负载均衡算法

在分布式系统中，多台服务器同时提供一个服务，并统一到服务配置中心进行管理，消费者通过查询服务配置中心，获取到服务到地址列表，需要选取其中一台来发起RPC远程调用。如何选择，则取决于具体的负载均衡算法，对应于不同的场景，选择的负载均衡算法也不尽相同。负载均衡算法的种类有很多种，常见的负载均衡算法包括轮询法、随机法、源地址哈希法、加权轮询法、加权随机法、最小连接法、Latency-Aware等，应根据具体的使用场景选取对应的算法

1、轮询（Round Robin）法

轮询很容易实现，将请求按顺序轮流分配到后台服务器上，均衡的对待每一台服务器，而不关心服务器实际的连接数和当前的系统负载。轮询法适用于机器性能相同的服务，一旦某台机器性能不好，极有可能产生木桶效应，性能差的机器扛不住更多的流量。

2、源地址哈希法

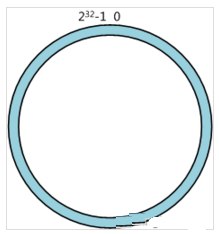
源地址哈希法的思想是根据服务消费者请求客户端的IP地址，通过哈希函数计算得到一个哈希值，将此哈希值和服务器列表的大小进行取模运算，得到的结果便是要访问的服务器地址的序号。

3、加权轮询（Weight Round Robin）法

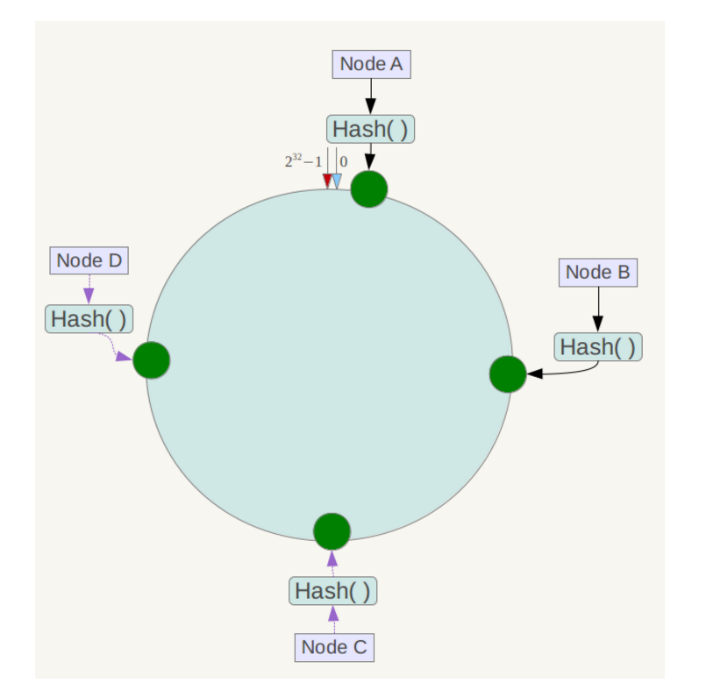
不同的后台服务器可能机器的配置和当前系统的负载并不相同，因此它们的抗压能力也不一样。跟配置高、负载低的机器分配更高的权重，使其能处理更多的请求，而配置低、负载高的机器，则给其分配较低的权重，降低其系统负载，加权轮询很好的处理了这一问题，并将请求按照顺序且根据权重分配给后端。Nginx的负载均衡默认算法是加权轮询算法。

4、**一致性哈希算法**

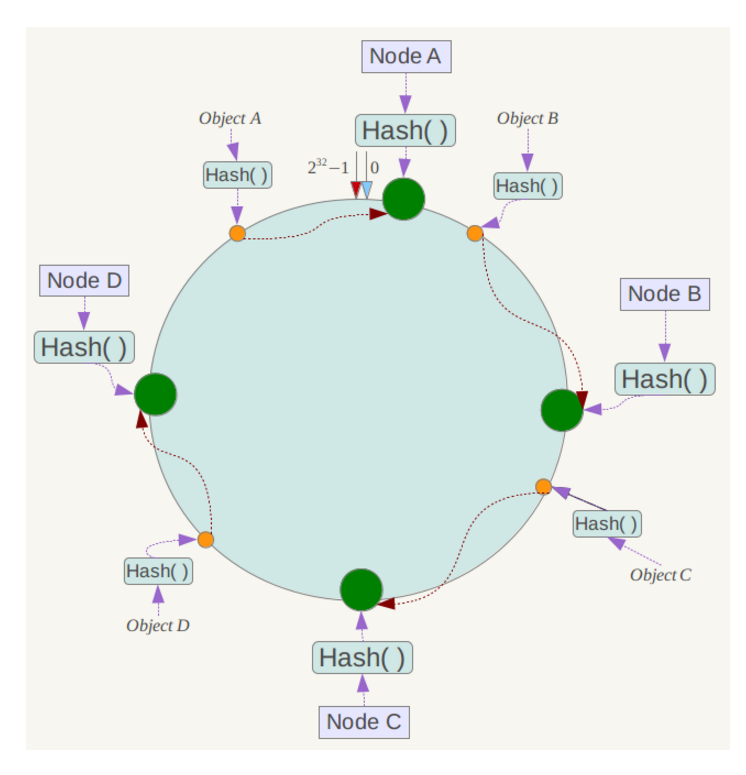
一致性哈希将整个哈希值空间组织成一个虚拟的圆环，如假设某哈希函数H的值空间为0 - (2^32)-1（即哈希值是一个32位无符号整形），整个哈希空间环如下：



下一步将各个服务器使用Hash进行一个哈希，具体可以选择服务器的ip或主机名作为关键字进行哈希，这样每台机器就能确定其在哈希环上的位置，这里假设将上文中四台服务器使用ip地址哈希后在环空间的位置如下：



将数据key使用相同的函数Hash计算出哈希值，并确定此数据在环上的位置，从此位置沿环顺时针“行走”，第一台遇到的服务器就是其应该定位到的服务器。



一般的，在一致性哈希算法中，如果一台服务器不可用，则受影响的数据仅仅是此服务器到其环空间中前一台服务器（即沿着逆时针方向行走遇到的第一台服务器）之间数据，其它不会受到影响。

分布式数据库系统

数据分配方式

（1）集中式：所有数据片段都安排在同一个场地上。

（2）分割式：所有数据只有一份，它被分割成若干逻辑片段，每个逻辑片段被指派在一个特定的场地上。

（4）全复制式：数据在每个场地重复存储。也就是每个场地上都有一个完整的数据副本。

（5）混合式：这是一种介乎于分割式和全复制式之间的分配方式。

