QianBase 高可用性指南 1.6.6

2021 年 03 月

版权

公告

本文档包含的信息如有更改，恕不另行通知。

易鲸捷产品和服务附带的正式担保声明中规定的担保是该产品和服务享有的唯一担保。本文中的任何信息均不构成额外的保修条款。

声明

Microsoft® 和 Windows® 是美国微软公司的注册商标。Java® 和 MySQL® 是 Oracle 及其子公司的注册商标。Bosun 是 Stack Exchange 的商标。Apache®、Hadoop®、HBase®、Hive®、openTSDB®、Sqoop® 和 Trafodion® 是 Apache 软件基金会的商标。Esgyn，EsgynDB 和 QianBase 是易鲸捷的商标。

目录

前言

关于本文档

本指南介绍了 QianBase 中的高可用性以及如何配置和启用极端可用性功能。

目标读者

本指南适用于 QianBase 1.6.6 或更低版本系统和数据库管理员。

修订历史

版本	日期	摘要
1.6.6	2020 年 3 月	添加并修正参数列表
1.0.0	2020 年 3 月	第一版
1.1.0	2020 年 6 月	安装步骤做了调整，加入了 addon 自动化安装，删除了手动安装部分，并更新了检查部分，见附件。
1.2.0	2020 年 7 月	增加手动清理 TLOG 的情况和操作步骤。
1.6.6	2020 年 12 月	时间副本和 TLOG 不再适用，增减 HA 参数列表

批评与建议

我们支持您对本指南做出的任何批评与建议，并尽力提供符合您需求的文档。若您发现任何错误、或有任何改进建议，请发邮件至 support@esgyn.cn。

相关文档

本指南为 QianBase 文档库的一部分，QianBase 文档库**包括但不限于**以下文档:

文档名称	说明
QianBase 安装部署指南	本文介绍安装 QianBase，包括安装前准备、安装 Hadoop 发行版、故障排除、配置、启用安全功能、提高安全性和卸载 QianBase 等。
易鲸捷 Designer 用户指南	本文介绍易鲸捷图形化数据库管理工具
易鲸捷迁移工具用户指南	本文介绍如何安装和使用易鲸捷迁移工具。
ODB 用户指南	本文介绍了如何使用 odb（一种基于 ODBC 的多线程命令行工具）在易鲸捷数据库上执行各种操作。
QianBase 技术白皮书	本文介绍 QianBase 技术架构，组件介绍，技术特点等。
QianBase 数据库规划文档	本文介绍节点数量规划、数据目录和安装部署目录规划、集群角色分配规划等。

QianBase 管理员手册	本文介绍 QianBase 的日常运维常用系统命令、常用检查 SQL，用户权限配置，连接设置等内容。
QianBase 常见问题提排查与解决	本文介绍如何排查和解决 QianBase 的常见问题。
QianBase 灾难恢复手册	本文介绍 QianBase 灾难恢复设计原理，方案建议以及使用手册。
QianBase 备份恢复手册	本文介绍 QianBase 备份恢复设计原理，方案建议以及使用手册。
QianBase 数据库扩容指南	本文介绍 QianBase 如何更换节点，增加节点，删除节点等操作。
QianBase 数据库参数调优建议	本文介绍如何进行数据模型优化，CQD 参数优化等 。
QianBase 客户端安装手册	本文介绍 QianBase JDBC， ODBC 以及 Trafci 驱动安装。
QianBase JDBC 程序员参考指南	本文介绍 QianBase JDBC 驱动连接设置，开发人员指南。
QianBase ODBC 程序员参考指南	本文介绍 QianBase ODBC 驱动连接设置，开发人员指南。
QianBase SPSQL 存储过程用户手册	本文介绍 QianBase SPSQL 存储过程的使用。

Esgyn DBManager 用户手册	本文介绍图形化数据库监控运维工具 DB Manager 的使用。
QianBase 数据库迁移指南	本文介绍如何将常见关系型数据库（Oracle、MySQL、SQL server 等）迁移至 QianBase。
QianBase SQL 用户手册	本文是 QianBase 的 SQL 使用手册。
QianBase 命令行工具指南	本指南适用于维护和监管 QianBase 数据库的数据库管理员和支持人员。

1. 概述

QianBase 产品具有强大的 NonStop®传统，并且是为任务关键型企业数据库市场而构建的，高可用性是其体系结构的基础。

尽管基于没有对高可用性专门支持的商品硬件，但故障和恢复对于 QianBase 用户而言往往是透明的。当无法将故障与用户隔离开来时，该平台通过将故障对一个或多个特定查询的负面影响隔离并控制，从而最大限度地降低了总体影响，同时为其他查询提供了连续的可用性。

2. 架构

为了实现高可用性，QianBase 平台遵循以下设计原则

- 避免故障：识别并消除软件中的单点故障
- 快速恢复：检测到特定故障后，提供简单快速的机制来从故障中恢复
- 自动快速接管：检测到故障后，如果在平台中配置了辅助或备用组件，平台会快速自动执行切换

支持 HA 的基础架构功能包括：

功能	高可用性好处
名称节点冗余 Name Node Redundancy	防止名称节点故障
存储引擎主冗余 Storage Engine Master Redundancy	防止存储引擎主进程（Storage Engine Master process）失败
复制（数据块副本）Replication (data block	提供数据保护，防止节点和磁盘故障或数据损坏

copies)	
数据快照 Snapshot	在特定时间拍摄表快照，以将损坏的表及时恢复到先前已知的良好时间点
Zookeeper	实现基础设施服务的高度可靠的分布式协调
无共享架构 Shared nothing architecture	限制或消除节点之间的硬件或软件资源共享。 与共享导致竞争加剧和性能下降的“共享”体系结构不同，“无共享”显着提高了可伸缩性并提供针对单点故障的保护。
持久的过程保护 Persistent process protection	确保当持久受保护的软件进程失败或中止时，如果原始节点不可用，则会在同一节点或另一个节点中自动创建替换进程。 提供高可用性的服务。
浮动 IP Floating IP	QianBase 连接服务使用的抽象层,即使基础 IP 连接丢失并重新启动，它也允许应用程序客户端始终使用相同的 IP 地址。 提供客户端保护,并增强 IP 结构中潜在故障的可用性。
自动查询重试 Automatic Query Retry	检测到 QianBase 中特定类别的故障后，在数据库清理操作完成后自动重新提交查询的技术。 为最终用户提供内部数据库故障的透明性。
跨数据中心复制 Cross Datacenter Replication	使两个 QianBase 实例能够以事务完整性在主动-

