|  |  |
| --- | --- |
| **工业时序大数据评测基准** | **V0.7** |
| **需求规格说明书** | **2017.05.07** |

北京市中国人民大学

**版本历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 日期 | 作者 | 变更操作 | 说明 |
| 0.7 | 2017.05.07 | 覃雄派 | 开始撰写 | 1，场景与数据。2，数据导入。3，性能测试。 |
|  |  |  |  |  |

目录

[1. 引言 4](#_Toc482031242)

[1.1. 系统名称 4](#_Toc482031243)

[1.2. 系统描述 4](#_Toc482031244)

[1.3. 本系统使用者 4](#_Toc482031245)

[1.4. 本文读者 4](#_Toc482031246)

[1.4.1. 系统需求提出者 4](#_Toc482031247)

[1.4.2. 系统设计者 4](#_Toc482031248)

[1.4.3. 系统开发者 4](#_Toc482031249)

[1.5. 参考资料 5](#_Toc482031250)

[2. 名词解释和对象描述 6](#_Toc482031251)

[2.1. 设备 6](#_Toc482031252)

[2.2. 测点 6](#_Toc482031253)

[2.3. 数据点 6](#_Toc482031254)

[2.4. 数据生成器 6](#_Toc482031255)

[2.5. 时序数据库 6](#_Toc482031256)

[2.6. 负载生成器 6](#_Toc482031257)

[2.7. 性能评测模块 7](#_Toc482031258)

[3. 工业时序大数据的实际应用场景和数据生成 8](#_Toc482031259)

[3.1. 工业时序大数据的实际应用场景 8](#_Toc482031260)

[3.2. 数据生成 8](#_Toc482031261)

[3.3. 数据注入 9](#_Toc482031262)

[4. 工业时序大数据评测的负载 11](#_Toc482031263)

[4.1. 数据写入append 11](#_Toc482031264)

[4.2. 数据读取read 11](#_Toc482031265)

[4.3. 聚合操作 12](#_Toc482031266)

[4.4. 数据Update 12](#_Toc482031267)

[4.5. 数据random insert 12](#_Toc482031268)

[4.6. 数据delete before some timestamp 12](#_Toc482031269)

[4.7. 数据data export 13](#_Toc482031270)

[5. 系统的性能要求 14](#_Toc482031271)

[6. 系统的扩展性要求 14](#_Toc482031272)

# 引言

## 系统名称

本系统的名称为“工业时序大数据评测基准”。

## 系统描述

研发本系统的目的是，模拟生成真实场景下的工业时序大数据，并且持续地注入目标时序数据库，并且在目标时序数据库上，在较长的时间里，运行查询负载，观察性能的变化。在交叉时间范围里，对查询负载性能进行评测，目的是了解系统在不同数据规模下、以及在系统对自身进行维护的时候，系统性能是否有所下降。

本系统可以对各种时序数据库，比如influxdb、opentsdb、graphite、tsfile等，进行性能评测，以便用户对这些系统进行评估和选择。

## 本系统使用者

本系统的使用者包括时序数据库开发者，他们使用本系统对他们开发的时序数据库进行评测，验证新技术是否提高系统的性能。

本系统的使用者包括时序数据库的最终用户，他们使用本系统对若干时序数据库进行评测，以便选择符合他们业务需求的时序数据库产品。

## 本文读者

### 系统需求提出者

本系统的读者包括需求提出者、需求分析者。

需求规格说明书撰写完毕，需要经过评审阶段(review)，以便确认本需求规格说明书，已经把事先讨论的需求都正确地进行了描述。以便后续的设计和实现，不至于偏离原先的需求。

### 系统设计者

系统的设计人员，根据本文档，进行系统概要设计和详细设计。

### 系统开发者

系统开发者需要深入理解详细设计文档，并进行系统实现。他们一般不需要直接阅读本文档。

## 参考资料

无。

# 名词解释和对象描述

负载生成器

数据生成器

性能评测

1，评价系统是否具备某个功能

2，测试性能指标

提交查询

数据生成器

时序数据库

数据注入

图1. 时序大数据评测基准的主要构件

## 设备

一个工厂或者一个部门，有多个设备需要进行监控。

## 测点

每个设备有多个测点。测点以一定的速率，产生数据点。

## 数据点

所谓数据点，对应到某个设备、某个测试点(传感器)、某一个时刻生成的数据。这个数据除了包含device id、point id、timestamp等描述信息之外，还包含一个value。Value由一系列的key/value pair构成。

即data point = < device id, point id, timestamp, {key1=value1, key2=value2, …}>。

## 数据生成器

数据生成器，负责按照真实场景下的设备数量、传感器数量、时序数据生成频率、时序数据的数据类型、时序数据的波形特点等参数，持续地生成数据。并且以单条、小批量、文件导入等方式注入目标系统。

