可行性报告

一.去中心化

1. Hash算法

哈希的英文名字为 Hash，意思为散列，它将任意 长度的输入值通过散列算法，变换成固定长度的输出 值。这个值就是散列值，即哈希值。哈希算法的方式 有很多，在分布式存储系统中主要是对文件名和路径 进行哈希，决定数据的分布策略。目前广为流行的有 经 典 的 一 致 性 哈 希 算 法 、字 符 串 哈 希 算 法 、M D 4 、M D 5 、 SHA-1、Davies-Meyer 等。

文献[1]提出了一致性哈希算法，现在在分布式 系统中应用非常广泛。一致性哈希算法在移除或者添加一个服务器时，能够尽可能小地改变已存在的服务 请求与处理请求服务器之间的映射关系。常用的字符 串哈希算法有 BKDRHash、APHash、DJBHash、JSHash 等。MD4 算法是哈希算法中较为成熟的算法之一。 MD4 算法可以对任意长度不超过 264 的消息进行处 理，生成一个 128 bit 的哈希值。消息在处理前，首先 要进行填充，保证 MD 填充后的 bit 位长度是 512 bit 的整数倍。填充结束后，利用迭代结构和压缩函数来 顺序处理每个512bit的消息分组。MD5算法是 MD4 算法的升级版。在 MD5 算法中，原始消息的预处 理操作和 MD4 是完全相同的，都需要进行补位、补位 长度操作，它们的信息摘要的大小都是 128 bit。MD5 在 MD4 的基础上加入了第四轮的计算模式，每一步骤 都是一一对应的固定值，改进了 MD4 在第二轮、第三 轮计算中的漏洞，完善了访问输入分组的次序，从而减 小其对称性和相同性。

SHA-1算法是在MD5算法基础上发展而来 的，其主要功能是从输入长度不大于 264 bit 的明文消 息中得到长度为 160 bit 的摘要值。该算法通过计算 明文信息得到固定长度的信息摘要，只要原始信息改变，摘要也随之发生改变，而且变化很大。这种发散性 可以检测数据的完整性，因此 SHA-1 算法在数字签名 中有着广泛的应用。Davies-Meyer 算法是基于对 称分组算法的单向散列算法。分组算法在设计和实现上成本较低，可以构建一个基于该分组密码的哈希函数。

2.一致性Hash算法

2.1一致性哈希算法

一致性哈希算法是一种分布式算法，在分布式存储中做数据分布时比较常用，Memcached client 也选择 这种算法，解决将 key-value 均匀分布到众多 Mem- cached server 上的问题。该算法实现思路如下:

首先求出每个存储节点的哈希值，并将其配置到 一个 0 ~ 232 -1(哈希值是一个 32 位的无符号整型)的 圆环区间上。其次使用同样的方法求出所需要存储的 键的哈希，也将其配置到这个圆环上。然后从数据映 射到的位置开始顺时针查找，将数据映射到找到的第 一个存储节点上。如果超过 232 仍然找不到存储节点， 就会映射到第一个存储节点上。

2.2数据倾斜问题

一致性哈希算法具有随机性。当存储节点组数量

较少时节点在环上分布不够均匀，会使得部分节点组 分布的数据量较少，部分节点组分布的数据明显增多 的 情 况 。 为 解 决 这 个 问 题 ， 提 出 了 基 于 虚 拟 节 点 的 改进算法。其核心思路是引入虚拟节点，每个虚拟节 点都有一个对应的物理存储节点，而每个物理存储节 点 可 以 对 应 若 干 个 虚 拟 节 点 。 引 入 “ 虚 拟 节 点 ”后 ， 映 射关系就从对{ 数据->节点} 转换到了{ 数据->虚拟 节点} 。

二.动态负载均衡

1.负载均衡策略设计思想及算法

1.轮询：每个请求按时间顺序逐一分配到不同的后端服务器，如果后端某台服务器宕机，则自动剔除故障机器，使用户访问不受影响。

2.weight：指定轮询权重，weight值越大，分配到的几率就越高，主要用于后端每台服务器性能不均衡的情况。

3.ip_hash：每个请求按访问IP的哈希结果分配，这样每个访客固定访问一个后端服务器，可以有效的解决动态网页存在的session共享问题

4.fair：更智能的一个负载均衡算法，此算法可以根据页面大小和加载时间长短智能地进行负载均衡，也就是根据后端服务器的响应时间来分配请求，响应时间短的优先分配。

5.url_hash：按访问URL的哈希结果来分配请求，使每个URL定向到同一台后端服务器，可以进一步提高后端缓存服务器的效率。

6.一种用于n个存储节点的非共享型分布式存储系统为m个节点的客户端服务器提供s个数据单位的存储服务的负载均衡策略，其包括以下步骤：步骤1：统计如下信息：统计各存储节点上对每一个客户端连接的数据存取访问，统计s个数据单位中的每一个数据单位通过不同存储节点的访问次数，统计每个节点的空间使用率；步骤2：根据步骤1中的上述统计数据，预先制定带宽阈值，远程访问阈值和容量阈值，所有的阈值均为百分比，判断分布式存储系统的各节点是否出现网络负载不均衡，跨节点访问次数过多导致的高延迟和容量极度不均衡，并根据判断结果选择迁移数据或者通过路由重定向客户端接入访问点。

[1]Devine R. Design and implementation of DDH: a distributed dynamic hashing algorithm[M]//Foundations of data organi-zation and algorithms. Berlin: Springer，1993: 101-114.

一致性Hash算法的解释

负载均衡的常见算法及实现