(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102693315 A
(43)申请公布日 2012.09.26

(21)申请号 201210171316.8

(22)申请日 2012.05.29

(71)申请人 上海家配电子商务有限公司
地址 200062 上海市普陀区中江路106号北岸长风i座17层1711

(72)发明人 司贺华 孔凡兵

(51)Int.Cl.
G06F 17/30(2006.01)

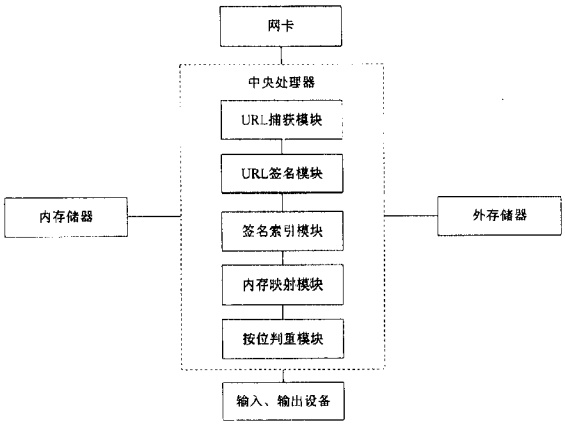
权利要求书 2 页 说明书 6 页 附图 2 页

(54)发明名称

一种基于共享内存映射的URL去重方法及装置

(57)摘要

本发明的基于共享内存映射的URL去重方法及装置，去重方法包括初始化、URL签名、签名索引、共享内存映射和按位判重。本发明的有益效果在于通过签名，将URL最终以签名文件中的1位来存储，极大的压缩了URL所占的存储空间；通过共享内存映射，将对内存的操作同步到外存签名文件中，提高了URL去重的速度；将URL对应的位存储于签名文件中，长期保存了去重结果。本发明还提供了一种实现上述方法的装置，装置包括中央处理器、内存储器、外存储器、网卡和输入输出设备，这些设备均与中央处理器连接，所述的中央处理器又包括连接在一起的URL捕获模块、URL签名模块、签名索引模块、共享内存映射模块和按位判重模块。



CN 102693315 A

1. 基于共享内存映射的URL去重方法，包括初始化模块、URL签名模块、签名索引模块、共享内存映射模块和按位判重模块；

其特征包括：

装置初始化模块：装置所需要相关数据的加载与共享内存映射的初始化；

URL签名模块：用于将不定长的URL转换成定长的签名，此签名在一定范围内是唯一对应的；签名的方法包括但不限于CRC32(Cyclical Redundancy Check 32，共有32位二进制数)，SHA1(Secure Hash Algorithm 1，共有160位二进制数)，MD5(Message Digest Algorithm5，共有128位二进制数)等算法；

签名索引模块：根据签名的字节数x(x = x11+x12)，选取前x11字节作为文件名，剩余x12字节作为文件内容；签名字节数x比较大，则拆分为3级甚至更多(若3级以上，则分为多级目录)，以3级为例：x = x21+x22+x23，将签名的前x21字节作为目录名，中间x22字节作为文件名，剩余x23字节作为文件内容；索引的方法就是根据签名指定的字节查找即可；

共享内存映射模块：由于存储为文件形式，但读写文件系统开销很大，故采用共享内存映射的方法，将内存和上述签名文件一一对应；直接对内存操作，即由共享内存映射模块完成内存与文件的同步；

按位判重模块：根据签名索引，找到签名文件内容对应共享内存位置，读取内存的值，并判断此URL是否重复。

2. 根据权利要求1所述的基于共享内存映射的URL去重方法，其特征在于包括以下步骤：

1) 初始化步骤：根据预定的签名索引方案，初始化文件到内存，完成共享内存映射的初始化；

2)URL捕获步骤，是从网卡获取的信息中的URL识别出来供装置去重。

3)URL签名步骤，是从获取URL开始，将URL通过签名算法转换为定长的字节，然后将签名字节传给签名索引步骤，根据签名算法的不同，采用不同的详细步骤。

4) 签名索引步骤，是根据所选的目录级别，选定的索引方案，其实现直接影响装置初始化具体实现；

5) 共享内存映射步骤，使得进程之间通过映射同一个签名文件实现共享内存，签名文件被映射到进程地址空间后，进程可以像访问普通内存一样对文件进行访问，不必再调用read()，write() 等操作；换句话说就是把一个文件的内容在内存里面做一个映像，内存操作比磁盘操作要快；

