

VLDB实验文档

下载代码

```
1 git clone
  https://github.com/zhanglingzhe0820/LSMCodeAndBenchmarkForVLDB.git
2 cd LSMCodeAndBenchmarkForVLDB
```

目录结构解析

本实验全部基于Apache IoTDB 0.12版本进行

iotdb-add_lsm_compaction: 基于0.12版本加入了顺乱序不分离的lsm合并

iotdb-add_tlsm_compaction: 基于0.12版本修改了顺乱序分离的tlsm合并, 更方便本论文实验

benchmark: 数据生成脚本、写入脚本、测试脚本等所在目录

benchmark脚本解析

benchmark中的脚本主要分为以下5个部分

数据生成

ty.txt: 模拟工厂数据的分布规律

data_prepare_iotdb_ty.py: 根据ty.txt生成符合相应要求的数据 (生成.csv文件, 之后会在写入部分使用)

```
1 python data_prepare_iotdb_ty.py ${生成点数} ${乱序时间间隔}
```

generate_data.sh: 使用data_prepare_iotdb_ty.py一键生成符合论文中实验条件的数据

iotdb实验配置

lsm_write.properties: lsm写入时的iotdb参数配置

lsm_compaction.properties: lsm合并时的iotdb参数配置

tlsm_write_1000.properties: tlsm按seq_memtable_size=1000写入时的iotdb参数配置

tlsm_compaction_1000.properties: tlsm按seq_memtable_size=1000合并时的iotdb参数配置

tlsm_write_2000.properties: tlsm按seq_memtable_size=2000写入时的iotdb参数配置

tlsm_compaction_2000.properties: tlsm按seq_memtable_size=2000合并时的iotdb参数配置

tlsm_write_3000.properties: tlsm按seq_memtable_size=3000写入时的iotdb参数配置

tlsm_compaction_3000.properties: tlsm按seq_memtable_size=3000合并时的iotdb参数配置

tlsm_write_4000.properties: tlsm按seq_memtable_size=4000写入时的iotdb参数配置

tlsm_compaction_4000.properties: tlsm按seq_memtable_size=4000合并时的iotdb参数配置

写入

write_iotdb-1.0-SNAPSHOT.jar: iotdb写入数据脚本

```
1 java -jar write_iotdb-1.0-SNAPSHOT.jar 127.0.0.1 6667 root.storage_group  
   ${需要写入的.csv文件}
```

run_lsm.sh: lsm写入数据调度脚本

```
1 ./run_lsm.sh ${需要写入的.csv文件} ${生成点数} ${乱序时间间隔}
```

run_lsm_compaction: lsm合并调度脚本

```
1 ./run_lsm_compaction.sh ${生成点数} ${乱序时间间隔}
```

run_tlsm.sh: tlsm写入数据调度脚本

```
1 ./run_tlsm.sh ${需要写入的.csv文件} ${生成点数} ${乱序时间间隔}
```

run_tlsm_compaction: tlsm合并调度脚本

```
1 ./run_tlsm_compaction.sh ${生成点数} ${乱序时间间隔}
```

查询

query文件夹: 查询所需的所有sql语句

read_iotdb-1.0-SNAPSHOT.jar: iotdb查询数据脚本

```
1 java -jar read_iotdb-1.0-SNAPSHOT.jar ${需要查询的sql语句文件(即query文件夹中  
   的任意一个文件)}
```

query_lsm.sh: lsm查询调度脚本

```
1 ./query_lsm.sh ${生成点数} ${乱序时间间隔}
```

query_tlsm.sh: tlsm查询调度脚本

```
1 ./query_tlsm.sh ${生成点数} ${乱序时间间隔}
```

整体调度

stop_server.sh: 终止所有实验(包括iotdb进程)的脚本

run_test.sh: 实验一键调度脚本(除了数据生成部分)

论文实验启动流程

编译Apache IoTDB

```
1 cd iotdb-add_tlsm_compaction  
2 mvn clean package -DskipTests //等待编译完成  
3 cd ../iotdb-add_lsm_compaction  
4 mvn clean package -DskipTests //等待编译完成
```

生成数据

```
1 cd ../benchmark  
2 ./generate_data.sh //等待数据生成完成
```

运行测试

```
1 nohup ./run_test.sh > data.log &
```

等全部测试运行完后，根据data.log文件统计写放大和查询性能