WHITE PAPER

Delivering High Sustained Performance with Enterprise Class Reliability

Vijay Karamcheti, CTO, Virident Systems

Explosive I/O Growth Driven By a New Breed of Applications

We are seeing a fundamental shift in computing, triggered by an explosion of data driving the applications that run on the world's largest Internet and Enterprise data centers. The continuing popularity of Internet portal, search, and e-commerce sites and the exponential growth in social networking sites has resulted in a quadrupling of the size of data stores at these sites every 18 months. Similarly, even traditional Enterprise workflows are being transformed due to the rapid adoption of virtualization and private clouds and the need to harvest value from the huge repository of structured and unstructured data generated across the enterprise. Looking into the future, the trend towards internet-enabled mobile computing devices (smart phones, Netbooks), and the growing relevance of cloud computing infrastructures in all of its forms promises to only accelerate this growth.

Accompanying the growth in data volumes is also a shift in how applications are interacting with this data. In contrast to traditional data-intensive applications such as database analytics, which have evolved over the years into disk-friendly access patterns (medium-to-large sized blocks, sequential scans, seek penalty-aware data structures, etc.), a new breed of data-intensive applications are emerging, which impose different, more challenging performance requirements on data storage devices. Examples of such applications include search, messaging, ad hoc data analytics, and social graph traversals–these applications tend to typically interact with unstructured data, performing a large number of small granularity accesses to random locations in the data collection. Such access patterns are markedly different from the carefully coordinated large-granularity accesses that characterize the traditional structured data applications.

Both of these trends–growth of data volumes, and the growing random access nature applications outstripping advances in mechanical hard-drive technologies–are exposing shortcomings in traditional disk drive-based storage infrastructures, which are unable to keep up with the growing performance demands. Even currently, the most performance-intensive applications require fairly complex storage deployments, involving hundreds to thousands of disk spindles, and use of techniques such as multi-way striping and "short stroking" to meet the bandwidth and IOPS (I/O Operations Per Second) requirements, respectively. The cost, footprint, power requirements, and operational complexity of such deployments are fast making them untenable for the wide variety of situations that demand such performance levels.

Opportunity For Non-volatile Memory Components

Recent advances in solid-state memory technologies, particularly NAND Flash memories, offer an attractive solution to this problem. Although DRAM and disk have established themselves as the most durable alternatives for server storage, this new class of Storage-Class Memories (SCM) offers an interesting blending of the characteristics of both DRAM and disk. Table 1 compares these three storage alternatives[1], and highlights the fact that SCM technologies such as NAND Flash can be used to build storage systems with the performance and random-access capabilities of DRAM, and the persistence and capacity of disk drives. Note that SCM is not limited to NAND Flash. It can comprise of other technologies like NOR Flash, and emerging technologies like Phase Change Memory, MRAM, etc.

Attribute	DRAM	NAND Flash	Disk
Capacity	10's of GB	1-2TB	1-10TB
Persistence	No	Yes	Yes
Cost/GB[2]	$10.00–$100.00	$1.00–$10.00	$0.10–$10.00
Performance			
Reads	~100ns, 10+ GB/s	~50 us, 2+ GB/s	10 ms, 200-500 MB/s (seq), 10-20 MB/s (random)
Writes		~500 us, 1+ GB/s (seq), ~500 us, 300+ MB/s (random)	
Access Granularity	Byte, word, cache line (10s of bytes)	Block (1000s of bytes)	

Table 1. Comparing NAND Flash memory with DRAM and Disks

NAND Flash-based Solid State Drive (SSD) is the first instance of SCM that is going mainstream. A large number of companies, ranging from non-volatile memory manufacturers to server OEMs are offering NAND Flash-based SSDs in various form factors. All of these products are riding the Moore's Law influenced scaling of NAND Flash capacities (with accompanying reductions in cost/GB), which is making possible disk drive-like capacities for a moderate cost multiple. When one takes into consideration the hundred to thousand-fold improvements in random access performance possible with NAND Flash devices and their sharply dropping cost, it appears inevitable that SSDs will replace traditional disk drives, used as the performance storage tier in majority of data-intensive applications in a few years.

