时序数据库OPENTSDB使用总结

# OPENTSDB简介

何为时序数据？

时序数据是一组按照时间发生先后顺序进行排列的数据点序列，通常一组时间序列的时间间隔为一恒定值（如1秒、1分钟、1小时等）。时序数据具有如下特点：

* 每一个时间序列通常为某一固定类型的数值
* 数据按一定的时间间隔持续产生，每条数据拥有自己的时间戳信息
* 通常只会不断的写入新的数据，几乎不会有更新、删除的场景
* 在读取上，也往往倾向于读取最近写入的数据。

因为时序数据的这些特点，通常使用专门的时序数据库来存储，OpenTSDB就是一种时序数据库实现。其是基于Hbase的分布式的、可伸缩的时间序列数据库。存储到OPENTSDB的数据，是以metric为单位的，metric就是一个监控项，如监控服务器时，会有CPU使用率，内存使用率这些metric。除了metric外，OPENTSDB存储时还需要包括value、timestamp与tags。OPENTSDB采用metric + timestamp + tags组合rowkey，能唯一确定一条数据。

# OPENTSDB常用方法汇总

由于时序数据通常只会不断的写入新的数据，几乎不会有更新、删除的特点，所以并未提供更新的api，也未单独提供删除api。以下将主要从读、写两个方面来介绍，最后简要介绍删除方法。

## 2.1 数据写入OPENTSDB

OPENTSDB采用Post请求方式，提供了put接口以供数据的写入。Post请求地址格式如下：http://opentsdb-hofijmxaj857z14.cloudtable.com:4242/api/put。OPENTSDB存储数据时必须包含4个字段：”metric”、”timestamp”、”value”与”tags”。

* **Metric**：表示监控项，如服务器CPU使用率，服务器内存使用率等；
* **Value**：表示metric的值；
* **Timestamp**：value对应的时间戳，及描述value是什么时候的；
* **Tags**：一些标签值，用于描述metric的。

下面使用Postman来展示OPENTSDB方法的使用。

实例一：单条写入

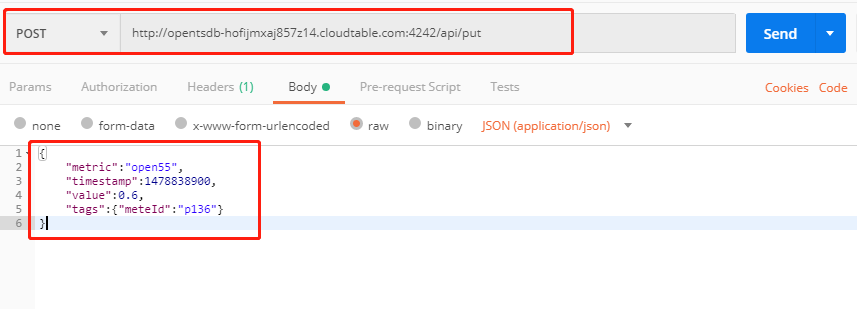


图1

输入格式为JSON，“tags”包含一个小JSON。若写入成功，则无任何返回。

实例二：多条写入

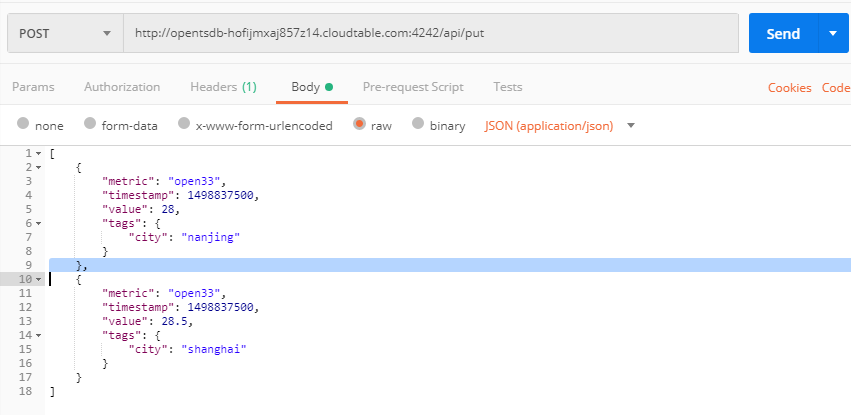


图2

OPENTSDB也提供批量写入，通过将多条JSON数据组成一个array来写入。

注意：对于相同metric、timestamp、tags的数据，先前的数据会被后写入的数据覆盖。

## 2.2 OPENTSDB数据查询

OPENTSDB的查询使用query接口，采用Post请求方式。Post请求格式如下：<http://opentsdb-hofijmxaj857z14.cloudtable.com:4242/api/query。>OPENTSDB数据查询接口包含以下参数：

