

RocksDB - Write batch

`WriteBatch` trong RocksDB được sử dụng để thực hiện nhiều thao tác ghi (put, delete, merge) cùng một lúc, giúp tăng hiệu suất ghi dữ liệu. Điều này có thể đặc biệt hữu ích khi bạn muốn thực hiện nhiều thao tác ghi một cách liên tục mà không muốn mỗi thao tác phải chờ cho đến khi dữ liệu được ghi vào ổ đĩa trước khi thực hiện thao tác tiếp theo.

Dưới đây là một ví dụ về cách sử dụng `WriteBatch` trong RocksDB bằng C++:

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

int main() {
    rocksdb::DB* db;
    rocksdb::Options options;
    options.create_if_missing = true;
    rocksdb::Status status = rocksdb::DB::Open(options, "/path/to/database", &db);
    if (status.ok()) {
        rocksdb::WriteBatch batch;
        batch.Put("key1", "value1");
        batch.Delete("key2");
        batch.Merge("key3", "value3");
        rocksdb::WriteOptions writeOptions;
        rocksdb::Status write_status = db->Write(writeOptions, &batch);
        if (write_status.ok()) {
            // Thực hiện thành công
        } else {
            // Xử lý lỗi khi thực hiện ghi không thành công
        }
    } else {
        // Xử lý lỗi khi mở database không thành công
    }
    delete db;
    return 0;
}
```

Java:

```

import org.rocksdb.*;

public class RocksDBWriteBatchExample {
    public static void main(String[] args) {
        RocksDB.loadLibrary();
        try (final Options options = new Options().setCreateIfMissing(true);
            final RocksDB db = RocksDB.open(options, "/path/to/database")) {
            try (final WriteBatch batch = new WriteBatch()) {
                byte[] key1 = "key1".getBytes();
                byte[] value1 = "value1".getBytes();
                batch.put(key1, value1);

                byte[] key2 = "key2".getBytes();
                batch.delete(key2);

                byte[] key3 = "key3".getBytes();
                byte[] value3 = "value3".getBytes();
                batch.merge(key3, value3);

                WriteOptions writeOptions = new WriteOptions();
                db.write(writeOptions, batch);
            } catch (RocksDBException e) {
                // Xử lý lỗi khi thực hiện ghi không thành công
            }
            } catch (RocksDBException e) {
                // Xử lý lỗi khi mở database không thành công
            }
        }
    }
}

```

Trong ví dụ trên, chúng ta tạo một RocksDB database và sau đó sử dụng `WriteBatch` để thực hiện các thao tác ghi (put, delete, merge) trên các key tương ứng. Sau khi thêm các thao tác ghi vào `WriteBatch`, chúng ta sử dụng phương thức `write` của RocksDB để thực hiện ghi tất cả các thao tác trong `WriteBatch` cùng một lúc.

Lưu ý rằng `WriteBatch` là một thao tác ghi đồng bộ trong RocksDB. Điều này có nghĩa là thao tác `write` sẽ chờ cho đến khi tất cả các thao tác ghi trong `WriteBatch` được ghi vào ổ đĩa trước khi trả về.

Đó là một số thông tin cơ bản về cách sử dụng `WriteBatch` trong RocksDB. Bạn có thể tìm hiểu thêm về các tùy chọn và tính năng khác của RocksDB trong tài liệu chi tiết của nó.

Revision #1

Created 23 September 2023 09:25:05 by Laptrinh.vn

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