

**数据立方与HBase**

**性能对比测试报告**

目 录

一、 概述 4

1.1目的 4

1.2测试内容 4

二、 测试环境 9

2.1测试组网 9

2.2设备配置 9

2.3测试工具 10

2.3.1 Ganglia监控工具 10

2.4测试方法 10

三、 测试用例 11

3.1 数据立方测试用例 11

3.1.1 数据立方可靠性测试 11

3.1.2数据立方入性能 12

3.1.3数据立方查询性能 17

3.2 HBase测试用例 22

3.2.1HBase可靠性测试 22

3.2.2HBase入库性能 23

3.2.3 HBase查询性能测试 27

四、 测试结果 34

4.1测试用例执行情况 34

4.2 性能测试结果 35

4.2.1数据立方与HBase数据入库速率 35

4.2.2数据立方与HBase数据入库对比 37

4.2.3数据立方与HBase查询性能对比 38

五、 测试总结 40

# 概述

## 1.1目的

本次将数据立方(Datacube)与HBase在相同硬件配置、网络环境下，进行性能对比测试，从而验证数据立方在计算处理方面优越于HBase。

## 1.2测试内容

本次将数据立方(Datacube)与HBase进行性能对比测试，测试内容为：产品可靠性、单客户端数据入库速率、多客户端数据入库速率、数据查询性能。

测试的记录结构为:

hbase(main):001:0> describe 'usertable'

DESCRIPTION :

'usertable', {NAME => 'f1', DATA\_BLOCK\_ENCODING => 'NONE', BLOOMFILTER => 'NONE', REPLICATION\_SCOPE => '0', VERSIONS => '3', COMPRESSI true ON => 'NONE', MIN\_VERSIONS => '0', TTL => '2147483647', KEEP\_DELETED\_CELLS => 'false', BLOCKSIZE => '65536', IN\_MEMORY => 'false', ENCODE\_ON\_DISK => 'true', BLOCKCACHE => 'true'}, {NAME => 'f2', DATA\_BLOCK\_ENCODING => 'NONE', BLOOMFILTER => 'NONE', REPLICATION\_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE', MIN\_VERSIONS => '0', TTL => '2147483647', KEEP\_DELETED\_CELLS => 'false', BLOCKSIZE => '65536', IN\_MEMORY => 'false', ENCODE\_ON\_DISK => 'true', BLOCKCACHE => 'true'}, {NAME => 'f3', DATA\_BLOCK\_ENCODING => 'NONE', BLOOMFILTER => 'NONE', REPLICATION\_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE', MIN\_VERSIONS => '0', TTL => '2147483647', KEEP\_DELETED\_CELLS => 'false', BLOCKSIZE => '65536', IN\_MEMORY => 'false', ENCODE\_ON\_DISK => 'true', BLOCKCACHE => 'true'}

数据生成方式:

* HBase数据生成：

dataload的配置方式为:

# Copyright (c) 2010 Yahoo! Inc. All rights reserved.

#

# Licensed under the Apache License, Version 2.0 (the "License"); you

# may not use this file except in compliance with the License. You

# may obtain a copy of the License at

#

# http://www.apache.org/licenses/LICENSE-2.0

#

# Unless required by applicable law or agreed to in writing, software

# distributed under the License is distributed on an "AS IS" BASIS,

# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

# implied. See the License for the specific language governing

# permissions and limitations under the License. See accompanying

# LICENSE file.

# Yahoo! Cloud System Benchmark

# Workload C: Read only

# Application example: user profile cache, where profiles are constructed elsewhere (e.g., Hadoop)

# Read/update ratio: 100/0

# Default data size: 1 KB records (10 fields, 100 bytes each, plus key)

# Request distribution: zipfian

recordcount=5000000

operationcount=5000000

workload=com.yahoo.ycsb.workloads.CoreWorkload

readallfields=true

readproportion=1

updateproportion=0

scanproportion=0

insertproportion=0

requestdistribution=zipfian

HBase数据生成方式如下:

java -cp build/ycsb.jar:db/hbase/lib/\* com.yahoo.ycsb.Client -p columnfamily=f1 -db com.yahoo.ycsb.db.HBaseClient -P workloads/dataload -threads 1 -load

* 数据立方（Datacube）数据生成：

public void selectScanFilter(String usertable, String[] args) throws IOException {

for (int i = 0; i < args.length; i=i+2)

{

HTable table = new HTable(configuration, usertable);

Scan scan = new Scan();// 新建一个scan

scan.setStartRow(Bytes.toBytes(args[i]));

scan.setStopRow(Bytes.toBytes(args[i+1]));

ResultScanner ResultScannerFilterList = table.getScanner(scan);

FileOutputStream fos = null;

BufferedWriter bw = null;

fos = new FileOutputStream(file,true);

bw = new BufferedWriter(new OutputStreamWriter(fos));

int j =0;

for (Result rs = ResultScannerFilterList.next(); rs != null; rs = ResultScannerFilterList

.next())

{

StringBuilder sBuilder = new StringBuilder();

for (KeyValue kv : rs.list())

{

sBuilder.append(new

String(kv.getRow())).append(",").append(new

String(kv.getFamily()))

.append(",").append(new

String(kv.getQualifier())).append(",")

.append(new String(kv.getValue())).append("\n");

}

bw.write(sBuilder.toString());

if (j%1000==0)

{

bw.flush();

}

j++;

bw.newLine();

}

ResultScannerFilterList.close();

# 测试环境

## 2.1测试组网

本次测试，数据立方、HBase在相同的硬件环境上进行测试

## 2.2设备配置

**软件配置**

|  |  |  |  |
| --- | --- | --- | --- |
| 软件名称 | 软件版本 | 数量（套） | 说明 |
| Datacube | 1.0版本 | 1 |  |
| HBase | HBase-0.94.7 | 1 |  |

