

# SAN Cache: SSD In The SAN

BY MICHAEL CASEY

Solid-State Disk (SSD) modules will be installed as shared file-caching facilities on Storage Area Networks (SANs). These SAN cache appliances will enable rapid growth of the SAN infrastructure by multiplying application performance and by supporting virtual storage addressing. Conversely, SAN architectures will expand the range of applications and environments that can exploit the full benefits of SSD by making it easy to share and easy to manage.

Storage Area Network (SAN) discussions generally extol the benefits of sharing storage devices and subsystems—including disk drives, RAID subsystems, tape drives, automated tape libraries, and perhaps, other storage media such as CD-ROM or DVD. I would like to add another hard-

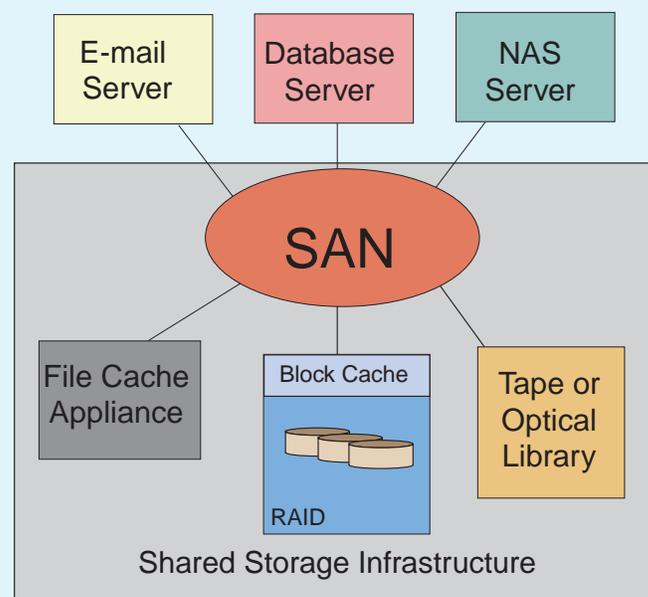
ware category to the standard SAN architectural sketch: the Solid-State Disk (SSD) file cache appliance or “SAN cache” (Fig 1).

Earlier articles have documented the impact of disk drive access density on I/O performance (“Access Density—Key to Disk Perfor-

mance” by Randy Kerns, Storage, Inc., Q2 1999) and the use of SSD to multiply application-server performance and scalability (“Disk I/O Performance Scaling: The File Caching Solution” by Michael Casey, [www.soliddata.com/whitepapers/file\\_caching.html](http://www.soliddata.com/whitepapers/file_caching.html)). The latter article introduced a key distinction—between block caching and file caching (see Sidebar) and described the benefits of server-attached SSD as a performance multiplier in transaction-intensive applications such as e-mail, messaging, and e-business.

This article outlines the synergies between SSD and SAN technologies: SSD will become a key-enabling technology for SAN performance and scalability and SANs will enable enterprises and system integrators to exploit the full potential of SSD as a performance multiplier. These benefits

Fig 1. File Cache on the SAN



will be most fully realized when the SAN infrastructure is automatically configured and managed as part of a virtual storage architecture.

### SSD As Enabling Technology For SANs

When an intelligent solid-state disk subsystem is added to a SAN, it becomes a shared file caching facility for the application servers attached to the storage network. It also becomes available as part of the storage infrastructure and system integrators and ISVs can use the file cache to enable functionality and performance that would be impossible to achieve with mechanical disk drives alone.

One key benefit is modular scalability of capacity and performance in the storage infrastructure. In a modular SAN architecture, a storage administrator or system integrator can configure the desired amount of capacity by adding disk drives or disk array modules. By adding SSD modules that are separate from the disk drive arrays, a SAN architect can independently “dial in” the desired amount of performance for transaction-intensive applications. Fig 2 illustrates this concept of managing capacity and performance as two separate dimensions with a separate “control dial” for each. Initially, this is a metaphor for a manual configuration process; however, the process will ultimately become an automated capability of virtual storage architectures.

By exploiting an optimized mix of RAID and SSD modules, a SAN architecture can deliver cost-effective configurations for a wide range of applications—including those for

which a single, monolithic approach is ineffective or needlessly expensive. Fig 3 positions a number of applications in terms of their high-water-mark requirements for response time and for bandwidth.

