

Dubbo源码解析（二十八）远程调用——memcached协议

[memcached](#) [dubbo](#) 阅读约 12 分钟

远程调用——memcached协议

目标：介绍memcached协议的设计和实现，介绍dubbo-rpc-memcached的源码。

前言

dubbo实现memcached协议是基于Memcached，[Memcached](#) 是一个高效的 KV 缓存服务器，在dubbo中没有涉及到关于memcached协议的服务暴露，只有服务引用，因为在访问Memcached服务器时，Memcached客户端可以在服务器上存储也可以获取。

源码分析

(一) MemcachedProtocol

该类继承AbstractProtocol，是memcached协议实现的核心。

1. 属性

```
/**  
 * 默认端口号  
 */  
public static final int DEFAULT_PORT = 11211;
```

2. export

```
@Override  
public <T> Exporter<T> export(final Invoker<T> invoker) throws RpcException {  
    // 不支持memcached服务暴露  
    throw new UnsupportedOperationException("Unsupported export memcached service. url: " + invoker.getUrl());  
}
```

可以看到，服务暴露方法直接抛出异常。

3. refer

```

@Override
public <T> Invoker<T> refer(final Class<T> type, final URL url) throws RpcException {
    try {
        // 获得地址
        String address = url.getAddress();
        // 获得备用地址
        String backup = url.getParameter(Constants.BACKUP_KEY);
        // 把备用地址拼接上
        if (backup != null && backup.length() > 0) {
            address += "," + backup;
        }
        // 创建Memcached客户端构造器
        MemcachedClientBuilder builder = new XMemcachedClientBuilder(AddrUtil.getAddresses(address));
        // 创建客户端
        final MemcachedClient memcachedClient = builder.build();
        // 到期时间参数配置
        final int expiry = url.getParameter("expiry", 0);
        // 获得值命令
        final String get = url.getParameter("get", "get");
        // 添加值命令根据类型来取决是put还是set
        final String set = url.getParameter("set", Map.class.equals(type) ? "put" : "set");
        // 删除值命令
        final String delete = url.getParameter("delete", Map.class.equals(type) ? "remove" : "delete");
        return new AbstractInvoker<T>(type, url) {
            @Override
            protected Result doInvoke(Invocation invocation) throws Throwable {

```

该方法是服务引用方法，基于MemcachedClient的get、set、delete方法来对应Memcached的get、set、delete命令进行对值的操作。

后记

该部分相关的源码解析地址：<https://github.com/CrazyHZM/i...>

该文章讲解了远程调用中关于memcached协议实现的部分，逻辑比较简单。接下来我将开始对rpc模块关于redis协议部分进行讲解。

阅读 501 · 更新于 11月8日

 赞 1  收藏 1  赞赏  分享

本作品系原创，作者保留所有权利，未经作者允许，禁止转载和演绎



crazyhzm

◆ 265 

关注作者

0 条评论

得票 · 时间



撰写评论 ...

提交评论

推荐阅读

dubbo负载均衡策略及对应源码分析

在集群负载均衡时，Dubbo提供了多种均衡策略，缺省为random随机调用。我们还可以扩展自己的负责均衡策略，前提是已经...

bali · 阅读 757