Realtime publishers

The Shortcut Guide to Balancing Storage Costs and Performance with Hybrid Storage

sponsored by

FUSION-IO

Dan Sullivan

Copyright Statement

Chapter 4: Best Practices for Deploying Hybrid Storage

Hybrid storage offers the combined performance benefits of flash storage along with the cost efficiencies of disk-based storage solutions. Throughout, this guide has considered the advantages of hybrid storage, the use of hybrid storage to improve virtualization performance, and the benefits of hybrid storage in database applications.

The benefits of hybrid storage apply to many types of workloads and applications, however, it is important to remember that there are multiple ways of implementing hybrid storage. This final chapter turns attention to best practices for deploying hybrid storage.

Key considerations for deploying hybrid storage include:

- Workload analysis—Workload analysis focuses on understanding the distribution of read-and-write operations and methods for improving both.
- Architectural issues associated with deploying hybrid storage devices—There are several architecture issues that should be understood when evaluating hybrid storage systems, including options for deploying flash storage in a hybrid environment.
- Real-time quality of service (QoS) controls—Real-time QoS controls address the need for varying performance levels according to an application's importance to business operations.
- Data management practices—Data management practices address issues such as data reduction, data protection, and dynamic data placement.
- Hybrid storage deployment planning—Finally, the planning section discusses requirements, deployment planning, and the need for policies specific to hybrid storage management and use.

Workload Analysis

Workload analysis is a process used to understand the read-and-write operations performed by an application or server with multiple applications. It is helpful to understand workload analysis in terms of:

- The IO blender effect
- The distribution of read-and-write operations
- Methods to improve the speed of both

The IO blender effect is a property of IO operations seen in virtualized environments.

IO Blender Effect

The widespread use of server virtualization has introduced a new set of challenges for systems administrators. Mixed workloads on servers can lead to different IO operation patterns than those seen on non-virtualized servers. Known as the IO blender effect, the mix of reads and writes can lead to lower levels of performance.

For example, an extraction, transformation, and load (ETL) operation can generate a large number of writes to sequential blocks on a disk. On a non-virtualized server, these operations could be executed in sequence and, assuming sufficient free space, in sequential blocks. As the data blocks are sequential, minimal time is required to position read-write heads.

On a virtualized server, the set of write operations for the ETL process are likely intermixed in the IO queue with operations from other virtualized servers. Each time a set of ETL write operations are performed, the disk read write heads have to be positioned appropriately leading to longer latencies.

In addition to understanding how mixed workloads can influence the order in which IO operations are performed, it is important to understand the relative amounts of read-and-write operations.

Understanding Distribution of Reads and Writes

Workloads vary in the proportion of read-and-write operations. An ETL process, for example, is write intensive while business intelligence reporting applications are read intensive. Transaction processing applications typically have a mix of read-and-write operations, such as 65% read and 35% write operations. Understanding the mix of read-and-write operations can help optimize the configuration of a hybrid storage system.

Consider an application with a high proportion of reads relative to writes. In such a case, a large amount of flash storage can implement a high-speed read cache for the application. As data is read from disk, it is sent to the application as requested and written to flash memory. Next time that data is needed, it can be retrieved from the flash memory instead of from a hard disk drive. As the cache fills and new data is added to the cache, some data is removed from flash memory. The larger the amount of flash memory, the more likely that data to be read will be in the cache.

