PebblesDB: Building Key-Value Stores using Fragmented Log-Structured Merge Trees (II)

NOTE: Your slides/presentation only need to cover background information necessary to answer the given questions (in a clear and well-organized manner). You are allowed to borrow contents from other resources, such as online slides, as long as you acknowledge them. The presentation should be mostly question-focused and proceed mostly in a Q&A format. Please include the questions in your slides. Don’t write detailed answers in the slides and read them to the class. Instead, use bullet points, graphs, or animations to explain your answers to the class.

In your Q&A report, use text to more thoroughly answer the questions. Include a short paragraph at the beginning of the report to summarize the paper.

1. “FLSM can be viewed as a generalization of the LSM data structure.” Please explain this statement.

2. “New sstables are simply added to the correct guard in the next level. There are two exceptions to the no-rewrite rule.” Describe these two exception compaction scenarios.

3. “Deleting a guard will involve a significant amount of compaction work” Please describe the operations involved in a guard deletion.

4. Describe how a range query is served.

5. “In Workload F, all writes are read-modify-writes: the workload does a get() before every put() operation. As a result, the full write throughput of PebblesDb is not utilized, resulting in performance similar to that of other key-value stores.” Explain why for this workload PebbleDB has similar performance.