主动或主动-被动操作模式下运行

下表说明了不同故障场景对用户的影响以及用于提供可用性的基础技术。

注意事项:

运行查询：发生故障时正在执行的用户 SQL 查询（事务性或非事务性）

新查询：检测到故障后启动的用户 SQL 查询（事务性或非事务性）

故障场景	种类	用户影响	使用的技术
CPU	HW	**运行查询:** 在故障节点上启动的那些查询将被中止。客户端应用程序必须重新发出查询。 其他运行查询可能会自动重试. **新查询：**新查询将正常运行，但性能可能会下降。	- 无共享架构 - 浮动 IP 地址 - 持续的过程 - 软件重试 - 自动查询重试
存储驱动器 Storage drives	HW	**运行查询**和**新查询：**继续进行而不会中断。在某些工作负载条件下可能会降级。	- 具有 n 个数据块副本的软件复制（可配置）
Network	HW	**运行查询**和**新查询**：取决于网络不可用的持续时间发生超时错误	- Zookeeper

HBase Region Server	SW	**运行查询：**在受影响的 Region Server 上具有活动事务的所有查询都将无法防止写丢失。 **新查询：**不受影响。	- 故障转移 Failover
监控过程 Monitor process	SW	**运行查询：**在计算机上执行的受影响查询将中止，并且将错误消息发送到客户端。客户端必须重新提交查询。某些活动查询可能会自动重试。 **新查询：**新查询的性能可能会有所下降，因为可用于处理查询的逻辑机较少。	- 无共享架构 - Zookeeper
分布式 TM 流程 Distributed TM process	SW	**活动查询：**来自 DTM 进程失败的节点的事务查询将被中止。其他事务查询和不使用事务的查询将不会有影响。 **新查询：**无影响	- 无共享架构 - 持久进程
Executor Server Process (ESP)	SW	**运行查询（由失败的 ESP 进程提供服务）：**通常没有影响。 在某	- 自动查询重试 Auto Query Retry

		些状态下的查询可能会出错，并且客户端将需要重新提交查询。 **新查询：**无影响
Master Executor process (MXOSRV)	SW	**运行查询：**（被失败的 MXOSRV - 自动重启进程控制）将中止，并向客户端发送错误消息； 客户端必须重新连接并重新提交查询。 **新查询：**无影响。

3. 硬件故障检测

快速硬件故障检测功能是一个可选的 HA 插件软件包。 该软件包可在不到一秒的时间内检测到硬件或软件（存储引擎）故障，并加快恢复速度。 它包括扩展 Red Hat Linux 操作系统和 QianBase 使用的存储引擎的模块。

3.1. 包组件

包含以下模块

- 存储引擎（HBase 和 HDFS）
- Linux watchdog，用于监视硬件故障
- SNMP 陷阱接收器

3.2. 前提

- 基本环境要求
- H3C server 硬件
- Red Hat Enterprise Linux 7.4
- Addon-installer 安装包
- 每个 server 的 IPMI 端口 IP 地址和管理员帐户。
  - 此 IP 地址是服务器特定的，与服务器 IP 地址不同。
- root 和 trafodion 用户访问权限

3.3. 安装

3.3.1. 运行 addon installer

```
[root@esgzb-qa-n113 20200608]# ll
total 16396
-rw-r--r--. 1 root root 16783438 Jun    8 18:01 addon-installer-redhat7-0608.tar.gz

[root@esgzb-qa-n113 20200608]# tar zxf addon-installer-redhat7-0608.tar.gz
[root@esgzb-qa-n113 20200608]# ll
total 16400
drwxr-xr-x. 4 root root        4096 Jun    8 17:56 addon-installer
-rw-r--r--. 1 root root 16783438 Jun    8 18:01 addon-installer-redhat7-0608.tar.gz

[root@esgzb-qa-n113 20200608]# cd addon-installer

[root@esgzb-qa-n113 addon-installer]# ./addon_install.py
*******************************
   Add-On Installation ToolKit
*******************************
Enter HDP/CDH web manager URL:port, (full URL, if no http/https prefix, default prefix is http://): 10.10.14.113
Enter HDP/CDH web manager user name [admin]:
Enter HDP/CDH web manager user password:
Confirm Enter HDP/CDH web manager user password:
Enter trafodion user name [trafodion]:
** CDH Cluster 1 node list: [esgzb-qa-n115.esgyncn.local,esgzb-qa-n116.esgyncn.local,esgzb-qa-n113.esgyncn.local,esgzb-qa-n114.esgyncn.local]
Enter list of Trafodion Nodes' hostname separated by comma, support simple numeric Regular Expression,
 i.e.          "n[01-12],n[21-25]","n0[1-5].com":          esgzb-qa-n115.esgyncn.local,esgzb-qa-n116.esgyncn.local,esgzb-qa-n114.esgyncn.local
                                                    --此处填写主机名，逗号隔开，也支持正则表达式格式
*****************
   Final Configs
*****************
+-------------+-----------------------------------------------------------------------------------+
|                    config                    type                    |                    value
|
+-------------+-----------------------------------------------------------------------------------+
|          mgr_url                                          |          http://10.10.14.113:7180
|
|               mgr_user                                                |               admin
|
|    node_list             |    esgzb-qa-n115.esgyncn.local,esgzb-qa-n116.esgyncn.local,esgzb-qa-n114.esgyncn.local |
```

```
|                traf_user                |                trafodion
|
+-------------+-----------------------------------------------------------------------------+
Confirm result (Y/N)    [N]: y

** Generating config file [/root/ha-addon/20200608/addon-installer/add_on_config] to save configs ...

**********
  Installation Start
**********
** Log file location: [/var/log/trafodion/install/add_install_20200608_180230.log]

TASK:              Copy              Trafodion              package              file
**********
Copy watchdog.rpm to all trafodion nodes ...
Copy hbase-server.jar and hadoop-hdfs.jar to all cluster nodes ...

Host [localhost]: Script [copy_files.py] ..... [    OK    ]

TASK:                          Hadoop                          modification
**********
Host [esgzb-qa-n115.esgyncn.local]: Script [hadoop_mods.py] ..... [    OK    ]

TASK:              Deploy              SNMP              Trap              Receiver
**********
Host [esgzb-qa-n116.esgyncn.local]: Script [deploy_SNMP.py] ..... [    OK    ]
Host [esgzb-qa-n114.esgyncn.local]: Script [deploy_SNMP.py] ..... [    OK    ]
Host [esgzb-qa-n115.esgyncn.local]: Script [deploy_SNMP.py] ..... [    OK    ]

TASK:                    Deploy                    watchdog                    service
**********
Host [esgzb-qa-n116.esgyncn.local]: Script [deploy_watchdog.py] ..... [    OK    ]
Host [esgzb-qa-n114.esgyncn.local]: Script [deploy_watchdog.py] ..... [    OK    ]
Host [esgzb-qa-n115.esgyncn.local]: Script [deploy_watchdog.py] ..... [    OK    ]

TASK:                                Stop                                cluster
**********
Host [esgzb-qa-n115.esgyncn.local]: Script [stop_cluster.py] ..... [    OK    ]

TASK:              Replace              hbase-server.jar              and              hadoop-hdfs.jar
**********
```

```
Host [esgzb-qa-n116.esgyncn.local]: Script [replace_jar.py] ..... [    OK    ]
Host [esgzb-qa-n114.esgyncn.local]: Script [replace_jar.py] ..... [    OK    ]
Host [esgzb-qa-n113.esgyncn.local]: Script [replace_jar.py] ..... [    OK    ]
Host [esgzb-qa-n115.esgyncn.local]: Script [replace_jar.py] ..... [    OK    ]

TASK:                                Trafodion                              modification
************************************************************
Host [esgzb-qa-n116.esgyncn.local]: Script [traf_mods.py] ..... [    OK    ]
Host [esgzb-qa-n114.esgyncn.local]: Script [traf_mods.py] ..... [    OK    ]
Host [esgzb-qa-n115.esgyncn.local]: Script [traf_mods.py] ..... [    OK    ]

TASK:                                  Start                                  cluster
************************************************************************
Host [esgzb-qa-n115.esgyncn.local]: Script [start_cluster.py] ..... [    OK    ]

Time Cost: 0 hour(s) 6 minute(s) 28 second(s)
*************************
   Installation Complete
*************************
```

