

# 小米HBase实践

刘绍辉

小米云存储组

China Hadoop Summit 2013

1. HBase简介
2. HBase在小米的使用
3. 小米开发的重要特性
4. 未来开发计划

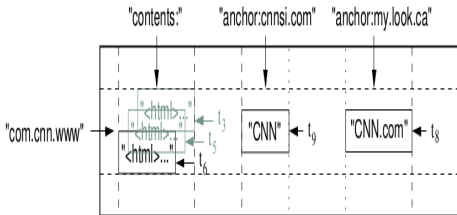
1. HBase简介
2. HBase在小米的使用
3. 小米开发的重要特性
4. 未来开发计划

# HBase是什么？



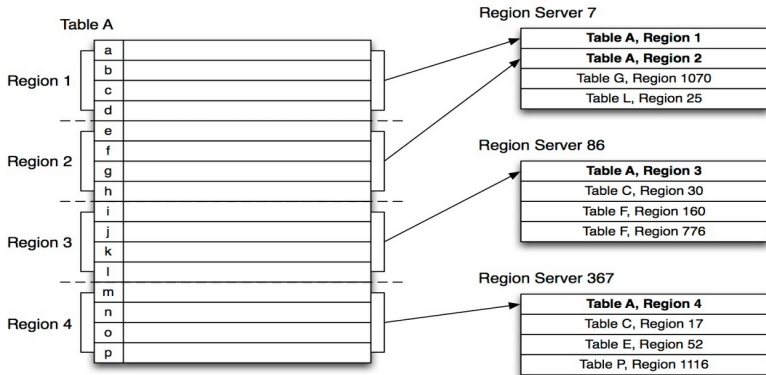
- ▶ Google Bigtable系统的开源实现
- ▶ 分布式的，可扩展的，一致性的，半结构化数据存储系统
- ▶ 稀疏的，一致性的，分布式的，多维有序的映射表

- ▶ 表
- ▶ 行
- ▶ 列簇
- ▶ 列
- ▶ 版本(时间戳)



多维映射表：

(行key, 列簇, 列, 版本) → 值



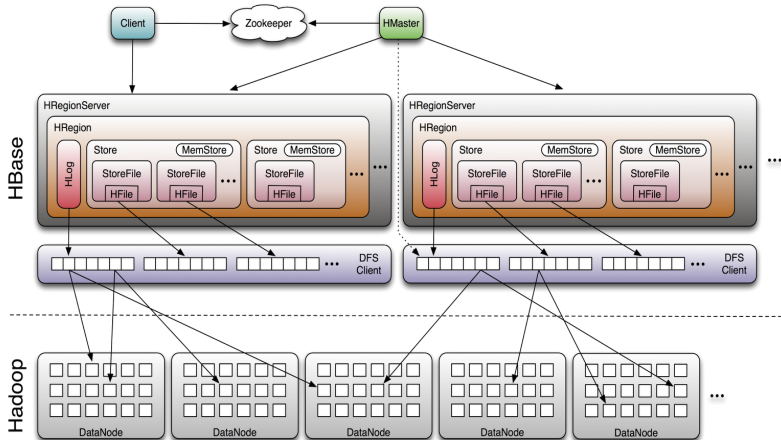
**Legend:**

- A single table is partitioned into Regions of roughly equal size.
- Regions are assigned to Region Servers across the cluster.
- Region Servers host roughly the same number of regions.

