凯多份™
CARBONITE

HBase应用与优化分析

微博：@卢亿雷
邮箱：johnlya@163.com
2013-4-13

提纲

- HBase分布式数据库特点
- Hbase经验分享
- HDFS经验分享
- Hbase案例Rowkey设计分享
- Hbase案例Filter设计分享

HBase分布式数据库特点

- 行强一致性
- 水平自动伸缩
  - Region的自动分裂，生产系统需要看具体情况
- 任意增加列
- 高性能随机写
- 支持Thrift框架

➢ 合理设计RowKey—极其重要

- OpenTSDB
- 查询地理信息系统

➢ 充分利用Filter功能

- SingleColumnValueFilter
- SubstringComparator
- BinaryPrefixComparator
- FamilyFilter
- QualifierFilter
- ColumnPrefixFilter
- ColumnPaginationFilter

- 应用端需要做数据安全检查

- RegexStringComparator
- STARTROW/STOPROW

- 可根据应用需求重写某些方法

- 在org.apache.hadoop.hbase.filter创建自己应用filter
- 打包成jar文件拷贝至Hbase环境目录
- 重新启动Hbase集群

➢ 考虑容量开启压缩

- 目前主要是lzo方式

➢ 提高随机读性能

- 前端增加一个分布式缓存Radis系统

➢ 系统参数优化

- GC策略：-XX:+UseConcMarkSweepGC -XX:+UseParNewGC -XX:CMSInitiatingOccupancyFraction=70
- 读写策略优化

系统参数优化

- 读优化：
  - hbase.regionserver.handler.count
  - hbase.regionserver.global.memstore.upperLimit/lowerLimit
  - hbase.hregion.memstore.block.multiplier
  - hbase.hstore.blockingStoreFiles
  - hbase.hregion.max.filesize

- 写优化：
  - Bloomfilter
  - in-memory
  - Blockcache
  - hfile.block.cache.size

➢ 系统参数优化

- GC策略

➢ 带宽策略优化

- 区分带内与带外心跳
- NameNode的备份不影响正常带宽使用

➢ 同步锁机制尽量少用

- 所有文件IO操作的地方尽可能不要加同步锁
- 大锁尽可能拆成小锁

➢ 文件复本数设置

- 根据应用的访问频率设置不同份数

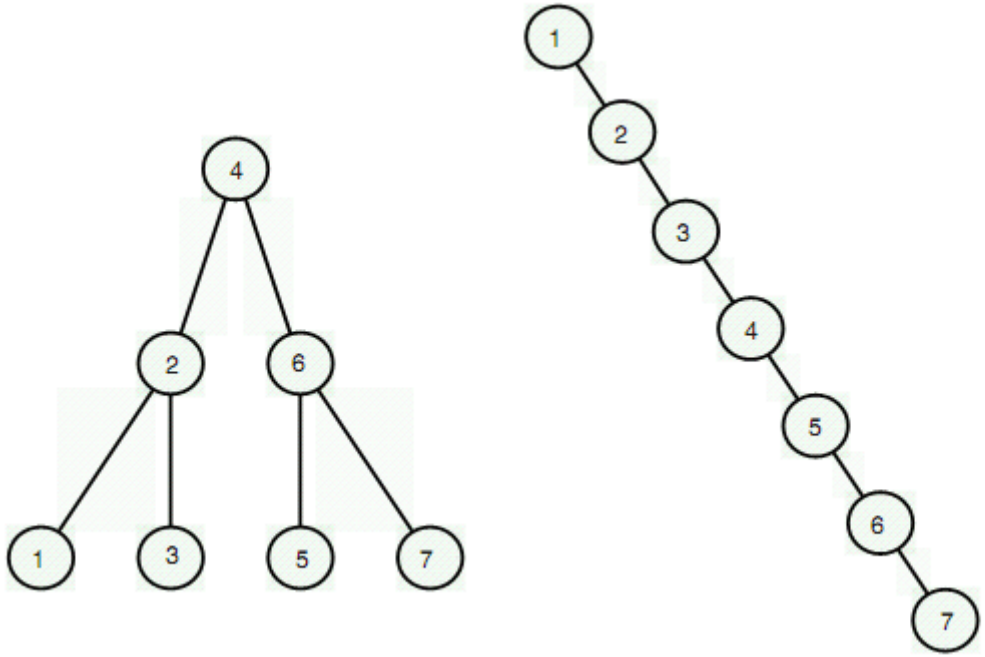
NameNode采取人工切换模式

NameNode启动失败分析方法

- 查看日志，确定问题位置
- 备份fsimage，edits等
- 使用OfflineImageViewer方法查看（可能不行）
- 编译相应Hadoop版本
- 在相关读取文件信息的地方适当加一些try/catch
- 多次查找分析元数据信息
- Replay重新生成fsimage