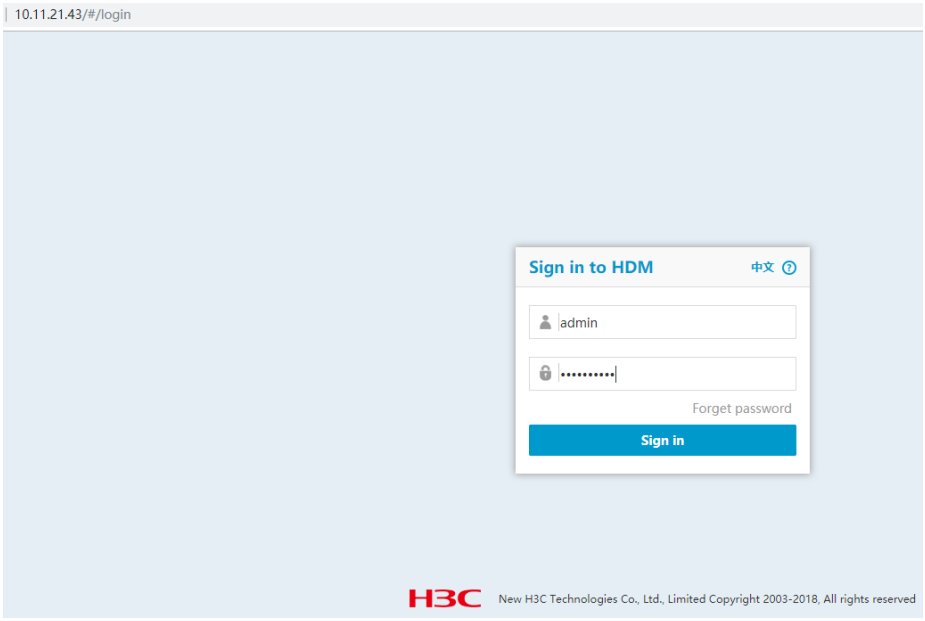
3.3.2. 为 H3C server 配置 IPMI

对于每个 H3C / IPMI GUI

- 为 SNMP 陷阱设置三个目标
- 启用 SNMP 版本 V1
- 这些目标应该是不受此 H3C / IPMI 管理的 Linux 主机

在 H3C / IPMI GUI 上配置 SNMP

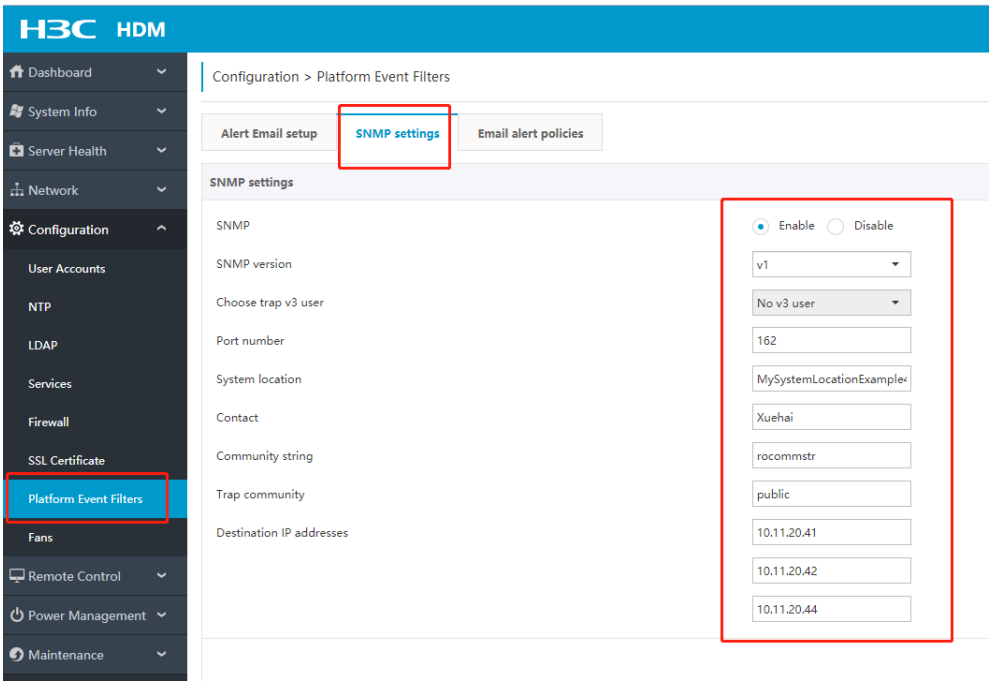
a) 登录 H3C / IPMI GUI：



b) 【configuration】-->【Platform Event Filters】-->【SNMP settings】

c) 配置目标 IP 地址：

例如：如果您在 10.11.20.43 Linux 主机上的 H3C / IPMI GUI 上，则在群集中配置其他三个 Linux 主机地址（10.11.20.41、10.11.20.42、10.11.20.44）



