

06-Dubbo最佳实践

1 推荐用法

<http://dubbo.apache.org/zh-cn/docs/user/configuration/configuration-load-process.html>

2 服务化最佳实践

<http://dubbo.apache.org/zh-cn/docs/user/best-practice.html>

3 容量规划

以下数据供参考：

使用 Dubbo 的会员服务项目

- 每天接收 4 亿次远程调用
- 使用 12 台网站标配机器提供服务（8 核 CPU，8G 内存）
- 平均负载在 1 以下（对于 8 核 CPU 负载很低）
- 平均响应时间 2.3 到 2.5 毫秒，网络开销约占 1.5 到 1.6 毫秒（和数据包大小有关）

使用 Dubbo 的产品授权服务项目

- 每天接收 3 亿次远程调用
- 使用 8 台网站标配机器提供服务（8 核 CPU，8G 内存）
- 平均负载在 1 以下（对于 8 核 CPU 负载很低）

- 平均响应时间 1.4 到 2.8 毫秒，网络开销约占 1.0 到 1.1 毫秒（和数据包大小有关）

4 优雅停机

Dubbo 是通过 JDK 的 ShutdownHook 来完成优雅停机的，所以如果用户使用 `kill -9 PID` 等强制关闭指令，是不会执行优雅停机的，只有通过 `kill PID` 时，才会执行。

原理

服务提供方

- 停止时，先标记为不接收新请求，新请求过来时直接报错，让客户端重试其它机器。
- 然后，检测线程池中的线程是否正在运行，如果有，等待所有线程执行完成，除非超时，则强制关闭。

服务消费方

- 停止时，不再发起新的调用请求，所有新的调用在客户端即报错。
- 然后，检测有没有请求的响应还没有返回，等待响应返回，除非超时，则强制关闭。

设置方式

设置优雅停机超时时间，缺省超时时间是 10 秒，如果超时则强制关闭。

```
# dubbo.properties
dubbo.service.shutdown.wait=15000
```

如果 ShutdownHook 不能生效，可以自行调用，使用tomcat等容器部署的场景，建议通过扩展ContextListener等自行调用以下代码实现优雅停机：

```
ProtocolConfig.destroyAll();
```