

天翼云 HBASE 数据库 用户指南

中国电信股份有限公司云计算分公司





1	购买指南和快速入门	. 1
1.1	产品介绍	.1
1.2	产品特征	.1
1.3	版本说明	.2
1.4	应用场景	.2
1.5	快速入门	.2
1. 5.	1 单节点版本规格选型	.2
1. 5.	2 企业高可用版本规格选型	. 3
1. 5.	3 访问准备	. 4
1. 5.	4	. 4
1. 5.	5 从零开始使用 HBase	. 6
1. 5.	<i>6 续费和退订</i>	. 7
2	用户指南	. 8
2.1	名词定义	. 8
2.2	HBASE 实时监控和历史指标	. 8
2. 2.	1 集群概览	. 8
2. 2.	2	11



2.3	预警管理	
2. 3. 1	<i>设置预警</i>	
2. 3. 2	查看集群后台操作进度	21
2. 3. 3	查看系统消息	
2.4	表信息管理	23
2. 4. 1	可视化表操作	23
2. 4. 2	可视化 SHELL	26
2.5	二级索引	27
2. 5. 1	进入二级索引页面	27
2. 5. 2	新建索引	29
2. 5. 3	删除索引	
2. 5. 4	查看索引操作日志	
2.6	HBASE 冷热存储	
2. 6. 1	WAL 日志热存储	
2. 6. 2	表冷热分离	
2.7	配置管理	
2. 7. 1	配置修改	
2.8	场景推荐配置优化	
2. 8. 1	集群级别推荐配置优化	
2. 8. 2	表级推荐配置优化	41
2.9	安全访问控制	42



1 购买指南和快速入门

2.9.1 配置访问控制	
2.10 自助导入导出	43
2.10.1 集群迁移	
2.10.2 后台导入导出	
3 开发指南	49
3.1 常用概念	
3.2 开发流程	
3.3 环境准备	
3.3.1 开发和运行环境简介	
3.3.2 配置并导入样例	
3.3.3 开发 HBASE 应用	
3.3.4 样例代码说明	
4 常见问题	92
A 免责声明	96



购买指南和快速入门

1.1 产品介绍

概述

天翼云 HBASE 数据库(以下简称云 HBASE) 是基于 hdp HBase 优化和完善,支持 NOSQL 和二级索引(兼容 phoen i x5. x),提供高性能、低延迟、易运营实时分析云服务,适合上百 TB 数据存储,百万级 QPS 的分布式计算应用场景。

单节点版本

基于 hbase2.0 优化适用企业测试、个人学习、培训机构/测试开发应用。

企业高可用版本

云 HBASE 发行版紧随开源社区的最新引擎 2.0 并做问题修复,通过严格测试和配置优化快速集成 hbase 生态的最新组件,并在可靠性、安全性、管理性等方面做企业级的增强,自研智能分析引擎实现自动巡检预 警和问题诊断,场景配置优化推荐,智能交互提示并持续改进相关功能。

1.2 产品特征

云 HBASE 定时收集 Hbase 相关运行的指标和参数配置,结合天翼云多年的运营经验,实现业务场景自适应,保障系统更平稳,获得系统性能最大化。

自研的日志指标分析引擎定时巡检系统并收集系统状态日志和预警消息,对日志进行分析和问题判 断,减少了用户的专业技术人员投入,提高系统的可用性。



1.3 版本说明

云 HBASE 是基于开源 hdp hbase2.0 版本上进行功能优化。

1.4 应用场景

实时动态处理分析-不同应用场景的差异化需求

行业	应用场景	能力需求
🄊 安防	 ・ 公安警务:平安城市、天罗地网、雪亮工程 ・ 智慧城市:智慧交通,车联网 ・ 民生服务:水利、气象等监控与指挥调度 阳光厨房、平安校园 	数据里超大,监控视频要求实时 分析存储、全文检索,能够支撑 时空建模、AI建模
建基 工业互联网、 旅游商圈	 物联网工厂: 传感器、机器控制 旅游商圈: 人流分析管控,客流分析管控 	延时低,实时查询响应,时序要求 按时空地理信息格式存放 时空库实时建模
命 政府、金融	・ 社保、税务: 个人画像,企业画像 ・ 金融: 风控	数据 <mark>望</mark> 大,能够支持多种建模工具, 多维度分析建模,可靠度高
角 其它	• 传媒、新闻、社交分发、聊天信息	数据存储格式要求丰富、 数据里大、并发大

1.5 快速入门

1.5.1 单节点版本规格选型

单节点版本适用于企业测试、个人学习、培训机构/测试开发应用

- 1. CPU/内存规格: 2C4GB、4C8GB、8C16GB
- 2. 存储类型: 普通 IO、高速 IO、超高速 IO
- 3. 存储大小: 普通 IO 和高速 IO 最小 100GB,超高速 IO 最小 20GB, 以 20G 为单位递增,最大 2TB。



🛄 说明

单节点版本的数据副本数为1个副本。

1.5.2 企业高可用版本规格选型

云 hbase 可以支撑多种业务场景,不同业务场景对集群的各项指标需求不同,因此我们为

不同规模的企业和不同的业务场景提供了多种组合的集群规格组合:

1. 管理节点 CPU/内存规格: 8C16GB、16C32GB

2. 数据节点 CPU/内存规格: 8C16GB、16C32GB、32C64GB、32C128GB

3. 管理节点存储类型:普通 IO 存储大小: 500G, 管理节点为高可用 2 个节点。

4. 数据节点存储类型: 普通 IO、高速 IO、超高速 IO 存储大小: 普通 IO 和高速 IO 最小 200G, 以 20G

为单位递增, 最大 2TB。超高速 IO 最小 20GB,以 20GB 为单位递增, 最大 1TB.存储个数: 增加磁盘个数,

磁盘 IO 可以按照个数递增,具体上限受存储网络的限制。

🛄 说明

管理节点默认双节点高可用模式。数据节点单个订单允许添加最大的数据节点数为10个。单个数据节点最多支持5个 磁盘测试进行再验证确认。数据节点规格32C时,管理节点只能为16C。

1.5.2.1 管理节点选择

管理节点不带存储,默认是主备保障单点容灾

数据节点的规模	master 选择
10 <= 数据节点数 < 20	8cpu16g
20 <= 数据节点数 < 50	16cpu32g



1.5.2.2 数据节点选择

数据节点阶段选择: 8cpu16g 起步, 最大 32cpu128g

以 1KB 为准:

QPS	core 的个数及选择	单个数据节点
0 ~ 1w	2 台 8cpu16g	
1w ~ 10w	>=2 16cpu32g	可以按照每台 2.5w 算,批量 12w
10w ~5000w	>=2 32cpu64g	可以按照每台 5w 算,批量 24w

1.5.3 访问准备

访问准备

第一步 准备同一 vpc 内的 ECS 云主机

处于安全考虑, HBase 仅提供内网访问模式。

第二步 设置虚拟子网和安全组

通过虚拟私有云(VPC)和网络安全组实现网络隔离。VPC 允许租户通过配置 VPC 入站 IP 范围,来控制连接数据库的 IP 地址段

1.5.4 快速创建集群

用户登录天翼云官方网站,首次在云数据库 HBASE 产品页面,点击购买按钮进入订购页面。

您可以根据您的需要选取单节点版本或者企业版。示例选取企业版,见下图:

1	购买指南和快速入]
---	----------	---

	云数据库 Hbase		
O.	基础配置		
	* 类型:	HBase-单节点版 HBase-企业版 @	
0	管理节点配置		
	* CPU/内存规格:	16核32GB 8核16GB	
	*节点数量:	2	
Ģ	数据节点配置		
	* CPU/内存规格:	32核128GB 32核64GB 16核32GB 8核16GB	
	*存储:	普通IO × 磁盘容量 200 수 GB 磁盘数量 1 수 增加一块磁盘	×

管理节点数量固定为2个,

天

異日 **e** Cloud

数据节点的数量预估请参考 1.5.2.2 数据节点选择一节。

数据节点的存储需要根据您的业务预估数据量,再乘以3副本,除以数据节点数量。

	* 节点数量:	5	Ø
0	网络 * 虚拟私有云:	vpc-e098	✓ 査看虚拟私有云 ᢗ
	*子网:	subnet-e098-0726	
	*安全组:	default	~
0	集群配置		
	* 集群名称: * 管理员密码:	test0828]
	* 确认密码:]

填写上面的信息后,选取购买时长

													1	购买打	自南和快	速入门
0	购买量 * 购买时长:	 1个月	〇 2个月	 3个月	 4个月	 5个月	 6个月	 7个月	 8个月	 9个月	 10个月	 11个月	 1年	 2年	 3年	<mark>2</mark> 个月

然后勾选用户协议(请仔细阅读用户协议),提交。

1.5.5 从零开始使用 HBase

1 导入表结构

用户在可视化 shell 导入已经准备好的建表文件。详细格式规范见表规范模板。

如需临时建表,可在表可视化管理进行建表操作。建表请参考 rowkey 设计原则

2 准备用户接入和查询分析程序

用户通过集群,请先下载开发运行依赖包,配置文件和示例代码,创建接入程序请参见开发指南。

3 创建二级索引

在百万级数据以下建议用户在表可视化创建业务需要查询条件索引,来提高查询速度。

4 存放和读取数据

申请单独的 ecs, 用于存放用户的, 基于用户的应用程序, 把云 HBASE 集群作为存储后端, 存放和读取数据。

5日常指标巡检和场景适配

用户可以在<u>集群概览和用例信息</u>进行集群运行情况和主机运行情况巡检。系统指标采集分析程序会对系统 进行读写场景分析,并发出系统消息,提示用户,用户可以根据场景,参考天翼云优化的配置参数,对系 统做出合理调整,以提高系统稳定性和性能。



1.5.6 续费和退订

当集群处于服务中或者您可在集群列表内选取您需要续费或者退订的集群进行对应的操作。

集群名称	类型	版本	集群状态	计费方式	付费状态	创建时间	描述	操作
Hbase105	集群	2.0	● 服务中	后付费企业版 /按月付费	已付费	2019-07-23 10:41:17		明细 概览 更多
Hbase231	集群	2.0	▶ 施工中	后付费企业版 /按月付费	已付费	2019-08-19 15:13:03		<mark>续费</mark> 明细 概览 退订

如果您要续费,点击续费链接,进入续费页面:

续费									
hbase将进行续订	hbase将进行续订操作,续订产生续订订单,请与48小时内支付订单,否则续订操作失败								
atta inclusion	~*	d Date:	4745						
实例ID	各称	状态	规恒	王山伯田山					
		暂无数据							
总条数:0									
* 续订时长:	月 ~ 3 ~)								
配置费用:									
茶									
了解计费详情									
取消	确认								

选择续订时长,点击确认提交。

如果您要退订,点击退订链接,进入退订页面:

<mark>退订</mark> hbase将进行退订操作,在申请退订前,请做好数据备份工作,退订后数据将保留15个自然日,15天后相关数据将不予保留									
实例ID	名称	状态	规格	到期时间					
		暂无数据							
总条数: 0									
取消 确认									

仔细阅读完信息后,点击确认提交。



2 用户指南

2.1 名词定义

Phoenix: apache 的一款可以支持在 hadoop 做 0LTP 操作的查询引擎,鉴于 hbase 本身 无法支持 sql,云 HBASE 引入后进行封装,可提供二级索引和在百万级数据下毫秒级查询。

可视化 shell: 对 hbase shell 做了优化并实现 web 可视化。避免用户繁琐的登录和切换

运维。

Rowkey: hbase 数据库数据记录的主键,用于区分数据的唯一性。

hbase 数据是 kv 的数据结构。

2.2 HBase 实时监控和历史指标

2.2.1 集群概览

在集群概览中,展示了各组件的健康状态及其节点情况,针对关键组件指标通过可视化展现,动态实 时更新,同时支持不同时间区间的历史展示。

用户可以针对特定指标,通过弹窗查看不同时间区间的指标历史趋势,及特定时间点的指标具体数值。

对于 HBase 相关组件的具体指标,用户可以通过集群概览中的组件链接,进入组件指标详细界面。在 组件界面中,提供组件节点及指标的历史趋势、控制组件的功能,方便用户更准确地诊断问题和解决 问题。

2.2.1.1 进入集群概览

点击左边的导航栏的集群列表页面,显示了当前登录用户所在资源池的所有集群。

翼 己 Cloud								2 用户指南
● 集群列表	+ 添加集群							0
集群名称	类型	版本	集群状态	计费方式	付费状态	创建时间	描述	操作
Hbase105	集群	2.0	● 服务中	后付费企业版 /按月付费	已付费	2019-07-23 10:41:17		明细 概览 更多
Hbase231	集群	2.0	▶ 施工中	后付费企业版 /按月付费	已付费	2019-08-19 15:13:03		明细 概览 更多
共2条 1	0条/页 > 《	1 >	前往 1 页					

通过集群名字或者对应集群操作列中概览链接,可以跳转到该集群的概览页面。

HBase 数据库	♀ 当前集群: Hba	se105 V	0	5. 110					
集群列表	•集群列表 +	添加集群							0
集群概览	集群名称	类型	版本	集群状态	计费方式	付费状态	创建时间	描述	操作
实例信息	Hbase105	集群	2.0	 ● 服务中 	后付鶈企业版 /按月付费	已付费	2019-07-23 10:41:17		明细 概览 更多
HBase 概范 表信息管理		集群	2.0	▶ 施工中	后付费企业版 /按月付费	已付费	2019-08-19 15:13:03		明细概觉更多
预警管理访问控制	共2条 10条/	ī, ∨ <	1 >	前往 1 页					
配置管理									