3.4. 验证

3.4.1. 配置 SNMP Trap Receiver

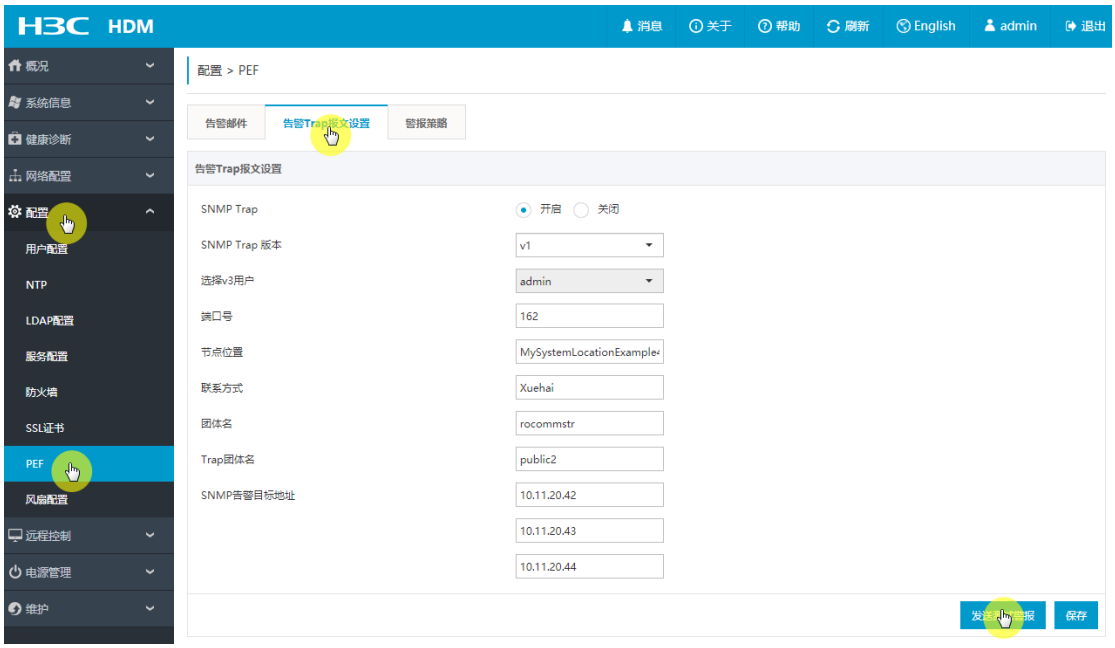
1. 检查 IPMI 配置

作为 QianBase 的数据库 linux 用户（例如 trafodion），检查**$TRAF_CONF/ipmi_host_map** 文件是否配置成功，如下表所示，其中的 ip 是 H3C 管理地址：

```
[trafodion@esggk012 conf]$ cat ipmi_host_map
10.0.11.141 esggk01.esgyncn.local
10.0.11.142 esggk02.esgyncn.local
10.0.11.143 esggk03.esgyncn.local
10.0.11.144 esggk04.esgyncn.local
10.0.11.145 esggk05.esgyncn.local
```

3.4.2. 验证 SNMP/Watchdog

1. 从 H3C / IPMI GUI 上发送测试警报，以验证 SNMP 是否正常工作



发送测试警报之后，切换到 trafodion 用户下查看相关日志记录，提示了接收到来自 xx agent

的信号:

```
[trafodion@esggk02 ~]$ cdl
[trafodion@esggk02 trafodion]$ vi snmp_trap_receiver.log
…
2020-06-08 18:29:45.86: receivedV1Trap, Agent: 10.0.11.141, generic code: 6, specific code: 0, timestamp: 215904400
oid: 1.3.6.1.2.1.1.5.0, val: HDM210200A00QH194002688
oid: 1.3.6.1.4.1.25506.13.1.1.1.1, val: Remote insight trap test.
2020-06-08 18:29:45.861: receivedV1Trap, Agent: 10.0.11.141, Linux Host: esggk01.esgyncn.local
2020-06-08 18:29:45.861: receivedV1Trap, No further action for the trap with the specific code: 0
2020-06-08 18:29:45.861: receivedV1Trap Exit.
```

2. 测试 watchdog 是否正常工作

```
ps -eaf| grep wd_keepalive          << to find the watchdog process
kill -9 <pid of wd_keepalive process>
```

注意事项:

不要选择任何内核 watchdog 线程 ([watchdog/0] [watchdog/1] ... etc.)，因为它们不是 wd_keepalive daemon 进程

预期结果：该节点应在 1 秒钟内重置，并且您应该在 Windows PC 上运行的“ SNMP Trap Watcher”中看到 snmp 陷阱事件。

4. 只读表的副本[1]

通常，不管 RegionServer 是否与可本地访问同一块（block）的其他 DataNodes 并置，RegionServer 都服务于来自客户端的读取请求。这样可以确保所读取数据的一致性。但是，由于 RegionServer 的性能不佳，网络问题或其他可能导致读取速度慢的原因，RegionServer 可能成为瓶颈。启用只读副本后，HMaster 会将区域（副本）的只读副本分发到群集中的不同 RegionServer。一个 RegionServer 服务于默认副本或主副本，这是唯一可以满足写请求的副本。如果为主副本提供服务的 RegionServer 宕机了，则写入将失败。其他 RegionServer 服务于辅助副本，紧随主 RegionServer 之后，仅查看已提交的更新。辅助副本是只读的，无法处理写请求。通过以固定的时间间隔读取主副本的 HFile 或通过复制，可以使辅助副本保持最新状态。如果他们使用第一种方法，则在进行更新并且 RegionServer 尚未将内存存储刷新到 HDFS 时，辅助副本可能不会反映数据的最新更新。如果客户端从辅助副本接收到读取响应，则通过将读取标记为“陈旧”来表明这一点。客户端可以检测读取结果是否陈旧，并做出相应的反应。 QianBase 不使用间隔或复制这两种方法，因为该功能用于只读表。

