KIOXIA

Technical Brief

KIOXIA Software-Enabled Flash™ Technology:

Improve Performance and Efficiency of Hyperscale Data Center Flash Storage

KIOXIA has unlocked the potential of flash memory with a new Application Programming Interface (API) that is available as open source software to better control and manage flash-based storage device behaviors. KIOXIA Software-Enabled Flash technology will feature a Software Development Kit (SDK) that enables hyperscale developers and architects to improve storage performance based on latency management, tenant isolation, data placement, endurance, Quality of Service (QoS), and other capabilities. The software API helps flash memory achieve its full potential and is designed to easily adapt to multiple generations and vendor implementations of flash memory. This capability provides rapid, cost-effective and efficient transitions to new flash generation deployments.

One of the challenges that hyperscale developers and architects face is the ability to better control and customize their flash storage device deployments. They develop solutions to achieve peak application performance while also trying to meet the specific needs of their hyperscale data centers. Software-Enabled Flash technology is a new solution that provides the development tools and answers that hyperscale developers and architects require.

When deployed within flash storage, the open source API can dynamically address changing storage needs, or lifecycle and usage requirements, or cost efficiencies, or the need to develop specialized or custom hyperscale applications. Software-Enabled Flash technology was developed to maximize the capabilities of flash memory so it can achieve its full potential with the goal of enabling new solutions from the data storage community.

Introducing Software-Enabled Flash Technology

The performance advantages and capabilities of flash memory are yet to be tapped in real world hyperscale applications. New Software-Enabled Flash technology not only provides hyperscale developers and architects with control over their flash memory to meet specific application needs, but also enables full use of their native capabilities. The open source software API is available to hyperscale developers with the associated definition and specification document downloadable from the KIOXIA repositories on the GitHub® site[1]. A home page with white papers, presentations and demonstrations is also available at https://softwareenabledflash.com.

The software API allows the host to be focused on the needs of applications versus having to micro-manage flash memory. It is also designed to enable applications to be portable across different flash generations, technologies and vendor implementations. Software-Enabled Flash technology provides a unique set of application-focused functions and capabilities that include virtual devices and QoS domains, weighted fair queueing (WFQ), and garbage collection primitive offloads that help to reduce write amplification (WA).

The API also enables users to define and isolate die and channel configurations into one or more ‘virtual devices’ (Figure 1) by which applications can define a specific die to use across one or more specific channels. Users can set up virtual devices to create hardware isolation that eliminates I/O impact from one virtual device to another.

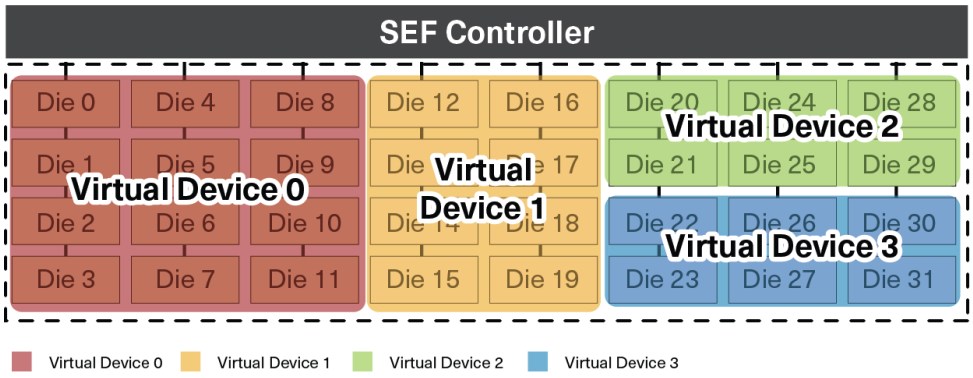

Figure 1: The software API assigns dies and channels into virtual devices

To further enable workload or tenant isolation, a software isolation layer is defined as a QoS domain within the API. These QoS domains are defined by capacity inside an isolated virtual device and deployed as the actual user space within Software-Enabled Flash units (Figure 2). The virtual device can have one or more QoS domains defined as they are independently managed by software applications.