参考: <http://www.slideshare.net/xefyr/h-base-for-architectspptx>

HMaster 负责控制流

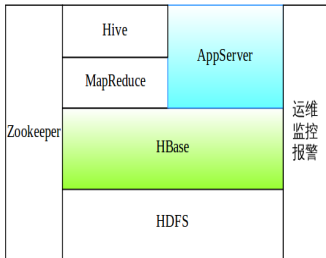
HRegionServer 负责数据流



1. HBase简介
2. HBase在小米的使用
3. 小米开发的重要特性
4. 未来开发计划



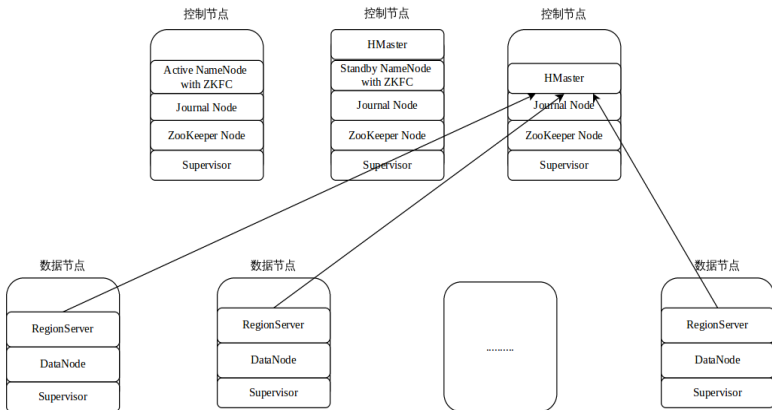
- ▶ 米聊消息全存储
- ▶ MiCloud: 短信, 通话记录同步
- ▶ 小米Push服务
- ▶ 其他一些离线分析业务



## 配置

节点类型	数量	CPU	Memory	Disk
控制节点	3 - 5	16核	64G	6 * 600G SAS, RAID10
数据节点	5 - N	16核	64G	12 * 2T SATA/SAS

- ▶ 控制节点：ZooKeeper, NameNode/ZKFC/JournalNode, HMaster
- ▶ 数据节点：DataNode, RegionServer



## 命令行工具

- ▶ 初始化: `deploy.py install/bootstrap hbase bjsrv-test`
- ▶ 启/停: `deploy.py start/stop hbase bjsrv-test`
- ▶ 展示: `deploy.py show hbase bjsrv-test`
- ▶ 升级: `deploy.py restart/rolling_update hbase bjsrv-test`
- ▶ 删除: `deploy.py cleanup hbase bjsrv-test`

```
2013-11-13 17:53:21 Showing task 0 of regionserver on 10.237.12.13(0)
2013-11-13 17:53:21 Task 0 of regionserver on 10.237.12.13(0) is RUNNING
2013-11-13 17:53:21 Showing task 1 of regionserver on 10.237.12.14(0)
2013-11-13 17:53:21 Task 1 of regionserver on 10.237.12.14(0) is RUNNING
2013-11-13 17:53:21 Showing task 2 of regionserver on 10.237.12.15(0)
2013-11-13 17:53:21 Task 2 of regionserver on 10.237.12.15(0) is RUNNING
2013-11-13 17:53:21 Showing task 0 of master on 10.237.12.13(0)
2013-11-13 17:53:21 Task 0 of master on 10.237.12.13(0) is RUNNING
2013-11-13 17:53:21 Showing task 1 of master on 10.237.12.14(0)
2013-11-13 17:53:21 Task 1 of master on 10.237.12.14(0) is RUNNING
```

## Dashboard

10.237.101.59:8080/monitor/



OWL- Cluster Monitor Business Longhaul

admin logout

## All clusters for all services.

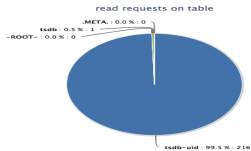
name	job (running/total tasks)	cluster entry	version	description
✓ hdfs / dptst-example	✓ journalnode (3/3) ✓ namenode (2/2) ✓ datanode (3/3)	10.237.101.59:12201	2.0.0-mdh1.0.0-SNAPSHOT, r115665	
✓ hbase / dptst-example	✓ master (1/2) ✓ regionserver (3/3)	10.237.101.65:12501	0.94.3-mdh1.0.0-SNAPSHOT, r115665	
! impala / dptst-example	! statelstored (0/1) No running statelstored! ! impalad (0/3) Too few running impalad!			
! yarn / dptst-example	! resourcemanager (0/1) No running resourcemanager! ! nodemanager (0/3) Too few running nodemanager! ! historyserver (0/3) Too few running historyserver! ! proxyserver (0/1) No running proxyserver!			

