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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 December 2020 (24.12.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/256782 A1

(51) International Patent Classification:
G06F 3/06 (2006.01)

(21) International Application Number:
PCT/US2019/067398

(22) International Filing Date:
19 December 2019 (19.12.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
16/444,461 18 June 2019 (18.06.2019) US

(71) Applicant: **WESTERN DIGITAL TECHNOLOGIES, INC.** [US/US]; 5601 Great Oaks Parkway, San Jose, CA 95119 (US).

(72) Inventor: **MUTHIAH, Ramanathan**; c/o Western Digital Technologies, Inc., 5601 Great Oaks Parkway, San Jose, CA 95119 (US).

(74) Agent: **HETZ, Joseph F.**; Brinks Gilson & Lione, P.O. Box 10087, Chicago, IL 60610 (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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, 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).

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

(54) Title: STORAGE SYSTEM AND METHOD FOR MEMORY BACKLOG HINTING FOR VARIABLE CAPACITY

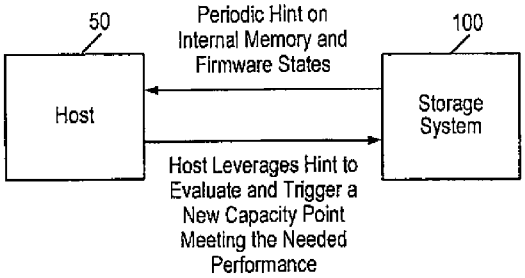

FIG. 3

(57) Abstract: A storage system and method for memory backlog hinting for variable capacity are provided. In one embodiment, a method for memory backlog hinting for variable capacity is provided that is performed in a storage system comprising a memory. The method comprises: sending information regarding a state of the memory to a host device; receiving an instruction from the host device to alter memory capacity in order to alter memory performance, wherein the instruction is based on the information regarding the state of the memory sent to the host device; and altering memory capacity in order to alter memory performance in response to receiving the instruction from the host device. Other embodiments are provided.

WO 2020/256782 A1

Storage System and Method for Memory Backlog Hinting for Variable Capacity

Background

[0001] Manufacturers market storage systems with various characteristics, such as capacity and performance. In some environments, a host device can request that a storage system trade off some capacity to meet a performance requirement. However, there is no guarantee that the storage system can satisfy such a request. In some systems, such as Flexible Flash File Systems, predetermined single level cells (SLC) and multi-level cells (MLC) can be changed dynamically by the storage system to change the capacity-performance tradeoff. To do this, the storage system can use complex firmware to internally manage these heterogeneous memory regions with a data migration policy to attempt to meet the host's request for a desired capacity.

Brief Description of the Drawings

[0002] Figure 1A is a block diagram of a non-volatile storage system of an embodiment.

[0003] Figure 1B is a block diagram illustrating a storage module of an embodiment.

[0004] Figure 1C is a block diagram illustrating a hierarchical storage system of an embodiment.

[0005] Figure 2A is a block diagram illustrating components of the controller of the non-volatile storage system illustrated in Figure 1A according to an embodiment.

[0006] Figure 2B is a block diagram illustrating components of the non-volatile memory storage system illustrated in Figure 1A according to an embodiment.

[0007] Figure 3 is a block diagram of a storage system and host of an embodiment for memory backlog hinting for variable capacity.

[0008] Figure 4 is a diagram illustrating a process flow of an embodiment.

[0009] Figure 5 is a diagram of an embodiment illustrating different performance modes.

Detailed Description

[0010] Overview

[0011] By way of introduction, the below embodiments relate to a storage system and method for memory backlog hinting for variable capacity. In one embodiment, a method for memory backlog hinting for variable capacity is provided that is performed in a storage system comprising a memory. The method comprises: sending information regarding a state of the memory to a host device; receiving an instruction from the host device to alter memory capacity in order to alter memory performance, wherein the instruction is based on the information regarding the state of the memory sent to the host device; and altering memory capacity in order to alter memory performance in response to receiving the instruction from the host device.

[0012] In some embodiments, the information regarding the state of the memory comprises one or more of the following: a number of free blocks in the memory, a number of used blocks in the memory, a number of single level cell (SLC) blocks in the memory, a number of triple level cell (TLC) blocks in the memory, a garbage collection workload, and backlog information.

[0013] In some embodiments, the method further comprises sending one or more of the following secondary information to the host device: memory age, memory quality, and memory endurance, wherein the instruction from the host device is also based on the secondary information.

[0014] In some embodiments, the method further comprises sending updated information about an updated state of the memory to the host device that indicates that it is no longer possible to reduce memory capacity in order to increase memory performance.

[0015] In some embodiments, the updated state indicates one or more of the following: urgent compaction, a wear level operation pending for longer than a threshold amount of time, and an urgent read scrub command.

[0016] In some embodiments, the method further comprises periodically sending updated information about the state of the memory to the host device.

[0017] In another embodiment, a storage system is provided comprising a memory and a controller configured to be in communication with the memory. The controller is further configured to: provide a hint to a host device on a condition of the memory; receive a command from the host device to alter memory capacity in order to alter memory performance, wherein the command is in response to the hint provided to the host device; and alter memory capacity in order to alter memory performance in response to receiving the command from the host device.

[0018] In some embodiments, the information regarding the state of the memory comprises one or more of the following: a number of free blocks in the memory, a number of used blocks in the memory, a number of single level cell (SLC) blocks in the memory, a number of triple level cell (TLC) blocks in the memory, a garbage collection workload, and backlog information.

[0019] In some embodiments, the controller is further configured to send one or more of the following secondary hints to the host device: memory age, memory quality, and memory endurance, wherein the command from the host device is also in response to the secondary information.

[0020] In some embodiments, the controller is further configured to send an updated hint to the host device that indicates that it is no longer possible to reduce memory capacity in order to increase memory performance.

[0021] In some embodiments, the updated hint indicates one or more of the following: urgent compaction, a wear level operation pending for longer than a threshold amount of time, and an urgent read scrub command.

[0022] In some embodiments, the controller is further configured to periodically send updated hints about the state of the memory to the host device.

[0023] In some embodiments, the memory comprises a three-dimensional memory.

[0024] In some embodiments, the storage system is configured to be integrated in the host device.

[0025] In some embodiments, the storage system is configured to be removably connected with the host device.

[0026] In another embodiment, a storage system is provided comprising: a memory; means for sending information regarding a state of the memory to a host device; means

for receiving an instruction from the host device to alter memory capacity in order to alter memory performance, wherein the instruction is based on the information regarding the state of the memory sent to the host device; and means for altering memory capacity in order to alter memory performance in response to receiving the instruction from the host device.

[0027] In some embodiments, the information regarding the state of the memory comprises one or more of the following: a number of free blocks in the memory, a number of used blocks in the memory, a number of single level cell (SLC) blocks in the memory, a number of triple level cell (TLC) blocks in the memory, a garbage collection workload, and backlog information.

[0028] In some embodiments, the storage system further comprises means for sending one or more of the following secondary information to the host device: memory age, memory quality, and memory endurance, wherein the instruction from the host device is also based on the secondary information.

[0029] In some embodiments, the storage system further comprises means for sending updated information about an updated state of the memory to the host device that indicates that it is no longer possible to reduce memory capacity in order to increase memory performance.