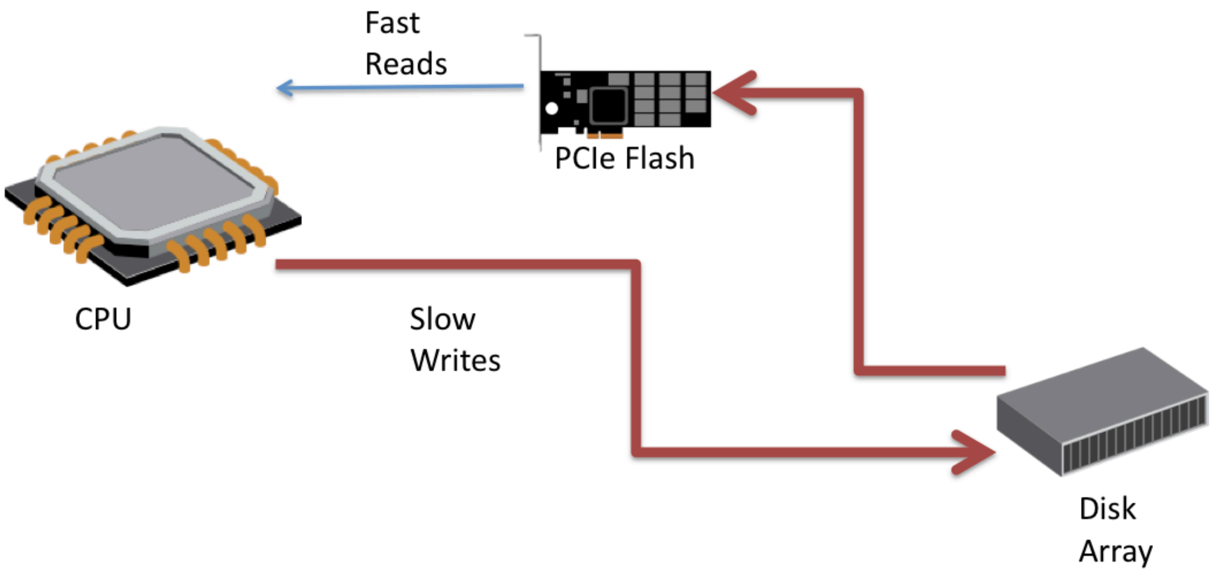

Figure 4.1: Read operations are faster with flash storage. After initial reading from disk, data is cached in flash memory for faster retrieval next time that data is requested by an application.

Write operations can also complete at faster rates when using flash storage. The key to optimizing writes with flash storage is having sufficient storage to hold all write data long enough for it to be written to disk after the write operation requested by an application completes. More details on how to improve the speed of IO operations are presented in the next section.

Methods for Improving the Speed of Reads and Writes

Reads are optimized by having frequently read data in some form of cache, such as a Peripheral Component Interconnect Express (PCIe) flash device on a server. This configuration eliminates network latency and allows data to be transferred to the CPU faster than if the data were stored, even locally, on a disk drive.

The speed of write operations can be improved if the complete write operation, including creating redundant copy or calculating and saving parity data, is performed more quickly. Flash devices can complete write operations faster than disk drives but the speed advantage has to be balanced with high-per-gigabyte cost. Completing a write operation using a flash device and then copying the data to disk can achieve fast write times while quickly freeing memory used by data already copied to disk.

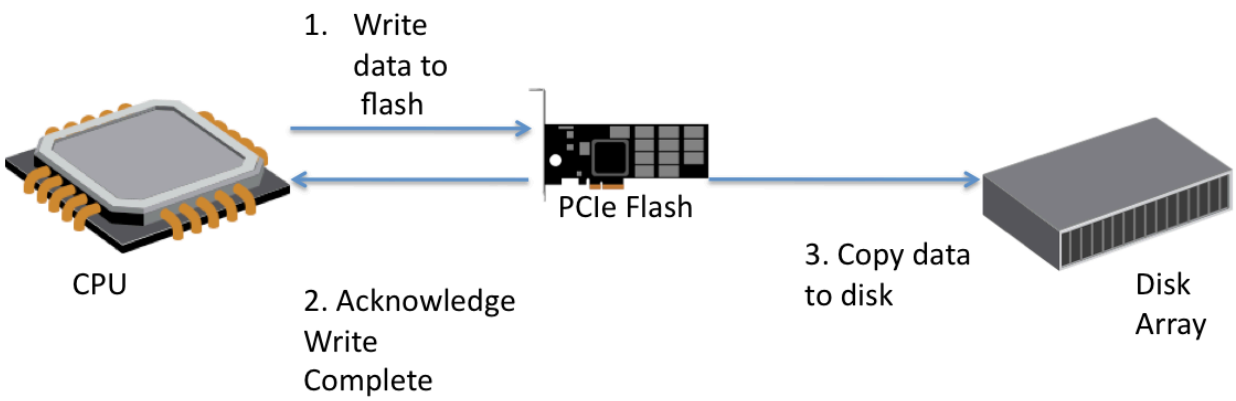

Figure 4.2: Write operations can complete faster on flash devices than on hard disk drives. To optimize flash storage use, data can be written to hard disk drives after the write operation is complete. This allows the application to continue while data is written to disk storage.