## Dashboard

hbase / dptst-example

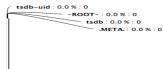
Tasks Basic Tables RegionServers Replication

### QPS distribution on tables



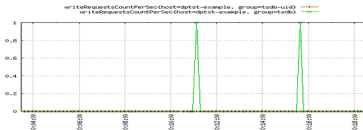
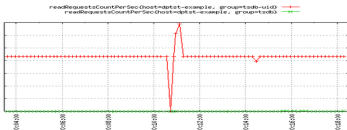
Highcharts.com

write requests on table



Highcharts.com

### QPS Comparison



### System Table

table	memstore size(MB)	storefile size(MB)	read qps	write qps	Availability
.META.	0	0	0	0	
-.ROOT-	0	0	0	0	

### User Table

table	memstore size(MB)	storefile size(MB)	read qps	write qps	Availability
tsdb-uid	1	0	216	0	
tsdb	245	302	1	0	

性能指标: 分类展示

基于OpenTSDB: <http://opentsdb.net/>

Operation

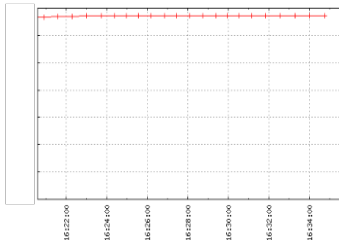
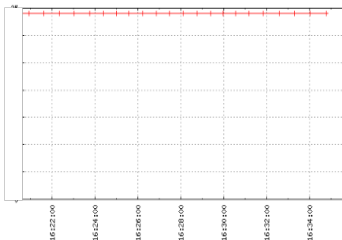
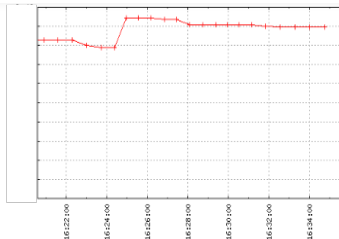
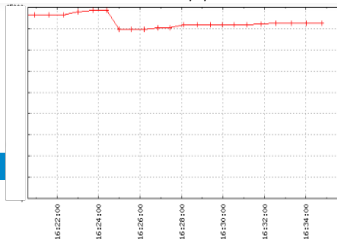
RPC

Store

BlockCache

FileSystem

JvmStatistics



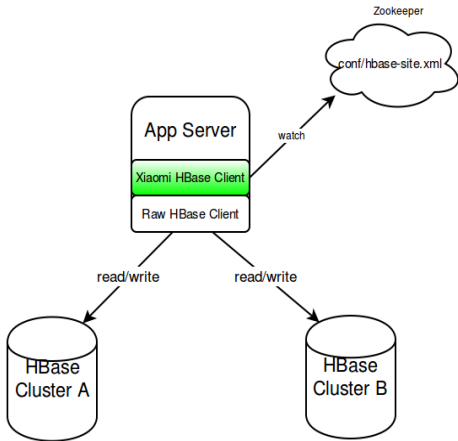
开源 <https://github.com/xiaomi/Minos>





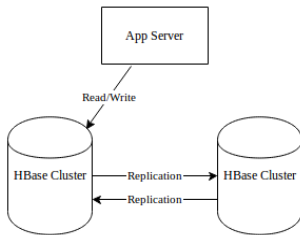
## HBase Client 封装

- ▶ 线程安全
- ▶ 自动添加性能指标
- ▶ 跨表、跨集群对用户透明
- ▶ 动态更新客户端配置

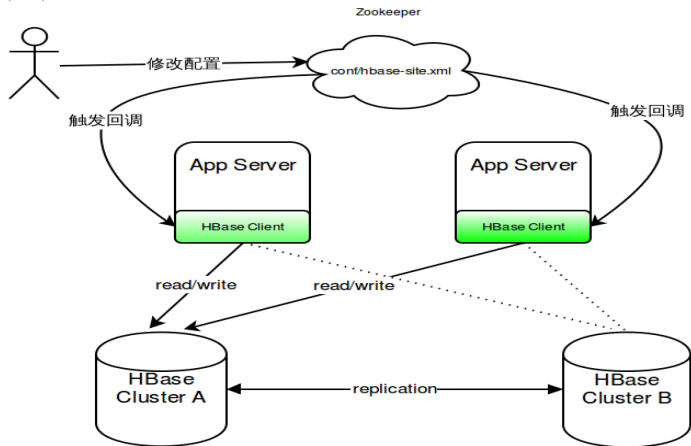


## 跨机房：双主复制

- ▶ 中途添加一个复制  
add\_peer + CopyTable
- ▶ 复制数据一致性验证  
VerifyReplication



## 主备集群自动切换



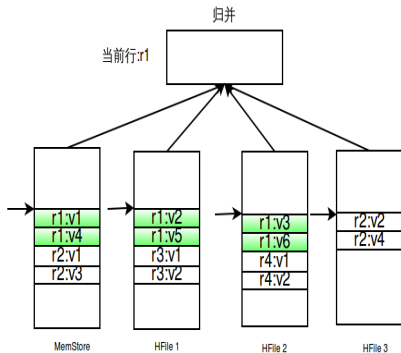
- ▶ 平滑升级  
基于move region脚本, 减少不可用时间
- ▶ Full GC  
每天低峰期触发Full GC: `jmap -histo:live $pid`
- ▶ Compaction  
`hbase.offpeak.start/end.hour`
- ▶ Shortcircuit Read  
`dfs.client.read.shortcircuit`
- ▶ 安全  
Kerberos + HBase ACL