**硬件配置**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 设备名称 | 数量 | CPU | 内存 | 硬盘 | 说明 |
| 1 | 主控服务器 | 2 | Intel(R) Xeon(R) CPU E5606 @ 2.13GHz | 32G | 2\*3T |  |
| 2 | 处理节点 | 7 | Intel(R) Xeon(R) CPU E5606 @ 2.13GHz | 32G | 2\*3T |  |

**网络配置**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 设备名称 | 设备型号 | 数量 |
| 1 | 千兆交换机 | SD2008T | 1 |
| 2 | 千兆连接口 | 10/100/1000BASE-T口 | 24 |

## 2.3测试工具

### 2.3.1 Ganglia监控工具

Ganglia是设计用于检测数以千计的节点。Ganglia的核心包含gmond、gmetad以及一个Web前端。主要是用来监控系统性能，如：cpu 、mem、硬盘利用率， I/O负载、网络流量情况等，通过曲线很容易见到每个节点的工作状态，对合理调整、分配[系统资源](http://baike.baidu.com/view/53557.htm)，提高系统整体性能起到重要作用。

## 2.4测试方法

HBase测试是采用YCSB benchmark测试的，HBase入库数据量：5000万条、1亿条、5亿条、10亿条、20亿、40亿、80亿、100亿条；HBase数据查询是采用测试代码实现的，本次HBase查询是以行键+列族+列名进行数据查询的。

数据立方（Datacube）数据入库分别将HBase中的5000万、1亿条、5亿条、10亿条、20亿、40亿、80亿、100亿条数据，以文本格式导入到数据立方hdfs中的。数据立方中的查询条件与HBase中的查询条件相同。

# 测试用例

## 3.1 数据立方测试用例

### 3.1.1 数据立方可靠性测试

#### 3.1.1.1 Primary（AvatarNode1）节点故障

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 主namenode宕机 |
| 用例编号 | Datacube-fun-001 | 重要性 | 重要 |
| 测试目的 | 验证主namenode宕机后, Standby（AvatarNode2）节点是否能正常转换为主节点，并且系统稳定运行 | | |
| 预置条件 | 1. Datacube系统运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端向数据立方写数据  2、写数据过程中，构造主节点服务器故障：重启（reboot）、网络异常、掉电、服务关闭  3、客户端继续写入数据  4、检测写入的数据是否丢失 | | |
| 预期结果 | 1、AvatarNode2自动切换为Primary，且系统稳定。切换完成时间少于10s  2、数据写入成功  3、切换后写入的数据无丢失 | | |
| 备注 |  | | |

#### 3.1.1.2 datanode节点故障

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 写数据过程中，datanode节点宕机 |
| 用例编号 | Datacube-fun-002 | 重要性 | 重要 |
| 测试目的 | 验证客户端写入数据过程中，将datanode故障情况下，测试写入的数据是否成功 | | |
| 预置条件 | 1. Datacube系统运行正常 2. 客户端运行正常 3. 设置副本数为2 | | |
| 测试步骤 | 1、客户端向数据立方写数据  2、写数据过程中，构造datanode节点服务器故障：重启（reboot）、网络异常、掉电、服务关闭  3、客户端继续写入数据  4、检测数据写入是否成功 | | |
| 预期结果 | 写数据过程中，在机器宕机的那一瞬间写入的某个文件写失败，之后数据写入成功 | | |
| 备注 |  | | |

### 3.1.2数据立方入性能

#### 3.1.2.1 单客户端数据入库

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 单个客户端数据入库性能测试 |
| 用例编号 | Datacube-pre-001 | 重要性 | 重要 |
| 测试目的 | 验证单个客户端向数据立方写5亿条数据时的带宽 | | |
| 预置条件 | 1. Datacube系统运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、启用单个客户端将HBASE中的5亿条数据写入到数据立方  2、启动ganglia监控工具  3、记录5亿条数据入库速率 | | |
| 预期结果 | 1、数据入库正确无误  2、数据入库速率正常 | | |
| 备注 |  | | |

#### 3.1.2.2 多个客户端数据入库速率

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 多个客户端数据入库性能测试 |
| 用例编号 | Datacube-pre-002 | 重要性 | 重要 |
| 测试目的 | 验证多个客户端向数据立方写5亿条数据，通过ganglia工具，监控多个客户端数据入库速率 | | |
| 预置条件 | 1. Datacube系统运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、启用多个客户（不同服务器），将HBase中写入的5亿条数据并发写入数据立方  2、启动ganglia监控工具  3、记录数据库入库速率 | | |
| 预期结果 | 1、数据入库正确无误  2、多个客户端数据入库速率正常 | | |
| 备注 |  | | |

#### 3.1.2.3 5000万条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 数据立方5000万记录入库测试 |
| 用例编号 | Datacube-pre-003 | 重要性 | 重要 |
| 测试目的 | 测试统计5000万记录写到数据立方中所用的时长 | | |
| 预置条件 | 1. 数据立方系统运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端运行测试程序将HBase中写入的5000万条记录写到数据立方中  2、记录5000万记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、5000万条记录入库时长正常 | | |
| 备注 |  | | |

#### 3.1.2.4 1亿条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 数据立方1亿条记录入库测试 |
| 用例编号 | Datacube-pre-004 | 重要性 | 重要 |
| 测试目的 | 测试统计1亿条记录写到数据立方中所用的时长 | | |
| 预置条件 | 1. 数据立方系统运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端运行测试程序将HBase中写入的1亿条记录写到数据立方中  2、记录1亿条记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、1亿条记录入库时长正常 | | |
| 备注 |  | | |

#### 3.1.2.5 5亿条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 数据立方5亿条记录入库测试 |
| 用例编号 | Datacube-pre-005 | 重要性 | 重要 |
| 测试目的 | 测试统计5亿条记录写到数据立方中所用的时长 | | |
| 预置条件 | 1. 数据立方系统运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端运行测试程序将HBase中写入的5亿条记录写到数据立方中  2、记录5亿条记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、5亿条记录入库时长正常 | | |
| 备注 |  | | |