* Start：查询的开始时间，必备参数；
* End：查询结束时间，可选参数。若不指定结束时间，则表示从开始时间到当前时间；
* Queries: 子查询条件，必备参数，为数组形式，至少包含一个或多个条件。Queries中的每个条件中必须包含aggregator与metric两个字段；
* Delete：可选参数，根据给定查询条件来删除数据点。

其它参数可参考文档中的“OPENTSDB查询”，这里不做详细介绍。常用的Queries查询子条件主要包含以下参数：

* **Metric**：必选参数，和写入时一致；
* **Aggregator：**必选参数，聚合函数（一般使用none）；
* **Tags**：可选参数，Map类型；
* **Downsample：**可选参数，对查询结果作进一步采样聚合。

下面从实例来讲解opentsdb的使用。

实例三：查询一段时间内的全量数据：

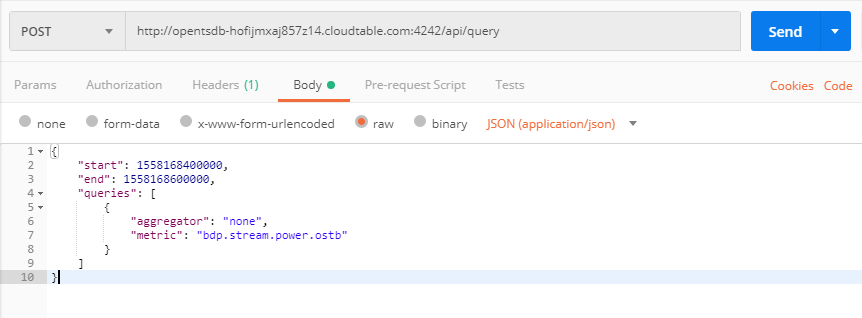


图3

"start": 1558168400000为开始时间戳，也有便捷写法，如1d-ago（1天前）、10m-ago（10分钟前）等，常用的时间单位参数还有s-秒、h-小时、d-天、w-星期、n-月、y-年。end参数为可选参数，若有则以指定时间参数为准；若未指定此参数，默认当前时间。“metric”参数与写入时一致。若有返回值，返回类型为JSON数组类型。每个json中包含“metric”、“tags”与“dps”等字段。“dps”中包含不同时间段的监测值。若无返回值，则返回空数组。

