

•数据赋能•价值创新•

# ZQPool第一讲

中启乘数科技（杭州）有限公司

唐成



01. 个人介绍

# 目录

CONTENTS

02. ZQPool  
研发初衷



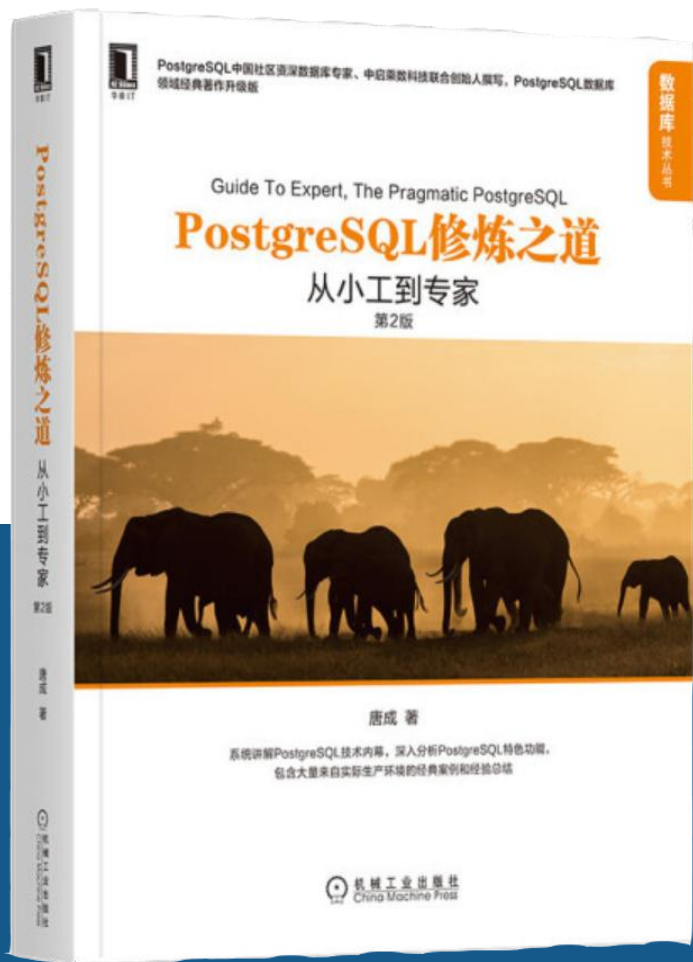


NO.1

# 个人介绍

唐成

# 个人介绍



## 唐成

- 《PostgreSQL修炼之道：从小工到专家》的作者，现任PostgreSQL中文社区常委。从业20多年，拥有20年数据库、操作系统、存储领域的工作经验，历任过网易研究院技术专家、阿里巴巴高级数据库专家，从事过阿里巴巴Greenplum、PostgreSQL数据库的架构设计和运维。
- 资深架构师：既熟悉数据库的运维工作，又精通开发工作。是最早的Oracle 9i的OCP，精通c/c++和python，擅长使用C语言做数据库内核心开发，做过数百TB以上的Greenplum集群的维护和扩容工作，有丰富的Greenplum的经验，解决过很多PostgreSQL、Greenplum方面的疑难杂症。

## 中启乘数科技（杭州）有限公司

- 是以“数据赋能、价值创新”为发展战略和业务定位的高科技创新企业，围绕大中型企业用户的关系型数据库及分布式存储的相关技术领域，提供产品解决方案
- 中启乘数推出数据库私有云平台、分布式存储云平台等一系列自主知识产权的企业级产品，为金融、通信、能源、政府行业等大型用户构建安全可控、性能高效的数据环境
- 中启乘数创始团队为中国数据库社区核心成员，拥有兼具专业技术和丰富经验的研发团队，其核心成员来自阿里、Oracle、Teradata、网易等企业领域的技术专家。



NO.2

# ZQPool研发初衷

应对业务高并发访问

# 数据库侧的连接池的作用



减少到数据库上的连接数。应用程序到连接池软件上有M个连接，这M个连接不是同时都繁忙的，这M个连接上同一个时刻发来的并发SQL可能只有N个（N通常大大小于M），这样连接池软件只需要在后端数据库上建N个连接。就可以满足了要求。这个场景通常是java应用。

减少短连接应用花在新建数据库连接的时间。PostgreSQL数据库对每一个连接需要fork出一个新的进程来提供服务，而每次fork一个进程是需要时间的。而连接池软件可以预先建好到数据库的连接，应用程序连接到连接池软件后，连接池软件可以从池中取一个已经建好的连接马上提供服务，这样就大大减少了新连接的时间。这个场景的典型应用是php应用。php应用到数据库通常是短连接。