6) 按位判重步骤，根据签名文件内容字段的值，索引找到所在位，判断此位是否为1，若为1则此URL为重复URL，若为0则将此位置1，并返回非重URL的结果。

3. 根据权利要求1所述的方法制作成的URL去重装置，其特征包括中央处理器、内存储器、外存储器、网卡和输入输出设备，这些设备均与中央处理器连接，所述的中央处理器又包括连接在一起的URL捕获模块、URL签名模块、签名索引模块、共享内存映射模块和按位判重模块。

4. 根据权利要求1所述的基于共享内存映射的URL去重方法，其特征在于总体设计如下：首先装置初始运行时进行初始化，将存储URL的签名文件全部打开，如果文件不存在则创建并将所有位置0，共享内存映射模块进行共享内存映射获取SMM指针，然后即可关闭文

件；以后对 SMM 指针的读写操作，就会映射到 URL 存储文件的读写；将 SMM 指针全部放到一个数组中管理，以后根据 CRC32 码的前 8 位，从这个数组中找到相应的 SMM 指针，再判断此 SMM 指针对应的内存区的相应位的值，如果是 1，则表示为重复的 URL，如果是 0，就是非重复的 URL，同时将此位置为 1；重复如上流程，即是基于共享内存映射的 URL 去重方法的实现。

5. 根据权利要求 4 所述的共享内存映射，其特征在于先要判断签名文件是否已经被创建，若是则直接打开，否则需要创建文件，并固定文件大小且按位置 0；然后再进行共享内存映射，同时需要 smm 同步，并返回 smm 指针，供调用去重装置所用。

6. 根据权利要求 1 所述的基于共享内存映射的 URL 去重方法，其特征在于有益效果在于通过签名，将 URL 最终以签名文件中的 1 位来存储，极大的压缩了 URL 所占的存储空间；通过共享内存映射，将对内存的操作同步到签名文件中，提高了 URL 去重的速度；将 URL 对应的位存储于签名文件中，长期保存了去重结果。

一种基于共享内存映射的 URL 去重方法及装置

技术领域

[0001] 本发明属于计算机网络信息处理领域，涉及统一资源定位符 URL(Uniform Resource Locator) 的去重方法与产品，具体涉及一种基于共享内存映射的 URL 去重方法及装置。

背景技术

[0002] 近年来互联网技术的发展，对重复统一资源定位符 URL(Uniform Resource Locator) 的采集过滤与去重处理，显得越来越重要。URL 去重技术广泛应用于网络审计系统、搜索引擎系统中。如果能够快速识别出重复 URL，免去重复 URL 的后续处理步骤，将大大提高性能。

[0003] URL 去重技术，需要从两个方面考虑 :URL 存储空间和 URL 匹配速度。URL 的存储空间是指可以处理非重复 URL 的最大数目和每条 URL 所占用的内存空间。URL 匹配速度是通过判断一条 URL 记录是否为重复 URL 所用的时间来衡量的。

[0004] 国外应用 URL 去重技术的相关产品很多，所有的爬虫系统都需要用到此功能。开源的搜索引擎 Larbin 使用了 Bloom Filter 算法实现了 URL 的去重过滤功能，这是一种 URL 所占存储空间低而且 URL 过滤速度比较快的算法。Bloom Filter 算法是 Burton Bloom 在 1970 年提出的 (Burton Bloom.Bloom filter[EB/OL].http://en.wikipedia.org/wiki/Bloom_filter,1970)。然而，Bloom Filter 算法有一个缺陷，即有一定程度的误判，即将非重复的 URL 判定为重复的 URL。

