RAID 7: Architecture and Functionality

ABSTRACT: The Berkeley RAID paper postulated and described five levels of architecture for Redundant Arrays of Inexpensive Disks (RAID) and many authors have described levels 1-5 in great detail. Published details on the higher number RAIDs have been less plentiful. This paper presents a comprehensive examination of the architecture of a RAID 7, and also analyzes its functionality and derived benefits.

Introduction

All RAID levels are comprised of **arrays of disks** and each incorporates a **measure of redundancy** into its data storage capabilities. These two primary attributes define the very essence of a RAID. The definition does not stop with those two properties, however. The "measure of redundancy" attribute, is a RAID feature bought and paid for with two distinct currencies: (1) storage capacity, and (2) speed performance of the array. While many readers will find this double expenditure readily apparent, any true understanding of different RAID level performance must incorporate and address the existence of this dual tradeoff.

Many publications[1] [2] have discussed and summarized RAIDs 1 through 5 and it is felt there is no need to review that material here. Rather we will focus on the architecture of RAID 7, examine the functional performance characteristics which emanate from that architecture, and compare and contrast the differences with the lower level RAIDS.

RAID 7: An Uncompromising Architecture

Each RAID level reflects a different design architecture. Associated with each is a backdrop of imposed architectural limitations, as well as possibilities which might be exploited within the architectural constraints of that level. For example, RAID 1 is a data mirroring approach which defines that each byte of data is stored twice. Therefore the usable capacity of the RAID 1 drive array can never exceed 50% of total formatted storage capacity, as this is an architectural design constraint of the RAID 1. In terms of design possibilities within RAID 1, it is possible -- but not required -- for a RAID 1 array to make use of dual controllers and data paths so that read accesses may be executed with faster access times as viewed by the host.

RAID 7 likewise supports an architecture where it is possible to include several design features which increase performance but are not strictly required to satisfy the RAID 7 definition. What are the unique architectural features that differentiate a RAID 7 from other levels? There are three: (1) RAID 7 is asynchronous with respect to IO data paths, (2) RAID 7 is asynchronous with respect to bus utilization, and (3) RAID 7 is asynchronous with respect to a real time process oriented OS.

More specifically the unique features of RAID 7 are as follows:

(1) **RAID 7 is asynchronous with respect to usage of I/O data paths**. Each I/O drive (includes all data and 1 or more parity drives) as well as each host interface (there may

[1] D. Patterson, G. Garth, R. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)", University of California, Berkeley, Report No. UCB/CSD/87/391, December 1987.

[2] J. Moad, "Relief for Slow Storage Systems", Datamation, pp 22-28, September 1, 1990

be multiple host interfaces) has independent control and data paths. This means that each can be accessed completely independently of the other. This is facilitated by a separate device cache for each device/interface as well.

(2) **RAID 7 is asynchronous with respect to device hierarchy and data bus utilization**. Each drive and each interface is connected to a high speed data bus and controlled by the embedded operating system to make independent transfers to and from central cache.

(3) **RAID 7 is asynchronous with respect to the operation of an embedded real time process oriented operating system** . This means that exclusive of and independent of the host, or multiple host paths, the embedded operating system manages all I/O transfers asynchronously across the RAID 7 data and parity drives.

Figure 1 depicts the RAID 7 block diagram. As illustrated, each device and interface is symmetrically positioned as a peer. The separate device cache and separate device control shown in the figure, permit asynchronous I/O transfers which are independently controlled and cached. This applies to multiple host interfaces as well.

A dedicated single parity drive (P1), is shown in figure 1, and it may occupy any channel of the array because its function can be defined by the embedded-OS at power-up. Also shown are channels for up to N number of data drives, as well as up to M number of Resource drives. The Resource drives represent optional system resources; these may be one or more hot stand-by drives, or a second (P2), a third (P3) or more parity drive(s). Note that while RAID 7 provides for multiple hot stand-by and secondary, tertiary, and beyond parity calculation and associated drives, this architecture does not require those options.

