# W-IO文档

版本号 1.1

w-io是一款集http解析，分布式集群注册，请求转发，接口调用，负载均衡，分布式锁，分布式事务于一身的网络框架，是基于高性能的netty4和十倍于json传输效率的protobuf3开发的一款网络框架，采用多路复用的设计思想，所有请求转发和接口调用，都是由服务器到服务器之间的一个连接(channel)进行数据传输，以达到多路复用的目的，所有业务处理都由业务线程池进行线程分配，为了不阻塞网络IO。目前，在安装了win7的虚拟机上压测，四代i7分配5个核心，8G内存，map给1000个Key，一万并发下，平均返回时间3秒左右，非压测情况下不超过100毫秒。**本套代码完全开源，将会一直坚持完全开源，拒绝独享的原则**

## w-io技术介绍

1. http请求解析：不再采用servlet规范，业务代码中不需要区分是get还是post请求，因为handle中已经处理好了，http请求地址将会转发到各个业务服务器中，而页面文件放到入口，由返回的字符串判断请求哪个html，而js，css等文件则只需要前端请求即可。
2. 注册中心：注册中心放在入口处，统一对注册的服务器进行管理，并保持和业务服务器的连接，避免重新创建连接带来的性能损耗。由controller注册的路径进行判断，服务器在注册的时候会带一些信息，比如路径，负载能力，是否有锁等，http请求或者内部服务器进行接口调用都会通过注册中心统一调配，判断是否有锁。当所有服务器都被认为满载，则返回503。**不要配置相同的路径名，否则会出问题**
3. 分布式锁：注册中心调配请求的时候，如果判断路径上有锁，则这个路径只能被一个请求所持有，当这个请求返回时，锁会被释放，如果出现问题导致不能返回，由定时器释放，前提是配置了超时策略
4. 服务器之间接口调用：和http一样由注册中心统一调配，基本上和http请求没啥区别，但接口调用分有同步接口调用和异步接口调用，同步接口调用会有返回值，目前是一个map，异步接口调用没有返回值，无需等待。
5. 分布式事物：分布式事务在事务调度器上，每个业务服务器都需要连接到事务调度器。目前事务的提交由事务调度器控制事务的提交，第一次确认事务，在业务方法执行之前判断方法是否启用事务，方法内由thransThread进行提交，当方法返回上一个方法之后再进行阻塞等待最终确认，如果最终确认需要回滚，则需要执行补偿程序，补偿程序的定义还没设计好，需要下一个版本才能解决。

## 主要操作

注册中心：proxyHs项目中的HttpServer进行入口的启动，其配置文件基本都有中文注释，下一版本将完善所有配置文件。业务服务器是RegServer1由OtherServer类进行启动和注册，可以自己改名，配置自己的端口和ip以及线程池。事务调度器是Transer由TransServer进行启动。启动流程：注册中心—事务调度器—业务服务器。Controller的配置在RefController的方法controllerSetting，从这里配置的controller才能到注册中心进行注册。

目前是采用java7+myeclipse10进行开发，代码拷贝下来后，在myeclipse上建立三个项目，一个是注册中心，一个是事务调度器，其他的是业务服务器。

## 其他说明

本套程序坚持完全开源，拒绝独享。坚持简单易学，易上手，易配置。

## 下一个版本

版本1.1将提供完整配置，事务补偿设计，更多的优化，敬请期待