#### 3.1.2.6 10亿条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 数据立方10亿条记录入库测试 |
| 用例编号 | Datacube-pre-006 | 重要性 | 重要 |
| 测试目的 | 测试统计10亿条数据写到数据立方中所用的时长 | | |
| 预置条件 | 1. 数据立方系统运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端运行测试程序将HBase中写入的10亿条记录写到数据立方中  2、记录10亿条记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、10亿条记录入库时长正常 | | |
| 备注 |  | | |

#### 3.1.2.7 20亿条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 数据立方20亿条记录入库测试 |
| 用例编号 | Datacube-pre-007 | 重要性 | 重要 |
| 测试目的 | 测试统计20亿条记录写到数据立方中所用的时长 | | |
| 预置条件 | 1. 数据立方系统运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端运行测试程序将HBase中写入的20亿条记录写到数据立方中  2、记录20亿条记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、20亿条记录入库时长正常 | | |
| 备注 |  | | |

#### 3.1.2.8 40亿条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 数据立方40亿条记录入库测试 |
| 用例编号 | Datacube-pre-008 | 重要性 | 重要 |
| 测试目的 | 测试统计40亿条记录写到数据立方中所用的时长 | | |
| 预置条件 | 1. 数据立方系统运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端运行测试程序将HBase中写入的40亿条记录写到数据立方中  2、记录40亿条记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、40亿条记录入库时长正常 | | |
| 备注 |  | | |

#### 3.1.2.9 80亿条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 数据立方80亿条记录入库测试 |
| 用例编号 | Datacube-pre-009 | 重要性 | 重要 |
| 测试目的 | 测试统计80亿条记录写到数据立方中所用的时长 | | |
| 预置条件 | 1. 数据立方系统运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端运行测试程序将HBase中写入的80亿条记录写到数据立方中  2、记录80亿条记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、80亿条记录入库时长正常 | | |
| 备注 |  | | |

#### 3.1.2.10 100亿记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 数据立方100亿条数据入库测试 |
| 用例编号 | Datacube-pre-010 | 重要性 | 重要 |
| 测试目的 | 测试统计100亿条记录写到数据立方中所用的时长 | | |
| 预置条件 | 1. 数据立方系统运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端运行测试程序将HBase中写入的100亿条记录写到数据立方中  2、记录100亿条记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、100亿条记录入库时长正常 | | |
| 备注 |  | | |

### 3.1.3数据立方查询性能

#### 3.1.3.1 5000万条记录中查询10条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 5000万记录查询10条数据 |
| 用例编号 | Datacube-pre-011 | 重要性 | 重要 |
| 测试目的 | 5000万记录中查询10条记录，查询10条记录正确、时间正常 | | |
| 预置条件 | 1. Datacube系统运行正常 2. 查询框架运行正常 3. ZK启动正常 | | |
| 测试步骤 | 1、客户端向数据立方写入5000万条记录  2、客户端发送记录查询请求，请求查询10条记录  3、记录查询时长 | | |
| 预期结果 | 1、查询结果正确  2、查询时间正常 | | |
| 备注 | 数据立方测试5000万条记录中查询10条记录与HBase中的查询条件一致 | | |

#### 3.1.3.2 1亿条记录中查询1条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 1亿条记录中查询1条记录 |
| 用例编号 | Datacube-pre-012 | 重要性 | 重要 |
| 测试目的 | 1亿条记录中查询1条记录，查询1条记录正确、时间正常 | | |
| 预置条件 | 1. Datacube系统运行正常 2. 查询框架运行正常 3. ZK启动正常 | | |
| 测试步骤 | 1、客户端写入1亿条记录  2、发送记录查询请求，请求查询1条记录  3、记录1条记录查询时长 | | |
| 预期结果 | 1、查询结果正确  2、查询时间正常 | | |
| 备注 | 数据立方测试1亿条记录中查询1条记录的条件与HBase中的查询条件一致 | | |

#### 3.1.3.3 1亿条记录中查询10条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 1亿条记录中查询10条记录 |
| 用例编号 | Datacube-pre-013 | 重要性 | 重要 |
| 测试目的 | 1亿条记录中查询10条记录，验证查询10条记录正确、时间正常 | | |
| 预置条件 | 1. Datacube系统运行正常 2. 查询框架运行正常 3. ZK启动正常 | | |
| 测试步骤 | 1、客户端向数据立方写入1亿条记录  2、发送记录查询请求，请求中查询10条记录  3、记录10条记录查询时长 | | |
| 预期结果 | 1、查询结果正确  2、查询时间正常 | | |
| 备注 | 数据立方测试1亿条记录中查询10条记录与HBase中的查询条件一致 | | |

#### 3.1.3.4 5亿条记录中查询1条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 1亿条记录中查询1条记录 |
| 用例编号 | Datacube-pre-014 | 重要性 | 重要 |
| 测试目的 | 5亿条记录中查询1条记录，验证查询1条记录正确、时间正常 | | |
| 预置条件 | 1. Datacube系统运行正常 2. 查询框架运行正常 3. ZK启动正常 | | |
| 测试步骤 | 1、客户端写入5亿条记录  2、发送记录查询请求：查询1条记录  3、记录查询1条记录时长 | | |
| 预期结果 | 1、查询结果正确  2、查询时间正常 | | |
| 备注 | 数据立方测试查询5亿条记录中的1条记录与HBase中的查询条件一致 | | |

#### 3.1.3.5 5亿条记录中查询10条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 5亿条记录中查询10条记录 |
| 用例编号 | Datacube-pre-015 | 重要性 | 重要 |
| 测试目的 | 5亿条记录中查询10条记录，验证查询10条记录正确、时间正常 | | |
| 预置条件 | 1. Datacube系统运行正常 2. 查询框架运行正常 3. ZK启动正常 | | |
| 测试步骤 | 1、客户端向数据立方写入5亿条记录  2、发送记录查询请求：请求查询10条记录  3、记录查询10条记录时长 | | |
| 预期结果 | 1、查询结果正确  2、查询时间正常 | | |
| 备注 | 数据立方测试查询5亿条记录中的10条记录与HBase中的查询条件一致 | | |

