



(51) International Patent Classification:

G06F 11/14 (2006.01) G06F 9/44 (2006.01)
G06F 11/22 (2006.01)

(21) International Application Number:

PCT/US2016/013317

(22) International Filing Date:

14 January 2016 (14.01.2016)

(25) Filing Language:

English

(26) Publication Language:

English

(71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 11445 Compaq Center Drive W., Houston, Texas 77070 (US).

(72) Inventors: **LIU, Wei Ze**; 11445 Compaq Center Drive W, Houston, Texas 77070 (US). **BRADUKE, Rosilet Retnamoni**; 11445 Compaq Center Drive W, Houston, Texas 77070 (US). **SHAH, Rajesh A.**; 11445 Compaq Center Drive W, Houston, Texas 77070 (US). **WANG, Lan**; 11445 Compaq Center Drive W, Houston, Texas 77070 (US).

(74) Agents: **MAISAMI, Ceyda Azakli** et al.; HP Inc., Mail Stop 35, 3390 E. Harmony Road, Fort Collins, Colorado 80528 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, 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, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

[Continued on next page]

(54) Title: MANAGEMENT WITH RESPECT TO A BASIC INPUT/OUTPUT SYSTEM POLICY

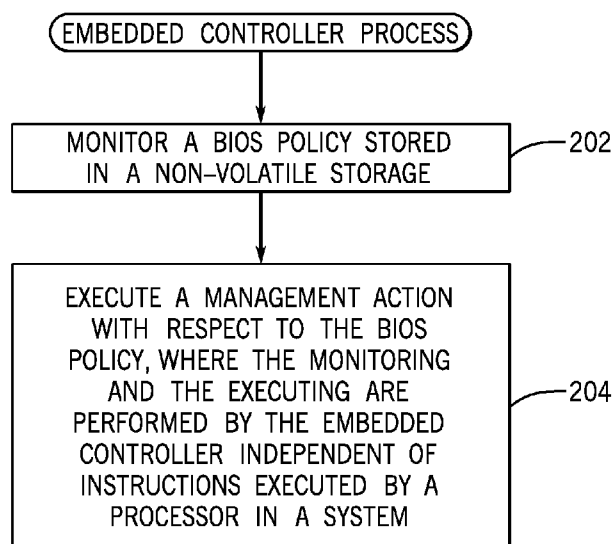


FIG. 2

(57) Abstract: In some examples, an embedded controller in a system monitors a basic input/output system (BIOS) policy stored in a non-volatile storage, and executes a management action with respect to the BIOS policy, where the monitoring and the executing are performed by the embedded controller independent of instructions executed by a processor in the system.

WO 2017/123225 A1



Published:

— *with international search report (Art. 21(3))*

MANAGEMENT WITH RESPECT TO A BASIC INPUT/OUTPUT SYSTEM POLICY

Background

[0001] A computing system can include code to perform various startup functions of the computing system. This code can include Basic Input/Output System (BIOS) code. The BIOS code is executed when the computing system initially starts up, and can be executed in response to other events.

Brief Description Of The Drawings

[0002] Some implementations of the present disclosure are described with respect to the following figures.

[0003] Fig. 1 is a block diagram of an example computing system according to some implementations.

[0004] Fig. 2 is a flow diagram of an example process of an embedded controller, according to some implementations.

[0005] Fig. 3 is a block diagram of another example computing system according to further implementations.

[0006] Fig. 4 is a block diagram of an example non-transitory storage medium storing machine-readable instructions according to some implementations.

Detailed Description

[0007] System code used to perform startup of a computing system can include system firmware, which can be in the form of machine-readable instructions executable on a processor (or processors) of the computing system. "System firmware" can refer to any machine-readable instructions that are able to perform startup of a computing system and other tasks. Examples of computing systems include desktop computers, notebook computers, tablet computers, personal digital assistants (PDAs), smart phones, game appliances, wearable devices (e.g. smart

watches, smart eyeglasses, etc.), server computers, storage nodes, network communication nodes, vehicles, and so forth.

[0008] System firmware can include Basic Input/Output System (BIOS) code, which can initialize various components of the computing system, and load an operating system (OS) of the computing system. The BIOS code can perform checking of hardware components to ensure that the hardware components are present and functioning properly. This can be part of a power-on self-test (POST) procedure, for example. After the POST procedure, the BIOS code can progress through the remainder of a booting sequence, after which the BIOS code can load and pass control to the OS. In some cases, the BIOS code can also perform certain tasks (runtime services) after control of the computing system has been passed to the OS.

