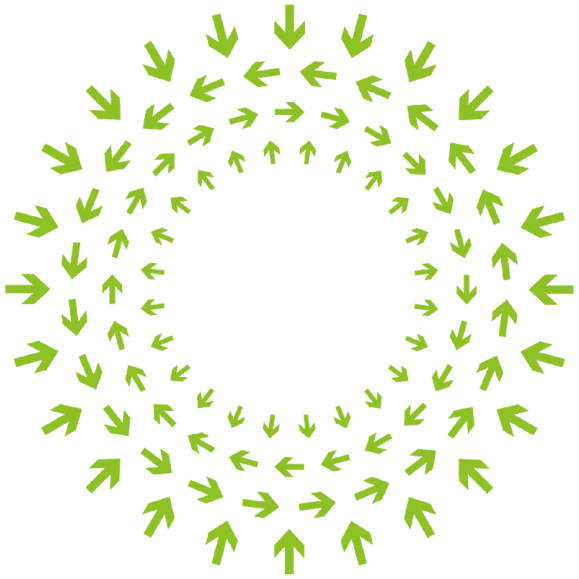


OPEN

Compute Project

Cloud HDD - Fast Fail Read

Rev 1.0d

Date: Jun 7, 2018

Co-Authors are listed on the second page

Joint Development License

The following organizations have executed the Joint Development Agreement:

- **Broadcom Inc**
- **Facebook, Inc**
- **Google LLC**
- **Huawei Technologies Co., Ltd**
- **Microsoft Corp**
- **Seagate Technology PLC**
- **Toshiba Corp**
- **Western Digital Corp**

A list of the corresponding contributing individuals, representing each company, can be found here.

License

Contributions to this Specification are made under the terms and conditions set forth in **Open Web Foundation Contributor License Agreement** ("Contribution License") by:

- **Broadcom Inc**
- **Facebook, Inc**
- **Google LLC**
- **Huawei Technologies Co., Ltd**
- **Microsoft Corp**
- **Seagate Technology PLC**
- **Toshiba Corp**
- **Western Digital Corp**

You can review the signed copies of the applicable Contributor License(s) for this Specification on the OCP website at http://www.opencompute.org/products/specsanddesign

Usage of this Specification is governed by the terms and conditions set forth in **Open Web Foundation Final Specification Agreement ("OWFa 1.0")** ("Specification License").

You can review the applicable Specification License(s) executed by the above referenced contributors to this Specification on the OCP website at http://www.opencompute.org/participate/legal-documents/

Note: The following clarifications, which distinguish technology licensed in the Contribution License and/or Specification License from those technologies merely referenced (but not licensed), were accepted by the Incubation Committee of the OCP:

None.

Revision History

Rev 1.0a	Apr 12, 2018	Initial Co-authored Draft
Rev 1.0b	Apr 27, 2018	First Major Document Clean-up
Rev 1.0c	Jun 7, 2018	Final (Minor) Document clean-up
Rev 1.0d	Jun 13, 2018	Added NCQ Clarification

Table of Contents

High-level Overview

In a typical large-scale distributed file system, a piece of data is stored across multiple HDDs. Thus, if a given HDD cannot complete a read request quickly, it is often much better for that HDD to abandon the read request, and have the distributed file system read from another HDD instead.

“Fast Fail Read” is the very first product feature in a family of OCP Cloud-optimized HDD products, designed to specifically address the desire described above. It is designed to be plugged into and work OCP HDD Trays, Servers, and Systems. An example of one such OCP HDD tray would be the Bryce Canyon Storage Server.

Fast Fail Read Scope

The scope of this product specification is to cover the command interface needed to achieve the goal of enabling fast-fail read commands to be sent from a distributed file system host, down to the Cloud HDD device.