1. HBase简介
2. HBase在小米的使用
3. 小米开发的重要特性
4. 未来开发计划

正向scan的基本过程

eg: scan  $[r1, r3]$

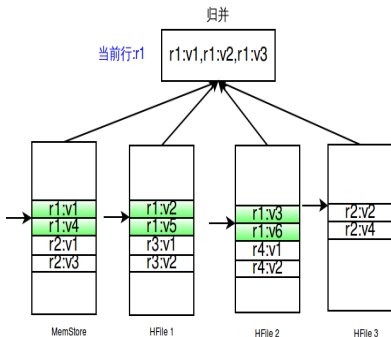
1. 顺序读取当前行数据后，自然读到下一行开头
2. 选取最小行作当前行的“下一个行”，重复上述过程



正向scan的基本过程

eg: scan  $[r1, r3]$

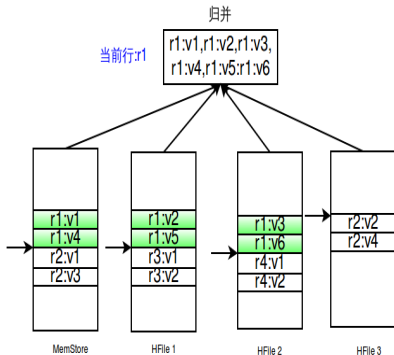
1. 顺序读取当前行数据后，自然读到下一行开头
2. 选取最小行作当前行的“下一个行”，重复上述过程



正向scan的基本过程

eg: scan  $[r1, r3]$

1. 顺序读取当前行数据后，自然读到下一行开头
2. 选取最小行作当前行的“下一个行”，重复上述过程

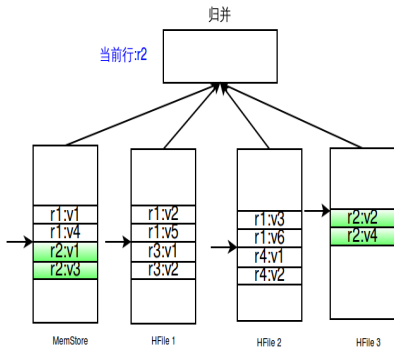




正向scan的基本过程

eg: scan  $[r1, r3]$

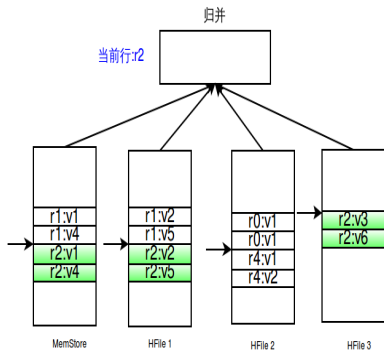
1. 顺序读取当前行数据后，自然读到下一行开头
2. 选取最小行作当前行的“下一个行”，重复上述过程



基本过程:

eg: 反向scan  $[r2, r0]$

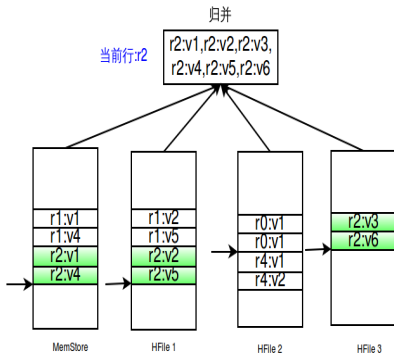
1. 顺序读取完当前行数据后,
2. 构造当前行的最小kv, 对每个HFile, SeekBefore  $\rightarrow$  获取这个最小kv的上一个kv
3. 选取所有kv中最大行作为当前行的“上一个行”, 构造并seek到上一个行的最小kv位置



