# Couchbase questions

1. Even though Couchbase is considerably more flexible with its Bucket abstraction (i.e. SQL table), how should my domain entities be represented? Should I lean more towards representing higher level flattened entities as buckets or should the abstraction level be lower with key-based references (like relational DBs)?
2. I ran into major performance issues attempting to ensure the state of buckets prior to specific test cases (I do Test Driven Development). Using flush() takes prohibitively longer (the runtime of many test cases went from less than a second to 10-15 seconds. Having over 300 automated test cases so far means that I could go out to lunch😊 I also received intermittent “indexer” errors after flushes. Any ideas for this situation would be appreciated. Using Mock objects for the DB is unwarranted as it is the database I am partially testing.
3. Recently, I ran into an overall performance problem. My 320 test cases run time went from 55 seconds to over 4 minutes. I ended up re-installing your DB, which did not fix the problem. Neither did doing compactions. What finally worked was to drop and recreate the buckets. I would love to understand what I’m doing wrong that is causing this…
4. I have multiple automated test cases that involve the logging of data followed by the checking of that data and its relationship to other data. Unfortunately, I have not found a way to synchronously log and ensure the writes have completed before moving to the next portion of the test case. That means these test cases are non-deterministic and are successful only 95% of the time by inserting short “sleep” intervals between the logging and the query calls. Additionally, that type of logic also adds potentially unnecessary seconds to the overall test suite execution time. Any help with this is appreciated.
5. Do you have any recommendations on my configuration parameters below? They work except for the occasional “connect timeout” error:

continuousKeepAliveEnabled=false

keyValueServiceConfig=2

connectTimeout=10000

socketConnectTimeout=2000

disconnectTimeout=2000

kvTimeout=2000

keepAliveTimeout=2000

I am currently using the singleton design pattern to ensure that the couchbase connection is only done once for all test cases. Therefore, I do not suspect the connection and bucket opening logic to be a source of performance issues. For production code this will be changed to dependency injection of the couchbase connection.

Thanks in advance for your assistance with any of these issues/questions.

Larry Avery

210-724-5531