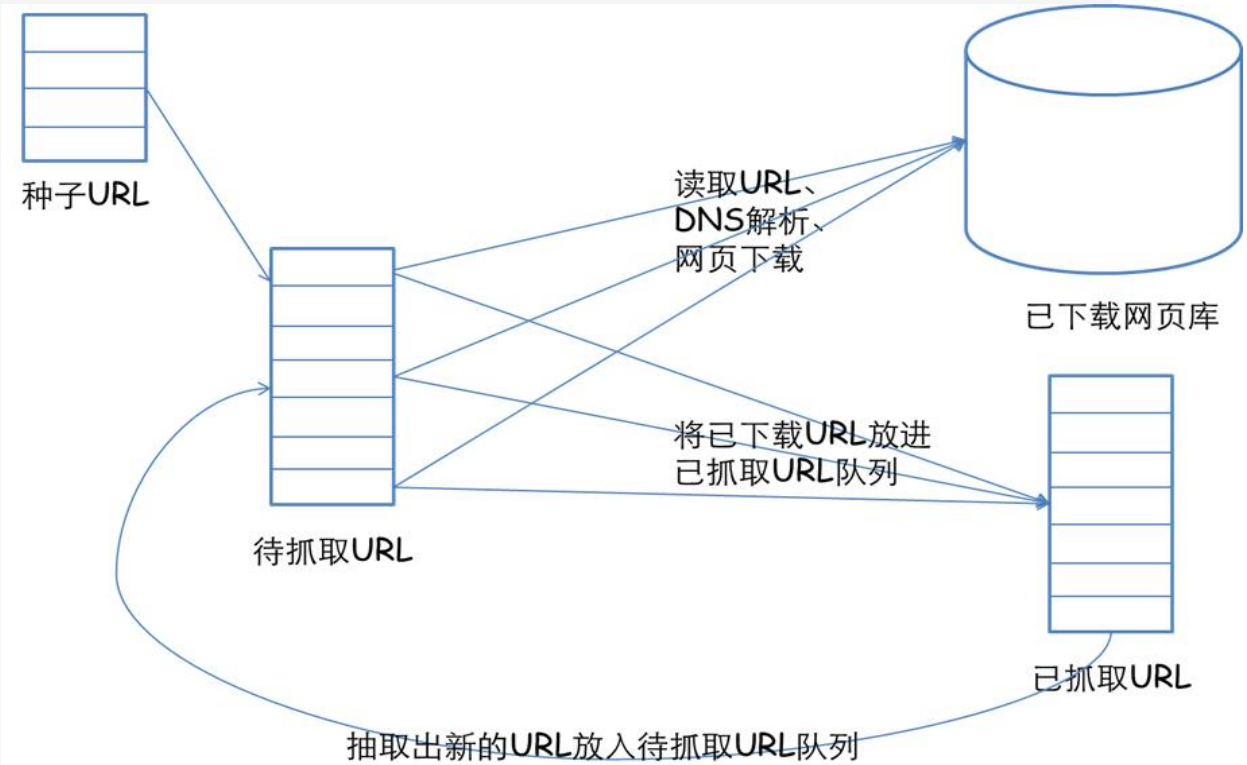
# 爬虫技术原理

说明：网络爬虫是搜索引擎抓取系统的重要组成部分。爬虫的主要目的是将互联网上的网页下载到本地形成一个或联网内容的镜像备份

一、网络爬虫的基本结构及工作流程

一个通用的网络爬虫的框架如图所示：




网络爬虫的基本工作流程如下：

1.首先选取一部分精心挑选的种子 URL；

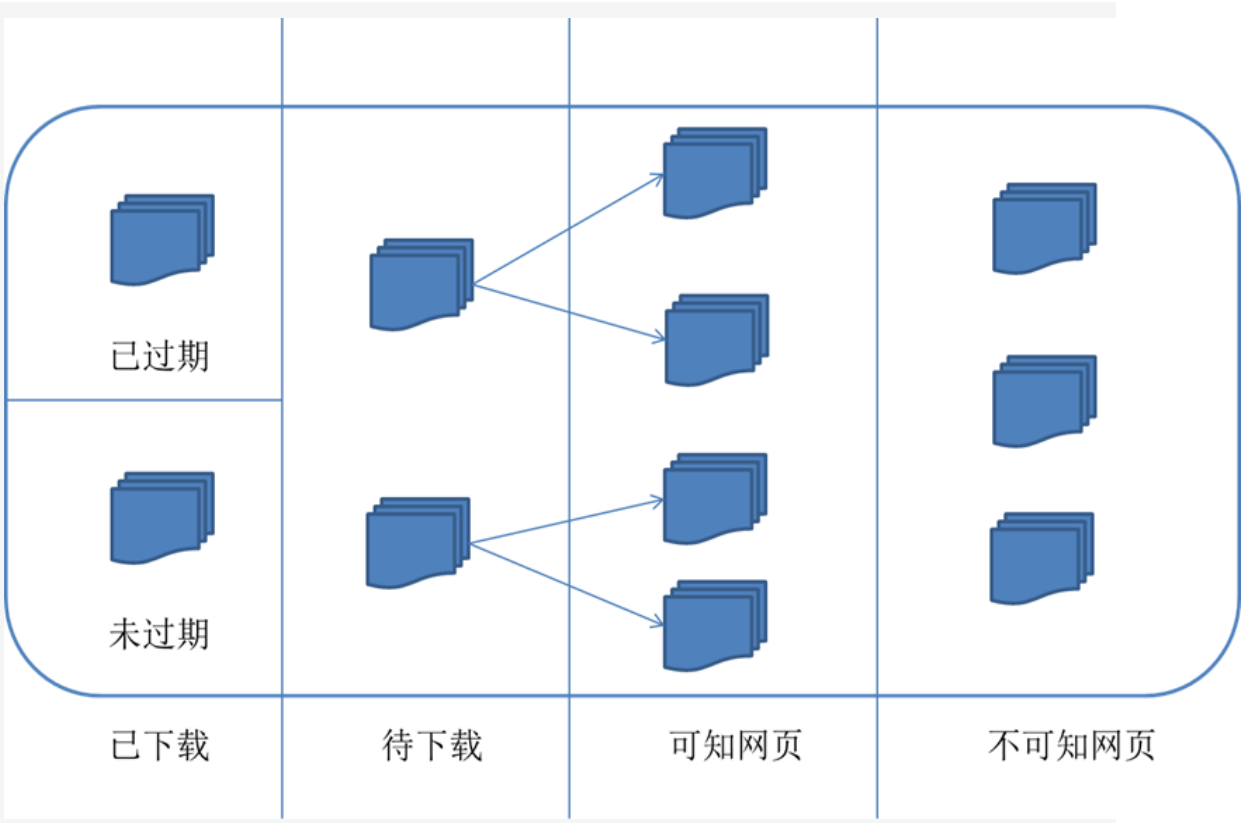
2.将这些 URL 放入待抓取 URL 队列；

3.从待抓取 URL 队列中取出待抓取在 URL，解析 DNS，并且得到主机的 ip，并将 URL 对应的网页下载下来，存储进已下载网页库中。此外，将这些 URL 放进已抓取 URL 队列。

4.分析已抓取 URL 队列中的 URL，分析其中的其他 URL，并且将 URL 放入待抓取 URL 队列，从而进入下一个循环。

二、从爬虫的角度对互联网进行划分

对应的，可以将互联网的所有页面分为五个部分：




1.已下载未过期网页

2.已下载已过期网页：抓取到的网页实际上是互联网内容的一个镜像与备份，互联网是动态变化的，一部分互联网上的内容已经发生了变化，这时，这部分抓取到的网页就已经过期了。

