微信开源PhxSQL：高可用、强一致的MySQL集群

作者：R

原文链接：https://ld246.com/article/1472629693746

来源网站：链滴

PhxSQL是由微信后台团队自主研发的一款服务高可用、数据强一致的分布式数据库服务。该服务基Percona5.6搭建，目标在于解决MySQL在容灾和数据一致性方面的不足，并大幅简化了MySQL容切换的运维操作。

作者：Junchao Chen (junechen@tencent.com), Haochuan Cui (lynncui@tencent.com), Duokai Huang (mariohuang@tencent.com), Ming Chen (mingchen@tencent.com) 和 Sifan Liu (stephnliu@tencent.com)

总览

- PhxSQL具有服务高可用、数据强一致、高性能、运维简单、和MySQL完全兼容的特点。
- 服务高可用：PhxSQL集群内只要多数派节点存活就能正常提供服务；出于性能的考虑，集群会选出一个Master节点负责写入操作；当Master失效，会自动重新选举新的Master。
- 数据强一致：PhxSQL采用多节点冗余部署，在多个节点之间采用paxos协议同步流水，保证了集内各节点数据的强一致。
- 高性能：PhxSQL比MySQL SemiSync的写性能更好，得益于Paxos协议比SemiSync协议更加高效；
- 运维简单：PhxSQL集群内机器出现短时间故障，能自动恢复数据，无需复杂的运维操作；PhxSQL提供一键更换（新增/删除）集群内的机器，简化运维的工作。
- MySQL完全兼容：PhxSQL是基于Percona的研发，完全兼容MySQL的操作命令。 可通过MySQL供的mysqlclient/perconaserverclient直接操作PhxSQL。

项目中包含PhxSQL源代码，源代码编译时所需要的一些第三方库，及可直接在Linux环境下运行的二制包。其中代码使用到了微信团队自研的另外三个开源项目（phxpaxos,phxrpc,colib)。若需编译源码，需额外下载，也可以在clone时通过--recurse-submodule获得代码。

phxpaxos项目地址： http://github.com/tencent-wechat/phxpaxos

phxrpc项目地址： http://github.com/tencent-wechat/phxrpc

colib项目地址： http://github.com/tencent-wechat/libco

PhxSQL编译

如果是直接使用二进制包，请跳过这节。

PhxSQL主要目录结构

- PhxSQL
  - phxsqlproxy
  - phxbinlogsvr
  - percona
  - phx_percona
    - plugin
    - phxsync_phxrpc
    - semisync

- third_party
    - glog
    - leveldb
    - protobuf
    - phxpaxos
    - colib
    - phxrpc
  - tools
  - phxrpc_package_config

主要目录介绍

目录名	
phxsqlproxy	phxsqlproxy模块，负责接入请求。
phxbinlogsvr	负责MySQL binlog数据同步，master管理等。
percona	percona5.6.31-77.0的官方源代码
phx_percona/plugin/phxsync_phxrpc	percon用于跟phxbinlogsvr同步binlog的插件。
phx_percona/plugin/semisync	因兼容问题，改了semisync的部分代码，目录内为修改过的文件。
third_party/glog	Google Glog 第三方库
third_party/leveldb	Google Leveldb 第三方库
third_party/protobuf	Google Protobuf 3.0+三方库
third_party/phxpaxos	paxos协议库。用于phxbilogsvr之间同步binlog
third_party/colib	协程基础库。phxsqlproxy使用
third_party/phxrpc	rpc框架。phxbinlogsvr使用

事前准备

安装第三方库

PhxSQL需要用到一些第三方库（glog, leveldb, protobuf, phxpaxos, colib, phxrpc）,下载后安装到hxSQL的third_party目录下或者把安装目录链接到PhxSQL的third_party目录下。

需保证第三方库glog, protobuf在configure时带上-fPIC选项(configure CXXFLAGS=-fPIC)，并定--prefix=当前目录绝对路径。

比如configure CXXFLAGS=-fPIC --prefix=/home/root/phxsql/third_party/glog.