[0009] In the present disclosure, a “BIOS” can also refer to a Unified Extensible Firmware Interface (UEFI), and “BIOS code” can also refer to “UEFI code.” UEFI code can perform boot services during boot time of a computing system. UEFI code can additionally provide certain runtime services (during system runtime in which an OS is active). A version of UEFI is described by the Unified Extensible Firmware Interface Specification, Version 2.5 (April 2015).

[0010] The behavior of a BIOS code can be based on a BIOS policy (or multiple BIOS policies). The BIOS policy can be stored in a non-volatile storage. The non-volatile storage can include a non-volatile memory device (or multiple non-volatile memory devices). Examples of non-volatile memory devices include any or some combination of the following: a flash memory device, a memristor memory, spin-transfer torque memory device, a phase change memory device, or other type of memory device in which content of the memory device is maintained even when power is removed from the memory device. In further examples, the non-volatile storage can include a disk-based storage device.

[0011] Examples of BIOS policies include any or some combination of the following: a BIOS policy that specifies whether a computing system is allowed to be

awakened from a lower power state (e.g. sleep state) by a message transmitted to the computing system over a network (in some examples, this feature is referred to as a wake-on-LAN (local area network) feature); a BIOS policy specifying whether to use Universal Serial Bus (USB) 2 or USB 3 mode; a BIOS specifying a boot order of different storage devices (e.g. attempt to boot from an external storage device first followed by boot from an internal storage device); and other policies.

[0012] A BIOS policy can be set by a user, such as by using a startup screen of the BIOS code. In the startup screen, the user can enable or disable certain features of the computing system, which causes corresponding BIOS policies to be stored in the non-volatile storage. In other examples, a BIOS policy can be set by code executing in the computing system, or can be provided to the computing system from an external source, such as a management server or other management entity.

[0013] A BIOS policy can become corrupted due to various causes. For example, a BIOS policy may be corrupted by malware executing in a computing system or in an external system, where the malware can tamper with the BIOS policy by changing content in the BIOS policy or replacing the BIOS policy with a different BIOS policy. A BIOS policy is also considered to be corrupted if the BIOS policy includes a setting or other information that causes the BIOS code to malfunction or return an error. Thus generally, a BIOS policy is “corrupted” if the BIOS policy has been tampered with in an unauthorized manner, or the BIOS policy includes information that causes BIOS code (or a computing system in which the BIOS code executes) to not function in an expected manner.

[0014] In some cases, a BIOS policy can be managed by a BIOS code executed on a processor of a computing system. In the ensuing discussion, reference to execution of a BIOS code or other machine-readable instructions on a processor is a reference to execution of the BIOS code or other machine-readable instructions on one processor, or on multiple processors.

[0015] The management of the BIOS policy can include any or some combination of the following example actions: monitoring the BIOS policy to

determine corruption in the BIOS policy, restoring the BIOS policy (which may be corrupted) to a different version of the BIOS policy (e.g. a known good version of the BIOS policy), and reclaiming storage space in a region of a non-volatile storage used to store BIOS policies. Although specific example management actions are listed above, it is noted that in other examples, other types of management actions can be performed with respect to BIOS policies.

[0016] Performing management actions using the BIOS code can place a processing burden on the processor(s) of a computing system. While the BIOS code is performing the management actions, the processor(s) may not be available, or may have a reduced processing capacity, to perform other tasks of the computing system. Also, if the BIOS code itself becomes compromised for any reason (e.g. due to malware attack or for some other reason), then the BIOS code may no longer be able to perform management actions with respect to BIOS policies.

[0017] In accordance with some implementations, instead of or in addition to performing management actions with respect to a BIOS policy by BIOS code executed on a processor of a computing system, an embedded controller can perform management actions with respect to the BIOS policy.

[0018] Fig. 1 shows an example of a computing system 100 that includes an embedded controller 102 that is able to perform management actions with respect to a BIOS policy (or multiple BIOS policies) 104. The computing system 100 also includes a processor (or processors) 106. In the ensuing discussion, reference is made to “a processor”; however, it should be understood that “a processor” can refer to one processor or to multiple processors. A processor can include a microprocessor, a core of a multi-core microprocessor, a microcontroller, a programmable integrated circuit, a programmable gate array, or any other hardware processing circuit.

[0019] The embedded controller 102 is physically separate from the processor 106 of the computing system 100. Note that the embedded controller 102 can also refer to a super input/output (I/O) chip used to control I/O devices. The processor

106 is used for executing a BIOS code 105, an operating system (OS), an application code, and other code in the computing system 100. The term “code” can refer to machine-readable instructions that can be executed. Due to hardware separation, an OS, an application, and other system code in the computing system 100 are unable to execute in the embedded controller 102 or access any data in the embedded controller 102.