Note that the followings are specifically out of scope under this proposal:

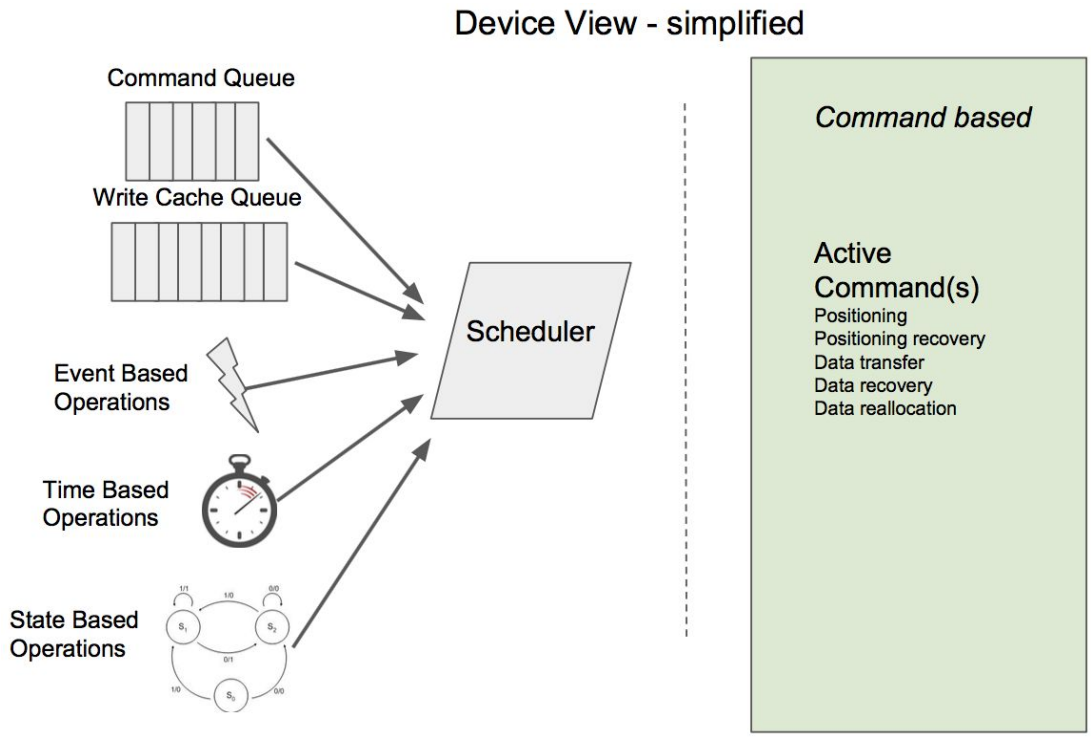
- Changes to (generic) command queueing and/or caching in order to support different read traffic classes or quota system, as well as changes to HDD queue management in general
- Generic host management of disk background commands, tasks, or services
- Generic logging or health monitoring

Device Level Behavior

Introduction to Two Additive Timeouts

The Cloud HDD device supports the control of read command latency with two separate and independent timeout values: Inactive Command Limit and Active Command Limit. The Inactive Command Limit mechanism constrains the amount of time that a command can exist after reception and prior to activation. Commands may be inactive for a variety of reasons such as queuing, writeback cache operations, and internal operations. The Active Command Limit mechanism constrains the time consumed by the command while active or executing. For a Cloud HDD, active time includes the time to access the media (i.e. seek and rotational latency), transfer time, servo and media recovery.

The illustration below defines the domains of the command limit values. Left of the dashed line depicts the domain of the Inactive Command Limit while the right side depicts the domain of the Active Command Limit.



The purpose of utilizing two time limit values is to provide the system with the ability to determine the acceptable amount of command execution time and corresponding error recovery. If a single all encompassing limit value was used, the device would internally decide when to schedule a command and the corresponding time available for execution. A device based decision would not allow systems to tailor the amount of command execution to their system needs such as acceptable Unrecoverable Error Rate (UER).

Systems shall tolerate command completion times that marginally exceed the aggregate of the limit values as acceptable device behavior. The HDD FW must meet this absolute requirement: Within 5x 9's

of the number of read IOs with the Active Command Limit Set, the HDD FW must adhere and fail an IO within the active command limit set, by no more than a +/- 20ms error.

The expected command failure rate is expected to scale as a function of the limit value. Smaller limit values will likely result in higher failure rates. It is the responsibility of the system to properly trade-off control of command latency and command failure rates.

The Cloud HDD device shall not count the Inactive Command Limit or the Active Command Limit failures towards SMART trips or related drive health parametrics.

Inactive Command Limit

The Inactive Command Limit specifies the maximum amount of time a read command may persist in an inactive state. Inactive means that no data has been read or written to the media. The Inactive Command Limit is a per command timeout value. The full definition of this mechanism is beyond the scope of this document. Some examples of the protocol used to specify the Inactive Command Limit are Isochronous Command Completion (ICC) for SATA and Duration Limited Commands for SAS.