3.待下载网页：也就是待抓取 URL 队列中的那些页面

4.可知网页：还没有抓取下来，也没有在待抓取 URL 队列中，但是可以通过对已抓取页面或者待抓取 URL 对应页面进行分析获取到的 URL，认为是可知网页。

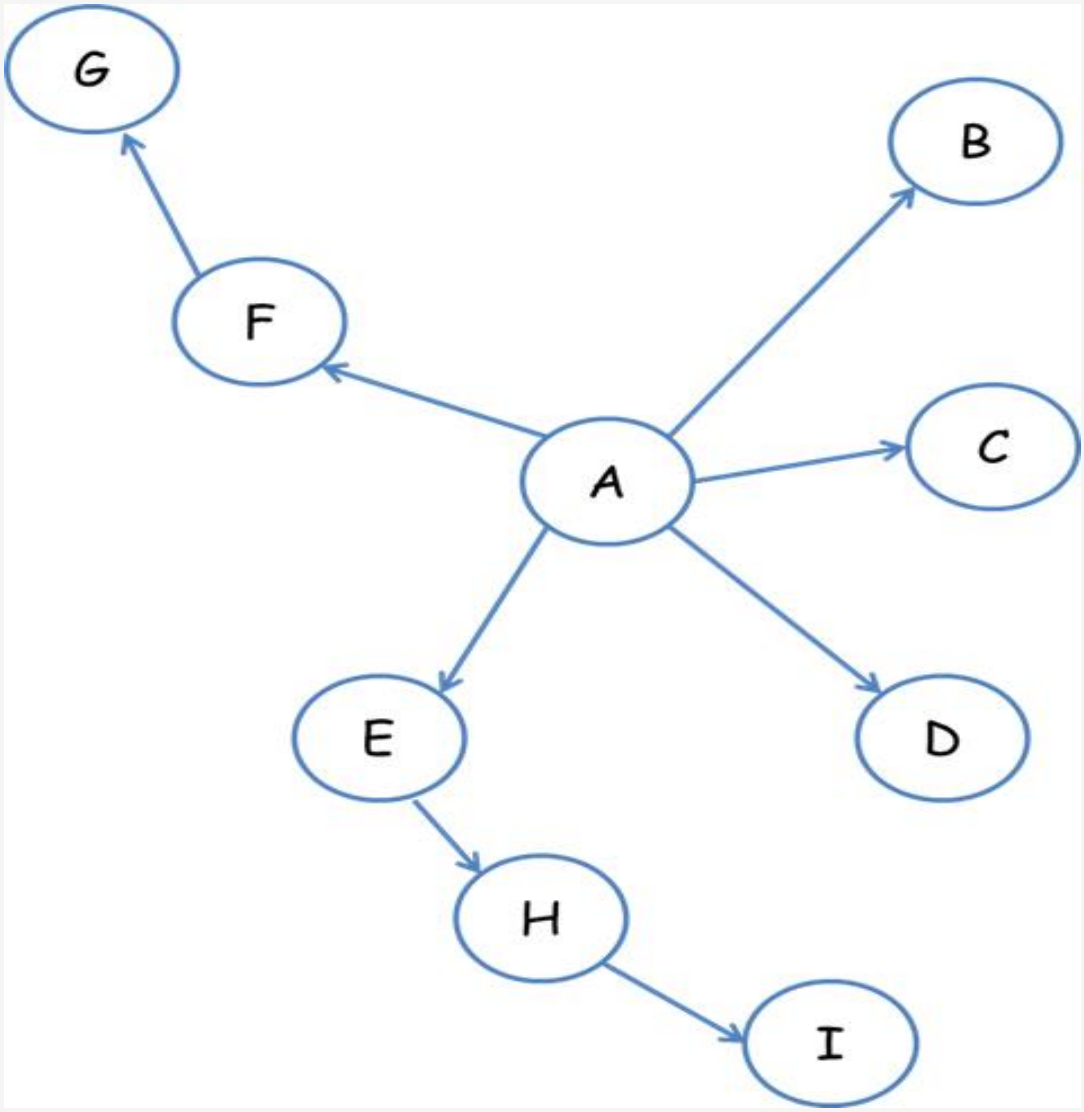
5.还有一部分网页，爬虫是无法直接抓取下载的。称为不可知网页。

三、抓取策略

在爬虫系统中，待抓取 URL 队列是很重要的一部分。待抓取 URL 队列中的 URL 以什么样的顺序排列也是一个很重要的问题，因为这涉及到先抓取那个页面，后抓取哪个页面。而决定这些 URL 排列顺序的方法，叫做抓取策略。下面重点介绍几种常见的抓取策略：

