

Dubbo 视频教程--高级篇--第 30 节--Dubbo 负载均衡策略

Dubbo 官方文档：

用户指南 >> 示例 >> 负载均衡

负载均衡

(+) (#)

✓ 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。

可以自行扩展负载均衡策略，参见：[负载均衡扩展](#)

Random LoadBalance

- 随机，按权重设置随机概率。
- 在一个截面上碰撞的概率高，但调用量越大分布越均匀，而且按概率使用权重后也比较均匀，有利于动态调整提供者权重。

RoundRobin LoadBalance

- 轮循，按公约后的权重设置轮循比率。
- 存在慢的提供者累积请求问题，比如：第二台机器很慢，但没挂，当请求调到第二台时就卡在那，久而久之，所有请求都卡在调到第二台上。

LeastActive LoadBalance

- 最少活跃调用数，相同活跃数的随机，活跃数指调用前后计数差。
- 使慢的提供者收到更少请求，因为越慢的提供者的调用前后计数差会越大。

ConsistentHash LoadBalance

- 一致性Hash，相同参数的请求总是发到同一提供者。
- 当某一台提供者挂时，原本发往该提供者的请求，基于虚拟节点，平摊到其它提供者，不会引起剧烈变动。
- 算法参见：http://en.wikipedia.org/wiki/Consistent_hashing。
- 缺省只对第一个参数Hash，如果要修改，请配置<dubbo:parameter key="hash.arguments" value="0,1" />
- 缺省用160份虚拟节点，如果要修改，请配置<dubbo:parameter key="hash.nodes" value="320" />

配置如：

```
<dubbo:service interface="..." loadbalance="roundrobin" />
```

或：

```
<dubbo:reference interface="..." loadbalance="roundrobin" />
```

或：

```
<dubbo:service interface="...">
  <dubbo:method name="..." loadbalance="roundrobin"/>
</dubbo:service>
```

或：

```
<dubbo:reference interface="...">
  <dubbo:method name="..." loadbalance="roundrobin"/>
</dubbo:reference>
```



龙果学院微信公众号：ron-coo

负载均衡扩展

(+) (#)

(1) 扩展说明:

从多个服务提供者中选择一个进行调用。

(2) 扩展接口:

```
com.alibaba.dubbo.rpc.cluster.LoadBalance
```

(3) 扩展配置:

```
<dubbo:protocol loadbalance="xxx" />
<dubbo:provider loadbalance="xxx" /> <!-- 缺省值设置, 当<dubbo:protocol>没有配置loadbalance时, 使用此配置 -->
```

(4) 已知扩展:

```
com.alibaba.dubbo.rpc.cluster.loadbalance.RandomLoadBalance
com.alibaba.dubbo.rpc.cluster.loadbalance.RoundRobinLoadBalance
com.alibaba.dubbo.rpc.cluster.loadbalance.LeastActiveLoadBalance
```

(5) 扩展示例:

Maven项目结构

```
src
|-main
|  |-java
|     |-com
|         |-xxx
|             |-XxxLoadBalance.java (实现LoadBalance接口)
|  |-resources
|     |-META-INF
|         |-dubbo
|             |-com.alibaba.dubbo.rpc.cluster.LoadBalance (纯文本文件, 内容为: xxx=com.xxx.XxxLoadBalance)
```

XxxLoadBalance.java

```
package com.xxx;

import com.alibaba.dubbo.rpc.cluster.LoadBalance;
import com.alibaba.dubbo.rpc.Invoker;
import com.alibaba.dubbo.rpc.Invocation;
import com.alibaba.dubbo.rpc.RpcException;

public class XxxLoadBalance implements LoadBalance {
    public <T> Invoker<T> select(List<Invoker<T>> invokers, Invocation invocation) throws RpcException {
        // ...
    }
}
```

META-INF/dubbo/com.alibaba.dubbo.rpc.cluster.LoadBalance

```
xxx=com.xxx.XxxLoadBalance
```