点击概览链接后,系统显示集群的服务组件概览和集群的 cpu,内存,负载,hdfs 磁盘使用率等各项 基本指标。

フ e	Cloud						
						2)	用户指南
	● 集群概览						最近1小时 🗸
	服务概览					HDFS 磁盘使用率	
	服务	健康状态	角色数				
	HBASE	 已启动 	4 hbase_clie	nt 2 hbase_master 2 hbase_regionserver 1 pho	penix_query_ser		
	HDFS	 已启动 	2 datanode	4 hdfs_client 3 journalnode 2 namenode 2 zkf	c	77 1 3	
	YARN	 已启动 	1 app_timeli	ne_server 2 nodemanager 1 resourcemanager	1 timeline_rea	1.1.5	%
	ZOOKEEPER	 已启动 	4 zookeeper	_client 3 zookeeper_server			
	集群负载		Q	集群 CPU 使用量	Q	集群网络使用量	Q
	8]			100% -		244 KB -	1
	6 -			80% -		195 KB -	
	4 -			60% -		146 KB -	
				40% -		97 КВ	4 4 4 4
			4.4	20% -		48 КВ-	
	0	2019-09-06 15:29:30		0% 2019-09-06 14:56:08 2019-09-06 1	5:29:30	0 B 2019-09-06 14:56:08 2019-09-06	5 15:29:30

您也可以通过在顶部的当前集群下拉框,选择不同的实例,来查看不同集群的概览。

♀ 当前集群: Hbase231 ~	Ęº 🥂	5	
• 集群概 20190215 Hbase231			

2.2.1.2 查看组件服务和指标

HBase 需要 Hadoop 体系的组件支持,在服务概览中,我们可以查看到 Hadoop 相关组件的健康状态和 各角色数量,其中也包括了 HBase 的状态。

服务概览		
服务	健康状态	角色数
HBASE	 已启动 	2 hbase_client 1 hbase_master 1 hbase_regionserver 1 phoenix_que
HDFS	 已启动 	1 datanode 2 hdfs_client 1 namenode 1 secondary_namenode
YARN	• 已启动	1 app_timeline_server 1 nodemanager 1 resourcemanager 1 timelin
ZOOKEEPER	• 已启动	2 zookeeper_client 1 zookeeper_server

2.2.1.3 查看详细指标

在服务概览里,各种图表展示了各种关键指标的详细情况。

图表有趋势图、饼状图两种类型。

饼状图用以显示变化不大的数据,例如下图的 HDFS 磁盘使用率,在鼠标未点击之前,只是显示了大



2 用户指南

概的使用率,当点击后则切换为详细的数据。



趋势图主要是用来展示变化频繁的指标,而且是随着时间变化而变化的。

例如下图的 CPU 使用量,在正常情况下是显示最近一小时的大致趋势,当需要进一步查看详细的趋势 时,点击图示右上角的放大镜图标,弹窗显示最近一小时的详细趋势,还可以在图上移动鼠标,了解 某个时间点具体的指标值。

还可以通过弹窗左上角的区间选择框,选择不同时间区间,来切换不同时间的指标趋势。



2.2.2 实例信息

当集群出现异常或者需要升级扩容时,用户需要对集群的主机进行分析,确定故障原因和性能瓶颈, 这需要系统提供关于主机的一系列指标展示。

用户可以对选取的集群查看集群内的主机 CPU,内存,磁盘,负载状态。

1 添加节点:对选取的集群进行节点添加,为集群扩容和升级提供易用的交互。



て で Cloud

2 搜索功能:使用主机名和主机 IP 进行复合查询。同时用户可以通过主机链接,跳转到 ECS 的主机详情页面。

当用户需要对系统进行进一步的分析时,如故障发生或者系统推送告警信息,用户可以利用实时指标 辅助分析,包括 CPU,网络,内存,磁盘读写,RPC 请求数和响应时间,磁盘使用率等。而主机的历史 趋势图则可以为用户提供时间维度上的指标趋势,方便用户提前发现问题或升级集群。

在使用过程中,还可以对时间区间、CPU 使用量、RAM 使用量、磁盘剩余量、网络负载,处理请求数 进行条件过滤,实现对某一区间不同主机的指标(实时指标和历史趋势)过滤。

2.2.2.1 进入实例信息

点击左边导航栏的实例信息,即可进入实例信息界面,显示了当前集群的所有实例信息和实例指标。 在实例信息页面,显示了所有实例的关键指标实时信息。

• 实例信息							🖶 实例信息	@ 历史趋势图
实例名称	清榆入实例名称	IP 地址	请输入IP地址	搜索	查			
指标条件	+ 过滤条件							
集群名称	实例名称	IP 地址	CPU 使用量	RAM 使用量	磁盘利用率	剩余磁盘容量	平均负载	网络平均负载
Hbase105	t73	192.168.1.73	3.2 % / 8(8) cores	13.25 GB / 32 GB	75.76%	8.56GB	0.1	接收 534.40 KB, 发送 180.15 KB,
Hbase105	t135	192.168.1.135	32.1 % / 8(8) cores	16.79 GB / 32 GB	49.5%	17.83GB	0.79	接收 620.19 KB, 发送 674.65 KB,
Hbase105	t142	192.168.2.142	3.6 % / 8(8) cores	12.49 GB / 32 GB	71.85%	9.94GB	0.19	接收 654.74 KB, 发送 857.21 KB,
Hbase105	t105	192.168.1.105	4.4 % / 8(8) cores	17.20 GB / 32 GB	44.97%	19.43GB	0.31	接收 103.70 KB, 发送 36.82 KB/s
•								•

在页面最右侧的操作栏,可以点击明细按键,可以查看主机更多实例的基本信息。

实例明细			
实例名称	t73	CPU	8(8)
IP 地址	192.168.1.73	内存	32 GB
操作系统	redhat7-centos7(x86_64)	Rack	/default-rack
组件版本	2.0	心跳	4 分钟前





2.2.2.2 筛选特定条件的主机

点击指标条件旁边的过滤条件按键,则弹出条件筛选框,我们可以按需填入条件,

点击搜索后,界面将显示同时符合所有条件的实例,方便我们预防和定位问题。

+ 过滤条件		
CPU 使用量	> ~	%
RAM 使用量	> ~	GB
磁盘剩余量	> ~	GB
网络平均负载	> ~	kb/s
处理请求数	> ~	/s
	关闭 确定	

点击指标条件旁边的过滤条件按键,则弹出条件筛选框,我们可以按需填入条件,可以通过 CPU 使用量、RAM 使用量、磁盘剩余量、网络平均负载和处理需求数来进行筛选

输入过滤条件完毕,点击搜索即可显示满足过滤条件的主机列表。

可以点击重置按键来清除所有过滤条件,或者点击特定过滤条件上的关闭按键,来自定义过滤条件, 界面将显示同时符合所有条件的主机,方便我们预防和定位问题。

	指标条件	〇 CPU 使用量 > 6 %	+ 过演条件						
Í	集群名称	实例名称	IP 地址	CPU 使用量	RAM 使用量	磁盘利用率	剩余磁盘容量	平均负载	网络平均负载
	Hbase105	t135	192.168.1.135	6.5 % / 8(8) cores	16.80 GB / 32 GB	49.5%	17.83GB	1.28	接收 112.09 KB, 发送 458.01 KB,
4									Þ
	共1条	10条/页 🗸 🗸	1 > 前往	1 页					

2.2.2.3 进入实例历史趋势

点击图上右侧的小图标 // 即可切换至主机历史趋势。点击历史趋势图后界面显示:

で Cloud	2 用户指南
• 历史趋势图	受 实例信息
实例名称 请输入实例名称 时间区间 最近 1 小时 ~	被实 重量
指标条件 + 过滤条件	
过滤结果 173 ~ 最近1小时 ~	
CPU 使用量	磁盘使用量
100% 80% 40% 20% 0% 20% 20% 20% 20% 2019-08-07 13:53:07 2019-08-07 14:10:07 2019-08-07 14:44:07	40 GB 30 GB 20 GB 10 GB 0 GB 0 GB 2019-08-07 13:53:07 2019-08-07 14:42:07
Idle User Nice System 1/0 Idle	Total Free

2.2.2.4 选择特定主机查看

点击实例名字旁边的文本框,界面将弹出当前集群的所有主机列表,选择对应主机,并选择时间区间, 可以查看该主机在特定时间内的历史趋势。

• 历史趋势图				
指标条件	十 过减条件			
实例名称	请输入实例 名称	时间区间 最近1小时	2 捜索 重置	
	t140			
	t160			
	t231			
	t36			
	t66			
4				



2.2.2.5 过滤符合历史条件的主机

在界面上部是条件筛选区域,即可以过滤在特定时间内指标符合条件的主机。

界面下部的是主机选择区域,可以通过下拉框选择符合条件的主机,及其指标的时间区间

和实例信息类似,可以通过筛选条件来过滤主机,这里不同的是,增加了时间维度,即可以过滤在一定时间内符合条件的主机。



2 用户指南



在这里选择了两个指标作为过滤条件,时间区间选择为最近两个小时,即过滤最近两个小时内符合 CPU 使用量大于 5%, RAM 使用量大于 10GB 的主机。

翼 己 Cloud							2 用户指南	<u>i</u>
• 历史趋势图	R							
指标条件	〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇	RAM 使用量 > 10 GB	× + 过滤条件					
实例名称	请输入实例名称	时间区间	最近2小时	~	搜索	重置		

点击搜索后,显示了符合条件的主机,一共有两台主机符合条件。

● 历史趋势图		
指标条件 CPU 使用量 > 5 % RAM 使用量 > 10 GB	十 过滤条件	
实例名称 请输入实例名称 时间区间	最近2小时 🗸	搜索 重置
	136	∧ 最近1小时 ∨
CPU 使用量	t36	磁盘使用量
100% -	t140	Вд
80% -		3.9 TB -
60%		2.92 TB -
40% -		1.95 TB -
20% -		1000 GB -
0%		0 GB
2019-03-28 09:35:07 2019-03-28 09:58:07 201	9-03-28 10:21:07	2019-03-28 09:35:07 2019-03-28 09:58:07 2019-03-28 10:21:07
Idle User Nice S	ystem I/O Idle	Total Free
11 昭之书		古左海田目
机箭贝彩		内存使用重
3 1		33 GB -
2.5 -		28 GB
2		19 GB
		14 GB -

可以通过切换主机和不同的时间区间,查看特定主机在不同时间下的指标趋势。



如果有多台主机符合条件,在浏览指标时,可以直接点击指标趋势图旁边的切换按键,快速切换不同 主机的趋势图。

2.3 预警管理

在集群使用过程中,磁盘利用率、主机 CPU 和内存、客户请求数往往是集群健康状态最关键的几个指标。用户通过这些指标,可以提前对磁盘进行规划,对业务使用情况进行评估,预防后续突发事件的发生。

预警管理为用户提供了以上这些指标的监控预警,用户可以对这些指标设置检查周期,警告、危险阀 值,以及达到阀值时的通知方式。

预警平台会按照用户设置的检查周期定时的扫描指标,当达到用户设置的阀值时,会及时通知用户, 方便用户进行后续的处理。

2.3.1 设置预警

预警管理可以对常见指标进行监控,当监控平台发现指标达到阀值时,会及时发送系统消息给用户, 用户可以查看了解系统的运行情况并对系统采取措施处理。



步骤如下:

- ① 点击添加预警规则,见图 2.3.1
- ② 选择预警规则,设置检查周期、警告阀值、危险阀值,见图 2.3.2
- ③ 查看消息列表新消息,浏览预警详细内容,见图 2.3.3
- ④ 对设置过的预警规则,进行修改、关闭/启动、删除,见图 2.3.4

♀ 当前集群: 20190215 ∨	<u>e</u> (4) s i	Ö			
● 预警管理 + 添加预警规则		步骤1:点击添加预警规则			
规则名称		集群名称	预警状态	创建时间	操作
		暫无数据			

图 2.3.1

う た 算 口 e Cloud	
	2 用户指南
• 预警管理 > 预警规则定义 步骤2:选择预警规则,设置检查周期	,警告阀值,危险阀值
→ 设置预警规则	
*预警规则 HBASE 集群内存预警 V	
* 检查周期 30 分钟	
* 警告阀值 70 %	
- 危险阀值 80 %	
提文 重要	



消息列表		×
选择级别 所有 🗸 查询	: 查看消息列表新消息,	浏览预警详细内容
消息	级别	时间
<mark>张警管理</mark> 您设置的预警规则[HBASE集群主机内存预警],已达到警告值[60%]	警告	16 小时前
根据HBase读写指标,推荐您使用(读少写多)配置	提示	2 天前
表操作 drop表 backupTest2	提示	2 天前
表操作 drop表 backupTest1	提示	2 天前
HBase已重启完毕	提示	3 天前
共 26 条 5条/页 ~ 〈 1 2 3 4 5	6 > 前往 1	页

关闭

图 2.3.3





2.3.2 查看集群后台操作进度

用户在任一页面的顶部,有下面两个图标,左侧为进度框,右侧为消息框。

图标 数字表示后台正在执行的服务和命令操作数量,图标 数字表示未读消息数。



点击 后,可以查看系统具体的服务和命令操作的进度情况:

选择状态 所有 🗸 🗸	查询			* 0 个操作正在运行
操作	状态		开始时间	运行时长
删除集群[2000]表[TEST231]的 emo_index]成功 正在删除集群[2000]表[TEST231]的 mo_index]	較引[id_d 평][id_de 100%	运行成功	2019-08-07 16:35:15	1分钟33秒
创建集群[2000]表[TEST231]的统 emo_index]成功 正在创建集群[2000]表[TEST231]的参 mo_index]	乾引[id_d 평l[id_de 100%	运行成功	2019-08-07 16:30:33	1分钟55秒

2.3.3 查看系统消息

系统会推送必要的信息给用户,以提示用户系统的状态。点击——后,可以查看系统消息。点击某条系 统消息,可以跳转到消息发生对应的模块页面

2 用户指南

消息列表		×
选择级别 所有 > 重词		
消息	级别	时间
根据HBase读写描标,[test4]表推荐您使用[读写均衡]配置	提示	20 分钟前
根据HBase读写指标,[test2]表推荐您使用[读写均衡]配置	提示	20 分钟前
根据HBase读写指标,[test3]表推荐您使用[读写均衡]配置	提示	20 分钟前
根据HBase读写指标,[test1]表推荐您使用[读写均衡]配置	提示	1 天前
根据HBase读写指标,推荐您使用[读少写多]配置	提示	14 天前