[0005] 在提高 URL 匹配去重速度的研究领域中，哈希表 (Hash Table) 是一个重要的研究方向。由于哈希表在查找信息时速度非常快，所以在内存中存储 URL 时一般都采用基于哈希表的存储结构，因而在 URL 去重技术中针对哈希表和哈希函数的研究比较有价值。西安交通大学的郑卫斌等人实现了基于哈希表的 URL 去重器，使用位图法提高了 URL 过滤的性能 (郑卫斌，张德运，丁会宁，李继华，高磊 . 基于哈希表的高性能 URL 过滤器研究 . 小型微型计算机系统 .2005,26(2) ;17-180 页)。郑卫斌的 URL 去重器选取适用于字符串哈希的位哈希函数，也实现了简单的存储优化和高速缓存优化，并通过实验证明了这种基于位图法哈希表的 URL 去重器的高效性。使用哈希表来进行 URL 匹配去重的技术，保存的 URL 数目取决于哈希表桶的大小，到最后会受限于内存的大小，同时哈希表方案完全存储于内存，并未保存上一次的结果，去重的记录未能保存，因此完全使用哈希表的 URL 去重装置在海量 URL 去重中不适合。

[0006] 除了将海量的 URL 存储到硬盘上，对 URL 本身的存储进行优化，也是研究的内容之一。Kasom 等人提出了一种基于 Delta Encoding 的方法，并使用 AVL 树来提高查找的效率，取得了大约 50％的压缩率，达到的效果非常明显 (Koht-arsa K,Sanguanpong S.In-memory URL Compression, National Computer Science and Engineering Conference, Chiang Mai,Thailand,November 7-9,2001 :425-428)。Genova 等人提出了使用签名来提高 URL 查找速度和降低存储空间的方法 (Genova Z, Christensen K.Efficient Summarization of

URLs using CRC32 for Implementing URL Switching, CONFERENCE ON LOCAL COMPUTER NETWORKS, 2002 :102-105P)。他所谓的签名，实际上就是使用 CRC32(Cyclic Redundancy Check 32) 将 URL 编码成固定长度，使用 CRC32 码来代替 URL 字符串作为哈希表的键，这样既降低的 URL 的存储空间，又提高了查找和去重速度。然而，海量 URL 的 CRC32 编码全部存储内存中，仍然受限于硬件条件，同时匹配速度也是个问题。

发明内容

[0007] 针对现有技术的上述不足，本发明的目的在于提供高效的基于共享内存映射 SMM(Shared Memory-Mapped) 的 URL 去重方法，解决现有 URL 去重方法的不足。本发明的目的还在提供一种实现上述方法的 URL 去重装置，实现了 URL 去重的功能，提高了去重速度、降低了存储空间（每条 URL 占 1 位），同时保存了去重结果，可以应用于搜索引擎、网络审计系统等。

[0008] 本发明的基于共享内存映射的 URL 去重方法及装置，其特征包括 ：

[0009] 去重方法包括初始化、URL 签名、签名索引、共享内存映射和按位判重 ；装置包括中央处理器、内存储器、外存储器、网卡和输入输出设备，这些设备均与中央处理器连接，所述的中央处理器又包括连接在一起的 URL 捕获模块、URL 签名模块、签名索引模块、共享内存映射模块和按位判重模块。

[0010] 初始化 ：装置所需要相关数据的加载与共享内存映射的初始化 ；

[0011] 本发明的目的是通过下列技术方案实现的 ：

[0012] 本装置所述的中央处理器用于连接控制装置的各个模块，完成装置的功能 ；

[0013] 本装置所述的网卡用于捕获存储 URL 信息的数据包 ；

[0014] 本装置所述的内存储器是装置运行中数据的存储介质 ；

[0015] 本装置所述的外存储器是装置保存去重结果文件的存储介质 ；

[0016] 本装置所述的 URL 捕获模块通过网卡将网络数据包解析，并分析出相应的 URL，供装置其它模块使用 ；

[0017] URL 签名模块 ：用于将不定长的 URL 转换成定长的签名，此签名在一定范围内是唯一对应的 ；签名的方法包括但不限于 CRC32(Cyclical Redundancy Check 32，共有 32 位二进制数），SHA1(Secure Hash Algorithm 1，共有 160 位二进制数），MD5(Message Digest Algorithm5，共有 128 位二进制数）等算法 ；