The central cache shown in Figure 1, supports reads and writes via the high speed bus. Please observe the central cache connects directly through the high speed bus to the device cache of each device/interface. This device cache is separate and distinct from any "on board" drive cache. Thus RAID 7 enjoys internal cache-to-cache high speed transfers for all I/O operations. The complete asynchrony and the device independence of these high speed transfers are some of the sources of RAID 7's speed and scalable reliability advantages.

The embedded RAID 7 microprocessor control logic, bus control logic, the high speed bus control logic, the cache control logic, the parity generation and check control logic, and the resource drive(s) control logic, are all tied together and managed under the control of the embedded operating system. This self-contained and embedded OS is the glue that cements everything together. Central cache is optimized and managed by the RAID 7 Operating System. High speed I/O transfers occur to and from device cache, asynchronously, under the control of the RAID 7 Operating System. Each read/write to the parity drive is managed, scheduled, and executed, in an optimized asynchronous manner, via the RAID 7 Operating System.

Also depicted in Figure 1, is a two way serial link between the host-OS and the embedded-OS in the RAID 7 device. This is an optional connection. Its purpose is to provide a communication path for special processes that the host wishes the RAID 7 device to perform. Implicit in this design is that the host vendor/sophisticated user would author and link code into the open-system embedded-OS present in the RAID 7. Storage Computer, for example, has written its embedded-OS, called SOS (StorComp Operating System), to execute on an Intel

286/386/486 ISA PC compatible board. The capability to write processes for both the host and RAID 7 puts more control into the hands of the host vendor/sophisticated user. This dramatically lowers the development costs for all kinds of innovative features such as: background archiving, background defragmentation, mission critical archiving, write journals, audit trails, and security features, to name just a few. Many RAID 7 users may not need this serial host connection, but the elegance of the RAID 7 architecture is that such a powerful set of features can be had via a software construct. Simply said, one single facet of the RAID 7 architecture, the host-OS to RAID 7 embedded-OS serial link, provides a world of performance and feature enhancement opportunities that will dramatically extend the useful life of the RAID 7 subsystems.

One of the more distinguishing architectural features of the RAID 7, is the integration of the process oriented embedded-OS for the real time control of all I/O through independent data paths. This feature explicitly recognizes and capitalizes on the inherently asynchronous nature of host viewed disk storage. This is a subtle concept that deserves an expanded exposition.

Host Viewed Disk Storage - Or, How To Avoid the System Boundary Trap

Individual disks achieve their greatest throughput when they receive large sequential reads/writes. No one disputes that. However, it is the nature of individual disks to be always connected to something - usually something that has a host operating system. Therefore when users and vendors conceptualize the performance of a "large write" or "random read/writes" they may allow themselves to be misled unless the question is asked, "where do we draw the system boundary?". Clearly, the most predictive results will be gained by considering host viewed disk storage, since the host fulfills all of its I/O transfers looking through the prism of its respective host operating system.

The demands made on a storage unit by the host -- especially a multiuser host -- are not heavily weighted with large data transfers. In fact even when a large transfer request is made by the end user, the host operating system will translate that request into slices of "more reasonable size", and intersperse those component requests with other disk access requests. This approach simultaneously represents three things: (1) good host-OS methodology (2) a disk storage device's worst nightmare, (3) every-day reality.

Consider this host access activity from the perspective of the storage unit. The host appears to make a heavily read weighted, and visibly asynchronous serial stream of transfer requests. Those requests, although often unexpected, are not truly random. They are characterized by two kinds of locality: spatial and temporal. An example of spatial locality might be accesses to sectors A,B,and C, of Figure 2, in succession. While the sequence I,X,J,A,B,K,Y shows a temporal locality since the contiguous locations I,J,K were accessed within a relatively short timeframe. (Note that this figure represents the multiple disk array disks the way the host views them -- as one large integrated disk.)

Inherent in the RAID 7 is an ability to make use of both spatial and temporal locality to improve host I/O access times. The embedded-OS can anticipate a "new" access of the disk and prefetch it. Thus in Figure 2, after the access to sectors I and A and X. The embedded-OS would use available time slices to automatically move sectors J,K and B,C and Y,Z to device cache, and then to central cache (see Figure 1). Subsequent accesses -- even accesses which have some time delta between them -- would find the requested data in cache and hence be transferred to the requesting host at functionally zero latency.

