ZooKeeper实现负载均衡

|  |  |  |  |
| --- | --- | --- | --- |
| 编写人 | 编写时间 | 版本 | 备注 |
| 张贵东 | 2017年8月 | V1.0 |  |

目 录

[1. 负载均衡 2](#_Toc490550939)

[1.1. 负载均衡介绍 2](#_Toc490550940)

[1.1.1. 详细的流程分析 2](#_Toc490550941)

[1.1.2. 负载均衡算法 4](#_Toc490550942)

[1.2. 实现思路 5](#_Toc490550943)

[1.3. 代码实现 8](#_Toc490550944)

[1.4. 存在问题： 8](#_Toc490550945)

[2. 同步队列 8](#_Toc490550946)

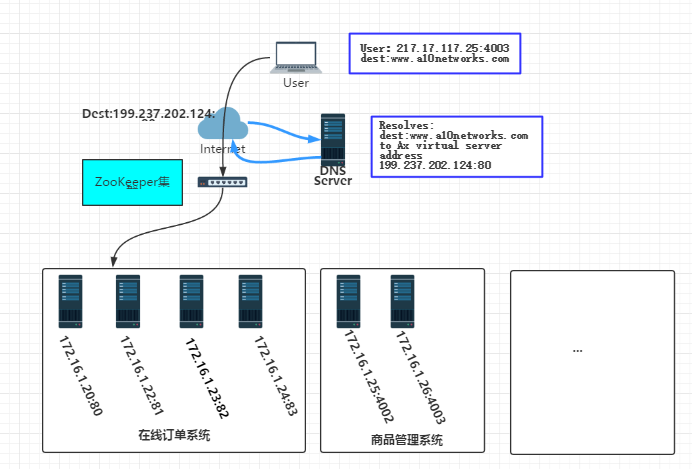
1. 负载均衡
   1. 负载均衡介绍

负载均衡设备作为纵跨网络2-7层协议的设备，往往放置在网络设备和应用设备的连接处，对工程师在网络和应用基本知识方面的要求远高于其他设备，所以我们要在基本功能的理解上下更多的功夫。负载均衡设备还有另外一个称呼：4/7层交换机，但它首先是个2-3层交换机，这要求我们首先掌握2-3层的基本知识，然后才是本文介绍的内容。

服务器负载均衡有三大基本Feature：负载均衡算法，健康检查和会话保持，这三个Feature是保证负载均衡正常工作的基本要素。其他一些功能都是在这三个功能之上的一些深化。下面我们具体介绍一下各个功能的作用和原理。负载均衡设备作为纵跨网络2-7层协议的设备，往往放置在网络设备和应用设备的连接处，对工程师在网络和应用基本知识方面的要求远高于其他设备，所以我们要在基本功能的理解上下更多的功夫。负载均衡设备还有另外一个称呼：4/7层交换机，但它首先是个2-3层交换机，这要求我们首先掌握2-3层的基本知识，然后才是本文介绍的内容。

详细的流程分析

我们来做一个详细的访问流程分析:



用户(IP:207.17.117.20)访问域名www.a10networks.com，首先会通过DNS查询解析出这个域名的公网地址：199.237.202.124，接下来用户207.17.117.20会访问199.237.202.124这个地址，因此数据包会到达负载均衡设备，接下来负载均衡设备会把数据包分发到合适的服务器，看下图：

负载均衡设备在将数据包发给服务器时，数据包是做了一些变化的，如上图所示，数据包到达负载均衡设备之前，源地址是：207.17.117.20，目的地址是：199.237.202.124, 当负载均衡设备将数据包转发给选中的服务器时，源地址还是：207.17.117.20,目的地址变为172.16.20.1，我们称这种方式为目的地址NAT(DNAT)。一般来说，在服务器负载均衡中DNAT是一定要做的（还有另一种模式叫做服务器直接返回-DSR，是不做DNAT的，我们将另行讨论），而源地址根据部署模式的不同，有时候也需要转换成别的地址，我们称之为：源地址NAT(SNAT)，一般来说，旁路模式需要做SNAT，而串接模式不需要，本示意图为串接模式，所以源地址没做NAT。

我们再看服务器的返回包，如下图所示，也经过了IP地址的转换过程，不过应答包中源/目的地址与请求包正好对调，从服务器回来的包源地址为172.16.20.1，目的地址为207.17.117.20,到达负载均衡设备后，负载均衡设备将源地址改为199.237.202.124，然后转发给用户，保证了访问的一致性。

以上是单个数据包的处理流程。那么负载均衡设备是怎么选择服务器的呢？ 这就是我们要介绍的第一个Feature:

负载均衡算法

一般来说负载均衡设备都会默认支持多种负载均衡分发策略，例如：

Ø 轮询（RoundRobin）将请求顺序循环地发到每个服务器。当其中某个服务器发生故障，AX就把其从顺序循环队列中拿出，不参加下一次的轮询，直到其恢复正常。

Ø 比率（Ratio）：给每个服务器分配一个加权值为比例，根椐这个比例，把用户的请求分配到每个服务器。当其中某个服务器发生故障，AX就把其从服务器队列中拿出，不参加下一次的用户请求的分配，直到其恢复正常。

