基于 Counting Bloom Filter 的海量网页快速去重研究

吴家奇,刘年国,李 雪,谢 翔,王 涛
(国网淮南供电公司,安徽 淮南 232007)

摘 要:网页去重是从给定的大量的数据集合中检测出冗余的网页,然后将冗余的网页从该数据集合中去除的过程,可以有效地减少检索和存储的压力。其中基于同源网页的 URL 去重方法、基于网页结构和特征的抽取指纹方法和基于网页内容的聚类方法的研究都已经取得了很大的发展,但是针对海量网页去重问题,上述三种方法,目前还是很难解决网页去重的时间和空间问题,本文在基于 MD5 指纹库网页去重算法的基础上,结合 Counting Bloom filter 算法的特性,提出一个节省空间的大规模数据表示和快速去重策略,实现了一种快速去重算法 IMP-CM Filter,大大降低了网页去重算法的时间复杂度和空间复杂度。该算法通过减少 I/O 频繁操作,来提高海量网页去重的效率。最后通过实验表明,IMP-CM Filter 算法的有效性。

关键词:网页去重;MD5 指纹库;Counting Bloom filter;IMP-CM Filter 算法
文章编号:2096 - 4633(2018)12 - 0037 - 06 **中图分类号**:TM744 **文献标志码**:B

随着网络技术和电力信息化业务的的发展,网络信息膨胀,搜索引擎已成为人们从互联网获取信息的重要手段,与此同时,人们对搜索引擎的质量要求越来越高。然后现在大量的重复网页充斥着互联网,将严重影响网页检索效率。据统计数据表明,近似网页的比例占全部网页的 29%。清华大学 IT 可用性实验室对 Google、Baidu 搜索引擎的研究表明,Google 和 Baidu 的重复网页占全部网页的比率分别为 3.4% 和 2.1%[1]。而针对国家电网相关业务网页进行统计,发现重复网页占全部网页的比率约为 7.8%

与此同时,网页去重又能带来很多好处。首先,重复网页的去除能够节省存储空间,并且可以提高检索效率;其次,对重复网页进行分析,可以在网页下载时预先发现重复网页,提高下载效率和准确度;最后,更少的重复资源可以提高搜索引擎的整体质量,提高可用度,为用户提供方便。

综上所述,在海量网页中快速消除重复的网页已经成为信息搜索中的研究重点。文献[2]指出网页去重的方法有三类:基于网页结构和特征的抽取指纹信息方法、基于网页内容的聚类方法、基于同源网页的 URL 方法。本文只针对基于同源网页的 URL 去重的研究。为了实现海量网页的快速去重,该文在基于 MD5 指纹库的网页去重算法的基础上,利用 Counting Bloom Filter 算法,提出一个节省空间的大规模数据表示和快速去重策略,以应对海量网页去重的需求。在不影响去重效率的前提下,提出了 IMP-CB Filter 算法,解决了 Counting Bloom Filter 在处理海量网页去重时所需要解决的计数器值溢出和更新问题。

1 相关研究

1.1 基于同源网页的 URL 去重技术。

文献[3 - 4]提出了基于 Hash 算法的网页去重,需要维护一个 Hash 表,如果 Hash 函数设计的不好,在进行映射的时候,发生碰撞的几率很大,此外使用的是 URL 作为键,URL 字符串也占用了很大的存储空间。当 URL 重复率较高的时候,需要进行较多的字符串匹配操作,这将严重影响 URL 的检索速率。MD5 指纹的网页去重算法是 Hash 算法去重的一种,可以对 URL 字符串进行压缩,得到一个压缩字符串,并且 MD5 进行 Hash 映射碰撞的几率非常小,但是它需要维持一个 MD5 指纹库,对于海量的网页(一份报告指出,截止 2005 年 1 月,网页数量至少达到 115 亿[5])的去重处理,将会产生频繁地 I/O 操作瓶颈,严重影响网页去重的速率。

文献[6 - 7]提出基于查找树的网页 URL 去重算法,也可以节约存储空间,用于 URL 检索时,检索算法的时间复杂度是 O(n),其中 n 是树的层数,但是对于海量较长的 URL,这也需要很大的空间,一

般内存空间是无法承受的。而且每次检索都需要做很长的字符比较,使得检索速率较慢。

文献[8]提出基于 Rabin 指纹方法的 URL 检索算法,该算法将通过 Rabin 算法计算得到的指纹映射成 16 进制数字成的字符串,再把此字符串存储在一颗键树中,然后利用键树对 URL 检索,该算法可以有效减少了存储空间。但是检索效率不是太理想。

由上述所知,我们需要一种节省存储空间的数据表示和快速判重的解决方案。本文在基于 MD5 指纹库的网页去重算法的基础上,结合 Counting Bloom filter 算法,提出了满足上述需求的网页去重算法 IMP-CB Filter。