Figure 3 presents a combined view of the architectural aspects of RAID 7 (shown in Figure 1) and the various disk sectors-of-interest requested by a group of unrelated users -- (shown in Figure 2). Thus Figure 3 represents a kind of practical usage of the RAID 7 device. As far as the host-OS is concerned, in Figure 3, it is dealing with a single disk with sectors of interest represented by Figure 2; A,B,C are sequential sectors. The RAID 7 embedded-OS preserves the meaning of these sequential sectors while mapping them to different physical disk drives in the array. Access to sector A causes an automatic anticipation and prefetch of sector B by the embedded-OS. The sectors I,J,K are served upon request and remain in central cache -- as long as permitted by the embedded-OS mechanisms. Thus, from the point of view of user 2, a "process" has been created for him which will result in optimizing both spatial and temporal accesses to his disk sectors of interest. Subsequent accesses -- even accesses after other requests -- to I,J,K are served from cache. Aggregating this operation across all users results in a very efficient servicing of host I/O transfers.

This ability to have the embedded-OS to anticipate and match host-OS I/O usage patterns is another principal benefit that the embedded-OS contributes to RAID 7. Because the system boundary was drawn corresponding to its use in the system, the RAID 7 architecture both complements and augments the manner in which the host makes storage demands. The RAID 7 OS component merges with the asynchronous hardware structure of the RAID 7 architecture to achieve synergy between host expectations and disk drive behavior. This yields the following three heretofore mutually exclusive benefits.

(1) From the perspective of the host computer, it appears as a normally connected BFD (Big Fast Disk).

(2) From the perspective of the individual disk devices in the array, it appears as a kinder and gentler host, that minimizes the total number of accesses and optimizes read/write transfer requests.

(3) From the perspective of the RAID 7 device itself, it smoothly integrates the "random demands" of independent users with the principles of spatial and temporal locality. This optimizes small, large, and time sequenced I/O requests which results in users having an I/O performance which is nearer to main memory performance.

RAID 7 Functional Performance Characteristics - The Six Yardsticks

Implicit in any discussion of performance is a set of measurement criteria which will be used to evaluate the device under study. We argue that there are six appropriate performance criterion which ought to be used in any RAID discussion. These six yardsticks are as follows:

(1) Usable Storage Capacity

(2) Sustained Host I/O Transfer Rates

(3) Small-R/W Large-R/W Metrics: Bandwidth and IOPs

(4) Access Times

(5) Scalable Reliability

(6) Scalable Performance

Let's examine each criterion in greater detail, and see what functional performance is offered by the RAID 7 architecture.

Usable Storage Capacity

At the business end of every RAID is a finite amount of unformatted storage. After all the formatting and data striping, and sector mapping, and parity overhead is accounted for, a single question remains; "what is my usable storage capacity?".

One of the features of the RAID 7 architecture is that the usable storage capacity is not a function of sector, block, striping, or write request size. Unlike a RAID 3, which depending on implementation can waste fully 50% to 90% of the storage capacity if the sector size does not match write request size, the RAID 7 pays no such penalty. RAID 7 places no requirements on formatting its element disks so that the total usable storage capacity is always the sum of the standard formatted capacity of the drives less one.

RAID 7 likewise places no other requirements regarding drive form factor, rotation speed, transfer rate, drive synchronization, or drive capacity. The embedded-OS will normalize all drives in the array to the same capacity. If a drive is placed into service which is larger than the other drives, RAID 7 will proceed to use it as the same capacity as the other drives. This unrestricted approach to disk usage sharply contrasts to the lower level RAIDs. This results in the following benefits by using drives that are: (1) standard off-the-shelf, (2) not synchronized, (3) already installed in use, (4) of mixed form factor, (5) of mixed manufacturer, (6) of varying capacity, and (7) of varying performance. The RAID 7 device epitomizes the classic definition of economics, "doing the best with whatever it has".