[0030] In some embodiments, the storage system further comprises means for periodically sending updated information about the state of the memory to the host device.

[0031] Other embodiments are possible, and each of the embodiments can be used alone or together in combination. Accordingly, various embodiments will now be described with reference to the attached drawings.

[0032] Embodiments

[0033] Storage systems suitable for use in implementing aspects of these embodiments are shown in Figures 1A-1C. Figure 1A is a block diagram illustrating a non-volatile storage system 100 according to an embodiment of the subject matter described herein. Referring to Figure 1A, non-volatile storage system 100 includes a controller 102 and non-volatile memory that may be made up of one or more non-volatile memory die 104. As used herein, the term die refers to the collection of non-volatile memory cells, and

associated circuitry for managing the physical operation of those non-volatile memory cells, that are formed on a single semiconductor substrate. Controller 102 interfaces with a host system and transmits command sequences for read, program, and erase operations to non-volatile memory die 104.

[0034] The controller 102 (which may be a non-volatile memory controller (e.g., a flash, resistive random-access memory (ReRAM), phase-change memory (PCM), or magnetoresistive random-access memory (MRAM) controller)) can take the form of processing circuitry, a microprocessor or processor, and a computer-readable medium that stores computer-readable program code (e.g., firmware) executable by the (micro)processor, logic gates, switches, an application specific integrated circuit (ASIC), a programmable logic controller, and an embedded microcontroller, for example. The controller 102 can be configured with hardware and/or firmware to perform the various functions described below and shown in the flow diagrams. Also, some of the components shown as being internal to the controller can also be stored external to the controller, and other components can be used. Additionally, the phrase "operatively in communication with" could mean directly in communication with or indirectly (wired or wireless) in communication with through one or more components, which may or may not be shown or described herein.

[0035] As used herein, a non-volatile memory controller is a device that manages data stored on non-volatile memory and communicates with a host, such as a computer or electronic device. A non-volatile memory controller can have various functionality in addition to the specific functionality described herein. For example, the non-volatile memory controller can format the non-volatile memory to ensure the memory is operating properly, map out bad non-volatile memory cells, and allocate spare cells to be substituted for future failed cells. Some part of the spare cells can be used to hold firmware to operate the non-volatile memory controller and implement other features. In operation, when a host needs to read data from or write data to the non-volatile memory, it can communicate with the non-volatile memory controller. If the host provides a logical address to which data is to be read/written, the non-volatile memory controller can convert the logical address received from the host to a physical address in the non-volatile memory. (Alternatively, the host can provide the physical address.) The non-

volatile memory controller can also perform various memory management functions, such as, but not limited to, wear leveling (distributing writes to avoid wearing out specific blocks of memory that would otherwise be repeatedly written to) and garbage collection (after a block is full, moving only the valid pages of data to a new block, so the full block can be erased and reused).

[0036] Non-volatile memory die 104 may include any suitable non-volatile storage medium, including resistive random-access memory (ReRAM), magnetoresistive random-access memory (MRAM), phase-change memory (PCM), NAND flash memory cells and/or NOR flash memory cells. The memory cells can take the form of solid-state (e.g., flash) memory cells and can be one-time programmable, few-time programmable, or many-time programmable. The memory cells can also be single-level cells (SLC), multiple-level cells (MLC), triple-level cells (TLC), or use other memory cell level technologies, now known or later developed. Also, the memory cells can be fabricated in a two-dimensional or three-dimensional fashion.

[0037] The interface between controller 102 and non-volatile memory die 104 may be any suitable flash interface, such as Toggle Mode 200, 400, or 800. In one embodiment, storage system 100 may be a card based system, such as a secure digital (SD) or a micro secure digital (micro-SD) card. In an alternate embodiment, storage system 100 may be part of an embedded storage system.

[0038] Although, in the example illustrated in Figure 1A, non-volatile storage system 100 (sometimes referred to herein as a storage module) includes a single channel between controller 102 and non-volatile memory die 104, the subject matter described herein is not limited to having a single memory channel. For example, in some storage system architectures (such as the ones shown in Figures 1B and 1C), 2, 4, 8 or more memory channels may exist between the controller and the memory device, depending on controller capabilities. In any of the embodiments described herein, more than a single channel may exist between the controller and the memory die, even if a single channel is shown in the drawings.

[0039] Figure 1B illustrates a storage module 200 that includes plural non-volatile storage systems 100. As such, storage module 200 may include a storage controller 202 that interfaces with a host and with storage system 204, which includes a plurality of non-

volatile storage systems 100. The interface between storage controller 202 and non-volatile storage systems 100 may be a bus interface, such as a serial advanced technology attachment (SATA), peripheral component interconnect express (PCIe) interface, or double-data-rate (DDR) interface. Storage module 200, in one embodiment, may be a solid state drive (SSD), or non-volatile dual in-line memory module (NVDIMM), such as found in server PC or portable computing devices, such as laptop computers, and tablet computers.

[0040] Figure 1C is a block diagram illustrating a hierarchical storage system. A hierarchical storage system 250 includes a plurality of storage controllers 202, each of which controls a respective storage system 204. Host systems 252 may access memories within the storage system via a bus interface. In one embodiment, the bus interface may be a Non-Volatile Memory Express (NVMe) or fiber channel over Ethernet (FCoE) interface. In one embodiment, the system illustrated in Figure 1C may be a rack mountable mass storage system that is accessible by multiple host computers, such as would be found in a data center or other location where mass storage is needed.

[0041] Figure 2A is a block diagram illustrating components of controller 102 in more detail. Controller 102 includes a front end module 108 that interfaces with a host, a back end module 110 that interfaces with the one or more non-volatile memory die 104, and various other modules that perform functions which will now be described in detail. A module may take the form of a packaged functional hardware unit designed for use with other components, a portion of a program code (e.g., software or firmware) executable by a (micro)processor or processing circuitry that usually performs a particular function of related functions, or a self-contained hardware or software component that interfaces with a larger system, for example. Modules of the controller 102 may include a memory backlog hint generator 111, which is discussed in more detail below, and can be implemented in hardware or software/firmware. The memory backlog hint generator 111 can be configured to perform the algorithms and methods discussed below and shown in the attached drawings.

[0042] Referring again to modules of the controller 102, a buffer manager/bus controller 114 manages buffers in random access memory (RAM) 116 and controls the internal bus arbitration of controller 102. A read only memory (ROM) 118 stores system

boot code. Although illustrated in Figure 2A as located separately from the controller 102, in other embodiments one or both of the RAM 116 and ROM 118 may be located within the controller. In yet other embodiments, portions of RAM and ROM may be located both within the controller 102 and outside the controller.

[0043] Front end module 108 includes a host interface 120 and a physical layer interface (PHY) 122 that provide the electrical interface with the host or next level storage controller. The choice of the type of host interface 120 can depend on the type of memory being used. Examples of host interfaces 120 include, but are not limited to, SATA, SATA Express, serially attached small computer system interface (SAS), Fibre Channel, universal serial bus (USB), PCIe, and NVMe. The host interface 120 typically facilitates transfer for data, control signals, and timing signals.

