

刘洋,高鹏,徐娟,等. 基于 Mycat 的分布式技术在收集与分发系统中的应用[J]. 气象研究与应用,2021,42(1):90-95.

Liu Yang,Gao Peng,Xu Juan,et al. Application of distributed technology based on Mycat in collection and distribution system[J]. Journal of Meteorological Research and Application,2021,42(1):90-95.

# 基于 Mycat 的分布式技术在收集与分发系统中的应用

刘 洋,高 鹏,徐 娟,陈旭辉

(甘肃省气象信息与技术装备保障中心,甘肃 兰州 730020)

**摘要:** 基于 Mycat 的分布式数据库集群,采用了 Mycat+Mysql+Haproxy+Keepalived 的技术架构,实现了一个高可用、高性能和高扩展的分布式数据库集群系统,并对单机版 Mysql 数据库和基于 Mycat 分布式数据库集群进行性能比较。结果表明,基于 Mycat 的分布式技术研发数据库集群系统具有高效性和可行性,可以解决单机版的 Mysql 数据库运行因硬件故障、软件错误、存储的损毁等导致主机系统异常,数据服务无法响应,数据的可用性和时效性降低的问题。

**关键词:** Mycat;分布式;集群;数据库;性能

**中图分类号:** TP391

**文献标识码:** A

**doi:** 10.19849/j.cnki.CN45-1356/P.2021.1.16

**OSID:**



## 引言

随着科技的进步,气象数据的种类和数量在不斷上升,由原来单一的、小规格的数据累积成了海量数据<sup>[1]</sup>。数据量的急剧增长会导致单机的存储能力严重不足,特别是在汛期,高频率的查询使单机的处理能力和响应性能严重的下降,数据不能实时入库,给用户的使用带来极大的不便,同时给数据库的管理者带来了很大的负担,而分布式数据库<sup>[2-3]</sup>是在集中式数据库基础上发展而来的,其最显著的特点就是数据分布在地理位置不同的机器上,但是所有的数据在逻辑上是集中管理的,分布式数据库的出现克服了传统集中式数据库的许多缺点,成为大数据处理和存储的方案之一<sup>[4-6]</sup>,分布式数据库已被广泛的应用到各行各业<sup>[7-9]</sup>。目前,气象业务中的收集与分发系统所采用的 Mysql 数据库是单机版的,高并发的数据量会导致单机的存储能力和响应性能严重下降,极大的削弱了系统的服务能力。

针对现有数据库存在的问题,本文提出了基于 Mycat 的分布式数据库集群架构,利用分布式存储

中间件 Mycat 对数据进行了读写分离,从而极大的提高了数据库的读写性能以及可扩展性;利用 Haproxy<sup>[10-11]</sup> 实现了 Mycat 的高可用性;同时使用 keepalived<sup>[12]</sup> 实现了 Haproxy 的高可用,解决了传统的集中式数据库单点故障的问题,保障了气象数据传输的稳定性和有效性。

## 1 基于 Mycat 分布式数据库的设计

采用 Mycat+Mysql+Haproxy+Keepalived 的技术架构来构建关系型分布式集群系统,其整体架构如图 1 所示。为了保证系统的高性能、高并发和高可用,所有的模块都是基于集群进行构建,遵循自底向上的技术路线。

### 1.1 Mysql 数据库集群的实现

主从复制是实现 Mysql 高可用性、负载均衡以及读写分离的基础,Mysql 的主从复制有多种形式,本文采用一主一从的形式,即一个写节点(Master)和一个读节点(Slave),写节点负责所有应用的写操作,读节点负责所有应用的读操作,一主一从的架构部署简单并且高效,不仅可以实现高可用,还能够实

收稿日期: 2021-01-04

基金项目: 2020 年甘肃省气象局气象科学技术研究项目(MsCg2020-05)