2.4 表信息管理

2.4.1 可视化表操作

可视化显示了当前表运行的基本信息,指标和列族的相关属性配置信息。在表的场景发生变化的时候, 推荐栏的配置信息会做出相关场景的优化配置推荐。与此同时系统会发出系统消息,提示用户做优化 修改。

可视化还显示了表对应的 region 的所属和生命活动信息和相关的队列,内存,请求, memstore,日志 等相关指标信息。

2.4.1.1 批量建表

hbase 上云,首先要做的是建表。云 HBASE 提供了批量建表功能,具体操作流程如下图: 1:准备好导入文件(语法遵从单行单句,语法目前只支持 create, drop, disable 3 种操作)。

create 'hbase_test1', {NAME=>'cf1'}, {NAME=>'cf2'}

create 'hbase_test2', {NAME=>'cf1'}, {NAME=>'cf2'}

注意:建表文件一个表一行。建议每个文件最大不超过100张表。

2: 点击导航栏图标



3: 选择文件建进行导入。

HBase Shell @ ① ①		Θ	F
Velcome to ⊞age She \$ 导入表结构 + 选择文件 □ 出措不4	白街		
上传			
() the reaction		3±48=	8
(19) 中文 (18)(為)	©2016中國电信版防有限公司公式算力公司版权所有 FICP留 12022351号 海脑电信业务经营许可业A2.81.82-20090001 大英云目贝 用户协议	法僅戶	98



4: 获得导入进度

HBase Shell 🛞 🔂 🔂	Θ	2
<pre>WRITE => 'false', PREFETCH_BLOCKS_ON_OPEN => 'false', COMPRESSION => 'SNAPPY', BLOCKCACHE ue', BLOCKSIZE => '65536', METADATA => {'IN_MEMORY_COMPACTION' => 'NONE'}}, {NAME => 'infol SIONS => '1', EVICT_BLOCKS_ON_CLOSE => 'false', NEW_VERSION_BEHAVIOR => 'false', KEEP_DELE LLS => 'FALSE', CACHE_DATA_ON_WRITE => 'false', DATA_BLOCK_ENCODING => 'NONE', TTL => '214 7', MIN_VERSIONS => '0', REPLICATION_SCOPE => '0', BLOOMFILTER => 'ROW', CACHE_INDEX_ON_WR 'false', IN_MEMORY => 'false', CACHE_BLOOMS_ON_WRITE => 'false', PREFETCH_BLOCKS_ON_OPEN lse', COMPRESSION => 'SNAPPY', BLOCKCACHE => 'true', BLOCKSIZE => '65536', METADATA => {'I RY_COMPACTION' => 'NONE'}}</pre>	=> ' TED 7483 TEE 7483 TE =>	'tr VER CE 364 => 'fa EMO
<pre>\$ create 'backupTest119' successfully \$ create 'backupTest120', {NAME => 'info0', VERSIONS => '1', EVICT_BLOCKS_ON_CLOSE => 'fa NEW_VERSION_BEHAVIOR => 'false', KEEP_DELETED_CELLS => 'FALSE', CACHE_DATA_ON_WRITE => 'fa DATA_BLOCK_ENCODING => 'NONE', TTL => '2147483647', MIN_VERSIONS => '0', REPLICATION_SCOPE ', BLOOMFLITER => 'ROW', CACHE_INDEX_ON_WRITE => 'false', IN_MEMORY => 'false', CACHE_BLOO WRITE => 'false', PREFETCH_BLOCKS_ON_OPEN => 'false', COMPRESSION => 'SNAPPY', BLOCKCACHE ue', BLOCKSIZE => '65536', METADATA => {'IN_MEMORY_COMPACTION' => 'NONE'}}, {NAME => 'info1 SIONS => '1', EVICT_BLOCKS_ON_CLOSE => 'false', NEW_VERSION_BEHAVIOR => 'false', KEEP_DELE LLS => 'FALSE', CACHE_DATA_ON_WRITE => 'false', DATA_BLOCK_ENCODING => 'NONE', TTL => '214</pre>	lse se => MS_0 => [, TED 748	', '0 N 'tr VER _CE 364
<pre>7', MIN VERSIONS => '0', REPLICATION SCOPE => '0', BLOOMFILTER => 'ROW', CACHE_INDEX_ON_WR 'false', IN_MEMORY => 'false', CACHE_BLOOMS_ON_WRITE => 'false', PREFETCH_BLOCKS_ON_OPEN lse', COMPRESSION => 'SNAPPY', BLOCKCACHE => 'true', BLOCKSIZE => '65536', METADATA => {'I RY_COMPACTION' => 'NONE'}} \$ create 'backupTest120' successfully \$ 导入结果: 共 11 条语句, 11 成功, 0 失败</pre>	ITE => ' N_ME	=> 'fa EMO

2.4.1.2 单表操作优化

• 建表,通过可视化页面进行建表操作。

表信息管理	基本信息				
新増 清空 删除表 🔇	* 表名称 user:test				
Q 请输入表名筛选	表属性 属性	值		描述	
huda:back40 huda:back5	Enabled	TRUE 🗸		Is the table enabled.	
huda:back6 huda:back7 huda:back8	Schema 新增	删除 🥑 操作后还需点击下方按	钮提交		
huda:back9	□ 列族	Property 属性	Value 值		推荐值
ns1:t1	C1	VERSIONS	1		1
		EVICT_BLOCKS_ON_CLOSE	false	\sim	false
۰	4				Þ
©2018中国	电信股份有限公司云计算分公司	版权所有 京ICP备 12022551号 增值电信业务	经营许可证A2.B1.B2	-20090001 天翼云首页	用户协议 法律声明

表信息管理	基本信息			
新增 清空 删除表	<	COMPACTION_POLICY	RatioBasedCi 🖂	RatioBasedCompac
Q 请输入表名筛选		添加		
huda:back40	共1条 < 1 >			
huda:back5				
huda:back6	Phoenix 列名 新	增 🚺 * Phoenix 建表主键默认为 id		
huda:back7				
huda:back8	名称	类型	长度	
huda:back9				
ns1:t1	共0条 〈 1 〉			
user:test				
user:test2		18 z	*	

• 修改表,通过可视化页面进行修改表操作。

表信息管理	基本信息 二级索	뤽 Region		
新增 清空 删除表 禁用	<	EVICT_BLOCKS_ON_CLOSE	false 🗸	false
○ 清輸入表名筛洗		NEW_VERSION_BEHAVIOR	false 🗸	false
		KEEP_DELETED_CELLS	FALSE 🗸	FALSE
huda:back5		CACHE_DATA_ON_WRITE	false 🗸	false
huda:back6		DATA_BLOCK_ENCODING	NONE ^	
huda:back7		IN_MEMORY_COMPACTION	NONE	BASIC
huda:back8 huda:back9		TTL	Prefix	2147483647
ns1:t1		MIN_VERSIONS	Diff	0
user:test		REPLICATION_SCOPE	Fast_Diff	0
4 >>	4	BLOOMFILTER	Pretix_Tree	ROW

2.4.2 可视化 SHELL

云 HBASE 在 web 页面模拟 hbase shell 界面的操作,用户在 web 页面即可进行 hbase shell 的相关命 令操作,目前基本支持除 exit 和 quit 以外的所有的 hbase shell 操作。具体相关命令,可以参考 hbase 官方文档。



2.5 二级索引

HBase 的一级索引就是 rowkey,只能通过 rowkey 进行检索。如果相对 hbase 里面列族的列列进行一些组合查询,就需要采用 HBase 的二级索引方案来进行多条件的查询。天翼云 Hbase 提供基于 phoen ix 的二级索引方案。

在使用 phoenix 二级索引前,应确保 hbase 表存在对应的 phoenix 表,可以使用数据可视化功能对表进行维护。

索引提供本地索引、全局索引、函数索引和覆盖索引等的创建功能。用户可根据数据场景自行选择索引类型。用户可以查看创建和修改删除索引的进度日志。

2.5.1 进入二级索引页面

点击数据可视化标签,选择想要维护的表,切换到索引选项卡:

♀ 当前集群: 20190215 ∨ 🗐 🔮		
表信息管理 新增 清空 删除表 募用 <	基本信息 二级索引 Region	
Q 请输入表名筛选	⑦ 请先在 基本信息 页面中新增 Phoenix 列名	
TEST231		
petest2		
test1		
test2		
test3		
test4		
4		
1		•

在基本信息新增列



2 用户指南

表信息管理	基本信息 二级索	引 Region		
新增清空删除表禁用	<	STORAGE_POLICY	Hot 🗸	ONE_SSD
Q 请输入表名筛选		添加		
TEST231	共1条 〈 1 〉			
petest1				
petest2	Phoenix 別名 新	A * Phoenix 建素主a # 新认为 id	点击新增按钮,添加列,提交	
test1	постіх узд			
test2	名称	業型	长度	
test3				
test4	c1	UNSIGNED_INT ~	8	删除
	共1条 < 1 >			
4	4			Þ



2.5.2 新建索引

点击新建索引按钮弹出创建索引弹窗

	基本信息	二级索引	Region						
>				索引列表	索引日志		/		所增索引
	索引名称	3	素引信息				索引类型	操	l'E
				智力	无数据				
	4								►

输入索引名称,选择索引类型,点击索引字段栏可以弹出字段供选择,点击保存即开始创建索引:

创建索引		×
* 索引名称	id_demo_index	
* 索引类型	本地索引 ~	
*索引字段	INF01.c1 O	
	取消 保存	

创建索引提交后等待完成:



♀ 当前集群:	20190215 🗸 🗐 🔮			
基本信息	二级索引 Region			
>		繁引例表 素引日志	1	新増索引
索引名称	索引信息		素引类型	操作
		智无数据		
4			✓ 成功提示 任务已提交,请稍候查询结果	×
,				

点击消息按钮查看进度信息:

「后台操作				×
选择状态 所有 > 查询				* 0 个操作正在运行
操作	状态		开始时间	运行时长
创建集群[2000]录[TEST231]的索引[id_d emo_index]成功 正在创建集群[2000]汞[TEST231]的索引[id_de mo_index]	100%	运行成功	2019-08-07 16:30:33	1分钟55秒
HBase外部导入	100%	运行成功	2019-08-07 14:34:25	2分钟53秒
HBase外部导入	100%	运行失败	2019-08-07 14:31:43	1分钟48秒
HBase外部导入	100%	运行失败	2019-08-07 14:30:28	1分钟32秒
HBase外部导入	100%	运行失败	2019-08-07 14:23:39	1分钟51秒

索引已经成功建立:

基本信息	二级索引	Region						
>			索引列表	索引日志				
								新増索引
索引名称		索引信息				索引类型	握	們自
id_demo	_index	INFO1.c1				本地索引	删除	

2.5.3 删除索引

选择想要删除的索引,点击删除按钮,在弹出的确认窗口中选择确认:



2 用户指南

×

基本信息	二级索引	Region			
		索引列表	索引日志		新增索引
索引名称	素	删除索引	\times	索引类型	操作
id_demo_ind	lex IN	是否删除索引 id_demo_index?		本地索引	删除
		取消 确定			
4					Þ

与创建索引类似,可以看到提交提示与进度消息:

后台操作

选择状态 所有 🗸 🧃				* 0 个操作正在运行
操作	状态		开始时间	运行时长
删除集群[2000]表[TEST231]的索引[id_d emo_index]成功 正在删除集群[2000]表[TEST231]的索引[id_de mo_index]	100%	运行成功	2019-08-07 16:35:15	1分钟33秒
创建集群[2000]表[TEST231]的索引[id_d emo_index]成功 正在创建集群[2000]表[TEST231]的索引[id_de mo_index]	100%	运行成功	2019-08-07 16:30:33	1分钟55秒
HBase外部导入	100%	运行成功	2019-08-07 14:34:25	2分钟53秒
HBase外部导入	100%	运行失败	2019-08-07 14:31:43	1分钟48秒

2.5.4 查看索引操作日志

进入索引日志:

Cloud					
					2 用户指南
基本信息 二级索引	Region				
		索明表 索明表		新繼泰引	
索引名称	索引信息		索引类型	状态	
id_demo_index	"INFO1"."c1"		本地索引	创建成功	
id_demo_index	"INFO1"."c1"		本地索引	删除成功	
共2条 10条质 >	〈 1 〉 前往 1 页				

2.6 HBASE 冷热存储

大数据存储中一种常见的场景就是冷热数据,冷数据由于其访问频率低的特性,可以存放到价格低磁 盘速率慢的存储设备上,热数据由于访问频率高,基于性能的考虑可以存放到磁盘速率快的存储设备 上。在实际使用中,根据数据的热度选择不同的存储设备可以提升集群整体资源的使用效率,同时降 低使用成本。

2.6.1 WAL 日志热存储

HBase 的 Write Ahead Log (WAL)为数据的写入提供了可靠性的保证。每一个业务数据的写入操作 (PUT / DELETE)执行前,都会先记录到 WAL 中。对 WAL 进行热存储实施,将极大的提升 HBASE 写入请求响应速度。

用户在购买高性能存储设备后,可以直接选择设置 WAL 热存储,或是等待监控平台对系统读写能力的 分析后,判断是否需要设置 WAL 热存储。

用户可以在配置管理内进行配置日志存储策略:
							2 用户	^白 指i
▶ 配置管理							可修改参数	修
参数名 wal	参数值	请输入参数值		搜索				
参数名	参数分类	运行参数值	修改值	推荐值	是否需要重启	可修改范围	参数含义	
hbase.regionserver.wal.durable.syn c	RegionServe r/性能	false		false	是	true, false	每一条wal是否持… ▼	
hbase.wal.provider	RegionServe r/性能	multiwal		multiwal	是	asyncfs,filesystem,m ultiwal	单个regionserver 🔻	
hbase.wal.regiongrouping.numgro ups	RegionServe r/性能	4		4	是	0~20	单个regionserver 🔻	
hbase.wal.storage.policy	RegionServe r/性能	НОТ		НОТ	是	ALL_SSD,One_SSD,H ot,Warm,Cold	ALL_SSD:所有副 🔻	