[0044] Back end module 110 includes an error correction code (ECC) engine 124 that encodes the data bytes received from the host, and decodes and error corrects the data bytes read from the non-volatile memory. A command sequencer 126 generates command sequences, such as program and erase command sequences, to be transmitted to non-volatile memory die 104. A RAID (Redundant Array of Independent Drives) module 128 manages generation of RAID parity and recovery of failed data. The RAID parity may be used as an additional level of integrity protection for the data being written into the memory device 104. In some cases, the RAID module 128 may be a part of the ECC engine 124. A memory interface 130 provides the command sequences to non-volatile memory die 104 and receives status information from non-volatile memory die 104. In one embodiment, memory interface 130 may be a double data rate (DDR) interface, such as a Toggle Mode 200, 400, or 800 interface. A flash control layer 132 controls the overall operation of back end module 110.

[0045] The storage system 100 also includes other discrete components 140, such as external electrical interfaces, external RAM, resistors, capacitors, or other components that may interface with controller 102. In alternative embodiments, one or more of the physical layer interface 122, RAID module 128, media management layer 138 and buffer management/bus controller 114 are optional components that are not necessary in the controller 102.

[0046] Figure 2B is a block diagram illustrating components of non-volatile memory die 104 in more detail. Non-volatile memory die 104 includes peripheral circuitry 141 and non-volatile memory array 142. Non-volatile memory array 142 includes the non-volatile memory cells used to store data. The non-volatile memory cells may be any suitable non-volatile memory cells, including ReRAM, MRAM, PCM, NAND flash memory cells and/or NOR flash memory cells in a two dimensional and/or three dimensional configuration. Non-volatile memory die 104 further includes a data cache 156 that caches data. Peripheral circuitry 141 includes a state machine 152 that provides status information to the controller 102.

[0047] In one embodiment, the memory 104 comprises memory cells that can be configured either as single level cells (SLC) or as multi-level cells (MLC), and that configuration can change when the storage system 100 in the field (i.e., after manufacturing). The configuration of some or all of the memory 104 as either SLC or MLC affects both the capacity of the memory 104 and the performance of the storage system 100. More specifically, MLC cells can store more data than SLC cells; however, it takes longer to program an MLC cell than an SLC cell, resulting in slower performance. For example, to store extra bits of data in an MLC cell, the MLC cell may need four units of charge on the floating gate in the memory cell during the same voltage programming window that an SLC uses. To do this, the MLC cell may require a more time-consuming programming algorithm, which makes programming an MLC cell up to four times slower than programming an SLC cell. Additionally, unlike an SLC cell, a MLC cell may not support copyback or partial programming, which also reduces system performance.

[0048] As mentioned above, manufacturers market storage systems with various characteristics, such as capacity and performance. Sometimes, the host device may be willing to trade off some capacity (e.g., using SLC cells instead of MLC cells, or foregoing garbage collection to free-up blocks of memory) to meet an urgent performance requirement. To do this, the host device can send a request to the storage system to alter the configuration of its memory cells. However, the host device does not know the internal details of the memory. Without this information, the host device makes a blind request assuming that the memory would be able to meet it. However, the

memory may or may not be able to meet such a requirement, and the host device stands the chance of not being able to leverage the capacity of the memory in the desired way. Hence, there is no guarantee that even if a special request is made by the host device, it will be fulfilled by the memory. To address this problem, in some systems, such as Flexible Flash File Systems, predetermined single level cells (SLC) and multi-level cells (MLC) can be changed dynamically. The storage system internally manages heterogeneous memories regions and uses complex firmware with data migration policy as it tries to make sure the capacity is intact. However, the complexity of the firmware can be disadvantageous.

[0049] The following embodiments can be used to address these issues. Turning again to the drawings, Figure 3 is a block diagram of a host 50 and storage system 100 of an embodiment. As shown in this drawing, in this embodiment, the storage system 100 provides a "hint" on the storage system's internal memory 104 and firmware states (e.g., via an acknowledgement channel specified by a protocol addendum). To support this functionality, the controller 102 (e.g., firmware) in the storage system 100 can be configured to realize its current memory state in terms of garbage collection handling, memory quality, and wear-levelling requirements to periodically make estimates on the workload.

[0050] The host 50 leverages this hint to evaluate and trigger a new capacity point that meets a needed performance requirement. Because of this interaction, the host 50 can dynamically influence the storage system 100 towards varying memory capacity to meet a performance requirement by sending a command or instruction (e.g., an administrative command) to the storage system 100 to reduce memory capacity in order to increase memory performance. In response to this instruction/command, the storage system 100 can make the relevant changes to memory operation to affect this change. For example, the controller 102 in the storage system 100 can configured some or all of the memory 104 as SLC cells instead of MLC cell, can forego or reduce garbage collection, etc.

[0051] With these embodiments, the storage system's controller 102 (e.g., the hint generator 111) can gather information that affects memory performance and generate a "hint" to provide an indication of backlog to the host 50 to enable the host 50 to make a better decision, if appropriate. This results in a simpler and more affordable storage

system than the one mentioned above that use complex firmware. Further, with these embodiments, the host 50 does not have to speculate on performance or capacity. The host 50 can use the hint to figure out the available capacity for a given performance requirement. That is, since only the storage system 100 is aware of its underlying memory resources and workload, a hint provided by the storage system 100 to the host 50 can be leveraged by the host to make a fair estimate of the amount of time it can make the memory 104 work at a particular performance mode. As another advantage, these embodiments can provide variable capacity of the memory 104 to provide a better guarantee for burst performance even towards the end of the life of the memory 104.

[0052] As indicated in Figure 3, the hint can be updated periodically. That way, if the state of the memory 104 changes over time, the storage system 100 can generate an updated hint to the host 50, so that it can tune the capacity/performance tradeoff and mapping. In this way, the host 50 can opt to continue a certain capacity for a long duration with periodic hints. Additionally, if the host 50 is connected to a plurality of storage systems that have the ability to generate and provide hints, the host 50 can make a consolidated call to map its requirement to the relevant storage system.

[0053] The "hint" can take any suitable form. For example, in one embodiment, the hint (or, more generally, "information regarding the state of the memory," can take the form of one or more of the following: a number of free blocks in the memory 104, a number of used blocks in the memory 104, a number of single level cell (SLC) blocks in the memory 104, a number of multi-level cell (MLC) blocks in the memory 104, a garbage collection workload, and backlog information (e.g., due to a previous burst mode request). As such, the hint/information will sometimes be described herein as a memory backlog hint and/or workload hint.

[0054] As other memory parameters can play a role in absolute latency in moving blocks (e.g., due to different trim parameters), information about those other parameters can be sent to the host 50 as well for use in its decision. Examples of such secondary information can include, but are not limited to, memory age, memory quality, and memory endurance. When such secondary information is sent to the host 50, the host 50 can base its capacity/performance decision both on the state of the memory 104 and on the secondary information.

