SkimpyStash: RAM Space Skimpy Key-Value Store on Flash-based Storage

NOTE: Your slides/presentation need to cover the assigned sections and 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. For a slide that covers a given question, please print the question on the slide where it is covered. However, you don’t have to answer the question using a long paragraph of text on the slide. Instead, use bullet points, graph, animation, or oral explanation to answer the question. In your Q&A report, use text to more thoroughly answer the questions.

Only need to cover sections before Section 4.6

1. “Our base design uses less than 1 byte in RAM per key-value pair and our enhanced design takes slightly more than 1 byte per key-value pair.” In FAWN, even a pointer to a KV pair needs a 4-byte pointer. How can possibly SkimpyStash achieve such a low memory cost for metadata?

2. “SkimpyStash uses a hash table directory in RAM to index key-value pairs stored in a log-structure on flash.” Why are key-value pairs on the flash organized as a log?

3. “The average bucket size is the critical design parameter that serves as a powerful knob for making a continuum of tradeoffs between low RAM usage and low lookup latencies.” Please explain this statement.

4. “The client [write] call returns only after the write buffer is flushed to flash.” Why cannot such a call be acknowledged earlier?

5. “Basic functions: Store, Lookup, Delete” Use Figure 2(a) to explain how these basic functions are executed?

6. “The chain of records on flash pointed to by each slot comprises the bucket of records corresponding to this slot in the HT directory. This is illustrated in Figure 3.”. Please use the figure to describe SkimpyStash’s data structure. Also explain how lookup, insert, and delete operations are executed.

7. “Because we store the chain of key-value pairs in each bucket on flash, we incur multiple flash reads upon lookup of a key in the store.” Please explain how this issue can be alleviated. [Hint: please refer to Section “Compaction to Reduce Flash Reads during”]

8. “two-choice based load balancing strategy is used to reduce variations in the number of keys assigned to each bucket”. Explain how this is achieved.

9. “... when the last record in a bucket chain is encountered in the log during garbage collection, all valid records in that chain are compacted and relocated to the tail of the log.”. Please explain how garbage is collected.