### SANs Enhance The Scope Of SSD

SAN developments will make SSD file caching easier to connect and share, thus making SSD cost-effective for a wider range of operating environments and applications. Fibre Channel provides a number of benefits, even when it is used simply as a replacement for SCSI connections in server-attached storage configurations (which is what most server vendors support today). These FC benefits include longer supported distances from server to storage and improved robustness and flexibility in hot-plug configurations.

The benefits of Fibre Channel are more fully exploited in server clusters that employ FC connections between the server nodes and a shared storage facility. A cluster architecture can use a shared SSD to make key files available to all servers in the cluster; this provides shared, non-volatile storage while avoiding the mechanical access latencies that are introduced by shared disk storage. High-availability clus-

ter software—such as VERITAS Cluster Server and Hewlett-Packard’s MC/ServiceGuard—will dramatically increase application recovery speed by maintaining file system journals (write logs) and related data structures on a shared, high-speed file cache.

As the industry moves toward switched-fabric SANs with heterogeneous server and operating system support, the deployment of SAN management software will enable enterprises to realize the benefits of file caching in a much broader range of applications. Whereas server-attached SSD is best suited

Fig 2. Independent Scalability of Application Performance and Storage Capacity

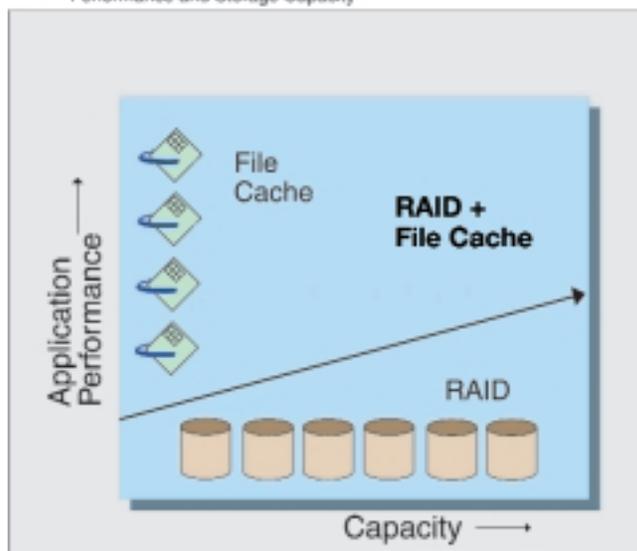
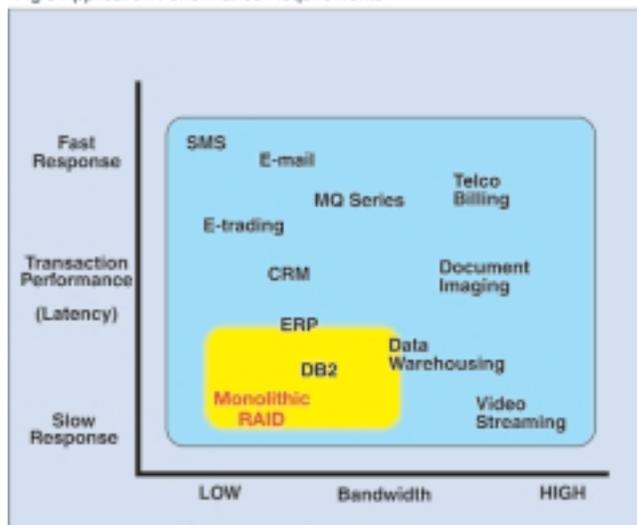


Fig 3 Application Performance Requirements



# File Caching With SSD

For many transaction-intensive applications, it is possible to identify a small set of files that consume most of the I/O activity and place those files in a high-performance cache. This approach typically uses Solid-State Disk (SSD) for file caching and increases overall transaction performance by a factor of 200% or more on the existing servers.

File caching differs from block caching in several respects. RAID cache—"block caching" in a RAID controller—is based on data blocks. Each block is identified by a SCSI block address (for example) and the RAID controller has no knowledge of which blocks are part of which file. It chooses what to cache (and what to flush from cache) based on historical usage statistics on the individual blocks (or disk tracks, in the

case of EMC). The caching algorithm looks at the usage history and tries to determine what will be needed next by the application. Since the controller cache is much smaller than the total amount of data stored on the disk drives, only a small percentage of the data can be kept in the cache. A given data block may reside in cache for only a few seconds or minutes before it is flushed from cache.

