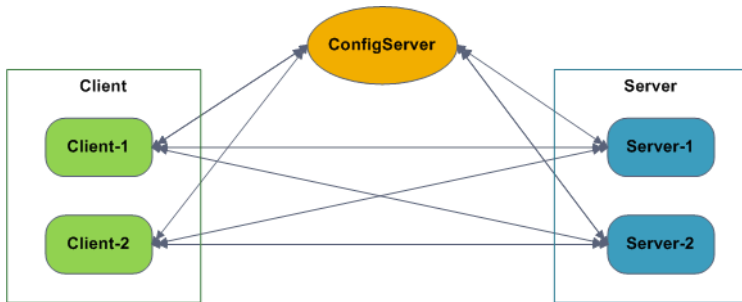


分布式框架原理

alibaba有好几个分布式框架，主要有：进行远程调用(类似于RMI的这种远程调用)的(dubbo、hsf)，jms消息服务(napoli、notify)，KV数据库(tair)等。这个框架/工具/产品在实现的时候，都考虑到了容灾，扩展，负载均衡，于是出现一个配置中心(ConfigServer)的东西来解决这些问题。

基本原理如图：



在我们的系统中，经常会有一些跨系统的调用，如在A系统中要调用B系统的一个服务，我们可能会使用RMI直接来进行，B系统发布一个RMI接口服务，然后A系统就来通过RMI调用这个接口，为了解决容灾，扩展，负载均衡的问题，我们可能会想很多办法，alibaba的这个办法感觉不错。

本文只说dubbo，原理如下：

• ConfigServer

配置中心，和每个Server/Client之间会作一个实时的心跳检测（因为它们都是建立的Socket长连接），比如几秒钟检测一次。收集每个Server提供的服务的信息，每个Client的信息，整理出一个服务列表，如：

service Name	serverAddressList	clientAddressList
UserService	192.168.0.1, 192.168.0.2, 192.168.0.3, 192.168.0.4	172.16.0.1, 172.16.0.2
ProductService	192.168.0.3, 192.168.0.4, 192.168.0.5, 192.168.0.6	172.16.0.2, 172.16.0.3
OrderService	192.168.0.10, 192.168.0.12, 192.168.0.5, 192.168.0.6	172.16.0.3, 172.16.0.4

当某个Server不可用，那么就更新受影响的服务对应的serverAddressList，即把这个Server从serverAddressList中踢出去（从地址列表中删除），同时将推送serverAddressList给这些受影响的服务的clientAddressList里面的所有Client。如：192.168.0.3挂了，那么UserService和ProductService的serverAddressList都要把192.168.0.3删除掉，同时把新的列表告诉对应的Client 172.16.0.1, 172.16.0.2, 172.16.0.3；

当某个Client挂了，那么更新受影响的服务对应的clientAddressList

ConfigServer根据服务列表，就能提供一个web管理界面，来查看管理服务的提供者和使用者的。

新加一个Server时，由于它会主动与ConfigServer取得联系，而ConfigServer又会将这个信息主动发送给Client，所以新加一个Server时，只需要启动Server，然后几秒钟内，Client就会使用上它提供的服务

• Client

调用服务的机器，每个Client启动时，主动与ConfigServer建立Socket长连接，并将自己的IP等相应信息发送给ConfigServer。

Client在使用服务的时候根据服务名称去ConfigServer中获取服务提供者信息（这样ConfigServer就知道某个服务是当前哪几个Client在使用），Client拿到这些服务提供者信息后，与它们都建立连接，后面就可以直接调用服务了，当有多个服务提供者时，Client根据一定的规则来进行负载均衡，如轮询，随机，按权重等。

一旦Client使用的服务它对应的服务提供者有变化（服务提供者有新增，删除的情况），ConfigServer就会把最新的服务提供者列表推送给Client，Client就会依据最新的服务提供者列表重新建立连接，新增的提供者建立连接，删除的提供者丢弃连接

• Server

真正提供服务的机器，每个Server启动时，主动与ConfigServer建立Socket长连接，并将自己的IP，提供的服务名称，端口等信息直接发送给

ConfigServer，ConfigServer就会收集到每个Server提供的服务的信息。

优点：

- 1，只要在Client和Server启动的时候，ConfigServer是好的，服务就可调用了，如果后面ConfigServer挂了，那只影响ConfigServer挂了以后服务提供者有变化，而Client还无法感知这一变化。
- 2，Client每次调用服务是不经过ConfigServer的，Client只是与它建立联系，从它那里获取提供服务者列表而已
- 3，调用服务-负载均衡：Client调用服务时，可以根据规则在多个服务提供者之间轮流调用服务。
- 4，服务提供者-容灾：某一个Server挂了，Client依然是可以正确的调用服务的，当前提是这个服务有至少2个服务提供者，Client能很快的感知到服务提供者的变化，并作出相应反应。
- 5，服务提供者-扩展：添加一个服务提供者很容易，而且Client会很快的感知到它的存在并使用它。

顺便说一下，**hadoop**里面的中心节点跟这里的**configServer**作用类似，在维护节点列表方面，不过它的相关计算都需要通过中心节点，让它来分配任务。