# Java应用已经有连接池，为什么还需要数据库侧连接池？



Java应用通常部署在非常多的机器上，无法降低到数据库的连接数。



我们可以想象一个场景：一个java应用可能部署在200台主机上，而每个主机上java应用自身会开启一个java连接池，这个java连接池假设开20个连接，这时到数据库上就有

$200 \times 20 = 4000$ 个连接，这些连接实际上多数时间都是空闲的，少数时间才是活跃的。4000个连接，PostgreSQL数据库就需要启动4000个进程，太多连接会降低数据库的效率。



把到数据库访问的服务都微服务化？然后让这些微服务部署在少数机器上。这样可以降低到数据库的总连接数。但比较麻烦。



# 为什么pgbouncer无法降低java应用到数据库的连接数?

## pgbouncer的三种连接池模式

要减少到数据库上的连接数，pgbouncer连接池的模式只能配置成语句级或事务级，不能配置成会话级。因为pgbouncer在会话级下，前面来多少个连接，到数据库也必须建多少个连接，起不到减少数据库连接的目的。

当我们把pgbouncer配置成语句级或事务级时，java应用连接pgbouncer会报错：

**org.postgresql.util.PSQLException: ERROR: prepared statement "S\_1" already exists**



语句级



事务级



会话级



# 为什么pgbouncer无法降低java应用到数据库的连接数?



**ZQPool通过记录一个连接上的Prepare SQL的名字并替换成不重复的名字的方式解决了这个问题**

## Jdbc执行SQL的步骤

- Prepare的SQL执行的两个步骤：“prepare S\_1 as select \* from test01 where id=\$1;” 然后再 “execute S1(1);”
- 当 “prepare S\_1 as select \* from test01 where id=\$1;” 这条SQL来的时候，从连接池中拿一个连接，执行了后，就释放了此连接；
- 然后再从连接池中取一个连接，然后执行：execute S1(1); 这个连接如果不是之前的连接，则因为无prepare语句S1而报错。
- 又执行另一个SQL，可能从连接池中取到的时之前的连接，然后再执行 “prepare S\_1 as select \* from test02 where id=\$1;” ，
- 但这个prepare SQL 的名字S\_1已经被前面的SQL占用，这时就报上面的错了。
- 注：jdbc实际的行为比上面的更复杂，但原理大致就是上面描述的这个过程。

# ZQPool解决pgbouncer不能利用多核的问题

PgBouncer处理SQL的转发只能用到CPU的一个核，因为pgbouncer是单线程程序。对于高并发的情况下，超过单核的性能时，就会立即出现瓶颈。而ZQPool是使用golang的协程技术，可以利用了多核的性能

