EXTENDED MODEL ANSWER

Operating Systems II 2003 Paper 3 Question 5 (SMH)

Basic motivation

Basic motivation is that it's much cheaper to provide a large amount of storage by using lots of average sized disks that by using one massive one (assuming you can even find a disk of the requisite size). Another valid motivation is the desire to allow parallelism in reads and writes, but this is less straightforward to justify.

RAID Level 0

This is simple striping (e.g. write blocks round-robin to the set of N disks, with N generally $= 2$). Benefits include increased parallelism and aggregate throughput. Main drawback is reduced reliability since MTBF(logical device) is now MTBF(physical device) / N.

RAID Level 1

This is simple mirroring (i.e. write blocks to all N disks, read from any individual disk). Benefits include high reliability, and potential for parallelism on read. Drawbacks include the high level of aggregate write bandwith required and of course the cost (we get no additional logical space).

RAID Levels 3, 4 and 5

The first two are simple parity schemes with a fixed parity disk or disks (3 does bit- or byte-interleaving and hence needs costly hardware including spindle sync, 4 does block-level interleaving and hence needs cheaper hardware, or can even by done in software). The problem with both of these is the extra load on the parity disk or disks. RAID5 avoids this by writing data and parity in a round robin fashion so that every disk has some data and some parity. This avoids the bottleneck.

Difficulties in Software RAID 5

There are a number of problems here. For example the read-modify-write issue: when writing less than a full 'stripe', the operating system must first read the existing parity information so that it can compute the new values to write back. This can be avoided by ensuring the file system block size is an integer multiple of the stripe size [less parity], or by using a log-structured filesystem or other deferred-write technique. Another problem is scheduling access to the disks – since reads or writes may require operations on several disks to complete before they are deemed finished, some coordination of layout is useful. An easy solution here involves using the 'same' locations on all disks as part of the same stripe (modulo sector remapping, etc). However since there is no spindle synchronization, even accesses the 'same' locations on multiple disks may take considerably different amounts of time. This can be avoided by e.g. using rotational replicas.

Marks will also be given for any plausible alternatives.