

昵称：南极山
 园龄：10年8个月
 粉丝：34
 关注：5
[+加关注](#)

<	2019年4月						>
日	一	二	三	四	五	六	
31	1	2	3	4	5	6	
7	8	9	10	11	12	13	
14	15	16	17	18	19	20	
21	22	23	24	25	26	27	
28	29	30	1	2	3	4	
5	6	7	8	9	10	11	

搜索

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

我的标签

[java\(39\)](#)
[mysql\(10\)](#)
[spring\(7\)](#)
[tomcat\(7\)](#)
[linux\(5\)](#)

JedisPool资源池优化

转载自：[https://yq.aliyun.com/articles/236383?](https://yq.aliyun.com/articles/236383?spm=a2c4e.11153959.blogcont531067.14.64f445b52u5nWH)
 spm=a2c4e.11153959.blogcont531067.14.64f445b52u5nWH

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：

```

<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(..);
  
```

jvm(4)
mvc(4)
redis(3)
Lucene(3)
算法(3)
更多

随笔分类

Asp.NET(4)
c#.Net(8)
database(1)
Javascript(4)
mvc(1)
mysql(1)
Oracle
Sliverlight(1)
SqlServer(2)
Uml学习(3)
其它(15)
算法(1)

随笔档案

2016年3月 (1)
2015年11月 (2)
2015年6月 (1)
2015年1月 (1)
2014年12月 (1)
2014年11月 (1)
2014年10月 (2)
2014年8月 (1)
2014年7月 (2)
2013年8月 (1)
2012年12月 (1)
2012年11月 (3)
2012年10月 (2)
2012年9月 (1)
2012年8月 (1)
2011年9月 (2)

```
jedisPoolConfig.setMaxIdle(..);  
jedisPoolConfig.setMinIdle(..);  
jedisPoolConfig.setMaxWaitMillis(..);
```

JedisPool的初始化如下:



```
// 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的构造函数
```

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



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。

2010年5月 (2)
2010年4月 (2)
2010年3月 (2)
2010年2月 (1)
2010年1月 (1)
2009年10月 (1)
2009年9月 (2)
2009年5月 (2)

文章分类

android
ankaban(1)
commons-pool(2)
elasticsearch(2)
guice
Java(73)
java虚拟机(7)
jsp(2)
linux(10)
log4j(1)
Lucene(6)
maven(3)
mongodb(1)
mybatis(2)
mysql(14)
netty(1)
nginx(1)
rabbitmq(2)
redis(3)
Shiro(8)
spring(11)
Sql Server(1)
tcp/ip(2)
thrift(1)
tomcat(7)
其它(8)
前端开发(2)
算法(11)

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
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

开发文档

阿里开发文档

在线工具

ImportNew-Java技术
Json在线工具
正则在线测试

积分与排名

积分 - 108056
排名 - 4204

最新评论

1. Re:Linux源码安装步骤
写得真好
--王希知
2. Re:第三方App接入微信登录 解读
谢谢 我是来找这个api地址的
--sdlkfjsdkfj
3. Re:Eclipse常用快捷键
楼主我拿走了哈
--煮酒问寒秋 J
4. Re:JPEG图像压缩
@ostartechts...
--南极山
5. Re:JPEG图像压缩
排版舒服, , 23333
--ostartech

阅读排行榜

1. C#调用windows API的一些方法(48084)
2. 关于jquery跨域请求方法(29178)
3. Javascript 中判断对象为空(23968)

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，也可以使用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JeidsPoolConfig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);
    }
}
```

4. 关于sql函数返回表(7111)
5. app推送中的通知和消息区别(6996)

评论排行榜

1. 计算正整数被标示为连续正整数的和(4)
2. 关于两个杯子到水的问题(3)
3. 也谈一下关于兔子的问题(3)
4. 在若干个整数中找到相加之和为某个整数的所有组合的算法(2)
5. 关于Request.Form获取listbox所有项的问题(2)

推荐排行榜

1. C#调用windows API的一些方法(4)
2. 关于jquery跨域请求方法(3)
3. Asp.net 程序优化js, css合并与压缩(3)
4. sqlserver中DATE类型的数据转化 C ONVERT(1)
5. 关于两个杯子到水的问题(1)

```
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 the pool
...
Caused by: java.util.NoSuchElementException: Timeout waiting for idle object
at org.apache.commons.pool2.impl.GenericObjectPool.borrowObject(GenericObjectPool.java:449)
```

或者

```
redis.clients.jedis.exceptions.JedisConnectionException: Could not get a resource from the pool
...
Caused by: java.util.NoSuchElementException: Pool exhausted
at org.apache.commons.pool2.impl.GenericObjectPool.borrowObject(GenericObjectPool.java:464)
```

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 {
    }
}
```



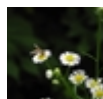
分类: [Java](#), [redis](#), [commons-pool](#)

标签: [redis](#), [JedisPool](#)

好文要顶

关注我

收藏该文



南极山

关注 - 5

粉丝 - 34

+加关注

1

0

« 上一篇: [阿里开发文档](#)

» 下一篇: [CentOS7中源码安装Mariadb](#)

posted @ 2018-03-21 11:47 南极山 阅读(7296) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#)。

【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码

【培训】IT职业生涯指南, Java程序员薪资翻3倍的秘密

【培训】架构师开发十年, 就只剩下这套架构体系了

【推荐】专业便捷的企业级代码托管服务 - Gitee 码云

相关博文:

- Redis JedisPool
- Jedis 连接池JedisPool 解决connection timeout问题
- JedisPool连接池实现难点
- jedispool资源释放
- JedisPool无法获得资源问题

最新新闻:

- 滴滴与自动驾驶产业联盟BDD战略合作 促自动驾驶落地
 - 一线 | 美团打车上线“聚合模式” 首汽、曹操、神州多平台接入
 - 研究发现 DNA 修复能力与长寿相关
 - 任天堂社长: Switch短时间不会入华 E3不发布新硬件
 - 洋葱数学宣布完成D轮3亿元融资 将加大人工智能投入
- » 更多新闻...