下载 percona-server-5.6.31-77.0.tar.gz

将percona-server-5.6_5.6.31-77.0源代码放到PhxSQL目录下，更名或链接为percona**（请注意能使用percona-server-5.6_5.6.31-77.0版本）**

生成安装环境

1. 在PhxSQL目录下执行 sh autoinstall.sh && make && make install
2. 若想打包二进制运行包（集群运行时所需要的所有文件和配置） make package
3. install完成后，二进制会生成到PhxSQL目录下的sbin目录，运行所需要的相关文件和配置会安装到hxSQL目录下的install_package目录。打包二进制运行包会把install_package进行tar格式的打包，生成phxsql.tar.gz。若想更改install的安装目录，可在sh autoinstall.sh 后加入-prefix=路径

PhxSQL部署和运行

准备机器

PhxSQL需要在2台或以上的机器集群上运行（建议集群内机器数目n>=3 且 n为奇数）

初始化PhxSQL

1. 把 phxsql.tar.gz传到集群内的所有机器,对集群内每台机器按以下步骤进行安装
    1. 解压 phxsql.tar.gz
    2. 进入 phxsql/tools, 并使用python install.py --help查看安装参数。(例子:python2.7 install.p -i"your_inner_ip" -p 54321 -g 6000 -y 11111 -P 17000 -a 8001 -f/tmp/data/)
2. 任意找一台机器， cd phxsql/sbin, 执行

./phxbinlogsvr_tools_phxrpc -f InitBinlogSvrMaster -h"ip1,ip2,ip3" -p 17000 (17000为phxbinlosvr监听的端口)

3. 看到master初始化完成的消息后，集群即可使用。
4. 可通过 mysql -uroot -h"your_inner_ip" -P$phxsqlproxy_port进入PhxSQL执行读写操作确认PxSQL已正常运行

简单测试

1. 在PhxSQL目录下进入tools目录
2. 执行 test_phxsql.sh port ip1 ip2 ip3，port为phxsqlproxy_port, ip为集群机器ip

配置文件说明

PhxSQL一共有3个配置文件

1. my.cnf： MySQL的配置,请根据你的业务需求进行修改（安装前请修改tools/etc_template/my.nf，安装后请修改etc/my.cnf)

2. phxbinlogsvr.conf

Section name	Key name	comment
AgentOption	AgentPort	Phxbinlogsvr监听MySQL访问的端口，用于MySQL和binlogsvr之间的通信
	EventDataDir	Phxbinlogsv数据存放目录
	MaxFileSize	Phxbinlogsvr每数据文件的大小，数据文件过大会导致启动过慢，数据文件过小会导致文件数过多，单位为B
	MasterLease	Phxbinlogsv的master租约时间，单位为s
	CheckPointTime	Phxbinlogsv会删除CheckPointTime时间前的数据，但如果被删数据中存在其他MySQL还没学到的，则不会删除部分数据，单位为分钟
	MaxDeleteCheckPointFileNum	hxbinlogsvr删数据时，每次删除的最大文件数
	FollowIP	机器为folloer机器，负责拉取FollowIP机器上的数据，不参与集群的投票
PaxosOption	PaxosLogPath	Phbinlogsvr中paxos库的数据目录
	PaxosPort	Phxbinlogsvr中paxo库的通信端口
	PacketMode	Phxbinlogsvr在pxos协议中是否增大包的大小限制， 1为每个包的大小为100m，但超时限制变为1分钟，0 为每个包大小为50m，超时限制2s起（动态变化）
Server	IP	Phxbinlogsvr的监听ip
	Port	Phxbinlogsvr的监听端口
	LogFilePath	Phxbinlogsvr的日目录
	LogLevel	Phxbinlogsvr的日志别

3. phxsqlproxy.conf

Section name	Key name	comment
Server	IP	phxsqlproxy的监听ip
	Port	phxsqlproxy的监听端口
	QSLogFilePath	phxsqlproxy日志目录
	QSLogLevel	phxsqlproxy的日级别

PhxSQL使用

phxsqlproxy为PhxSQL的接入层，所有的请求均经过phxsqlproxy,再透传给MySQL。

phxsqlproxy提供两个端口进行读写

读写端口

该端口号为phxsqlproxy.conf配置中的端口号, 用户连接上proxyA的此端口，proxyA会自动把请求由到Master机器，然后再对Master机器上的MySQL进行操作。

只读端口

该端口号为读写端口号+1, 用户连接上此端口时，会对本机的MySQL进行操作(但若本机为master，phxsqlproxy会把请求转发到其他phxsqlproxy的只读端口）。

PhxSQL的使用

1. 通过mysql命令连接上 phxsqlproxy，然后执行命令mysql -uroot -h$phxsqlproxyip -P$phxsqproxyport -ppwd

2. 进行sql命令操作

phxsqlproxyip为集群内的任意一台phxsqlproxy的ip

phxsqlproxyport为phxsqlproxy端口(读写/只读)

PhxSQL管理

PhxSQL提供一个工具phxbinlogsvr_tools_phxrpc来方便管理者对PhxSQL的运营管理。

PhxSQL集群中对MySQL的管理使用两个账号管理员帐号和数据同步账号。管理员账号默认账号密码（"root"，""），数据同步账号默认账号密码为（"replica","replica123"）。账号密码的更改通过工才执行（工具会直接操作MySQL数据，不需要人工进行MySQL操作）。

phxbinlogsvr_tools -f GetMasterInfoFromGlobal -h <host> -<port>

功能：集群的master机器ip和超时时间

参数:

- **Host:** 集群中的其中一台机器ip
- **Port:** phxbinlogsvr的监听port

phxbinlogsvr_tools -f SetMySqlAdminInfo -h <host> -p <port> -u <admin username> -d <admin pwd> -U <new admin username> -D <new admin pwd>

功能： 设置 PhxSQL管理员账号密码

参数：

- **Host:** 集群中的其中一台机器ip
- **Port:** phxbinlogsvr的监听port
- **Admin username:** 当前的管理员账号（默认为root）
- **Admin pwd:** 当前的管理员密码（默认为空）
- **New admin username:** 新的管理员账号
- **New admin pwd:** 新的管理员密码

phxbinlogsvr_tools -f SetMySqlReplicaInfo -h <host> -p <port> -u <admin username> -d <admin pwd> -U <new replica sername> -D <new replica pwd>

功能： 设置PhxSQL同步数据账号密码

参数：

- **Host:** 集群中的其中一台机器ip
- **Port:** phxbinlogsvr的监听port
- **Admin username:** 当前的管理员账号（默认为root）
- **Admin pwd:** 当前的管理员密码（默认为空）
- **New replica username:** 新的同步数据账号
- **New replica pwd:** 新的同步数据密码

phxbinlogsvr_tools_phxrpc -f GetMemberList -h <host> -port>

功能：集群的master机器ip和超时时间

参数：

- **Host:** 集群中的其中一台机器ip
- **Port:** phxbinlogsvr的监听port

phxbinlogsvr成员管理

移除成员

执行工具命令将机器A移除集群phxbinlogsvr_tools_phxrpc -f RemoveMember -h<host> -p<port> -m <机器A的ip>

执行成功后，机器A将会在一段时间后不在接收数据

添加成员

1. 执行工具将机器A加入到集群　phxbinlogsvr_tools -f AddMember -h<host> -p<port> -m <器A的ip>

2. 在新机器A上安装好PhxSQL

根据 "初始化PhxSQL" 步骤安装PhxSQL

3. 执行成功后，机器A的phxbinlogsvr将会在一段时间后开始接收数据

4. 从集群内任意一台机器的percona导出一份镜像数据

5. kill掉机器A上的phxbinlogsvr, 并通过本地的MySQL端口进入MySQL，执行　set super_read_onl = 0; set read_only = 0;

6. 将镜像数据导入到机器A上的percona，并kill掉机器A上的phxbinlogsvr

7. 一段时间（~1分钟）后，机器A开始正常工作

phxbinlogsvr故障处理

当机器出现问题，PhxSQL无法正常启动时，可以选择在该机器上重装PhxSQL，重装过程可参考4.2

在重装过程中，phxbinlogsvr可能会拉取集群内其他机器checkpoint来启动，待checkpoint拉取结后，phxbinlogsvr会自杀（为了确保数据安全），日志中会出现"All sm load state ok, start to exit p ocess"，此时须重新启动phxbinlogsvr，启动后会正常运行。

当MySQL数据出现问题时，phxbinlogsvr会停止工作，此时需要检查MySQL的binlog数据是否正。

当出现问题时，可观察日志中带有err标志的红色字体的日志，来确认是否有异常。

性能测试

机型信息

CPU :　Intel(R) Xeon(R) CPU E5-2420 0 @ 1.90GHz * 24

内存 : 32G

磁盘 : SSD Raid10

网络互Ping耗时

Master -> Slave : 3 ~ 4ms

Client -> Master : 4ms

压测工具和参数

sysbench　--oltp-tables-count=10 --oltp-table-size=1000000 --num-threads=500 --max-requ sts=100000 --report-interval=1 --max-time=200

压测结果

Client线程数	测试集群	测试集合				
		insert.lua (100%写)		select.lua (0%写)		LTP.lua (20%写)
		QPS	耗时	QPS	耗时	QPS
200	PhxSQL	5076	39.34/56.93	46334	.21/5.12	25657
140.16/186.39						
200	MySQL半同步	4055	49.27/66.64	47528	.10/5.00	20391
176.39/226.76						
500	PhxSQL	8260	60.41/83.14	105928	.58/5.81	46543
192.93/242.85						
500	MySQL半同步	7072	70.60/91.72	121535	.17/5.08	33229
270.38/345.84						

注：耗时分别为测试结果的平均耗时/95%分线耗时，单位ms