[0020] The embedded controller 102, on the other hand, can be used to perform specific predefined tasks, as programmed into embedded controller (EC) firmware (machine-readable instructions) executable by the embedded controller 102. In addition to the ability to perform management actions with respect to the BIOS policy 104, examples of other tasks that can be performed by the embedded controller 102 include any one or some combination of the following: power supply control in the computing system 100 (for controlling a power supply that supplies power supply voltages to various components in the computing system 100), charging and control of a battery in the computing system 100, thermal monitoring (to monitor a temperature in the computing system 100), fan control (to control a fan in the computing system 100), and interaction with a user input device (such as performing a scan of a keyboard of the computing system 100 or interaction with a pointing device such as a mouse, touchpad, touchscreen, and so forth). The embedded controller 102 can be implemented with a microcontroller, an application-specific integrated circuit (ASIC), a programmable gate array (PGA), or any other type of programmable circuit.

[0021] More generally, an “embedded controller” can refer to a controller, separate from the processor(s) of a computing system, that is designated to perform specified tasks in the computing system. In some examples, the embedded controller can be mounted on the same circuit board as the processor(s). In other examples, the embedded controller can be mounted on a separate circuit board than the circuit board on which the processor(s) is mounted; however, both the embedded controller and the processor(s) are provided within the same housing of the computing system.

[0022] As shown in Fig. 1, the BIOS code 105 and the BIOS policy 104 are stored in a non-volatile storage 108. The non-volatile storage 108 can be implemented with one or multiple non-volatile storage devices.

[0023] The BIOS code 105 is executable on the processor 106 during startup of the computing system 100 (prior to loading of an OS) and/or during runtime of the computing system 100 (after the OS of the computing system 100 has been loaded for execution).

[0024] As noted above, the embedded controller 102 is able to perform management actions with respect to the BIOS policy 104, based on monitoring the BIOS policy 104 that is stored in the non-volatile storage 108. In some examples, the embedded controller 102 can detect an issue associated with the BIOS policy, where the issue can be corruption or other issue of the BIOS policy that affects operation of the BIOS code 105 or the computing system 100. The monitoring that is performed by the embedded controller 102 can include monitoring for tampering of the BIOS policy 104. For example, the embedded controller 102 can compare the BIOS policy 104 to a known good version (or a backup copy) of the BIOS policy. If the compared copies do not match, then that is an indication that the BIOS policy 104 has been tampered with. In other examples, the embedded controller 102 is able to monitor for other issues associated with the BIOS policy 104.

[0025] In response to detecting the issue associated with the BIOS policy 104, the embedded controller 102 can restore a different version of the BIOS policy to the non-volatile storage 108. The restored different version of the BIOS policy can replace the BIOS policy 104, in which case the BIOS policy 104 can be removed from the non-volatile storage 108, or alternatively, the restored version of the BIOS policy can be marked as the active BIOS policy and the BIOS policy 104 can be marked as the inactive or obsolete (invalid) version.

[0026] By using techniques according to some implementations of the present disclosure, the embedded controller 102 is able to perform management actions with respect to the BIOS policy 104 even when the BIOS code 105 and/or the processor

106 are not available. For example, the BIOS code 105 may have been compromised by malware, such that the BIOS code 105 is no longer able to perform management actions with respect to the BIOS policy 104.

[0027] Although reference has been made to the embedded controller 102 performing management actions with respect to the BIOS policy 104, it is noted that in some implementations, the BIOS code 105 when executing on the processor 106 can also perform management actions with respect to the BIOS policy 104.

[0028] The non-volatile storage 108 can be a shared non-volatile storage that is accessible by both the embedded controller 102 and the processor 106. As a result, certain content (the BIOS policy 104) in the non-volatile storage 108 can be accessed (and possibly updated) by multiple entities, such as by the embedded controller 102 and the BIOS code 105 executed on the processor 106.

[0029] To avoid conflict between the processor 106 and the embedded controller 102, the embedded controller 102 can read the BIOS policy 104 in the non-volatile storage 108 at any time, but can write a BIOS policy in the non-volatile storage 108 only when the processor 106 is inactive with respect to BIOS policies (in other words, the processor 106 is not executing any code, such as BIOS code 105, which can perform rights of BIOS policies in a non-volatile storage 108).