点击修改后,对 wal 存储策略进行修改,见下图

┃ 修改参	数		×						
參数名	nbase.wal.storage.policy								
范围	ALL_SSD,One_SSD,Hot,Warm,Cold								
▪ 修改值	HOT	 ▲ 撤销 							
	ALL_SSD								
	One_SSD	存							
r/性能	Hot	4	분						
.,	Warm								
r/性能	Cold	HOT	是						

2.6.2 表冷热分离

用户在使用集群建表前或者系统运行中,可以对表的重要度和业务需求进行冷热表划分,

使用频率低的表可以在建表时设置为冷表,使用频率高的表可以在建表时设置为热表。

具体操作步骤如下:

第一步:进入表可视化操作(请先熟悉 2.4.1.2 单表操作优化章节)

第一步:在表的列族属性选择 STOREAGE_POLICY 进行设置,见下图。

を Cloud 基本信息 二级索引	Region		2 用户指南
▼ 列族	Property 履性	Value 值	推荐值
info0	STORAGE_POLICY	Hot	ONE_SSD

2.7 配置管理

用户可以通过修改配置参数以达到更好的性能或者稳定性,在配置管理里,用户可以选择修改某些参数提交,也可以查看修改历史,选择参数进行回滚提交。

2.7.1 配置修改

第一步:点击左边的导航栏的配置管理,进入配置管理列表页面。

● 配置管理							可修改参数	修改历史
参数名 请输入参数名	参数值	请输入参数值		搜索				
参数名	参数分类	运行参数值	修改值	推荐值	是否需要重启	可修改范围	参数含义	操作
ha.zookeeper.session-timeout.ms	HBase/性能	45000		45000	是	1~3600000	ZK连接超时。HBa 🔻	修改
dfs.client.datanode-restart.timeout	HBase/性能	60		60	是	1~65535	从接收到DataNod 🔻	修改
dfs.client.file-block-storage-locatio ns.timeout.millis	HBase/性能	600000		600000	是	1000~6000000	分布式文件系统中 🔻	修改
dfs.datanode.socket.write.timeout	HBase/性能	600000		600000	是	1~600000	DataNode等待so 🔻	修改
hbase.client.pause	HBase/性能	100		100	是	1~2147483647	一般客户端暂停值 ▼	修改
hbase.client.retries.number	HBase/性能	35		35	是	1~2147483647	最大重试次数。用 🔻	修改
hbase.client.scanner.caching	HBase/性能	100		100	是	1~2147483647	Scanner一次从服 🔻	修改
hbase.client.keyvalue.maxsize	HBase/性能	10485760		10485760	是	0~2147483647	—个KeyValue实 🔹	修改
hbase.server.thread.wakefrequency	HBase/性能	10000		10000	是	1~2147483647	服务线程的sleep 🔻	修改

第二步:在相关需要修改的配置上点击"修改"

hbase.client.pause	HBase/性能	100	100	是	1~2147483647	一般客户端暂停值 🔻	修改

第三步:输入修改值,点击"暂存"

2 用户指南 修改参数 × 参数名 hbase.client.pause 范围 1~2147483647	
修改参数 × 参数名 hbase.client.pause 范围 1~2147483647	
参数名 hbase.client.pause 范围 1~2147483647	
范围 1~2147483647	
* 修改值 1000 1000	
取消 誓存	

第四步:拖到底部,点击"提交"

 配置管理 					可修改参	数 修改历史
参数名 hbase.client.pa	多数值 请输入参数	值搜索				
参数名	参数分类 运行参数	收值 修改值 推荐值	且 是否需要重启	可修改范围	参数含义	操作
hbase.client.pause	HBase/性能 100	1000 100	是	1~2147483647	一般客户端暂停值	▼ 修改
共1条 10条页 ∨	〈 1 〉 前往 1	<u></u> д				

提示"提交成功后",切换到"修改历史",可以看到修改记录

 ■ 配置管理 						可修改参数	修改历史
参数名 请输入参数名	版本号	E	日期 ① 开始 建奏 回滾	_{行期} 至	结束日期		
历史版本号	参数名	参数分类	是否需要重启	历史参数值	修改人	修改时间	生效时间
O TEMP-20190325150440	ha.zookeeper.session-timeout.ms	HBase/性能	是	45000	admin	2019-03-25 15:04:40	
共1条 10条/页 ∨ 《	1 〉 前往 1 页						
4)

这里历史号以 TEMP 开头, 生效时间为空, 提示需要重启才能生效。



2.8 场景推荐配置优化

系统会根据 Hbase 实际运行情况,周期性地给用户提供集群级别和表级别的最优参数推荐。

日志分析引擎以 15 天为周期收集集群运行指标和检查 Hbase 运行状态,给出推荐系统配置。

2.8.1 集群级别推荐配置优化

第一步: 先在消息列表里查看相关消息

消息列表		×
选择级别 所有 > 查询		
消息	级别	时间
根据HBase读写指标,[test4]表推荐您使用[读写均衡]配置	提示	2 小时前
根据HBase读写指标,[test2]表推荐您使用[读写均衡]配置	提示	2 小时前
根据HBase读写指标,[test3]表推荐您使用[读写均衡]配置	提示	2 小时前
根握HBase读写指标,[lest1]表推荐您使用[读写均衡]配置	提示	1 天前
根据HBase读写指标,推荐您使用[读少写多]配置	提示	14 天前
共 5 条 5条 (五) · · · · · 前往 1 页		
关闭		

第二步:点击消息进入相关配置界面(集群配置)



消息列表	×
选择级别 所有 > 查询	
消息	时间
表操作 提示 drop 表 backuptest	7 天前
根据HBase读写指标,推荐您使用读多写少配置集群级别推荐消息 提示	10 天前
表操作 提示 drop 表 pe_read_write	11 天前
表操作 提示 drop 表 pewrite	11 天前
表操作 提示 drop 表 peread	11 天前
共 20 条 5条/页 ~ < 1 2 3 4 > 前往 4 页	
	关闭

第三步: 在相关配置界面上点击"修改"

天翼云 e Cloud

♀ 当前集群: 20190215	~ 🗳 🕻	9 5						
 配置管理 							可修改参数修改	历史
参数名 请输入参数名	3	参数值 请编	入参数值	搜索				
参数名	参数分类	运行参数值	修改值	推荐值	是否需要重启	可修改范围	息 古 修 仪 参数含义 描	l/F
ha.zookeeper.session-ti meout.ms	HBase/性能	45000		45000	是	1~3600000	ZK连接超时。HB 🔹 🧌	瞅
dfs.client.datanode-rest art.timeout	HBase/性能	60		60	是	1~65535	从接收到DataNod 🔻 🧌	皲
dfs.client.file-block-stora ge-locations.timeout.mill is	HBase/性能	600000		600000	是	1000~6000000	分布式文件系统中 🔻 🎁	暾
dfs.datanode.socket.writ e.timeout	HBase/性能	600000		600000	是	1~600000	DataNode等待soc 🔻 🧌	敬
hbase.client.pause	HBase/性能	100		100	是	1~2147483647	一般客户端暂停值 🔻 🧌	B政
hbase.client.retries.num ber	HBase/性能	35		35	是	1~2147483647	最大重试次数。用 🔻 🧌	礅
hbase.client.scanner.ca ching	HBase/性能	100		100	是	1~2147483647	Scanner—次从服… 🔻 🇌	敬
hbase.client.kevvalue.m								

第四步: 输入修改值, 点击"暂存"

天翼 Cloud	
	2 用户指南
修改参数 ×	
参数名 dfs.datanode.socket.write.timeout	
范围 1~600000	
*修改值 600000 《 撤销	
取消 暫存	
第五步:拖到底部,点击"提交"	

 配置管理 					可修改参数	修改历史
参数名 kookeeper 💿	参数值 请输入参数值	搜索				
参数名 参数分类	运行参数值 修改值	推荐值	是否需要重启	可修改范围	参数含义	操作
ha.zookeeper.session-ti HBase/性能 meout.ms	45000 45001	45000	是	1~3600000	ZK连接超时。HB	▼ 修改
zookeeper.session.time HBase/性能 out	90000	90000	是	1~2147483647	Zookeeper连接超	▼ 修改
共2条 10条页 ∨ < 1	> 前往 1 页	提交				
A						Þ

提示"提交成功后",切换到"修改历史",可以看到修改记录

 配置管理 						2 用户 ^{可修改参数}	□ 指雨
参数名 请输入参数名	版本号 版本号		日期 ① 开 2 一 元 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	白期 至	结束日期		
历史版本号	参数名	参数分类	是否需要重启	历史参数值	修改人	修改时间	生效時
○ TEMP-20190325150440	ha.zookeeper.session-timeout.ms	HBase/性能	是	45000	admin	2019-03-25 15:04:40	
共1条 10条/页 ~ <	1 〉 前往 1 页						

这里历史号以 TEMP 开头,生效时间为空,提示需要重启才能生效。

第六步:重复第一步打开消息列表,点击消息进入集群管理页面

消息列表		×
选择级别 所有 > 查询		
消息	级别	时间
配置已更新,需要重启以生效	提示	几秒前
表操作 drop表 TEST231	提示	6 小时前
根据HBase读写指标,[test2]表推荐您使用[读写均衡]配置	提示	1 天前
根据HBase读写指标,[test1]表推荐您使用[读写均衡]配置	提示	4 天前
根据HBase读写指标,推荐您使用[读少写多]配置	提示	4 天前
共13条 5条页 ~ 1 2 3 > 前往 1 页		
关闭		



第七步:在 HBase 管理页面点击重启



收到消息"Hbase 已重启完毕",则修改生效。

■消息列表		×
选择级别 所有 > 查询		
消息	级别	时间
HBase已重启完毕	提示	1 分钟前
HBase已重启完毕	提示	1 分钟前
配置已更新,需要重启以生效	提示	7 分钟前
表操作 drop 表 TEST231	提示	7 小时前
根据HBase读写指标,[test2]表推荐您使用[读写均衡]配置	提示	1 天前
共15条 5条页 > 《 1 2 3 > 前往 1 页		

关闭

e Cloud							2 用户	指南
HBase 数据库	♀ 当前集群: Hbase231 ∨	Eº 🗗 🖬 🛱						
集群列表	 配置管理 						可修改参数	修改历史
▼ 集群信息 集群病院	参数名 请输入参数名	版本号版本号	日期) 开始日期	至	结束日期	搜索	夏 回滾
实例信息	历史版本号	参数名	参数分类	是否需要重启	历史参数值	修改人	修改时间	生效时间
HBase 概览	ACTIVE-20190807165601	dfs.client.datanode-restart.time out	HBase/性能	是	60	admin	2019-08-07 16:50:41	2019-08-07 16:56:02
表信息管理 预警管理	ACTIVE-20190802170436	ha.zookeeper.session-timeout. ms	HBase/性能	是	45001	admin	2019-08-02 17:04:13	2019-08-02 17:04:37
访问控制 配置管理	ACTIVE-20190802170341	ha.zookeeper.session-timeout. ms	HBase/性能	足	45000	admin	2019-08-02 17:02:57	2019-08-02 17:03:41
▶ 备份还原	共3条 10条页 > <	1 〉 前往 1 页						

这里历史版本号变成 ACTIVE 开头,生效时间也不为空。

2.8.2 表级推荐配置优化

第一步:打开消息列表(同 2.8.1 第一步)

第二步:点击消息进入相关配置界面(表级配置)

消息列表			×
选择级别 所有 > 宣询			
消息	级别	时间	
HBase已重启完毕	提示	4 分钟前	
HBase已重启完毕	提示	4 分钟前	
配置已更新,需要重启以生效	提示	10 分钟前	
表操作 drop表 TEST231	提示	7 小时前	
根据HBase读写指标,[test2]表推荐您使用[读写均衡]配置	提示	1 天前	
共15条 5条页 > 〈 1 2 3 > 前往 1 页			
关闭			

第三步: 在相关配置界面上修改属性值



表信息管理	基本信息	二级索引 Region		
新增 清空 删除表 禁用	< Schema	新増 删除 🥑 操作后边	还需点击下方按钮提交	
Q 请输入表名筛选	▼ 列族	Property 属性	Value 值	推荐值
TEST231 petest1	f1	STORAGE_POLICY	ONE_SSD 🗸	ONE_SSD
test1		SPLIT_POLICY	IndexRegion5 ∨	IndexRegionSplitPo
test2		VERSIONS	1	4
		EVICT_BLOCKS_ON_CLO	false 🗸	false
		NEW_VERSION_BEHAVI	OR false ~	false
		KEEP_DELETED_CELLS	FALSE \lor	FALSE

	NEW_VERSION_BEHAVIOR	false 🗸	false
	KEEP_DELETED_CELLS	FALSE V	FALSE
	CACHE_DATA_ON_WRITE	false 🗸	false
	DATA_BLOCK_ENCODING	NONE V	
	IN_MEMORY_COMPACTION	BASIC V	BASIC
	TTL	2147483647	2147483647
Þ. d			Þ

第四步:拖到底部点击"提交"按钮

表信息管理	基本信息 二级感	형태 Region		
新增清空删除表禁用	<	COMPACTION_POLICY	RatioBasedCr $$	RatioBasedCompac
Q 请输入表名筛选		IS_MOB	true 🗸	true
TECT221		HSTORE_ENGINE	DefaultStoreE 🗸	DefaultStoreEngine
petest1		MOB_COMPACT_PARTITION_POLICY	DAILY V	DAILY
test1		添加		
test2	共1条 〈 1 〉			
	Phoenix 列名 新	<mark>増</mark> * Phoenix 建表主键默认为 id		
	名称	类型	长度	
	共0条 〈 1 →		_	
		提考	2	
∢ →	4			Þ

提示成功则配置修改完成。

2.9 安全访问控制

具体业务场景如下

2 用户指南



- 用户下载客户端配置后,通过自购的 ECS 配置后访问已购买的 HBase 集群时,需要配置访问 控制,放开 HBase 集群对自购 ECS 的 IP 和端口的访问限制。
- 用户开通外网访问和 Rest 服务后,需要配置访问控制,将本机的公网 IP 加入到访问控制中, 以方便调试。

