# LAB 3 – Advanced Databases –Kieran Hogan – C12561353

# RAID, B-TREES, Constraints and Triggers

RAID

Q1)

Show how to implement different level of RAID (0,1,3,5,10,0+1) and for the ones that can be implemented compute:

RAID 0: Is implemented by striping blocks on a minimum of 2 disks. Excellent performance, no redundancy.

RAID 1: Is implemented by mirroring blocks on a minimum of 2 disks. Ok performance, excellent redundancy.

RAID 3: Is implemented by striping bytes instead of blocks on a minimum of 3 disks, one of which is a parity disk. Sequential read and right will have good performance, but random read and write will have worst performance. Parity here is good for redundancy, but not as good as mirroring.

RAID 5: Is implemented by striping blocks and distributing the parity over a minimum of 3 disks. Good performance, good redundancy with dist parity. Best cost effective option for both perf and redund. Good for DB that’s heavily read oriented, write ops will be slow.

RAID 10: Is implemented by mirroring blocks, and then striping blocks over a minimum of 4 disks. Excellent redundancy and excellent performance. Best option but more expensive than RAID 5.

RAID 0+1: Is implemented by striping blocks, and then mirroring the blocks over a minimum of 3/4 disks. Same performance as RAID 10, however RAID 0+1 fault tolerance is much less. If a disk in each group of striped fails, the whole RAID will fail. Therefore RAID 10 will always be better than RAID0+1.

RAID 0 can be used here:

1. Storage efficiency here would be 100%. 2 disks can be discarded (the two slow ones), as it would only need 8 disks, 750gb spread over the 8000gb striped.
2. Read performance is very good on RAID 0. Ideally, with the two slow disks discarded, RAID 0 would read at n times the speed of the single disk, 1000mbits per second per disk, which is 8000mbits per second.

RAID 3 can be used here:

1. Storage efficiency here would be good. 8 disks are used for storage, these would be the faster disks, and 1 used for parity. Bytes are striped across the 8 fast disks, 8000gb with 750 used.
2. Read performance is good for sequential reads. 8

RAID 5 can be used here:

1. Storage efficiency here would be close to 100%. Again, not all 10 disks would be needed. 2 can be discarded. Parity distributed across 8 good disks.
2. Read performance is very good on RAID 5, almost as good as RAID 0. It reads at n times the speed of the single disk, 1000mbits per second per disk, which is 8000mbits per second.

3 and can be used but is less popular and less efficient than 0 and 5.

Q2)

A RAID 4 configuration (parity disk at block level) is composed by 5 disks + the parity disk. The time for reading a block is 1 second, writing 5 second. The database needs to modify (write): a) A data spread over 2 blocks in the same stripe

a) READ DATA, PARITY READ, WRITE DATA, WRITE DATA, PARITY WRITE

b) You cannot do the write operation at the same time as doing the read at part a). RAID 4 has bad write performance.

c) You can do a read operation at the same time as doing the read at part a). RAID 4 has great read performance.

d) 1 second, 1 seconds, 5 second, 5 seconds, 5 seconds = 17 seconds

RAID 5

The result is the same since the parity is distributed, as the stripe is only striped on 2 blocks. Having the parity distributed has not changed anything.

B TREES

Q1)

2, 4, 5, 6, 8, 10, 9, 14, 16, 18, 7, 22, 1

a)b)

c)

These two trees look completely different. The main difference is how they are spread out. The standard tree is very one-sided, but is easier to read. The B Tree is however much more balanced, but more difficult to read.