副本放置在不同的 RegionServers 上，并尽可能放置在不同的机架上。就读取而言，这提供了一种高可用性（HA）的措施。如果 RegionServer 不可用，则即使在使用其他副本之一由另一个 RegionServer 接管该区域之前，客户端仍可以访问它所服务的区域。直到新的 RegionServer 处理给定区域的整个 WAL 之前，读取可能都是过时的。

4.1. 启用只读副本支持

Esgyn 强烈建议为只读用户表启用此功能。

[1] Cloudera CDH 文档: https://docs.cloudera.com/documentation/enterprise/5-16-x/topics/admin_hbase_read_replicas.html

重要事项:

在启用只读副本支持之前，请确保考虑到它们增加的堆内存需求。尽管没有创建 HFile 数据的其他副本，但是只读副本区域具有与普通区域相同的内存占用量，因此在计算所需的增加的堆内存量时需要考虑这些副本。例如，如果您的表需要 8 GB 的堆内存，则在启用三个副本时，大约需要 24 GB 的堆内存。

要启用对副本读取的支持，必须设置以下属性。

属性名称	默认值	描述
hbase.ipc.client.allowsInterrupt	true Recommended: true	是否在客户端启用RPC线程中断。 默认值true使主要RegionServers可以访问其他区域的辅助副本中的数据。
hbase.ipc.client.specificThreadForWriting	TBD Recommended: true	是否在客户端启用 RPC 线程中断。具有回退 RPC 到辅助区域的区域副本时需要执行此操作 [Jason：这一个和前一个不确定使用哪个，目前我们都设置了这两个]。
hbase.client.primaryCallTimeout.get	10 ms Recommended: 1 ms	如果读取请求允许时间轴一致性，则超时时间（以

		毫秒为单位），HBase客户端将在读取提交到辅助副本之前等待响应。默认值为10。较低的值增加了远程过程调用的数量，同时降低了延迟 。
hbase.client.primaryCallTimeout.multiget	10 ms Recommended: 1 ms	在HBase客户端的多重获取请求（例如 HTable.get（List <GET>））之前，如果该多重获取请求允许时间轴一致性，以毫秒为单位的超时时间将会被提交给辅助副本。较低的值增加了远程过程调用的数量，同时降低了延迟。
hbase.client.replicaCallTimeout.scan	100 ms Recommended: 10 ms	在次要回退RPC's 之前，超时（以微秒为单位）给具有Consistency.TIMELINE的扫描请求被提交到区域的次要副本。将此值设置得

		较低会增加RPC的数量，但会降低p99延迟。
CQD HBASE_READ_REPLICA	OFF Recommended: ON	这是一个QianBase属性，应在QianBase”_MD_”.DEFAULTS表中设置。 启用该选项后，辅助region servers中的复制区域中将允许区域复制表的任何具有跳过读取冲突的读取。该数据可能与主要区域不同步。基于启动读取的节点ID，此读取内容分布在辅助区域中。没有跳过读取冲突的其他读取将来自主要region server。主要region server服务器将始终用于服务写入请求。

4.1.1. 使用 Cloudera Manager 配置只读副本

1. 在使用复制使副本保持最新之前，必须使用 HBase Shell 或客户端 API 将 QianBase 表的列属性 REGION_MEMSTORE_REPLICATION 设置为 false。 请参阅在表上激活只读副本。
2. 选择群集 **Cluster**> **HBase**。
3. 单击配置（**Configuration**）选项卡。
4. 选择“作用域（**Scope**）”>“ **HBase**”或“**HBase Service-Wide**”。
5. 选择类别（**Category**）>高级（**Advanced**）。
6. 找到 **HBase Service Advanced Configuration Snippet (Safety Valve) for hbase-site.xml** 属性“搜索”框中键入其名称进行搜索。
7. 使用与“使用命令行配置只读副本”相同的 XML 语法，和上图表创建配置并将其粘贴到文本字段中。
8. 单击保存更改以提交更改。

4.1.2. QianBase 配置

通过添加以下参数来编辑文件/etc/trafodion/conf/trafodion-site.xml

```
<property>
       <name>hbase.client.primaryCallTimeout.get</name>
       <value>1000</value>
</property>
<property>
       <name>hbase.client.primaryCallTimeout.multiget</name>
       <value>1000</value>
</property>
<property>
       <name>hbase.client.replicaCallTimeout.scan</name>
       <value>10000</value>
</property>
<property>
     <name>trafodion.hbase.read.specific.replica</name>
     <value>false</value>
```

```
</property>
```

通过添加以下参数来编辑/var/lib/trafodion/ms.env

```
SHARED_CACHE_SEG_SIZE=200
TMCLIENT_RETRY_ATTEMPTS=1
TM_TRANSACTIONAL_TABLE_RETRY=0
TM_JAVA_THREAD_POOL_SIZE=384
```

4.2. 激活表上的只读副本

在 RegionServers 上启用只读副本支持后，配置要为其创建只读副本的表。请记住，每个副本都会增加 HBase 在 HDFS 中使用的存储量。

要创建启用了读取复制功能的新表，请在表上设置 REGION_REPLICATION 属性。 QianBase CREATE TABLE 命令接受 HBase 选项 REGION_REPLICATION ='n'