2.9.1 配置访问控制

点击左侧导航栏的访问控制,进入配置页面

• 访问控制 + 7	添加网络白名单 🚽			
IP 🔶				
172.168.235.1	172.168.3.12	127.0.0.1		

点击"增加网络白名单"后,弹出配置对话框。

在对话框中输入需要放开的内/外网 IP 地址(支持 ipv4 和 ipv6)。

▌添加网络白名单	1	×
* 网络访问白名单	172. 168. 253. 1	
	IP白名单设置为 127.0.0.1 或者 ::1 代表禁止所有地址访问。多个白名单之间 用逗号或换行分隔。 <mark>不建议使用 0.0.0.0</mark>	
	关闭 确认	

2.10 自助导入导出

2.10.1 集群迁移

集群迁移发生在如下业务场景:



- 业务拆分,将部分业务迁移到新的 HBase 集群
- 集群缩容

Cloud

• 双集群备份切换

云 hbase 目前支持对同租户下的新建集群实行一键迁移,具体流程如下:

点击左侧导航栏的集群迁移,进入集群迁移列表页面,见下图

● 集群迁移	+ 添加迁移任务					S
ID	任务名称	目标集群	任务状态	开始时间	结束时间	备注
117	导入4	hdfs://192.168.1.231:8603/user/hbase/backup/0807/ 527/backup_1565158863453	✔ 导入成功	2019-08-07 14:35:56	2019-08-07 14:37:18	
116	导入3	hdfs://192.168.1.231:8603/user/hbase/backup/0807/ 527	■ 导入失败	2019-08-07 14:33:13	2019-08-07 14:33:31	
115	导入2	hdfs://192.168.1.231:8603/user/hbase/backup/0807/ 527	■ 导入失败	2019-08-07 14:31:59	2019-08-07 14:32:00	测试备份导入
114	导入1	hdfs://192.168.1.231:8603/user/hbase/backup/0807	■ 导入失败	2019-08-07 14:25:10	2019-08-07 14:25:30	导入231的备份1

点击按钮添加迁移集群,填写迁移任务。

■ 添加迁移任务					
*任务名称	迁移当前集群]			
目标集群	Hbase231]			
任务备注	Hbase231				
	关闭 一键迁移				

目标集群仅支持相同资源池下同一账户的不同云 HBASE 集群。如有其它特殊需求,请参考后台导入导 出章节进行处理。填写好迁移任务信息后,点击按钮一键迁移,集群会对当前源集群创建快照,并将 集群全量备份到目标集群的 HDFS 上。然后以此全量备份对目标集群做还原。

注意事项

1. 迁移前请确认目标集群的 HDFS 容量为备份数据的两倍,防止迁移过程中出现容量写满的情况

2. 增量迁移为全量备份,为避免数据遗漏。建议在迁移期间暂停业务数据录入集群



- 用户无需在目标集群创建表,迁移会自动创建和源集群一样的表,包括分区的信息。客户自建的 目标表可能会和源表分区很不一致,这可能会导致迁移完成之后,目标表会进行频繁的 split、 compaction,如果表的数据量十分庞大,可能会导致这个过程非常耗时
- 4. 如果源表带有 coprocessor 和索引, 在创建目标表的时候可能会报错, 因为目标集群不包含对应 的 coprocessor
- 5. 如果迁移无论成功失败,迁移程序会自动清理当前备份数据

2.10.2 后台导入导出

具体业务场景如下

- 自建 HBase 集群迁移上云 HBASE
- 跨资源池迁移
- 迁移部分业务表
- 业务拆分,将部分业务迁移到离线集群的 hdfs
- 退订,将数据导出到对应 ECS

注意:迁移前请确认目标集群的 HDFS 容量,防止迁移过程中出现容量写满的情况

场景	迁移方式	环境
企业版单表, 或者少量表, 数 据量小于 5TB 导出	CopyTable 命令	需要确保其它集群的 vpc 网络与当前集群网络互通。 建议停止应用对集群对应表
		的写操作。
企业版全库或者部分表导出	Export 命令	需要确保其它集群的 vpc 网 络与当前集群网络互通。 建议停止应用对集群对应表 的写操作。
单节点表数据导出	snapshot	需要确保其它集群的 hdfs 或 者兼容 S3 协议接口可访问



2 用户指南

企业版全库或者部分表导出	Backup 命令	需要确保其它集群的 hdfs 或
		者兼容 \$3 协议接口可访问

具体步骤:

(1) 下载 hbase client 客户端

在集群列表内点击需要导入或者导出的集群明细,进入明细页面。

 ● 集群列表 > 明细 	概览	续费	退订
▶ 基本信息			
▶ Master节点信息			
▶ Core节点信息			
		1	
▼ 配置信息 本地开发	補置 有	印端安装包	本地开发包

点击按钮客户端安装包下载安装包文件到本地。您可通过 ftp 软件上传客户端安装包到对应可连接到 您购买的 ecs 机器上。(请确认该 ecs 与导入导出集群对应的 vpc 是否可以联通)

(2) HBase 数据导出到 HDFS 或者本地文件

在 hbase client /bin 目录下运行

hbase org.apache.hadoop.hbase.mapreduce.Export emp file:///usr/hbase_data/bak

HBase 数据导出到本地文件

hbase org. apache. hadoop. hbase. mapreduce. Export emp /hbase/emp_bak

导入 hbase 表(#默认不写 file://的时候就是导出到 hdfs 上了))

将 hdfs 上的数据导入到备份目标表中

\$ hbase org.apache.hadoop.hbase.mapreduce.Driver import emp_bak /hbase/emp_bak/*

将本地文件上的数据导入到备份目标表中

hbase org.apache.hadoop.hbase.mapreduce.Driver import emp_bak file:///usr/hbase_data/bak/*

(3) 导出时可以限制 scanner. batch 的大小



如果在 hbase 中的一个 row 出现大量的数据,那么导出时会报出 ScannerTimeoutException 的错误。这时 候需要设置 hbase. export. scanner. batch 这个参数。这样导出时的错误就可以避免了。

hbase org.apache.hadoop.hbase.mapreduce.Export -Dhbase.export.scanner.batch=2000 emp
file:///usr/hbase_data/bak

(4) 为了节省空间可以使用 compress 选项

hbase 的数据导出的时候,如果不适用 compress 的选项,数据量的大小可能相差 5 倍。因此使用 compress 的选项,备份数据的时候是可以节省不少空间的。

hbase org.apache.hadoop.hbase.mapreduce.Export -Dhbase.export.scanner.batch=2000 -D mapred.output.compress=true emp file:///usr//hbase_data/bak

命令说明:

2.10.1. hbase backup create full/incremental path

full:创建全量备份

incremental:创建增量备份

path: 备份到的文件系统以及对应的路径

可以跟的选项: -t table;如果不写就是全部的表都备份

-w 后面跟的数字,表示多少个工作任务

-s 备份的 set

-b 控制备份的带宽

2.10.2. hbase restore path backupid

path: 备份的路径

backupid: 备份的 id;

后面可以的选项: -o 覆盖之前的表

-t 备份的表,其他选项类似

2.10.3.hbase backup merge <backup_ids>

把一些 backup 的 id 进行 merge, 可以是每日的 backup merge 成一个;



2.10.4.hbase backup progress $<\!backup_id\!>$

展示 backup 进度

(5)迁移部分表业务,请联系客户经理。



3.1 常用概念

| 过滤器

过滤器提供了非常强大的特性来帮助用户提高 HBase 处理表中数据的效率。用户不

仅可以使用 HBase 中预定义好的过滤器,而且可以实现自定义的过滤器。

| 协处理器

允许用户执行 region 级的操作,并且可以使用与 RDBMS 中触发器类似的功能。

l Client

客户端直接面向用户, 可通过 Java API、HBase Shell 或者 Web 可视化 Shell 访问服务端, 对

HBase 的表进行读写操作。本文中的 HBase 客户端指 HBase client 的安装包或者 Web 可视化 Shell, 里面包含通过 Java API 访问 HBase 的样例代码。



3.2 开发流程

开发流程

1. 了解基本概念 在开始开发应用前,需要了解 HBase 的基本概念,了解场景需求,设计表

等。

2. 准备开发和运行环境 HBase 的应用程序当前推荐使用 Java 语言进行开发。可使用 Eclipse

工具。

3. 准备工程 HBase 提供了不同场景下的样例程序,您可以导入样例工程进行程序学习。

4. 根据场景开发工程,可根据提供了 Java 语言的样例工程,包含从建表、写入到删除表

全流程的样例工程。

5. 编译并运行程序, 用户将开发好的程序编译并提交运行。

6. 查看程序运行结果,程序运行结果会写在用户指定的路径下。用户还可以通过表可视化查看应用读 写表的情况。

3.3 环境准备

3.3.1 开发和运行环境简介

在进行二次开发时,要准备的开发和运行环境如下表所示。

准备项	说明
操作系统	开发环境:Windows 系统或 Linux 系统,推荐 Windows 7 以上



	版本。
	运行环境: Windows 系统或 Linux 系统。
安装 JDK	开发和运行环境的基本配置。版本要求如下:
	云 HBASE 的服务端和客户端仅支持自带的 0racle JDK
	(版本为1.8.161),不建议替换。
	对于客户应用需引用 SDK 类的 Jar 包运行在客户应用进程中
	的,
	支持 Oracle JDK。
	I Oracle JDK: 支持 1.8.161 版本。
安装和配置 Eclipse	用于开发 HBase 应用程序的工具。版本要求:4.8 或以上。
	说明
	I 若使用 IBM JDK, 请确保 Eclipse 中的 JDK 配置为 IBM JDK。
	若使用 0racle JDK,请确保 Eclipse 中的 JDK 配置为 0racle
	JDK.
	Ⅰ 不同的 Eclipse 不要使用相同的 workspace 和相同路径下
	的示例工程。
安装 Junit 插件	开发环境的基本配置。
下载安装开发包	"登录 web hbase 控制台,选择集群,进入集群明细,定位
	到配置信息页签,下载本地开发包和本地配置文件。



3.3.2 配置并导入样例

背景信息

HBase 客户端安装程序目录中包含了 HBase 开发样例工程,将工程导入到 Eclipse 开始样

例学习。

前提条件

确保本地 PC 的时间与云 HBASE 集群的时间差要小于 5 分钟,若无法确定,请联系系

统管理员。

操作步骤

步骤 1 获取客户端解压文件 "HBase" 目录下

的样例工程"hbase-example"。

步骤 2 导入样例工程到 Eclipse 开发环境。

1. 选择"File > Import > General > Existing Projects into Workspace > Next >

Browse"。

显示"浏览文件夹"对话框。

2. 选择样例工程文件夹,单击"Finish"。

步骤 4 设置 Eclipse 的文本文件编码格式, 解决乱码显示问题。

1. 在 Eclipse 的菜单栏中,选择 "Window > Preferences"。

弹出 "Preferences" 窗口。

 2. 在左边导航上选择"General > Workspace",在"Text file encoding"区域,选中 "Other",并设置参数值为"UTF-8",单击"Apply"后,单击"OK",如图



所示。

下图设置 Eclipse 的编码格式

	Workspace	⟨¬ ▼ ¬		
> User Storage Servic ^ Web Browser	See <u>'Startup and Shutdown'</u> for workspace start	tup and shutdown preferences.		
> Workspace	Refresh using native books or polling			
Ant	Refresh on access			
Cloud Foundry	Always slose uprelated projects without pror	mot		
Code Recommenders	Always close unrelated projects without profi	npt		
Data Management	Workspace save interval (in minutes): 5			
Gradle				
Help	Window title			
Install/Update	Show workspace name: hbaseworkspace	ie in the second se		
Java		~		
Java EE	Show perspective name			
Java Persistence	Show full workspace path: D:\hbaseworks	space		
JavaScript	✓ Show product name			
JSON				
LiClipse				
Maven	Open referenced projects when a project is op	pened: Prompt V		
Mylyn	Report unknown project nature as:	Report unknown project nature as:		
Oomph				
Plug-in Development				
Remote Systems	Command for launching system explorer: expl	lorer /E,/select=\${selected_resource_loc}		
Run/Debug				
Server				
leam	Text file encoding	New text file line delimiter		
Terminal	O Default (GBK)	Default (Windows)		
Validation	● Other: UTF-8 ∨	O Other: Windows ∨		
Web				
Web Services				
Web Services		Restore Defaults Apply		
Web Services XML V		Restore Defaults Apply		

-----结束

3.3.3 开发 HBASE 应用

3.3.3.1 典型场景说明

通过实例说明,我们可以初步掌握 HBase 的开发过程,对关键的接口函数和指令有所了解。

假定用户开发一个应用程序,用于网址管理上网用户的用户信息,如下表所

示,业务流程如下:

| 创建用户信息表。



| 在用户信息中新增用户的邮箱、账号、工作等信息。

| 根据用户编号查询用户账号和邮箱。

I 根据用户账号进行查询。

| 查询年龄段在[16-40]之间的用户信息。

| 数据统计,统计用户信息表的人员数、年龄最大值、年龄最小值、平均年龄。

| 用户销户, 删除用户信息表中该用户的数据。

| 删除用户信息表。

用户信息

主键 账号 邮箱 年龄 工作

1000101 molika tt@hotmail.cn 29 IT

1000102 kouda kouda@126.com 24 销售

1000103 liusha liusha@173.com 36 经理

1000104 baifei baifei@189. 16 学生

合理地设计表结构、行键、列名能充分利用 HBase 的优势。本样例工程以唯一主键作为

RowKey,列都存储在 info 列族中。

HBase 是以 RowKey 为字典排序的分布式数据库系统, RowKey 的设计对性能影响很大,

具体的 RowKey 设计参考 4.7 rowkey 设计原则。

序号	步骤	代码实现
1	创建用户信息表	请参考 5.3 创建表代码示例
2	用户信息中新增用户的邮箱、账号、工作等信息	请参考 5. 6 插入数据代码示 例



3 开发指南

3	根据用户编号查询用户账号和邮箱	请参考5.8使用Get读取数据
		代码示例
4	根据用户账号进行查询	请参考 4.8 使用过滤器
		Filter 代码示例
5	查询年龄段在[16-40]之间的用户信息	请参考 4.8 二级索引代码示
		例
6	用户销户,删除用户信息表中该用户的数据	请参考 4.8 删除数据代码示
		例
7	删除用户信息表	请参考 4.8 删除表代码示例