[0018] 签名索引模块 ：根据签名的字节数 x(x ＝ x11+x12)，选取前 x11 字节作为文件名，剩余 x12 字节作为文件内容 ；签名字节数 x 比较大，则拆分为 3 级甚至更多（若 3 级以上，则分为多级目录），以 3 级为例 ：x ＝ x21+x22+x23，将签名的前 x21 字节作为目录名，中间 x22 字节作为文件名，剩余 x23 字节作为文件内容 ；索引的方法就是根据签名指定的字节查找到相应的目录及文件 ；

[0019] 共享内存映射模块 ：由于存储为文件形式，但读写文件系统开销很大，故采用共享内存映射的方法，将内存和上述签名文件一一对应 ；直接对内存操作，即由共享内存映射模块完成内存与文件的同步 ；

[0020] 按位判重模块 ：根据签名索引，找到签名文件内容对应共享内存位置，读取内存的值，并判断此 URL 是否重复。

[0021] 本发明的基于共享内存映射的 URL 去重方法，包括以下步骤：

[0022] 1) 初始化步骤：根据预定的签名索引方案，初始化文件到内存，完成共享内存映射的初始化；

[0023] 2)URL 捕获步骤，是从网卡获取的信息中的 URL 识别出来供装置去重。

[0024] 3)URL 签名步骤，是从获取 URL 开始，将 URL 通过签名算法转换为定长的字节，然后将签名字节传给签名索引步骤，根据签名算法的不同，采用不同的详细步骤。

[0025] 4) 签名索引步骤，是根据所选的目录级别，选定的索引方案，其实现直接影响装置初始化具体实现；

[0026] 5) 共享内存映射步骤，使得进程之间通过映射同一个签名文件实现共享内存，签名文件被映射到进程地址空间后，进程可以像访问普通内存一样对文件进行访问，不必再调用 read()，write() 等操作；换句话说就是把一个文件的内容在内存里面做一个映像，内存操作比磁盘操作要快；

[0027] 6) 按位判重步骤，根据签名文件内容字段的值，索引找到所在位，判断此位是否为 1，若为 1 则此 URL 为重复 URL，若为 0 则将此位置 1，并返回非重 URL 的结果。

[0028] 本发明的有益效果在于通过签名，将 URL 最终以签名文件中的 1 位来存储，极大的压缩了 URL 所占的存储空间；通过共享内存映射，将对内存的操作同步到签名文件中，提高了 URL 去重的速度；将 URL 对应的位存储于签名文件中，长期保存了去重结果。

附图说明

[0029] 图 1 是本发明结构示意图；

[0030] 图 2 是基于共享内存映射的 URL 去重方法的流程图；

[0031] 图 3 是共享内存映射的处理流程；

[0032] 图 4 是基于位图法哈希表的 URL 去重流程；

具体实施方式

[0033] 下面结合附图举例对本发明做进一步详细描述：

[0034] 图 1 所示为一种基于共享内存映射的 URL 去重方法的流程，包括：

[0035] 装置初始化模块：装置所需要相关数据的加载与共享内存映射的初始化；

[0036] 中央处理器：用于连接控制装置的各个模块，完成装置的功能；

[0037] 网卡：用于捕获存储 URL 信息的数据包；

[0038] 内存储器：装置运行时数据的存储介质；

[0039] 外存储器：装置保存去重结果文件的存储介质；

[0040] URL 捕获模块：获取所需要去重的 URL；

[0041] URL 签名模块：用于将不定长的 URL 转换成定长的签名；

[0042] 签名索引模块：根据签名的字节分级索引到对应的签名文件；

[0043] 内存映射模块：将签名文件与内存进行共享映射，使装置对内存的操作同步更新到签名文件中；

[0044] 按位判重模块：根据签名文件内容中对应位的值来判断此 URL 是否重复。

[0045] 实施例

[0046] 图2给出了一种基于共享内存映射的URL去重方法的流程，以签名算法使用CRC32为例。

[0047] 1、总体设计

[0048] 基于共享内存映射的URL去重装置的实施例总体设计如下：首先装置初始运行时进行始化，将存储于外存的URL签名文件在内存中全部打开，如果文件不存在则创建并将所有位置0，共享内存映射模块进行共享内存映射获取SMM指针，然后即可关闭文件。以后对SMM指针的读写操作，就会映射到URL存储文件的读写。将网卡捕获的URL进行CRC32签名，获取CRC32码。将SMM指针全部放到一个数组中管理，根据CRC32码的前8位，从这个数组中找到相应的SMM指针，再判断此SMM指针对应的内存区的相应位的值，如果是1，则表示为重复的URL，如果是0，就是非重复的URL，同时将此位置为1。重复如上流程，即是基于共享内存映射的URL去重装置的实现流程。