QianBase 建议设置 n = 2

将数据加载到新表中，然后执行主要压缩。

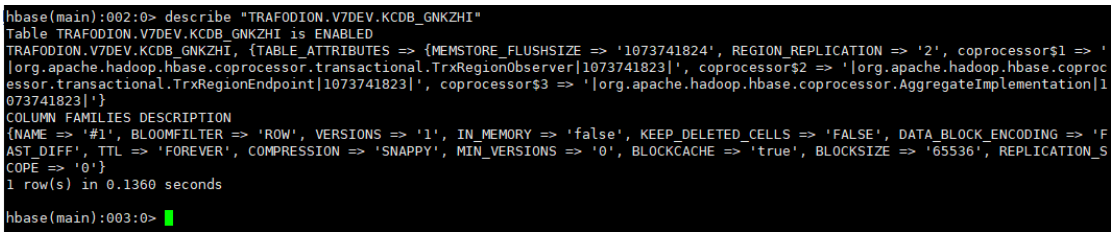
举例

```
CREATE TABLE TRAFODION.SEABASE.T2
  (
    ID            LARGEINT NO DEFAULT NOT NULL NOT DROPPABLE NOT SERIALIZED
  , COMMENTS       CHAR(20) CHARACTER SET ISO88591 COLLATE DEFAULT DEFAULT NULL
NOT SERIALIZED
  , REGISTER_DATE   DATE DEFAULT NULL NOT SERIALIZED
  , PRIMARY KEY (ID ASC)
  )
ATTRIBUTES ALIGNED FORMAT NAMESPACE 'TRAF_RSRVD_3'
  HBASE_OPTIONS
  (
    DATA_BLOCK_ENCODING = 'FAST_DIFF',
    COMPRESSION = 'SNAPPY',
    MEMSTORE_FLUSH_SIZE = '1073741824'
```

```
  )
;

-- GRANT SELECT, INSERT, DELETE, UPDATE, REFERENCES, ALTER, DROP ON
TRAFODION.SEABASE.T2 TO DB__ROOT WITH GRANT OPTION;

--- SQL operation complete.

SQL>alter table t2 alter HBASE_OPTIONS
+>  (
+>    DATA_BLOCK_ENCODING = 'FAST_DIFF',
+>    COMPRESSION = 'SNAPPY',
    MEMSTORE_FLUSH_SIZE = '1073741824',
    REGION_REPLICATION = '2'
  );+>+>+>

--- SQL operation complete.

SQL>showddl t2;


CREATE TABLE TRAFODION.SEABASE.T2
  (
    ID                              LARGEINT NO DEFAULT NOT NULL NOT DROPPABLE
      NOT SERIALIZED
  , COMMENTS                        CHAR(20) CHARACTER SET ISO88591 COLLATE
      DEFAULT DEFAULT NULL NOT SERIALIZED
  , REGISTER_DATE                   DATE DEFAULT NULL NOT SERIALIZED
  , PRIMARY KEY (ID ASC)
  )
ATTRIBUTES ALIGNED FORMAT NAMESPACE 'TRAF_RSRVD_3'
  HBASE_OPTIONS
  (
    DATA_BLOCK_ENCODING = 'FAST_DIFF',
    COMPRESSION = 'SNAPPY',
    MEMSTORE_FLUSH_SIZE = '1073741824',
    REGION_REPLICATION = '2'
  )
;

-- GRANT SELECT, INSERT, DELETE, UPDATE, REFERENCES, ALTER, DROP ON
TRAFODION.SEABASE.T2 TO DB__ROOT WITH GRANT OPTION;
```

```
--- SQL operation complete.
```

这是一个通过 HBase Shell 查看 QianBase 只读副本表上副本设置的示例。



5. 防止 OOM

Linux 内核根据系统上运行的应用程序的需求分配内存。由于许多应用程序预先分配了内存，并且通常不使用分配的内存，因此内核被设计为具有过量使用内存的能力，从而使内存使用效率更高。这种过量使用模型允许内核分配比其实际可用物理空间更多的内存。如果某个进程利用了分配给它的内存，则内核随后将这些资源提供给应用程序。当太多的应用程序开始利用分配的内存时，过量使用模型有时会出现问题，内核必须开始终止进程才能保持运行状态。内核用于在系统上恢复内存的机制简称为内存不足杀手或简称 OOM 杀手。

启用此配置可在恶意应用或进程消耗计算机上所有物理内存的情况下保护实例。此配置将导致系统在内存不足的情况下崩溃并重新启动。

在每个节点上，以 root 用户身份执行以下命令。 将“ X”设置为 30（持续 30 秒）。

```
echo "vm.panic_on_oom=1" >> /etc/sysctl.conf
 echo "kernel.panic=X" >> /etc/sysctl.conf
```

6. 附录

附录 A

```
[root@esggk09 ~]# systemctl status wd_keepalive

wd_keepalive.service - watchdog daemon
   Loaded: loaded (/usr/lib/systemd/system/wd_keepalive.service; enabled; vendor
preset: disabled)
   Active: active (running) since Thu 2020-01-23 17:33:46 CST; 1 day 14h ago
  Process: 410522 ExecStart=/usr/sbin/wd_keepalive (code=exited,
status=0/SUCCESS)
  Process: 410513 ExecStartPre=/bin/bash -c lsmod | grep ipmi_watchdog; if [ $? -
eq 0 ];then exit 0; else exit 1; fi (code=exited, status=0/SUCCESS)
  Process: 410507 ExecStartPre=/bin/bash -c lsmod | grep ipmi_watchdog; if [ $? -
ne 0 ]; then /usr/sbin/modprobe -v ipmi_watchdog timeout=10 pretimeout=0
action=reset nowayout=0 start_now=0; fi; exit 0 (code=exited, status=0/SUCCESS)
  Process: 410502 ExecStartPre=/bin/bash -c lsmod | grep TCO; if [ $? -eq 0 ];
then exit 1; else exit 0; fi (code=exited, status=0/SUCCESS)
  Process: 410494 ExecStartPre=/bin/bash -c lsmod | grep TCO; if [ $? -eq 0 ];
then /usr/sbin/modprobe -r iTCO_wdt; fi; exit 0 (code=exited, status=0/SUCCESS)
 Main PID: 410524 (wd_keepalive)
   Memory: 0B
   CGroup: /system.slice/wd_keepalive.service
           └─410524 /usr/sbin/wd_keepalive

Jan 23 17:33:46 esggk09.esgyncn.local systemd[1]: Starting watchdog daemon...
Jan 23 17:33:46 esggk09.esgyncn.local bash[410507]: ipmi_watchdog         25058
0
Jan 23 17:33:46 esggk09.esgyncn.local bash[410507]: ipmi_msghandler       46608
4 ipmi_ssif,ipmi_devintf,ipmi_watchdog,ipmi_si
Jan 23 17:33:46 esggk09.esgyncn.local bash[410513]: ipmi_watchdog         25058
0
Jan 23 17:33:46 esggk09.esgyncn.local bash[410513]: ipmi_msghandler       46608
4 ipmi_ssif,ipmi_devintf,ipmi_watchdog,ipmi_si
Jan 23 17:33:46 esggk09.esgyncn.local systemd[1]: Started watchdog daemon.
```

附录 B

```
[root@esggk09 ~]# ipmitool mc watchdog get
```

```
Watchdog Timer Use:     SMS/OS (0x44)
Watchdog Timer Is:      Started/Running
Watchdog Timer Actions: Hard Reset (0x01)
Pre-timeout interval:   0 seconds
Timer Expiration Flags: 0x10
Initial Countdown:      1 sec
Present Countdown:      0 sec
```