1.深度优先遍历策略

深度优先遍历策略是指网络爬虫会从起始页开始，一个链接一个链接跟踪下去，处理完这条线路之后再转入下一个起始页，继续跟踪链接。我们以下面的图为例：




遍历的路径：A-F-G  E-H-I B C D

2.宽度优先遍历策略

宽度优先遍历策略的基本思路是，将新下载网页中发现的链接直接插入待抓取 URL 队列的末尾。也就是指网络爬虫会先抓取起始网页中链接的所有网页，然后再选择其中的一个链接网页，继续抓取在此网页中链接的所有网页。还是以上面的图为例：

遍历路径：A-B-C-D-E-F G H I

3.反向链接数策略

反向链接数是指一个网页被其他网页链接指向的数量。反向链接数表示的是一个网页的内容受到其他人的推荐的程度。因此，很多时候搜索引擎的抓取系统会使用这个指标来评价网页的重要程度，从而决定不同网页的抓取先后顺序。

在真实的网络环境中，由于广告链接、作弊链接的存在，反向链接数不能完全等他我那个也的重要程度。因此，搜索引擎往往考虑一些可靠的反向链接数。

4.Partial PageRank 策略

Partial PageRank 算法借鉴了 PageRank 算法的思想：对于已经下载的网页，连同待抓取 URL 队列中的 URL，形成网页集合，计算每个页面的 PageRank 值，计算完之后，将待抓取 URL 队列中的 URL 按照 PageRank 值的大小排列，并按照该顺序抓取页面。

如果每次抓取一个页面，就重新计算 PageRank 值，一种折中方案是：每抓取 K 个页面后，重新计算一次 PageRank 值。但是这种情况还会有一个问题：对于已经下载下来的页面中分析出的链接，也就是我们之前提到的未知网页那一部分，暂时是没有 PageRank 值的。为了解决这个问题，会给这些页面一个临时的 PageRank 值：将这个网页所有入链传递进来的 PageRank 值进行汇总，这样就形成了该未知页面的 PageRank 值，从而参与排序。下面举例说明：

5.OPIC 策略策略

该算法实际上也是对页面进行一个重要性打分。在算法开始前，给所有页面一个相同的初始现金（cash）。当下载了某个页面 P 之后，将 P 的现金分摊给所有从 P 中分析出的链接，并且将 P 的现金清空。对于待抓取 URL 队列中的所有页面按照现金数进行排序。