[0055] It should be noted that it is not always practically possible for the host 50 to make the memory 104 work on a performance mode for a stipulated time without compromising the capacity point; at least it may not be guaranteed for long. The capacity of the memory 104 varies at various points as and when the host 50 makes a new mode request based on a hint from the storage system 100. As such, the capacity is dynamically defined and varied by the host 50. Once that mode is no longer possible (e.g., due to internal obligations, such as urgent compaction, wear leveling pending for longer than a threshold amount of time, long or urgent read scrubs, etc.), the storage system 100 can send an updated hint/information to the host 50 (if the host 50 has not already changed the mode based on its fair estimate). The host 50 can request a different capacity point (e.g., mapping to a lower performance mode) based on the new information of the memory's operating limit. The storage system 100 can send the updated information periodically, as a change happens, or at some other time interval.

[0056] Turning again to the drawings, the overall operation of the examples set forth above is shown in the flow diagram of Figure 4. As shown in Figure 4, first, the host 50 requests a performance mode (act 1). In one implementation, this is fixed per a requirement of a protocol being used. Next, the storage system 100 provides primary (and, optionally, secondary) hints to the host 50 (act 2). As mentioned above, the hints can be based on, for example, backlog work due to burst mode, garbage collection workload, and/or memory health). Based on this information, the host 50 can present a different capacity point to attempt to reach its desired performance requirement (act 3). In this example, acts 2 and 3 involve periodically-changing variables according to the requested performance in act 1. Further, the state machine/generator 111 in the storage system 100 that generates the hints continues until the host 50 or storages system 100 resets it, such as when the host 50 ignores a hint.

[0057] Further, to help with performance when SLC cells are used, the storage system 100 can freeze some of its activities. This will be discussed in conjunction with Figure 5. Figure 5 shows a diagram 500 that illustrates various acts that are taking place in response to an SLC write operation. These acts include Flash Translation Layer (FTL) updates, pseudo/hybrid/direct write, MLC write/fold, SLC compaction, hybrid compaction, MLC compaction, FTL compaction, and wear leveling/read scrub/other

work. Although not shown, balanced garbage collection can occur along with the writes. To increase performance even when SLC cells are used, the storage system 100 can freeze some of these activities. For example, for better performance, the storage system 100 can suspend garbage collection, wear levelling, and/or read scrub for some period of time.

[0058] In some cases, the host 50 may want a bumped-up performance for an amount of time for a given piece of data. The host 50 could have such a requirement for processing edge decisions for artificial intelligent or machine learning for certain high-end gaming applications. The host's performance requirement may be based on learning from those applications needing it. For example, if the host 50 is handling multiple applications, based on learning or just pure history, it may know the performance requirement for the applications under use. The host 50 can put to use the memory backlog hint (and, optionally, the second information), along with its own learning, to determine the performance and the capacity point for further use of storage system 100 at a given point of time for multiple such applications.

[0059] There are many other applications that can benefit from these embodiments. For example, as noted above, using these embodiments to leverage a larger capacity of a storage system can allow processing edge decisions fast enough for artificial intelligence / machine learning applications irrespective of the state of the memory 104. As another example, these embodiments can be used by Non-Volatile Memory Express (NVMe) devices to support multiple physical stores to make use of different media to provide different qualities of service for certain logical block addresses (LBAs)/namespaces.

[0060] The following paragraphs describe other example use cases. In one example use case, a retail storage system with one form factor can be used in various performance modes based on the memory backlog hint. For example, a 400 gigabyte (GB) Secure Digital (SD) card can be used in default mode and in a high-performance mode capable of 4K video recording through a host command that is based on a memory backlog hint from the SD card. By providing an interface channel to communicate the memory hint, these embodiments can be used to leverage a larger memory capacity, making sure that endurance and other factors are met per specification of the SD card.

[0061] In another use case, these embodiments are used to ensure that processing edge decisions are fast enough for artificial intelligence and machine learning. For example, when a machine learning application is utilizing the memory and needs a momentary performance burst, even it means at a cost of capacity, these embodiments can provide that without speculation on performance for that capacity. Further, as mentioned above, in the case of a storage pool, if the host has access to multiple storage systems, with backlog hints from each of them, the host can decisively decide the next correct course of action.

[0062] It should be noted that these embodiments provide extra performance without the need for dynamic overprovisioning of blocks of memory. The extra performance provided by these embodiments is achieved through lowering the capacity point itself of the memory. The lower capacity point can be brought back if the host requests by requesting a lower-than-average performance for subsequent commands. This is a win-win situation since there may not be a top performance gear requirement all the time. With the host having full control owing to backlog hinting, the host can work around multiple performance limitations through leveraging capacity and not through any overprovisioning of blocks.

[0063] There are several alternatives that can be used with these embodiments. For example, instead of reducing capacity for increased performance, these embodiments can be used to increase capacity for decreased performance. Accordingly, these embodiments can be said to alter capacity to alter performance.

[0064] Finally, as mentioned above, any suitable type of memory can be used. Semiconductor memory devices include volatile memory devices, such as dynamic random access memory ("DRAM") or static random access memory ("SRAM") devices, non-volatile memory devices, such as resistive random access memory ("ReRAM"), electrically erasable programmable read only memory ("EEPROM"), flash memory (which can also be considered a subset of EEPROM), ferroelectric random access memory ("FRAM"), and magnetoresistive random access memory ("MRAM"), and other semiconductor elements capable of storing information. Each type of memory device may have different configurations. For example, flash memory devices may be configured in a NAND or a NOR configuration.

[0065] The memory devices can be formed from passive and/or active elements, in any combinations. By way of non-limiting example, passive semiconductor memory elements include ReRAM device elements, which in some embodiments include a resistivity switching storage element, such as an anti-fuse, phase change material, etc., and optionally a steering element, such as a diode, etc. Further by way of non-limiting example, active semiconductor memory elements include EEPROM and flash memory device elements, which in some embodiments include elements containing a charge storage region, such as a floating gate, conductive nanoparticles, or a charge storage dielectric material.

[0066] Multiple memory elements may be configured so that they are connected in series or so that each element is individually accessible. By way of non-limiting example, flash memory devices in a NAND configuration (NAND memory) typically contain memory elements connected in series. A NAND memory array may be configured so that the array is composed of multiple strings of memory in which a string is composed of multiple memory elements sharing a single bit line and accessed as a group. Alternatively, memory elements may be configured so that each element is individually accessible, e.g., a NOR memory array. NAND and NOR memory configurations are examples, and memory elements may be otherwise configured.

[0067] The semiconductor memory elements located within and/or over a substrate may be arranged in two or three dimensions, such as a two dimensional memory structure or a three dimensional memory structure.

[0068] In a two dimensional memory structure, the semiconductor memory elements are arranged in a single plane or a single memory device level. Typically, in a two dimensional memory structure, memory elements are arranged in a plane (e.g., in an x-z direction plane) which extends substantially parallel to a major surface of a substrate that supports the memory elements. The substrate may be a wafer over or in which the layer of the memory elements are formed or it may be a carrier substrate which is attached to the memory elements after they are formed. As a non-limiting example, the substrate may include a semiconductor such as silicon.

[0069] The memory elements may be arranged in the single memory device level in an ordered array, such as in a plurality of rows and/or columns. However, the memory

elements may be arrayed in non-regular or non-orthogonal configurations. The memory elements may each have two or more electrodes or contact lines, such as bit lines and wordlines.

