

# RocksDB - Atomic Flush

Atomic Flush là một tính năng trong RocksDB cho phép bạn đảm bảo rằng tất cả các thay đổi dữ liệu đã được ghi vào bộ nhớ đệm sẽ được ghi xuống đĩa cùng một lúc. Điều này đảm bảo tính nhất quán của dữ liệu và tránh mất mát dữ liệu do quá trình ghi bị hỏng.

Trong RocksDB, bạn có thể kích hoạt Atomic Flush bằng cách sử dụng `SetAtomicFlush()` trong `Options`.

Dưới đây là một ví dụ về việc sử dụng Atomic Flush trong RocksDB bằng Java:

```
import org.rocksdb.*;

public class RocksDBAtomicFlushExample {

    public static void main(String[] args) {
        RocksDB.loadLibrary();
        try (final Options options = new Options().setCreateIfMissing(true)) {
            options.setAtomicFlush(true);
            try (final RocksDB db = RocksDB.open(options, "/path/to/database")) {
                // Thực hiện các thao tác ghi dữ liệu
                byte[] key = "myKey".getBytes();
                byte[] value = "myValue".getBytes();
                db.put(key, value);
                // Kích hoạt Atomic Flush
                db.flush(new FlushOptions().setWaitForFlush(true));
            } catch (RocksDBException e) {
                // Xử lý lỗi khi mở database không thành công
            }
        }
    }
}
```

Trong ví dụ trên, chúng ta sử dụng `setAtomicFlush(true)` để kích hoạt Atomic Flush trong RocksDB. Sau đó, chúng ta thực hiện các thao tác ghi dữ liệu và sử dụng `flush()` để ghi xuống đĩa. Bằng cách sử dụng `setWaitForFlush(true)`, RocksDB đảm bảo rằng tất cả các thay đổi đã được ghi xuống đĩa trước khi tiếp tục thực thi chương trình.

Đây là một ví dụ về việc sử dụng Atomic Flush trong RocksDB bằng Java.

## C++:

```
#include "rocksdb/db.h"

int main() {
    rocksdb::DB* db;
    rocksdb::Options options;
    options.create_if_missing = true;
    options.atomic_flush = true;
    rocksdb::Status status = rocksdb::DB::Open(options, "/path/to/database", &db);
    if (status.ok()) {
        // Thực hiện các thao tác ghi dữ liệu
        rocksdb::WriteOptions writeOptions;
        rocksdb::Slice key = "myKey";
        rocksdb::Slice value = "myValue";
        rocksdb::Status putStatus = db->Put(writeOptions, key, value);
        if (putStatus.ok()) {
            // Kích hoạt Atomic Flush
            rocksdb::FlushOptions flushOptions;
            flushOptions.wait = true;
            rocksdb::Status flushStatus = db->Flush(flushOptions);
            if (!flushStatus.ok()) {
                // Xảy lỗi khi ghi xuống đĩa không thành công
            }
        } else {
            // Xảy lỗi khi ghi dữ liệu không thành công
        }
    } else {
        // Xảy lỗi khi mở cơ sở dữ liệu không thành công
    }
    delete db;
    return
```

---

Revision #1

Created 23 September 2023 10:22:58 by Laptrinh.vn

Updated 23 September 2023 10:56:18 by Laptrinh.vn