# 一 需求分析

## 1.1 核心功能(生产中用的较多的功能)

### 1.1.1 数据导入测试

通过offline模式生成数据，计划手动导入数据或者利用python脚本进行文件各个数据库的文件导入，判断各个数据库的导入性能

用java执行python脚本进行代码实现

### 1.1.2 数据压缩比测试

根据配置文件，读取数据导入后的文件路径，判断前后数据导入的大小差别，进行压缩比计算，导入前的文件大小为数据生成器的数据文件大小，导入后的文件问数据库文件大小

1.1.4 负载写入(append)功能测试

(有历史数据，并且同时又写入和读取的负载，进行输入写入)

测试设备数，选择单设备，多设备(可配置，默认10个)

以单设备为例（两个测试结果，1，lines/s 平均每秒的写入数目 2，timeout 延迟，平均插入每k条的延迟毫秒）

十个模式下分别插入1,100,1000,5000,10000,30000，100000 条各100次，计算avg,max,min,95%,99%

1.1.5负载读取测试功能测试

(有基本历史数据，并且同时有写入和读取的负载)，进行聚合读取测试(对设备，测试点进行聚合(某个时间段的avg,min,max,sum))

先插入前一周的数据;

查询限制为单设备

十个模式下分别对1min,1Hour,1day,1week数据全字段进行查询100次，计算出 min,max,avg,95%,99%;

测试指标为

1. 延迟时间: ms /1000Line（每1000行延迟时间）；
2. 查询速度:lines/s(每秒的查询函数)

导出结果时候带上测试条件

1.1.6负载数据(ms->sec->min->hour)的聚合操作

(有基本历史数据，并且同时有写入和读取的负载)

先插入前一周的数据

查询限制为单设备，同一属性测试聚合条件 ，聚合查询完就一条记录

分别是十个模式下进行计算

1,1分钟内对设备进行max,min,avg,sum聚合查询各100次，聚合字段为float值字段，分组为sec，设备，设备属性。

2,1小时内对设备进行max,min,avg,sum聚合查询各100次，聚合字段为float类型字段，分组为minute，设备，设备属性。

3,1day内对设备进行max,min,avg,sum聚合查询各100次，聚合字段为float类型字段，分组为hour，设备，设备属性。

3,1week内对设备进行max,min,avg,sum聚合查询各100次，聚合字段为float类型字段，分组为day，设备，设备属性。  
测试指标为延迟时间: ms (对应的时间段，聚合条件,where条件)

## 1.2 辅助功能(生产中用的较少的功能）

1.2.1数据update测试

十种负载模式下，online插入数据，0.1几率错掉一个字段的数据，后面update计算延迟时间

1.2.2数据random insert

插入数据时候，随机漏掉数据并且记录，在后面进行批量random insert 操作，计算insert延迟时间，每秒的insert数目

十种模式下，online插入数据，0.1几率漏掉数据，后面再插入漏掉的数据

1.2.3 删除某个时间之前的数据

首先插入1周内的数据

删除昨天同时间之前的数据所用的延迟ms,已经平均每sec的所删除的数目

删除所有数据

进行100次操作，计算出min,max,avg,95%,99%

性能指标

1,延迟时间:ms(每1000条)

2，每秒的行数 lines/s

1.2.4 数据导出

首先判断目标数据库是否支持导出功能，然后测试导出性能

如果支持，先导入1周数据，然后导出1周数据，100次，计算延迟时间

# 二 负载模式定义

**模式一**

没有任何历史数据，没有任何负载

**模式二**

测试设备没有历史数据，没有任何负载

**模式三**

测试设备有历史数据，没有任何负载

**模式四**

测试设备有历史数据，负载为同设备每秒插入10条数据，聚合查询10次(聚合为max,以分钟聚合)

**模式五**

测试设备有历史数据，负载为同设备每秒插入10条数据，异设备每秒插入10条数据，聚合查询10次(聚合为max,以分钟聚合)

**模式六**

测试设备有历史数据，负载为同设备每秒插入10条数据，异设备每秒插入100条数据，聚合查询10次(聚合为max,以分钟聚合)

**模式七**

测试设备有历史数据，负载为同设备每秒插入100条数据，异设备每秒插入100条数据，聚合查询100次(聚合为max,以分钟聚合)

**模式八**

测试设备有历史数据，负载为同设备每秒插入100条数据，异设备每秒插入100条数据，聚合查询100次(聚合为max,以分钟聚合) PS：这个模式不太科学，实现过程中测试一下

**模式九**

测试设备有历史数据，负载为同设备每秒插入100条数据，异设备每秒插入1000条数据，聚合查询100次(聚合为max,以分钟聚合) PS：这个模式不太科学，实现过程中测试一下

**模式十**

测试设备有历史数据，负载为同设备每秒插入100条数据，异设备每秒插入1000条数据，聚合查询100次(聚合为max,以分钟聚合) PS：这个模式不太科学，实现过程中测试一下

# 三 配置文件说明

**基本函数设置（base-function.conf）**

* 1. 配置各个基本函数参数，基本函数为系统内置，这里可以配置各个函数的参数，各个函数的类型可以重复，但是各个函数的id不可以重复，这个配置文件中的函数id，用于其他配置文件中的引用
  2. 数据类型为数字型(整型，浮点型)的基本函数(均为周期函数)包含正弦，单调斜率唯一函数(包含(波和锯齿波)，抛物线，指数，对数

数字类型为非数字型(枚举类型，字符串类型，bool类型) 均显示配置在文件中

* 1. 每个函数均有自己的id，后面所有用到的函数，均需要在这里配置，需要时引用这里的id

应用场景:用于offline数据生成，和online数据负载的函数引用

**传感器值配置(sensor.conf)**

配置各个传感器的基本函数，以及名称

传感器波形可引用base-function.conf中的各个函数id进行函数组合，设置好连续点可自由组合函数生成数据(必须是相同数据类型，如果所配置的连续点不存在，则按照该不连续的点之前的函数生成数据)

应用场景:用于设备配置引用

**设备配置(mertric.conf)**

配置各个设备的参数 包含设备的元数据信息(位置，各个传感器的位置及引用)，设备的有效时间，对应各个传感器的有效时间

Ps:配置时候，数据结构要满足各个数据库

**offline 数据生成(generate-offline.conf)**(多线程生成，一个设备最多支持一个线程，一个设备对应一个目标文件)

配置目标数据库类型，可配置多个设备，配置每个设备的采集时间段，采集的步长，设备来自mertric.conf，最后根据设备及目标数据库类型生成相应数据库的目标导入文件

应用场景:Offline数据导入，主要针对相应的数据库import，向数据库种导入数据

**online 数据负载(generate-online.conf)**(多线程负载)

配置十种模式中，各个模式所对应的设备，设备的各个参数(步长，缓存时间等)

**数据库连接配置(database.conf)**

数据库类型，url，端口，账号，密码信息