## 时序数据库

时序数据库是我们要评测的目标系统，我们要评测的系统主要包括influxdb、opentsdb、graphite、tsfile等。

## 负载生成器

负载生成器，按照一定的负载混合比例，生成系统的负载。我们生成的负载，不仅包括数据的读取查询，还包括少量的随机插入(random insert)等。具体的负载类型，将在本文的后续章节进行详细介绍。

## 性能评测模块

性能评测模块，取得负载生成器的负载，施加于目标系统，并且评测目标系统(1)是否具备该功能。(2)系统执行某项功能的性能指标。

# 工业时序大数据的实际应用场景和数据生成

## 工业时序大数据的实际应用场景

工业时序大数据的实际应用场景，包括地铁车辆的监控、风力发电机监控、智能电表、桥梁/隧道监控、车联网等。

在地铁车辆监控中，一个城市正在运行的地铁有上1000辆，每辆地铁车辆有近400个测试点(测点)，采用频率一般为秒级。

在风力发电机监控中，一个发电厂的风力发电机数量达到500台，每台设备有多达上100个测试点，采样频率一般为50hz，少量测点的频率达到KHz级别(也就是高频传感器，数量并不是很多)。

桥梁/隧道监控场景下，采集频率并不是很大，一般是几秒钟一次，但是需要监控的桥梁/隧道以及测点的数量大。

允许用户通过设定设备数量/测点数量，采样频率等参数，模拟生成时间序列数据。由于工业时间序列数据场景下，设备的类型不同，我们可以从设备类型入手(描述其测点数量、采样频率)，然后给出一个不同类型设备的混合比例，模拟真实场景的设备构成。

对设备数量、测点数量、采样频率进行设定的时候，可以选择10/100/1000等数量级。

经过抽象，我们可以对少数几类工业时序大数据场景进行建模，也就是需要覆盖多个应用场景。比如如下的两个场景(当然还可以增加其它场景)。

(1)设备数量/测点数量少，采样频率高，达到几千Hz。

(2)设备数量/测点数量多，采样频率不高，比如秒级采样。比如桥梁/隧道监控。

这两种场景，通过设备数量/测点数量多，或者采样频率高，产生大量的数据。而对目标系统的测试，只要在数据量大的时候，才能了解到系统的极限，才有意义。

## 数据生成

工业时序数据的主要特点有：

(1)生成速率方面，包括固定频率、可变频率、以及基于事件触发的数据生成。

(2)数据具有灵活的schema：即便是同一个设备的同一个测试点，由于可能设备更换了零件，所以前段时间和现在该设备该测试点，生成的数据的schema有可能是不一样的。所谓可变的schema，可以理解为每行数据，列数的多少有变化。这一点，在测试中要进行模拟。也就是从设备过来的数据包，包含schema不一样的测点数据。

Schema的灵活性和不规范性，有时候还体现在，不同批次到达数据库的时序数据，相同的数据列，名称不一样，对于这一点，我们在评测的时候一般不给予特别的关注。

(3)设备向数据库传递数据的时候，有可能是一个数据包一个测点数据，也可能是一个数据包包含多个测点数据，也就是把多个测点某个时刻的数据打个包，发送到时序数据库。

(4)数据类型方面，包括开关类型(0/1)、枚举类型、整数类型、单精度浮点数、双精度浮点数等。在数据生成时，对数据类型的比例可以进行配比。

(5)数据具有某种分布形状，也就是具有某种波形，包含波峰、波谷，并且带有季节性特点。

(6)工业时序数据可能有错，数据中的**错误**需要经过清洗修正。我们在测试的时候，不涉及清洗，但是需要在负载中，时不时地(不是那么频繁)进行update操作，模拟把错误数据修正过来。

如果时序数据的时间戳字段，不是按照预定的生成频率对应的时刻点，而是有所**偏差**，一般来讲，我们直接在数据库里进行记录，不做修改。

时序数据可能有**遗漏**情况，原因可能是设备发送过来了，但是在网络上丢失了。由于有数据遗漏状况，我们在负载里实现random insert，模仿对遗漏的数据进行插补操作。

时序数据有可能有**重复**状况。对于接收到重复数据的状况，测试基准必须保证只有一条数据进入数据库中。

此外，通过UDP网络进行数据传输，有可能造成数据**乱序**情况。即早先的时序数据后到达，后续的时序数据先到达。数据里包含时间戳，记录了事件真实发生的时间。

比如在12：30：31秒，“12：30：30秒”的数据到达了，我们对其进行append操作。在12：30：32秒，“12：30：29秒”的数据才到达，我们对其进行append操作。系统能否正确处理这种情况，我们需要进行考察。我们希望，数据都入库了，查询的时候，数据的顺序是对的。