3.3.3.2 数据压缩与编码

天翼云支持 LZO GZ LZ4 SNAPPY ZSTD NONE,其中 NONE 就代表不开启压缩。系统默认开启 LZ4. 云 HBASE 支持编码方式有 Prefix | Diff | Fast_Diff | Prefix_Tree。DIFF encoding 对小 kv 场景使用效果较好,可以减少 2-5 倍的数据量。我们推荐 DATA_BLOCK_ENCODING 使用 diff

3.3.3.3 读写取优化

BlockSize 设置用户平均读取数据的大小。理论上讲,如果用户平均读取数据的大小较小, 建议将块大小设置较小,这样可以使得内存可以缓存更多 block,读性能自然会更好。 相反,建议将块大小设置较大。随着 BlockSize 的增大,系统随机读的吞吐量不断降低, 延迟不断增大。随着 BlockSize 增大,scan 的吞吐量逐渐增大,延迟不断降低。可见, 如果业务请求以 Get 请求为主,可以考虑将块大小设置较小;如果以 Scan 请求为主,





可以将块大小调大; 默认的 64M 块大小是在 Scan 和 Get 之间取得的一个平衡。

3.3.3.4 预分区

当 hbase 的 region 分区过多,数据分布不均衡,或者有热点。我们就可以对该表进行预分区

来避免以上问题。减少由于 region split 带来的资源消耗。从而提高 HBase 的性能。

3.3.3.5 如何预分区

===方法 1===

通过可视化 shell 来创建。命令样例如下:

create 't1', 'f1', SPLITS => ['10', '20', '30', '40']

create 't1', {NAME =>'f1', TTL => 180}, SPLITS => ['10', '20', '30', '40']

create 't1', {NAME =>'f1', TTL => 180}, {NAME => 'f2', TTL => 240}, SPLITS => ['10', '20', '30', '40']

۲	HBase	♥ 当前集群: 20190215 ∨ 🗳	5			
۲	▼ 集群信息	表可视化	基本信息 二级感	契引 Region		
®₁ ₽₽	集群列表	新增 清空 删除表 禁用	<	BLOCKCACHE	true ~	true
	集群概览	Q 请输入表名筛选		BLOCKSIZE	65536	65536
×	HBase 概览			MOB_THRESHOLD	1000000	1000000
⊗	实例信息	SYSTEM.SEQUENCE SYSTEM.STATS		COMPACTION_POLICY	RatioBasedCt 🗸	RatioBasedCompact
	预警管理	backup:BackupRule		IS_MOB	true 🗸	true
	备份管理	backupTest1 backupTest2		HSTORE_ENGINE	DefaultStoreE >>	DefaultStoreEngine
	还原管理	pewrite		MOB_COMPACT_PARTITION_POLICY	DAILY	DAILY
	表可视化	test		SPLITS	"['10', '20', '30', '4	删除
	配置管理	tsdb-meta		添加		
		tsdb-uid	共1条 〈 1 〉			
		4	4			E.

===方法 2===



仍然是通过可视化 shell 来创建,不过是通过读取文件

1、在任意路径下创建一个保存分区 key 的文件, 我这里如下

路径: /home/hadmin/hbase-2.0.3/txt/splits.txt

2、通过可视化 shell 创建表

命令样例:

create 't1', 'f1', SPLITS_FILE => '/home/hadmin/hbase-1.3.1/txt/splits.txt'

create 't1', {NAME =>'f1', TTL => 180}, SPLITS_FILE =>

'/home/hadmin/hbase-2.0.3/txt/splits.txt'

create 't1', {NAME =>'f1', TTL => 180}, {NAME => 'f2', TTL => 240}, SPLITS_FILE => '/home/hadmin/hbase-2.0.3/txt/splits.txt'

۲	HBase	♀ 当前集群: 20190215 ∨ 🗗 🖆	5 - 5			
<u>د</u>	▼ 集群信息	表可视化	基本信息 二级索·	31 Region		
••	集群列表	新增清空删除表禁用		MOB_THRESHOLD	1000000	1000000
•	集群概览	Q 请输入表名筛选		COMPACTION_POLICY	RatioBasedCr 🗸	RatioBasedCompact
8	HBase 概览			IS_MOB	true 🗸	true
8	实例信息 	SYSTEM.STATS		HSTORE_ENGINE	DefaultStoreE >	DefaultStoreEngine
	<u> </u>	backup:BackupRule		MOB_COMPACT_PARTITION_POLICY	DAILY V	DAILY
	备份管理	backupTest1		SPLITS_FILE	"/home/hadmin/t	删除
	还原管理	pewrite		添加		
	表可视化	test				
	配置管理	tsdb	共1条 〈 1 〉	$\langle 1 \rangle$		
		tsdb-tree				
		tsdb-uid	 Phoenix 列名 	* Phoenix 建表主键默认为 id		
		< >>	4			Þ



3.3.3.6 columnFamily 设计原则

列族 column family: 它是 column 的集合,在创建表的时候就指定,不能频繁修改。值得注意的是, 列族的数量越少越好,因为过多的列族意味着数据库存放需要更多单独的 storefile 文件来存放,文 件越多,数据库读写的性能就会越差。一般设置一个到两个列族。

3.3.3.7 rowkey 设计原则

HBase 的数据是按照 RowKey 的字典顺序进行全局排序的,所有的查询都只能依赖于这一个排序维度。 RowKey 设计的好坏直接决定了数据库读写的性能和数据保存的均衡性。数据库数据不均衡,虽然数据 库本身可以开启自动均衡功能,但其均衡过程会间接影响数据库的读写,从而导致了数据库使用的稳 定性体验。

Rowkey 是一个二进制码流,长度建议按照 8 字节倍数进行设计,一般控制在 16 字节。

Rowkey 高位建议进行散列,低位按照 rowkey 的规律进行处理。假如低位是按照时间规律生成,那么低位可以在 regionserver 内部达到顺序化的目的。

Rowkey 避免热点的常用方式有:

加盐

在 rowkey 的前面增加一个前缀,一般是随机数,使得它和之前的 rowkey 的开头不同。分配的前缀种 类数量应该和你想使用数据分散到不同的 region 的数量一致。加盐之后的 rowkey 就会根据随机生成 的前缀分散到各个 region 上,从而达到 Region 负载均衡的目标。

例如: 有 4 个 Region(注: 以 [, f1)、[f1, f2)、[f3, f4)、[f4,)为 Region 起至

加 Salt 前的 Rowkey: a001、a002、a003

f1、f2、f3、f4 为前缀, 加 Salt 后 Rowkey 为: f1-a001、f2-a002、f3-a003 、f4-a004

实验可以看到,加盐后的 Rowkey 数据会分布在 4 个 region 中,理论上处理后的吞吐量应是之前的 4 倍。由于前缀是随机的,读这些数据时需要耗费更多的时间,所以 Salt 增加了写操作的吞吐量,不 过缺点是同时增加了读操作的开销。



天翼 E Cloud

哈希

哈希会使同一行永远用一个前缀加盐。哈希也可以使负载分散到整个集群,但是读却是可以预测的。 使用确定的哈希可以让客户端重构完整的 rowkey,可以使用 get 操作准确获取某一个行数据

反转

第三种防止热点的方法时反转固定长度或者数字格式的 rowkey。这样可以使得 rowkey 中经常改变的部分(最没有意义的部分)放在前面。这样可以有效的随机 rowkey,但是牺牲了 rowkey 的有序性。 常用的类似工号,电话号码前段部分比较固定这类容易引起数据热点。

3.3.4 样例代码说明

3.3.4.1 创建 Configuration

功能介绍

HBase 通过 login 方法来获取配置项。包括用户登录信息、安全认证信息等配置项。

代码样例

下面代码片段在 cn. chinatelecom. hbase. sample. client 包的 "HbaseConf" 类的 init 方法中。

private static void init() {

conf = HBaseConfiguration.create();

conf.set("hbase.zookeeper.quorum", HBASE_ZK_QUORUM);

conf.set("hbase.zookeeper.property.clientPort", HBASE_ZK_CLIENT_PORT);

conf.set("zookeeper.znode.parent", ZK_ZNODE_PARENT);



}

3.3.4.2 创建 Connection

功能介绍

HBase 通过 ConnectionFactory.createConnection(configuration)方法创建 Connection 对象。

传递的参数为上一步创建的 Configuration。

Connection 封装了底层与各实际服务器的连接以及与 ZooKeeper 的连接。Connection 通过

ConnectionFactory 类实例化。创建 Connection 是重量级操作, Connection 是线程安全

的,因此,多个客户端线程可以共享一个 Connection。

典型的用法,一个客户端程序共享一个单独的 Connection,每一个线程获取自己的

HbaseClient 实例,然后调用 HbaseClient 对象提供的操作接口。不建议缓存或

者池化 HbaseClient。HbaseClient 的生命周期由调用者维护,调用者通过调用 release(),

释放资源。

代码样例

以下代码片段是登录, 创建 Connection 并创建表的示例, 在

下面代码片段在 cn. chinatelecom. hbase. sample. client 包的"HbaseClient"类的构造方法中。

private Connection conn;

private Admin admin;

public HbaseClient(Configuration conf) {

try {

conn = ConnectionFactory.createConnection(conf);

admin = conn.getAdmin();



```
} catch (IOException e) {
   LOGGER.error(e.getMessage());
   System.exit(-1);
}
```

说明

构造代码要避免重复调用。释放资源的代码片段在 cn. chinatelecom. hbase. sample. client 包的 "HbaseClient"类的 release 方法中。

```
public void release() {
    try {
        if (admin != null) {
            admin.close();
        }
        if (conn != null) {
            conn.close();
        }
    }
    catch (10Exception e) {
        LOGGER.error(e.getMessage());
    }
}
```

3.3.4.3 创建表

功能简介



HBase 通过 org. apache. hadoop. hbase. client. Admin 对象的 createTable 方法来创建表,并指 定表名、列族名。创建表有两种方式(强烈建议采用预分 Region 建表方式):

| 快速建表,即创建表后整张表只有一个 Region,随着数据量的增加会自动分裂成

多个Region。

| 预分 Region 建表,即创建表时预先分配多个 Region,此种方法建表可以提高写入

大量数据初期的数据写入速度。

说明

表的列名以及列族名不能包含特殊字符,可以由字母、数字以及下划线组成。

代码样例

以下代码片段在 cn. chinatelecom. hbase. sample. client 包的"HbaseClient"类的 createTable 方法中。

public void createTable(TableName tableName) throws IOException{

ColumnFamilyDescriptor columnFamilyDescriptor =

ColumnFamilyDescriptorBuilder.newBuilder(Bytes.toBytes("info"))//(1)

.setCompressionType(Compression.Algorithm.LZ4)//注[1]

.setDataBlockEncoding(DataBlockEncoding.DIFF)

.build();

TableDescriptor tableDescriptor = TableDescriptorBuilder.newBuilder(tableName)//(2)

.setColumnFamily(columnFamilyDescriptor)//(3)

.setCompactionEnabled(true)

.setMaxFileSize(10 * 1024 * 1024 * 1024)





```
.build();//注[2]
```

admin.createTable(tableDescriptor);//(4)

}

解释

- (1) 创建列族描述符
- (2) 创建表描述符
- (3) 添加列族描述符到表描述符中
- (4)调用 Admin 的建表方法。

注意事项

- | 注[1] 可以设置列族的压缩方式,代码片段如下:
- //设置编码算法, HBase 提供了 DIFF, FAST_DIFF 和 PREFIX 三种编码算法
- ${\tt setDataBlockEncoding}\ ({\tt DataBlockEncoding}.\ {\tt DIFF})$
- //设置文件压缩方式, HBase 默认提供了 LZ4、BZ IP2、GZ、LZ0、SNAPPY 和 ZSTD 压缩算法
- //其中 ZSTD 的压缩率高,但压缩和解压性能相对 LZ4 低,适用于冷数据
- //LZ4 压缩率低,但压缩解压性能高,适用于热数据
- //建议默认开启 LZ4 压缩
- setCompressionType(Compression.Algorithm.LZ4);
- | 注[2] 可以通过指定起始和结束 RowKey,或者通过 RowKey 数组预分 Region 两种方
- 式建表,代码片段如下:
- // 创建一个预划分 region 的表



byte[][] splits = new byte[4][];

splits[0] = Bytes.toBytes("A");

splits[1] = Bytes.toBytes("H");

splits[2] = Bytes.toBytes("0");

splits[3] = Bytes.toBytes("U");

admin.createTable(htd, splits);

3.3.4.4 修改表

功能简介

HBase 通过 org. apache. hadoop. hbase. client. Admin 的 modifyTable 方法修改表信息。 代码样例

以下代码片段在 cn. chinatelecom. hbase. sample. client 包的"HBaseClient"类的 modifyTable 方法中。

public void modifyTable(TableName tableName) throws IOException {

HTableDescriptor hTableDescriptor = admin.getTableDescriptor(tableName);

HTableDescriptor copy = new HTableDescriptor(hTableDescriptor);

HColumnDescriptor hColumnDescriptor = new HColumnDescriptor (Bytes.toBytes("new-cf"));

copy.addFamily(hColumnDescriptor);

admin.disableTable(tableName);//注[1]

admin.modifyTable(tableName, copy);



admin.enableTable(tableName);

}

注意事项

注[1] modifyTable 只有表被 disable 时,才能生效。

3.3.4.5 删除表

功能简介

HBase 通过 org. apache. hadoop. hbase. client. Admin 的 deleteTable 方法来删除表。

代码样例

以下代码片段在 cn. chinatelecom. hbase. sample. client 包的 "HBaseClient" 类的 dropTable

方法中。

public void dropTable(TableName tableName) throws IOException {

if (admin.tableExists(tableName)) {

admin.disableTable(tableName);//注[1]

admin.deleteTable(tableName);

}

}

注意事项

注[1] 只有表被 disable 时,才能被删除掉,所以 deleteTable 常与 disableTable, enableTable, tableExists, isTableEnabled, isTableDisabled 结合在一起使用。