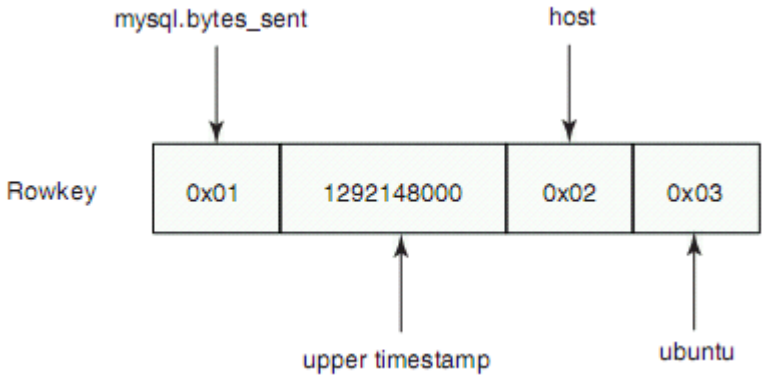
OpenTSDB

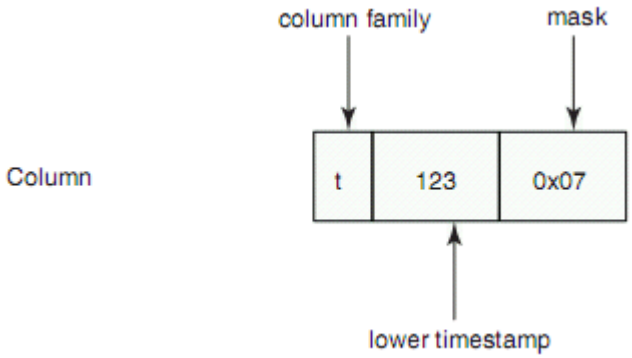
- 用HBase存储所有的时序（无须采样）来构建一个分布式、可伸缩的时间序列数据库

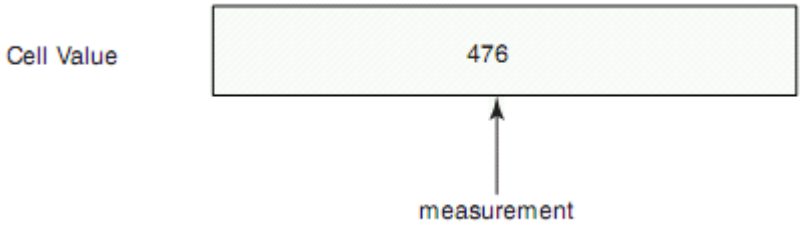


Hbase案例Rowkey设计分享

- OpenTSDB







查询地理位置信息

- 经度和纬度定位某个地理位置信息
- 通过geohash方法将经纬度生成rowkey
  - ✓ geohash是一种把几个值转换成单个值的函数。且这些值中的每一个必须来自于一个有固定范围的维度。

查询某一行的列，需符合如下条件

1、分页返回

2、列前缀限制

3、列必须返回含有某字符串的值

4、过滤某些不符合条件的列

```
HTable table = new HTable(hbaseConf, "TestTable");
Get get = new Get(Bytes.toBytes(rowKey));

FilterList filterList = new FilterList();

//column page filter
filterList.addFilter(new ColumnPaginationFilter(maxKeys,
    0));
```

```
//prefix filter
filterList.addFilter(new
    ColumnPrefixFilter(Bytes.toBytes(prefixColumn)));

//column filter
filterList.addFilter(new
    QualifierFilter(CompareOp.EQUAL,new
    SubstringComparatorLast(key, 1)));
```

Hbase案例Filter设计分享

凯备份™ CARBONITE

```
for (String filter : strArray) {
filterList.addFilter(new QualifierFilter(
CompareOp.NOT_EQUAL,
new
    BinaryPrefixComparator(Bytes.toBytes(exclusiveColum
    n))));
}

get.setFilter(filterList);
```

```
Result rs = table.get(get);

for (KeyValue kv : rs.raw()) {
    System.out.println(kv.getValue());
}
```

THANKS