# **OPENTSDB整改方案**

## **原数据样例**

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | tag | timestamp | value |
| Bdp.stream.power.ostb | Deviceid,meteid | XXX | XXX |

由于opentsdb底层采用HBASE存储，最终映射为HBASE数据结构为：

|  |  |
| --- | --- |
| Rowkey | value |
| Bdp.stream.power.ostb,14xxxxx{divicedi=123,meteid=456} | XXX |

引发问题：

按照此数据格式，HBASE表格中前缀固定为Bdp.stream.power.ostb那么在批量查询（指定特定设备ID，查询某些特定监控量值）HBASE需要全表SCAN，带来的性能损耗非常巨大，需要进行优化ROWKEY设计。

## **优化方案**

根据目前业务场景，对于OPENTSDB聚合操作多半是指定特定监控量，查询N个设备聚合值，固OPENTSDB优化结构如下：

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | tag | timestamp | value |
| meteid | Deviceid | XXX | XXX |

以meteid作为Metric监控指标取代原先自定义无意义常量：Bdp.stream.power.ostb

优化后HBASE rowkey结构如下：

|  |  |
| --- | --- |
| Rowkey | value |
| meteid=456,14xxxxx{divicedi=123} | XXX |

指定meteid进行多个设备进行聚合值，HBASE SCAN操作只需指定特定的METEID分区，不需要全表扫描，

大大提升检索性能。

## **具体方案**

1. 实时数据切割

修改OPENTSDB数据写入流处理程序，将原先Metric值更改为meteid并且从原先tags标签中移除meteid。

1. 历史数据恢复

历史数据恢复按照数据切割时间点，采用离线MR任务，定期从HIVE中按天读取历史数据，插入OPENTSDB中，历史数据恢复完成后从hbase tsdb表清楚Bdp.stream.power.ostb数据。

1. 应用端访问逻辑修改

Post请求格式如下：<http://opentsdb-hofijmxaj857z14.cloudtable.com:4242/api/query。>OPENTSDB数据查询接口变动参数

* 1. **Metric**：必选参数，由原来固定的Bdp.stream.power.ostb改为具体的meteid；
  2. **Aggregator：**必选参数，聚合函数（不变）；
  3. **Tags**：可选参数，Map类型（删除meteid标签）；
  4. **Downsample：**可选参数，对查询结果作进一步采样聚合（不变）。

## **数据流程**