作者简介: 刘洋(1987—),男,硕士,工程师,主要从事智能优化算法等工作。E-mail: 1036646681@qq.com

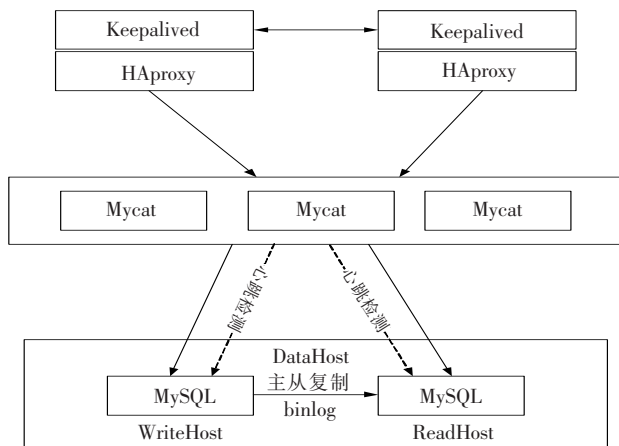


图 1 基于 Mycat 的分布式集群架构

现读写分离,进而提升集群的并发能力。

### 1.2 Mycat 实现 Mysql 的读写分离和高可用性

Mycat 是基于阿里巴巴 Cobar 项目的开源分布式数据库系统,其架构如图 2 所示,Mycat 是介于数据库和应用之间,负责资源的调度和服务,后端是物理数据库,Mycat 支持 Mysql,SQL Server,Oracle,DB2 等主流数据库。其原理是对用户发送的 SQL 语句进行拦截,首先对 SQL 语句做一些特定的分析,如路由分析、读写分离分析等,根据分析的结果,将对应的 SQL 语句发送至后端的物理数据库,并将返回结果进行处理后再输出给用户,对用户而言 Mycat 完全是透明的,为了防止 Mycat 服务器单点故障,Mycat 的部署采用灾备方式,当其中的一台服务器宕机后,另外的一台会立即接管相应的服务,避免因单机故障导致服务中断,从而实现了 Mycat 的高可用性。同时,Mycat 内置了一些功能,可以对后端真实的物理节点进行定期的检测,如果发现有故障节点,Mycat 可以进行节点之间的自动切换,从而

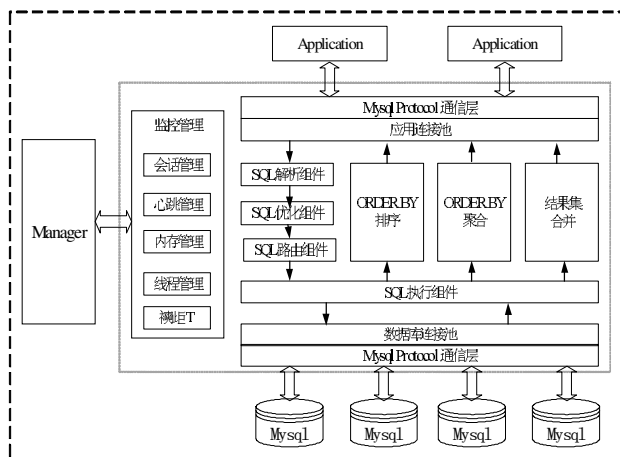


图 2 Mycat 的原理架构图

提高了数据库的可靠性。

### 1.3 Haproxy 实现 Mycat 负载均衡和高可用性

Haproxy 是一款提供高可用和负载均衡的代理软件。为了实现 Mycat 的高可用性,避免 Mycat 服务器的单点故障,文中采用 Haproxy 来检测 Mycat 的存活,如果 Haproxy 发现其中的某个 Mycat 服务器已经宕机,它会把用户的请求自动转发到存活的 Mycat 节点上,从而实现了 Mycat 服务的高可用性,保证了服务的可靠性。同时,随着气象数据的越来越多,种类越来越复杂,单一的 Mycat 节点承载能力有限,过多的数据量随时都有可能单个节点宕机,所以需要再增加一个 Mycat 节点,在这种情况下,就必须要通过负载均衡来实现流量的合理调度。Haproxy 提供了多种负载均衡的算法,文中采用了最常用的轮询算法,Haproxy 会把请求轮询的转发到每一个服务器上,按照权重来分配。

