In the meantime, for some KV stores, such as SILT [24], major efforts are made to optimize reads by minimizing metadata size, while write performance can be compromised without conducting multi-level incremental compactions.” Explain how high write amplifications are produced in SILT.

"Note that LSM-trie uses hash functions to organize its data and accordingly does not support range search.” Do FAWN and LevelDB support range search?

Use Figure 1 to explain the difference between linear and exponential growth patterns.

"Because 4KB block is a disk access unit, it is not necessary to maintain a larger index to determine byte offset of each item in a block.” Show how a lookup with a given key is carried out in LevelDB?

“Instead, we first apply a cryptographic hash function, such as SHA-1, on the key, and then use the hashed key, or hashkey in short, to make the determination.” Assuming a user-provided key has 160 bits, what’s the issue if LSM-trie used the user keys, instead of hashed keys, in its data structure and operations?

“Among all compactions moving data from $L_k$ to $L_{k+1}$, we must make sure their key ranges are not overlapped to keep any two SSTables at Level $L_{k+1}$ from having overlapped key ranges. However, this cannot be achieved with the LevelDB data organization ...” Please explain why LevelDB cannot achieve it?

Use Figures 2 and 3 to describe the LSM-trie’s structure and how compaction is performed in the trie.