Challenges with using NAND Flash to build high-performance I/O subsystems

Unfortunately, when one takes stock of ongoing deployments, reality has not quite lived up to the promise, particularly in the server context. Despite the initial euphoria, NAND Flash-based SSDs seem to have hit the "trough of disillusionment" soon after a "peak of inflated expectations". Although NAND Flash-based SSDs delivered high performance on short running, carefully controlled tests, the typical user experience has been, for the most part, disappointing. Figure 1 shows a typical variation of the performance of a NAND Flash SSD, as it is subjected to workloads that range from short, controlled tests to long running, real-world workloads.

[1] The performance comparisons are at the subsystem level, and assume present-day densities, form-factors, and power considerations.

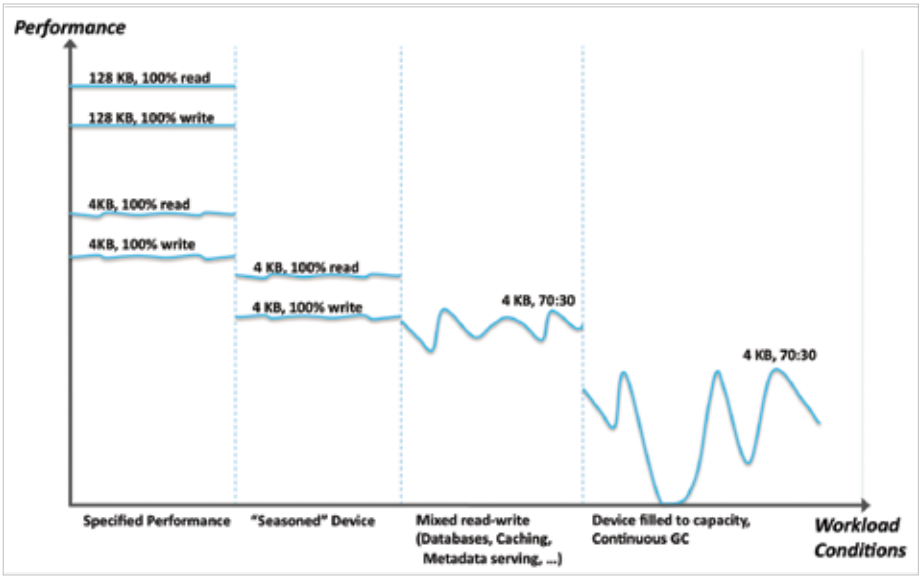

Figure 1. Performance variations in current generation of NAND Flash-based SSDs

The first generation of NAND Flash-based SSDs were typically advertised as delivering extremely high read and write bandwidths, particularly compared to disk subsystems. However, what most users experience is lower than advertised bandwidth on the smaller block sizes that typically motivate use of SSDs. On such block sizes, not only is the peak performance lower, but perhaps more disconcerting to most users is the fact that performance varies with usage patterns, over time, and with mixed read-write requests of the form typically encountered in I/O intensive applications. Of these variations, the ones that average users get most confused about is the fact that most NAND Flash-based SSDs on the market show markedly different behavior depending on whether or not the device is filled to capacity and what background activities are ongoing. Over the last few years, the user community has grown increasingly aware of these performance variations and the need for different benchmarking methodologies, both of which have led to the current skeptical view of the benefits such devices can bring to real-world application workloads.

To address this situation, we first need to understand the primary reasons for such behavior. One can group the underlying causes into three broad categories:

- The 'physics' of NAND Flash devices
- System architectures for high-performance NAND Flash-based storage devices
- Operating system and application interactions

Physics of NAND Flash Devices

At the device level, NAND Flash memories possess certain fundamental characteristics that have performance and reliability implications. These characteristics are similar to those of other non-volatile Flash memory technologies. At the elemental level, NAND Flash memories store information using a "charge trapping" mechanism whereby a certain amount of electrons are trapped in special structures within the semiconductor substrate. Different mechanisms are used for storing the charge ("writing") and for sensing how much charge is stored ("reading"). Additionally, the Flash memory cell structure requires an intervening operation to rewrite it–the operation consists of draining the charge trapped in the semiconductor substrate, before allowing new information to be stored in the cell.

