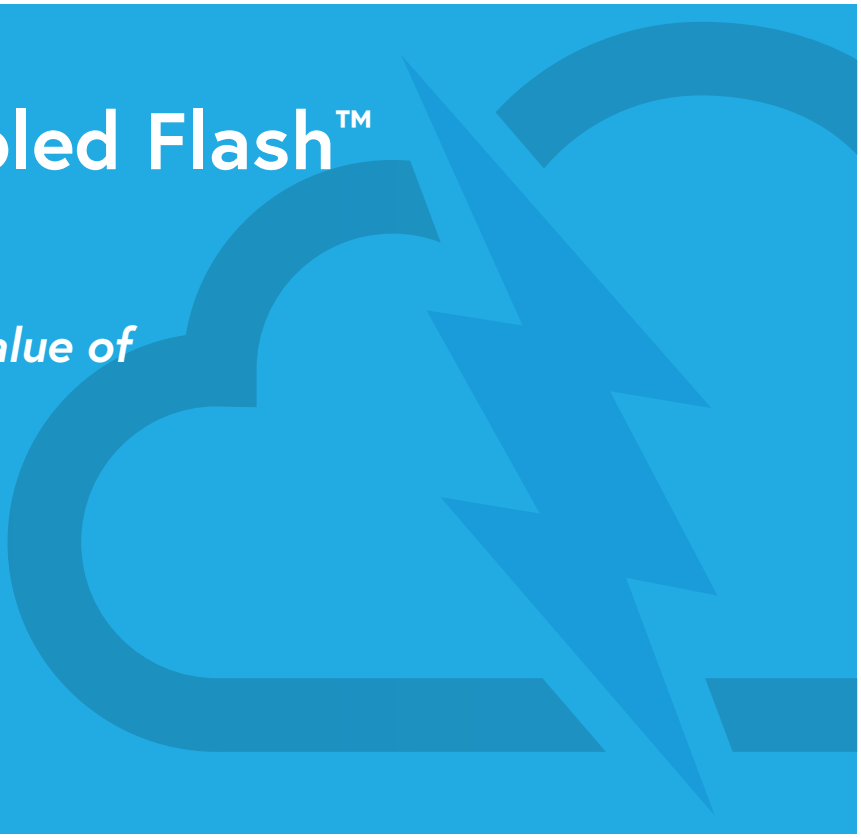
SEF

SOFTWARE-ENABLED FLASH™

Software-Enabled Flash™ Technology:

How to Maximize the Value of Flash Memory



TECHNICAL BRIEF

Software-defined applications are monopolizing data center hardware stacks as hardware is managed by software, changing the paradigm of fixed-functional hardware. Hardware functionality can be expanded by making it software-definable as demonstrated by technologies including software-defined networks, accelerators and storage devices – all of which are now common in the data center. These software-defined capabilities are now being applied to flash storage devices with Software-Enabled Flash (SEF) technology. This software-defined flash native technology can help maximize the value of flash memory and is the focus of this tech brief.

Software-Enabled Flash Technology Overview

Software-Enabled Flash technology is an open source community focused project of The Linux Foundation®, encouraging multi-vendor, cooperative development for future applications, capabilities and beyond. It addresses the fixed functionality in traditional data center hardware where there are no mechanisms in place to unlock any additional capabilities within the device. Therefore, when a new or specific storage capability is needed, the user may need to purchase additional hardware, take a server offline and/or manage the inventory. All of these activities are expensive and time-consuming in the traditional data center.

In today's modern data center with software-definable and modifiable hardware, Software-Enabled Flash technology is designed to unlock the full potential of the flash memory within compatible SEF storage drives. When applied, new capabilities can be deployed with the SEF drive in place that may only require a simple software driver change. The ability of not having to swap drives can result in no hardware replacement costs, no server down time and no inventory management.

Unlocking the potential of flash memory can make a compatible SEF drive significantly more functional and valuable. The restrictions of legacy hard drive protocols that were used for traditional flash storage devices can be unleashed in Software-Enabled Flash technology enabling data center applications to be flash-optimized. As a result, SEF drives are capable of addressing more demanding applications as well as ever-changing cloud storage requirements.

Benefits of Flash Abstraction

Prior to Software-Enabled Flash technology, legacy hard drive protocols[1] required a lot of development time to refine and tweak code, addressing the changing requirements of a particular storage drive (such as the priorities associated with read, write and latency operations). The code also needed to be refined in order to align data or minimize the write amplification of a specific Flash Translation Layer (FTL) that was in use.

