determine if any of the data has been replaced with newer data before the data is written to the low speed third memory device, and

once an appropriate write size of data for the third memory device is stored in the higher speed second memory device, waiting until a period of low system activity to write the data to the lower speed third memory device.

2. The computer system of claim 1, wherein, when the processor executes the set of instructions stored in the first memory device, an index of data is kept in the higher speed second memory device that is to be written to the lower speed third memory device and, if the data written to be written to the lower speed third memory device is resident in the higher speed second memory device, read requests for the data from the higher speed second memory device are served without regard as to whether the data has been written to the lower speed third memory device.

3. The computer system of claim 1, wherein the data to be written to the lower speed third memory device is backed up on an additional low speed storage device.

4. The computer system of claim 1, wherein the lower speed third memory device is a flash storage device.

5. The computer system of claim 4, wherein the write size is the erase block size of the flash.

6. The computer system of claim 1, wherein, when the set of instructions is executed by the processor, a user is warned that the data has yet to be written to the lower speed third memory device when the lower speed memory device has been removed from the computer system before the data has been written from the higher speed second memory device to the lower speed third memory device.

7. The computer system of claim 6, wherein, when the set of instructions is executed, a user is presented the option to write the data from the higher speed second memory device to the lower speed third memory device before the lower speed memory device is removed.

8. The computer system of claim 1, wherein, when the set of instructions is executed, the data in the higher speed second memory device is organized such that the write to the

2006331830 19 Apr 2011

lower speed third memory device is more efficient.

9. The computer system of claim 1, wherein, when the set of instructions is executed, data in the higher speed second memory device is analyzed to determine if any of the data has been replaced with newer data before the data is written to the lower speed third memory device.

10. The computer system of claim 1, wherein, when the set of instructions is executed by the processor, the data in the higher speed second memory device is analyzed to determine whether any of the data to be written to the lower speed third memory device is duplicate data wherein the older duplicate data is not written to the lower speed third memory device.

11. The computer system of claim 1, wherein, when the set of instructions is executed by the processor, invalidating blocks in the higher speed second memory device are invalidated when write operations impact those blocks before they have been committed to the lower speed third memory device.

12. The computer system of claim 1, wherein, when the set of instructions is executed by the processor, the data is moved from the higher speed second memory device to an additional memory device that has a lower speed than the higher speed second memory device but has a higher speed than the lower speed third memory device.

13. The computer system of claim 1, wherein the rotatable disk based storage device comprises a hard disk drive.

14. The computer system of claim 1, wherein the rotatable disk based storage device includes at least one rotatable storage disk.

