



## 阿里云数据库ApsaraDB

手机版

关注

云栖社区 > 阿里云数据库ApsaraDB > 博客 > 正文

编程语言系列讲座，四大编程语言全面盘点

报名直播

# JedisPool资源池优化

carlosfu | 2017-11-08 17:22:25 | 浏览1937 | 评论0

redis | 监控 | Apache | 配置 | 云数据库 | exception | Ping | 分布式系统 | jedis | 云数据库Redis版

**摘要：**合理的JedisPool资源池参数设置能为业务使用Redis保驾护航，本文将对JedisPool的使用、资源池的参数进行详细说明，最后给出“最合理”配置。

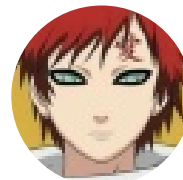
## 背景

合理的JedisPool资源池参数设置能为业务使用Redis保驾护航，本文将对JedisPool的使用、资源池的参数进行详细说明，最后给出“最合理”配置。

## 一、使用方法

以官方的2.9.0为例子(Jedis Release)，Maven依赖如下：

## 达人介绍



carlosfu



文章 5篇 | 关注 44

关注

## 文中提到的云产品

### 云数据库 Redis 版

一种稳定可靠、性能卓越、可弹性伸缩的数据库服务。基于飞天分布式系统和全SSD盘高性能存储，支... 查看详情

## 博主其他文章

更多>

- 阿里云Redis开发规范
- Redis大数据应用场景
- Redis c#客户端StackExchange.Redis简单使用
- Jedis常见异常汇总

```
<dependency>
  <groupId>redis.clients</groupId>
  <artifactId>jedis</artifactId>
  <version>2.9.0</version>
  <scope>compile</scope>
</dependency>
```

Jedis使用apache commons-pool2对Jedis资源池进行管理，所以在定义JedisPool时一个很重要的参数就是资源池GenericObjectPoolConfig，使用方式如下，其中有很多资源管理和使用的参数(具体看第二节)

注意：后面会提到建议用**JedisPoolConfig**代替**GenericObjectPoolConfig**

```
GenericObjectPoolConfig jedisPoolConfig = new GenericObjectPoolConfig();
jedisPoolConfig.setMaxTotal(..);
jedisPoolConfig.setMaxIdle(..);
jedisPoolConfig.setMinIdle(..);
jedisPoolConfig.setMaxWaitMillis(..);
...
```

JedisPool的初始化如下：

## 相关话题

[更多>](#)

- 体验全新的备份数据上云功能，赢阿里云代金券！
- 传统行业的数据库如何迎接云时代，互动送云栖大会好礼
- 云数据库：未来10年指数级数据增长背后的技术要塞，云栖互动送好礼
- 云数据库POLARDB劲风来袭，选出你最爱的产品功能
- 顶级数据库ICDE会议POLARDB首开专场



```
// redisHost和redisPort是实例的IP和端口
// redisPassword是实例的密码
// timeout, 这里既是连接超时又是读写超时, 从Jedis 2.8开始有区分connectionTimeout和soTimeout

JedisPool jedisPool = new JedisPool(jedisPoolConfig, redisHost, redisPort, timeout, redisPassword);

执行命令如下：
Jedis jedis = null;
try {
    jedis = jedisPool.getResource();
    //具体的命令
    jedis.executeCommand()
} catch (Exception e) {
    logger.error(e.getMessage(), e);
} finally {
    //注意这里不是关闭连接, 在JedisPool模式下, Jedis会被归还给资源池。
    if (jedis != null)
        jedis.close();
}
```

二、参数说明

JedisPool保证资源在一个可控范围内，并且提供了线程安全，但是一个合理的GenericObjectPoolConfig配置能为应用使用Redis保驾护航，下面将对它的一些重要参数进行说明和建议：

在当前环境下，Jedis连接就是资源，JedisPool管理的就是Jedis连接。

1. 资源设置和使用

序号	参数名	含义	默认值	使用建议
1	maxTotal	资源池中最大连接数	8	设置建议见下节

序号	参数名	含义	默认值	使用建议
2	maxIdle	资源池允许最大空闲的连接数	8	设置建议见下节
3	minIdle	资源池确保最少空闲的连接数	0	设置建议见下节
4	blockWhenExhausted	当资源池用尽后，调用者是否要等待。 只有当为true时，下面的maxWaitMillis才会生效	true	建议使用默认值
5	maxWaitMillis	当资源池连接用尽后，调用者的最大等待时间(单位为毫秒)	-1：表示永不超时	不建议使用默认值
6	testOnBorrow	向资源池借用连接时是否做连接有效性检测(ping)，无效连接会被移除	false	业务量很大时候建议设置为false(多一次ping的开销)。
7	testOnReturn	向资源池归还连接时是否做连接有效性检测(ping)，无效连接会被移除	false	业务量很大时候建议设置为false(多一次ping的开销)。
8	jmxEnabled	是否开启jmx监控，可用于监控	true	建议开启，但应用本身也要开启