[0070] A three dimensional memory array is arranged so that memory elements occupy multiple planes or multiple memory device levels, thereby forming a structure in three dimensions (i.e., in the x, y and z directions, where the y direction is substantially perpendicular and the x and z directions are substantially parallel to the major surface of the substrate).

[0071] As a non-limiting example, a three dimensional memory structure may be vertically arranged as a stack of multiple two dimensional memory device levels. As another non-limiting example, a three dimensional memory array may be arranged as multiple vertical columns (e.g., columns extending substantially perpendicular to the major surface of the substrate, i.e., in the y direction) with each column having multiple memory elements in each column. The columns may be arranged in a two dimensional configuration, e.g., in an x-z plane, resulting in a three dimensional arrangement of memory elements with elements on multiple vertically stacked memory planes. Other configurations of memory elements in three dimensions can also constitute a three dimensional memory array.

[0072] By way of non-limiting example, in a three dimensional NAND memory array, the memory elements may be coupled together to form a NAND string within a single horizontal (e.g., x-z) memory device levels. Alternatively, the memory elements may be coupled together to form a vertical NAND string that traverses across multiple horizontal memory device levels. Other three dimensional configurations can be envisioned wherein some NAND strings contain memory elements in a single memory level while other strings contain memory elements which span through multiple memory levels. Three dimensional memory arrays may also be designed in a NOR configuration and in a ReRAM configuration.

[0073] Typically, in a monolithic three dimensional memory array, one or more memory device levels are formed above a single substrate. Optionally, the monolithic three dimensional memory array may also have one or more memory layers at least partially within the single substrate. As a non-limiting example, the substrate may

include a semiconductor such as silicon. In a monolithic three dimensional array, the layers constituting each memory device level of the array are typically formed on the layers of the underlying memory device levels of the array. However, layers of adjacent memory device levels of a monolithic three dimensional memory array may be shared or have intervening layers between memory device levels.

[0074] Then again, two dimensional arrays may be formed separately and then packaged together to form a non-monolithic memory device having multiple layers of memory. For example, non-monolithic stacked memories can be constructed by forming memory levels on separate substrates and then stacking the memory levels atop each other. The substrates may be thinned or removed from the memory device levels before stacking, but as the memory device levels are initially formed over separate substrates, the resulting memory arrays are not monolithic three dimensional memory arrays. Further, multiple two dimensional memory arrays or three dimensional memory arrays (monolithic or non-monolithic) may be formed on separate chips and then packaged together to form a stacked-chip memory device.

[0075] Associated circuitry is typically required for operation of the memory elements and for communication with the memory elements. As non-limiting examples, memory devices may have circuitry used for controlling and driving memory elements to accomplish functions such as programming and reading. This associated circuitry may be on the same substrate as the memory elements and/or on a separate substrate. For example, a controller for memory read-write operations may be located on a separate controller chip and/or on the same substrate as the memory elements.

[0076] One of skill in the art will recognize that this invention is not limited to the two dimensional and three dimensional structures described but cover all relevant memory structures within the spirit and scope of the invention as described herein and as understood by one of skill in the art.

[0077] It is intended that the foregoing detailed description be understood as an illustration of selected forms that the invention can take and not as a definition of the invention. It is only the following claims, including all equivalents, that are intended to define the scope of the claimed invention. Finally, it should be noted that any aspect of

any of the embodiments described herein can be used alone or in combination with one another.

What is claimed is:

1. A method for memory backlog hinting for variable capacity, the method comprising:

performing the following in a storage system comprising a memory:

sending information regarding a state of the memory to a host device;

receiving an instruction from the host device to alter memory capacity in order to alter memory performance, wherein the instruction is based on the information regarding the state of the memory sent to the host device; and

altering memory capacity in order to alter memory performance in response to receiving the instruction from the host device.

2. The method of Claim 1, wherein the information regarding the state of the memory comprises one or more of the following: a number of free blocks in the memory, a number of used blocks in the memory, a number of single level cell (SLC) blocks in the memory, a number of multi-level cell (MLC) blocks in the memory, a garbage collection workload, and backlog information.

3. The method of Claim 2, further comprising sending one or more of the following secondary information to the host device: memory age, memory quality, and memory endurance, wherein the instruction from the host device is also based on the secondary information.

4. The method of Claim 1, further comprising sending updated information about an updated state of the memory to the host device.

5. The method of Claim 4, wherein the updated state indicates one or more of the following: urgent compaction, a wear level operation pending for longer than a threshold amount of time, and an urgent read scrub command.

6. The method of Claim 4, further comprising periodically sending the updated information to the host device.

7. A storage system comprising:

a memory; and

a controller configured to be in communication with the memory, wherein the controller is further configured to:

provide a hint to a host device on a condition of the memory;

receive a command from the host device to alter memory capacity in order to alter memory performance, wherein the command is in response to the hint provided to the host device; and

alter memory capacity in order to alter memory performance in response to receiving the command from the host device.

8. The storage system of Claim 7, wherein the information regarding the state of the memory comprises one or more of the following: a number of free blocks in the memory,

a number of used blocks in the memory, a number of single level cell (SLC) blocks in the memory, a number of multi-level cell (MLC) blocks in the memory, a garbage collection workload, and backlog information.

9. The storage system of Claim 8, wherein the controller is further configured to send one or more of the following secondary hints to the host device: memory age, memory quality, and memory endurance, wherein the command from the host device is also in response to the secondary information.

10. The storage system of Claim 7, wherein the controller is further configured to send an updated hint to the host device.

11. The storage system of Claim 10, wherein the updated hint indicates one or more of the following: urgent compaction, a wear level operation pending for longer than a threshold amount of time, and an urgent read scrub command.

12. The storage system of Claim 10, wherein the controller is further configured to periodically send the updated hint to the host device.

13. The storage system of Claim 7, wherein the memory comprises a three-dimensional memory.

14. The storage system of Claim 7, wherein the storage system is configured to be integrated in the host device.

15. The storage system of Claim 7, wherein the storage system is configured to be removably connected with the host device.

16. A storage system comprising:

a memory;

means for sending information regarding a state of the memory to a host device;

means for receiving an instruction from the host device to alter memory capacity in order to alter memory performance, wherein the instruction is based on the information regarding the state of the memory sent to the host device; and

means for altering memory capacity in order to alter memory performance in response to receiving the instruction from the host device.

17. The storage system of Claim 16, wherein the information regarding the state of the memory comprises one or more of the following: a number of free blocks in the memory, a number of used blocks in the memory, a number of single level cell (SLC) blocks in the memory, a number of multi-level cell (MLC) blocks in the memory, a garbage collection workload, and backlog information.

18. The storage system of Claim 17, further comprising means for sending one or more of the following secondary information to the host device: memory age, memory

quality, and memory endurance, wherein the instruction from the host device is also based on the secondary information.

19. The storage system of Claim 16, further comprising means for sending updated information about an updated state of the memory to the host device.

20. The storage system of Claim 19, further comprising means for periodically sending the updated information to the host device.