### 1.4 Keepalived 实现 Haproxy 高可用性

为了实现一个高性能,高可靠的分布式数据库服务系统,Haproxy 采用了热备模式,防止单节点故障的发生,文中利用 Keepalived 并编写监测程序实时监控 Haproxy 的健康状态,随时对故障进行转移,从而实现 Haproxy 的高可用,Keepalived 主要是通过虚拟路由冗余协议实现高可用功能。

## 2 性能测试及分析

为了验证基于 Mycat 的分布式数据库的性能,文中搭建了六台虚拟机并部署了分布式数据库测试环境,虚拟机的配置为 CentOS-7.3.1611-x86\_64,32Core,16G Mem。所有应用的部署都是在 Windows 下通过 Xshell 来实现的,所需要的依赖包都是通过 Yum 方式进行安装。

### 2.1 Mysql 数据库读写分离和主从切换测试

读写分离,即把应用对数据库的读操作和写操作放在不同的节点上执行,其中写操作在 Mysql 的主节点上执行,包括 Insert,Delete,Update 操作,读操作在从节点上执行,包括 Select 操作。在测试的过程中,开启 debug 模式,进行 Insert 语句和 Select 语句操作,通过查看日志 mycat.log 确认读和写 SQL 被路由到哪个服务器上,从而实现了数据的读写分离。

主从切换,即当 Mysql 的其中一个节点宕机之后会自动切换到备用节点上,读和写的操作全部由备用节点来承担。主从切换测试之前需要将 dataHost 标签中的 switchType 属性值设置为 1,并配

置心跳检测,停止 Mysql 写节点服务之后,进行写和读操作,查看 mycat.log 确认读和写的 SQL 被路由到 Mysql 的哪个节点上,测试结果可知当 Mysql 写节点宕机之后,读和写的操作全部由 Mysql 的从节点来承载,从而实现了主从切换。

2.2 Mycat 高可用性测试

HAproxy 是专门用来检测 mycat 状态和进行负载均衡的软件,HAproxy 负责将请求分发到 Mycat 上,实现了 Mycat 多节点的集群高可用和负载均衡,通过监听 Mycat 服务器的 48700 端口来实现,通过检测可知 Mycat 状态是否正常,如果正常则显示为

UP,如果异常则显示为 DOWN,同时可以查看 Mycat 服务运行的时长。通过 HAproxy 所在的其中一台服务器访问双机热备的 mycat,本文选取第一台 HAproxy 的服务器,对访问的几种可能性进行了测试,测试结果如表 1 所示。从表 1 可知 HAproxy 只会将请求转发到存活的 Mycat 上,HAproxy 会自动将宕机的 Mycat 服务器从集群中移除,故障恢复后会自动将 Mycat 服务器添加到集群中,HAproxy 这些操作对用户来说是完全透明的,即只要有一台 Mycat 的服务器正常,用户都能够对数据进行访问。

表 1 HAProxy 高可用性测试结果

Mycat 主机名称	Mycat 服务器 IP	Mycat 服务是否启动	HA 访问 Mycat 是否正常
mycat-node1 mycat-node2	192.168.6.113 192.168.6.119	节点 1 和节点 2 上的 Mycat 服务都启动	正常
mycat-node1 mycat-node2	192.168.6.113 192.168.6.119	停止节点 1 上的 Mycat 服务 停止节点 2 上的 Mycat 服务	正常 正常
mycat-node1 mycat-node2	192.168.6.113 192.168.6.119	节点 1 和节点 2 上的 Mycat 服务都停止	不正常

2.3 Keepalived+HAproxy 高可用性测试

为了避免 HAproxy 服务单点故障,文中将 HAproxy 部署为双机热备模式,即当两个节点上的任何一个 HAproxy 服务在线,都能够对外提供服务。HAproxy 的高可用性是通过 Keepalived 来实现的,Keepalived 主要作用是负责抢占虚拟 IP,抢占虚拟 IP 有优先级,一般主服务器比备服务器的优先级高。通过模拟故障,对几种组合进行了测试,测试结果如表 2,首先停止 HAproxy 进程,然后通过虚拟 IP 访问 HAproxy 的 Web 页面,检测该页面是否能够正常提供服务,比如,节点 keep-ha-node1 上的 HAproxy 服

