参考：<http://www.jianshu.com/p/0bafd0168647>

**Opentsdb简介（一）**

[](http://www.jianshu.com/u/6ca93a173ea2)

作者 [沉默剑士](http://www.jianshu.com/u/6ca93a173ea2) 关注

2014.09.03 10:14\* 字数 1625 阅读 12214评论 5喜欢 24

**1、OpenTSDB介绍**

**1.1、OpenTSDB是什么？主要用途是什么？**

官方文档这样描述：OpenTSDB is a distributed, scalable Time Series Database (TSDB) written on top of HBase；-OpenTSDB是一个分布、可扩展的基于HBase时间序列数据库。

翻译过来就是，基于Hbase的分布式的，可伸缩的时间序列数据库。

主要用途，就是做监控系统（me：ELK同样可以作为日志监控系统）；譬如收集大规模集群（包括网络设备、操作系统、应用程序）的监控数据并进行存储，查询。

**1.2、介绍continue**

存储到OpenTSDB的数据，是以metric（me：度量的）为单位的，metric就是1个监控项，譬如服务器的话，会有CPU使用率、内存使用率这些metric；

OpenTSDB使用HBase作为存储，由于有良好的设计，因此对metric的数据存储支持到秒级别；

OpenTSDB支持数据永久存储，即保存的数据不会主动删除；并且原始数据会一直保存（有些监控系统会将较久之前的数据聚合之后保存）

**2、OpenTSDB存储相关的概念**

介绍这些概念的时候，我们先看一个实际的场景。

譬如假设我们采集1个服务器（hostname=qatest）的CPU使用率，发现该服务器在21:00的时候，CPU使用率达到99%

下面结合例子看看OpenTSDB存储的一些核心概念

1）Metric：即平时我们所说的监控项。譬如上面的CPU使用率

2）Tags：就是一些标签，在OpenTSDB里面，Tags由tagk和tagv组成，即tagk=takv。标签是用来描述Metric的，譬如上面为了标记是服务器A的CpuUsage，tags可为hostname=qatest

3）Value：一个Value表示一个metric的实际数值，譬如上面的99%

4）Timestamp：即时间戳，用来描述Value是什么时候的；譬如上面的21:00

5）Data Point：即某个Metric在某个时间点的数值。

     Data Point包括以下部分：Metric、Tags、Value、Timestamp

     上面描述的服务器在21:00时候的cpu使用率，就是1个DataPoint

保存到OpenTSDB的，就是无数个DataPoint。

下面讲一下，OpenTSDB是如何保存DataPoint的。

Me：综合来讲，openTSDB保存的就是某个监测项，在某一个时刻某个指标的值，这样形成一个时间记录的序列。

**3、OpenTSDB的设计**

还是以例子来说明，譬如保存这样的1个DataPoint：

metric：proc.loadavg.1m

timestamp：1234567890

value：0.42

tags：host=web42，pool=static

**3.1、简单的设计**

那么，如果是一般的设计，会怎么做呢，可能就是：RowKey=metric|timestamp|value|host=web42|pool=static，Column=v，Value=0.42