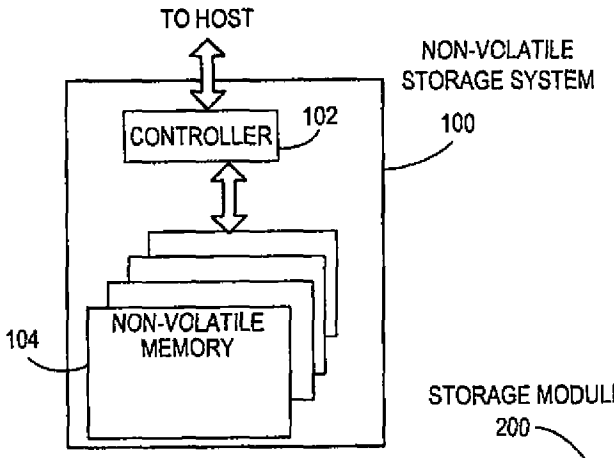

FIG. 1A

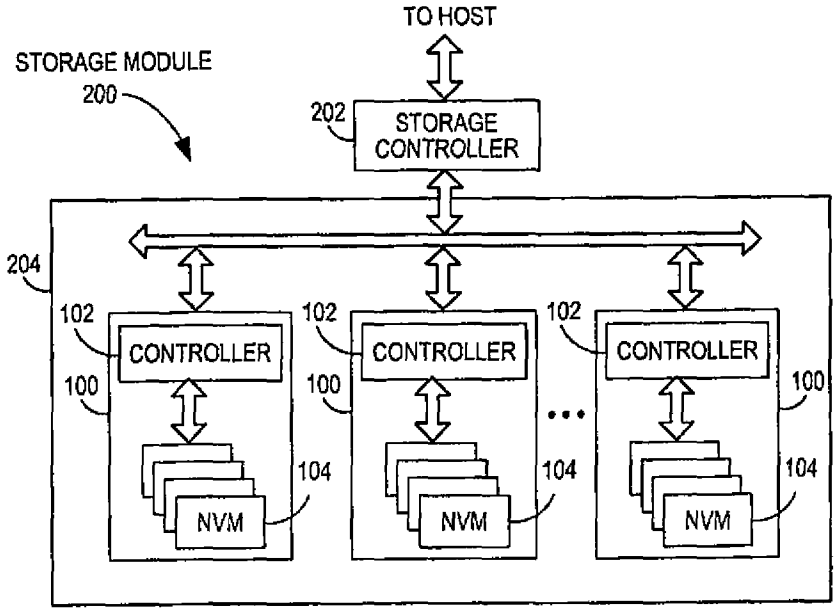

FIG. 1B

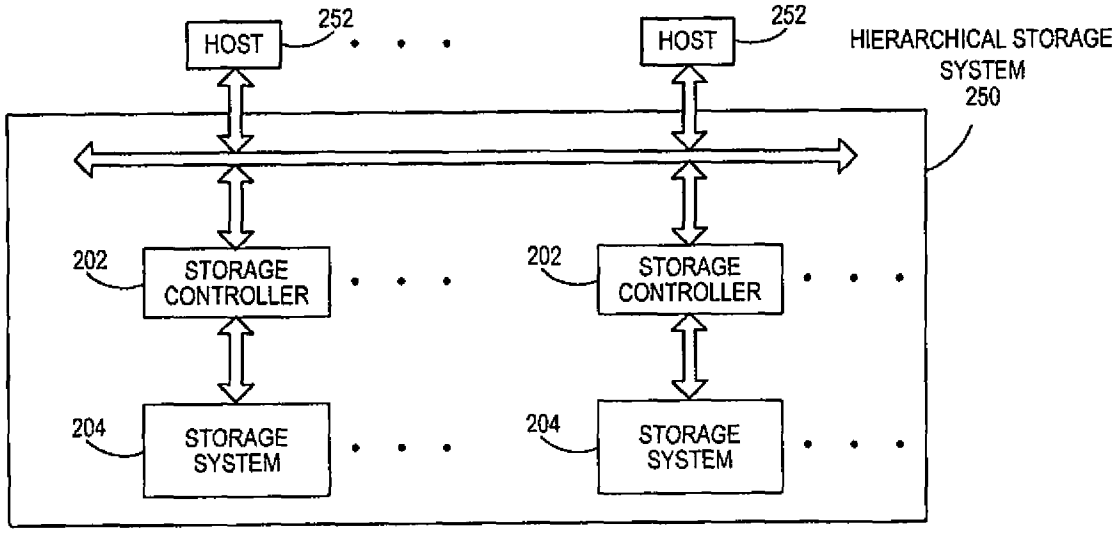

FIG. 1C

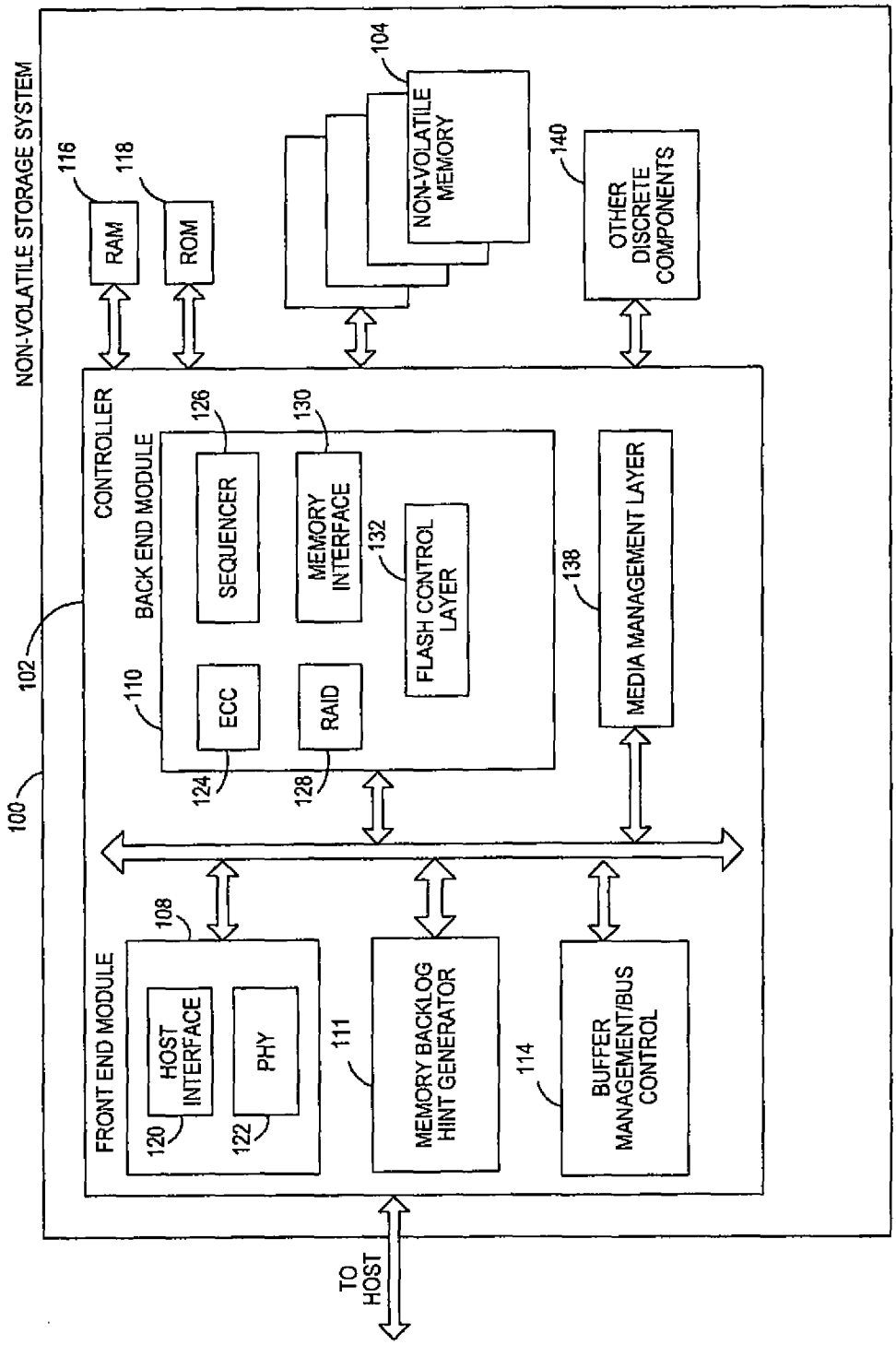
NON-VOLATILE STORAGE SYSTEM
100
102
CONTROLLER
FRONT END MODULE
108
HOST INTERFACE
120
PHY
122
111
MEMORY BACKLOG HINT GENERATOR
114
BUFFER MANAGEMENT/BUS CONTROL
BACK END MODULE
110
ECC
124
RAID
128
SEQUENCER
126
MEMORY INTERFACE
130
FLASH CONTROL LAYER
132
138
MEDIA MANAGEMENT LAYER
RAM
116
ROM
118
NON-VOLATILE MEMORY
104
OTHER DISCRETE COMPONENTS
140
TO HOST