Ø 优先权（Priority）：给所有服务器分组，给每个组定义优先权，将用户的请求分配给优先级最高的服务器组（在同一组内，采用预先设定的轮询或比率算法，分配用户的请求）；当最高优先级中所有服务器或者指定数量的服务器出现故障，AX将把请求送给次优先级的服务器组。这种方式，实际为用户提供一种热备份的方式。

Ø 最少连接数（LeastConnection）：AX会记录当前每台服务器或者服务端口上的连接数，新的连接将传递给连接数最少的服务器。当其中某个服务器发生故障，AX就把其从服务器队列中拿出，不参加下一次的用户请求的分配，直到其恢复正常。

Ø 最快响应时间（Fast Reponse time）：新的连接传递给那些响应最快的服务器。当其中某个服务器发生故障，AX就把其从服务器队列中拿出，不参加下一次的用户请求的分配，直到其恢复正常。

以上为通用的负载均衡算法，还有一些算法根据不同的需求也可能会用到，例如：

Ø 哈希算法( hash): 将客户端的源地址，端口进行哈希运算，根据运算的结果转发给一台服务器进行处理，当其中某个服务器发生故障，就把其从服务器队列中拿出，不参加下一次的用户请求的分配，直到其恢复正常。

Ø 基于策略的负载均衡：针对不同的数据流设置导向规则，用户可自行编辑流量分配策略，利用这些策略对通过的数据流实施导向控制。

Ø 基于数据包的内容分发：例如判断HTTP的URL，如果URL中带有.jpg的扩展名，就把数据包转发到指定的服务器。

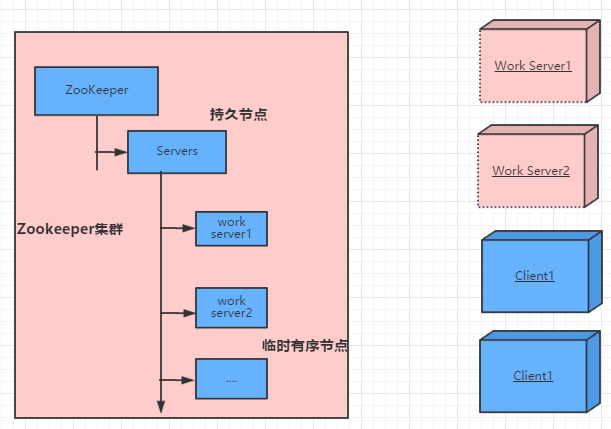
继续看图分析，第二个用户207.17.117.21也访问www.a10networks.com，负载均衡设备根据负载均衡算法将第二个用户的请求转发到第二台服务器来处理。

