

java反射机制性能优化

```
import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;
import java.util.HashMap;
import java.util.Map;
import org.apache.log4j.Logger;

public class DynamicServer {
    private static Logger log = Logger.getLogger(DynamicServer.class);
    public static Map<String,Class> clazzMap = new HashMap<String,Class>();
    public static Map<String,Object> objMap = new HashMap<String,Object>();
    public static Map<String,Method> metMap = new HashMap<String,Method>();
    public static Map<String,Map<String,String>> typeMap = new HashMap<String,Map<String,String>>();

    /**
     * 通过反射+缓存高效的调用某个类里面的某个方法
     * */
    public static Object cacheExce(String clazz,String method,Object[] os,Class[] cs) throws
NoSuchMethodException{
        try {
            int size = 0;
            if(cs!=null){
                size = cs.length;
            }

            Method m = metMap.get(clazz+"_"+method+"_"+size);//用于区分重载的方法
            Object obj = objMap.get(clazz);

            if(m==null||obj==null){
                Class cl = clazzMap.get(clazz);
                if(cl==null){
                    cl = Class.forName(clazz);
                    clazzMap.put(clazz, cl);//缓存class对象
                }

                if(obj==null){
                    obj = cl.newInstance();
                    objMap.put(clazz, obj);//缓存对象的实例
                }

                if(m==null){
                    m = cl.getMethod(method, cs);
                    metMap.put(clazz+"_"+method+"_"+size, m);//缓存Method对象
                }
            }
            return m.invoke(obj , os);//动态调用某个对象中的public声明的方法
        } catch (Exception e) {
            e.printStackTrace();
            throw new NoSuchMethodException();
        }
    }
}
```

昵称：RUN_TIME
园龄：2年6个月
粉丝：14
关注：2

< 2017年10月 >

日	一	二	三	四	五	六
24	25	26	27	28	29	30
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	31	1	2	3	4

搜索

找找看

谷歌搜索

常用链接

- 我的随笔
- 我的评论
- 我的参与
- 最新评论
- 我的标签
- 更多链接

我的标签

- Could Not Connect(1)
- get(1)
- http(1)
- httpClient(1)
- java(1)
- JavaScript(1)
- post(1)
- tomcat负载均衡(1)
- weblogic(1)
- webservice(1)
- 更多

随笔分类

- 工作经历(5)
- 学习笔记(2)
- 有点趣儿

随笔档案

2017年5月 (1)

```
/**
 * 通过反射+缓存获取指定类里面 的指定方法的返回类型
 * */
public static String cacheType(String clazz,String method) throws ClassNotFoundException{
    Map<String,String> clazzs = typeMap.get(clazz);
    if(clazzs==null){
        Map<String,String> mmap = new HashMap<String, String>();
        Class cl = Class.forName(clazz);
        Method[] ms = cl.getMethods();//获取某个类里面的所有的公共的方法
        for(Method m:ms){
            mmap.put(m.getName(), m.getGenericType().toString()); //遍历出所有的方法，将方法名
和返回类型存在静态的map中（缓存）
        }
        clazzs = mmap;
        typeMap.put(clazz, mmap);
    }
    return clazzs.get(method);
}
}
```

代码部分应该没啥要说的了吧，注释已经写的很清楚了，剩下的就需要参照jdk文档了。那么现在来扯扯代码之外的技术话题。

一，九个活动，一人一周

其实是这样的。我的新公司，是给电信做运营支撑的，每到节假日都会做一些送流量送话费的活动。这一次，是针对八月份的里约奥运会来的，给新疆那边做了九个活动，什么奥运游泳，奥运射击等等等、、反正就九个了。

