数据库进阶 day2

Mysql读写分离

把客户查询的请求分别给不同的数据库服务器处理

能够实现数据读写分离软件有很多个（如：mycat mysql-porxy maxscale） 通称中间件

原理

多台mysql服务器

-分别提供读，写服务，均衡流量

-通过主从复制保持数据一致性

Mysql 代理面向客户端

-收到sql写请求时，交给服务器A处理

-收到sql读请求时，交给服务器B处理

-具体区分策略由服务器设置

构建思路

已搭建好mysql主从复制

-基于上一个实验的结果

-其中slave为只读

添加一台mysql代理服务器

-部署/启用maxscale

客户端通过代理主机访问mysql数据库

-访问代理服务器

部署maxscale服务

安装一个软件包 maxscale

修改配置文件

9 [maxscale] #定义服务启动线程的数量

10 threads=auto

18 [server1] #指定第1台数据库服务

19 type=server

20 address=192.168.4.51

21 port=3306

22 protocol=MySQLBackend

24 [server2] #指定第2台数据库服务

25 type=server

26 address=192.168.4.52

27 port=3306

28 protocol=MySQLBackend

36 [MySQL Monitor] #定义监控的数据库服务器

37 type=monitor

38 module=mysqlmon

39 servers=server1, server2 //服务器列表

40 user=scalemon //用户名

41 passwd=123qqq...A //密码

42 monitor\_interval=10000

53 #[Read-Only Service] #不定义只读服务

54 #type=service

55 #router=readconnroute

56 #servers=server1

57 #user=myuser

58 #passwd=mypwd

59 #router\_options=slave

64 [Read-Write Service] #定义读写分离服务

65 type=service

66 router=readwritesplit

67 servers=server1, server2 #数据库服务器列表

68 user=maxscale #用户

69 passwd=123qqq...A #密码

70 max\_slave\_connections=100%

76 [MaxAdmin Service] 定义管理服务

77 type=service

78 router=cli

86 #[Read-Only Listener] 不定义只读服务

87 #type=listener

88 #service=Read-Only Service

89 #protocol=MySQLClient

90 #port=4008

92 [Read-Write Listener] #定义读写分离服务使用的端口

93 type=listener

94 service=Read-Write Service

95 protocol=MySQLClient

96 port=4006

98 [MaxAdmin Listener] #定义管理服务使用的端口

99 type=listener

100 service=MaxAdmin Service

101 protocol=maxscaled

102 socket=default

103 port=4016

:wq

数据库调优

搭建监控服务器：监控服务的硬件使用率

Cpu 内存 存储 网络流量

数据库服务运行参数：

提高mysql系统的性能，响应速度

-替换有问题的硬件（cpu/磁盘/内存等）

-服务程序的运行参数调整

-对sql查询进行优化

并发及连接控制

连接数，连接超时

Max\_connections 允许的最大并发连接数

Connect\_timeout 等待连接超时，默认10秒 仅登陆时有效

Wait\_timeout 等待关闭连接的不活动超时秒数 默认28800秒

缓存参数控制

缓冲区，线程数量，开表数量

Key\_buffer\_size #用于myisam引擎的关键索引缓存大小

Sort\_buffer\_size #为每个要排序的线程分配此大小的缓存空间

-增大此值可提高order 和 group 的速度

Read\_buffer\_size #为顺序读取表记录保留的缓存大小

-此缓存值影响SQL查询的响应速度

Thread\_cache\_size #允许保存在缓存中被重用的线程数量

查看可重用线程数

Table\_open\_cache #为所有线程缓存的打开的表的数量

查看当前的线程重用状态

查看已打开，打开过多少个表

Open\_tables

Opened\_tables

查看可缓存多少个打开的表

Tables\_open\_cache

查看缓存的大小

Query\_cache\_limit #大于这个缓存不往缓存里放

Query\_cache\_min\_res\_unit #从物理单元里拿出多少k出来当缓存

Query\_cache\_size #

Query\_cache\_type #

Query\_cache\_wlock\_invalidate #查询缓存写锁有没有开

查看当前查询缓存统计

Show global status like “%qcache%”

日志类型

错误日志

Log-error[=name] #记录启动/运行/停止过程中的错误信息

查询日志

General-log #记录客户端连接和查询操作

General-log-file= #记录客户端连接和查询操作

慢查日志

记录耗时较长或不使用索引的查询操作

Slow-query-log 启用慢查询

Slow-quer-log-file= 指定慢查询日志文件

Long-query-time= 查过时间（默认10秒）

Log-queries-not-using-indexes 记录未使用索引的查询