1.2 Counting Bloom Filter 算法

Counting Bloom Filter 算法是为了支持删除操作而由标准 Bloom Filter 算法改进而来的。标准 Bloom Filter 算法是 1970 年由布隆提出的,它实际上是由一个很长的二进制数组和一系列随机散列函数构成[9-11]。标准 Bloom Filter 算法的主要作用是判断一个集合中是否存在某个元素,它的空间效率和时间效率都比较好,比较适合海量数据集的表示和检测,缺点是有一定的误判率和不支持删除操作。Counting Bloom Filter 算法和标准 Bloom Filter 算法不同主要在位数组上,标准 Bloom Filter 算法是对每一 bit 操作,但是 Counting Bloom Filter 算法是将位数组的每个 bit 扩展为多个 bit 来表示一个计数器。在插入元素的时候,通过对计数器的值进行加 1 操作来代替置 1 操作;在删除元素的时候,不是进行置 0,而是在计数器的值上减 1,这样就实现了删除操作。

Counting Bloom Filter 算法也具有很好的时间和空间效率,利用这个特性可以解决海量网页去重的频繁 I/O 操作瓶颈问题,从而提高海量网页去重效率。此外,Counting Bloom Filter 算法还支持删除操作,因此它支持对网页删除操作。虽然 Counting Bloom Filter 算法可以提高海量网页去重效率和支持对网页的删除操作,但是它并没有解决标准 Bloom Filter 算法存在的误判率问题,这会导致用户进行网页去重时,网页去重机制会误判 MD5 指纹库中已经存在该网页。本文将结合 Counting Bloom Filter 算法和 MD5 指纹库检测技术,来研究海量网页去重技术,从而提高网页去重效率,其中涉及 Counting Bloom Filter 算法的计数器大小分析,Counting Bloom Filter 算法的计数器溢出问题,网页删除时引起的 Counting Bloom Filter 算法的计数器值更新问题。

2 基于 Counting Bloom Filter 的海量网页去重技术

2.1 基于 MD5 指纹库的网页去重过程

基于 MD5 指纹库的网页去重技术主要是对网页 URL 进行 MD5 指纹处理,然后根据 MD5 指纹库,来进行下一步处理,具体流程如图 1 所示。

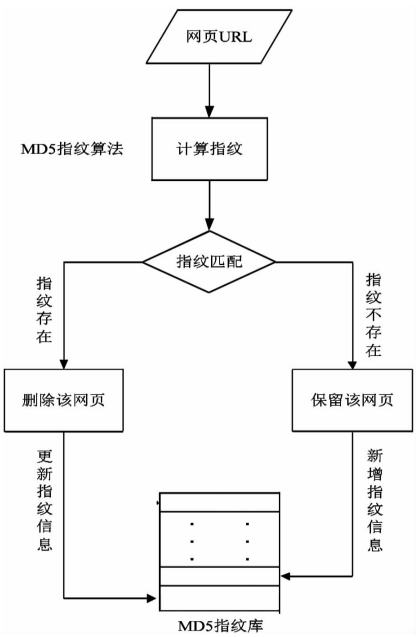

图 1 基于指纹库的网页去重流程图

Fig. 1 The flow chart of deduplication based on web-based fingerprint

首先,通过 MD5 指纹算法计算即将被处理的网页 URL 的指纹值 MD5-value,然后根据指纹值 MD5-value 查询 MD5 指纹库,如果 MD5 指纹库中不存在指纹值 MD5-value,说明该网页不是重复网页,则将该网页保留下来,并将对应的指纹值 MD5-value 写入 MD5 指纹库中;如果 MD5 指纹库中存在指纹值 MD5-value,说明已经存在相同的网页,则将该网页删除,并更新指纹库信息,使指纹值 MD5-value 引用次数加 1。

2.2 Counting Bloom Filter 算法的计数器大小分析

上述基于 MD5 指纹库的网页去重过程直接对指纹库进行查询是否存在即将被处理的网页 URL 的指纹值 MD5-value,在网页数据量比较小时,指纹库的指纹数据信息可以完全存放到服务器内存中,查找速度比较快,但是随着不断地网页 URL 被检

测,指纹库中的指纹数据信息也会不断增加,会超过服务器内存的大小,这样在指纹库中进行查询指纹 MD5-value 时,会频繁地进行 I/0 操作,以致于影响海量网页去重的效率,可以利用 Counting Bloom Filter 算法的特性来解决这个问题。虽然 Counting Bloom Filter 算法可以支持文件的删除操作,但是 Counting Bloom Filter 算法是以使用更多内存空间的代价换取支持删除操作。本文接下来探讨 Counting Bloom Filter 算法需要设置多大的计数器来满足一般需要的删除操作。

