1. Log(options\_.info\_log, "Level-0 table #%llu: started",

(unsigned long long) meta.number);

这个日志是由Status DBImpl::WriteLevel0Table(MemTable\* mem, VersionEdit\* edit,Version\* base)生成，主要是在由immutable进行compaction到level-0时的日志输出。标志着minor compaction的开始。

1. Log(options\_.info\_log, "Level-0 table #%llu: %lld bytes %s",

(unsigned long long) meta.number,

(unsigned long long) meta.file\_size,

s.ToString().c\_str());

这个日志是由Status DBImpl::WriteLevel0Table(MemTable\* mem, VersionEdit\* edit,Version\* base)生成,紧跟在上一个日志之后。用来说明刚才的minor compaction的状态。

1. Log(options\_->info\_log,

"Expanding@%d %d+%d (%ld+%ld bytes) to %d+%d (%ld+%ld bytes)\n",

level,

int(c->inputs\_[0].size()),

int(c->inputs\_[1].size()),

long(inputs0\_size), long(inputs1\_size),

int(expanded0.size()),

int(expanded1.size()),

long(expanded0\_size), long(inputs1\_size));

这个日志出现在void VersionSet::SetupOtherInputs(Compaction\* c)中，这个函数主要是设置Inputs\_[1]中的内容，就是填充将要参加合并的level+1和level中有覆盖部分的sstable到inputs\_[1]中。

1. Log(options\_.info\_log, "Compacting %d@%d + %d@%d files",

compact->compaction->num\_input\_files(0),

compact->compaction->level(),

compact->compaction->num\_input\_files(1),

compact->compaction->level() + 1);

这个日志在Status DBImpl::DoCompactionWork(CompactionState\* compact)中调用，代表真正合并的开始。

1. Log(options\_.info\_log,

"Generated table #%llu@%d: %lld keys, %lld bytes",

(unsigned long long) output\_number,//生成的sstable文件编号

compact->compaction->level(),//合并的level,其实是将新的sstable放到level+1中

(unsigned long long) current\_entries,//sstable中的记录数

(unsigned long long) current\_bytes);//sstabl占据的大小

这个日志是在FinishCompactionOutputFile(CompactionState\* compact,Iterator\* input)中，是在将合并生成的新sstable落盘时写的内容。

1. Log(options\_.info\_log, "Compacted %d@%d + %d@%d files => %lld bytes",

compact->compaction->num\_input\_files(0),//level中要删除文件的总数

compact->compaction->level(),//level的层数

compact->compaction->num\_input\_files(1),//level+1中要删除文件的总数

compact->compaction->level() + 1,//level+1的层数

*static\_cast*<long long>(compact->total\_bytes));//要删除的总大小(inputs\_[2]数组的总大小)

这个日志由Status DBImpl::InstallCompactionResults(CompactionState\* compact)生成，主要是用来纪录哪些level要合并，以及每一个Level的文件大小。代表真正的合并基本完成。

1. Log(options\_.info\_log,

"compacted to: %s", versions\_->LevelSummary(&tmp));

这个日志是由Status DBImpl::DoCompactionWork(CompactionState\* compact)中生成，代表真正的合并已经完成了。它会打印当前version的每一层sstable文件的数量。如compacted to: files[ 0 9 1 0 0 0 0 ]。这里正好有七层，代表着每一层有多少个sstable文件。

1. Log(options\_.info\_log, "Delete type=%d #%lld\n",

int(type),//要删除的文件类型

*static\_cast*<unsigned long long>(number));//要删除的文件编号

这个日志是由void DBImpl::DeleteObsoleteFiles()生成的，而这个函数又是在执行完DoCompactionWork()之后做的垃圾回收工作，这个日志的输出是将type类型的文件删除掉。而type的值就是

*enum* FileType {

*kLogFile*,

*kDBLockFile*,

*kTableFile*,

*kDescriptorFile*,//这个标志就是MANIFEST文件

*kCurrentFile*,

*kTempFile*,

*kInfoLogFile* *//* *Either* *the* *current* *one,* *or* *an* *old* *one*

};

分别从0到6个数来表示不同的文件类型。

}