# 基本原理

rocksdb是基于leveldb1.5进行修改的，基本原理是基于leveldb，以下的介绍的内容是根据leveldb的资料进行整理和总结。

## leveldb简介

1、leveldb是kv系统

2、leveldb具有持久化功能

3、数据分层

4、每一层key是排序

## 架构

外存

level0

level1

levelN

.sst文件

Immutable

Memtable

Memtable

内存

数据在leveldb中的存储架构，数据首先写入memtable中，当memtable达到阈值后，变为不可写的Immutable memtable并写入level0的磁盘文件中。

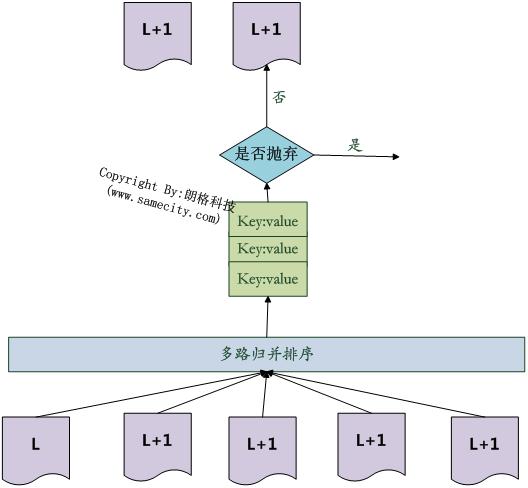
memtable和Immutable memtable是相同结构的，以skip list的方式对key进行排序。

level0的sst文件是Immutable memtable的内容保存至外存

## Compaction

compaction分为三种，minor、major和full。minor是将Immutable memtable写入外存的；major是外存中的sst文件中数据的合并；full是所有外存文件的合并。

如下图是major合并图示



除了level0层以外，其他层次的sst文件中，不存在key重叠的情况，也就是第N个sst文件的key，都小于N+1个sst文件的key

越频繁的compaction会形成越多的写操作

## 读取

依次在memtable、immutable memtable、level 0到level N中进行key的搜索

参考：

http://www.samecity.com/blog/Index.asp?SortID=12

# rocksdb

## 编译

1、依赖组件

zlib、bzip2、snappy、gflags

2、GCC版本

支持C++11特性

3、生成目标代码

编译静态库：make static\_lib

编译动态库：make shared\_lib

参考

https://github.com/facebook/rocksdb/blob/master/INSTALL.md

## mysql引擎

https://github.com/MySQLOnRocksDB/mysql-5.6/

9个月前最后更新