The effectiveness of block caching depends on the application and, usually, an application will reach a point of diminishing returns—a point where adding more cache will not deliver much additional performance improvement. Once the application reaches that point, the next step is to start caching entire files based on an understanding of application structure.

File caching depends on an understanding of the application structure—the identification of "hot files." Once the hot files have been identified, selected files are moved to the file cache—as a policy decision, not a statistical extrapolation. The hot files may reside on the SSD for days, weeks, or months. The "hit ratio" on data in the file cache is always 100%, since the entire file is always in the cache and available for access.

Two conditions are necessary to make solid-state file caching a good bet: (1) the application server must be I/O bound; (2) the I/Os must be skewed: a small percentage of the files must drive a large percentage of the I/O activity. For example, in many e-mail and messaging applications, the message queues represent a high percentage of

the total I/O on a small percentage of the data. They are also very write-intensive files. This skewed I/O distribution is a feature of the application design and, thus, the application is a good fit for architectural adoption of solid-state file caching. Typically, a small, I/O-intensive fraction of the data is moved from cached RAID to a separate, non-volatile file cache.

In suitable applications, addition of a solid-state file cache to an existing server configuration can boost throughput by a factor of four or even eight. This enables the system administrator to deliver the required performance and service levels without purchasing and managing several additional servers and their associated storage.

to applications that can justify investment in a dedicated file-caching subsystem, a shared SAN cache can be used by applications that could not justify dedicated file caches. These include smaller NT and Linux servers, each of which might need only a fraction of the capacity provided by a robust SSD product. In a SAN, many servers can share one file cache and storage management software can allocate appropriate amounts to each server as the workload changes.

The expanded range of applications will also include applications that only need a fast file cache for an occasional workload spike such as a month-end financial close that runs for 30 hours and needs to

complete in less than three hours. It might be difficult to justify a dedicated file cache to speed up the month-end close, but it will be easy to justify allocating part of a shared SSD facility to that application for a few hours each month. The shared SAN connection, together with SAN management software, will make this dynamic allocation easy enough for widespread adoption. Thus, file caching in the form of a shared SAN cache will be deployed to serve a much wider range of servers and applications. These developments will be further enhanced by virtual storage architectures.

## Virtual Storage Architectures

The holy grail of SAN evolution is the Virtual Storage Architecture

(VSA) in which SAN management software presents virtual disk volumes to the application servers and maps those virtual address ranges to physical storage devices connected through the SAN. Early examples include storage subsystems and SAN domain servers developed by XIOTech (recently acquired by Seagate) and ConvergeNet (recently acquired by Dell Computer). Other server and storage vendors such as Compaq Computer and Sun Microsystems are also developing virtual storage architectures.

Ultimately, virtual storage architectures will accept high-level requests from a SAN administrator and will automatically allocate physical resources based on storage policies

defined for each class of storage. For example, an administrator might use a storage control console to request creation of a 50GB virtual disk volume that can deliver 10,000 I/Os per second (at an 8KB block size) and a maximum response time of one millisecond. The management software would, then, configure the appropriate combination of physical resources available on the SAN such as fast disk, tape storage, and SSD.

Virtual storage architectures will make SSD easy to configure and easy to use. For example, VSA software facilities and services will enable easy migration of data from slow storage to fast storage without dis-

rupting application availability.

SSD will also be a crucial component of the VSA infrastructure. Effective operation of a VSA requires very fast conversion of logical addresses to physical addresses. In a distributed SAN, the obvious place to store the address lookup tables will be a shared, high-speed SAN cache.

In the future, as Storage Area Networks (SANs) are widely deployed and supported by sophisticated storage management tools—such as virtual storage architectures and policy-based storage management consoles—solid-state file cache will become an easily managed, shared

facility on the SAN. As such, it will become attractive and cost-effective for architectural deployment in a wide range of applications.

*Michael Casey is the vice president of marketing at Solid Data Systems (Santa Clara, CA).*



*Reprinted by permission from the publisher of the Quarter 1, 2000 edition of Storage Inc.*

*For FREE subscription information, please call 310/276-9500, ext. 36 or reply via the World Wide Web <http://www.wvpi.com>.*

©2000 West World Productions, Inc.



2945 Oakmead Village Court  
Santa Clara, CA 95051

Tel: +1.408.845.5700  
Tel: +1.800.287.0373  
Fax: +1.408.727.5496  
Email: [info@soliddata.com](mailto:info@soliddata.com)  
[www.soliddata.com](http://www.soliddata.com)