[0049] 2、签名文件索引

[0050] 本实施例使用CRC32算法签名。CRC32是一个32位的值，将此信息存入文件，文件总共2^{32}位，即$2^{32}/8$等于512MB字节，所以可以考虑将其全部存入一个字节大小为512MB的文件中。然而此时有个问题，尽管一直读写的是一个文件，但是在2^{32}大小的文件中寻址，肯定会比较慢。因此，本实施例的方案是：取CRC32值前8位作为文件名，即能产生名为00到FF的256个文件，将其余24位存储在一个2MB大小的文件，单个文件寻址空间为2^{24}。

[0051] 3、共享内存映射

[0052] 基于共享内存映射的URL去重方法中，共享内存映射的处理流程如图3所示。模块先要判断签名文件是否已经被创建，若是则直接打开，否则需要创建文件，并固定文件大小且按位置0；然后再进行共享内存映射，同时需要SMM同步，并返回SMM指针，供调用去重装置所用。

[0053] 4、实验结果与分析

[0054] 1) 实验步骤

[0055] 实验的测试数据来源于一次网络捕包存储的包含URL信息的文件，所含URL数目为75012，则平均每条URL长度为50.5字节。

[0056] 实验的流程是：将此数据包文件中的URL进行URL去重流程，查询此URL(若查询不到此记录则插入此记录)，记录查询时间，然后读取下一条记录，重复上述操作。最后在75012条记录全部查完后，再重新读取此文件，记录第二次查询一遍全部URL的时间。

[0057] 实验方案：本实施例的测试结果与基于位图法哈希表的URL去重过滤器和基于Bloom Filter算法实现的URL去重器进行实验对比，论证了本发明实现URL去重方法的高效性。另外，根据每种方法，修改相关参数来获得相应的实验数据。每个方法实验若干次，取平均值。本发明中，每个装置的实验次数为4或5，然后得出一平均值，来证明实验的效果。

[0058] 2) 对比方案描述

[0059] 本发明的测试方案中实现了西安交通大学郑卫斌等人提出的基于位图法哈希表的高性能URL去重器，并与基于共享内存映射的URL去重装置进行了对比。基于位图法哈希表的URL去重流程如图4所示。

[0060] 基于Bloom Filter算法的URL去重方法设计：Bloom Filter算法的实现，影响其

处理速度的因素在于位数组大小 m 值和哈希函数个数 k 的选择，当然考虑误判率时，还要考虑元素的数量 n。在本实验实现的 Bloom Filter 算法中，对哈希函数个数 k 值的获取则采用了如下的方案：选择对 URL 字符串进行 CRC32 编码（对应 1 个无符号整形值）、MD5 编码（对应 4 个无符号整形值）、安全散列标准 SHA1 编码（对应 5 个无符号整形值），所以通过这三个算法编码过程所得的值，可以相当于 10 次哈希函数的值。而且，可以选择三个编码值的组合，就可以生成不同的 k 值，k 的最大值为 10，来获得不同的实验数据。

[0061] 3) 实验结果

[0062] 基于共享内存映射的 URL 去重装置实验结果如表 1 所示。

[0063] 表 1 基于共享内存映射的 URL 去重装置的实验结果

[0064]

运行次数	程序运行后第一次查询时间（单位：微秒）	查询一遍后再查询时间（单位：微秒）	URL 数目
1	2038838	170205	75012
2	342172	165876	75012
3	427925	169508	75012
4	342219	165811	75012
平均值	787788.5	167850	——

[0065] 使用基于位图法的 URL 去重器的实验结果如表 2 所示。

[0066] 表 2 使用基于位图法哈希表的 URL 去重器的实验结果

[0067]

哈希表大小（桶数）	第一次查询时间（单位：微秒）	查询并插入时间（单位：微秒）	全部插完后再查询时间（单位：微秒）
10	34742593	15632846	34547932
100	3497778	1116397	3487212
1000	484152	280649	482715
10000	174795	252472	185054