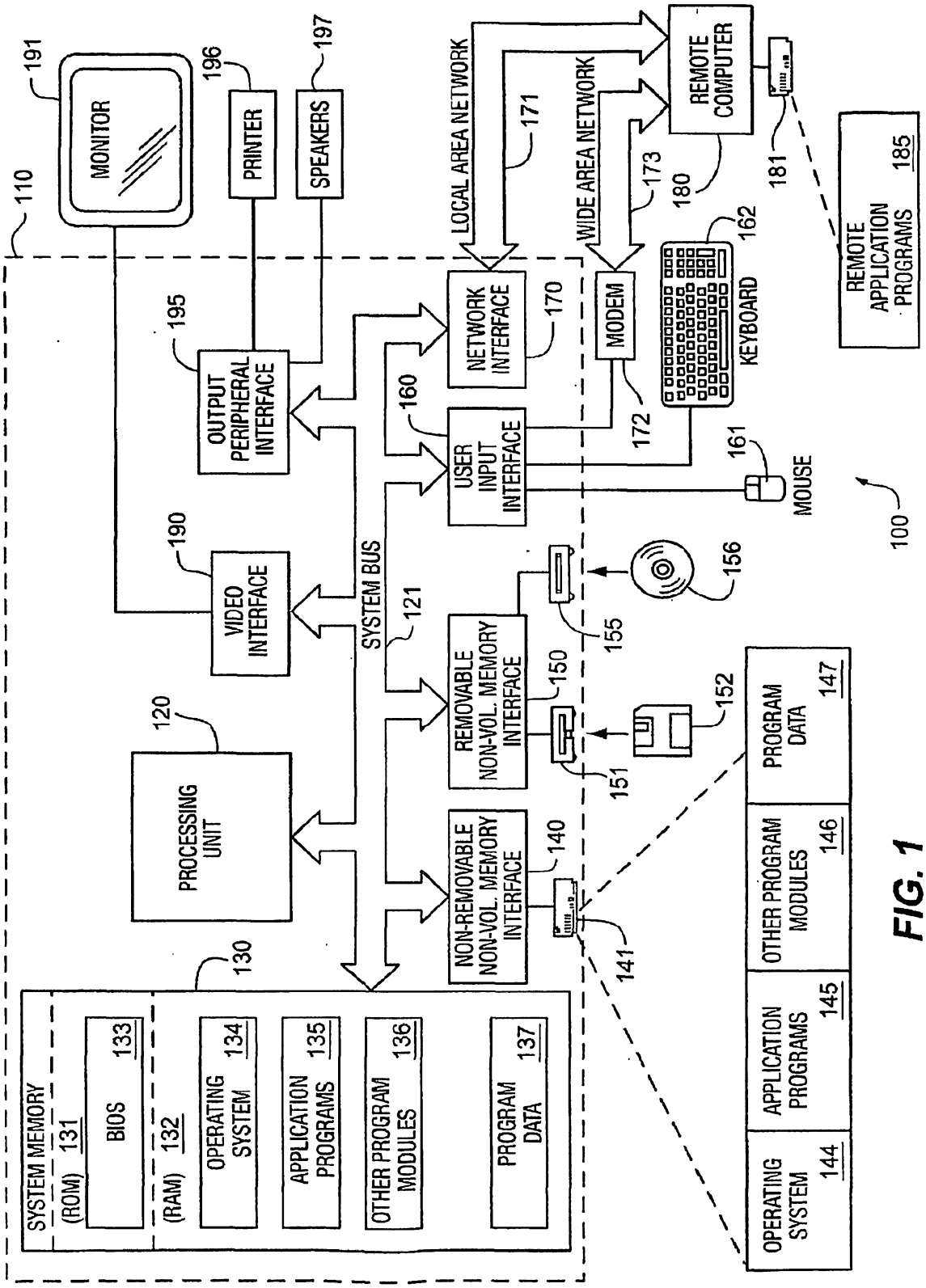

SYSTEM MEMORY
(ROM) 131
BIOS 133
(RAM) 132
OPERATING SYSTEM 134
APPLICATION PROGRAMS 135
OTHER PROGRAM MODULES 136
PROGRAM DATA 137
130
PROCESSING UNIT
120
VIDEO INTERFACE
190
OUTPUT PERIPHERAL INTERFACE
195
NETWORK INTERFACE
170
USER INPUT INTERFACE
160
MODEM
172
KEYBOARD
162
MOUSE
161
REMOTE COMPUTER
180
REMOTE APPLICATION PROGRAMS
185
181
LOCAL AREA NETWORK
171
WIDE AREA NETWORK
173
MONITOR
191
PRINTER
196
SPEAKERS
197
110
SYSTEM BUS
121
REMOVABLE NON-VOL. MEMORY INTERFACE
150
NON-REMOVABLE NON-VOL. MEMORY INTERFACE
140
155
156
151
152
141
100
OPERATING SYSTEM 144
APPLICATION PROGRAMS 145
OTHER PROGRAM MODULES 146
PROGRAM DATA 147

