论文讲了一个实现单个服务器爬取数以十亿计的网页的实现，并对它的表现进行了建模分析。文章着重于解决搜索引擎两个基本组成——网络爬虫和数据挖掘中的前者。网络爬虫面临着可扩展性、避免现如无用的内容、爬虫礼貌的三个具有挑战性的问题。文章第一部分提出了一套网络爬虫的算法来解决上述三个，第二部分简要检查在实际操作过程中的性能。

研究中遇到第一个瓶颈是检测url的唯一性问题和是否符合robots.txt，通过DRUM技术，在磁盘上存储大量哈希数据，通过桶排序实现查找和更新。第二个瓶颈有数万个页面站点造成，通过来自其他域的入度链接数量，动态分配每个域及其所有子域的允许页面的预算。

对于瓶颈体现在URL确认和机器人规则上的规模问题，主要是因为牵扯到硬盘和存储器的交互，使用以前的方法，随着爬取规模的增大，这一步开销会快速增长，为了降低这个开销，需要使用到更有效地存储结构。也就是文章提到的DRUM。它结合桶排序和哈希算法，将所有的桶中的数据和硬盘中的数据进行结合，从而实现大规模键值的数据存储，方便快速的查找和更新。此外，文章通过一系列参数来推导开销，证实了DRUM的优越性。

对于网站名声和垃圾网站问题，由于拥有大量动态网页的合法网站和制造了大量垃圾网页的恶意网站，使得爬取变得低效，带来巨大的带宽浪费，之前的研究中的爬取策略容易让爬虫陷入这种不断生成动态网页的网站中，文章在“网站名声”的基础上给每个域名分配预算，侦察垃圾网站，避免垃圾网站的困扰。

对于礼貌问题，由于短时间频繁对某一服务器访问会形成DOS攻击，之前的策略可能会导致“多主机公用”的服务器崩溃或者效率降低，使得在大规模的网页爬取中最终无法正常工作，文章使用BEATS，在确定链接的合法性之前，延迟下载这个链接的时机，合理地决定网页爬取顺序。这样保留了队列的不同优先级，另一方面使得预算较低地链接不断的推迟被爬取的时机。

然后是实验验证，在2.6GHz的4核AMDCPU上，41天的时间，通过76亿测尝试，得到74亿个有效响应，排除没有html内容和重定向，最终得到63亿个有效的页面。从结果中可以看出，域名的流行度，和它在爬虫运行时分配到的带宽有着明显的正相关，可以说明网站名声计算的正确性。