[0030] In some implementations, in examples where the non-volatile storage 108 stores multiple BIOS policies, the embedded controller 102 is able to perform management actions with respect to just a subset (less than all) of the BIOS policies. The BIOS policies of this subset can be considered “critical” BIOS policies, as designated by a user of the computing system 100 or by code executing in the computing system 100. In contrast, in some implementations, the BIOS code 105 is able to perform management actions with respect to all of the BIOS policies stored in the non-volatile storage 108.

[0031] Fig. 2 is a flow diagram of an example process that can be performed by the embedded controller 102 according to some implementations. The embedded

controller monitors (at 202) a BIOS policy (e.g. BIOS policy 104 in Fig. 1) stored in a non-volatile storage (e.g. non-volatile storage 108 in Fig. 1). The embedded controller 102 executes (at 204) a management action with respect to the BIOS policy, where the monitoring and the executing are performed by the embedded controller 102 independent of instructions executed by any processor (e.g. processor(s) 106 in the computing system 100). The executing of the management action can be in response to at least one trigger selected from among: the monitoring (at 202), a policy of the embedded controller 102, or a user request.

[0032] Fig. 3 is a block diagram of an example computing system 100A according to alternative implementations. The computing system 100A includes the processor 106, the embedded controller 102, and the non-volatile storage 108, which is a shared non-volatile storage that is accessible by both the processor 106 and the embedded controller 102.

[0033] In addition, the computing system 100A includes a private non-volatile storage 302, which is accessible by just the embedded controller 102 (i.e. the private non-volatile storage 302 is not accessible by the processor 106 or any other entity of the computing system 100A, other than the embedded controller 102). The private non-volatile storage 302 can be implemented with one or multiple non-volatile storage devices. The private non-volatile storage 302 stores a known good BIOS policy 304, which can be a prior version of the BIOS policy 104 that has been confirmed to not have been corrupted and that is not faulty. More generally, the private non-volatile storage 302 can store a copy of a BIOS policy (or copies of BIOS policies) that can be used to restore a BIOS policy in the shared non-volatile storage 108.

[0034] In some examples, in response to detecting an issue (e.g. corruption) with respect to the BIOS policy 104 in the shared non-volatile storage 108, the embedded controller 102 can restore the known good BIOS policy 304 from the private non-volatile storage 302 to the shared non-volatile storage 108, by writing the known good BIOS policy 304 to the shared non-volatile storage such that the BIOS code

105 when executed uses the known good BIOS policy 304 rather than the BIOS policy 104 associated with the issue.

[0035] As explained above, a management action with respect to the BIOS policy 104 can be triggered in response to monitoring the BIOS policy 104. In other examples, a management action with respect to the BIOS policy can have other triggers, including a policy 310 in the embedded controller, and/or a user request received through a user interface 312. The policy 310 can be stored in a storage medium of the embedded controller 102, or can be stored in a storage medium external of the embedded controller 102 but accessible by the embedded controller 102. As examples, the policy 310 can specify times at which management actions can be performed, such as at specific times of the day or periodic intervals.

[0036] The user interface 312 can include a touchscreen display, a mouse input device, a keyboard, or any other type of user input device along with a display screen through which information can be presented to a user. A user can submit a user request through the user interface 312 to perform a management action with respect to the BIOS policy 104.

[0037] In further implementations, the embedded controller 102 is able to perform maintenance of BIOS policies stored in the non-volatile storage 108. For example, the embedded controller 102 can read the BIOS policies in the non-volatile storage 108, delete obsolete (invalid) copies of BIOS policies to keep the BIOS policy data in the non-volatile storage clean, or other management tasks. Deleting obsolete copies of BIOS policies can increase the amount of available storage of a BIOS policy region 306 in the non-volatile storage 108 that is designated to store BIOS policies. In other words, BIOS policies are allowed to be stored in the BIOS policy region 306, but not in another part of the shared non-volatile storage 108.

[0038] As BIOS policies are added to the BIOS policy region 306, the BIOS policy region 306 can become full such that the addition of a further BIOS policy to the BIOS policy region 306 may not be possible. By deleting obsolete copies of

BIOS policies from the BIOS policy region 306, available storage space of the BIOS policy region 306 can be reclaimed for storing other BIOS policies.

[0039] In some examples, the embedded controller 102 can have access to configuration data 308 that specifies which of various management actions the embedded controller 102 is to perform with respect to BIOS policies. In some examples, the configuration data 308 can be stored in the private non-volatile storage 302, or alternatively in a different location, such as a storage that is part of the embedded controller 102.