Sustained Host I/O Transfer Rates

Those familiar with either specsmanship or with hardware design know all too well the difference between **sustained-rates**, and **burst-rates** and **rates**. The latter two are almost always identical. The real issue as far as RAID I/O performance is concerned is, "what is the sustained transfer rate to the host?" A brief re-visit to Figure 1 would be helpful at this point. In the RAID 7 device the N Data drives (shown in Figure 1) represent the available bandwidth to store data. If N were to be 5, and those drives were capable of a sustained transfer rate of 200 KB/s, then the RAID 7 could offer the host a 5*.95*200 KB/s or 950 KB/s sustained transfer rate. If N were to be 40 with the same 200 KB/s per drive, then the RAID 7 could offer the host a 7.6 MB/s sustained transfer rate. It is significant to note that unlike other RAID levels, RAID 7 offers a linear increase in sustained transfer capacity as the number of drives increase.

It is possible to put tremendous storage bandwidth in a RAID device, however, the only part of that bandwidth that is usable is that which can be offered to the host in a sustained fashion. One of the unique features of the RAID 7 is an ability to accommodate multiple numbers of host connections. Thus the 7.6 MB/s sustainable bandwidth of storage could be delivered to a host through two 4 MB/s controllers, or eight 1 MB/s controllers.

Unlike RAID 3 and RAID 5, the RAID 7 transfer rates are independent of the type, mix, and size of read/write requests. Unlike RAID 5 the RAID 7 device expends no overhead to support a rotated parity distribution scheme. Unlike RAID 3 and RAID 5, RAID 7 supports multiple host connections.

Small-R/W Large-R/W Metrics: Bandwidth and IOPs

Perhaps the easiest and simplest measure of a RAID device ought to be whether it can answer, "yes" to the following four questions:

1) Can the RAID perform SMALL READS better than a single spindle?

2) Can the RAID perform SMALL WRITES better than a single spindle?

3) Can the RAID perform LARGE READS better than a single spindle?

4) Can the RAID perform LARGE WRITES better than a single spindle?

Unlike the RAID 3, which offers an average access time longer than that of a single spindle and can not match single spindle performance for small writes, and unlike the RAID 5 which can not match single spindle performance for large writes, and which for some small writes can muster only 1/20th of single spindle performance, the RAID 7 can answer yes to all four questions since it exceeds single spindle performance in all four cases.

Most of the previously published discussions of RAID performance have taken place by comparing different measures with different RAID architectures. For example, MegaBytes/second (MB/s) are used to evaluate RAID 3 while I/O's per second (IOPs) are used to measure RAID 5. The problem with such comparisons is twofold: (1) they do not match the real world systems which most always have a continuous mix of small and large requests, and (2) they mask the disastrous performance of the untested measure.

The key to understanding RAID performance is to compare each RAID level to the performance you get using only a single spindle. The results indicate that RAID 3 fails miserably on small reads and small writes and RAID 5 handles large and small writes poorly. RAID 1 manages small and large reads as well as the single spindle level; and executes large and small writes less than a single spindle. RAID 7 consistently outperforms the single spindle rates, and frequently achieves this at substantial performance multiples.

Access Times

Long a figure of merit in evaluating single spindles, average access time remains an important parameter in gauging RAID performance. Yet the concept of access time for a multiple spindle device remains an elusive concept requiring a more refined perspective. When we discuss a RAID access time, are we gauging the access time to commence a single Input Output Operation (IOP)? How does cache effect this? How do other optimizing features of the RAID array effect this? What does this aggregate access rate really represent? Is it influenced by the bandwidth of the host controller, or is it independent of all host channel characteristics? Is it influenced by the host-OS performance in any way, or is it completely independent of the host-OS IOP transfer size? There are many questions and too few cut-and-dried answers.

When we talk of multiple spindle devices the access time measurement is influenced by host parameters as well as specific RAID parameters. More specifically, the access time performance of a RAID will be a function of the host-OS IOP transfer size, the channel capacity of the host adapter interface, the number of host interfaces to the RAID, the RAID level, the amount of cache in the RAID, the number of disks in the RAID, and the type of disks in the RAID. Perhaps the most reasonable and non-deceptive approach in considering RAID access times is to approach them from the following three different viewpoints: (1) worst case, (2) best case, and (3) the representative case.