Using flash memory as a new storage tier between RAM and hard disk drives can improve performance of read-and-write operations. How much of a performance improvement one achieves is determined, in part, by the storage system architecture. Options include:

- Deploying flash in a storage array
- Deploying a flash storage device between servers and a storage array as a cache
- Deploying flash in servers

Flash can be used in a storage array. A common practice is to use flash in hard disk drive forms so that they can fit into a storage array like a hard disk drive. This has advantages for storage vendors who can extend the life span of their storage arrays by adapting flash storage to their hardware. The cost, however, is the overhead of running unnecessary hard drive protocols on flash devices. Doing so degrades overall performance.

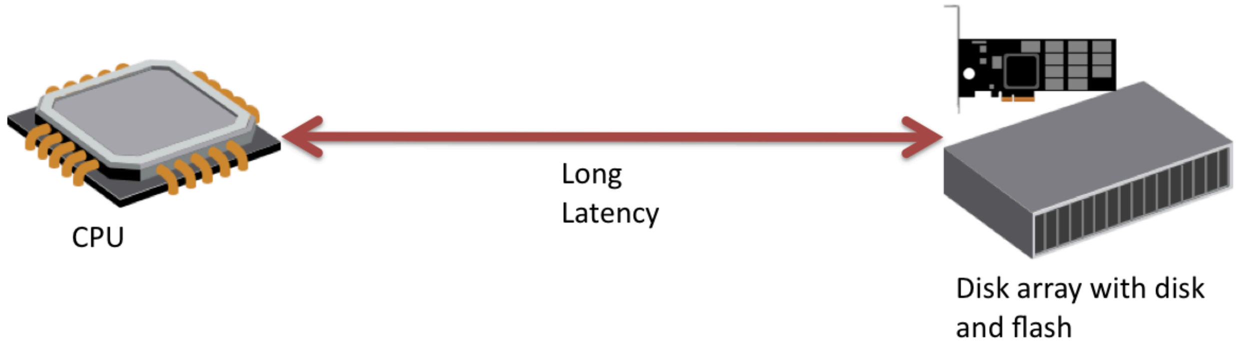

Figure 4.3: Deploying flash in storage arrays entails network latencies and slower read-and-write operations than necessary.

Some vendors choose to implement flash appliances between servers and storage arrays. This setup requires that data move from servers over the network to the flash appliance. Network latencies slow the overall performance in this architecture.

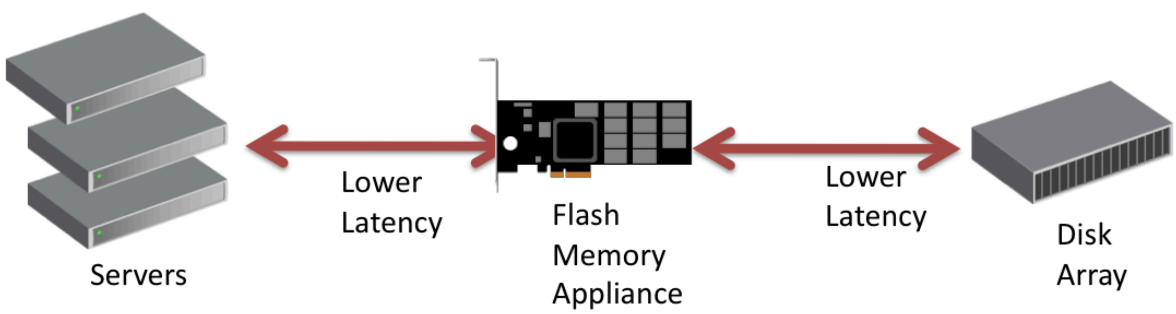

Figure 4.4: Flash appliances between servers and storage arrays incur adverse performance effects because of network latencies.

A third higher performance option is to install flash devices on the PCIe bus of a server. The PCIe bus was designed to reduce the latency of moving data to CPUs. This allows for fewer idle cycles and more efficient use of CPUs. It also avoids network latencies associated with flash in storage arrays or in flash appliances between servers and storage arrays.