[0040] The configuration data 308 can be set by a user or by code executed in the computing system 100A, or by an external entity that is external of the computing entity 100A. For example, the configuration data 308 can specify whether or not the embedded controller 102 is to perform any or some combination of the following actions: monitoring BIOS policies stored in the shared non-volatile storage 108; reclamation of the BIOS policy region 306 used to store BIOS policies (reclamation refers to deleting copies of BIOS policies to free up available space in the BIOS policy region 306 to store other BIOS policies); restoration of a known good (or different version of) BIOS policy to the shared non-volatile storage 108 in response to detecting an issue with a BIOS policy; and so forth.

[0041] The configuration data 308 can also specify the timing of when the embedded controller 102 is to perform management actions with respect to BIOS policies. For example, the configuration data 308 can specify that the embedded controller 102 is to perform management actions with respect to BIOS policies on a periodic basis (e.g. daily, weekly, or at certain times during the day).

[0042] In addition, the configuration data 308 can specify whether the embedded controller 102 is to perform backup of BIOS policies stored in the shared non-volatile storage 108 to the private non-volatile storage 302. The configuration data 308 can also specify which version of multiple known good BIOS policies to use to restore a given BIOS policy in response to detecting an issue with the given BIOS policy.

[0043] Fig. 4 is a block diagram of an example non-transitory storage medium 400 that stores BIOS policy management instructions 402 that are executable by an embedded controller, such as the embedded controller 102. The BIOS policy management instructions 402 in some examples can include machine-readable instructions in the form of firmware executable by the embedded controller 102. The non-transitory storage medium 400 can be included in the embedded controller 102, or alternatively, the non-transitory storage medium 400 can be external of the embedded controller 102.

[0044] The BIOS policy management instructions 402 upon execution by the embedded controller 102 causes the embedded controller 102 in a computing system to execute management actions with respect to BIOS policies stored in a non-volatile storage (e.g., non-volatile storage 108). The executing is performed by the embedded controller 102 independent of instructions executed by any processor in the computing system. The management actions include detecting an issue associated with a BIOS policy, recovering from the issue (e.g. by replacing a corrupted BIOS policy with a different version of the BIOS policy), and reclaiming a portion of a region of the non-volatile storage storing the BIOS policies.

[0045] The storage medium 400 can include one or multiple different forms of memory including semiconductor memory devices such as dynamic or static random access memories (DRAMs or SRAMs), erasable and programmable read-only memories (EPROMs), electrically erasable and programmable read-only memories (EEPROMs) and flash memories; magnetic disks such as fixed, floppy and removable disks; other magnetic media including tape; optical media such as compact disks (CDs) or digital video disks (DVDs); or other types of storage devices. Note that the instructions discussed above can be provided on one computer-readable or machine-readable storage medium, or alternatively, can be provided on multiple computer-readable or machine-readable storage media distributed in a large system having possibly plural nodes. Such computer-readable or machine-readable storage medium or media is (are) considered to be part of an article (or article of manufacture). An article or article of manufacture can refer to any manufactured

single component or multiple components. The storage medium or media can be located either in the machine running the machine-readable instructions, or located at a remote site from which machine-readable instructions can be downloaded over a network for execution.

[0046] In the foregoing description, numerous details are set forth to provide an understanding of the subject disclosed herein. However, implementations may be practiced without some of these details. Other implementations may include modifications and variations from the details discussed above. It is intended that the appended claims cover such modifications and variations.

What is claimed is:

- 1 1. A method comprising:
2 monitoring, by an embedded controller in a system, a basic input/output
3 system (BIOS) policy stored in a non-volatile storage; and
4 executing, by the embedded controller, a management action with respect to
5 the BIOS policy, wherein the monitoring and the executing are performed by the
6 embedded controller independent of instructions executed by a processor in the
7 system.
- 1 2. The method of claim 1, wherein executing the management action is in
2 response to at least one trigger selected from among the monitoring, a policy of the
3 embedded controller, or a user request.
- 1 3. The method of claim 1, wherein executing the management action comprises
2 recovering from an issue associated with the BIOS policy, wherein the recovering
3 comprises restoring a different version of the BIOS policy to the non-volatile storage.
- 1 4. The method of claim 3, wherein the different version of the BIOS policy is
2 stored in a private non-volatile storage that is accessible to the embedded controller
3 but inaccessible to the processor in the system.
- 1 5. The method of claim 1, further comprising:
2 reclaiming, by the embedded controller, a portion of a region of the non-
3 volatile storage used to store BIOS policies.
- 1 6. The method of claim 5, wherein reclaiming the portion of the region comprises
2 deleting at least one BIOS policy from the region.