6.大站优先策略

对于待抓取 URL 队列中的所有网页，根据所属的网站进行分类。对于待下载页面数多的网站，优先下载。这个策略也因此叫做大站优先策略。

参考书目：

1.《这就是搜索引擎—核心技术详解》　张俊林　电子工业出版社

2.《搜索引擎技术基础》　刘奕群等　清华大学出版社

四、更新策略

互联网是实时变化的，具有很强的动态性。网页更新策略主要是决定何时更新之前已经下载过的页面。常见的更新策略又以下三种：

1.历史参考策略

顾名思义，根据页面以往的历史更新数据，预测该页面未来何时会发生变化。一般来说，是通过泊松过程进行建模进行预测。

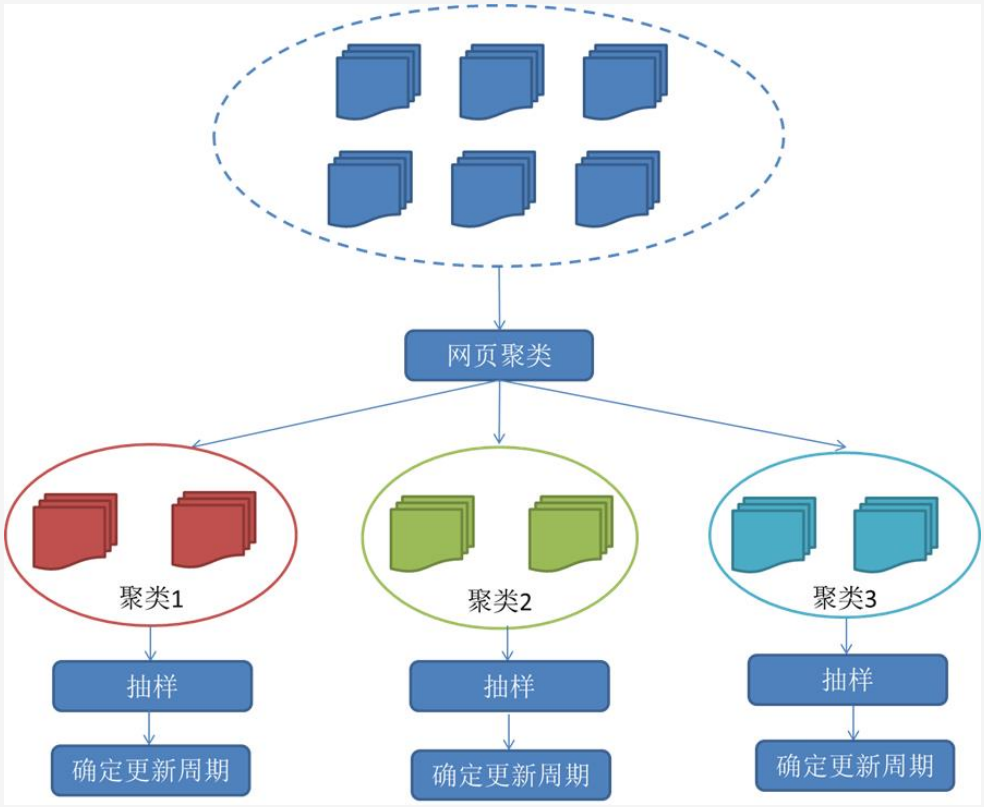
2.用户体验策略

尽管搜索引擎针对于某个查询条件能够返回数量巨大的结果，但是用户往往只关注前几页结果。因此，抓取系统可以优先更新那些现实在查询结果前几页中的网页，而后再更新那些后面的网页。这种更新策略也是需要用到历史信息的。用户体验策略保留网页的多个历史版本，并且根据过去每次内容变化对搜索质量的影响，得出一个平均值，用这个值作为决定何时重新抓取的依据。

3.聚类抽样策略

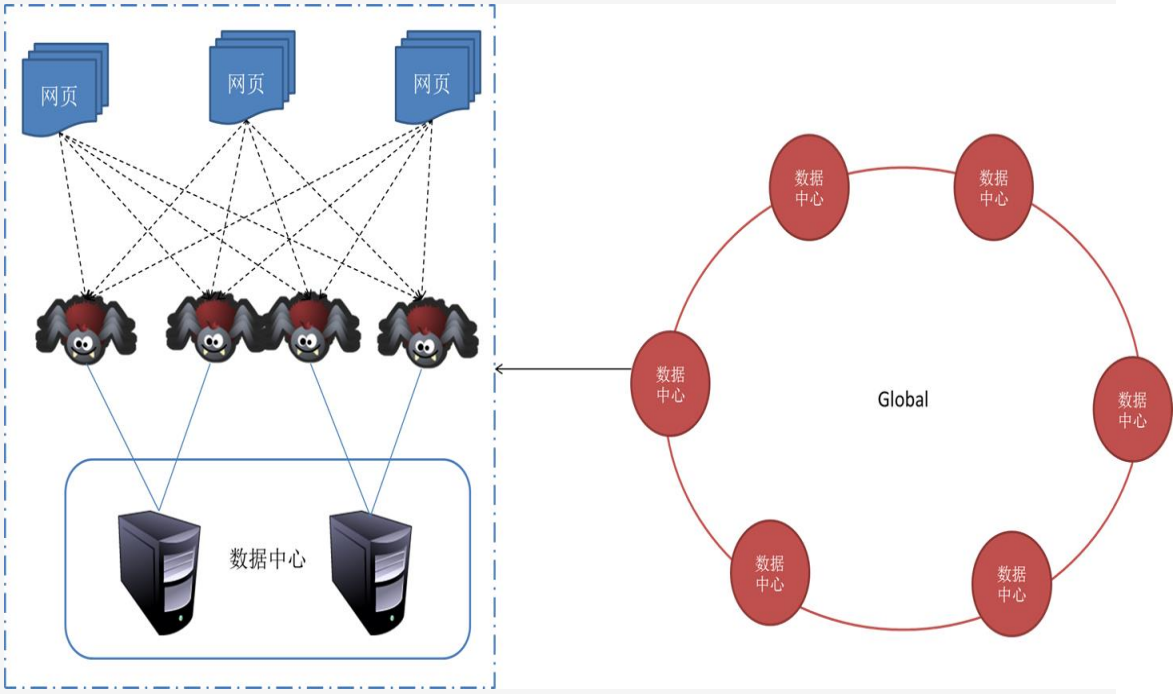
前面提到的两种更新策略都有一个前提：需要网页的历史信息。这样就存在两个问题：第一，系统要是为每个系统保存多个版本的历史信息，无疑增加了很多的系统负担；第二，要是新的网页完全没有历史信息，就无法确定更新策略。