These different mechanisms for reading, and erasing a Flash cell result in significant heterogeneity of corresponding operations at the device level. For example, a typical current-day single-level cell (SLC) NAND Flash device can read data from the memory array in 25-30 microseconds, can write data into a group of erased cells in 300-500 microseconds, but requires 1-2 milliseconds to erase a group of cells to permit rewrite. Beyond the performance differences, NAND Flash devices also work with different granularities at which these operations can be performed. For example, a current-day SLC device permits reads at byte-level granularity, but writes only at 512 B or 4 KB granularity, and erases at 256 KB granularity.

Additionally, as higher capacity and lower cost NAND Flash devices are being constructed using increasingly smaller semiconductor process geometries, the charge trapping mechanisms also introduce reliability and endurance considerations. The analog nature of the charge trapping and sensing mechanisms make reads and writes against Flash devices inherently unreliable–with each generation of NAND Flash device, the controller interfacing such devices is expected to take on responsibility for increased levels of error correction. In addition, Flash devices are characterized by limited endurance–a given Flash cell can only be erased and reprogrammed a certain number of times before it loses its ability to reliably store and retrieve information. Current-day SLC NAND Flash devices support 100,000 program (write) or erase cycles, but pressures on cost and capacity are rapidly driving this number down.

The System-level Impact

These characteristics-the heterogeneity of read/write/erase performance and the granularity, and the inherent unreliability and limited endurance of Flash–have multiple system-level implications.

First, mixed read-write workloads and workloads that require rewrites of previously written data will typically result in a mix of read, write, and erase operations happening at the device level. The order of magnitude difference in operation latencies means that one can encounter situations where a low-latency operation such as a read gets queued behind a high-latency operation such as an erase. Such “head of the line” blocking can produce significant unpredictability in mixed read-write workloads.

Second, to reduce having to perform expensive inline “read-modify-write” operations to rewrite data, most current-day Flash management software relies on a Flash Translation Layer (FTL), which maps logical blocks at the Flash subsystem level into (potentially changing) physical blocks in individual Flash devices. Writes at the subsystem level are directed to previously unused blocks in erased state, and result in FTL mapping changes from the old physical blocks to new physical blocks. This mechanism is intended to reduce the application-visible latency of writes, but requires a background “garbage collection” activity, which sweeps through previously used Flash locations copying and consolidating valid data in a new location before erasing the old location to place it back into service. Since both the FTL and the garbage collection activity compete with foreground read and write activity for usage of system Flash resources, the underlying algorithms and structures can significantly impact both the aggregate performance seen by write-intensive workloads and the predictability thereof, thus defeating the original intent of reducing application-visible latency of writes.

Third, the limited endurance nature of Flash places an increased emphasis on “wear leveling” algorithms. Wear leveling is intended to ensure that a given Flash-based device can provide reliable, predictable performance over its guaranteed lifetime independent of any localization of accesses in the presented workload. In particular, independent of whether a logical data block is read or written more frequently than others, wear leveling algorithms need to ensure uniform usage of the system’s physical Flash resources to ensure a predictable level of reliability and lifetime.

System Architectures for High-Performance NAND Flash-based Storage Devices

Individual Flash devices must be aggregated into a subsystem containing a collection of such devices to meet the capacity and performance requirements of high-performance, data-intensive applications. NAND Flash-based SSDs in this context primarily offer a performance value by introducing a layer in the memory hierarchy, which can bridge the gap between lower capacity but higher performance DRAM and higher capacity but lower performance disk drives. How effectively a particular SSD plays this role is influenced both by the system architecture it uses to aggregate a collection of Flash devices, as well as by how this device is interfaced with the rest of the server platform.