基本过程:

eg: 反向scan  $[r2, r0]$

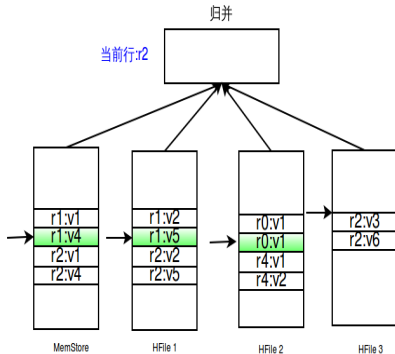
1. 顺序读取完当前行数据后,
2. 构造当前行的最小kv, 对每个HFile, SeekBefore  $\rightarrow$  获取这个最小kv的上一个kv
3. 选取所有kv中最大行作为当前行的“上一个行”, 构造并seek到上一个行的最小kv位置



基本过程:

eg: 反向scan  $[r2, r0]$

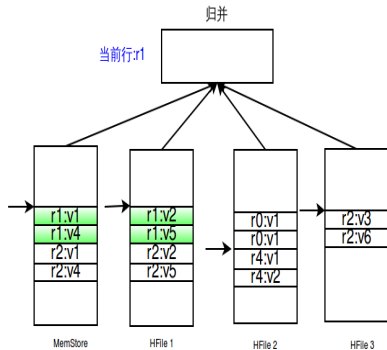
1. 顺序读取完当前行数据后,
2. 构造当前行的最小kv, 对每个HFile, SeekBefore  $\rightarrow$  获取这个最小kv的上一个kv
3. 选取所有kv中最大行作为当前行的“上一个行”, 构造并seek到上一个行的最小kv位置



基本过程:

eg: 反向scan  $[r2, r0]$

1. 顺序读取完当前行数据后,
2. 构造当前行的最小kv, 对每个HFile, SeekBefore  $\rightarrow$  获取这个最小kv的上一个kv
3. 选取所有kv中最大行作为当前行的“上一个行”, 构造并seek到上一个行的最小kv位置



相同点：读取每一行内数据过程，正/反向scan是一致的

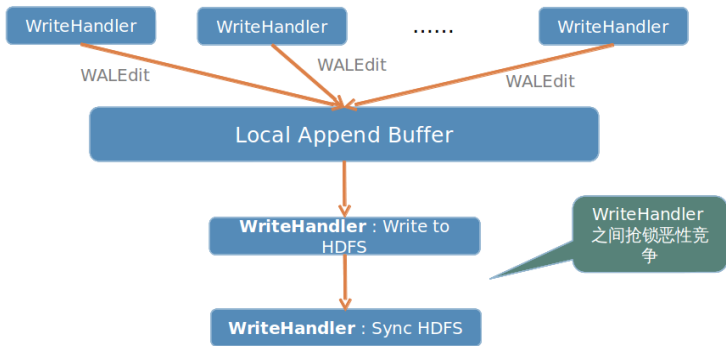
区别：如何当前行的"下一行"