(7)数据生成的频率以data points/sec进行度量。数据生成的模式包括定频、变频、和事件触发等三种类型。所谓定频，就是按照一定的频率生成数据，变频则是绝大部分时间按照某个频率生成数据，在某段时间、短暂地按照另外一个频率(一般是更高频率)生成数据，比如在预计设备将要发生故障的时候，短暂提升采样频率，采集某个测点数据，查看某个测点数据的变化。而事件触发类型则是，有事件发生的时候，生成时间序列数据data point，否则不生成。这三种类型，我们都需要模拟生成，重点模拟第一种。

(8)数据生成器，需要考虑若干方面，包括设备数量、测试点数量、每个数据点包含不同的数据项、每个数据项的类型可以配比等，数据是单条append、还是多条append、还是先生成文件再load，我们需要在配置文件中，对参数进行配置。数据生成程序根据配置文件，进行数据生成。

(9)数据生成器，可以配置成在单节点上以多线程方式运行。也可以配置成在若干节点上，以多进程方式运行。提高数据生成的速率，测试目标系统接收不同速率的数据，及时添加到数据库的能力。

## 数据注入

数据以三种方式注入到系统，分别介绍如下，

(1)第一种方式，是每生成一条数据，就及时插入。

(2)第二种方式，是在设备端，已经对数据进行缓存，每若干条数据(或者攒一段时间的数据，比如5秒)打一个包，提交到时序数据库，进行数据添加。

第一种方式，和第二种方式，将在下一节“负载”进行介绍，因为它是持续发生的。

第三种方式，则一般只在系统初始化的时候，做一次。

(3)第三种方式，是针对历史数据的导入的，类似于关系数据库管理系统的Load(或者Import)功能。数据以文件的方式提供，利用目标时序数据库的导入功能，实现数据添加。

一般来讲，在第三种情况下，我们要评测目标系统的数据**导入性能**，性能指标的单位一般取每条中多少个数据点(data points/sec)、或者每秒钟导入的数据量(MB/sec)。

并且考察导入以后的数据的**存储效率(压缩比)**，比如数据文件的形式是文本文件，大小是Size1。数据导入以后，使用某种数据结构以二进制方式存储，并且有可能进行了数据压缩，我们需要考察数据导入以后，占用的空间，大小为Size2。通过比较Size1和Size2，就可以了解到目标系统的存储效率。

在这里，需要注意，我们必须尽量按照真实场景生成时序数据，因为压缩存储效果，模拟数据和真实数据，有可能存在较大的差别。

# 工业时序大数据评测的负载

目前，在负载方面，主要考虑带设备id、测试点(传感器)id、时间范围、以及数值字段范围的查询，还包括聚合查询。目前阶段，不考虑频域计算，因为频域计算应该是由数据库系统负责还是由应用程序(应用层)负责，没有定论。目前阶段，也不考虑测试目标系统的join操作。

## 数据写入append

对系统进行数据写入的性能评测，写入的每条数据，列数可能是不一样的。也就是schema是动态变化的。

一般来讲，对系统的写入性能进行测试之前，首先利用数据生成器，生成一定量的数据，保存到数据文件中，然后进行数据导入(Load)。这时候，测试目标系统的数据导入性能，以及存储压缩比。

数据的写入性能测试和读取性能测试是同时进行的。数据写入测试的时间必须足够长，在这个期间，目标系统可能对数据进行compact(小文件合成大文件，对删除数据进行整理)操作，这时候对数据写入和读取都会产生不小的影响。

接下来，由数据生成器持续生成数据，模拟持续地单条的写入方式，以及持续地小批量的写入方式，测试目标系统的数据写入性能，性能指标的单位是每秒钟多少个数据点(data point/sec)或者每秒钟多少数据量(MB/sec)。

对于data point/sec性能指标来讲，每个data point的平均数据量的多少，也是我们需要关心的，所以，一般需要同时给出data point/sec和MB/sec两个性能指标。

## 数据读取read

对系统进行数据读取的性能评测。

数据读取的查询条件，包含设备id、测试点id、以及时间范围。通过改变这些参数，改变查询返回的数据条数。还可以对具体的**数值设定查询条件**，比如查询某个设备、某个测试点、某个时间范围内，其某个读数超过一定阈值的数据等。

查询要考虑对设备id、测试点id进行分组(group by)，以及计算一些聚集(count/min/max/sum/avg)等。

数据的写入性能测试和读取性能测试是同时进行的。数据读取测试的时间必须足够长，在这个期间，目标系统可能对数据进行compact操作(小文件合成大文件，对删除数据进行整理)，这时候对数据写入和读取都会产生不小的影响。

数据读取测试的性能指标，一般为每秒钟数据点数(data points/sec)，或者每秒钟数据量(MB/sec)。

备注：目标系统是否提供索引的支持，对数据读取的效率有重大影响。

## 聚合操作

在这里，所谓聚合操作，是从细粒度的数据获得粗粒度的数据，即改变时间精度。比如数据库里面存放的是秒级的数据，我们希望获得分钟级的汇总数据(mean、max、min、sum等)。

