

Cloud Data Bases

Milestone 2: Persistent Storage Server

PeristenceTest:

- **testCreateFile()**: Test create a file named "SomeKey=09" on the file system and assert that it is existed
- **testReadFile()**: test reading a file named "SomeKey=09" and assert that its value is "=="Abc09\$8".
- **testUpdateFile()**: try to update a file named "SomeKey=09" with a new value "New\"Val=10" and assert that the update is success file. Then reading that file again and assert that the file content has been updated.
- **testDeleteFile()**: try to delete a file named "SomeKey=09" and assert that the deletion was successful.

CacheTest:

getUpdateCacheMethod():Get private method for updating the cache from the {@link CacheManager}

* @param cm CacheManager

* @return {@link CacheManager#updateCache(K, V)}

* @throws NoSuchMethodException something was changed in the implementation and method doesn't exist anymore

testCacheStrategySimplePut(): Test simple put operation on cache without exceeding the capacity

* @param capacity capacity of the cache to test

* @param strategy displacemetn strategy to test

* @throws NoSuchMethodException for some reaqson {@link CacheManager#updateCache(K, V)} was not found

* @throws InterruptedException if {@link TestClient} thread was interrupted

testFIFOsimplePut(): Test simple put for {@link server.storage.cache.FIFO} strategy
{@link #testCacheStrategySimplePut(int, CacheDisplacementStrategy)}

testLRUSimplePut(): Test simple put for {@link server.storage.cache.LRU} strategy
{@link #testCacheStrategySimplePut(int, CacheDisplacementStrategy)}

testLFUSimplePut(): Test simple put for {@link server.storage.cache.LFU} strategy
{@link #testCacheStrategySimplePut(int, CacheDisplacementStrategy)}

testCacheOneItemEvict(): Test one of the edge cases when there is a cache of size one, check if eviction works

* @param strategy {@link CacheDisplacementStrategy}

testFIFOoneItemEvict(): Eviction test {@link
#testCacheOneItemEvict(CacheDisplacementStrategy)} for {@link
server.storage.cache.FIFO} strategy

testLRUoneItemEvict(): Eviction test {@link
#testCacheOneItemEvict(CacheDisplacementStrategy)} for {@link
server.storage.cache.LRU} strategy

testLFUoneItemEvict(): Eviction test {@link
#testCacheOneItemEvict(CacheDisplacementStrategy)} for {@link
server.storage.cache.LFU} strategy

testDelete(): Test removing an item from the cache when value of put operations is null

* @param cm {@link CacheManager} to test

testFIFODelete(): Cache item remove operation for cache {@link
#testDelete(CacheManager)} with {@link server.storage.cache.FIFO} strategy

testLRUDelete(): Cache item remove operation for cache {@link
#testDelete(CacheManager)} with {@link server.storage.cache.LRU} strategy

testLFUDelete(): Cache item remove operation for cache {@link
#testDelete(CacheManager)} with {@link server.storage.cache.LFU} strategy