**Week 4: LSM Trees**

**Overview**

According to the RUM conjecture, data structures can be optimized for at most two out of the three properties:

* Read overhead: the amount of additional data that needs to be read in order to read a record of interest.  For example, a database needs to load a search index to identify the record number and then load the record of interest.  Using a tree data structure, it make take O(log N) disk accesses to find the record number and an additional O(1) disk access to grab the record itself.
* Update overhead: the amount of additional data that needs to be written in order to update a record of interest. Similar to the read overhead, the database may need to search a tree to find the record of interest, perform a read of the record, and then overwrite the record on disk with the updated data.
* Memory overhead: the amount of additional memory required to store a data set in the data structure.  In the case of a tree, each leaf stores pointers to other nodes.

B-trees are generally considered to be optimized for read performance.  A naive B-tree implementation can be problematic if a user is inserting a larger number of sequential elements.  Variants of B-trees have been developed that optimize for particular situations.  Entirely different data structures like Log-Structured Merge (LSM) trees have been developed for write-optimized use cases and are used in databases like RocksDB and Apache Cassandra.  In this module, we're going to learn about LSM trees and compare their use cases those of B-trees.  Knowing the type of search data structures a database uses can tell you a lot about its appropriate use cases.

**Reflection Questions**

* How are data are inserted, searched, and stored with LSM trees?
* What are big-O time complexities of these operations?
* What are the relative advantages and disadvantages of naive B-trees and LSM trees?
* Why is a B-tree and LSM trees considered to be read- and write-optimized data structures, respectively?
* What does the RUM conjecture state? What are the implications?
* Where do B-trees and LSM trees fall in the read-update-memory space? Can you explain why?