* 1. 实现思路

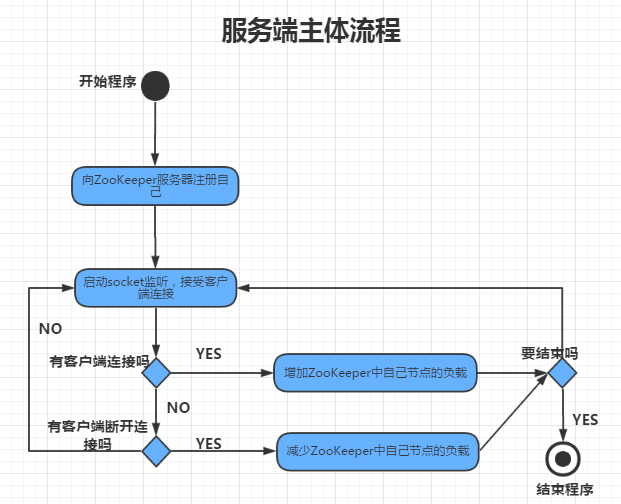
每台WorkServer启动的时候都会到Server创建临时节点。

每台ClientServer启动的时候，都会到Server节点下面取得所有WorksServer节点，并通过一定算法取得一台并与之连接。

zookeeper实现原理图：

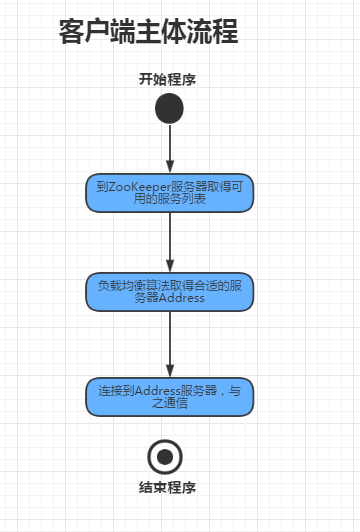


zookeeper实现流程图：

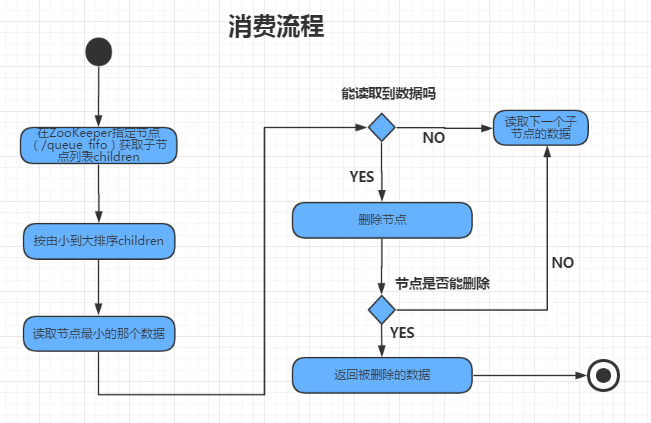


服务端主体流程

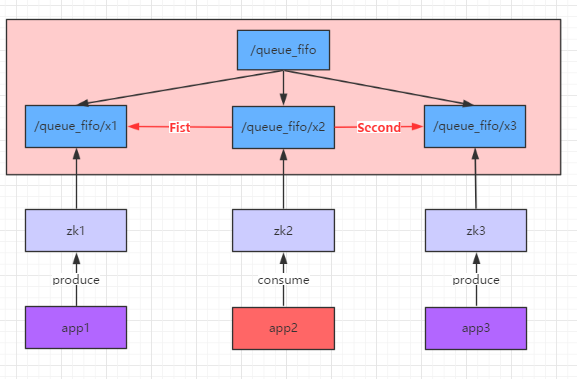
有ClientServer与之建立连接，这台WorksServer的负载计数器加一，断开连接负载计数器减一。负载计数器作为客户端负载均衡算法的依据，客户端会选择负载最轻的WorksServer建立连接。



客户端主体流程

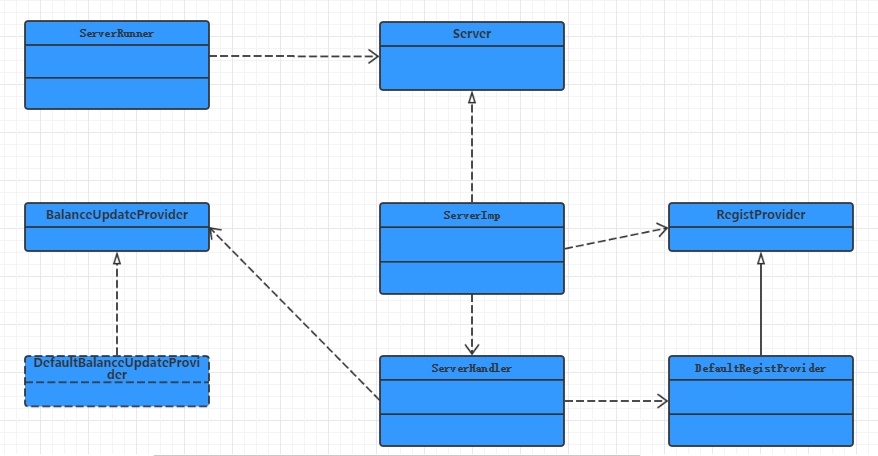


实现实例：



* 1. 代码实现

服务端代码设计：

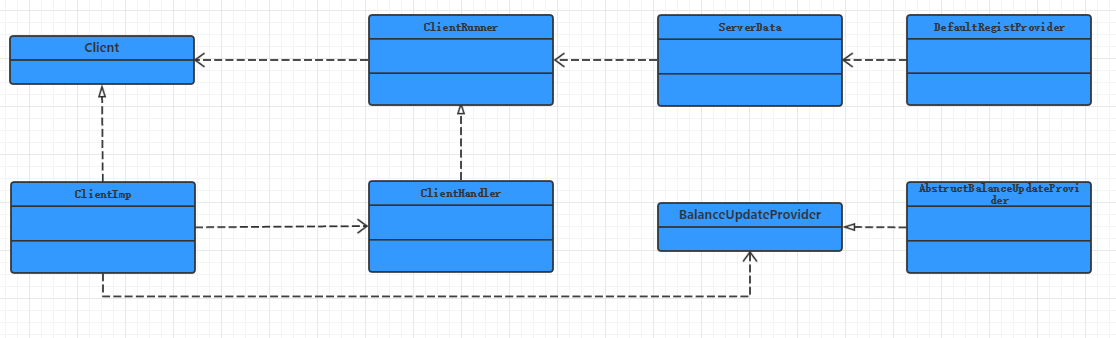


ServerRunner 调度类

RegistProvider 服务端启动时的注册过程

ServerHander 处理与客户端之间的连接

DefaultBalanceUpdateProvider 连接建立与断开，修改负载信息



  ClientRunner 调度类

  ClientHander 处理与服务器之间的通信

  BanceProvider 负载的算法

  ServerData 服务器和客户端公用的类，计算负载等使用

见GitHub地址: <https://git.oschina.net/BJZGD/zookeeper-demo>

* 1. 存在问题：