#### 3.1.3.6 10亿条记录中查询1000条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 10亿条记录中查询1000条记录 |
| 用例编号 | Datacube-pre-016 | 重要性 | 重要 |
| 测试目的 | 查询10亿条记录中的1000条记录，查询1000条记录正确、时间正常 | | |
| 预置条件 | 1. 数据立方运行正常 2. 客户端运行正常 3. ZK运行正常 | | |
| 测试步骤 | 1、客户端将HBase中的10亿条记录写入到数据立方  2、运行测试程序发送查询1000条记录请求:查询1000条记录  3、记录查询1000条记录时长 | | |
| 预期结果 | 1、1000条查询结果正确  2、1000条记录查询时间正常 | | |
| 备注 | 数据立方测试查询10亿条记录中的1000条记录与HBase中的查询条件一致 | | |

#### 3.1.3.7 20亿条记录中查询1000条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 10亿条记录中查询1000条记录 |
| 用例编号 | Datacube-pre-017 | 重要性 | 重要 |
| 测试目的 | 查询20亿条记录中的1000条记录，查询1000条记录正确、时间正常 | | |
| 预置条件 | 1. 数据立方运行正常 2. 客户端运行正常 3. ZK运行正常 | | |
| 测试步骤 | 1、客户端将HBase中的20亿条记录写入到数据立方系统中  2、运行测试程序发送查询1000条记录请求:查询1000条记录  3、记录查询1000条记录时长 | | |
| 预期结果 | 1、1000条查询结果正确  2、1000条记录查询时间正常 | | |
| 备注 | 数据立方测试查询20亿条记录中的1000条记录与HBase中的查询条件一致 | | |

#### 3.1.3.8 40亿条记录中查询1000记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 40亿条记录中查询1000条记录 |
| 用例编号 | Datacube-pre-018 | 重要性 | 重要 |
| 测试目的 | 查询40亿条记录中的1000条记录，查询1000条记录正确、时间正常 | | |
| 预置条件 | 1. 数据立方运行正常 2. 客户端运行正常 3. ZK运行正常 | | |
| 测试步骤 | 1、客户端将HBase中的40亿条记录写入到数据立方系统中  2、运行测试程序发送查询1000条记录请求: 查询1000条记录  3、记录查询1000条记录时长 | | |
| 预期结果 | 1、1000条记录查询结果正确  2、1000条记录查询时间正常 | | |
| 备注 | 数据立方测试查询40亿条记录中的1000条记录与HBase中的查询条件一致 | | |

#### 3.1.3.9 80亿条记录中查询1000条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 80亿条记录中查询1000条记录 |
| 用例编号 | Datacube-pre-019 | 重要性 | 重要 |
| 测试目的 | 查询80亿条记录中的1000条记录，查询1000条记录正确、时间正常 | | |
| 预置条件 | 1. 数据立方运行正常 2. 客户端运行正常 3. ZK运行正常 | | |
| 测试步骤 | 1、客户端将HBase中的80亿条记录写入到数据立方系统中  2、运行测试程序发送查询1000条记录请求: 查询1000条记录  3、记录查询1000条记录时长 | | |
| 预期结果 | 1、1000条记录查询结果正确  2、1000条记录查询时间正常 | | |
| 备注 | 数据立方测试查询80亿条记录中的1000条记录与HBase中的查询条件一致 | | |

#### 3.1.3.10 100亿条记录中查询1000条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 数据立方 | 用例名称 | 100亿条记录中查询1000条记录 |
| 用例编号 | Datacube-pre-020 | 重要性 | 重要 |
| 测试目的 | 查询100亿条记录中的1000条记录，查询1000条记录正确、时间正常 | | |
| 预置条件 | 1. 数据立方运行正常 2. 客户端运行正常 3. ZK运行正常 | | |
| 测试步骤 | 1、客户端将HBase中的100亿条记录写入到数据立方系统中  2、运行测试程序发送查询1000条记录请求: 查询1000条记录  3、记录查询1000条记录时长 | | |
| 预期结果 | 1、1000条记录查询结果正确  2、1000条记录查询时间正常 | | |
| 备注 | 数据立方测试查询100亿条记录中的1000条记录与HBase中的查询条件一致 | | |

## 3.2 HBase测试用例

### 3.2.1HBase可靠性测试

#### 3.2.1.1 Primary（AvatarNode1）节点故障

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | 主namenode宕机 |
| 用例编号 | HBase-fun-001 | 重要性 | 重要 |
| 测试目的 | 验证主namenode宕机后, 备namenode节点是否能正常转换为主节点，并且系统稳定运行 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端向HBase写数据  2、写数据过程中，构造主节点服务器故障：重启（reboot）、网络异常、掉电、服务关闭  3、检测写入的数据是否丢失 | | |
| 预期结果 | 1、AvatarNode2自动切换为Primary，且系统稳定。切换完成时间少于10s  2、数据写入成功  3、切换后写入的数据无丢失 | | |
| 备注 |  | | |

#### 3.2.1.2 datanode节点故障

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | 写数据过程中，datanode节点宕机 |
| 用例编号 | HBase-fun-002 | 重要性 | 重要 |
| 测试目的 | 验证客户端向HBase写入数据过程中，将datanode故障情况下，测试写入的数据是否成功 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 3. 设置副本数为2 | | |
| 测试步骤 | 1、客户端向HBase写数据  2、写数据过程中，构造datanode节点服务器故障：重启（reboot）、网络异常、掉电、服务关闭  3、检测数据写入是否成功 | | |
| 预期结果 | 写数据过程中，在机器宕机的那一瞬间写入的某个文件写失败，之后的数据写入成功 | | |
| 备注 |  | | |