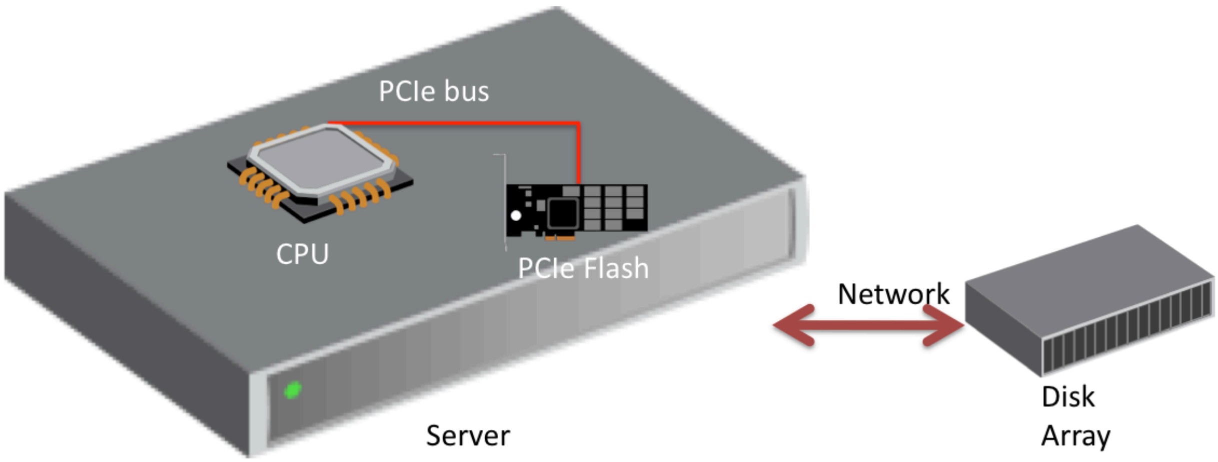

Figure 4.5: Flash on a PCIe bus of a server avoids network latencies and offers the highest level of performance.

One of the reasons flash devices are faster than hard disk storage systems is because flash devices support concurrent IO operations. For example, a PCIe flash device can support up to 24,000 IOPS while an NVRAM/solid state drive (SSD) disk can support only 3,000 IOPS. Disks support serial IO operations and this can lead to long queues of pending IO operations. The SSD can still perform faster than a hard disk drive can perform, but the full performance potential of the flash is not realized because of overhead associated with making the flash memory function within the constraints of hard disk drive protocols.

We have considered methods for improving the speed of read-and-write operations and in the process introduced architectural issues. The next section examines those issues in more detail.

Architecture Issues Associated with Deploying Hybrid Storage Devices

When deploying flash devices in an enterprise environment, it is important to consider how the flash will interact with other storage system components as well as with servers. Consider four ways to deploy flash storage in your enterprise environment:

- AND chips in a disk drive form factor (SSD)
- As read-only cache
- As all-flash arrays
- As PCIe-based flash devices

Technically, a flash-only storage device is not a type of hybrid storage, but it is included here for completeness. A flash-only storage system is an option that might be considered along with hybrid storage solutions, which is sufficient justification for including it in the discussion.

NAND Chips in Disk Drive Form Factor (SSD)

There are definite advantages to adapting a new technology to an older infrastructure. Your existing investment in the older infrastructure can still be used while you realize at least some additional benefit from the new technology. This is the case with using flash storage in hard disk drive form factors, commonly referred to as SSDs.

SSDs simply plug into drive slots much like hard disk drives would. The storage array treats the SSD as if it were a hard disk drive; all the protocols that run on disk drives are run for SSDs as well. This includes:

- Storage protocols
- Embedded controllers that are needed for disk drives
- Legacy RAID firmware

The net effect of this overhead is a bottleneck on performance.

In addition to the overhead associated with hard disk protocols, there can be mirroring or parity operations that require flash storage to implement. When mirroring or parity IS used, the average cost of usable gigabytes of flash storage increases.

Read-Only Flash Devices

Some flash vendors choose to implement read-only flash devices. Doing so improves read performance by caching data in high-speed flash memory. It does not help, however, with write operations. Depending on the mix of read-and-write operations in your applications, a significant amount of IO operations may still be subject to the latencies and performance limitations of disk-based systems.

