What is RAID?

- RAID is the use of multiple disks and data distribution techniques to get better Resilience and/or Performance
- RAID stands for:
  - Redundant Array of
  - Inexpensive / Independent Disks
- RAID can be implemented in Software or Hardware or any combination of both
- This presentation is a simple introduction to the RAID levels with some information on Caching and different I/O profiles
RAIDO - Striping

- Simple striping of data across disks in the array
- No “RAID” resilience - if one disk fails, then the entire array fails
- Good for load balancing & elimination of “hot spindles”
- Usually implemented in conjunction with RAID1 to get resilience
- Array capacity = disk capacity x n where n is the number of disks in the array

\[
\text{capacity} \times n = \text{array capacity}
\]
RAID1 - Mirroring

- Mirroring, shadowing, duplexing for resilience with 2nd copy of data
- Array capacity = disk capacity x n/2 where n is the number of disks in the RAID1 set (or n/3 for triple mirroring)
- Can be used to mirror RAID0 RAID sets with other identical RAID0 sets (RAID01 or RAID0+1)
- Can enable simultaneous reads on both parts of mirror

\[ \text{capacity} \times \frac{n}{2} \]
RAID3

- Fine granularity data striping with ECC based disk redundancy
- Dedicated ECC disk (original definition)
- Highest throughput and simple write process
- Array Capacity = disk capacity x (n-1) where n is the number of disks in the RAID set
RAID5

- Data striping with ECC based disk redundancy
- All disks have round-robin share of ECC data
- Good small block read I/O performance but complex write process
- Array Capacity = disk capacity x (n-1) where n is the number of disks in the array

`capacity x (n-1)`
RAID6

- Same as RAID5 but with additional spindle and double ECC redundancy
- Allows two disks to fail in the same RAID set
- Very complex write process
- Array capacity = disk capacity x (n-2) where n is the number of disks in the array

Array capacity = disk capacity x (n-2)

Data reliability with performance
RAID Application Theory

- The 1988 RAID paper came up with the following observations (without any commercial RAID systems to prove or disprove and without consideration for any enhancements such as caching)
  - RAID5 was promoted as best for TP and database workloads
  - RAID3 was considered best for scientific workloads
  - Other applications fall between the two extremes in terms of typical I/O size

![Diagram showing application range of RAID5 and RAID3]

- Small transfers
- Large transfers
- Transaction Processing
- NFS, Novell, Client Server
- Multi-Media
- Imaging, Seismic
- Scientific, Video
- Database Enquiry
- Database Enquiry
Using Cache to boost RAID5

- Cache can be used to temporarily buffer write-data from the host to allow faster I/O completion.
- However, RAID5 requires lots of cache to overcome the basic complexity of the write process. This reduces the amount of cache available for write buffering and allows the cache to become saturated in heavy load scenarios.
- Caching does not significantly boost throughput as small I/O’s benefit most - large transfers are typically ignored by cache.
- This means that RAID5 based systems can never be fully suited to the entire range of applications.
Using Cache to boost RAID3

- Just as with RAID5, cache can be used to buffer writes for fast I/O completion
- Unlike RAID5, parallel RAID systems do not need to use cache to overcome a complex write process
- This means that a much greater percentage of cache space is available for genuine work and as a result a RAID3 system with cache is not saturated as easily as an equivalent RAID5 system
- The extra cache space can also be used for hot read data
- Small I/O’s benefit most from caching, extending the suitability of a RAID3 system with cache into the OLTP and DBMS range
Summary

- This presentation just covers the relationship between disks in the basic RAID setups, and introduces the concept of caching.
- Data integrity issues are not covered