### 3.2.2HBase入库性能

#### 3.2.2.1 单客户端数据入库

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | 单个客户端入库性能测试 |
| 用例编号 | HBase-pre-001 | 重要性 | 重要 |
| 测试目的 | 验证单个客户端向HBase中写数据，通过ganglia监控工具，获知单个客户端数据入库带宽 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、启用单个客户端向HBase连续写入5亿条数据  2、启用ganglia监控程序  3、记录数据入库速率 | | |
| 预期结果 | 1、数据入库正确无误  2、数据入库速率正常 | | |
| 备注 |  | | |

#### 3.2.2.2 多客户端数据入库速率

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | 多个客户端入库性能测试 |
| 用例编号 | HBase-pre-002 | 重要性 | 重要 |
| 测试目的 | 验证多个客户端向HBase中写数据，通过ganglia监控工具，获知多个客户端数据入库带宽 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、启用多个客户（不同服务器）并发向HBase写5亿条数据  2、启用ganglia系统监控程序  3、记录数据库入库速率 | | |
| 预期结果 | 1、数据入库正确无误  2、多个客户端数据入库速率正常 | | |
| 备注 |  | | |

#### 3.2.2.3 5000万条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | HBase5000万条记录入库测试 |
| 用例编号 | HBase-pre-003 | 重要性 | 重要 |
| 测试目的 | 测试统计5000万条记录写到HBase中所用的时长 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端通过Benchmark向HBase写入5000万记录  2、记录5000万条记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、5000万条记录入库时长正常 | | |
| 备注 |  | | |

#### 3.2.2.4 1亿条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | HBase中1亿条记录入库测试 |
| 用例编号 | HBase-pre-004 | 重要性 | 重要 |
| 测试目的 | 测试统计1亿条记录写到HBase中所用的时长 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端通过Benchmark向HBase写入1亿条记录  2、记录1亿条记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、1亿条记录入库时长正常 | | |
| 备注 |  | | |

#### 3.2.2.5 5亿条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | HBase中5亿条记录入库测试 |
| 用例编号 | HBase-pre-005 | 重要性 | 重要 |
| 测试目的 | 测试统计5亿条记录写到HBase中所用的时长 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端通过Benchmark向HBase写入5亿条记录  2、记录5亿条记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、5亿条记录入库时长正常 | | |
| 备注 |  | | |

#### 3.2.2.6 10亿条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | HBase中10亿条记录入库测试 |
| 用例编号 | HBase-pre-006 | 重要性 | 重要 |
| 测试目的 | 测试统计10亿条记录写到HBase中所用的时长 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端通过Benchmark向HBase写入10亿条记录  2、记录10亿条记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、10亿条记录入库时长正常 | | |
| 备注 |  | | |

#### 3.2.2.7 20亿条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | HBase中20亿条记录入库测试 |
| 用例编号 | HBase-pre-007 | 重要性 | 重要 |
| 测试目的 | 测试统计20亿条记录写到HBase中所用的时长 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端通过Benchmark向HBase写入20亿条记录  2、记录20亿条记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、20亿条记录入库时长正常 | | |
| 备注 |  | | |

#### 3.2.2.8 40亿条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | HBase中40亿条记录入库测试 |
| 用例编号 | HBase-pre-008 | 重要性 | 重要 |
| 测试目的 | 测试统计40亿条记录写到HBase中所用的时长 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端通过Benchmark向HBase写入40亿条记录  2、记录40亿条记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、40亿条记录入库时长正常 | | |
| 备注 |  | | |

#### 3.2.2.9 80亿条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | HBase中80亿条记录入库测试 |
| 用例编号 | HBase-pre-009 | 重要性 | 重要 |
| 测试目的 | 测试统计80亿条记录写到HBase中所用的时长 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端通过Benchmark向HBase写入80亿条记录  2、记录80亿条记录入库时长 | | |
| 预期结果 | 1、数据入库正确  2、80亿条记录入库时长正常 | | |
| 备注 |  | | |

#### 3.2.2.10 100亿条记录入库测试

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | HBase中100亿条记录入库测试 |
| 用例编号 | HBase-pre-010 | 重要性 | 重要 |
| 测试目的 | 测试统计100亿条记录写到HBase中所用的时长 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端通过Benchmark向HBase写入100亿条记录  2、记录100亿条记录入库时长 | | |
| 预期结果 | 1、记录入库正确  2、100亿条记录入库时长正常 | | |
| 备注 |  | | |

### 3.2.3 HBase查询性能测试

#### 3.2.3.1 5000万记录中查询10条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | 从5000万记录中查询10条记录 |
| 用例编号 | HBase-pre-011 | 重要性 | 重要 |
| 测试目的 | 查询HBase的5000万条记录中的10条记录，查询10条记录正确、时间正常 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端向HBase写入5000万条记录，其中1条命中  2、通过测试程序发送查询记录请求:  selectRowKeyFamilyColumn("usertable", "user1000032235", "f1", "field0");  selectRowKeyFamilyColumn("usertable", " user1000032235", "f1", "field1");  selectRowKeyFamilyColumn("usertable", " user1000032235", "f1", "field2");  selectRowKeyFamilyColumn("usertable", " user1000032235", "f1", "field3");  selectRowKeyFamilyColumn("usertable", " user1000032235", "f1", "field4");  selectRowKeyFamilyColumn("usertable", " user1000032235", "f1", "field5");  selectRowKeyFamilyColumn("usertable", " user1000032235", "f1", "field6");  selectRowKeyFamilyColumn("usertable", " user1000032235", "f1", "field7");  selectRowKeyFamilyColumn("usertable", " user1000032235", "f1", "field8");  selectRowKeyFamilyColumn("usertable", " user1000032235", "f1", "field9");  3、记录查询时长 | | |
| 预期结果 | 1、查询结果正确  2、查询时间正常 | | |
| 备注 |  | | |

#### 3.2.3.2 1亿条记录中查询1条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | 1亿条记录中查询1条记录 |
| 用例编号 | HBase-pre-012 | 重要性 | 重要 |
| 测试目的 | 查询HBase的1亿条记录中的1条记录，查询1条记录正确、时间正常 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端向HBase写入1亿条记录  2、通过测试程序发送查询记录请求:  selectRowKeyFamilyColumn("usertable","user1000022285", "f1", "field0");  3、记录查询时长 | | |
| 预期结果 | 1、查询结果正确  2、查询时间正常 | | |
| 备注 |  | | |