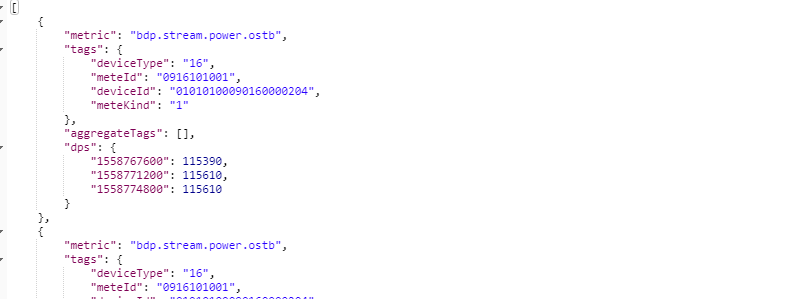


图4

实例四：指定设备信息

在tags里可以指定metric的描述信息，精确定位到某一设备。具体查询方法为：

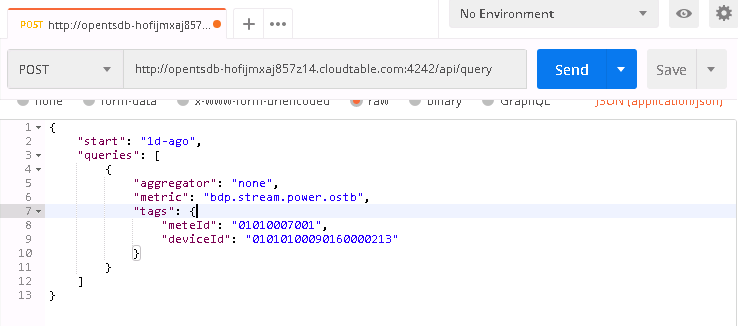


图5

查询结果为：



图6

实例五：查询最大值（最小值、平均值，统计，求和）

OPENTSDB提供了对查询数据的聚合，如最大值（max）、最小值（min）、求和（sum）、平均值（avg）以及统计（count）等。其查询方法为：

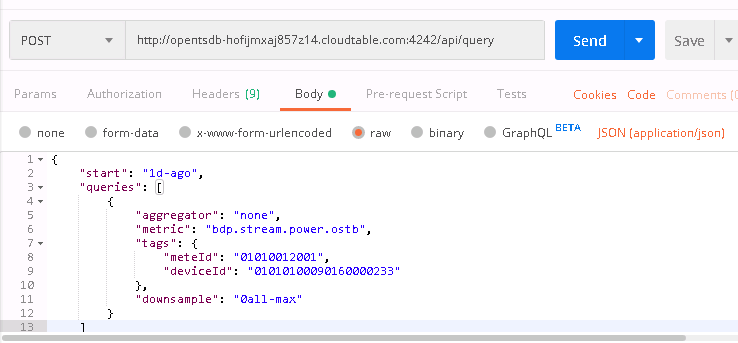


图7

“0all-max”表示查询的全量结果进行求最大值，可以根据具体需求，使用相应的聚合函数。图8展示了查询最大值的结果。



图8

实例六：查询一周内，每日数据的最大值

Downsample参数提供了对查询结果的进一步处理，可以对查询结果进行采样，减少数据量。下采样参数至少需要两个参数：

* 时间间隔（interval）-必选：用于聚合的时间范围，如1h、5m、10s、3n等；注意0all表示对start到end之间查询的所有值进行聚合。
* 聚合函数-必选：数学计算函数，常用的包括max、sum、count与dev等，其它聚合请参考：http://opentsdb.net/docs/build/html/user\_guide/query/aggregators.html。
* 填充参数-可选：如果希望所有查询的序列都返回值，可以采用填充参数。常用的填充参数包括：None（none）、NaN（nan）、Null（null）与Zero（zero）。

以查询一周内每天数据的最大值为例：

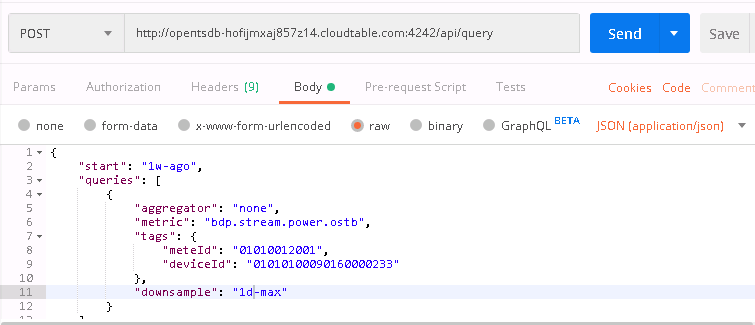


图9

查询结果：

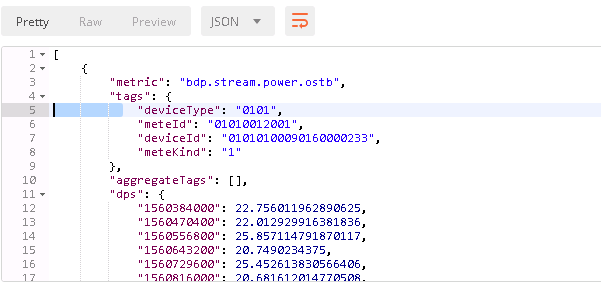


图10

## 2.3 多条件查询与聚合函数的使用

在2.2中所述的查询主要是单条件查询，而在实际应用场景中多条件的应用却十分广泛。如查询多个设备的功率值、查询某一设备的最大值、最小值、平均值等。使用filters字段可以实现多个设备值的聚合，filters参数为列表形式。列表中的每个元素为json格式。每个json必须包含“type”、“tagk”与“filer”字段。各字段含义如下：

* type：过滤器类型。常用的过滤器有正则表达式过滤器（regexp）、通配符过滤器（wildcard）、管道符过滤器（literal\_or）等。
* tagk：具体条件字段，如设备ID（deviceId）。
* filer：过滤条件表达式。