这种策略认为，网页具有很多属性，类似属性的网页，可以认为其更新频率也是类似的。要计算某一个类别网页的更新频率，只需要对这一类网页抽样，以他们的更新周期作为整个类别的更新周期。基本思路如图：




五、分布式抓取系统结构

一般来说，抓取系统需要面对的是整个互联网上数以亿计的网页。单个抓取程序不可能完成这样的任务。往往需要多个抓取程序一起来处理。一般来说抓取系统往往是一个分布式的三层结构。如图所示：

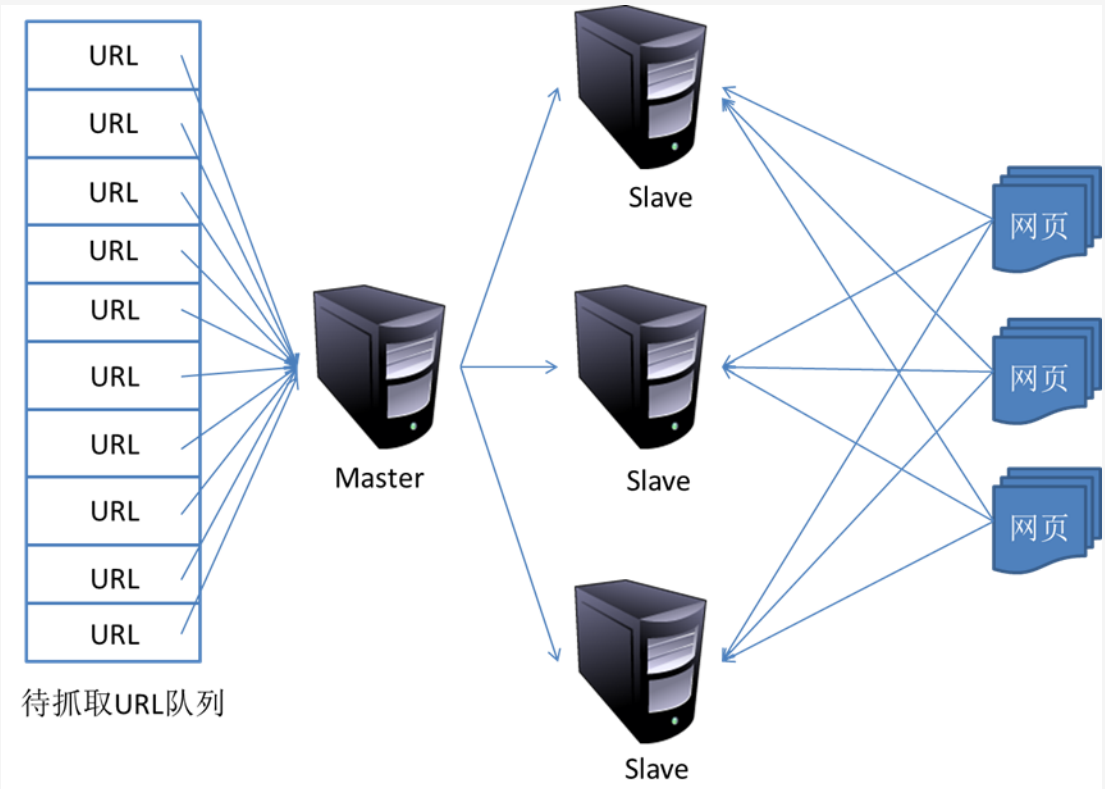


最下一层是分布在不同地理位置的数据中心，在每个数据中心里有若干台抓取服务器，而每台抓取服务器上可能部署了若干套爬虫程序。这就构成了一个基本的分布式抓取系统。