#### 3.2.3.3 1亿条记录中查询10条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | 1亿条记录中查询10条记录 |
| 用例编号 | HBase-pre-013 | 重要性 | 重要 |
| 测试目的 | 查询HBase的1亿条记录中的10条记录，查询10条记录正确、时间正常 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 | | |
| 测试步骤 | 1、客户端向HBase写入1亿条记录  2、发送数据查询10条记录请求:  selectRowKeyFamilyColumn("usertable", "user1000022285", "f1", "field0");  selectRowKeyFamilyColumn("usertable", " user1000022285", "f1", "field1");  selectRowKeyFamilyColumn("usertable", " user1000022285", "f1", "field2");  selectRowKeyFamilyColumn("usertable", " user1000022285", "f1", "field3");  selectRowKeyFamilyColumn("usertable", " user1000022285", "f1", "field4");  selectRowKeyFamilyColumn("usertable", " user1000022285", "f1", "field5");  selectRowKeyFamilyColumn("usertable", " user1000022285", "f1", "field6");  selectRowKeyFamilyColumn("usertable", " user1000022285", "f1", "field7");  selectRowKeyFamilyColumn("usertable", " user1000022285", "f1", "field8");  selectRowKeyFamilyColumn("usertable", " user1000022285", "f1", "field9");  3、记录查询10条记录时长 | | |
| 预期结果 | 1、查询结果正确  2、查询时间正常 | | |
| 备注 |  | | |

#### 3.2.3.4 5亿条记录中查询1条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | 1亿条记录中查询1条记录 |
| 用例编号 | HBase-pre-014 | 重要性 | 重要 |
| 测试目的 | 查询HBase的5亿条记录中的1条记录，查询1条记录正确、时间正常 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 3. 测试程序运行正常 | | |
| 测试步骤 | 1、客户端向HBase写入5亿条记录  2、通过测试程序发送查询1条记录请求:  selectRowKeyFamilyColumn("usertable", "user1000062287", "f1", "field0");  3、记录查询1条记录时长 | | |
| 预期结果 | 1、查询结果正确  2、查询时间正常 | | |
| 备注 |  | | |

#### 3.2.3.5 5亿条记录中查询10条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | 5亿条记录中查询10条记录 |
| 用例编号 | HBase-pre-015 | 重要性 | 重要 |
| 测试目的 | 查询HBase的5亿条记录中的10条记录，查询10条记录正确、时间正常 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 3. 测试代码运行正常 | | |
| 测试步骤 | 1、客户端向HBase写入5亿条记录  2、运行测试程序发送查询10条记录请求:  selectRowKeyFamilyColumn("usertable", "user1000062287", "f1", "field0");  selectRowKeyFamilyColumn("usertable", "user1000062287", "f1", "field1");  selectRowKeyFamilyColumn("usertable", "user1000062287", "f1", "field2");  selectRowKeyFamilyColumn("usertable", "user1000062287", "f1", "field3");  selectRowKeyFamilyColumn("usertable", "user1000062287", "f1", "field4");  selectRowKeyFamilyColumn("usertable", "user1000062287", "f1", "field5");  selectRowKeyFamilyColumn("usertable", "user1000062287", "f1", "field6");  selectRowKeyFamilyColumn("usertable", "user1000062287", "f1", "field7");  selectRowKeyFamilyColumn("usertable", "user1000062287", "f1", "field8");  selectRowKeyFamilyColumn("usertable", "user1000062287", "f1", "field9");  3、记录查询10条记录时长 | | |
| 预期结果 | 1、查询结果正确  2、查询时间正常 | | |
| 备注 |  | | |

#### 3.2.3.6 10亿条记录中查询1000条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | 10亿条记录中查询1000条记录 |
| 用例编号 | HBase-pre-016 | 重要性 | 重要 |
| 测试目的 | 查询HBase的10亿条记录中的1000条记录，查询1000条记录正确、时间正常 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 3. 测试代码运行正常 | | |
| 测试步骤 | 1、客户端向HBase写入10亿条记录  2、运行测试程序发送查询1000条记录请求:  Select (1000个RowKey)  3、记录查询1000条记录时长 | | |
| 预期结果 | 1、1000条记录查询结果正确  2、1000条记录查询时间正常 | | |
| 备注 |  | | |

#### 3.2.3.7 20亿条记录中查询1000条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | 20亿条记录中查询1000条记录 |
| 用例编号 | HBase-pre-017 | 重要性 | 重要 |
| 测试目的 | 查询HBase的20亿条记录中的1000条记录，查询1000条记录正确、时间正常 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 3. 测试代码运行正常 | | |
| 测试步骤 | 1、客户端向HBase写入20亿条记录  2、运行测试程序发送查询1000条记录请求:  Select (1000个RowKey)  3、记录查询1000条记录时长 | | |
| 预期结果 | 1、1000条记录查询结果正确  2、记下1000条记录查询时间 | | |
| 备注 |  | | |

#### 3.2.3.8 40亿条记录中查询1000条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | 40亿条记录中查询1000条记录 |
| 用例编号 | HBase-pre-018 | 重要性 | 重要 |
| 测试目的 | 查询HBase的40亿条记录中的1000条记录，查询1000条记录正确、时间正常 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 3. 测试代码运行正常 | | |
| 测试步骤 | 1、客户端向HBase写入40亿条记录  2、运行测试程序发送查询1000条记录请求:  Select (1000个RowKey)  3、记录查询1000条记录时长 | | |
| 预期结果 | 1、1000条查询结果正确  2、统计1000条记录查询时间 | | |
| 备注 |  | | |

