Place holder for 2021/2022 Exam Please change if you want as I'm probably wrong for all of them – I don’t know if I'm tripping but I swear there was already a document for this. I also remember lectures going through some of these questions. (Mandela effect?)

1.a.i) Similar access time to mid ssd but potentially as slow as best HDD performance. Some application that would enjoy equal sequential and random times being the same, maybe an application where writes are random and reads are always sequential or the reverse. OLTP – It's random write heavy?

ii) Write process on SSD degrades the block being written into as unless the block is empty, all the data must be erased on it by a strong charge being passed through the block and then the data is written back with the new write data being added. This strong charge is what damages the block and therefore if writes are not roughly evenly distributed you end up losing blocks faster than you would overwise. Someone else can explain how it works?

Flash translation layer keeps a record of ‘hot’ and ‘cold’ blocks (hot blocks are written to often, cold blocks are written to infrequently) and moves hot data to cold blocks so all blocks are written to reasonably evenly. Means one block isn’t written to very often and so lifetime of the SSD is improved

iii) basically the same as the cw

1.b.i) memory is 2 dimensional. Align data structures for cache, increase fan out, remove child pointers.

ii) Decomposed storage

1.c.i) sequential, no locking, no latching, no need for buffer pool

ii) no, it focuses on throughput over latency? You don’t know how your transaction will get scheduled?

1.d.i) Write once, read infrequently (one or even never), random reads?

ii) Reads on the same piece of data should be minimized, batch jobs?, try to access as much from the same group and not from the same domain?  
I think it should be :  
Reduce data trips to Pelican by smart caching techniques, build software to be able to handle out-of-order data from Pelican, access to data should be hardware-driven to minimize switching of disks.

Worst case workload : full scan of data in Pelican which causes all disk groups to be spun up. Scan done in an order that puts all accesses to the same domain sequentially?

iii) cost to synthesize and limited length of sequence that can be synthesized. Errors in the sequence

2.a) operations on DS are only complete once the slowest node is finished in most cases. You are as fast as the slowest part

2.b) logfile and memtable are super fast and fault tolerant

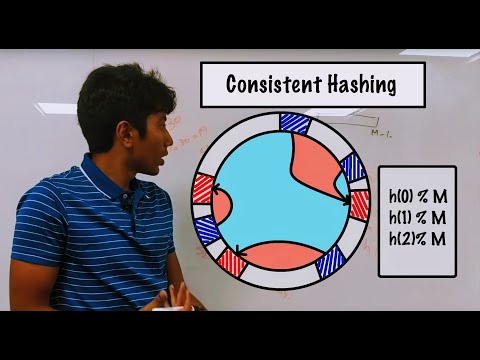
2.c.i) minor compaction

ii) compaction? Merge or major?

and/or?

Bloom filters – allow users to query whether a specified SSTable might contain any data for a row/column pair. Stored in tablet server memory and ensures that lookups for non-existent rows or columns won’t ever touch disk.

2.d) Describe consistent hashing if you wanna get A\*? [What is Consistent Hashing and Where is it used?](https://www.youtube.com/watch?v=zaRkONvyGr8)

[](https://www.youtube.com/watch?v=zaRkONvyGr8)

Just copy this dude he is goated