本来吧这跟我是一点关系搭不上，但是后来出了点小意外，，离上线还要一周多点的时间，九个活动大家已经开发的差不多了，突然接到上级的通知开个紧急会议，说活动需要整改，，，说是生产环境为了安全考虑，web应用不可以直接访问数据库，数据库需要放在内网服务器。目前的解决方案就是开发一套内网应用，专门用来做数据库读写操作，然后对外提供接口，供给web应用访问。然后领导接对我说，现在公司大部分人都投入到这期的活动中了，都抽不出时间，有几个新手，也不太放心他们去弄，所以这个艰巨的任务就交给你了，一周能搞定吧？然后我就恩啊恩啊的答应了。

下了班之后，突然觉得有点不对劲！一周！！！九个活动的数据库读写操作！！瞬间崩溃掉了，一周时间还不够我了解九个活动业务上的需求呢。于是赶紧去找领导问问能不能通过其他方式解决，就比如在最外层加个反向代理服务器做个请求转发什么的，这样就不用去改代码了。但是领导的回复是不行。。好吧，只能认栽了。

二，为什么要用反射

第二天过来，向参与活动开发的同事打听了一下情况。一开始是一点思路都没，后来有人提议，让他们提供sql，这样会大大减少我这边的工作量。然后我问了下，发现大部分活动是用jfinal写的，这样的话我就可以直接把他们的Dao层的文件直接拷贝过来了。接下来要做的就是写接口，将方法一一对应起来。于是新的问题来了,每个活动，数据库读写最少有15个方法以上，而且每个方法传的参数都不一样，开发的时候也没有考虑过要整合的问题，所以根本就不存在什么继承，什么多态。这就意味着，我要写接近200个方法与之一一对应，对于我来说，这是写代码以来，最大的灾难！告诉我，这种情况下，除了用反射，我还能怎么办？要我码两百个方法，干脆让我去shi吧。。。

三，性能堪忧

三天之后，算上换行，还有搬砖部分的，累积不超过800行代码结束了这场灾难。接下来的几天，是跟几个开发活动的同事调接口，然后就匆匆上线了。测试的时候感觉还行，但是上线之后，感觉活动打开特别卡。然后我们公司运维查了下服务器性能情况，发现部署后台接口的那台服务器CPU占用率飙到了90%以上。妈的，灾难又来了，肯定是程序出问题了，之前了解过jvm性能监控方面的一些知识，然后百度了一下利用jstack命令打印出堆栈信息分析是那部分代码出的问题。然而，堆栈信息是打印出来了，但是完全看不懂（希望能有个大神指点一下）。没办法最后只好一段代码，一段带代码的测了（感觉应该有更好的方法）。虽然办法有点笨，但最后终于不负所望找到了凶手，原来是反射部分的代码。

四，优化之路

百度了一下，发现在加载类的时候，还有查找和执行方法的时候是很占资源的。想了好一会，觉着可以把这几个比较耗资源的操作缓存起来，这样的话只有在一次调用这个方法的时候比较耗资源，之后除了执行method.invoke方法以外，其他比较好资源的参数，可以直接从缓存中取，应该能缓解一部分压力了吧。光想没用，不如干起来。完了以后测试了一下，执行时间比原来快了接近10倍，而且同一个方法调用次数越多，就越能感觉到差距。

2016年8月 (1)

2016年4月 (1)

2015年7月 (1)

2015年6月 (2)

2015年4月 (3)

最新评论

1. Re:设计模式六大原则
面向对象六大原则

--清香白莲素还真

2. Re:java反射机制性能优化
@GoQC这个的话最好是人为规避这个问题，如果非要代码解决的话，可以再嵌套个map，拿参数的类型数组（Class[] cs）做键。取值的时候通过对比cs的内容来判断。不知道你有没有什么比较好的解决.....

--RUN_TIME

3. Re:java反射机制性能优化
mmap.put(m.getName(), m.getGenericReturnType().toString()); //遍历出所有的方法，将方法名和返回类型存在静态的map中（缓存）重载的方法怎么办.....