Most SSD system architectures rely upon well-established striping and parallelism techniques to build high-performance Flash subsystems. These techniques have a distinguished history ranging from memory-level interleaving approaches to high-performance RAIDed disk subsystems, and are very effective at increasing the bandwidth delivered by the subsystem. However, such techniques must be adapted in non-trivial ways for Flash-based SSDs to deliver unique value over high-bandwidth disk subsystems. This value, which consists of delivering high random IOPS using small block sizes, is at odds with the underlying nature of striping techniques that commonly improve bandwidth by increasing the block granularity using which applications and/or the operating system interact with the device. Beyond resulting in transfer of unneeded data, the block size increase also has negative implications for write-intensive workloads where the larger block sizes can increase the demand for either inline read-modify-write operations, or alternately increase the garbage collection pressure. Both of these implications are captured by the "write amplification" metric, which represents the amount of data actually written to the Flash devices in response to a unit data write request at the application level.

Equally important is how Flash-based SSDs are interfaced into the server platform. Several standard interfaces have been defined for interfacing disk-based storage devices, with currently the most popular being SATA and SAS. However, these interfaces limit the performance delivered to end applications from both a hardware point of view and a larger system interaction viewpoint. Particularly for the server context, there appears to be a growing consensus towards the use of PCI Express (PCIe)-attached NAND Flash SSDs. In addition to leveraging the order of magnitude higher bandwidth and lower latencies, these latter interfaces offer[3],they are being directly integrated into processor packages in much the same way as DRAM interfaces. This integration offers significantly leaner and more efficient interactions than is possible via the SAS/SATA interfaces. The expectation is that, with growing server OEM support, PCIe-attached Flash-based SSDs will soon become the norm for the highest-performance storage requirements.

Operating System and Application Interactions

The third category of causes explaining the unpredictable behavior of NAND Flash-based SSDs consists of the impact of routing application block access requests via operating system layers that were designed and tuned for mechanical disk-based subsystems.

NAND Flash-based SSDs are capable of levels of performance that expose bottlenecks in current-day operating system storage stacks. For example, appropriately designed PCIe-based NAND Flash devices are capable of sustaining 300-400K IOPS from a pure hardware point of view. But when one takes into consideration the overheads inherent in various layers of the storage stack, the delivered IOPS may end up being significantly below that level. A recent Intel study characterized the impact of various levels of the storage stack in the Linux operating system shown in Figure 2,

[3] The comparison here is in terms of bandwidth achievable per slot or per unit volume, where PCIe offers a significant advantage over SATA and SAS interfaces.

finding that each level approximately brings down delivered IOPS by a factor of 1.5x4. These bottlenecks are not as important when one considers use of these stacks with lower-performance devices such as traditional hard disks or even consumer-grade SSDs, but they become critical when used with server-grade SSD devices that are capable of delivering much higher levels of performance.

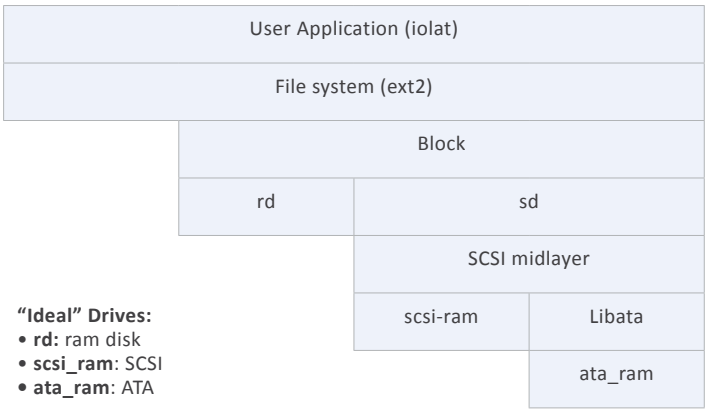

Figure 2. Overheads in the Linux Operating System stack

The developer communities of the most popular operating systems are working actively to address these bottlenecks and to tune internal policies to make them more SSD aware, and improvements are expected to roll out over the next few years. However, some fundamental bottlenecks are expected to remain in unified stacks that need to support different storage devices that exhibit a wide range of performance characteristics–from traditional hard drives capable of 250 IOPS, to consumer-grade SATA/SAS SSDs capable of 20K-40K IOPS, to server-grade PCIe SSDs capable of sustaining 200K or higher IOPS. Differences are also expected to remain with regards to I/O "Quality of Service" metrics, such as jitter and the distribution of request latencies.