## pgbouncer的情况

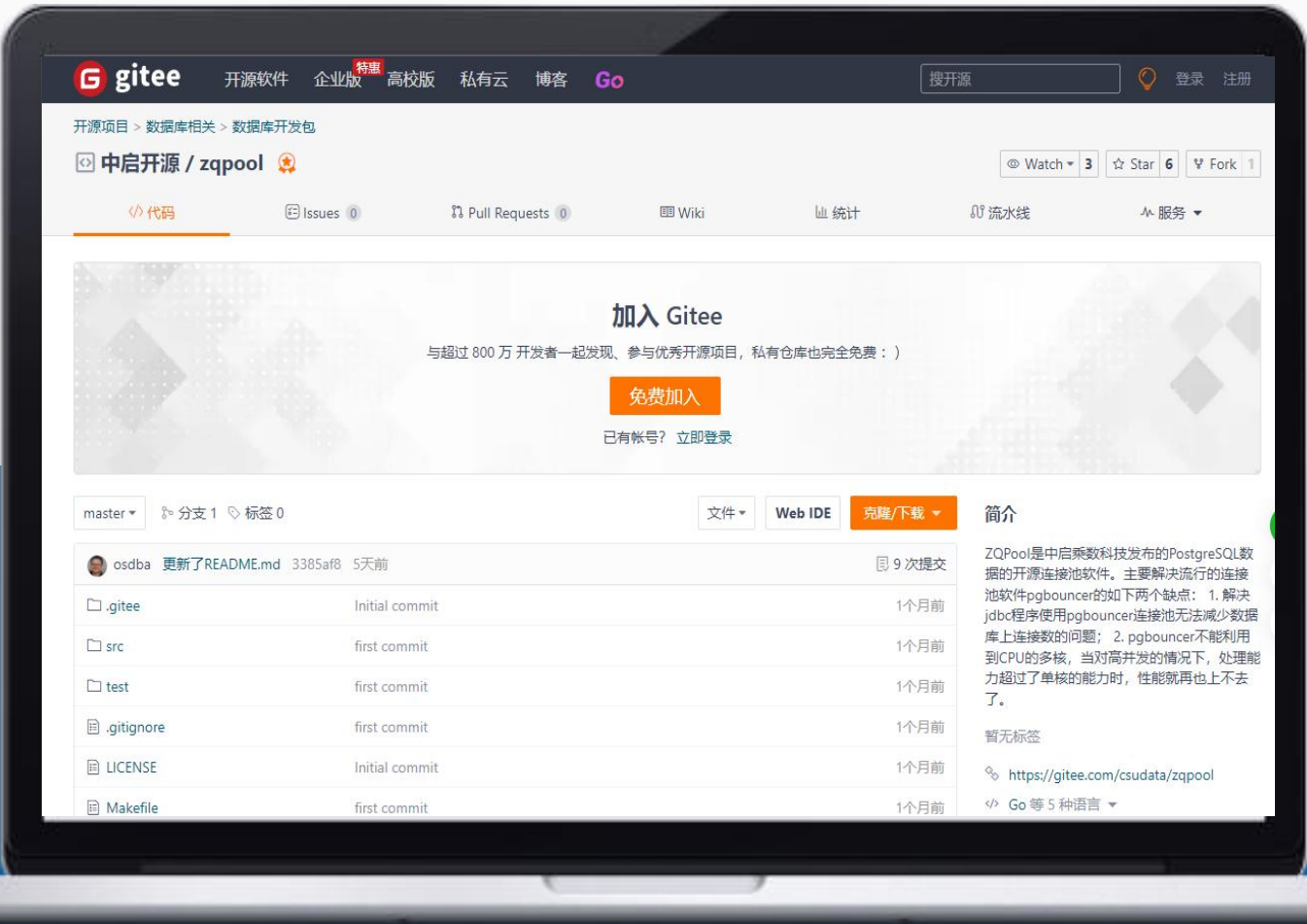
```
1. [postgres@csyun01 ~]$ pgbench -h 10.197.160.18 -p 6432 -Uu01 -S -P 2 -T 30 -c 32
2. pgbench (14.3)
3. starting vacuum...end.
4. progress: 2.0 s, 30407.5 tps, lat 1.050 ms stddev 0.180
5. progress: 4.0 s, 30108.6 tps, lat 1.062 ms stddev 0.182
6. progress: 6.0 s, 30231.5 tps, lat 1.058 ms stddev 0.179
7. progress: 8.0 s, 31157.9 tps, lat 1.026 ms stddev 0.176
8. progress: 10.0 s, 30491.7 tps, lat 1.049 ms stddev 0.178
9. progress: 12.0 s, 30463.0 tps, lat 1.050 ms stddev 0.180
10. progress: 14.0 s, 30366.2 tps, lat 1.053 ms stddev 0.179
11. progress: 16.0 s, 30177.5 tps, lat 1.060 ms stddev 0.180
12. progress: 18.0 s, 30067.1 tps, lat 1.064 ms stddev 0.181
13. progress: 20.0 s, 30420.1 tps, lat 1.051 ms stddev 0.177
14. ...
15. ...
16. ...
```

## ZQPool的情况

```
1. [postgres@csyun01 ~]$ pgbench -h 10.197.160.18 -p 5436 -Uu01 -S -P 2 -T 30 -c 32
2. Password:
3. pgbench (14.3, server 10.5)
4. starting vacuum...end.
5. progress: 2.0 s, 111134.7 tps, lat 0.213 ms stddev 0.058
6. progress: 4.0 s, 112688.1 tps, lat 0.209 ms stddev 0.058
7. progress: 6.0 s, 114570.8 tps, lat 0.207 ms stddev 0.054
8. progress: 8.0 s, 107305.3 tps, lat 0.216 ms stddev 0.066
9. progress: 10.0 s, 108680.1 tps, lat 0.215 ms stddev 0.063
10. progress: 12.0 s, 108867.6 tps, lat 0.214 ms stddev 0.064
11. ...
12. ...
13. ...
```



# ZQPool开源在gitee上



<https://gitee.com/csudata/zqpool>

## ZQPool交流微信群

---



欢迎各位扫码进群交流

个人微信号: osdba0



•数据赋能•价值创新•

# 感谢观看

中启乘数科技（杭州）有限公司

唐成