It is noted that commands may readily time out if this mechanism is used during certain HDD states such as an active firmware download or power mode transitions.

Active Command Limit

The Active Command Limit specifies the maximum time spent executing the read command. If multiple media operations are used to fulfill completion, the aggregate time spent shall be constrained by the limit value. The Active Command Limit is a per command timeout value. The protocol for SAS and SATA shall fulfill the following requirements:

- Read commands have the ability to indicate a limit value
- One to N per command limit values are supported
- If no limit value is indicated, the global error recovery limits are applied
- The limit value is time based
- Commands shall indicate a failure status upon encountering the time limit
- Logging counters reflecting the number of commands that exceeded each unique command time limit

The proposed command interface for the Active Command Limit is described in more details in the Command Interface and Behavior section.

Fast-Fail Reads and Normal Reads

Through the introduction of the Active Command Limit, “fast fail reads” can be achieved by setting a tight Active Command Limit. “Normal reads” can be achieved by having extremely relaxed or no Active Command Limit.

Limitation and Extent of Device Responsibility

Inactive Command Limit behaviors are out of scope for this document.

In terms of Active Command Limit, it is the sole responsibility of the host to set the proper command limit values, and it is the sole responsibility of the device to guarantee the appropriate Cloud HDD read bit error rate (or durability) characteristic when an Active Command Limit is not set for reads. For clarity, read bit error rate here refers to actual uncorrectable media read errors, and not Active or Inactive Command Limit timeouts for example.

For workloads where the fast-fail reads become the dominating read method, it is important that the Cloud HDD continues to maintain the same bit error rate specification for the rare normal reads. In other words, the HDD FW needs to have the proper sets of background operations and capabilities to ensure that the same Durability is maintained as normal reads are replaced by fast-fail reads.

In the event that the host sets unreasonable or nonoptimal command limit values, the storage controller and the device shall not do anything to correct mistakes that the host has made.

Finally, it is important that the Cloud HDD optimizes the Active Command Limit portion of both the fast-fail reads and the normal reads as equivalent ‘Fast Paths” for IO accesses in terms of code paths. In addition, the Cloud HDD should ensure that any ratio of fast-fail to normal read mixes (whether this is a normal service or a massive maintenance excursion) can be handled equally efficiently, without any performance or reliability degradation.

Host (and HBA Controller) Level Behavior

Expected Host Usage

In a typical large-scale distributed file system, a piece of data is stored across multiple HDDs. Thus, if a given HDD cannot complete a read request quickly, it is often much better for that HDD to abandon the read request, and have the distributed file system read from another HDD instead.

The host is expected to issue “fast-fail reads” for read accesses where the host cares about the command completion latency. If a fast-read were to fail (due to any reason, including read retry or high priority firmware background task interrupt), the host can choose to use data from other alternative Cloud HDDs instead. In the event that a mandatory read must be conducted from a particular Cloud HDD (such as a maintenance operation), the host can choose to issue a normal read operation at that point. From an optimal host performance perspective, having the ability for a fast-fail read failure (abort) to be self-contained without affecting the rest of the queued up IOs will be extremely important, from the interface perspective.

Please note that only a fast-fail read operation is desired for Cloud HDD. There is no practical use or desire for a “fast-fail write” operation at this point, and so this doc does not propose the need for fast-fail write commands.

Host usage behaviors and expectations around the inactive command limit are specifically out of scope in this document, and will be covered in a future Cloud HDD Specification instead.

Limitation and Extent of Host Responsibility

It is the sole responsibility of the host software and file system software to set reasonable command limit values that provide a balance between the ordinary execution time (e.g. seek, latency, transfer time) in addition to the amount of desired extraordinary time (e.g. read retries). Limit values that approach the tail of the distribution of ordinary execution times will result in a high rate of failures, and the device will execute based on these command limit settings at face-value.

It is highly advised that a proper study and characterization of the distributed file system and the individual storage nodes be done by the Cloud HDD user, before the desired command limit values and the distribution of regular to fast-fail reads are set in the host software.

Limitation and Extent of HBA Controller Responsibility

Similarly, it is the sole responsibility of the HBA (Host-Bus-Adaptor) Storage controller and its associated driver to ensure that the command priorities (fast-fail or regular) and the associated command limit values shall be maintained if command coalescence, fragmentation, or reprioritization were to take place within the controller.