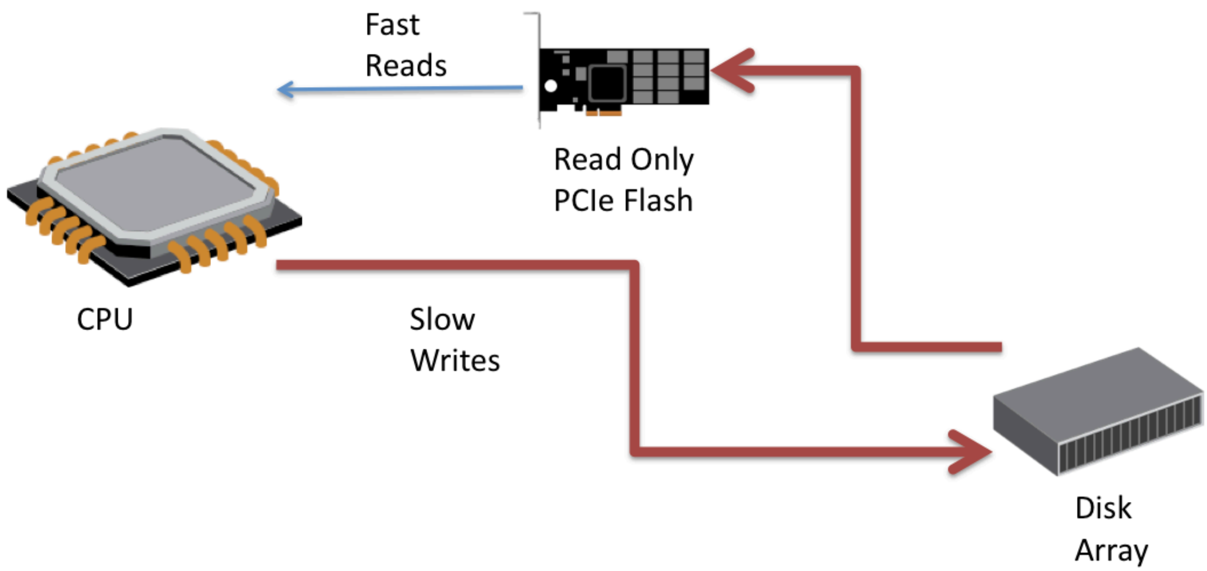

Figure 4.6: Read-only flash devices improve read performance but leave write operations to perform subject to the limitations of disk-based systems.

All-Flash Arrays

All-flash arrays can provide high performance for both read and write operations. The most significant disadvantage of this option is cost. The cost per gigabyte of flash storage is substantially higher than the cost per gigabyte of disk storage. Although there are performance advantages to using all flash, some of those gains can also be realized with a hybrid storage solution.

Consider write operations. Data written to an all-flash storage system would remain in flash even if it is not active. This could require more flash than needed in a hybrid system. With a flash-first hybrid storage system, once the write operation is completed in flash, data is copied to disk.

The application writing data to flash does not need to wait for the write to disk operation to complete. From a performance perspective, the write operation to flash-first hybrid storage takes the same elapsed time as writing to an all-flash device; data is written to flash and once the write operation is completed, data is copied to disk. This approach can leverage a smaller amount of flash memory than a more expensive all-flash device that provides similar performance (see Figure 4.7).