--GoQC

4. Re:java反射机制性能优化
Method m = metMap.get(clazz+"_"+method+"_"+size); //用于区分重载的方法如果重载的方法参数个数一样只是类型不同呢。。。...

--GoQC

5. Re:java反射机制性能优化
依赖注入 啊

--查克拉的觉醒

阅读排行榜

- 一份关于jvm内存调优及原理的学习笔记(11093)
- js调用百度地图接口(9717)
- java反射机制性能优化(5070)
- WebLogic部署集群和代理服务器(4361)
- Apache+Tomcat部署负载均衡（或集群）(3952)

评论排行榜

- js调用百度地图接口(6)
- java反射机制性能优化(4)
- 浅谈http请求数据分析(4)
- WebLogic部署集群和代理服务器(3)
- Apache+Tomcat部署负载均衡（或集群）(2)

现在项目已经上线了有一段时间了，基本没我啥事了。抽个空分享下那段时间的工作心得，顺便提醒一句，在并发量大的情况下，能不用反射尽量不用！还有，谁告诉我，这种情况下，除了用反射，我还能怎么办？？？

分类: [工作经历](#)

标签: [java](#), [反射](#)

好文要顶

关注我

收藏该文

[RUN_TIME](#)
[关注 - 2](#)
[粉丝 - 14](#)
[+加关注](#)

2

推荐

0

反对

« 上一篇: [一份关于jvm内存调优及原理的学习笔记](#)
» 下一篇: [设计模式六大原则](#)

posted @ 2016-08-17 23:03 RUN_TIME 阅读(5070) 评论(4) 编辑 收藏

发表评论

- #1楼 2016-08-18 11:04 | [查克拉的觉醒](#)

“
依赖注入 啊

支持(0) 反对(0)
- #2楼 2016-12-08 11:23 | [GoQC](#)

“
Method m = metMap.get(clazz+"_"+method+"_"+size);//用于区分重载的方法
如果重载的方法参数个数一样只是类型不同呢。。。

支持(1) 反对(0)
- #3楼 2016-12-08 11:29 | [GoQC](#)

“
mmap.put(m.getName(), m.getGenericReturnType().toString());//遍历出所有的方法，将方法名和返回类型存在静态的map中（缓存）
重载的方法怎么办？。。。

支持(0) 反对(0)
- #4楼[楼主] 2016-12-12 23:33 | [RUN_TIME](#)

“
@ GoQC
这个的话最好是人为规避这个问题，如果非要代码解决的话，可以再嵌套个map，拿参数的类型数组（Class[] cs）做键。取值的时候通过对比cs的内容来判断。不知道你有没有什么比较好的解决方案呢

支持(0) 反对(0)

[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】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
- 【推荐】报表开发有捷径：快速设计轻松集成，数据可视化和交互
- 【推荐】腾讯云 普惠云计算 0门槛体验

HUAWEI

软件开发云

移动应用测试有奖公测

抢先体验

- 最新IT新闻:
- 是我们改变了世界，还是世界改变了我和你
 - 自然语言处理技术，将会使机器从更人性化的视角来解决问题
 - 钛媒体独家曝光贾跃亭在美国的“金蝉脱壳”计划：FF破产卖掉
 - 高瓴资本·张磊：投资人要用长远的眼光看问题做选择，懂得滞后满足
 - 解构趣店：未杜绝校园贷、实际利率高于协议利率
- » 更多新闻...

推荐排行榜

1. [Apache+Tomcat部署负载均衡（或集群）](#) (6)
2. [WebLogic部署集群和代理服务器](#)(5)
3. [java反射机制性能优化](#)(2)
4. [同台电脑部署多组Tomcat负载均衡（或集群）](#) (2)
5. [js调用百度地图接口](#)(1)

Copyright ©2017 RUN_TIME



最新知识库文章:

- 实用VPC虚拟私有云设计原则
- 如何阅读计算机科学类的书
- Google 及其云智慧
- 做到这一点，你也可以成为优秀的程序员
- 写给立志做码农的大学生
- » 更多知识库文章...