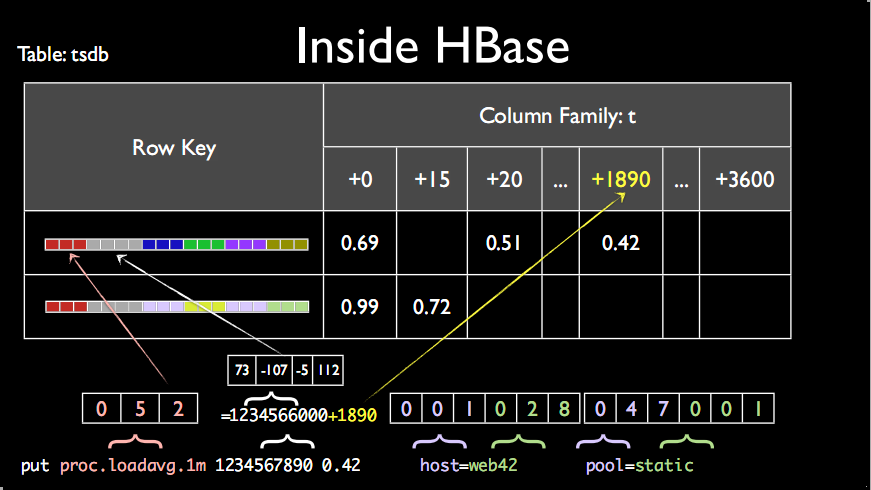
这是最简单的设计，那接下来看看，OpenTSDB是怎么做的吧。

**3.2、OpenTSDB的方案**

OpenTSDB使用HBase存储，核心的存储，是有两张表，tsdb和tsdb-uid

**3.2.1、表tsdb**

tsdb是保存数据的，看看该表的设计



tsdb的表结构

1）RowKey的设计

RowKey其实和上面的metric|timestamp|value|host=web42|pool=static类似；

但是区别是，OpenTSDB为了节省存储空间，将每个部分都做了映射（me:很多思想类似，如twitter的分布式数据库的设计和mongodb的ObjectId的设计）。

在OpenTSDB里面有这样的映射，metric-->3字节整数、tagk-->3字节整数、tagv-->3字节整数

上图的映射关系为，proc.loadavg.1m-->052、host-->001、web42-->028、pool-->047、static-->001

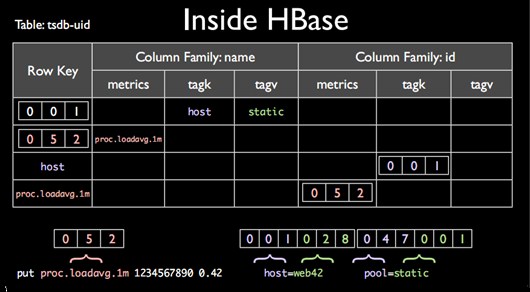
2）column的设计

为了方便后期更进一步的节省空间。OpenTSDB将一个小时的数据，保存在一行里面。

所以上面的timestamp1234567890，会先模上一个小时的时间戳查看过了多少秒，得出1234566000，然后得到的余数为1890，表示的是它是在这个小时里面的第1890秒；

然后将1890作为column name，而0.42即为column value

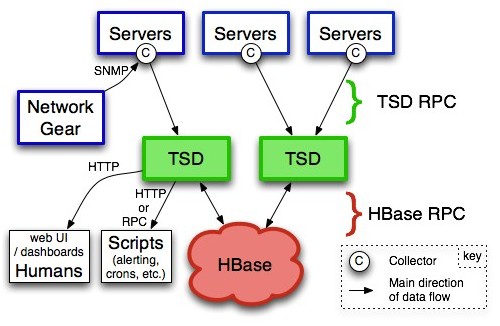
**3.2.2、表tsdb-uid**



表tsdb-uid

这里其实保存的就是一些metric，tagk，tagv的一些映射关系。

**4、OpenTSDB的总体架构**



openTSDB架构

Servers：就是服务器了，上面的C就是指Collector，可以理解为OpenTSDB的agent，通过Collector收集数据，推送数据；

TSD：TSD是对外通信的无状态的服务器，Collector可以通过TSD简单的RPC协议推送监控数据；另外TSD还提供了一个web UI页面供数据查询；另外也可以通过脚本查询监控数据，对监控数据做报警

HBase：TSD收到监控数据后，是通过AsyncHbase这个库来将数据写入到HBase；AsyncHbase是完全异步、非阻塞、线程安全的Hbase客户端，使用更少的线程、锁以及内存，可以提供更高的吞吐量，特别对于大量的写操作。