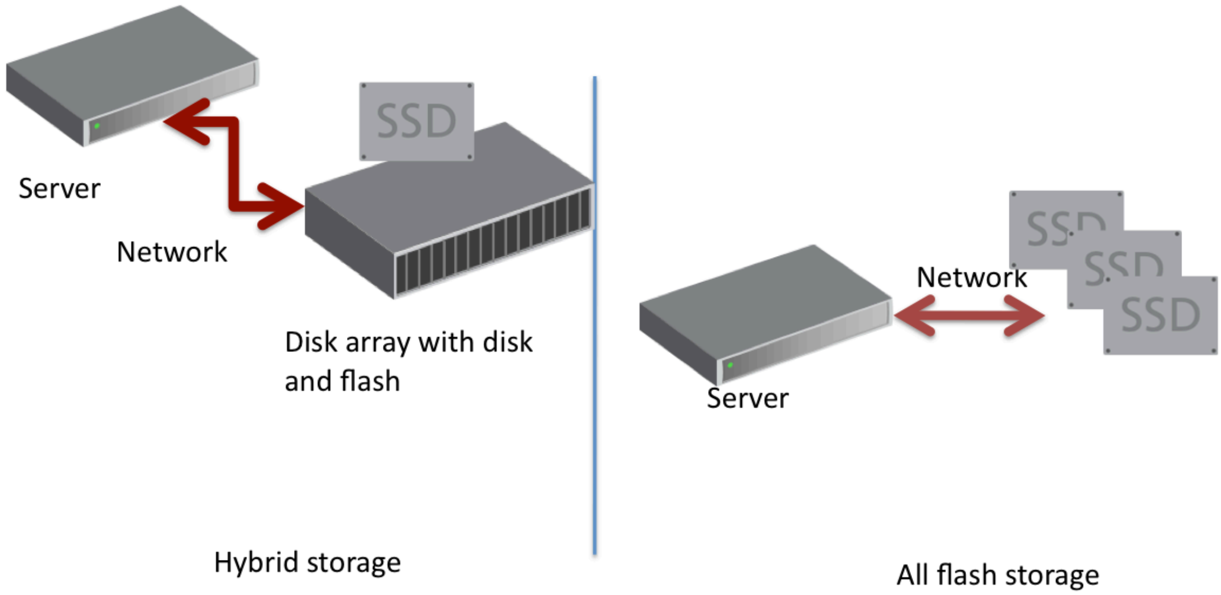

Figure 4.7: Hybrid storage devices can provide comparable write performance to all flash devices.

PCIe-based Flash Devices in Servers

Another option for deploying flash in hybrid storage is to install flash storage in servers using the PCIe bus. This bus standard has higher system throughput and better error detection than earlier bus standards. The PCIe bus uses full duplex communications between directly connected endpoints. This setup, which Figure 4.8 shows, allows flash storage to communicate with CPUs at higher rates than those possible with networked devices or internal hard drives.

In addition to the advantages provided by the PCIe bus, flash storage devices can be managed by the host system. This management enables support for QoS controls and memory optimization techniques to improve overall performance.

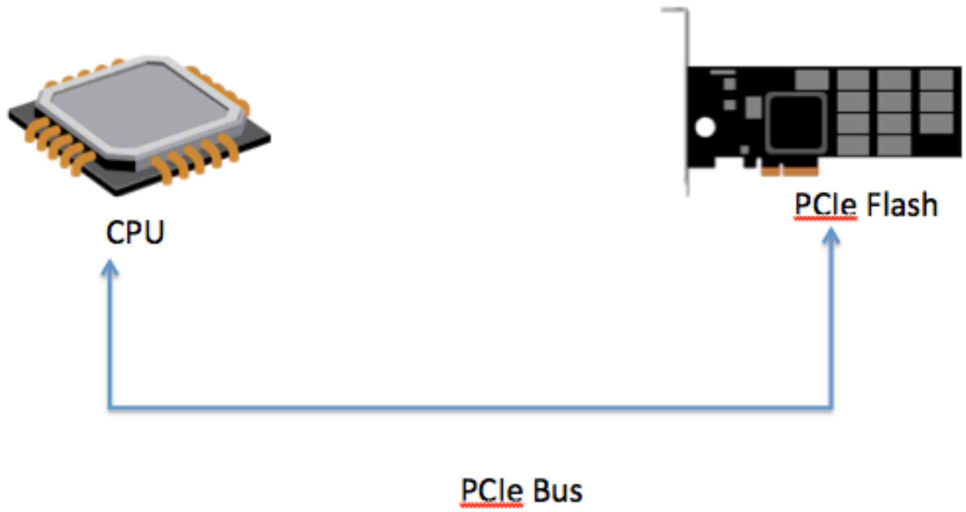

Figure 4.8: PCIe-based flash devices use a high-speed bus within a server to deliver high-performance storage.

Best Method for Deploying Flash in a Hybrid Storage System

The best method for deploying flash in a hybrid storage system takes advantage of the PCIe bus for fast data exchange with CPUs and for executing IO operations at high speeds on flash memory.

Flash memory is used like a new type of memory tier. The CPU still works with RAM at high speeds but now has access to additional high-speed storage with flash storage on the PCIe bus. The new memory tier acts as a temporary mirror for active data.

To improve availability, two PCIe storage devices can be used together. A write operation to such a configuration would work as follows (see Figure 4.9):

1. Application data is written to one of the PCIe flash storage devices
2. The written data is mirrored to the second PCIe flash storage device
3. A write acknowledgement is sent to the application
4. The application continues to execute while, in the background, the recently written data is copied to disk
5. As flash memory becomes filled, an appropriate algorithm (e.g., Least Recently Uses [LRU]) is applied to free memory in the PCIe flash device