- 1 7. The method of claim 1, wherein the management action executed by the
2 embedded controller is in addition to a management action performed by BIOS code
3 executed on the processor with respect to the BIOS policy.
- 1 8. The method of claim 1, wherein the monitoring comprises comparing the
2 BIOS policy stored in the non-volatile storage with another copy of the BIOS policy to
3 determining whether tampering has occurred with respect to the BIOS policy.
- 1 9. The method of claim 1, further comprising setting configuration data to select
2 which management action is to be performed by the embedded controller with
3 respect to the BIOS policy.
- 1 10. A computing system comprising:
2 a processor to execute basic input/output system (BIOS) code;
3 a non-volatile storage to store a BIOS policy that governs a behavior of the
4 BIOS code; and
5 an embedded controller separate from the processor and to:
6 detect an issue associated with the BIOS policy, and
7 in response to detecting the issue associated with the BIOS policy,
8 restore a different version of the BIOS policy to the non-volatile storage.
- 1 11. The computing system of claim 10, wherein the non-volatile storage is a
2 shared non-volatile storage accessible by the embedded controller and the
3 processor, the computing system further comprising:
4 a private non-volatile storage accessible by the embedded controller and
5 inaccessible by the processor,
6 the embedded controller to further back up a copy of the BIOS policy to the
7 private non-volatile storage.

1 12. The computing system of claim 11, wherein the restoring of the different
2 version of the BIOS policy comprises restoring the copy of the BIOS policy to the
3 shared non-volatile storage.

1 13. The computing system of 10, wherein the non-volatile storage includes a
2 region designated to store BIOS policies, and the embedded controller is to further
3 perform reclamation with respect to the region to remove an obsolete version of a
4 BIOS policy from the region.

1 14. A non-transitory storage medium storing instructions that upon execution by
2 an embedded controller cause the embedded controller in a computing system to:
3 execute management actions with respect to BIOS policies stored in a non-
4 volatile storage, wherein the executing is performed by the embedded controller
5 independent of instructions executed by any processor in the computing system, and
6 wherein the management actions comprise:
7 detecting an issue associated with a BIOS policy of the BIOS policies,
8 and recovering from the issue, and
9 reclaiming a portion of a region storing the BIOS policies in the non-
10 volatile storage.

1 15. The non-transitory storage medium of claim 14, wherein the management
2 actions executed by the embedded controller are in addition to management actions
3 with respect to the BIOS policies performed by BIOS code executed on the
4 processor.

