CAPACITY PLANNING

Is RAID 5 Really a Bargain?

Cary Millsap, *Hotsos LLC*

RAID 5 may look like a bargain compared to RAID 1, but for many Oracle systems it's not. Researchers in the 1980s invented the RAID 5 disk organization in response to the high cost per byte of RAID 1 (mirroring). To achieve a lower cost per byte, RAID 5 sacrifices several performance attributes that are relevant for most Oracle systems.

• **RAID 5 costs more for write-intensive applications than RAID 1.** The so-called "small-write penalty" inherent in the design of RAID level 5 disk arrays makes each Oracle DBWR write require *four* physical I/O operations. Consequently, to provide adequate throughput capacity for a write-intensive Oracle application, an architect must use about *twice as many disk drives* as if he were using RAID level 1 (mirroring).

Example: To obtain four disks' worth of storage capacity in a typical RAID 5 system, you must buy five disks. To obtain four disks' worth of storage capacity on a RAID 1 system, you must buy eight disks. By a cost-per-byte of storage capacity analysis, RAID 5 is only 63% the cost of RAID 1.

However, an eight-disk RAID 1 array can execute four small writes in parallel. A five-disk RAID 5 array can execute only two small writes at a time, and even then each RAID 5 small write will consume almost twice the service time of an equivalent RAID 1 small write. The cost of RAID 1 per unit of small-write throughput capacity is only 80% that of RAID 5.

The RAID 5 small-write penalty impacts the performance of all small-write operations. Architects of mostly read-only systems should assess the attributes of RAID 5, because the configuration does offer excellent read performance at a good price per byte of storage capacity. However, it is important not to neglect small-write performance penalty upon "extraordinary" events like data warehouse load processes or media recovery.

• **Caching helps, but at a price.** Some RAID 5 manufacturers claim that installing large amounts of battery-backed memory (called NVRAM) solves the problem. Cache does offset the performance penalty in two ways. First, caching helps defer the physical I/O operations. Hence, the small-write penalty doesn't necessarily bottleneck the individual I/O call that motivated it. Second, caching helps a disk array to group small writes into larger batches that don't incur the small-write penalty.

There are two downsides of this caching strategy. First, cache is expensive, which at least partially defeats the RAID 5 cost-per-byte-of-storage advantage over RAID 1.

Important: The cost-per-byte objection to RAID 1 is the motive for considering to endure the RAID 5 write-call performance penalty in the first place.

Second, your I/O throughput requirement may exceed your cache capacity. On a cached RAID 5 array, the cache of course fulfills I/O calls faster than the array can actually complete the calls to disk. If I/O calls keep coming quickly enough, the cache can fill, resulting in a phenomenon known as "cache cramming." When a RAID 5 array cache "crams," all I/O to the array will cease—potentially for dozens of seconds—until the array can resynchronize with the physical disk.

• **RAID 5 is less outage resilient than RAID 1.** A RAID 5 array is about three times more likely to incur data loss each year than a RAID 1 array with equivalent storage capacity. As you factor in the larger number of RAID 5 disks needed to provide similar performance to that of a high-throughput RAID 1 array, the RAID 1 resilience advantage widens even further.

• **RAID 5 suffers massive performance degradation during partial outage.** A RAID 5 array is resilient to a single disk outage, but I/O performance for the array degrades brutally during the outage. Until the array is repaired, every I/O call upon the failed disk will require a read or write from every surviving disk in the array.

The performance degradation will last until after the failed disk has been replaced *and* the replacement disk is resynchronized. The resynchronization event is extremely I/O intensive, requiring a read of every block from every disk in the array. Of course, the resynchronization reads compete for I/O service with the normal I/Os already being generated by your application.

By contrast, RAID 1 writes actually get *faster* during a disk outage. Resynchronization of a replacement RAID 1 disk involves I/O only to the replaced disk's mirror disk (called *resilvering* in RAID 1 vernacular).

• **RAID 5 is less architecturally flexible than RAID 1.** RAID 1 opens architectural opportunities that RAID 5 does not.

Example: Many sites use a "triple mirror" strategy to provide both fault resilience *and* a fast backup/recovery capability. Data copies #1 and #2 serve as mirror images in a regular RAID 1 array. Data copy #3 is resilvered nightly during the backup procedure (*alter tablespace… begin backup*, resilver, *alter tablespace… end backup*).

The resilver operation, a disk-to-disk copy, executes more quickly than a disk-to-tape copy. This speeds the backup and reduces the amount of undo generated by the database if it is active during the backup. Recovery is faster as well, because the media recovery operation is disk-to-disk instead of tape-to-disk.

- **Correcting RAID 5 performance problems can be very expensive.** Architects who don't understand these aspects of RAID 5 often find their RAID 5 systems unable to provide the performance that is required by their users. The fix is expensive, requiring either the purchase of many more RAID 5 disks than was originally planned, or the conversion of the whole disk subsystem over to a RAID 1 configuration that costs less per unit of write-throughput.

Cary Millsap is a manager of Hotsos LLC and the editor of *Hotsos Journal*. He is the author of several papers and presentations on system performance topics. Prior to joining Hotsos LLC, Mr. Millsap was a vice president at Oracle Corporation, where he founded the Core Technologies group and the System Performance Group.