#### 3.2.3.9 80亿条记录中查询1000条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | 80亿条记录中查询1000条记录 |
| 用例编号 | HBase-pre-019 | 重要性 | 重要 |
| 测试目的 | 查询HBase的80亿条记录中的1000条记录，查询1000条记录正确、时间正常 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 3. 测试代码运行正常 | | |
| 测试步骤 | 1、客户端向HBase写入80亿条记录  2、运行测试程序发送查询1000条记录请求:  Select (1000个RowKey)  3、记录查询1000条记录时长 | | |
| 预期结果 | 1、1000条记录查询结果正确  2、统计1000条记录查询时间 | | |
| 备注 |  | | |

#### 3.2.3.10 100亿条记录中查询1000条记录

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | HBase | 用例名称 | 100亿条记录中查询1000条记录 |
| 用例编号 | HBase-pre-020 | 重要性 | 重要 |
| 测试目的 | 查询HBase的100亿条记录中的1000条记录，查询1000条记录正确、时间正常 | | |
| 预置条件 | 1. HBase运行正常 2. 客户端运行正常 3. 测试代码运行正常 | | |
| 测试步骤 | 1、客户端向HBase写入100亿条记录  2、运行测试程序发送查询1000条记录请求:  Select (1000个RowKey)  3、记录查询1000条记录时长 | | |
| 预期结果 | 1、1000条记录查询结果正确  2、统计1000条记录查询时间 | | |
| 备注 |  | | |

# 测试结果

## 4.1测试用例执行情况

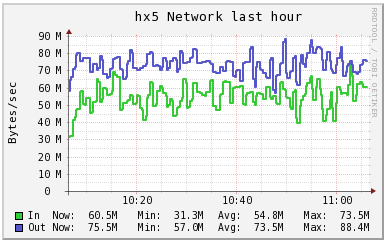
|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **用例章节** | **用例名称** | **测试结果** |
| 1 | 3.1 | 数据立方 |  |
| 2 | 3.1.1 | 数据立方可靠性测试 |  |
| 3 | 3.1.1.1 | Primary（AvatarNode1）节点故障 | 数据立方主节点故障时，系统切换时间为5s |
| 4 | 3.1.1.2 | datanode节点故障 | 测试通过 |
| 5 | 3.1.2 | 数据立方入库速率 |  |
| 6 | 3.1.2.1 | 单客户端入库速率 | 测试通过 |
| 7 | 3.1.2.2 | 多客户端并发入库速率 | 测试通过 |
| 8 | 3.1.2.3 | 5000万条记录入库测试 | 测试通过 |
| 9 | 3.1.2.4 | 1亿条记录入库测试 | 测试通过 |
| 10 | 3.1.2.5 | 5亿条记录入库测试 | 测试通过 |
| 11 | 3.1.2.6 | 10亿条记录入库测试 | 测试通过 |
| 12 | 3.1.2.7 | 20亿条记录入库测试 | 测试通过 |
| 13 | 3.1.2.8 | 40亿条记录入库测试 | 测试通过 |
| 14 | 3.1.2.9 | 80亿条记录入库测试 | 测试通过 |
| 15 | 3.1.2.10 | 100亿条记录入库测试 | 测试通过 |
| 16 | 3.1.3 | 数据立方查询性能 |  |
| 17 | 3.1.3.1 | 5000万条记录中查询1条记录 | 测试通过 |
| 18 | 3.1.3.2 | 1亿条记录中查询1条记录 | 测试通过 |
| 19 | 3.1.3.3 | 1亿条记录中查询10条记录 | 测试通过 |
| 20 | 3.1.3.4 | 5亿条记录中查询1条记录 | 测试通过 |
| 21 | 3.1.3.5 | 5亿条记录中查询10条记录 | 测试通过 |
| 22 | 3.1.3.7 | 10亿条记录中查询1000条记录 | 测试通过 |
| 23 | 3.1.3.8 | 20亿条记录中查询1000条记录 | 测试通过 |
| 24 | 3.1.3.9 | 40亿条记录中查询1000条记录 | 测试通过 |
| 25 | 3.1.3.10 | 80亿条记录中查询1000条记录 | 测试通过 |
| 26 | 3.1.3.11 | 100亿条记录中查询1000条记录 | 测试通过 |
| 27 | 3.2 | HBase |  |
| 28 | 3.2.1 | HBase可靠性 |  |
| 29 | 3.2.1.1 | Primary（AvatarNode1）节点故障 | HBase主节点故障时，主、备节点切换时间为10s |
| 30 | 3.2.1.2 | datanode节点故障 | 测试通过 |
| 31 | 3.2.2 | HBase入库速率 |  |
| 32 | 3.2.2.1 | 单客户端数据入库 | 测试通过 |
| 33 | 3.2.2.2 | 多客户端数据入库速率 | 测试通过 |
| 34 | 3.2.2.3 | 5000万条记录入库测试 | 测试通过 |
| 35 | 3.2.2.4 | 1亿条记录入库测试 | 测试通过 |
| 36 | 3.2.2.5 | 5亿条记录入库测试 | 测试通过 |
| 37 | 3.2.2.6 | 10亿条记录入库测试 | 测试通过 |
| 38 | 3.2.2.7 | 20亿条万记录入库测试 | 测试通过 |
| 39 | 3.2.2.8 | 40亿万记录入库测试 | 测试通过 |
| 40 | 3.2.2.9 | 80亿万记录入库测试 | 测试通过 |
| 41 | 3.2.2.10 | 100亿万记录入库测试 | 测试通过 |
| 42 | 3.2.3 | HBase查询性能测试 |  |
| 43 | 3.2.3.1 | 5000万条记录中查询10条记录 | 测试通过 |
| 44 | 3.2.3.2 | 1亿条记录中查询1条记录 | 测试通过 |
| 45 | 3.2.3.3 | 1亿条记录中查询10条记录 | 测试通过 |
| 46 | 3.2.3.4 | 5亿条记录中查询1条记录 | 测试通过 |
| 47 | 3.2.3.5 | 5亿条记录中查询10条记录 | 测试通过 |
| 48 | 3.2.3.6 | 10亿条记录中查询1000条记录 | 测试通过 |
| 49 | 3.2.3.7 | 20亿条记录中查询1000条记录 | 测试通过 |
| 50 | 3.2.3.8 | 40亿条记录中查询1000条记录 | 测试通过 |
| 51 | 3.2.3.9 | 80亿条记录中查询1000条记录 | 测试通过 |
| 52 | 3.2.3.10 | 100亿条记录中查询1000条记录 | 测试通过 |

