OPTIMIZING RAID PERFORMANCE WITH CACHE

Alex Bouzari, President
Mega Drive Systems, Inc.
489 S. Robertson Boulevard
Beverly Hills, CA 90211
phone: (310) 842-9616 fax: (310) 247-0006
e-mail: abouzari@uu1201.megadrive.com

We live in a world of increasingly complex applications and operating systems. Information is increasing at a mind-boggling rate. The consolidation of text, voice, and imaging represents an even greater challenge for our information systems. Which forces us to address three important questions: Where do we store all this information? How do we access it? And, how do we protect it against the threat of loss or damage?

Introduced in the 1980s, RAID (Redundant Arrays of Independent Disks) represents a cost-effective solution to the needs of the information age. While fulfilling expectations for high storage, and reliability, RAID is sometimes subject to criticisms in the area of performance. However, there are design elements that can significantly enhance performance. They can be subdivided into two areas: 1) RAID levels or basic architecture. And, 2) enhancement schemes such as intelligent caching, support of tagged command queuing, and use of SCSI-2 Fast and Wide features.

Host-independent hardware-based RAID

There are three types of disk arrays: 1) hardware-based, host-independent; 2) hardware-based, host-dependent; and, 3) software-based, host-dependent. Software- based disk arrays are very taxing on the CPU because most of the processing is done in the host computer. On the other hand, hardware-based, host-dependent RAID systems fall short by foregoing the host-side benefits of SCSI.

Therefore, this article will focus on host-independent, hardware-based disk arrays as they typically provide significantly improved overall performance (as measured by throughput and I/Os per second).

RAID levels

RAID 0 uses disk striping to distribute data evenly across all the disks in the array. There is no redundancy or duplication of any data, therefore data-security in minimized. The upside of this scheme is that it provides very high data transfer, and high I/0 rates for both read and write. Supplemented with a well implemented data integrity scheme, RAID 0 can significantly enhance performance in most general applications.

RAID 3 subdivides and distributes each data sector across all data disks, with redundant information stored on a dedicated parity disk. Data can be accessed on different drives concurrently, thereby offering very high data transfer rates, but no gains in I/O rates. RAID 3 is a great performance enhancer for large blocks of data such as video and multimedia type applications.

RAID 5 distributes data sectors as with disk striping, with additional independently computed redundant information. It significantly enhances data transfers and I/O reads, but penalizes writes. RAID 5 offers great performance in most business and database applications.

Other RAID levels that have been proposed to enhance performance, but they typically rely on complex and costly proprietary structures which have not gained broad market and industry acceptance.

Tools Available in RAID Systems to Enhance Performance

A well constructed caching algorithm is essential to a high performance RAID system. This article will cover methods to get the most out of caching.

Intelligent caching

Data transfers can be greatly improved by using adaptive techniques to allocate the optimal amount of cache memory to various read and write command blocks. Varying these segment block sizes will improve cache performance.

Caching and look-ahead

A look-ahead scheme ensures that when the host CPU requests data, the RAID caching algorithm provides the requested data. Look-ahead goes one step further and reads sequential data immediately following the request. That sequential data is written to a cache block on the array controller. If the host CPU requests that subsequent data, it can retrieve it from the cache nearly 1000 times faster than it normally would.

Studies have shown that 55 to 70 percent of all disk requests are sequential. As a result, well designed cache look-ahead schemes keep track of the sequential nature of data and continuously fill the cache with new data based on sequential patterns encountered in user storage activity.

Look-ahead caching eliminates the seek time and latency associated with non-cache transactions and keeps track of the type of drive activity (sequential versus non sequential) as well as the length of time a block of data resides in cache without being requested (FIFO implementation).

Effective array level caches typically range from 8 MB to 128 MB with the cost/performance ratio being optimized for most broad based applications in the 16 to 32 MB range.

It is worthy to note that the disk array's cache complements and greatly enhances the less sophisticated system level cache and smaller drive level caches found in most current hard disk drives.

Caching caveats

Caching offers significant performance gains in a disk array architecture. However, in order to maintain the RAID system's data integrity and fault tolerance requirements, it is critical that the data present in the cache during a system failure be gracefully recoverable.

This can be done by incorporating a UPS (uninterruptible power supply) in the RAID system and providing adequate firmware to flush the cache and carry through the rebuild process without data loss.

Multi-tasking environments

In a multi-tasking environment such as Unix, where a disk array typically services multiple CPU operations, the array must divide the available time among all operations, even though each might be requesting data sequentially from the disk.

Without an adequate caching implementation, the read/write heads in the array will typically seek from one location to another in order to service multiple data requests. With caching, the number of seeks required will be significantly reduced because of segmented cache. After the first seek and read has been performed for each cache, the disk array's cache on board typically takes over and transfers the data directly from the various segments of cache memory.

Tagged Command Queuing

Tagged command queuing (TCQ) allows the host to send multiple commands (from 8 to 64 depending on the implementation) to the disk array for processing. These commands are then tagged and can be reordered in the queue to reduce the time it takes for drives in the array to 1) access specific blocks of data (minimize latency); 2) optimize the use of sequential data; and, 3) increase the number of cache hits, and optimize the execution of a command stream. TCQ is most beneficial in environments which support that feature (such as Unix).

Handling large blocks of data

Another bottleneck in disk array performance has to do with the transmission of large quantities of data. Typical examples are emerging multimedia and related full motion video and video-on-demand applications, as well as traditional multi-tasking and LAN based database and general server applications.

An intelligent way to address these challenges is to eliminate the REQ/INIT/ACK CPU intensive steps usually present between blocks of data by implementing intelligent DMA (direct memory access) techniques.

Similarly, it is possible to reduce the number of interrupts handled during the processing of an I/O request, freeing valuable CPU time. The result is faster throughput to the system, especially for large blocks of data, such as those described above.

SCSI as a bottleneck

The SCSI standard alone can be a potential bottleneck to the RAID disk array (SCSI Fast is only 10 MB/sec. versus SCSI-2 Fast and Wide at 20 MB/sec.) As previously explained, the best disk array performance can usually be obtained in hardware-based, host independent implementations.

Most of the methods we have discussed offer significant enhancements and overcome the performance penalties inherent to the other two aspects of RAID: better data integrity and lower cost. With these tools, a hardware-based subsystem can come very close to the sustained throughput limit of SCSI-2 Fast and Wide (ie 20 MB/sec.).

The future

SCSI is becoming a limiting factor in our performance requirements. Full-motion video, multimedia, and increasingly complex business applications being right-sized from the glass room to personal computers and workstations--will need significantly more power than SCSI can harness.

Intel's P-6 and Motorola's future PowerPC chips will provide the needed processor power. RAID can and will provide high bandwidth storage. What is missing is a faster, more flexible and cost effective interface standard.

Faster Interface Standards

Fiber channel's Gigabit/sec. throughput, under any of its current four or five proposed implementations, shows every sign of fulfilling this promise in the next few years.

Fiber will tie in processor and RAID storage under one high power interface standard and provide us with the high-speed highway needed to support our exponentially growing information needs.

RAID technology is capable of offering the high performance needed to access and process large amount of information, when properly implemented. There are many factors that contribute to RAID performance. The key is to assess the specific storage and application requirements, and select the most appropriate RAID scheme. Once this is done, the RAID system can offer significant performance gains over JBOD (Just A Bunch of Drives) by using the tools such as the ones discussed here.