[0068] 基于 Bloom Filter 算法的哈希函数个数，kcrc32 = 1，kmd5 = 4，ksha1 = 5，选择三种编码的组合，则 k 就是相应 kx 值相加，可以组合出多种不同的 k 值，k 的最大值为 10。在本实验中，Bloom Filter 算法的 m 值分别 80000 和 8000000，n = 75012，k 根据不同的编码组合，取同的值。实验结果如表 3 所示。

[0069] 表 3Bloom Filter 算法的 URL 去重方法实验结果

[0070]

M	N	k	编码组合	误判 URL 数	第 1 次时间(5 次均值，单位：微秒)	第 2 次时间(5 次均值，单位：微秒)
80000	75012	4	md5	39978	292133.2	290936.0
	75012	5	crc32、md5	35892	326508.6	325907.2
	75012	5	sha1	35761	427177.2	430732.8
	75012	9	md5、sha1	25139	592211.4	591152.0
	75012	10	crc32、md5、sha1	23439	644839.8	640756.4
8000000	75012	4	md5	0	293967.8	290402.0
	75012	5	crc32、md5	0	347169.5	341385.5
	75012	5	sha1	0	451908.8	448344.8
	75012	9	md5、sha1	0	632018.0	625505.8
	75012	10	crc32、md5、sha1	0	684549.8	672470.2

[0071] 4) 实验结果对比分析

[0072] 采用基于位图法哈希表的 URL 去重方法，在哈希表大小为 10000 时，查询并插入 75012 条记录的时间是 252472 微秒，全部插入后再查询的时间是 185054 微秒。而基于共享内存映射的 URL 去重方法加入打开文件的时间为 787788.5 微秒，之后的再查询时间为 167850 微秒（以存入 256 个文件为例），可见，本发明的方法性能优于基于位图法的 URL 去重。对于 Bloom Filter 算法实现的 URL 去重方法，取 Bloom Filter 算法最快的 k 等于 4 即只选 MD5 编码的方案时间是最短的，但也高达 290936.0 微秒，仍然比另外两种方法慢。所以基于 Bloom Filter 算法实现的 URL 去重方法，时间主要用在了哈希值的生成上，k 值越大，计算的时间越大，性能也就越差。另外，由于使用位图法的 URL 去重方法，会随着 URL 记录数的增大而性能降低，而基于 Bloom Filter 算法的 URL 去重方法会随着 URL 记录数的增大而造成误判率的提高，基于共享内存映射的 URL 去重方法却不存在上述的任意一个问题。另外，当程序退出后再次运行时，由于基于共享内存映射的 URL 去重方法的记录存在文件中，再次去重时以此文件中的记录为准，而对于另外两种 URL 去重方法，数据全部存在内存中，程序退出后再次运行时，原有记录全部消失，需要重新开始进行去重。

[0073] 综合上面的实验结果，基于共享内存映射的 URL 去重方法是最优的方案，达到了本发明的目的。本实例说明无论是性能还是正确性方面，本发明都适用于搜索引擎、网络审计系统中的 URL 去重判断。

[0074] 以上所述，仅是本发明的较佳实例而已，并非对本发明任何形式上的限制，虽然本发明以较佳实例的图示方法描述，然而并非用以限定本发明，任何熟悉本专业的技术人员，在不脱离本发明技术方案的范围内，都可利用上述描述的方法及技术内容作出部分的改变和调整，调整之后均为等同调整的案例描述，但凡是未脱离本发明技术的内容，依据本发明的技术实质对以上描述案例所作的任何简单修改与调整，均仍属于本发明技术方案的范围内。