务停止后,通过虚拟 IP 地址 192.168.6.125 (URL: http://192.168.6.125:8099/stats) 访问 HAproxy 的 Web 页面正常,则表明当节点 keep-ha-node1 的 HAproxy 服务宕机之后,节点 keep-ha-node2 上的 Keepalived 抢占了虚拟 IP,虚拟 IP 漂移到节点 keep-ha-node2 上,保证了服务不中断,实现了 HAproxy 的高可用,其它组合情况测试类同。

2.4 分布式数据库读写性能测试

为了能够更准确的测试分布式数据库的性能,文中对单机 Mysql 数据库和基于 Mycat 的分布式数据库分别进行了性能测试,使用 sysbench-0.5<sup>[13]</sup>作

表 2 Keepalived+HAproxy 高可用性测试结果

HAproxy 主机名称	HAproxy 服务器 IP	虚拟 IP	HAproxy 服务是否启动	虚拟 IP 访问 HAproxy 的 Web 是否正常
keep-ha-node1 keep-ha-node2	192.168.6.120 192.168.6.132	192.168.6.125	节点 1 和节点 2 上的 HAproxy 服务都启动	正常
keep-ha-node1 keep-ha-node2	192.168.6.120 192.168.6.132		停止节点 1 上的 HAproxy 服务 停止节点 2 上的 HAproxy 服务	正常 正常
keep-ha-node1 keep-ha-node2	192.168.6.120 192.168.6.132		节点 1 和节点 2 上的 HAproxy 服务都停止	不正常

为压力测试工具,sysbench 是一个模块化的、跨平台和多线程基准测试工具,主要用于评估各种不同系统参数下数据库的负载情况。为了让测试结果更可靠,在整个测试过程中,不允许测试的虚拟机在物理机之间进行迁移,避免不同物理机对测试结果造成的影响。

sysbench-0.5 对数据库 OLTP 基准测试一般分为三个步骤:数据准备阶段;数据测试阶段;数据清理阶段。在数据准备阶段,共有 10 张表,每个表插入 1000 万行数据,数据都是随机产生的,10 张表的数据量大约为 23GB,插入数据耗时 26min。一般应用都会存在热点数据,同时 innodb 引擎也是支持热点数据,所以在测试过程中需要开启热点模式,取 5% 的热点数据,压测时间为 1800s,并发数依次递增 {8,16,32,64,128,256,512},每测试一个用例,需要清理主机的缓存和重启 Mysql 实例,防止缓存影响其它测试结果,sysbench 压力测试工具中的脚本 oltp.lua 中的参数 oltp-read-only=on 表示只进行读操作,oltp-read-only=off 表示进行读写操作,本文共进行了两组基准压力测试,分别记录了 TPS (Transactions Per Second,每秒处理的事务数)、QPS (Queries Per Second,每秒的查询数)、平均响应时间和 95% 的的平均的响应时间,响应时间的单位为秒

(s),在对每一个用例进行测试的同时要监控主机 CPU、内存和磁盘 IO 的使用情况,从而可以快速的定位系统性能的瓶颈。

2.4.1 基准测试一

基准测试一是对单机 Mysql 数据库进行性能测试,使用压力测试工具中的 oltp.lua 脚本,设置相关的参数,分别对 Mysql 数据库进行只读操作和读写操作的压力测试,在测试的过程中,发现主机 CPU 的占有率非常高,使用 perf 工具得知是热点函数 \_raw\_spin\_lock 和 ut\_delay 占用 CPU 资源,这两个函数占用了 78% 的 CPU 资源,这是由于锁的争用导致的。Mysql 提供了控制参数 innodb\_spin\_wait\_delay 和 innodb\_sync\_spin\_loops 需要修改这两个参数的值,修改后 CPU 的使用率降低了,对单机 Mysql 只读压力测试结果如表 3,对单机 Mysql 读写压力测试结果如表 4。