Similar considerations also apply to application structures, which were designed keeping in mind the performance characteristics of relatively slow hard disks on one end or relatively high performance DRAM on the other. By virtue of their ability to occupy an intermediate level in the memory hierarchy, NAND Flash-based SSDs are positioned to replace disk collections in one class of applications, and reduce the need for large volumes of DRAM in another. However, applications are slow to change, so NAND Flash-based devices need to deliver their performance with application structures that are tuned for one extreme or the other. This is in fact one of the primary reasons why the initial promise of NAND Flash-based SSDs has not been realized at the application level–despite showing good performance on synthetic, low-level I/O benchmarks, most current-day SSDs are unable to deliver good performance on real-world workloads. For the latter to happen, a number of additional considerations come into play such as the number of outstanding I/O operations, the application interfaces used for issuing such operations, and application-level synchronization structures that can cause the issuing structure to adapt to the behavior of the underlying I/O device. An example of the latter includes the log write pattern common to most database applications–this pattern requires the underlying storage device to deliver high performance with few outstanding I/Os (typically one), else the entire application can get serialized behind this operation preventing the aggregate benefits of the device from coming through at the application level.

[4] Linux* Storage Stack performance, Kristen Carlson Accardi, Matthew Wilcox, Open Source Technology Centre, Intel (http://www.usenix.org/event/lsf08/tech/IO_Carlson_Accardi_SATA.pdf)

Viridient's *tach*IO*n* Drive: A High Sustained Performance, Reliable I/O Subsystem

Although these challenges to obtaining sustained high performance from NAND Flash-based SSDs look daunting, they can be overcome by the appropriately designed product. Viridient's *tach*IO*n* drive is such a product which has been architected from the ground-up to address the performance, reliability, and usability issues experienced by the users of first generation SSD devices.

Viridient's *tach*IO*n* drive has been designed to meet the following objectives:

- Deliver the highest levels of sustained performance on real-world application workloads.
- Deliver the highest reliability, leveraging multiple modalities of redundancy and error correction.
- Support comprehensive storage lifecycle management, including incremental scaling, in-field replacement of faulty components, and (as appropriate) customization of internal policies to the needs of specific workloads.

Sustained Performance on Real-World Application Workloads

Sustained performance refers to a high degree of predictability in the performance characteristics seen by applications, independent of:

- whether the performance is being measured on the device just as it is placed into service, or towards the end of its advertised life;
- whether the device is lightly utilized, or filled to its advertised capacity; and
- what background maintenance activities (e.g. garbage collection) are ongoing in the device.

The focus on real-world workloads translates to an ability to deliver high performance on I/O characteristics commonly encountered in production applications. These characteristics include:

- small-to-medium block sizes in the range 4KB–32KB;
- thread counts and outstanding I/O counts that are a small multiple of the number of processor cores in the server system;
- mixed read-write workloads, where writes can make up a significant fraction of the overall workload; and
- both regular and bursty workloads, requiring the device to be capable of adapting its behavior to presented access patterns.

Figure 3 qualitatively illustrates these performance aspects the *tach*IO*n* drive. Unlike first-generation NAND Flash--based SSDs, the *tach*IO*n* drive is designed to deliver predictable, steady throughput across a range of workload conditions. Going beyond the focus on aggregate bandwidth, the *tach*IO*n* drive also emphasizes I/O Quality of Service (I/O-QoS) metrics such as the difference request latencies seen an unloaded vs. a loaded system, jitter i.e. the difference between the request latency and the 99th percentile statistic, etc. The latter metrics are particularly important in domains where a premium placed on guaranteeing tight bounds for application-level transactions, such as recording a financial trade.