3.3.4.6 插入数据

功能简介

HBase 是一个面向列的数据库,一行数据,可能对应多个列族,而一个列族又可以对应

多个列。通常,写入数据的时候,我们需要指定要写入的列(含列族名称和列名

称)。HBase 通过 HTable 的 put 方法来 Put 数据,可以是一行数据也可以是数据集。

代码样例

以下代码片段在 cn. chinatelecom. hbase. sample. client 包的 "HBaseClient" 类的 put 方 法中。

/*

插入数据:

```
主键 账号 邮箱 年龄 工作
```

1000101 molika tt@hotmail.cn 29 IT

1000102 kouda kouda@126.com 24 销售

1000103 liusha liusha@173.com 36 经理

1000104 baifei baifei@qq.com 16 学生

*/

public void put(TableName tableName) throws IOException{

```
Table table = null;
```

byte[] family = Bytes.toBytes("info");

byte[][] qualifiers =

{Bytes.toBytes("account"), Bytes.toBytes("email"), Bytes.toBytes("age"), Bytes.toBytes("ca





```
reer")};
```

try {

table = conn.getTable(tableName);

//单个 put

Put put = new Put(Bytes.toBytes("1000101"));

put.addColumn(family,qualifiers[0],Bytes.toBytes("molika"));

put. addColumn(family, qualifiers[1], Bytes. toBytes("tt@hotmail.cn"));

put. addColumn(family, qualifiers[2], Bytes. toBytes("29"));

put. addColumn(family, qualifiers[3], Bytes. toBytes("IT"));

table.put(put);

List<Put> puts = new ArrayList<>();

put = new Put(Bytes.toBytes("1000102"));

put.addColumn(family,qualifiers[0],Bytes.toBytes("kouda"));

put. addColumn(family, qualifiers[1], Bytes. toBytes("kouda@126. com"));

put. addColumn(family, qualifiers[2], Bytes. toBytes("24"));

put.addColumn(family,qualifiers[3],Bytes.toBytes("销售"));

puts.add(put);

put = new Put(Bytes.toBytes("1000103"));



put.addColumn(family,qualifiers[0],Bytes.toBytes("liusha")); put.addColumn(family,qualifiers[1],Bytes.toBytes("liusha@173.com")); put.addColumn(family,qualifiers[2],Bytes.toBytes("36")); put.addColumn(family,qualifiers[3],Bytes.toBytes("经理")); puts.add(put);

put = new Put(Bytes.toBytes("1000104"));

put. addColumn(family, qualifiers[0], Bytes. toBytes("baifei"));

put.addColumn(family,qualifiers[1],Bytes.toBytes("baifei@qq.com"));

put. addColumn(family, qualifiers[2], Bytes. toBytes("16"));

put.addColumn(family,qualifiers[3],Bytes.toBytes("学生"));

puts. add(put);

table.put(puts);

```
} catch (IOException e) {
```

LOGGER.error("put data failed, ", e);

```
} finally {
```

table.close();

}

}

注意事项




不允许多个线程在同一时间共用同一个 HTable 实例。HTable 是一个非线程安全类,因此,同一个 HTable 实例,不应该被多个线程同时使用,否则可能会带来并发问题。

3.3.4.7 删除数据

功能简介

HBase 通过 Table 实例的 delete 方法来 Delete 数据,可以是一行数据也可以是数据集。

代码样例

以下代码片段在 cn. chinatelecom. hbase. sample. client 包的 "HBaseClient" 类的 delete

方法中。

public void delete(TableName tableName) throws IOException{

```
Table table = null;
```

try {

```
table = conn.getTable(tableName);
```

//单个删除

Delete delete = new Delete(Bytes.toBytes("1000101"));

table. delete(delete);

//批量 delete

List<Delete> deletes = new ArrayList<>();

delete = new Delete(Bytes.toBytes("1000102"));

deletes.add(delete);



```
delete = new Delete(Bytes.toBytes("1000103"));
  deletes.add(delete);
  delete = new Delete(Bytes.toBytes("1000104"));
  deletes.add(delete);
  table.delete(deletes);
} catch (10Exception e) {
  LOGGER.error("delete failed.",e);
} finally {
  table.close();
}
```

如果被删除的 cell 所在的列族上设置了二级索引,也会同步删除索引数据。

3.3.4.8 使用 Get 读取数据

功能简介

}

说明

要从表中读取一条数据,首先需要实例化该表对应的 Table 实例,然后创建一个 Get 对

象。也可以为 Get 对象设定参数值,如列族的名称和列的名称。查询到的行数据存储在

Result 对象中, Result 中可以存储多个 Cell。

代码样例

以下代码片段在 cn. chinatelecom. hbase. sample 包的 "HBaseSample" 类的 sampleGet 方



法中。

```
public static List<String> sampleGet(Connection conn) {
        List<String> list = new ArrayList<>();
        TableName tableName = TableName.valueOf("hbase-sample");
        byte[] familyName = Bytes.toBytes("user");
        byte[][] qualifier = {Bytes.toBytes("name"), Bytes.toBytes("age")};
        byte[] rowKey = Bytes.toBytes("001");
        Get get = new Get(rowKey);
        get.addColumn(familyName, qualifier[0]);
        get.addColumn(familyName, qualifier[1]);
        try (Table table = conn.getTable(tableName)) {
            Result result = table.get(get);
            showData(list, result);
            log. info("Get data successfully. ");
        } catch (IOException e) {
            log.error("Get data failed ", e);
        }
        return list;
    }
```

private static void showData(List<String> list, Result result) {



}

3.3.4.9 使用 Scan 读取数据

功能简介

要从表中读取数据,首先需要实例化该表对应的 Table 实例,然后创建一个 Scan 对象, 并针对查询条件设置 Scan 对象的参数值,为了提高查询效率,最好指定 StartRow 和 StopRow。查询结果的多行数据保存在 ResultScanner 对象中,每行数据以 Result 对象形 式存储, Result 中存储了多个 Cell。

代码样例

以下代码片段在 cn. chinatelecom. hbase. sample 包的 "HBaseSample" 类的

sampleScan方法中。

public static List<String> sampleScan(Connection conn) {

List<String> list = new ArrayList<>();

TableName tableName = TableName.valueOf("hbase-sample");



```
Scan scan = new Scan();
        scan. addColumn(Bytes. toBytes("user"), Bytes. toBytes("name"));
        scan. setCaching(1000);
        scan.withStartRow(Bytes.toBytes("001"));
        // default not include stopRow
        scan.withStopRow(Bytes.toBytes("003"));
        try (Table table = conn.getTable(tableName); ResultScanner rScanner =
table.getScanner(scan)) {
            for (Result result : rScanner) {
                showData(list, result);
            }
            log. info("Scan data successfully");
        } catch (IOException e) {
            log.error("Scan data failed ", e);
        }
        return list;
   }
    private static void showData(List<String> list, Result result) {
            for (Cell cell : result.rawCells()) {
```

```
String temp = Bytes.toString(CellUtil.cloneRow(cell)) + ":"
```



```
+ Bytes.toString(CellUtil.cloneFamily(cell)) + ","
                      + Bytes.toString(CellUtil.cloneQualifier(cell)) + ","
                      + Bytes.toString(CellUtil.cloneValue(cell));
               log. info(temp);
               list.add(temp);
           }
1. 建议 Scan 时指定 StartRow 和 StopRow, 一个有确切范围的 Scan, 性能会更好些。
```

- 2. 可以设置 Batch 和 Caching 关键参数。
- Batch

注意事项

}

使用 Scan 调用 next 接口每次最大返回的记录数,与一次读取的列数有关。

- Caching

RPC 请求返回 next 记录的最大数量,该参数与一次 RPC 获取的行数有关。

3.3.4.10 使用过滤器 Filter

功能简介

HBase Filter 主要在 Scan 和 Get 过程中进行数据过滤,通过设置一些过滤条件来实现,如

设置 RowKey、列名或者列值的过滤条件。

代码样例

以下代码片段在 cn. chinatelecom. hbase. sample 包的 "HBaseSample" 类的



```
sampleScanFilter 方法中。
```

```
public static List<String> sampleScanFilter(Connection conn) {
```

```
List<String> list = new ArrayList<>();
```

TableName tableName = TableName.valueOf("hbase-sample");

```
Scan scan = new Scan();
```

scan.addColumn(Bytes.toBytes("user"), Bytes.toBytes("name"));

SingleColumnValueFilter filter = new SingleColumnValueFilter(

Bytes.toBytes("user"), Bytes.toBytes("name"), CompareOperator.EQUAL,

Bytes.toBytes("Zhang San"));

```
scan. setFilter(filter);
```

```
try (Table table = conn.getTable(tableName); ResultScanner rScanner =
table.getScanner(scan)) {
```

for (Result result : rScanner) {
 showData(list, result);
 }
 log.info("Scan filter data successfully");
} catch (IOException e) {
 log.error("Scan filter data failed ", e);
}
return list;





```
private static void showData(List<String> list, Result result) {
    for (Cell cell : result.rawCells()) {
        String temp = Bytes.toString(CellUtil.cloneRow(cell)) + ":"
            + Bytes.toString(CellUtil.cloneFamily(cell)) + ", "
            + Bytes.toString(CellUtil.cloneQualifier(cell)) + ", "
            + Bytes.toString(CellUtil.cloneValue(cell));
        log.info(temp);
        list.add(temp);
    }
}
注意事项
当前二级索引不支持使用 SubstringComparator 类定义的对象作为 Filter 的比较器。
```

Scan scan = new Scan();

filterList = new FilterList(FilterList.Operator.MUST_PASS_ALL);

filterList.addFilter(new SingleColumnValueFilter(Bytes

```
.toBytes(columnFamily), Bytes.toBytes(qualifier),
```

CompareOperator.EQUAL, new SubstringComparator(substring)));

scan. setFilter(filterList);



3.3.4.11 创建 Phoenix 表

功能简介

Phoenix 依赖 HBase 作为其后备存储, 支持标准 SQL 和 JDBC API 的强大功能, 使得 SQL

用户可以访问 HBase 集群。

代码样例

以下代码片段在 cn. chinatelecom. hbase. sample 包的 "PhoenixSample" 类的

sampleCreateTable、sampleDropTable方法中。

/**

* Create Table

*/

public static boolean sampleCreateTable(String url) {

//url is queryserver url, sample:

jdbc:phoenix:thin:url=http://localhost:8765;serialization=PROTOBUF

```
int result = -1;
```

String sql = "CREATE TABLE IF NOT EXISTS PHOENIX_SAMPLE (id integer not null primary
key, name varchar(10), age integer)";

try (Connection conn = DriverManager.getConnection(url); Statement stat =
conn.createStatement()) {

result = stat.executeUpdate(sql);

log. info("Create Table successfully. ");

} catch (Exception e) {



```
log.error("Create Table failed ", e);
        }
        return result \geq 0;
    }
/**
* Drop Table
 */
public static boolean sampleDropTable(String url) {
        //url is queryserver url, sample:
jdbc:phoenix:thin:url=http://localhost:8765;serialization=PROTOBUF
        int result = -1;
        String sql = "DROP TABLE IF EXISTS PHOENIX_SAMPLE";
        try (Connection conn = DriverManager.getConnection(url);
             Statement stat = conn.createStatement()) {
            result = stat.executeUpdate(sql);
            log. info("Drop Table successfully.");
        } catch (Exception e) {
            log.error("Drop Table failed ", e);
        }
        return result \geq 0;
   }
```



3.3.4.12 写 Phoenix 表

功能简介

使用 Phoen ix 实现写数据。

示例代码

以下代码片段在 cn. chinatelecom. hbase. sample. PhoenixSample. sampleUpsertData 方法中。

/**

* Upsert data

*/

public static boolean sampleUpsertData(Configuration conf) {

log.info("Entering sampleUpsertData.");

int result = -1;

//default assume zookeeper is stand-alone, and phoenix query server is same as

zookeeper

String url = "jdbc:phoenix:thin:url=http://" + conf.get("hbase.zookeeper.quorum") + ":" +

conf.get("phoenix.queryserver.http.port", "8765") +

";serialization=" + conf.get("phoenix.queryserver.serialization",

"PROTOBUF");

```
String sql = "UPSERT INTO PHOENIX_SAMPLE VALUES(1, 'Tom', 25)";
```

```
try (Connection conn = DriverManager.getConnection(url); Statement stat =
conn.createStatement()) {
```



```
result = stat.executeUpdate(sql);
log.info("Upsert data successfully.");
} catch (Exception e) {
log.error("Upsert data failed.", e);
}
log.info("Exiting sampleUpsertData.");
return result >= 0;
}
```

3.3.4.13 读 Phoenix 表

功能简介

使用 Phoenix 实现读数据。

示例代码

以下代码片段在 cn. chinatelecom. hbase. sample. PhoenixSample. sampleSelectData 方法中。

/**

* Select Data

*/

public static boolean sampleSelectData(Configuration conf) {

log.info("Entering sampleSelectData.");

int result = 0;

 $//default\ assume\ zookeeper\ is\ stand-alone,\ and\ phoenix\ query\ server\ is\ same\ as$

zookeeper

```
String url = "jdbc:phoenix:thin:url=http://" +
```

```
conf.get("hbase.zookeeper.quorum") + ":" +
```

conf.get("phoenix.queryserver.http.port", "8765") +

";serialization=" + conf.get("phoenix.queryserver.serialization",

"PROTOBUF");

```
String sql = "SELECT * FROM PHOENIX_SAMPLE WHERE id = ?";
```

```
try (Connection conn = DriverManager.getConnection(url); PreparedStatement
preStat = conn.prepareStatement(sql)
```

) {

```
preStat.setInt(1, 1);
try (ResultSet resultSet = preStat.executeQuery()) {
  while (resultSet.next()) {
    result++;
    int id = resultSet.getInt("id");
    String name = resultSet.getString(1);
    int age = resultSet.getInt("age");
    System.out.println("id: " + id);
    System.out.println("id: " + name);
    System.out.println("age: " + name);
    System.out.println("age: " + age);
  }
  log.info("Select data successfully.");
```



```
}
} catch (Exception e) {
    log.error("Select data failed.", e);
}
log.info("Exiting sampleSelectData.");
return result >= 0;
```