2.4.2 基准测试二

基准测试二是对基于 Mycat 的分布式数据库进行性能测试,分布式数据库的架构为 Mycat+Mysql+Haproxy+Keepalived,设置压力测试工具中 oltp.lua 脚本相关的参数,分别进行只读和读写压力测试,对分布式数据库只读的测试结果如表 5 所示,对分布式数据库读写的测试结果如表 6 所示。

表 3 单机 Mysql 只读压力测试

参数设置	并发数	TPS	QPS	平均响应时间 ( s )	95%平均响应时间( s )
oltp-read-only=on	8	1234.16	17278.29	6.48	7.82
oltp-read-only=on	16	2360.97	33053.54	6.78	7.65
oltp-read-only=on	32	4250.41	59505.62	7.53	8.66
oltp-read-only=on	64	6400.68	89609.56	10.00	11.94
oltp-read-only=on	128	8866.27	124127.75	14.44	22.99
oltp-read-only=on	256	5854.16	81958.28	43.72	71.95
oltp-read-only=on	512	5413.14	75784.00	94.58	158.67

表 4 单机 Mysql 读写压力测试

参数设置	并发数	TPS	QPS	平均响应时间 ( s )	95%平均响应时间( s )
oltp-read-only=off	8	1063.56	19144.41	7.52	9.11
oltp-read-only=off	16	1989.39	35810.86	8.04	9.15
oltp-read-only=off	32	3674.95	66156.51	8.71	10.41
oltp-read-only=off	64	6067.03	109222.15	10.55	15.42
oltp-read-only=off	128	7532.72	135609.28	16.99	30.62
oltp-read-only=off	256	7374.31	132757.43	34.71	76.27
oltp-read-only=off	512	6385.13	114952.29	80.17	241.48



表 5 基于 Mycat 分布式数据库只读压力测试

参数类型	并发数	TPS	QPS	平均响应时间 ( s )	95%平均响应时间( s )
oltp-read-only=on	8	770.83	10791.63	10.38	11.94
oltp-read-only=on	16	1338.21	18734.78	11.95	15.23
oltp-read-only=on	32	2480.46	34726.03	12.91	15.85
oltp-read-only=on	64	3866.44	54129.28	16.55	21.65
oltp-read-only=on	128	5318.82	74462.39	24.06	32.46
oltp-read-only=on	256	6937.35	97120.75	36.91	52.65
oltp-read-only=on	512	8058.76	112818.48	63.53	93.94

表 6 基于 Mycat 分布式数据库读写压力测试

参数类型	并发数	TPS	QPS	平均响应时间 ( s )	95%平均响应时间( s )
oltp-read-only=off	8	483.31	8699.37	16.55	19.88
oltp-read-only=off	16	835.35	15036.49	19.15	23.94
oltp-read-only=off	32	1655.06	29792.04	19.33	26.61
oltp-read-only=off	64	3020.22	54368.23	21.19	29.82
oltp-read-only=off	128	4524.54	81453.22	28.29	42.39
oltp-read-only=off	256	5354.91	96398.52	47.81	73.45
oltp-read-only=off	512	5462.59	98342.64	93.73	155.42

2.5 结果分析

表 3 和表 5 分别是对单点数据库和分布式数据库只读压力测试的结果，从表 3 可知，当并发量为 128 时,Mysql 数据库的性能最佳,QPS 可以达到 12.4 万，随着并发量的增加,QPS 和 TPS 在逐渐减小。平均响应时间在不断的增加,数据库的性能在不断的下降,系统出现了瓶颈,单点的数据库已经无法应对高并发的数据量,从对比实验表 5 可知,随着并发量的增加,TPS 和 QPS 的值一直在递增，并发量越大,性能对比越明显,平均响应时间也要比单点数据库小很多,表明了分布式数据库的优越性。表 4 和表 6 分别是对单点数据库和分布式数据库混合读写压力测试的结果,从表 4 可知,当并发量为 256 时,Mysql 数据库的性能最佳，随着并发量增加,TPS 和 QPS 在不断减小,平均响应时间在增加,单点的数据库性能已经出现了瓶颈,从对比实验表 6 可知,随着并发量的增加,TPS 和 QPS 的值一直在递增，在同等的并发条件下,数据库没有出现瓶颈,从而验证了分布式数据库集群架构 Mycat +Mysql+Haproxy +Keepalived 有效性和可行性。