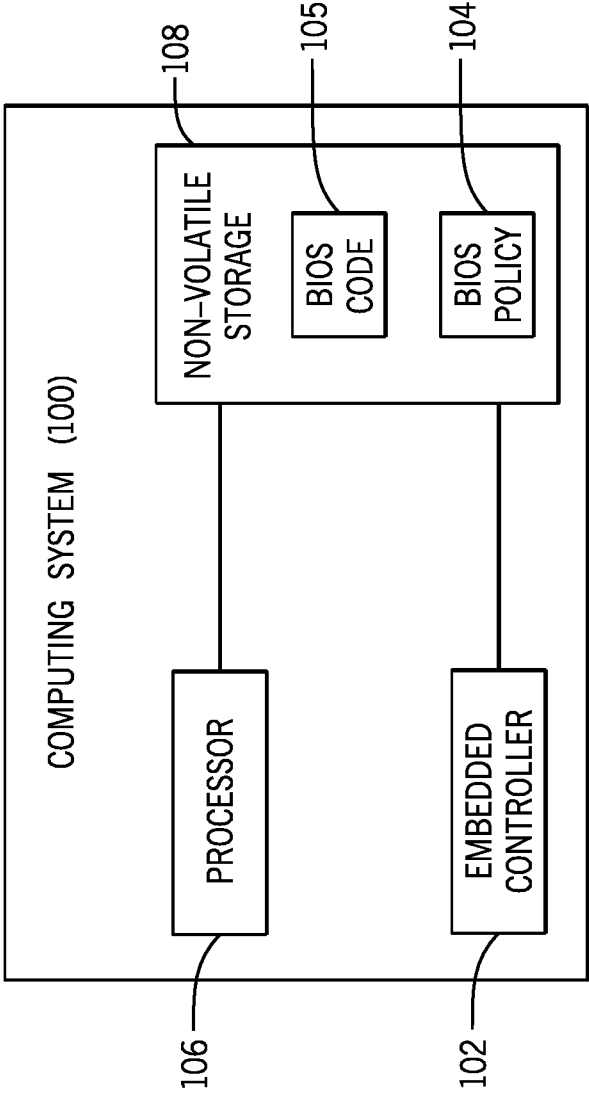


FIG. 1

2 / 4

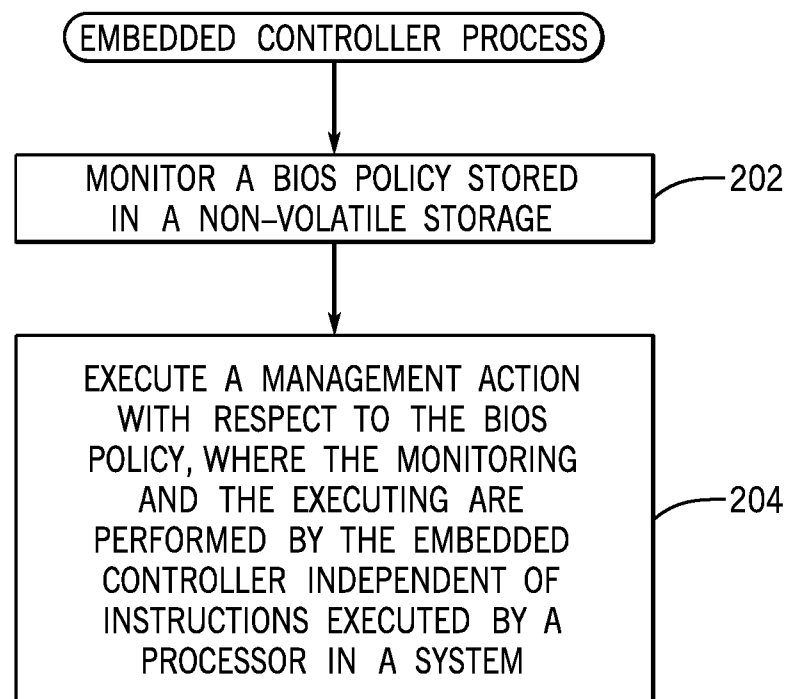


FIG. 2

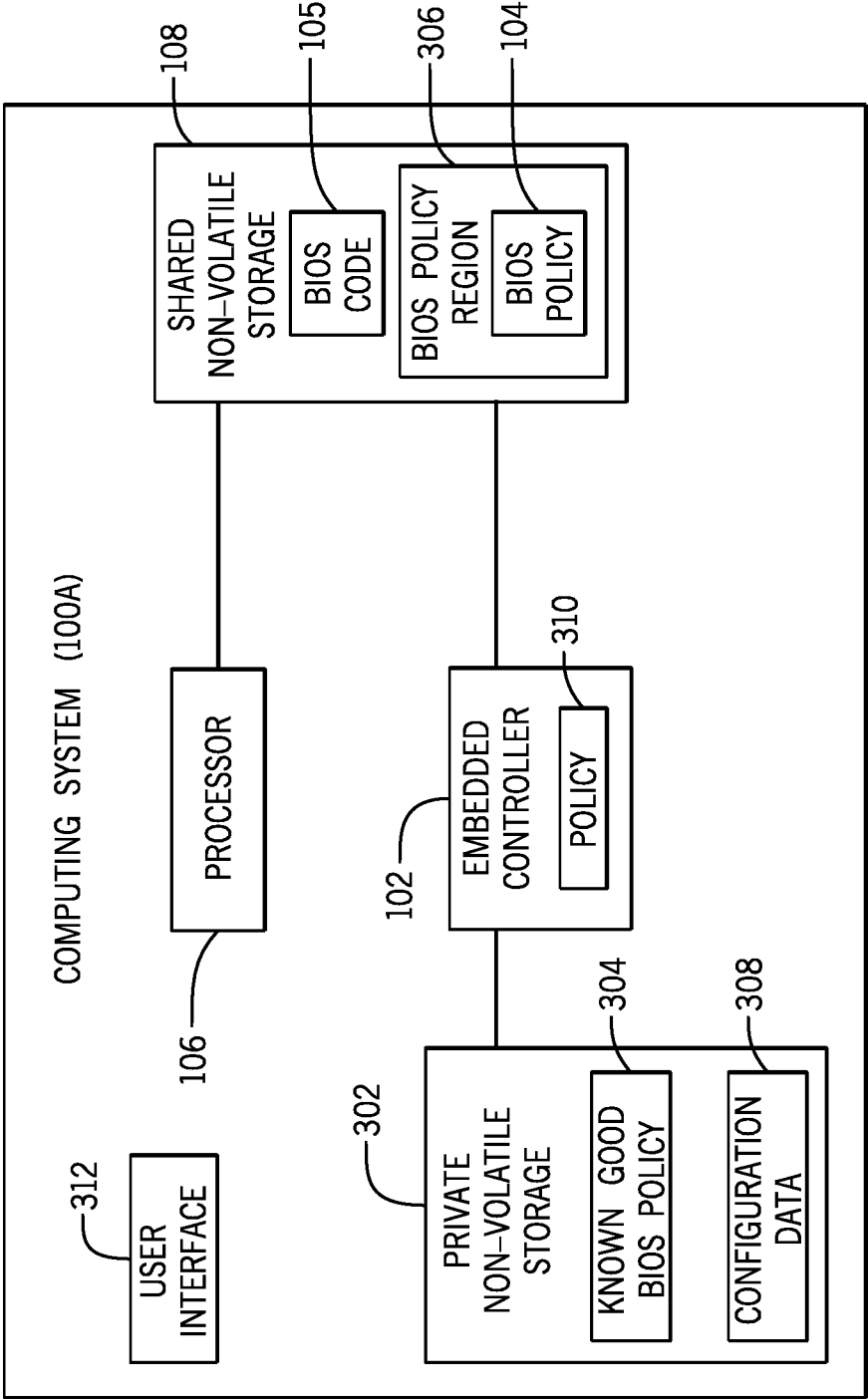


FIG. 3

4 / 4

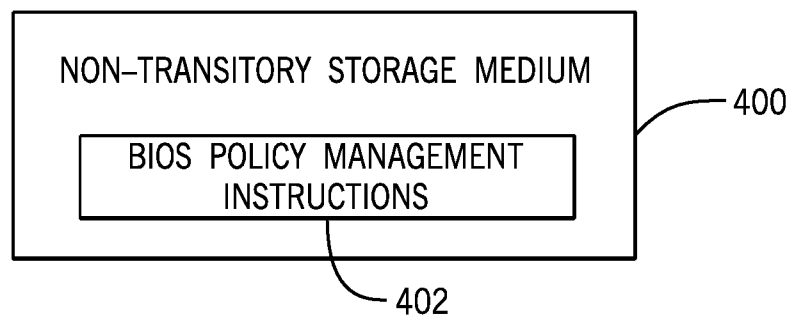


FIG. 4

A. CLASSIFICATION OF SUBJECT MATTER**G06F 11/14(2006.01)i, G06F 11/22(2006.01)i, G06F 9/44(2006.01)i**

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)

G06F 11/14; G06F 12/02; G06F 9/24; G06F 21/57; G06F 13/14; G06F 21/00; G06F 11/22; G06F 9/44

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & Keywords: BIOS, code, policy, management action, separate, embedded controller, and similar terms.**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2014-175864 A1 (HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.) 30 October 2014 See paragraphs [0014], [0018], [0022], [0024], [0034]-[0036], [0042], [0062], and [0086]-[0087]; and figures 2-3.	1-15
A	US 2015-0143095 A1 (AMERICAN MEGATRENDS, INC.) 21 May 2015 See paragraphs [0048], [0057]-[0058], and [0073]-[0075]; and figures 1 and 4.	1-15
A	US 8,745,364 B2 (MICHAEL A. ROTHMAN et al.) 03 June 2014 See column 5, line 33 - column 6, line 5; and figure 4.	1-15
A	WO 2014-175867 A1 (HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.) 30 October 2014 See paragraphs [0008]-[0025] and figure 1.	1-15
A	US 2014-0047229 A1 (WILLARD M. WISEMAN) 13 February 2014 See paragraphs [0025]-[0029] and figure 2.	1-15



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

"E" earlier application or patent but 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

12 September 2016 (12.09.2016)

Date of mailing of the international search report

12 September 2016 (12.09.2016)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

LEE, EUN KYU

Telephone No. +82-42-481-3580



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/013317

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2014-175864 A1	30/10/2014	None	
US 2015-0143095 A1	21/05/2015	None	
US 8745364 B2	03/06/2014	US 2006-0129744 A1	15/06/2006
WO 2014-175867 A1	30/10/2014	CN 105144185 A	09/12/2015
		TW 201443686 A	16/11/2014
		US 2016-0055332 A1	25/02/2016
US 2014-0047229 A1	13/02/2014	TW 201342115 A	16/10/2013
		US 9218490 B2	22/12/2015
		WO 2013-101178 A1	04/07/2013