FIG. 1

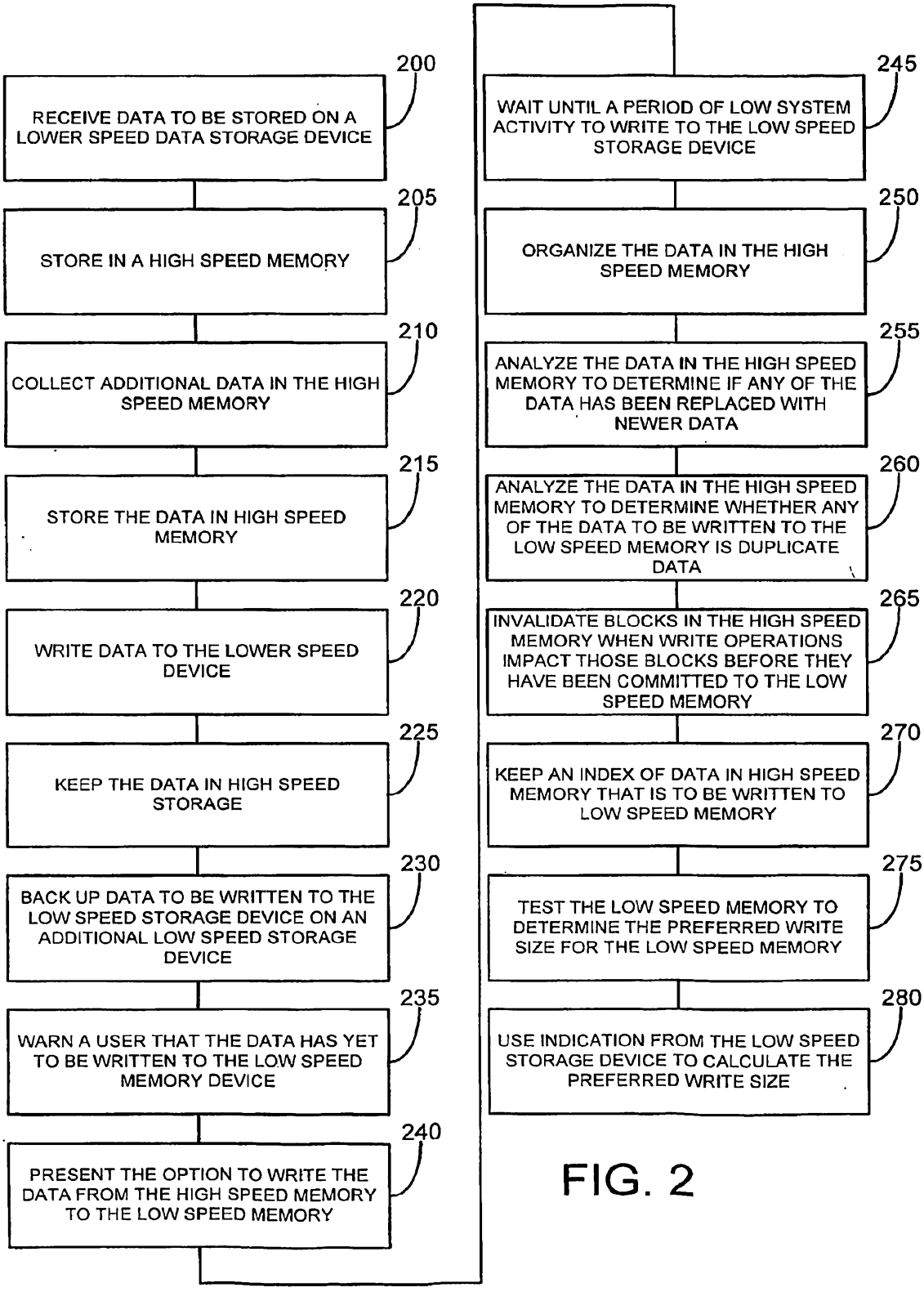

200
RECEIVE DATA TO BE STORED ON A LOWER SPEED DATA STORAGE DEVICE
205
STORE IN A HIGH SPEED MEMORY
210
COLLECT ADDITIONAL DATA IN THE HIGH SPEED MEMORY
215
STORE THE DATA IN HIGH SPEED MEMORY
220
WRITE DATA TO THE LOWER SPEED DEVICE
225
KEEP THE DATA IN HIGH SPEED STORAGE
230
BACK UP DATA TO BE WRITTEN TO THE LOW SPEED STORAGE DEVICE ON AN ADDITIONAL LOW SPEED STORAGE DEVICE
235
WARN A USER THAT THE DATA HAS YET TO BE WRITTEN TO THE LOW SPEED MEMORY DEVICE
240
PRESENT THE OPTION TO WRITE THE DATA FROM THE HIGH SPEED MEMORY TO THE LOW SPEED MEMORY
245
WAIT UNTIL A PERIOD OF LOW SYSTEM ACTIVITY TO WRITE TO THE LOW SPEED STORAGE DEVICE
250
ORGANIZE THE DATA IN THE HIGH SPEED MEMORY
255
ANALYZE THE DATA IN THE HIGH SPEED MEMORY TO DETERMINE IF ANY OF THE DATA HAS BEEN REPLACED WITH NEWER DATA
260
ANALYZE THE DATA IN THE HIGH SPEED MEMORY TO DETERMINE WHETHER ANY OF THE DATA TO BE WRITTEN TO THE LOW SPEED MEMORY IS DUPLICATE DATA
265
INVALIDATE BLOCKS IN THE HIGH SPEED MEMORY WHEN WRITE OPERATIONS IMPACT THOSE BLOCKS BEFORE THEY HAVE BEEN COMMITTED TO THE LOW SPEED MEMORY
270
KEEP AN INDEX OF DATA IN HIGH SPEED MEMORY THAT IS TO BE WRITTEN TO LOW SPEED MEMORY
275
TEST THE LOW SPEED MEMORY TO DETERMINE THE PREFERRED WRITE SIZE FOR THE LOW SPEED MEMORY
280
USE INDICATION FROM THE LOW SPEED STORAGE DEVICE TO CALCULATE THE PREFERRED WRITE SIZE

FIG. 2