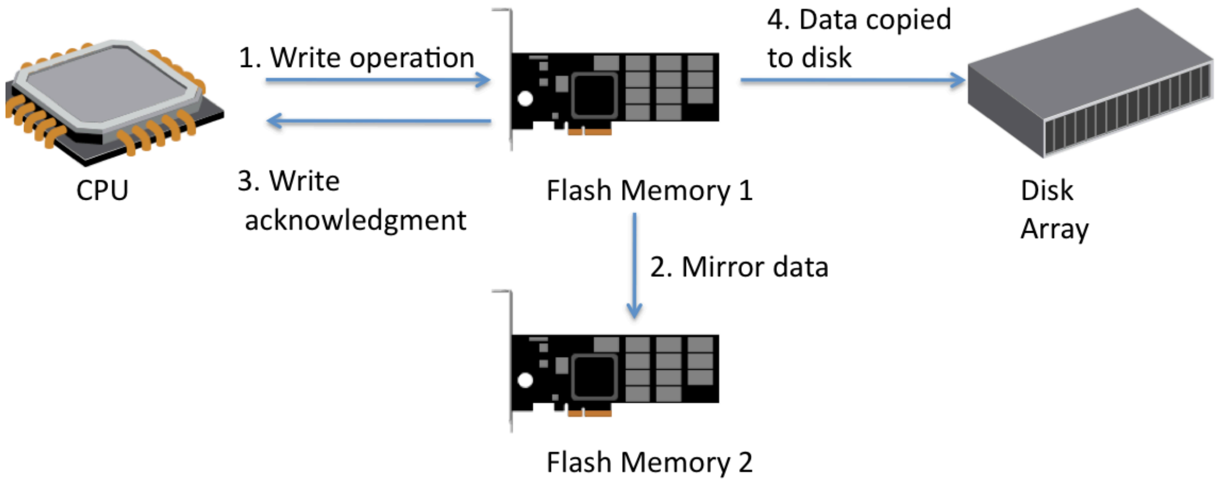

Figure 4.9: Write operation on high-availability hybrid storage system using PCIe flash devices.

The architecture of a hybrid storage system is a top concern when planning for performance. You should not forget, however, the need for realtime QoS controls when considering how to optimize overall performance.

Realtime Quality of Control Services

Between 80 percent and 90 percent of virtualized environments used shared network storage such as SAN or NAS devices. This configuration is efficient from a utilization perspective. Servers can run multiple virtual machines to maximize the utilization of server resources. Storage is shared among devices allowing for optimal use through techniques such as thin provisioning. There is, however, a potential drawback: contention.

Multiple applications may attempt to access a storage resource at the same time. This situation can lead to long queues of pending IO requests. Sometimes, high-priority applications might be held up while less important applications have their IO requests processed. Lack of control over performance during contention can lead systems designers to overprovision and spend more than is truly necessary for a storage system.

A better way to address this problem is to implement QoS controls. With QoS controls, each application can be assigned to a priority class, such as:

- Mission critical
- Business critical
- Non-critical

IOPS are then allocated to meet the needs of the most important applications first (i.e., mission critical) while less important applications are attended to if resources are available.

Consider an example: A transaction processing database used by customers is considered mission critical. If performance degrades, customers abandon shopping carts resulting in lost revenue. This application is clearly mission critical, so it is assigned a minimum of 30,000 IOPS.

The same storage system serves a document management system used by the human resources (HR) department. This system has a relatively light load at most times, but there are periods of peak demand, such as when large numbers of employee reports are generated and stored. This application can tolerate degraded performance without adversely affecting the overall business. This application is assigned 5,000 IOPS.

If there is a period of contention for storage resources, the high-priority database application is given preference. Mission-critical applications are unaffected by contention while business-critical applications degrade slightly and non-critical applications are affected the most (see Figure 4.10).

Realtime QoS controls provide an additional set of tools for managing performance in a mixed workload environment. In addition to QoS controls, a variety of data management practices are important elements of best practices for deploying hybrid storage systems.