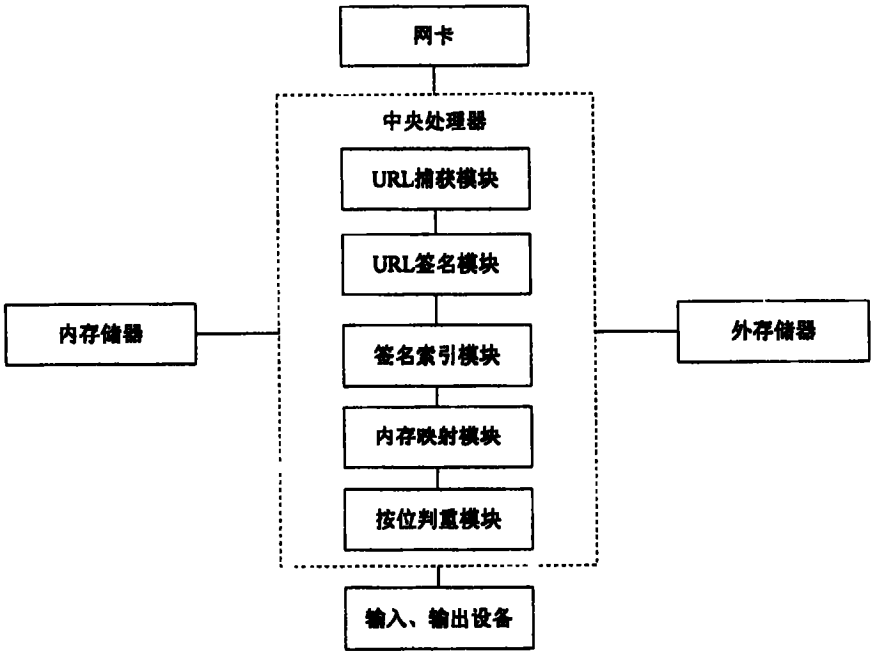

网卡
中央处理器
URL捕获模块
URL签名模块
签名索引模块
内存映射模块
按位判重模块
内存储器
外存储器
输入、输出设备

图 1

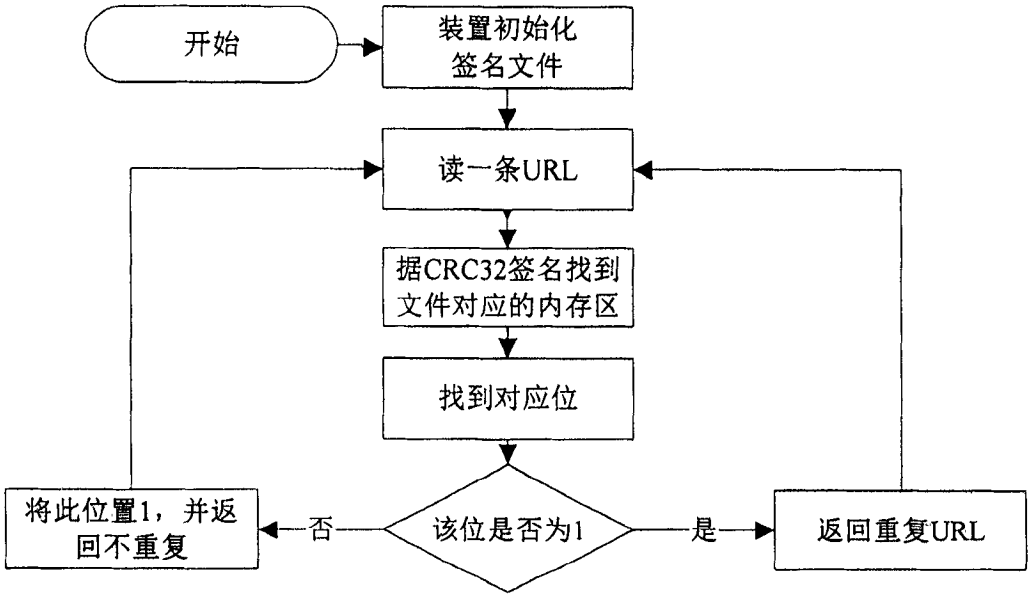

开始
装置初始化
签名文件
读一条URL
据CRC32签名找到
文件对应的内存区
找到对应位
该位是否为1
否
是
将此位置1，并返
回不重复
返回重复URL

图 2