**5、简单的应用**

**5.1、安装和部署**

OpenTSDB是以HBase作为存储的，那么，首先你得部署个Hbase。就好像对想要娶老婆生小孩的屌丝来说，首先，你得有个女朋友。

接下来OpenTSDB的安装和部署，这个官方文档已经说得很详细，也是比较简单的，几乎是一键部署。

**5.2、监控数据收集**

OpenTSDB已经自带了收集监控数据的一些脚本。主要由2个部分组成，TCollector和一些具体的collectors。

TCollector是1个客户端进程，它主要是收集具体的collectors收集到的监控数据，然后负责将数据推送到TSDB；

TCollector会处理和TSD的连接和协议处理，具体的collectors负责收集数据即可

目前OpenTSDB提供了一些已实现的collectors

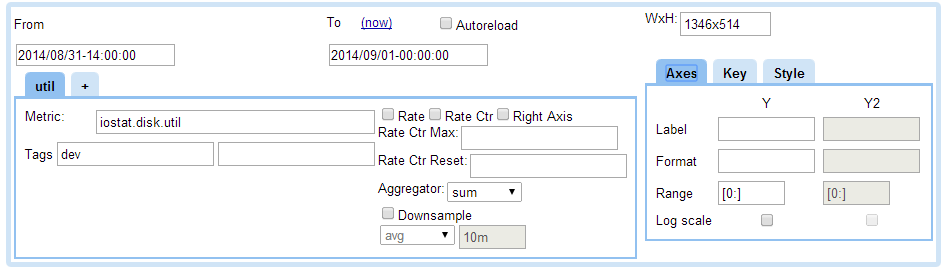
http://upload-images.jianshu.io/upload_images/58733-356340cfe3026fba.png

collectors

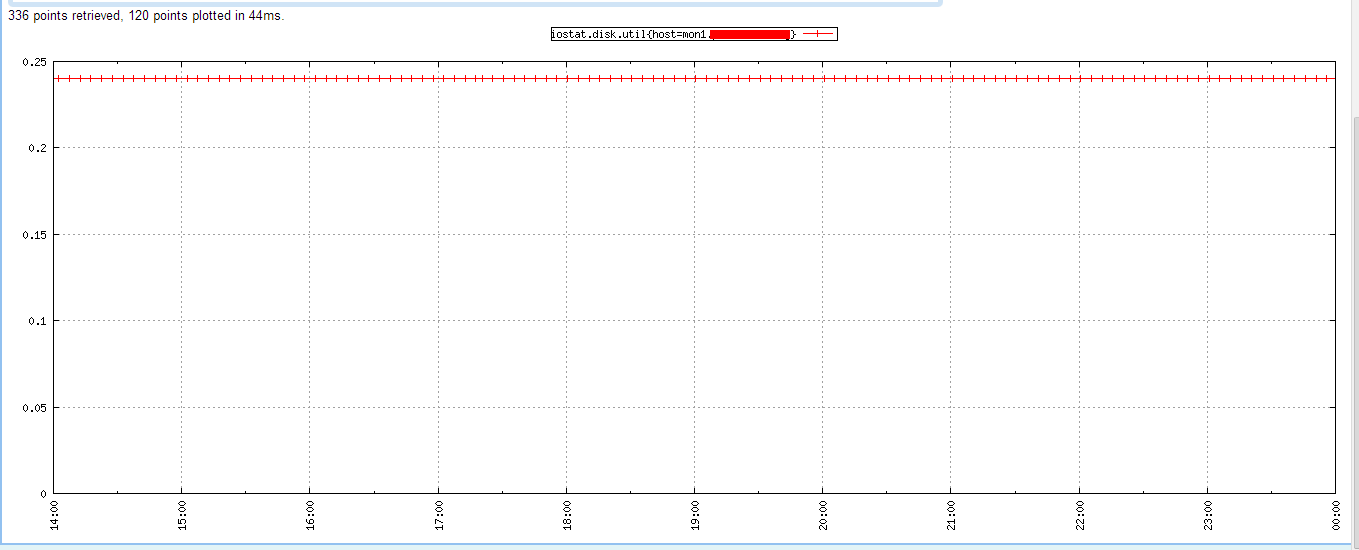
在一个服务器上部署1个TCollector之后，它会把这些collectors启动起来，collectors收集到监控数据之后，输出到stdout，TCollector接收后推送到OpenTSDB

**5.3、监控数据查看**

TSD除了接收数据之外，还提供了一个web页面供数据查询。不过页面的样式确实有点难以接受；



查询条件面板



查询结果面板

**5.4、报警**

目前OpenTSDB只是收集数据，并没有做报警。

或者说，OpenTSDB本身就是一个基于HBase的数据库，TCollector是数据收集器，报警不属于它的范畴。报警的话，可以接入到Nagios。

**6、总结**

这篇文章只是简单介绍一下OpenTSDB是什么东西，简单地谈了一下存储和架构，以及应用起来是如何的。接下来还有很多东西可以继续写。

譬如：

1）关于RowKey设计的更多东西（数据压缩、查询性能、数据均衡性、设计方案的缺点、metric和tagk长度）

2）OpenTSDB的更多特性（Aggregation、Interpolation 、Downsampling、Metadata、Tree）

可能接下来的文章再继续了。