FIG. 2A

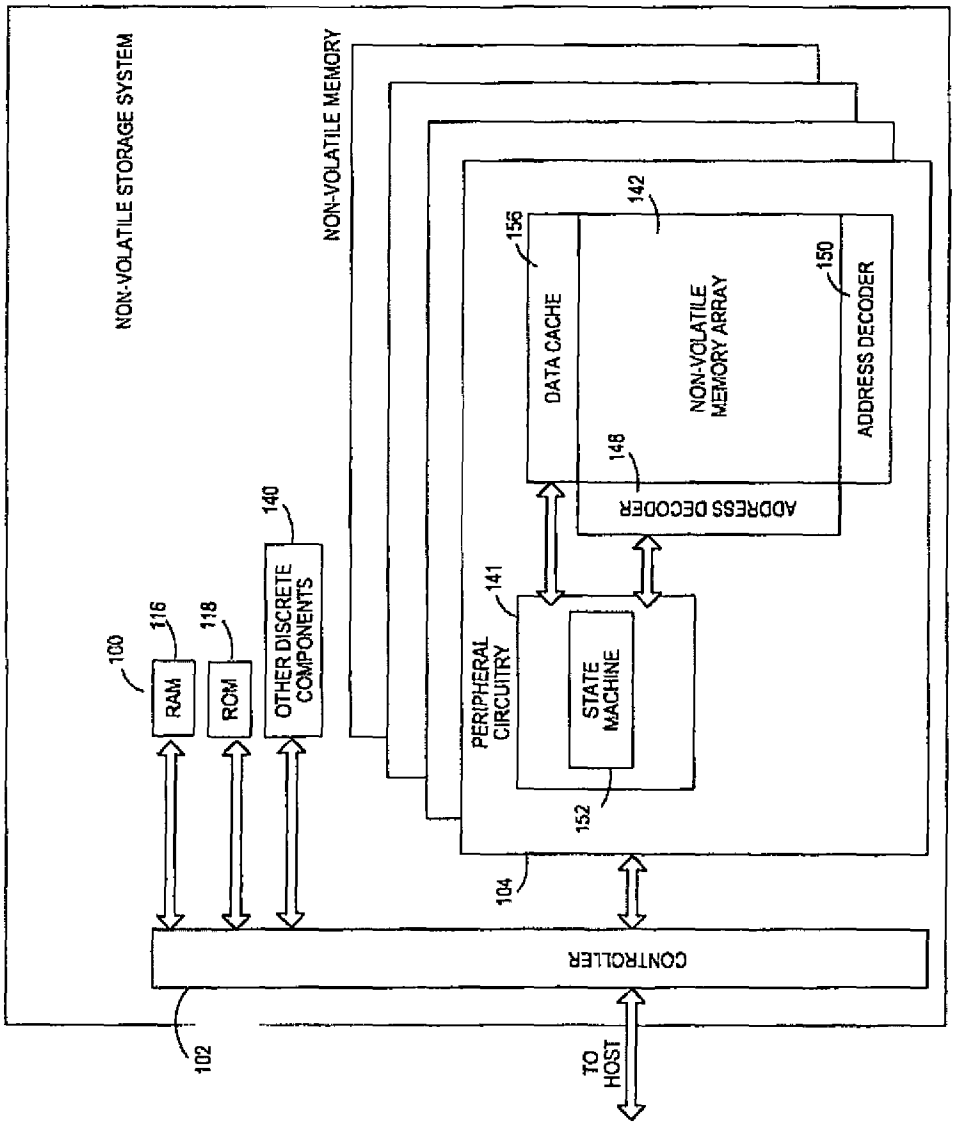
NON-VOLATILE STORAGE SYSTEM
NON-VOLATILE MEMORY
100
116
RAM
118
ROM
140
OTHER DISCRETE COMPONENTS
142
156
DATA CACHE
NON-VOLATILE MEMORY ARRAY
150
ADDRESS DECODER
148
ADDRESS DECODER
141
PERIPHERAL CIRCUITRY
STATE MACHINE
152
104
CONTROLLER
102
TO HOST

FIG. 2B

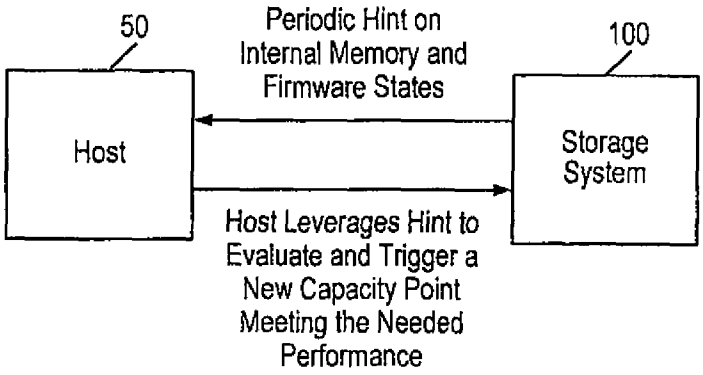
50
Periodic Hint on
Internal Memory and
Firmware States
100
Host
Storage
System
Host Leverages Hint to
Evaluate and Trigger a
New Capacity Point
Meeting the Needed
Performance