对于一个数据中心内的不同抓去服务器，协同工作的方式有几种：

1.主从式（Master-Slave）

主从式基本结构如图所示：

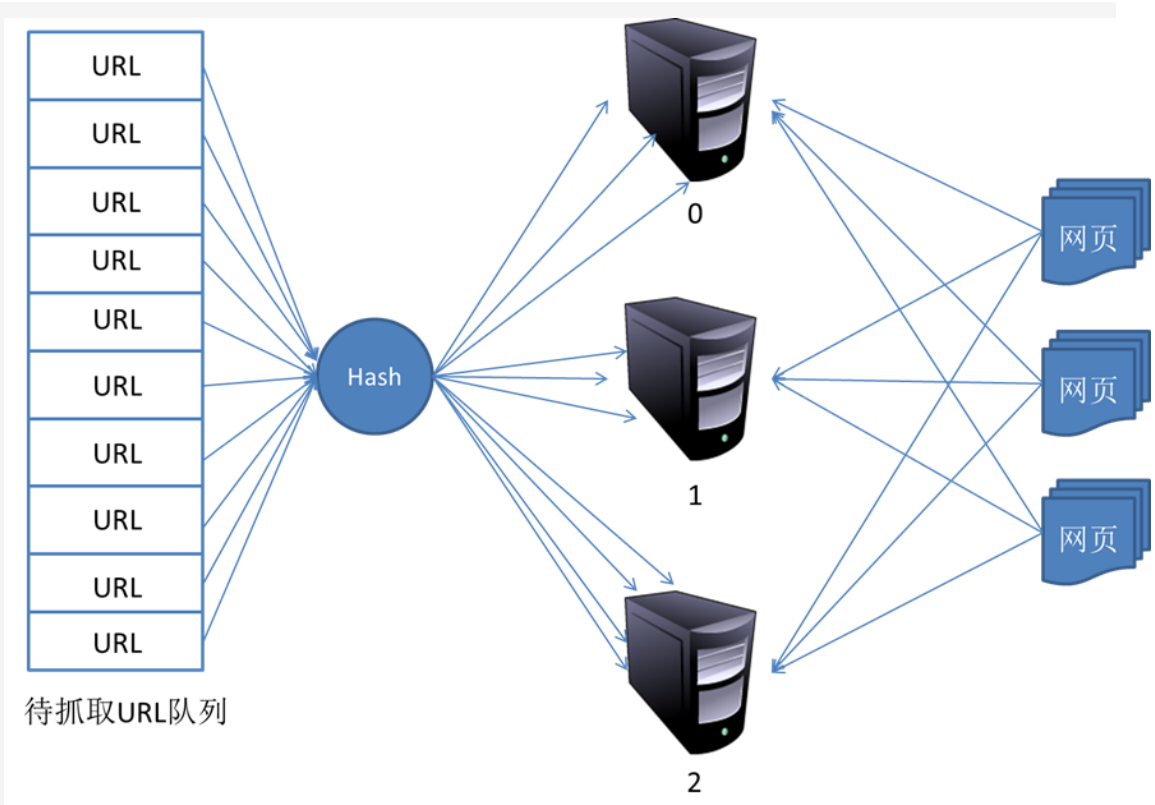



对于主从式而言，有一台专门的 Master 服务器来维护待抓取 URL 队列，它负责每次将 URL 分发到不同的 Slave 服务器，而 Slave 服务器则负责实际的网页下载工作。Master 服务器除了维护待抓取 URL 队列以及分发 URL 之外，还要负责调解各个 Slave 服务器的负载情况。以免某些 Slave 服务器过于清闲或者劳累。