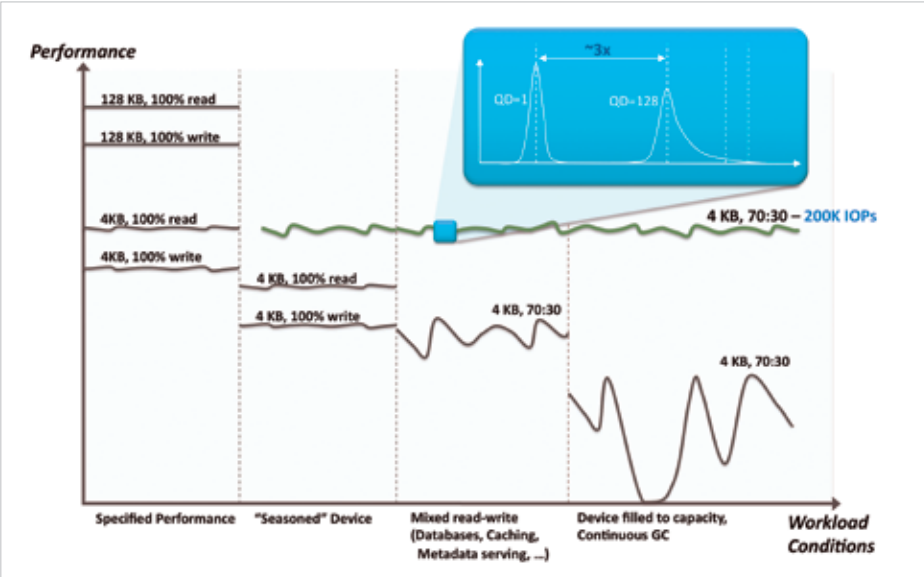

*Figure 3. Virident tach*IO*n is designed to deliver high, sustained performance on real-world application workloads.*

The *tach*IO*n* drive achieves these performance characteristics by bringing together several advances in its hardware and software architecture.

Key hardware advances that impact performance include:

- A Flash controller hierarchy enabling an increase in the level of parallelism that can be integrated on the device within the form factor and power constraints of a low-profile PCIe card. The parallelism can be exploited for higher performance, or alternately for increasing performance predictability for a given application workload.
- Proprietary logic in the controller hierarchy, which can optimally tradeoff the dual considerations of striping for performance and smaller block sizes for limiting write amplification.
- Advanced error correction techniques and data movement techniques to deliver high reliability with high performance, even for workloads with relatively small numbers of outstanding I/O operations.
- Capacitor-backed power-fail mechanisms, which permit early termination of write operations without having to wait for the data to be persisted in a Flash memory location. This attribute targets the requirements of real-world database and messaging workloads, by improving the latency of write operations and delivering high write band width in workloads with small block sizes and small numbers of outstanding I/O operations.

The main software advances that deliver high, sustained performance include:

- Sophisticated scheduling mechanisms to manage the flow of up to thousands of application requests to and from the device, with minimal control overhead and in a fashion that delivers very high levels of application-visible performance.
- Algorithms for minimizing read and write amplification, the amount of data that is read from or written to Flash in response to application-level read or write requests.
- Integrated algorithms that unify policies for block allocation, operation scheduling, garbage collection, and background maintenance activities to deliver high, sustained performance for mixed read-write workloads even when the device is fully utilized and garbage collection is continually active.

Multiple Modalities of Device Reliability

Virident's *tach*IO*n* drive is a highly reliable enterprise-grade I/O subsystem.

To achieve this objective, it includes multiple modalities of redundancy in its internal hardware and software architecture:

- Advanced error correction codes, which are aware of and protect against the device-level unreliability inherent in small geometry NAND Flash memories. This level of protection is analogous to the SECDED level of protection against soft errors in ECC DRAM modules, elevated to the levels mandated for NAND Flash devices. The *tach*IO*n* drive includes an order of magnitude higher error correction capability than that recommended by memory manufacturers for a specific process generation.
- System-level redundancy, which protects against data loss protection even in the event of catastrophic plane, die, or package-level failures. Such failures are difficult to eliminate completely, particularly in devices that can include hundreds of Flash dice. This level of protection is analogous to the Chipkill protected ECC DRAM modules, or RAID redundancy in disk subsystems, both of which provide one or more orders of magnitude higher reliability than basic error correction.
- Storage lifecycle management mechanisms, described in additional detail below, which help bring back a degraded system into its original, highly reliable state.
- Proactive software driver and firmware mechanisms, which detect and actively steer usage away from weaker regions of NAND Flash. As NAND Flash devices increasingly move to smaller and smaller process geometries, one expects a number of minor variations in how different regions of an otherwise good device end up behaving. Many of these variations are either not detectable either technically or in an economically viable fashion, with the consequence that one may well find such dice being qualified by the memory manufacturer as a production part. When one considers the hundreds to thousands of dice integrated into a high-performance SSD, even a miniscule probability of such issues translates into a relatively worrisome event, which needs to be protected against.

Comprehensive Storage Lifecycle Management

Virident's *tach*IO*n* drive employs a unique design where the Flash memory chips are packaged into replaceable modules that can be flexibly integrated with a base card carrying the controller circuitry.

This design offers multiple advantages:

- Physically partitioned view of card resources. The *tach*IO*n* drive driver can be configured to expose a subset of the replaceable modules as a standalone block device partition, whose usage can be customized in a flexible fashion. For example, it is common to find environments where multiple applications, such as a database and a search engine are hosted on the same server. In such environments, the *tach*IO*n* drive permits a card partition to be set aside for each application–each partition is physically isolated from other partitions, ensuring that there are no unexpected interactions when both applications are simultaneously active. The partitions can additionally be customized using *tach*IO*n* driver configuration options to best support the specific requirements of a particular application–configurability extends to support for more or less write intensive workloads, etc.
- Field replaceability of modules, to deal with the possibility that a Flash package undergoes more than anticipated wear, or (rarely) suffers a catastrophic failure. To protect against data loss, the *tach*IO*n* drive optionally supports RAID-5 like configuration over a subset of modules with online spares. In the event that a Flash package becomes faulty, data can be reconstructed on the fly using the redundant parity blocks. Additionally, similar to disk RAID configurations, the faulty module can be replaced with a new module

during a scheduled downtime, resetting the card to its original, highly reliable state. The *tach*IO*n* drivers provide support for seamlessly integrating the new module into the system, including rebalancing wear across old and new modules as appropriate.

- Incremental demand-based scaling of card capacity. As workload requirements grow, it is possible to upgrade the capacity of the card by adding on additional modules. The card can support up to 16 modules, which can be of more than one type or density.

Together, these advantages offer a flexible way of dealing with changing workload requirements and recovering from media degradation in a graceful fashion. The alternative, as is the case with first generation SSDs, would incur significant over provisioning at the start of the deployment, or unplanned interruptions over the deployment lifetime.

Virident Roadmap: Integrating Server-level Flash Into the Overall Storage Solution

Server-level Flash solutions are disruptive, but only the beginning. Today's dominant usage models for such devices are as a staging area for performance-critical data or as a cache for backend SAN/NAS storage. However, there is a larger opportunity for server-level Flash–to truly harness its latency and bandwidth advantages to become a distinguished tier of the overall storage solution. It has the potential to become the de facto Tier O storage layer, augmenting or in some cases replacing the traditional performance storage tier in the datacenter.

For this potential to be realized, there is a need for additional solutions at all levels–storage management software, caching and tiering software, as well as cluster-level solutions for pooling and high availability for server-level Flash to truly take on the role of Tier O storage. Future solutions from Virident will address some of the gaps in these areas, by relating application-and workload-level requirements from Tier O storage to the hardware and software architectures of *tach*IO*n* drive-like high-performance SSDs.

Corporate Headquarters
46600 Landing Parkway
Fremont, CA 94538
tel 510.933.8300
fax 408.321.0293
www.sgi.com

Global Sales and Support
North America +1 800.800.7441
Latin America +55 11.5185.2860
Europe +44 118.927.8000
Asia Pacific +61 2.9448.1463