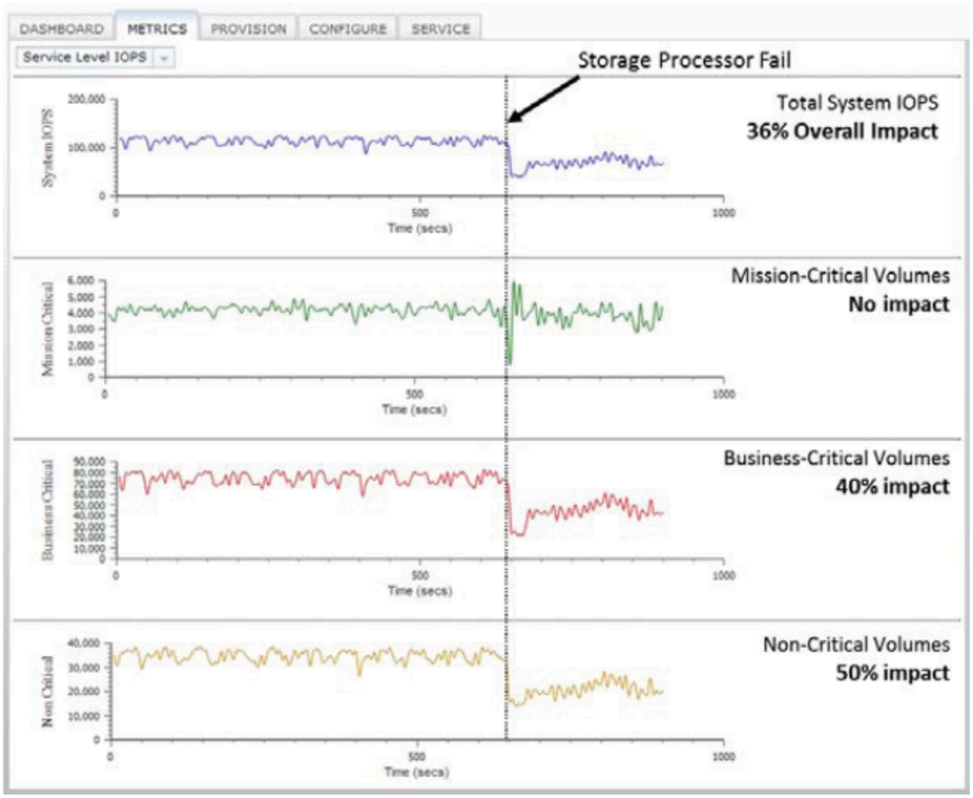

Figure 4.10: During loss of resources, high-priority applications are given preferential access to remaining resources. Only performance of less-important applications is adversely impacted.

In addition to the architectural and QoS controls, consider data management practices when deploying hybrid storage systems.

Data Management Practices

Data management practices are implemented in hybrid storage systems to optimize both the short-term performance and long-term operation of storage systems. These practices include features that provide:

- Data reduction
- Data protection
- Dynamic data placement

The combination of these three data management practices complement the architecture and QoS features to help improve overall storage performance.

Data Reduction

As flash is a high-cost, limited resource, anything that reduces the volume of data stored in flash can help improve the cost per gigabyte of provided storage. Data reduction strategies can help to minimize the amount of data stored in flash, allowing more data to fit into a flash device and increasing the likelihood that active data will be available from flash.

Data Protection

Data protection is an important element of ensuring availability of applications and data. To avoid this situation, many vendors use redundant devices in a single server. Unfortunately, many vendors keep one device active and one inactive in reserve in case of failure in the primary device. This setup provides improved availability but at the same time increases the cost per gigabyte of usable storage when applied to flash storage.

A more efficient use of resources is to deploy redundant PCIe devices in active-active mode. Doing so allows applications to take advantage of both storage devices while retaining the ability to continue to function if one of the two fails.

Dynamic Data Placement

Dynamic data placement is a process for optimizing where to place data in a storage system. Some vendors track block access and build profiles of historical patterns of access. The basic assumption is that if a block of data is hot at one point in time, it will be hot again at another time.

For example, if a data block is hot this week, it will likely be hot next week. Such is probably the case in some instances, but it is difficult to imagine realistic mixed workload scenarios where this generally applies. The high variability in data access patterns undermines this key assumption about data access and can lead to sub-optimal data placement.

A better approach is to allocate flash storage based on an application priority policy. Applications with high priorities are more likely to have data blocks in flash than low-priority applications. If flash becomes full and high-priority applications are not receiving appropriate QoS, data stored for lower-priority applications can be copied to disk to free more flash memory for the higher-priority application.

This kind of policy-driven dynamic data placement provides more control to systems administrators and makes data placement less subject to unpredictable variance in data access patterns.

Summary

Planning for hybrid storage starts with assessing requirements, planning deployments, and formulating guidelines for QoS. The architecture, QoS, and data management topics outlined in this chapter can help guide this process.

Hybrid storage systems can provide a balanced approach to achieving high-performance storage services without excessive costs. Not all hybrid storage systems are equally efficient. PCIe-based flash storage combined with adequate disk storage offers a compelling level of service without the excessive costs of all-flash solutions.