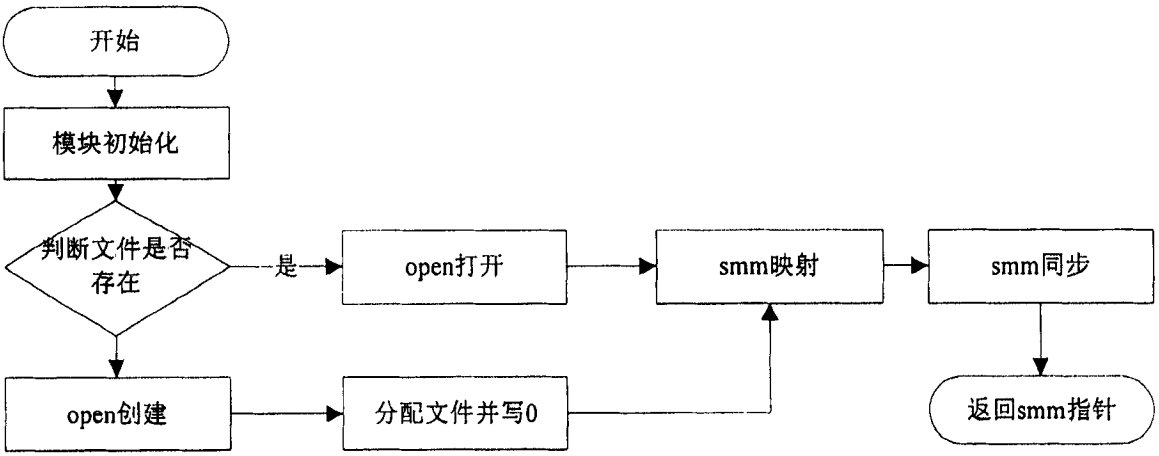

开始
模块初始化
判断文件是否存在
是
open打开
smm映射
smm同步
open创建
分配文件并写0
返回smm指针

图 3

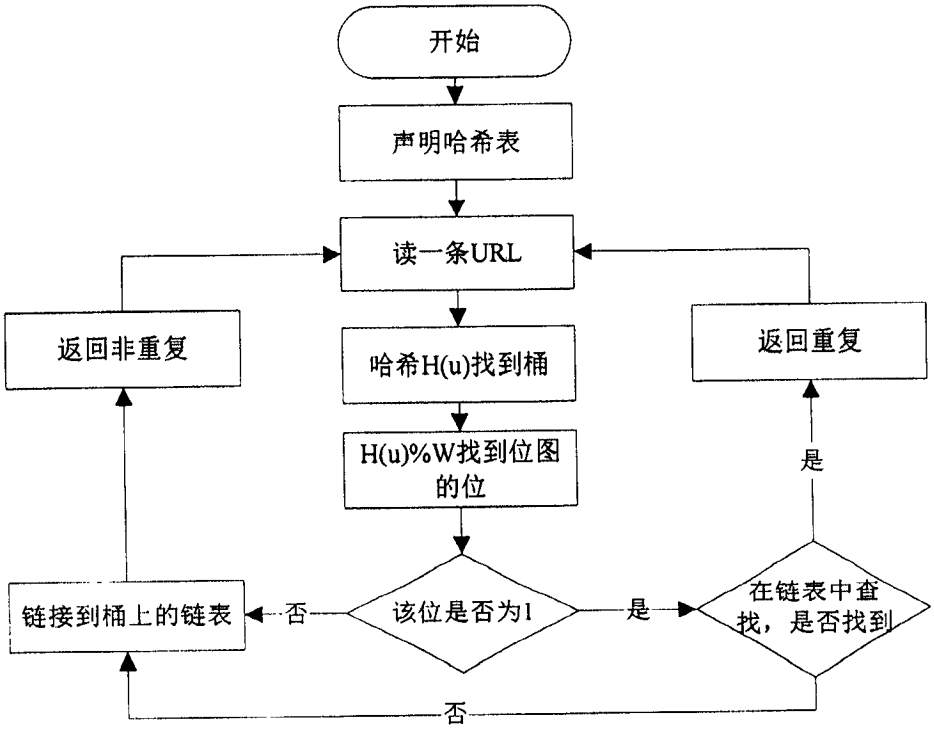

开始
声明哈希表
读一条URL
返回非重复
哈希H(u)找到桶
返回重复
H(u)%W找到位图的位
是
链接到桶上的链表
否
该位是否为1
是
在链表中查找，是否找到
否

图 4