## 调研SQLite本地数据库的读写速度，评估是否可以用于信号数据存储

目标：主要测试Real单精度浮点型和Blob二进制流两种数据类型存储的读写性能

以下有4种方案：

1. 环境：Windows本地；

存储数据类型：Real浮点型；

数据列:1个数据列，无索引

参数：num\_tables(通道数量), 线程池ThreadPool，num\_records（插入数据数量），线程池线程数量ThreadNum, frequency（每一事务提交执行语句的数量，也即采样频率）；

测试任务：开启事务，使用批量插入，同时向1024通道插入数据1000条数据，即1024个通道都要同时插入1000条数据，采样频率1KHZ；

测试指标：每一秒能存多少数据点，耗时，数据库文件大小。

测试结果：

总共用时为5.159s,每一秒能存197632个数据点(1024个通道累计)，平均每个通道一秒存储193个数据点，生成数据库文件大小20.1MB。

1. 环境：Windows本地；

存储数据类型：Blob二进制流；

数据列:1个数据列，无索引

参数：num\_tables(通道数量), 线程池ThreadPool，num\_records（插入数据数量），线程池线程数量ThreadNum, frequency（每一事务提交执行语句的数量，也即采样频率）；

测试任务：开启事务，使用批量插入，同时向1024通道插入数据1000条数据，即1024个通道都要同时插入1000条数据，采样频率1KHZ；

测试指标：每一秒能存多少数据点，耗时，数据库文件大小。

测试结果：

总共用时为4.162s,每一秒能存245760个数据点(1024个通道累计)，平均每个通道一秒存储240个数据点，生成数据库文件大小16.1MB。

1. 环境：Linux本地虚拟机；

存储数据类型：Real浮点型；

数据列:1个数据列，无索引

参数：num\_tables(通道数量), 线程池ThreadPool，num\_records（插入数据数量），线程池线程数量ThreadNum, frequency（每一事务提交执行语句的数量，也即采样频率）；

测试任务：开启事务，使用批量插入，同时向1024通道插入数据1000条数据，即1024个通道都要同时插入1000条数据，采样频率1KHZ；

测试指标：每一秒能存多少数据点，耗时，数据库文件大小。

测试结果：

总共用时为3.603s,每一秒能存284207个数据点(1024个通道累计)，平均每个通道一秒存储277个数据点，生成数据库文件大小21MB。

1. 环境：Linux本地虚拟机；

存储数据类型：Blob二进制流；

数据列:1个数据列，无索引

参数：num\_tables(通道数量), 线程池ThreadPool，num\_records（插入数据数量），线程池线程数量ThreadNum, frequency（每一事务提交执行语句的数量，也即采样频率）；

测试任务：开启事务，使用批量插入，同时向1024通道插入数据1000条数据，即1024个通道都要同时插入1000条数据，采样频率1KHZ；

测试指标：每一秒能存多少数据点，耗时，数据库文件大小。

测试结果：

总共用时为3.151s,每一秒能存324963个数据点(1024个通道累计)，平均每个通道一秒存储317个数据点，生成数据库文件大小17MB。

Windows环境测试评估结果：在参数集合为：{ num\_tables =1024, ThreadPool=True,ThreadNum=4, frequency =1KHZ, records =1000},使用Blob二进制流存储浮点型数据（方案2）比存储Real类型（方案1）快接近1s,文件存储大小也比方案1文件小4MB，方案2的每秒存储数据点数多出方案1近48127个数据点。

Linux环境测试评估结果：在参数集合为：{ num\_tables =1024, ThreadPool=True,ThreadNum=4, frequency =1KHZ, records =1000},使用Blob二进制流存储浮点型数据（方案4）比存储Real类型（方案3）快接近0.5s,文件存储大小也比方案3文件小4MB，方案4的每秒存储数据点数多出方案3近40756个数据点。