## 

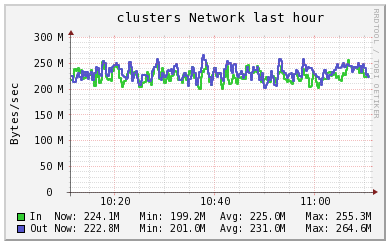
### 4.2 性能测试结果

### 4.2.1数据立方与HBase数据入库速率

#### 4.2.1.1 数据立方单个客户端入库速率

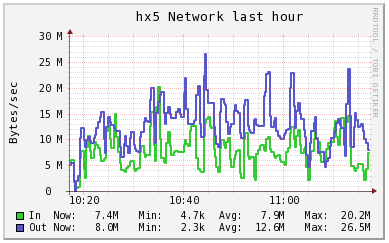


#### 4.2.1.2 数据立方多个客户端入库速率

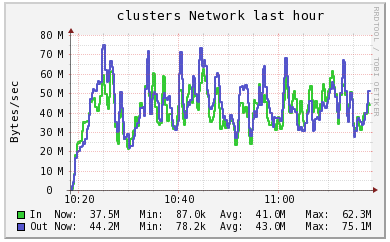


说明：上图是数据立方3个客户端的总带宽

#### 4.2.1.3 HBase单个客户端入库速率



#### 4.2.1.4 HBase多个客户端入库速率



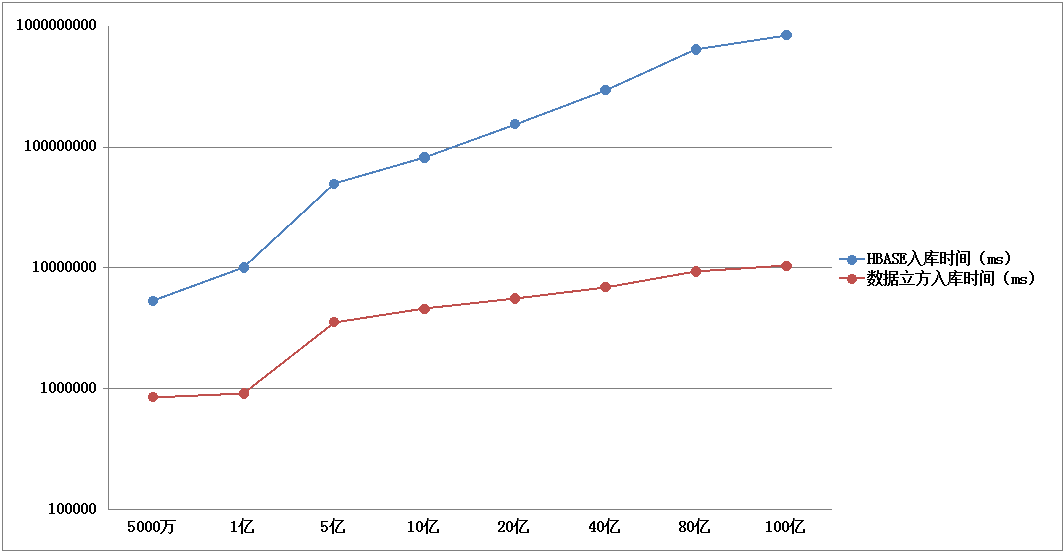
说明：上图是HBase测试中3个客户端的总带宽

### 4.2.2数据立方与HBase数据性能入库对比

#### 4.2.2.1数据立方与HBase数据入库时间表

|  |  |  |
| --- | --- | --- |
| 入库数据量（条） | HBASE入库时间（ms） | 数据立方入库时间（ms） |
| 5000万 | 5354522 | 850123 |
| 1亿 | 10120046 | 915136 |
| 5亿 | 49585509 | 3553514 |
| 10亿 | 82077432 | 4601800 |
| 20亿 | 155126436 | 5591187 |
| 40亿 | 296291321 | 6933071 |
| 80亿 | 641024743 | 9366578 |
| 100亿 | 845397549 | 10425641 |

#### 4.2.2.2数据立方与HBase数据入库性能对比图



### 4.2.3数据立方与HBase查询性能对比

#### 4.2.3.1数据立方与HBase数据查询时间表

|  |  |  |
| --- | --- | --- |
| 查询数据量（条） | HBase查询时间(ms) | 数据立方查询时间（ms） |
| 5000万条（查询10条） | 4112 | 1010 |
| 1亿条（查询1条） | 6414 | 1021 |
| 1亿条（查询10条） | 6734 | 1039 |
| 5亿条（查询1条） | 27092 | 1065 |
| 5亿条（查询10条） | 28305 | 1096 |
| 10亿条查询1000条 | 52257 | 1250 |
| 20亿条查询1000条 | 97145 | 1532 |
| 40亿条查询1000条 | 172918 | 2011 |
| 80亿条查询1000条 | 243814 | 2780 |
| 100亿条查询1000条 | 351892 | 3916 |

#### 4.2.3.2数据立方与HBase数据查询性能对比图

# 测试总结

本次测试分别对数据立方、HBase在不同数据量情况下，进行入库、入库流量、查询数据性能测试，从测试结果来看:

1、数据入库方面：数据立方与HBase在小数据量时两者的入库性能相差不明显,在100亿数据量入库时,数据立方入库性能是HBase的80倍。

2、数据查询方面：数据立方与HBase在5000万、1亿条数据量时，二者的查询性能相当，随着数据量的增大，在100亿数据量时，数据立方查询性能是HBase的近90倍。