3 结论与讨论

为了避免收集与分发系统中 Mysql 数据库单点

故障的问题,文中设计了基于Mycat+Mysql+Haproxy+Keepalived 的高可用分布式关系型数据库集群系统，该架构具有 Mycat 高可用、Haproxy 高可用、Mysql 高可用、Mysql 读写分离的特点,同时对 Mysql 单点数据库和基于 Mycat 的分布式数据库进行了压力测试，从不同的指标评价了分布式数据库架构的性能，从测试结果可知该架构可以实现对二维关系型气象数据分散存储和并发访问服务。但是，随着气象数据的不断增加，对分布式数据库的弹性扩展和性能调优是必然的趋势。

因此，在未来的研究工作中主要从以下几个方面进行开展:(1)分布式数据库节点的扩容,即对数据库进行弹性扩展。(2)为了能够提高数据查询的吞吐量,考虑对数据库进行分库和分表。(3)为了增强访问的并发量,对 Mycat 和Mysql 进行调优。

参考文献：

[ 1 ] 区永光. 大数据在气象服务中的应用研究[J]. 科技风, 2020,437(33):84-85.

[ 2 ] 申德荣. 分布式数据库系统原理与应用[M]. 北京:机械工业出版社, 2011.

[ 3 ] 王涛. 分布式数据库应用趋势分析[J]. 金融电子化,2016 (12):44-45.

[ 4 ] 陈宇收. 基于 Mycat 的交通大数据存储方案[J]. 电子技

- 术与软件工程, 2018, 138(16):186–187.
- [5] 张莹, 郑学智. 基于 Mycat 的大数据存储研究[J]. 电子设计工程, 2020, 28(5):1–6.
- [6] 高鹏祥, 陈文俊. 基于 Mycat 的分布式数据库的应用研究[J]. 金融科技时代, 2019, 27(1):50–53.
- [7] 姚丽丽, 万玉建, 朱峰. 基于 MyCat 的建筑能耗监管系统数据库设计与实现[J]. 电脑编程技巧与维护, 2015(24):5–7, 14.
- [8] 曾加帮. 基于 MyCAT 的智能水表系统数据库的研究与实现[D]. 武汉: 华中科技大学, 2017.
- [9] 周智. 基于 Mycat 的分布式数据库在运营商 IT 系统转型中的实现与探索[J]. 电脑知识与技术, 2018, 14(15):25–27.
- [10] 刘锴. 利用 HAProxy 实现选课系统 Web 负载均衡[J]. 电脑知识与技术, 2011, 7(1):35–36.
- [11] Prasetijo A B, Widiyanto E D, Hidayatullah E T. Performance comparisons of web server load balancing algorithms on HAProxy and Heartbeat [C]// 2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering(ICITACEE). IEEE, 2016.
- [12] 钱景辉, 廖锂. 基于 Keepalived 的动态浮动 IP 集群实现[J]. 化工自动化及仪表, 2012, 39(7):926–928.
- [13] 赵吉志. 服务器性能测试利器(之三) 平台处理性能工具 SYSBENCH[J]. 科技浪潮, 2006(12):19–21.

## Application of distributed technology based on Mycat in collection and distribution system

Liu Yang, Gao Peng, Xu Juan, Chen Xuhui

(Gansu Meteorological Information and Technical Equipment Support Center, Lanzhou Gansu 730020)

**Abstract:** The distributed database cluster based on MYCAT adopted the technical architecture of Mycat+Mysql+Haproxy+Keepalived, and realized a highly available, high-performance and high expansion distributed database cluster system. The performance comparison between the stand-alone version of Mysql database and the distributed database cluster based on Mycat was carried out. The results showed that the distributed technology research database cluster system based on Mycat was efficient and feasible, which could solve the problems of host system exception, data service failure, as well as data availability and timeliness reduction caused by hardware failure, software error, and storage damage.

**Key words:** Mycat; distributed; cluster; database; performance