## 2.空闲资源监测

空闲Jedis对象检测，下面四个参数组合来完成，testWhileIdle是该功能的开关。

序号	参数名	含义	默认值	使用建议
1	testWhileIdle	是否开启空闲资源监测	false	true
2	timeBetweenEvictionRunsMillis	空闲资源的检测周期(单位为毫秒)	-1 : 不检测	建议设置，周期自行选择，也可以默认也可以使用下面JedisPoolConfig中的配置
3	minEvictableIdleTimeMillis	资源池中资源最小空闲时间(单位为毫秒)，达到此值后空闲资源将被移除	10006030 = 30分钟	可根据自身业务决定，大部分默认值即可，也可以考虑使用下面JedisPoolConfig中的配置
4	numTestsPerEvictionRun	做空闲资源检测时，每次的采样数	3	可根据自身应用连接数进行微调，如果设置为-1，就是对所有连接做空闲监测

为了方便使用，Jedis提供了JedisPoolConfig，它本身继承了GenericObjectPoolConfig设置了一些空闲监测设置

```
public class JedisPoolConfig extends GenericObjectPoolConfig {  
    public JedisPoolConfig() {  
        // defaults to make your life with connection pool easier :)  
        setTestWhileIdle(true);  
        //  
        setMinEvictableIdleTimeMillis(60000);  
        //  
        setTimeBetweenEvictionRunsMillis(30000);  
        setNumTestsPerEvictionRun(-1);  
    }  
}
```

所有默认值可以从org.apache.commons.pool2.impl.BaseObjectPoolConfig中看到。

### 三、资源池大小(maxTotal)、空闲(maxIdle minIdle)设置建议

#### 1.maxTotal：最大连接数

实际上这个是一个很难回答的问题，考虑的因素比较多：

- 业务希望Redis并发量
- 客户端执行命令时间
- Redis资源：例如 nodes(例如应用个数) \* maxTotal 是不能超过redis的最大连接数。
- 资源开销：例如虽然希望控制空闲连接，但是不希望因为连接池的频繁释放创建连接造成不必靠开销。

以一个例子说明，假设：

- 一次命令时间 (borrow|return resource + Jedis执行命令(含网络)) 的平均耗时约为1ms，一个连接的QPS大约是1000
- 业务期望的QPS是50000

那么理论上需要的资源池大小是50000 / 1000 = 50个。但事实上这是个理论值，还要考虑到要比理

论值预留一些资源，通常来讲maxTotal可以比理论值大一些。

但这个值不是越大越好，一方面连接太多占用客户端和服务端资源，另一方面对于Redis这种高QPS的服务器，一个大命令的阻塞即使设置再大资源池仍然会无济于事。

## 2. maxIdle minIdle

maxIdle实际上才是业务需要的最大连接数，maxTotal是为了给出余量，所以maxIdle不要设置过小，否则会有new Jedis(新连接)开销，而minIdle是为了控制空闲资源监测。

连接池的最佳性能是maxTotal = maxIdle ,这样就避免连接池伸缩带来的性能干扰。但是如果并发量不大或者maxTotal设置过高，会导致不必要的连接资源浪费。

可以根据实际总OPS和调用redis客户端的规模整体评估每个节点所使用的连接池。

## 3.监控

实际上最靠谱的值是通过监控来得到“最佳值”的，可以考虑通过一些手段(例如jmx)实现监控，找到合理值。

# 四、常见问题

## 1.资源“不足”

```
redis.clients.jedis.exceptions.JedisConnectionException: Could not get a resource from
...
Caused by: java.util.NoSuchElementException: Timeout waiting for idle object
at org.apache.commons.pool2.impl.GenericObjectPool.borrowObject(GenericObjectPool.java
```

或者

```
redis.clients.jedis.exceptions.JedisConnectionException: Could not get a resource from
...
Caused by: java.util.NoSuchElementException: Pool exhausted
at org.apache.commons.pool2.impl.GenericObjectPool.borrowObject(GenericObjectPool.java
```

两种情况均属于无法从资源池获取到资源，但第一种是超时，第二种是因为blockWhenExhausted为false根本就不等。

遇到此类异常，不要盲目的认为资源池不够大，第三节已经进行了分析。具体原因可以排查：网络、资源池参数设置、资源池监控(如果对jmx监控)、代码(例如没执行jedis.close())、慢查询、DNS等问题。