7. 参数

当前 QianBase1.6.6 版本安装自带 HA 参数列表:

序号	参数名称	组件	默认值	推荐值	单位	描述	参数设置位置	备注
1	hbase.client.pause	HBase		1000	ms	常规客户端暂停值。主要用作在重试失败的获取、区域查找等之前等待的值。请参阅 hbase.client.retries.number，用于描述我们如何从该初始暂停量中回退，以及该暂停如何在重试的情况下工作	HBase CM 参数	
2	hbase.client.retries.number	HBase	3	3		最大重试次数。用作所有可重试操作的最大值，例如获取单元格值、开始行更新等。重试间隔是基于 hbase.client.pause 文件. 一开始我们在这个时间间隔重试，但是在 backoff 的情况下，我们很快就会达到每 10 秒重试一次。请参阅 HConstants#RETRY_BACKOFF 了解备份如何升级。更改此设置并 hbase.client.pause 文件以适应您的工作量	HBase CM 参数	最大重试次数。用作所有可重试操作的最大值，例如获取单元格值、开始行更新等。重试间隔是基于 hbase.client.pause 文件. 一开始我们在这个时间间隔重试，但是在退避的情况下，我们很快就会达到每 10 秒重试一次。请参阅 HConstants#RETRY_BACKOFF 了解备份如何升级。更改此设置和 hbase.client.pause 文件以适应您的工作量
3	zookeeper.recovery.retry	ZooKeeper		1		不重试	hbase-site.xml 的 HBase 客户端高级配置代码段（安全阀）	没有重试

4	zookeeper.session.timeout	ZooKeeper		90000		可能导致错误的故障检测	HBase CM 参数	可能导致错误的故障检测，该参数取决于机器性能
5	maxSessionTimeout	ZooKeeper		90000				在 zookeeper 中设置
6	hbase.regionserver.executor.openregion.threads	HBase	3	20		HBASE-21186	hbase-site.xml 的 HBase 客户端高级配置代码段（安全阀）	VM 不适用
7	hbase.master.executor.openregion.threads	HBase	5	10		在主服务器中处理区域打开的池线程数。	HBase CM 参数	VM 不适用
8	hbase.master.executor.closeregion.threads	HBase	5	10		在主服务器中处理区域关闭的池线程数。	HBase CM 参数	VM 不适用
9	hbase.master.executor.serverops.threads	HBase	5	10		用于处理主服务器中 RegionServer 恢复的池线程数	HBase CM 参数	VM 不适用
10	dfs.namenode.avoid.read.stale.datanode	HDFS	FALSE	TRUE		指明是否避免在多个指定时间间隔内未接收到其心跳消息的“stale” datanodes 上读取。Stale datanodes 将被移动到返回读取的节点列表的末尾。对于类似的写入设置，参考 dfs.namenode.avoid.write.stale.datanode。	HDFS CM 参数	参考连接: https://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml
11	dfs.namenode.avoid.write.stale.datanode	HDFS	FALSE	TRUE		指示是否避免在多次指定时间间隔内未接收到其心跳消息的 stale datanodes 上写入。写入将避免使用 stale datanodes，除非超过配置的比率	HDFS CM 参数	参考连接: https://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml

						(dfs.namenode.write.stale.datanode.ratio)的数据节点被标记为 stale。		
12	dfs.namenode.write.stale.datanode.ratio	HDFS	0.5f	1.0f		当 stale datanodes 数与标记的总数据节点数的比率大于此比率时，请停止避免写入 stale nodes，以防止引发热点。	HDFS CM 参数	
13	dfs.namenode.check.stale.datanode	HDFS	FALSE	TRUE		启用 stale datanode 功能 在 NameNode 处添加一个名为“stale”的新 DataNode 状态。如果 DataNodes 在使用配置以秒为单位参数 "dfs.namenode.stale.datanode.interval"配置的超时时间内未向 NameNode 发送心跳消息，则会将其标记为过时（默认值为 30 秒）。NameNode 在返回块位置进行读取时选择一个 stale datanode 作为最后一个要读取的目标。 默认情况下，如果一个盒子停了，datanode 将在 10:30 分钟后被 namenode 标记为 dead。同时，这个数据节点仍将由 nanenode 提议用于写块或读取副本。如果 datanode 崩溃也会发生这种情况：没有关闭钩子来告诉 nanemode 东西不在了。 尤其是对于 HBase 会是一个问题。HBase regionserver 的生产超	hdfs-site.xml 的 HDFS 客户端高级配置代码段（安全阀）	参考链接： https://issues.apache.org/jira/browse/HDFS-3703

						时通常为 30 秒。因此，使用这些配置，当一个盒子死机时，HBase 会在 30 秒后开始恢复，而 10 分钟后，namenode 会将同一个盒子上的块视为可用。除了写错误之外，这还会引发大量的漏读: -- 在恢复过程中，需要先在 HBase 上记录已用的块（在 HBase 中记录已用的块） --在恢复之后，读取这些数据块（“HBase region”）将有 33%的时间失败，使用默认的副本数，从而减慢数据访问速度，特别是当错误是套接字超时时（即大多数时间约为 60 秒）。		
14	dfs.datanode.socket.write.timeout	HDFS		10000	ms	10s (default = 8 * 60s?)	hdfs-site.xml 的 HDFS 客户端高级配置代码段（安全阀）	
15	ipc.client.connect.max.retries.on.timeouts	HDFS	45	2		指明客户端在套接字超时时建立服务器连接的重试次数。	hdfs-site.xml 的 HDFS 客户端高级配置代码段（安全阀）	
16	dfs.namenode.stale.datanode.interval	HDFS	30000	15000	ms	将 datanode 标记为“stale”的默认时间间隔，即如果 namenode 已经超过此时间间隔没有从 datanode 接收到 heartbeat 消息，那么 datanode 将被标记并默认视为“stale”。stale 时间隔不能太小，否则可能导致 stale 状态的更改过	HDFS CM 参数	参考链接: https://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml

						于频繁。因此，我们设置了一个最小失效间隔值（默认值是心跳间隔的 3 倍），并保证失效间隔不能小于最小值。在租约/块恢复期间，可以避免过时的数据节点。可以有条件地避免读取（请参阅 dfs.namenode.avoid.read.stale.datanode)对于写入（请参阅 dfs.namenode.avoid.write.stale.datanode).		
17	hbase.lease.recovery.first.pause	HBase		1000	ms	首次恢复 hdfs wal lease	hbase-site.xml 的 HBase 客户端高级配置代码段（安全阀）	
18	hbase.lease.recovery.pause	HBase		500	ms	恢复 hdfs wal lease	hbase-site.xml 的 HBase 客户端高级配置代码段（安全阀）	
19	hbase.regionserver.hlog.splitlog.writer.threads	HBase	3	20		写入 wal 的并发	hbase-site.xml 的 HBase 客户端高级配置代码段（安全阀）	对产品功能不会产生影响，对性能可能会有影响，需要现场适配 VM 不适用
20	hbase.regionserver.wal.max.splitters	HBase		20		handle wal 的并发	hbase-site.xml 的 HBase 客户端高级配置代码段（安全阀）	VM 不适用
21	hbase.regionserver.hlog.blocksize	HBase		10485 7600	byte (100M)	roll hlog to new file when up to the parameter	hbase-site.xml 的 HBase 客户端高级配置代码段（安全阀）	对产品功能不会产生影响，对性能可能会有影响，需要现场适配 VM 不适用
22	hbase.regionserver.executor.closeregion.threads	HBase		10		close region 并发	hbase-site.xml 的 HBase 客户端高级配置代码段（安全阀）	VM 不适用

23	hbase.regionserver.logroll.multiplier	HBase		0.9		log roll per	hbase-site.xml 的 HBase 客户端高级配置代码段（安全阀）	VM 不适用
24	Region Mover Threads	HBase	1	10		向 RegionServer 或从 RegionServer 加载和卸载区域时要使用的线程数。可用于**提高退役或滚动重启操作的速度**。	HBase CM 参数	VM 不适用
25	TMCLIENT_RETRY_ATTEMPTS=1	ms.env		1			所有节点的 ms.env	跟 DDL 有关，doCommitDDL 方法会使用到
26	TM_TRANSACTIONAL_TABLE_RETRY=0	ms.env		2			所有节点的 ms.env	该值会导致 SQL DDL&DML 某些功能(比如在线创建索引 M-15473)不可用
27	TM_JAVA_THREAD_POOL_SIZE=384	ms.env	384	32		(默认为 128)	所有节点的 ms.env	
28	hbase.status.published	trafodion-site.xml		TRUE		<property> <name>hbase.status.published</name> <value>true</value> </property>	所有节点的 trafodion-site.xml	
29	hbase.status.listener.class	trafodion-site.xml		org.apache.hadoop.hbase.client.ClusterStatus		<property> <name>hbase.status.listener.class</name> <value>org.apache.hadoop.hbase.client.ClusterStatusListener$ZKClientListener</value> </property>	所有节点的 trafodion-site.xml	

				Listener$ZKClientListener				
30	net.ipv4.tcp_retries2	linux	3	5			所有节点修改/etc/sysctl.conf 并 sysctl -p /etc/sysctl.conf 使其生效	
31	net.ipv4.tcp_syn_retries	linux	1	3			所有节点修改/etc/sysctl.conf 并 sysctl -p /etc/sysctl.conf 使其生效	
32	DTM_TRANS_HUNG_RETRY_INTERVAL	tmstart	20000	20000			所有节点的 tmstart （这个参数必须在 tmstart 脚本的第 36 行~44 行之间增加）	
33	hbase.wal.meta_provider	HBase		multiwal			hbase-site.xml 的 HBase 客户端高级配置代码段（安全阀）	
34	hbase.procedure.store.wal.max.retries.before.roll	HBase	1	2			hbase-site.xml 的 HBase 客户端高级配置代码段（安全阀）	
35	hbase.procedure.store.wal.sync.failure.roll.max	HBase	1	2			hbase-site.xml 的 HBase 客户端高级配置代码段（安全阀）	
36	hbase.rpc.write.timeout	HBase		10000			hbase-site.xml 的 HBase 客户端高级配置代码段（安全阀）	

37	dfs.client.block.write.replace-datanode-on-failure.policy	HDFS		NEVER			hdfs-site.xml 的 HDFS 客户端高级配置代码段（安全阀）	
38	dfs.client.block.write.replace-datanode-on-failure.enable	HDFS		TRUE			hdfs-site.xml 的 HDFS 客户端高级配置代码段（安全阀）	
39	hbase.client.pause	HBase	100	1000				
40	SQ_MON_ZCLIENT_MY_ZNODE_CHECKRATE	Monitor	5	2	s	monitor 心跳时间	sqenvcom.sh	
41	SQ_MON_ZCLIENT_MY_ZNODE_PING	Monitor	1	2	s	monitor 连接 zk 失败后，ping 的心跳	sqenvcom.sh	
42	SQ_MON_ZCLIENT_MY_ZNODE_DELAY	Monitor	3	7	s	断网其他服务器相对于检测本机延迟时间	sqenvcom.sh	
43	SQ_MON_ZCLIENT_SESSION_TIMEOUT	Monitor	60	39	s	monitor 和 zk 的超时时间。	sqenvcom.sh	
44	SQ_MON_ZCLIENT_TIMEOUT_FACTOR	Monitor	3	10	s	连接 ZooKeeper 丢失后，判断节点宕机的超时系数，宕机时间=超时时间*该值	sqenvcom.sh	
45	Tick	ZooKeeper	2000	2000	ms	心跳时间	ZooKeeper	
46	syncLimit	ZooKeeper	5	15	次	重试次数	ZooKeeper	
47	hbase.trafodion.asyncnc.wal	Hbase Coprocessor	1	2		Wal 的刷新策略，异步或同步	CDH 中 Hbase 参数设置	
48	zookeeper.session.timeout	HBase	20*ticktime	35000	ms	HMaster 客户度端 zookeeper.session.timeout	hbase-site.xml	该值设置须小于等于 ZooKeeper Server 的 maxSessionTimeout(参数 5)。

								Zookeeper 也有相同名字的参数(详见参数 4)，使用时请注意区分。

HA 测试环境手动增加参数列表：

序号	HA 测试环境手动增加参数	默认值	推荐值	设置位置	备注
1	ha.zookeeper.session-timeout.ms	5000	2000	hdfs core-site.xml 的群集范围高级配置代码段（安全阀）	
2	hbase.wal.regiongrouping.numgroups	1	3	HBase CM 参数	后期测试都基于 3 测试，无影响后合入 ha-installer