3.3.4.14 创建二级索引

}

功能简介

```
使用 Phoen ix 创建二级索引。
```

示例代码

以下代码片段在 cn. chinatelecom. hbase. sample. PhoenixSample. sampleCreateIndex 方法中。

/**

```
* Create Index
```

*/

public static boolean sampleCreateIndex(Configuration conf) {

log.info("Entering sampleCreateIndex.");

```
int result = -1;
```

//default assume zookeeper is stand-alone, and phoenix query server is same as zookeeper

String url = "jdbc:phoenix:thin:url=http://" + conf.get("hbase.zookeeper.quorum")



```
+ ":" +
```

conf.get("phoenix.queryserver.http.port", "8765") +

```
";serialization=" + conf.get("phoenix.queryserver.serialization",
```

"PROTOBUF");

String sql = "CREATE INDEX IF NOT EXISTS IDX_PHOENIX_SAMPLE_AGE ON PHOENIX_SAMPLE
(age)";

```
try (Connection conn = DriverManager.getConnection(url); Statement stat =
conn.createStatement()) {
```

result = stat.executeUpdate(sql); log.info("Create index successfully."); } catch (Exception e) { log.error("Create index failed.", e); } log.info("Exiting sampleCreateIndex."); return result >= 0;

注意事项

}

需要先创建 Phoenix 表再创建二级索引。

二级索引包含全局索引和本地索引两大类,又有函数索引、覆盖索引和联合索引等多种实现。本样例 仅展示创建二级索引的基本代码结构,不同索引 SQL 语法上的差异请查阅 Phoenix 官方文档。

3.3.4.15 删除索引

功能简介



```
使用 Phoen ix 删除二级索引。
```

示例代码

以下代码片段在 cn. chinatelecom. hbase. sample. PhoenixSample. sampleDropIndex 方法中。

/**

* Drop Index

*/

public static boolean sampleDropIndex(Configuration conf) {

log.info("Entering sampleDropIndex.");

int result = -1;

//default assume zookeeper is stand-alone, and phoenix query server is same as zookeeper

String url = "jdbc:phoenix:thin:url=http://" + conf.get("hbase.zookeeper.quorum")
+ "." +

conf.get("phoenix.queryserver.http.port", "8765") +

";serialization=" + conf.get("phoenix.queryserver.serialization",

"PROTOBUF");

```
String sql = "DROP INDEX IF EXISTS IDX_PHOENIX_SAMPLE_AGE ON PHOENIX_SAMPLE";
```

try (Connection conn = DriverManager.getConnection(url); Statement stat =
conn.createStatement()) {

result = stat.executeUpdate(sql);

log.info("Drop index successfully.");

} catch (Exception e) {



```
log.error("Drop index failed.", e);
}
log.info("Exiting sampleDropIndex.");
return result ≻= 0;
```

3.3.4.16 二级索引查询

功能简介

}

针对添加了二级索引的用户表,其数据查询性能高于针对无二级索引用户表的数据查询。

样例:使用二级索引查找数据

以下代码片段在 cn. chinatelecom. hbase. sample 包的 "PhoenixSample" 类的

sampleScanDataByIndex方法中。

public static boolean sampleScanDataByIndex(Configuration conf) {

log.info("Entering sampleScanDataByIndex.");

String url = "jdbc:phoenix:thin:url=http://" + conf.get("hbase.zookeeper.quorum") + ":"

+

conf.get("phoenix.queryserver.http.port", "8765") +

";serialization=" + conf.get("phoenix.queryserver.serialization",

"PROTOBUF");

String sql = "SELECT age FROM PHOENIX_SAMPLE";

try (Connection conn = DriverManager.getConnection(url); Statement stat =



```
conn.createStatement()) {
    stat.executeQuery(sql);
    log.info("Scan index successfully.");
    log.error("Scan index failed.", e);
    return false;
    log.info("Exiting sampleScanDataByIndex.");
    return true;
}
注意事项
```

```
需要预先对字段 age 创建二级索引。
```

3.3.4.17 写 MOB 表

功能介绍

HBase MOB 数据读出与普通 HBase 数据的读出没有什么区别,对客户来说是透明的。

为了使用 HBase MOB 功能需要在 web 控制台配置管理添加 HBase MOB 相关的配置项,

除此之外还需要在表可视化选定对应的表指定 columnfamily 上开启 MOB 功能。

以下代码片段在 cn. chinatelecom. hbase. sample 包的 "MOBSample" 类的

createMOBTable 方法中。

public void createMOBTable() {





```
log.info("Entering createMOBTable.");
```

```
try (Admin admin = conn.getAdmin()) {
```

ColumnFamilyDescriptor columnFamily = ColumnFamilyDescriptorBuilder

.newBuilder(CF.getBytes())

.setMobEnabled(true)

```
.setMobThreshold(10L).build();
```

TableDescriptor tableDescriptor = TableDescriptorBuilder

.newBuilder(tableName)

.setColumnFamily(columnFamily)

.build();

```
admin.createTable(tableDescriptor);
```

log.info("MOB Table is created successfully.");

} catch (Exception e) {

log.error("MOB Table is created failed ", e);

}

log.info("Exiting createMOBTable.");

}

样例:用 Put 接口写入 MOB 数据



```
天翼云
e Cloud
```

```
public void insertMOBData(byte[] value) {
    log.info("Entering insertMOBData.");
    try {
        Put p = new Put(Bytes.toBytes("row"));
        p.addColumn(Bytes.toBytes(CF), Bytes.toBytes("c1"), value);
        Table table = conn.getTable(tableName);
        table.put(p);
        log.info("MOB data inserted successfully.");
    } catch (Exception e) {
        log.error("MOB data inserted failed ", e);
    }
    log.info("Exiting insertMOBData.");
}
```

3.3.4.18 读 MOB 表

功能简介

为了使用 HBase MOB 功能需要在 web 控制台配置管理添加 HBase MOB 相关的配置项,

除此之外还需要在表可视化选定对应的表指定 columnfamily 上开启 MOB 功能。

以下代码片段在 cn. chinatelecom. hbase. sample 包的 "MOBSample" 类的

readMOBData 方法中。

public byte[] readMOBData() {

log.info("Entering readMOBData.");



```
byte[] value = "NULL".getBytes();
try (Table table = conn.getTable(tableName);
     Admin admin = conn.getAdmin();
     ResultScanner scanner = table.getScanner(new Scan())) {
    admin.flush(table.getName());
    for (Result result : scanner) {
         value = result.getValue(Bytes.toBytes(CF),
                Bytes. toBytes("c1"));
        String string = Bytes.toString(value);
        log. info("value:" + string);
    }
    log.info("MOB data read successfully.");
} catch (Exception e) {
    log.error("MOB data read failed ", e);
}
log.info("Exiting readMOBData.");
return value;
```

3.3.4.19 Region 预分区

功能简介

}

本例使用多点分割将 HBase 表按照 "-∞[~]A" 、 "A[~]C" 、 "C[~]G" 、



```
"G<sup>~</sup>I" 、 "I<sup>~</sup>+∞" 分为五个 Region。
```

以下代码片段在 cn. chinatelecom. hbase. sample 包的 "MOBSample" 类的

```
createTableBySplitKeys 方法中。
```

public void createTableBySplitKeys() {

log. info("Entering createTableBySplitKeys.");

```
try (Admin admin = conn.getAdmin()) {
```

ColumnFamilyDescriptor columnFamily = ColumnFamilyDescriptorBuilder

.newBuilder(CF.getBytes())

. build();

TableDescriptor tableDescriptor = TableDescriptorBuilder

```
.newBuilder(tableName)
```

.setColumnFamily(columnFamily)

. build();

```
byte[][] splitKeys = new byte[4][];
splitKeys[0] = "A".getBytes();
splitKeys[1] = "C".getBytes();
splitKeys[2] = "G".getBytes();
splitKeys[3] = "I".getBytes();
```



```
admin.createTable(tableDescriptor, splitKeys);
log.info("createTableBySplitKeys successfully.");
} catch (Exception e) {
log.error("MultiSplit failed.",e);
}
log.info("Exiting createTableBySplitKeys.");
```

}

注意:预分区操作只能在建表时起作用。



4 常见问题

Q:云 HBASE 适用什么场景?

A:云 HBASE 是基于开源 HBase 引擎支持 NOSQL 和二级索引,提供高性能、低延迟、易运营实时 分析云服务,适合上百 TB,百万级 QPS 的分布式计算应用场景。单节点定位开发和测试场景, 集群版定位于企业生产。

Q:创建云 HBASE 集群需要准备什么?

A:创建云 HBASE 集群时,请根据业务负载确定计算单元数量。在集群创建后,不支持计算单元减容。 如果集群处理业务一段时间后,云 HBASE 会自动对负载指标收集,如果超过警告值会做出预警,用 户可以考虑扩容增加计算单元数量。

Q:线下或者 ECS 上自建 CDH/HDP 怎么搬迁到云 HBASE?

A:HBase 提供搬迁不停机搬迁服务,可以在不影响业务的情况下实现数据搬迁。两种情况:1)线下 应用搬迁到天翼云对应资源池的 ecs 主机。2)已有云主机,开通云 HBASE 后下载更新 hbase-site.xml, zk 配置文件,重启应用程序生效。

Q: 什么是数据分布不均衡? 出现后如何处理?

A: Hbase 的表会被划分为 1....n 个 Region,被托管在 RegionServer 中。Region 二个重要的属性: Startkey 与 EndKey 表示这个 Region 维护的 rowkey 的范围,当我们要读写数据时,如果 rowkey 落在某个 start-end key 范围内,那么就会定位到目标 region 并且读写到相关的数据。

当数据库单表或者多表 region 读写负载长期发生极大偏差的时候,部分 region 负载很高的,我们称 热点。查看 memstore 的大小,如果 memstore 大小存在很大的偏差。那么我们称这种现象为 hbase 数据分布不均衡。



面对这种情况,我们可以手动地将热点 region 按特定的边界拆分出一个或多个新 region,然后将子 region 负载分布到多个 region 服务器上。

对于连续的行键的分布,可以采用盐析主键或者随机行键进行处理。

对于非连续的行键的分布,可以为 region 指定一个任意值做为拆分行键。

如果一个表的大部分 region 落在一个服务节点上,那么可以采用客户端的 move()指令把这些 region 迁移到别的服务节点上。也可以使用 unassign()指令让数据库从当前服务节点卸载,再进行重新分 配到别的服务节点上。

Q:我配置了 ZK 地址以后,还是无法连接上 HBASE

A: 可能的原因如下

1 忘记在云 HBASE 管理平台添加白名单。请登录进入云 HBASE 管理平台后

访问控制模块,添加访问机器的 ip.

2 两个不同的 VPC 环境默认是不通的(跨 VPC 请 配置 高速通道),可能是跨 Region(地域跨城), 当前不支持跨域机器访问云 HBASE。

Q: 客户端提示 "Could not resolve the DNS name of xxx"

现象描述

HBase 客户端程序运行中,出现类似以下的日志打印: "Could not resolve the DNS

name of xxx", 提示 DNS 无法解析。

A: 可能原因 HBase 客户端需要配置集群各节点的 HostName 和 IP 的对应关系才可以正常连接到 HBase

集群。客户端进程所在机器没有正确配置 HBase 集群中 HostName 和 IP 对应关系,可能

导致客户端无法解析 DNS。

定位思路

1. 检查客户端程序运行环境上是否正确配置了 HBase 集群 HostName 与 IP 的对应关

系。

2. 定位修改后,重启客户端程序即可恢复正常。

3. 恢复该故障正常耗时5分钟。

处理步骤





步骤1 登录天翼云 hbase 运营控制台。

步骤 2 点击 "集群列表>选择一行记录,点击明细 >定位配置信息,下载客户端安装包和配置信息", 下载客户端程序到本地机器。

步骤3 解压缩客户端程序,并双击该目录下的"install.bat"脚本,自动配置客户端工程。

解压目录下的"HBase/hbase-examples"文件夹即为样例工程文件夹。

步骤 4 拷贝解压目录下的"hosts"文件中除"casserver"项以外的内容到本地机器的 hosts 文件中。

本地 hosts 文件存放路径:

I Windows 环境: "C:\Windows\System32\drivers\etc\hosts"

I Linux 环境: "/etc/hosts"

步骤5 执行 quit 命令,退出并重启客户端程序。

----结束

参考信息

无。

Q: 由于网络或操作系统故障引起的找不到 hbase:meta 异常

现象描述

在执行 MapReduce 或者 Spark 等程序时,可能出现如下异常导致的任务执行失败:

Caused by: java.net.SocketTimeoutException: callTimeout=60000, callDuration=60304: row ''

on table 'hbase:meta' at region=hbase:meta,,1.1588230740,

hostname=host1, 21302, 1448886113294,

seqNum=0

at

org. apache. hadoop. hbase. client. RpcRetryingCaller. callWithRetries (RpcRetryingCaller. java :159)

at org.apache.hadoop.hbase.client.ResultBoundedCompletionService



\$QueueingFuture.run(ResultBoundedCompletionService.java:64)

... 3 more

oud

A: 可能原因

| HDFS 服务不可用。

| ZooKeeper 上存储的 meta region 位置数据和实际不符。

定位思路

无。

处理步骤

步骤 1 确认 HDFS 服务是否可用,如果 HDFS 服务不可用,请先排除 HDFS 故障。

步骤 2 如果 HDFS 服务无故障,从 HBase 原生网页中找到 hbase:meta 表所在节点,重启该节点的 RegionServer。

步骤3 如果故障仍然存在,请联系天翼云客服。





免责声明

免责声明

- 本文档可能包含第三方信息、产品、服务、软件、组件、数据或内容(统称"第三 方内容")。天翼云不控制且不对第三方内容承担任何责任,包括但不限于准确性、兼容性、可靠性、 可用性、合法性、适当性、性能、不侵权、更新状态等,除非本文档另有明确说明。在本文档中提及 或引用任何第三方内容不代表华为对第三方内容的认可或保证。
- 用户若需要第三方许可,须通过合法途径获取第三方许可,除非本文档另有明确说明。