为什么不要把ZooKeeper用于服务发现

作者 **谢丽** 发布于 2014年12月25日 | 讨论

ZooKeeper是Apache基金会下的一个开源的、高可用的分布式应用协调服务。许多公司都把它用于服务发现。但在云环境中，面对设备及网络故障时的恢复能力是需要重点考虑的问题。因此，将应用部署在云上，就必须要预见到硬件故障、网络延迟以及网络分区等问题，进而构建出恢复能力强的系统。Peter Kelley是个性化教育初创公司Knewton的一名软件工程师。他认为，从根本上讲，把ZooKeeper用于服务发现是个错误的做法，理由如下：

在ZooKeeper中，网络分区中的客户端节点无法到达Quorum时，就会与ZooKeeper失去联系，从而也就无法使用其服务发现机制。因此，在用于服务发现时，ZooKeeper无法很好地处理网络分区问题。作为一个协调服务，这没问题。但对于服务发现来说，信息中可能包含错误要好于没有信息。虽然可以通过客户端缓存和其它技术弥补这种缺陷，像Pinterest和Airbnb等公司所做的那样，但这并不能从根本上解决问题，如果Quorum完全不可用，或者集群分区和客户端都恰好连接到了不属于这个Quorum但仍然健康的节点，那么客户端状态仍将丢失。

更重要地，上述做法的本质是试图用缓存提高一个一致性系统的可用性，即在一个CP系统之上构建AP系统，这根本就是错误的方法。服务发现系统从设计之初就应该针对可用性而设计。

抛开CAP理论不说，ZooKeeper的设置和维护非常困难，以致Knewton多次因为错误的使用出现问题。一些看似很简单的事情，实际操作起来也非常容易出错，如在客户端重建Watcher，处理Session和异常。另外，ZooKeeper本身确实也存在一些问题，如ZOOKEEPER-1159、ZOOKEEPER-1576。

由于这些问题的存在，他们切换到了Eureka。这是一个由Netflix开发的、开源的服务发现解决方案，具有可用性高、恢复能力强的特点。相比之下，它有如下优点：

如果一个服务器出现问题，Eureka不需要任何类型的选举，客户端会自动切换并连接到一个新的Eureka服务器。当它恢复时，可以自动加入Eureka节点集群。而且，按照设计，它可以在零停机的情况下处理更广泛的网络分区问题。在出现网络分区的情况下，Eureka将继续接受新的注册并发布。这可以确保新增服务仍然可以供分区同侧的任意客户端使用。

Eureka有一个服务心跳的概念，可以阻止过期数据：如果一个服务长时间没有发送心跳，那么Eureka将从服务注册中将其删除。但在出现网络分区、Eureka在短时间内丢失过多客户端时，它会停用这一机制，进入“自我保护模式”。网络恢复后，它又会自动退出该模式。这样，虽然它保留的数据中可能存在错误，却不会丢失任何有效数据。

Eureka在客户端会有缓存。即使所有Eureka服务器不可用，服务注册信息也不会丢失。缓存在这里是恰当的，因为它只在所有的Eureka服务器都没响应的情况下才会用到。

Eureka就是为服务发现而构建的。它提供了一个客户端库，该库提供了服务心跳、服务健康检查、自动发布及

缓存刷新等功能。使用ZooKeeper，这些功能都需要自己实现。

管理简单，很容易添加和删除节点。它还提供了一个清晰简洁的网页，上面列出了所有的服务及其健康状况。

Eureka还提供了REST API，使用户可以将其集成到其它可能的用途和查询机制。

总之，云平台并不总是可靠，服务发现需要具备尽可能高的可用性和恢复能力，而Eureka恰恰是针对这种情况而设计的。

感谢郭蕾对本文的审校。