Here is an explicit example of what the HBA Controller shall not do: The host issues two reads. The first is a 4KB fast-fail read at LBA location X. The second is a 4KB normal read at the next 4KB LBA location. The HBA Controller must ensure that it does *not* merge these two read commands into a single 8KB read command, as two read commands with different active command limits are not “coalescable”.

Lastly, similar to the Cloud HDD device itself, the HBA Controller should optimize both the fast-fail reads and the normal reads as equivalent ‘Fast Paths” for IO accesses. Any ratio of fast-fail to normal read

mixes (whether this is a normal service or a massive maintenance excursion) should be handled equally efficiently, without any performance or reliability degradation.

Command Interface and Behavior

Desire for T13, SATA-IO, and T10

The essential elements of the protocol include a means of specifying for each IO command two command limit timer values – active and inactive as defined above, and a specification of the policy to be employed upon expiration of each such timer respectively.

It is desired that the implementation across standards be semantically similar. Due to limitations in the available (currently reserved) bits in the various command structures the approach taken will be to define an array of descriptors in a log page (ATA) or a mode page (SCSI). Each such descriptor will contain the active timer, the active time policy, the inactive timer, and the inactive timer policy. This array of descriptors will be indexed into by means of a field within the command structures. Using this index within the command structure, the host may specify which of the several descriptors – and thereby the timer values (active and inactive) and timer expiration policies (active and inactive) – shall be applied by the device to that particular command. One may note that this is the general template employed in SCSI Command Duration Limits.

Required enumerated values for policy to enact upon expiration of the Inactive Command Limit include:

- no action (no command limit policy)
- best effort (HDD tries its best to meet the command limit, but this is a hint and not a hard limit)
- abort (HDD must strictly comply with the command limit)

Required enumerated values for policy to enact upon expiration of the Active Command Limit include:

- no action (no command limit policy)
- abort (HDD must strictly comply with the command limit)

The various statistics are maintained within new log pages.

Finally, a new unique additional sense code will need to be defined to indicate an Active Command Limit timeout expiration. Similarly, one or more new unique additional sense codes may also be desired for the Inactive Command Limit, but the exact details are out of scope for this document.

Cloud-HDD Mode

Given that Cloud-HDD is a new product concept, it is desired to be able to set an HDD into this unique Cloud-HDD mode.

For SATA Non-Queued, support for the Cloud-HDD Fast-Fail Read feature will be indicated by a bit in the Identify Device log data. This feature will be enabled or disabled by the use of a new SET FEATURES subcommand.

Similarly, for SATA Queued, support for the Cloud-HDD Fast-Fail Read feature will be indicated by a bit in the Identify Device log data. This feature will be enabled or disabled by the use of a new SET FEATURES subcommand.

Finally, for SAS Queued, support for the Cloud-HDD Fast-Fail Read feature will be enabled or disabled by the value of a new mode bit. If set to 1b, the device shall enable this feature, and disable Command Duration Limits. If set to 0b, the device shall disable this feature, and if supported enable Command Duration Limits.

SATA Non-Queued

READ DMA EXT (and optionally WRITE DMA EXT if standards body desires) will be enhanced to support Fast Fail. The array of descriptors shall be indexed by the value in a three-bit field (currently reserved). A value of 00h in this field shall indicate the null Fast Fail descriptor (i.e., no active limit or inactive limit) is specified for this command.

SATA Queued

A three-bit field of each IO command shall specify the index into the array of descriptors that the device shall apply to that command. A value of 00h in this field shall indicate the null Fast Fail descriptor (i.e., no active limit or inactive limit) is specified for this command. An enumerated value (previously a reserved value) of the PRIO field shall specify that a Fast Fail descriptor is specified for this command.

It is worth noting that there is a strong desire for Cloud HDD to have the ability to abort a single IO at a time without aborting the entire queue. This is because the TCO optimal fast fail read operating point for Cloud HDD is likely to be one where fast fail read aborts are significantly more common. However, the exact interface change needed on that front is complex and beyond the scope of this document.

Given this document is not proposing that the actual SATA Queued IO cancellation behavior be changed, if the HDD user has a desire to cancel/abort individual IOs (due to any reason, including exceeding the fast fail active time limit), the HDD user will need to work around the SATA Queue limitations today in order to accomplish this.

SAS Queued

A three-bit field of the READ (16) (and optionally WRITE (16) if standards body desires) command shall specify the index into the array of descriptors that the device shall apply to that command.