正向scan自然的滑到下一行的开头。

反向scan需要多一次seekBefore和一次seek操作，才能定位到上一行开头。性能损失大约在30%左右

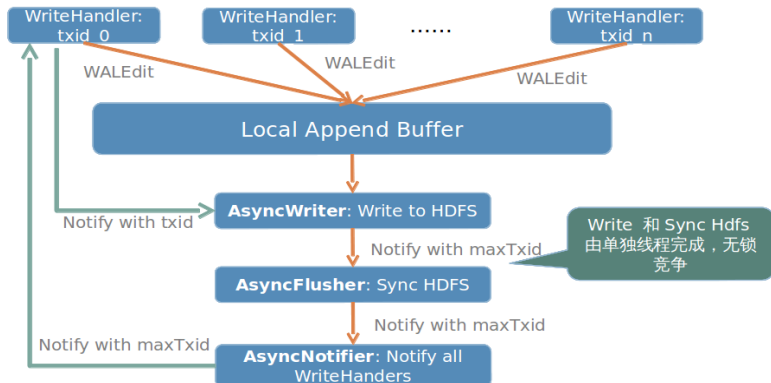
现在的写线程模型

问题： handler线程对sync操作前的锁竞争严重



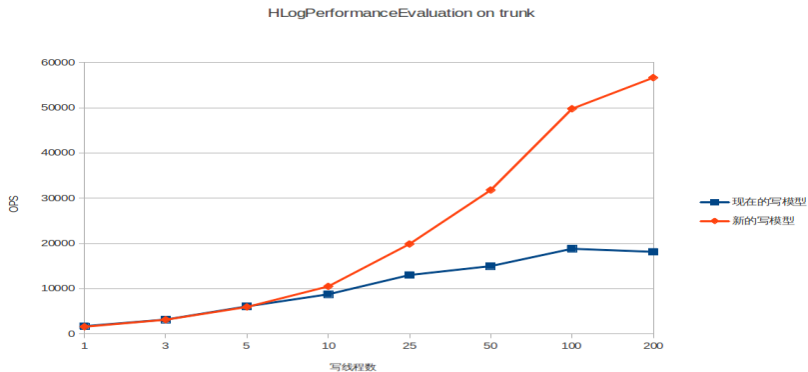
参见：JIRA HBASE-8755

单独的Write/Sync/Notify线程，消除锁竞争





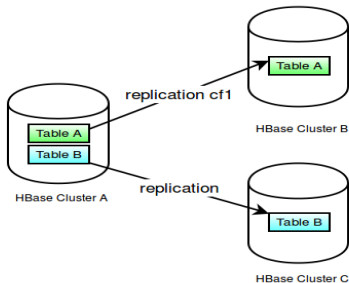
## 性能对比



# 表/列族级别复制

参见: jira HBASE-8751

add\_peer时指定表名和列族名  
eg: add\_peer '1', peer\_addr,  
'tableA:cf1'



1. HBase只支持行Key上的索引
2. 其他的索引需要用户自己建
3. 怎么保证数据和索引之间的一致性?

场景：用户为中心的数据，只需要局部二级索引。

类型	行	列族	列	版本	值
数据行	UserId-xxx	data	c	...	value
索引行	UserId-value	c-index	...	...	UserId-xxx

- ▶ 索引建立：客户端建索引，将数据行和索引行作为一个批量写，写入HBase
- ▶ 查询：先查询索引行获取数据行的Row Key，在根据Row Key查询实际数据

怎么保证数据和索引一致性？

## ► Region分割策略

前缀分割策略(KeyDelimiterPrefixRegionSplitPolicy)

123123-34141 → 123123-

保证同一个用户数据不会被分割在不同的region上

## ► Region内批量写的原子性

修改multi操作，对特定的表的multi操作，取到所有Row的行锁后才能继续，保证所有操作有相同的mvcc，并且写在hlog里面在同一个记录里面。

JIRA状态：即将提交

HBase集群的标识:

zkQuorum:zkClientPort:hbaseZnodeParent

例如: bjsrv-test 集群

bjsrv.hadoop.srv:2181:/hbase/bjsrv-test

扩展集群标识:

hbase://\$zkName-\$hbaseName:\$zkPort

## 映射规则

例如: hbase://bjsrv-test:2181/TestTable

- ▶ bjsrv → bjsrv.hadoop.srv
- ▶ 2181 → 2181, 默认是2181
- ▶ bjsrv-test → /hbase/bjsrv-test

## 使用

```
HTable table = new HTable(conf, "hbase://bjsrv-test/TestTable");
```

## 优势

简化客户端配置, 集群配置对业务透明

## 其他支持

### ▶ HBase Shell

eg: `bin/hbase-cluster hbase://bjsrv-test shell`

### ▶ Replication

eg: `add_peer '1', 'hbase://bjsrv-test', 'TestTable:CF'`

### ▶ Mapreduce

eg: `./hbase CopyTable -peer.adr=hbase://bjsrv-test TestTable`

JIRA状态: 即将提交



1. HBase简介
2. HBase在小米的使用
3. 小米开发的重要特性
4. 未来开发计划

- ▶ 同步复制
- ▶ 跨行跨表的原子性(参考: Google Percolator)
- ▶ 全局二级索引
- ▶ Compaction优化. 参见: JIRA HBASE-9528
- ▶ Failover相关的优化. 参见: JIRA HBASE-9873
- ▶ 多租户共享集群与共有云
- ▶ HMaster重构. 参见: JIRA HBASE-5487
- ▶ ...

- ▶ HBase修改反馈回社区(问题抽象/可配置)
- ▶ 紧跟社区最新进展
- ▶ 积极参与社区方案设计和讨论

# 谢谢！ 问题？

邮箱: liushaohui@xiaomi.com  
微博: lshmouse