根据需要，聚合操作在时间精度上，可以从ms🡪sec🡪minute🡪hour，细粒度的时间精度数据，可以向比它高的各级精度进行聚合。

聚合操作，也需要设定设备id、测试点id、以及时间范围。通过改变这些参数，改变查询返回的数据条数。但是在数据返回之前，首先对数据进行聚合，获得粗粒度的数据。

聚合操作本质上是基于时间的分组聚集操作，比如如下的SQL语句，查询出设定设备id、测试点id、以及时间范围内的，每分钟的测点数据的最大值。

Select max(metirc2) from ts\_table

Where device\_id = ? and sensor\_id = ?

And timestamp >=? And timestamp <=?

Group by minute(timestamp)

聚合操作的性能指标为每秒钟数据点数(data points/sec)，或者每秒钟数据量(MB/sec)。

备注：有些目标系统，通过物化视图(预先进行聚合计算)等机制，事先计算了不同时间分辨率的数据，可以获得极高的聚合操作效率。

## 数据Update

对系统进行数据Update的性能评测。

Update的数量是极少的，它的目的是对**错误**的数据进行修正。

数据Update和数据的写入、数据的读取同时进行。按一定的时间间隔，比如符合泊松分布的时间间隔，发起数据的Update操作。

对于Update操作，要考察目标系统**是否支持Update，以及Update的效率**，一般用响应时间来度量，单位是毫秒。

## 数据random insert

数据的random insert，模拟的是时序数据**遗漏**的时候，需要对数据进行插补。

对该功能的测试，首先看目标系统**是否支持random insert，其次考察其效率**，一般用响应时间来度量，单位是毫秒。

为了进行random insert进行测试，生成数据的时候，就需要故意遗漏若干条数据，然后通过random insert进行插补。对于数据的遗漏和random insert必须有日志记录，以便进行对数据进行检查，是否已经正确进行插补。

注意，文件导入使用了目标数据库的Load功能，而单挑数据和小批量数据的注入是append操作。和这里的random insert是不一样的。

## 数据delete before some timestamp

Delete before some timestamp功能的目的是把某个时间戳以前的**历史数据**全部删除。数据删除之前，可能需要事先导出(export)到文件中，以便归档。

对于该功能的测试，首先要考察目标系统是否支持该功能，其次考察其执行的效率。执行效率，用每秒钟删除多少数据点(data points/sec)，或者每秒钟删除多少数据量来表示(MB/sec)。

需要注意的是，在某些目标系统上，该功能的执行效率，有可能和当前系统的数据量和系统负载有关，所以我们需要设定已有的数据量，然后按照某个时间(隐含了一定的数据比例)以前进行删除。

## 数据data export

测试目标系统的数据导出功能，一般导出成文本文件，或者parquet文件格式。评价导出的效率，用每秒钟删除多少数据点(data points/sec)，或者每秒钟删除多少数据量来表示(MB/sec)。

# 系统的性能要求

评测基准自己也有性能指标，我们主要关注评测基准软件是否能够模拟高频的数据生成。在一个普通服务器上，我们要求评测基准的数据生成器，能够达到XXX万data points/sec的生成速度，以每个数据点XXX个字节计算，换算成数据量为XXXMB/s的数据生成速度。

# 系统的扩展性要求

评测基准软件，可以配置成在单节点上以多线程方式运行。也可以配置成在若干节点上，以多进程方式运行。提高数据生成的速率，以及对不同并发水平的系统负载进行模拟。

# 关于功能点的初步考虑

配置文件

测点配置模板

数据模式（数据项可以配比，数据波形可以配置），数据生成频率，数据传送模式单条或者小批量

设备配置模板

包含若干个测点

场景配置

包含若干个设备

数据生成与注入

Offline generator

Online generator for opentsdb

Online generator for influxdb

Online generator for graphite

Online generator for tsfile SQL

模仿数据错误，记日志，注入系统，并且通知负载生成器生成update

模仿数据遗漏，记日志，注入系统，并且通知负载生成器生成random insert

模仿数据重复，记日志，注入系统，并且通知性能测试模块验证

模仿乱序递送数据给时序数据库，记日志，注入系统，并且通知性能测试模块验证

多线程版本

多进程版本

负载生成

批处理作业生成

数据导入

数据导出

delete before some timestamp，

混合负载生成，包括

Append, read, 聚合，

update错误修正并验证，random insert 遗漏并验证，验证重复数据插入是否正确处理，验证设备乱序递送数据给时序数据库是否得到正确处理

性能测试

批处理作业测试

混合负载测试

Mysql

加一些测试标准