下面从两个应用应景来展示多设备的使用。

应用场景一：查询多个设备多个测点的值

查询多个设备（deviceId）和多个测点（meteId）的测量值可以通过filters的两个json子条件来控制，如图11所示。

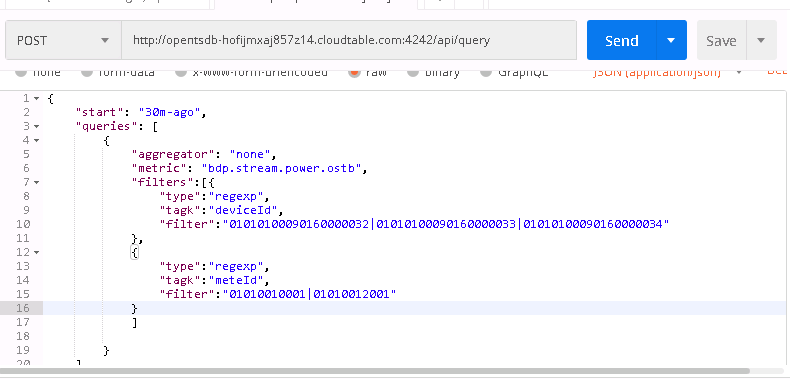


图11

应用场景二：多个设备间进行聚合

一些应用场景中，会使用多设备间的值进行聚合。如计算所有设备在同一时刻的能耗的总量。这时候需要用到聚合参数“aggregator”。Aggregator可以将在同一时间点的多个设备进行聚合，如下例中就是对多个设备进行求和。

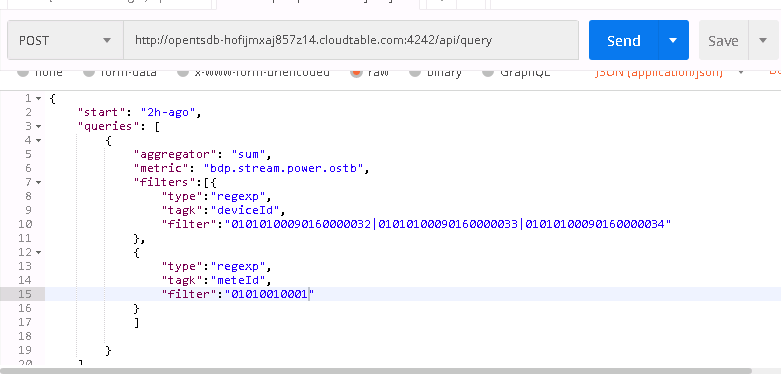


图12

应用场景三：同时查询设备测点的最大值、最小值、平均值等

一次性返回多个聚合值，可以通过Queries的子查询条件来实现。Queries参数是列表形式，可输入多个子条件。

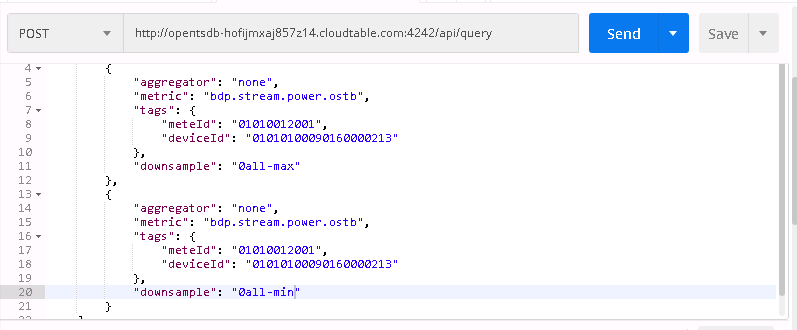


图13

## 2.4 数据删除

虽然时序数据库并未提供专属的删除api，但也将删除功能嵌入到查询方法中。首先，查询数据：

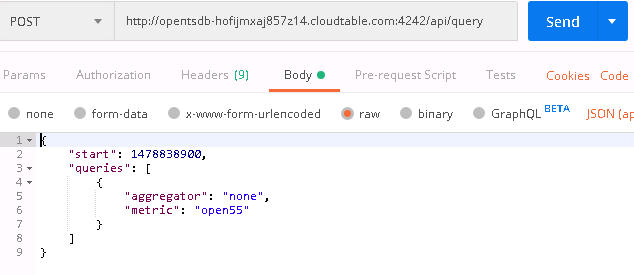


图14

结果如下：

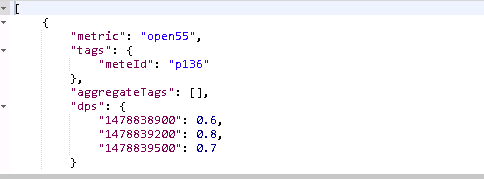


图15

删除方法：



图16

再次使用图14的方式查询时，可以发现查询结果为空。删除方式十分简单，采用与查询相同的方式，只是添加了“delete”字段。

# 参考文档

OPENTSDB写入：<http://opentsdb.net/docs/build/html/api_http/put.html>

OPENTSDB查询：<http://opentsdb.net/docs/build/html/api_http/query/index.html>

开发文档：<https://support.huaweicloud.com/devg-cloudtable/cloudtable_01_0066.html>