这种模式下，Master 往往容易成为系统瓶颈。

2.对等式（Peer to Peer）

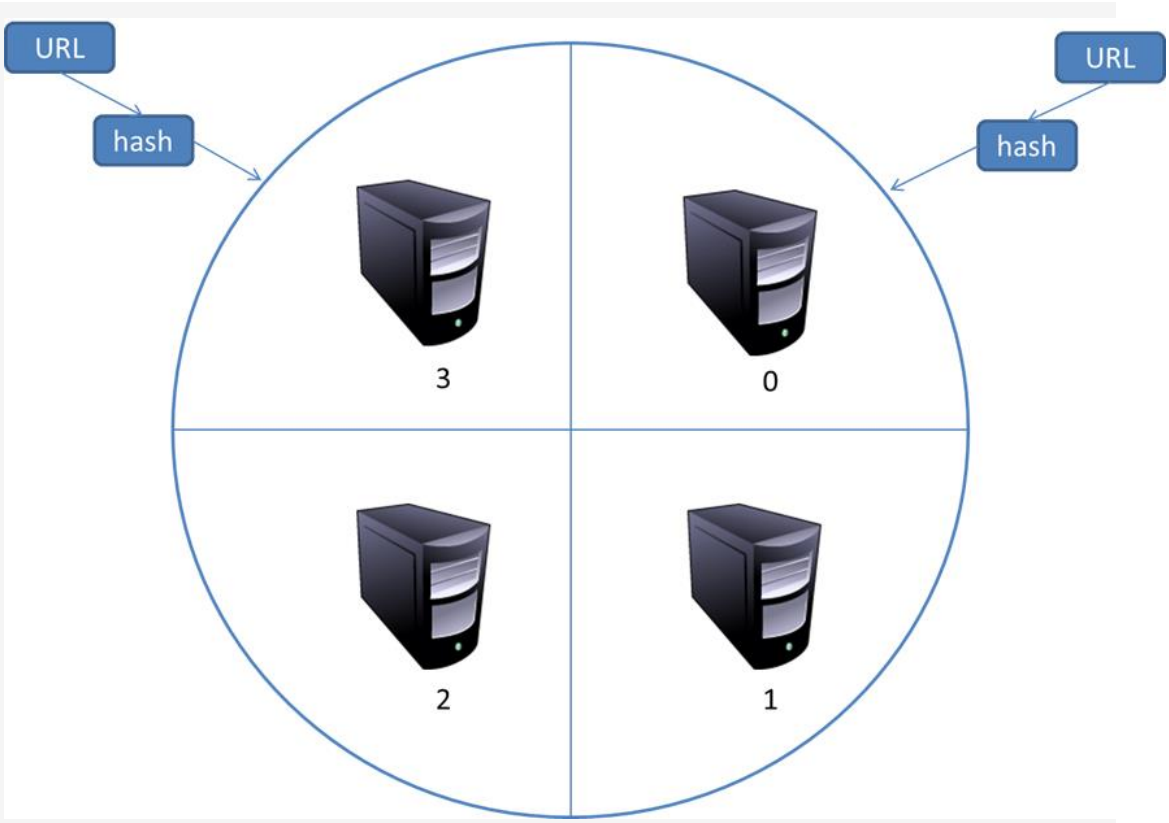
对等式的基本结构如图所示：




在这种模式下，所有的抓取服务器在分工上没有不同。每一台抓取服务器都可以从待抓取在 URL 队列中获取 URL，然后对该 URL 的主域名的 hash 值 H，然后计算 H mod m（其中 m 是服务器的数量，以上图为例，m 为 3），计算得到的数就是处理该 URL 的主机编号。

举例：假设对于 URL www.baidu.com，计算器 hash 值 H=8，m=3，则 H mod m=2，因此由编号为 2 的服务器进行该链接的抓取。假设这时候是 0 号服务器拿到这个 URL，那么它将该 URL 转给服务器 2，由服务器 2 进行抓取。

这种模式有一个问题，当有一台服务器死机或者添加新的服务器，那么所有 URL 的哈希求余的结果就都要变化。也就是说，这种方式的扩展性不佳。针对这种情况，又有一种改进方案被提出来。这种改进的方案是一致性哈希法来确定服务器分工。其基本结构如图所示：




一致性哈希将 URL 的主域名进行哈希运算，映射为一个范围在 0-$2^{32}$ 之间的某个数。而将这个范围平均的分配给 m 台服务器，根据 URL 主域名哈希运算的值所处的范围判断是哪台服务器来进行抓取。

如果某一台服务器出现问题，那么本该由该服务器负责的网页则按照顺时针顺延，由下一台服务器进行抓取。这样的话，及时某台服务器出现问题，也不会影响其他的工作。

参考书目：

1.《这就是搜索引擎—核心技术详解》　　张俊林　　电子工业出版社

2.《搜索引擎技术基础》　　刘奕群等　清华大学出版社