具体可以参考该文章：<https://www.atatech.org/articles/77799>

## 2. 预热JedisPool

由于一些原因(例如超时时间设置较小原因)，有的项目在启动成功后会出现超时。JedisPool定义最大资源数、最小空闲资源数时，不会真的把Jedis连接放到池子里，第一次使用时，池子没有资源使用，会new Jedis，使用后放到池子里，可能会有一定的时间开销，所以也可以考虑在JedisPool定义后，为JedisPool提前进行预热，例如以最小空闲数量为预热数量



```

List<Jedis> minIdleJedisList = new ArrayList<Jedis>(jedisPoolConfig.getMinIdle());

for (int i = 0; i < jedisPoolConfig.getMinIdle(); i++) {
    Jedis jedis = null;
    try {
        jedis = pool.getResource();
        minIdleJedisList.add(jedis);
        jedis.ping();
    } catch (Exception e) {
        logger.error(e.getMessage(), e);
    } finally {
    }
}

for (int i = 0; i < jedisPoolConfig.getMinIdle(); i++) {
    Jedis jedis = null;
    try {
        jedis = minIdleJedisList.get(i);
        jedis.close();
    } catch (Exception e) {
        logger.error(e.getMessage(), e);
    } finally {
    }
}
}

```

## 广告

云数据库Redis版（ApsaraDB for Redis）是一种稳定可靠、性能卓越、可弹性伸缩的数据库服务。基于飞天分布式系统和全SSD盘高性能存储，支持主备版和集群版两套高可用架构。提供了全套的容灾切换、故障迁移、在线扩容、性能优化的数据库解决方案。欢迎各位购买使用:云数据库 Redis 版

招聘：[阿里云-技术专家-KVstore](#)

## 岗位描述：

- 负责阿里云Redis源码开发维护
- 负责阿里云Redis cluster开发与设计

## 岗位要求：

- 精通C/C++，熟悉TCP，Linux Kernel等优先
- 数据结构，算法等基础知识扎实
- 5年后台系统的设计与开发，或3年分布式系统的设计与开发，运维过大型分布式系统
- 精通至少一项开源NoSQL产品。Redis，mongodb,memcached等优先。
- 有云服务产品或基于SSD的系统开发经验优先
- 善于创新，乐于挑战，有责任心，良好团队精神
- 良好的表达能力，能够清晰和准确地描述问题，发现并解决问题能力

► 本文为云栖社区原创内容，未经允许不得转载，如需转载请发送邮件至yqeditor@list.alibaba-inc.com；如果您发现本社区中有涉嫌抄袭的内容，欢迎发送邮件至：yqgroup@service.aliyun.com 进行举报，并提供相关证据，一经查实，本社区将立刻删除涉嫌侵权内容。



用云栖社区APP，舒服~

【云栖快讯】直播推荐——现在报名3月12日编程语言系列讲座，与行业资深专家一起学习Python、C++、JavaScript、Java！还可在活动页面领取红包，百分百中奖哦！[详情请点击](#)

评论 (0)    点赞 (3)    收藏 (3)

分享到:



相关文章

- Jedis常见异常汇总
- spring整合redis客户端及缓存接口设计（转）
- 阿里云Redis开发规范
- Redis Primer（1）基于JedisPool的R...
- NoSQL之Redis（五）--- Redis测试以及其...
- Jedis操作Redis数据库
- spring boot redis缓存JedisPool...
- Redis入门Jedis
- Jedis介绍及常见问题分析
- jedis连接池详解(Redis)

网友评论

登录后可评论，请 [登录](#) 或 [注册](#)

- 热点导航
- 用户关注
- 更多推荐
- 闲时流量包
- 自动化测试
- 用户体验
- 阿里云大学
- 云计算
- 解决方案
- 云数据库Rds
- cn域名
- 网络安全
- linux命令
- 负载均衡
- Js
- 互联网架构
- 云服务
- 域名注册
- Mysql
- ECS升级配置
- JavaScript 函数
- Whois查询
- 移动站
- 物联网
- 服务器监控
- 数据可视化
- IT论坛
- 教程
- Python语言
- ICP备案查询
- 企业邮箱
- PHP
- 移动数据分析
- 主题地图
- 签名文件

[阿里巴巴集团](#) [淘宝网](#) [天猫](#) [聚划算](#) [全球速卖通](#) [阿里巴巴国际交易市场](#) [1688](#) [阿里妈妈](#) [飞猪](#) [阿里云计算](#) [AliOS](#) [阿里通信](#) [万网](#) [高德](#) [UC](#) [友盟](#) [虾米](#) [阿里星球](#) [钉钉](#)  
[支付宝](#)

© 2009-2018 Aliyun.com 版权所有 ICP证：浙B2-20080101



浙公网安备 33010002000099号