FIG. 3

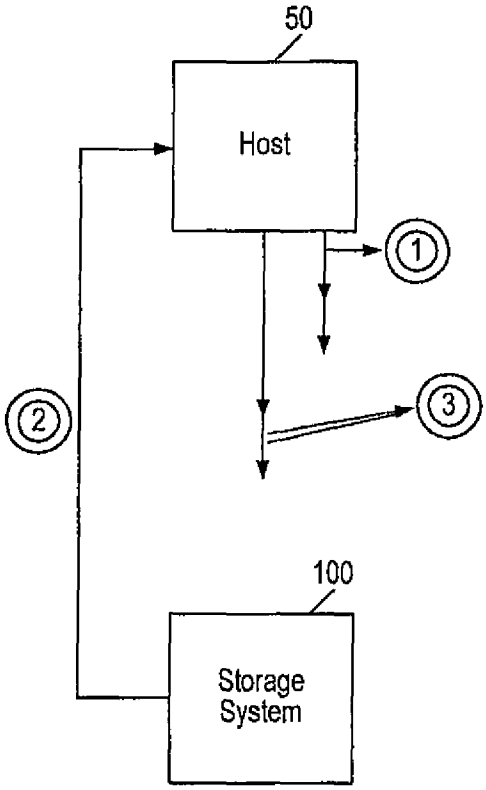

FIG. 4

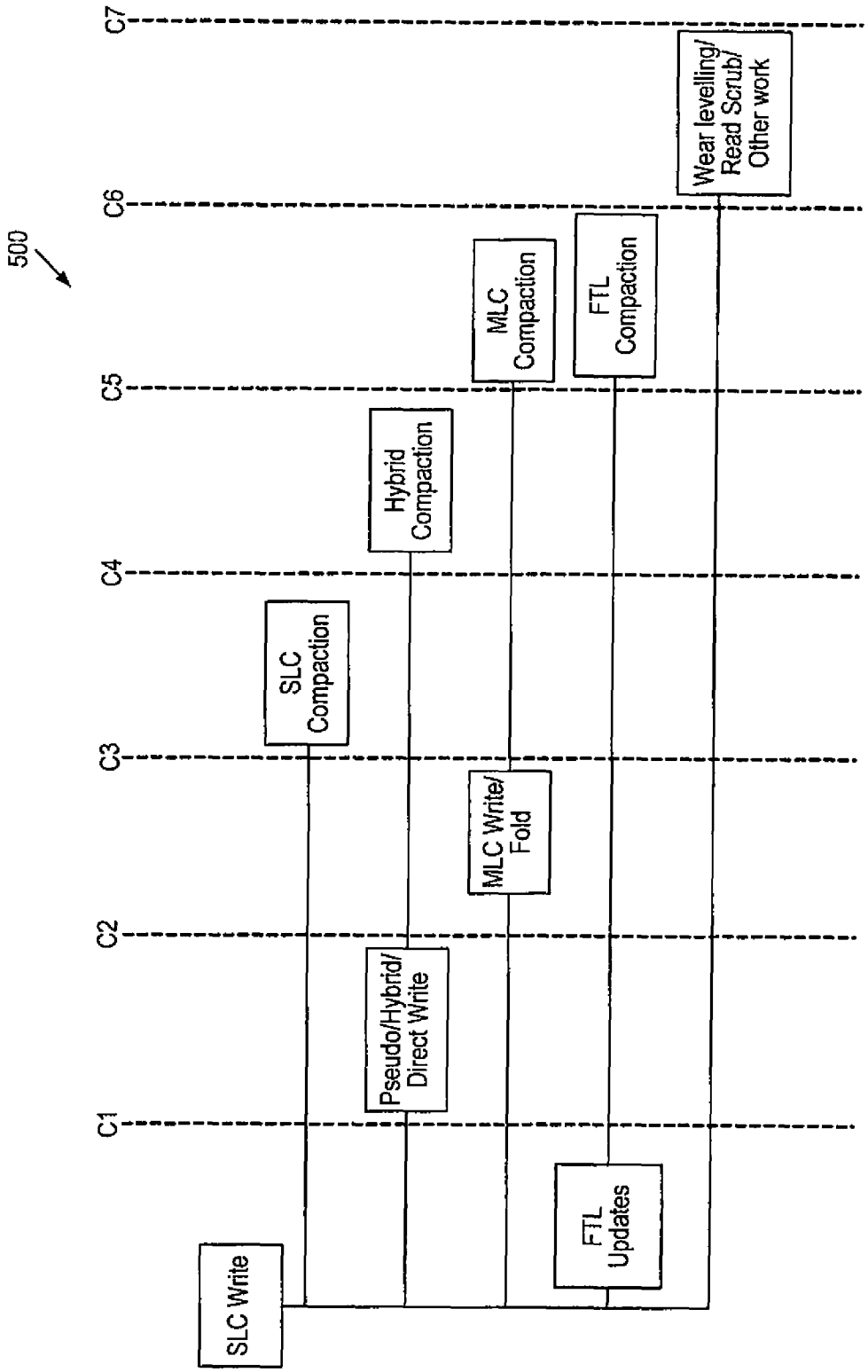
500
C7
C6
C5
C4
C3
C2
C1
Wear levelling/ Read Scrub/ Other work
MLC Compaction
FTL Compaction
Hybrid Compaction
SLC Compaction
MLC Write/ Fold
Pseudo/Hybrid/ Direct Write
FTL Updates
SLC Write

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2019/067398

A. CLASSIFICATION OF SUBJECT MATTER
IPC (20200101) G06F 3/06
CPC (20130101) G06F 3/0659, G11C 2211/5641, G06F 2212/7206, G06F 3/0631
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)
IPC (20200101) G06F 3/06
CPC (20130101) G06F 3/0659, G11C 2211/5641, G06F 2212/7206, G06F 3/0631

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)
Databases consulted: Esp@cenet
Search terms used: SLC binary bit single bit feedback host

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2018081594 A1 SAMSUNG ELECTRONICS CO.,LTD 22 Mar 2018 (2018/03/22) ¶¶46,97-101	1-5,7-11,13-19
X	US 2008250220 A1 Toshiba Corp 09 Oct 2008 (2008/10/09) abstract, ¶¶ 117,128,130,147	1,7,16
A	fig.6	6,12,20
A	US 2012005404 A1 SANDISK IL LTD., KFAR SABA (IL) 05 Jan 2012 (2012/01/05) abstract, ¶2	5,11

☐ 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
"D" document cited by the applicant in the international application
"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	Date of mailing of the international search report
23 Apr 2020	23 Apr 2020

Name and mailing address of the ISA:	Authorized officer
Israel Patent Office Technology Park, Bldg.5, Malcha, Jerusalem, 9695101, Israel Email address: pctoffice@justice.gov.il	MAUDA Nissim Telephone No. 972-73-3927247

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2019/067398

Patent document cited search report	Publication date	Patent family member(s)	Publication Date
US 2018081594 A1	22 Mar 2018	US 2018081594 A1	22 Mar 2018
		CN 107845394 A	27 Mar 2018
		KR 20180031289 A	28 Mar 2018
		TW 201814528 A	16 Apr 2018
US 2008250220 A1	09 Oct 2008	US 2008250220 A1	09 Oct 2008
		JP 2008276733 A	13 Nov 2008
US 2012005404 A1	05 Jan 2012	US 2012005404 A1	05 Jan 2012
		US 8683148 B2	25 Mar 2014
		TW 201209593 A	01 Mar 2012
		WO 2012001479 A1	05 Jan 2012