The worst case for RAID 7 is represented by a scenario where some of the RAID 7 optimizing features would not be invoked. This worst case can be characterized by a series of non-queued, non-contiguous accesses. If these access sizes are large the RAID 7 access rate would twice as fast as the single spindle rate; if the accesses are of small size (a sector) than the access rate performance would equal the single spindle rate. Note that in no case is it slower than the single spindle rate. This clearly differs from a RAID 3 device where even the average access rate is slower than single spindle performance.

Best cases for a RAID 7 are represented by the highest percentage of requests satisfied from cache. This is characterized by a queued request stream, or a request stream of contiguous accesses. In a RAID 7 device with N Data drives of 16 ms if N were to be 10 then an average access time figure, as seen from the host, would be 16/10 or 1.6 ms. The absolute best case access rate would be a function of the frequency of satisfying a request from central cache at 80 ns or 200,000 times faster than a single spindle.

The representative case in a RAID 7 device will tend towards the best case rates consistent with the queuing and degree of contiguous requests. The RAID 7 embedded-OS always anticipates and prefetches each read storing cache with probable accesses. Thus it is not possible to "miss the next sector" in terms of paying a rotational latency penalty. This means that subsequent accesses become "cache-hits" and performance tends to drift towards the best

case scenario. A relative poor hit rate of 80% would yield a host viewed average access time of .32 ms or 50 times faster than a single spindle. A more reasonable hit rate of 90% translates into .16ms or 100 times faster. Under extreme host request I/O loads the host I/O adapter will approach its saturation point and all access times for the RAID 7 will be masked from the host.

Scalable Reliability

By definition, data fault tolerance -- the ability of a disk array to withstand full or partial failure of a single disk and still provide access to the data on the damaged drive -- is provided by all RAID levels. The independent structure of the RAID 7 architecture -- with separate device cache and separate device control, and embedded-OS -- permit the optional addition of a secondary, tertiary, and beyond parity calculation. These could be used to protect against as many simultaneous drive failures as the user's pocketbook could afford. RAID 7 is unique in that it can provide this additional parity protection without a severe performance penalty. An additional RAID 7 reliability option is to provide full fault tolerance by linking two RAID 7 platforms to the same physical drives. This configuration also doubles read performance.

RAID 7 is likewise unique in its ability to support **multiple** hot stand-by drives. This improves maintainability repair times. The asynchronous architecture of RAID 7 permits the easy addition of hot stand-by drives at any time during the life cycle of the RAID 7 device. These RAID 7 optional features are unlike the other lower RAID levels where adding a second parity drive entails a significant disruption of the architecture and simultaneously devastates associated performance.

Scalable Performance

Most computers and systems serve a user community which is anything but static. The dynamic demands on type and pattern of usage -- not to mention capacity -- change at a dizzying rate. Those responsible for managing RAID usage in user environments frequently want to know what the effect is of adding additional capacity, or what can be done about increasing throughput.

Unlike RAID 5 which can only be scaled up in multiples of its specific write group size, the RAID 7 is capable of being linearly scaled up. Furthermore, the result of increasing the number of drives to a RAID 7 configuration is to improve performance at a non-linear rate. This is the **Ratio Engine** effect of the RAID 7 architecture. The RAID 7 performance improves as the number of drives increases. This is a direct function of the ability of the embedded-OS to supply many of the requests from cache, while optimizing the individual read/write performance of the individual drive elements.

This combined effect allows a RAID 7 device to mask individual drive characteristics from the host. This means that slower and lower cost drives are prudent choices to construct a RAID 7 implementation in many applications. The RAID 7 is not sensitive to drive rotational speeds or dependent on individual drive transfer speeds to present credible end use performance.

Conclusions

The design architecture of RAID 7 explicitly recognizes the asynchronous nature of disk drive usage, and thereby optimizes the (host viewed) process of asynchronous data transfers. RAID 7 achieves this in a manner which is absolutely uncompromising in terms of usable storage capacity, sustained host to I/O transfer rates, small reads and writes, large reads and writes, access times, reliability options, general performance, and optional performance features. Conclusion: by any yardstick, RAID 7 is the RAID of choice.