RAID- Redundant array of independent/inexpensive disk (can be implemented as software/hardware methods)

Speed, fault tolerance & Redundancy

SW RAID done at operating system level, cost effective &easy to implement, not great in terms of performance

HW RAID, dedicated hardware on host or array. Card installed on motherboard. RAID controller act as interface between host and disk

Techniques – Striping (Data is distributed across multiple disk and used in parallel)- Great performance, zero fault tolerance.

Stripe size= chunk size \* number of disk in RAID set (ex= 16000\*6=96k)

Mirroring (Two physical drives are placed in mirror pair. Data stored in one copied to another. Redundancy provided, medium write performance, good read performance)

Parity (provide data redundancy by error checking and data recovery. It can be odd or even)

**RAID 0(minimum 2 drives**)- **Striping** (Bit, byte & block)-**high read &Write Performance – No redundancy no availability**

**RAID1(2 drives)- Mirroring.** provides **redundancy and high availability (low to medium write speed, good read speed)- Expensive**

RAID2- (Minimum 3 drives) drives must spin at same angular orientation**. Error correcting code**. Split data bit wise. Faster reading. Slow write -Expensive(No longer used)

RAID3**(3Drives)-** Increase data transfer rate. Performance will be higher. Divide data bit wise. 1 parity is used. If it is lost, we will lose data

RAID4(more than 1 drive) -divide **data block wise**. - data transfer rate slow. Mainly for large write &Read options. 1 parity is used. If it is lost, we will lose data

**RAID 5(minimum 3 disks) (Striping with parity)- parity data shared eventually. reduces potential over use. 1 bit of parity used if one disk fails. Good performance, good redundancy. write option slow. Good read speed**

RAID 6 (Striping with double parity)-2 bits of parity corrects multiple disk failures

RAID 10(4 disks) (Striping + Mirroring)

**RAID1+0(mirror then strip)- minimum 4 disks. excellent redundancy &performance. Expensive.**