In addition to changing the Input/Output (I/O) priorities, developer tasks associated with data placement and garbage collection (GC) can become an onerous task, requiring workarounds for a specific drive type[2]. When a new drive model or a firmware upgrade to an existing model was deployed in the traditional data center, all of this extensive development work could be wasted. Now with Software-Enabled Flash technology and SEF drives, the data placement and GC tasks can be optimized through host control, which determines when and how these tasks are started and run. This flash abstraction of background functions controlled by the host enables developers with deeper control over the device management tasks, which in turn delivers Quality of Service (QoS) and latency control improvements on applications and workloads.

In addition to QoS and latency control improvements, Software-Enabled Flash technology delivers important new host control benefits, such as isolated tenant environments and tiered service levels. As data centers need to transition to the highest density media quickly when they become available, the ability of Software-Enabled Flash technology to abstract flash memory from the drive is a major benefit to developers. As SEF hardware technology handles the flash management tasks, the Application Programming Interface (API) and the application itself manage the rest. For each new flash memory generation, flash abstraction can minimize the amount of work needed when moving between flash technologies, drive models and even different vendor products, and enables fast time-to-market (TTM).

Software-Enabled Flash technology is designed so that the SEF hardware storage device is responsible for managing the low-level flash memory tasks such as erase times, patrol periods and pseudo-SLC (pSLC), TLC and QLC[3] programming. These tasks are hardware-configurable by each manufacturer to include different capacity levels of on-board RAM, power loss protection[4], flash interfaces, etc. The SEF compatible hardware also implements a unique, die-level I/O priority queueing system that enables a drive to prioritize its I/O operations at the request of the upper layer software stack. SEF compatible devices

can also be built with or without DRAM. For example, if DRAM is not needed in a read-intensive platform, the customer does not need to use a device with this added DRAM. This hardware design option that eliminates DRAM can result in significant cost savings at scale.

Additional information regarding the SEF hardware configurations and how they provide data center developers and architects with the ability to manage and define the behavior of their storage, is available from the 'Introducing Software-Enabled Flash Technology' white paper. These SEF behaviors can include host control over data placement and hardware/software workload isolation, latency control via dynamic queueing modes, physical address control over read processing, simplified adoption and migration of new flash memory generations and vendor differences, and flash abstraction of low-level tasks such as GC and wear-leveling. These behaviors are further explained in the Software-Enabled Technology 'Introducing the Software Stack' tech brief.

"When [Software-Enabled Flash technology is] applied, new capabilities can be deployed with the SEF drive in place that may only require a simple software driver change. The ability of not having to swap drives can result in no hardware replacement costs, no server down time and no inventory management."

Summary

The key benefits of Software-Enabled Flash technology include:

- *Efficient performance and capacity scaling with a common flash hardware storage device*
- *Simpler inventory management*
- *Faster TTM via flash abstraction support*
- *Less development time to refine changing requirements*
- *Removes legacy HDD protocols and associated limitations*
- *Supports any flash memory vendor and future flash generations that are software-enabled*

The Software Developer Kit (SDK) included with Software-Enabled Flash technology is expected to be available from the GitHub® site, and should include full source code for drivers, development and test tools, and the Command Line Interface (CLI). The SDK is also expected to include an industry-standard I/O testing tool. As it relates to evaluating performance regarding hardware or software isolation of virtual devices or latency control, the Software-Enabled Flash open source project also features developed tools to demonstrate the extended value of this new technology.

For more information on Software-Enabled Flash technology, the SEF SDK and the entire SEF ecosystem, visit the project website where white papers, videos and infographics are available.

NOTES:

[1] Legacy hard drive protocols were developed for HDDs, and adopted in SSDs, and are currently used in both.

[2] The specific drive types include HDDs and SSDs.

[3] Flash memory types include Single-Level Cell (SLC) which stores 1-bit of data per flash memory cell; Multi-Level Cell (MLC) which stores 2-bits of data per flash memory cell; Triple-Level Cell (TLC) which stores 3-bits of data per flash memory cell; and Quad-Level Cell (QLC) which stores 4-bits of data per flash memory cell.

[4] Power loss protection records data in buffer memory that is sent to flash memory and utilizes the backup power of the solid capacitor in case of a sudden supply shutdown.