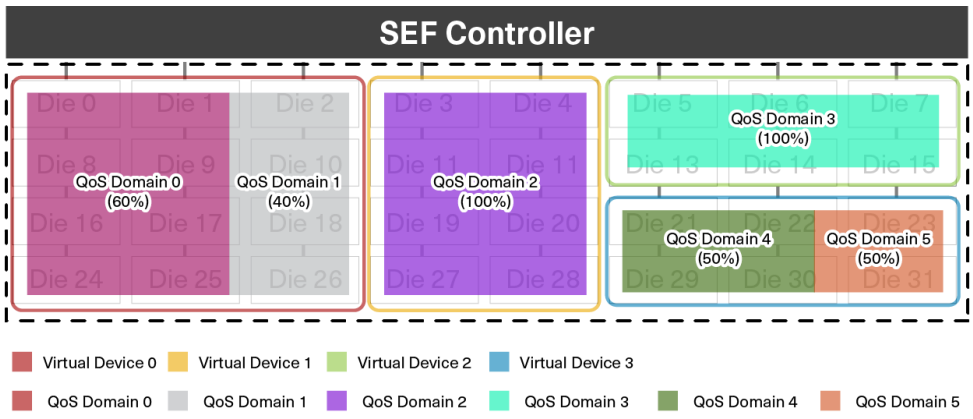

Figure 2: QoS domains are used as user space for Software-Enabled Flash units

Software-Enabled Flash technology (Figure 3) combines purpose-built hardware and software with an open source API that enables a dynamic relationship between host systems and flash-based storage devices. The technology also enables applications to deploy workload isolation and QoS on a fine-grained basis, manage data placement specifically based on flash locations, and control background and garbage collection operations to best suit application needs.

Developers and architects of hyperscale environments are provided access to an SDK that includes a variety of tools such as an open source Linux® reference block driver and flash translation layer[2] (FTL), C language and C++ wrappers, data structures, utility functions, and user space and kernel implementations. In addition, the Software-Enabled Flash technology project will use a neutral third-party, open source software host, to emphasize innovation and open collaboration.

The programmable API that is unlocked by Software-Enabled Flash technology enables hyperscale developers and architects with the ability to manage and control flash memory with customizable options. Hyperscalers will be able to manage flash memory with new expanded capabilities, regardless of the flash memory vendor or the flash memory generation being deployed.

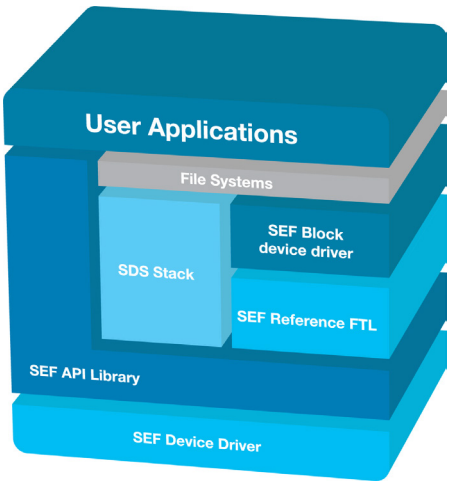

Figure 3: Software-Enabled Flash technology host software block diagram

Moving Beyond Hard Disk Drive Interface Technology

Flash-based storage delivers faster read and write speed performance and faster access latency than traditional hard disk drive (HDD) interface storage. It also generates less heat, has lower energy consumption, and much higher tolerance to shock and vibration versus hard drives. When Software-Enabled Flash technology is added, dynamic software customization can be applied and implemented within flash storage to address changing needs, or lifecycle and usage requirements, or to optimize for better total cost of ownership (TCO), or simply to create specialized hyperscale applications. For instance, endurance can be based on application requirements, which in turn, enables users to obtain higher densities or longer lifecycles for their flash-based storage devices.

Flash-based storage replaces traditional hard disk storage as the platform which supports the most intensive workloads and has led to an explosion of storage innovators and hyperscalers integrating flash memory to boost performance. To go beyond legacy hard drive interface technology and move data efficiently through the system, software engineering and intelligence can leverage the superior performance of flash memory and the flexibility of its digital nature.

The objective of Software-Enabled Flash technology as an open source API solution is to maximize the capabilities of flash memory so it can achieve its full potential with the goal of enabling additional solutions and deliverables from all flash memory manufacturers, storage innovators and hyperscalers.

Summary

Software-Enabled Flash technology is a new approach to flash storage that overcomes the unique challenges of today's high-performance, high-capacity data centers. It addresses these application performance challenges by providing functional control over endurance, latency, tenant isolation, data placement, and QoS, to name a few.

Hyperscale developers have attempted to custom-build storage devices that address these challenges, only to be confronted by additional problems that include addressing the varying characteristics of each flash memory generation and vendor. The Software-Enabled Flash API may enable hyperscale developers, cloud software developers and a wider community of application developers, data architects, and end users to solve their unique flash storage challenges. Flash memory becomes easier to manipulate and use, enabling developers to implement essential functions and capabilities, with solutions that can look, act and perform like a flash-based storage device but enabled, defined and managed through a software API.

Software-Enabled Flash technology is also ideally suited for all-flash arrays and proprietary storage solutions since the API can help create a common interface across multiple vendor flash-based devices.

The future of flash storage is software-enabled.

For more technology information, visit https://softwareenabledflash.com.

NOTES:

GitHub is a registered trademark of GitHub, Inc. Linux is a registered trademark of Linus Torvalds. All other trademarks or registered trademarks are the property of their respective owners.

[1] The GitHub branch platform includes a website and cloud-based service that stores and manages the code that a company develops, and tracks and controls any changes to their code. GitHub is an exclusive trademark registered in the United States by GitHub, Inc.

[2] The reference flash translation layer is a system that manages SSD operations and performs logical-to-physical address translation, garbage collection, wear-leveling, and other functions.