令集合元素个数为 N 个,Counting Bloom Filter 算法所需哈希函数[13-14]个数为 k 个,每个计数器占 n 个 bit,计数器个数为 m。假设第 i 个计数器被增加 j 次,那么它的概率为:

$$P(c(i)=j)=\binom{Nk}{j}(\frac{1}{m})^{j}(1-\frac{1}{m})^{NK-j}$$

其中 Nk 表示 N 个集合元素需要进行 Nk 次哈希函数计算,$\binom{Nk}{j}$表示从 N 次哈希函数计算中选择 j 次;每个计数器被选取的概率为$\frac{1}{m}$,所以$(\frac{1}{m})^{j}$表示选取的 j 次哈希函数计算值都映射到第 i 个计数器的概率;$(1-\frac{1}{m})$表示 1 次哈希函数计算值没有映射到第 i 个计数器的概率,因此$(1-\frac{1}{m})^{NK-j}$表示 NK-j 次哈希函数值没有映射到第 i 个计数器的概率。那么第 i 个计数器的值不小于 j 的概率为 $P(c(i)\geqslant j)\leqslant\binom{Nk}{j}(\frac{1}{m})^{j}(1-\frac{1}{m})^{NK-j}$

根据 Counting Bloom Filter 算法相关技术部分中[15],得到哈希函数个数 k 为$(\ln 2)\frac{m}{N}$可以得到最优解的结论,现在限定 k 不大于$(\ln 2)\frac{m}{N}$,可以得到下面结果:

$$P(max_i\, c(i)\geqslant j)\leqslant m(\frac{e\ln 2}{j})^{j}$$

如果每个计数器占用的 bit 个数 n 为 4 时,那么计数器最大可以表示的数字为 15,当大于 15 时就溢出。这个概率为:

$$P(max_i\, c(i)\geqslant 16)\leqslant 1.37\times 10^{15}\times m$$

由上面的结果可知,这个概率已经很小可以应用到大部分应用场景。

2.3　基于 IMP-CB Filter 算法的海量网页去重研究

由于 MD5 指纹库中存储海量 MD5 指纹值,如果每次检测网页是否已经存在都需要查询 MD5 指纹库中是否存在和即将进行去重处理的网页一样的指纹,并且服务器的内存是一定的,所以会产生频繁地 I/O 操作,严重影响网页去重的效率。为了解决这个问题,本文将在海量网页去重问题上使用 Counting Bloom Filter 算法。虽然 Counting Bloom Filter 算法可以支持删除操作,但是还是存在误判率问题。本文结合 Counting Bloom Filter 算法的可删除特性和指纹库的无误判率特性,提出了针对海量网页 URL 去重技术——IMP-CB Filter 算法,从而提高海量网页去重的效率。但是根据 2.2 节对 Counting Bloom Filter 算法的计数器大小的探讨可知,Counting Bloom Filter 算法存在小概率的计数器溢出问题,为了解决这个问题,本文对计数的值进行如下处理。

假设每个计数器占用 n 个 bit,这样可以表示整数的范围为 0 到 2^n-1。当一个网页 URL 的指纹 MD5-value 经过 Counting Bloom Filter 算法的哈希函数计算映射到第 i 个计数器 CountI。如果 CountI 的值为 0 到 2^n-2 时,对 CountI 的值加 1;如果 CountI 的值为 2^n-1 时,CountI 的值保持不变。由这个规定可知,对于每个计数器的值 value,当 value 的范围是 0 到 2^n-2 时,说明已经至多有 value 个网页 URL 的指纹 MD5-value 经过 Counting Bloom Filter 算法的哈希函数计算映射到这个计数器上;当 value 为 2^n-1,说明可能已经至少有 2^n-1 个网页 URL 的指纹 MD5-value 经过 Counting Bloom Filter 算法的哈希函数计算映射到这个计数器上。

虽然上述处理方法解决了 Counting Bloom Filter 算法的计数器溢出问题,但是进行网页删除操作时,如果被删除网页 URL 的指纹 MD5-value 经过 Counting Bloom Filter 算法的哈希函数计算映射到计数器 CountI 值 value 为 2^n-1 时,此时无法对 value 进行正确处理。为了解决这个问题,本文解决方法如下:

当进行网页去重处理时,如果通过 Counting Bloom Fiter 算法或者指纹库判定该网页为未出现的网页时,则将该网页 URL 的指纹值 MD5-value 对应的哈希值(计数器编号)添加到指纹库中。经过这样处理后,进行网页删除操作时,一旦遇到计数器的值 value 为 2^n-1 时,只需要统计指纹库中该计数器

的个数,并对计数器值 value 进行正确更新。

通过解决上述 Counting Bloom Filter 算法在处理海量网页去重时面临的问题后,接下来将要阐述网页去重技术的具体过程。具体流程如图 2 所示。

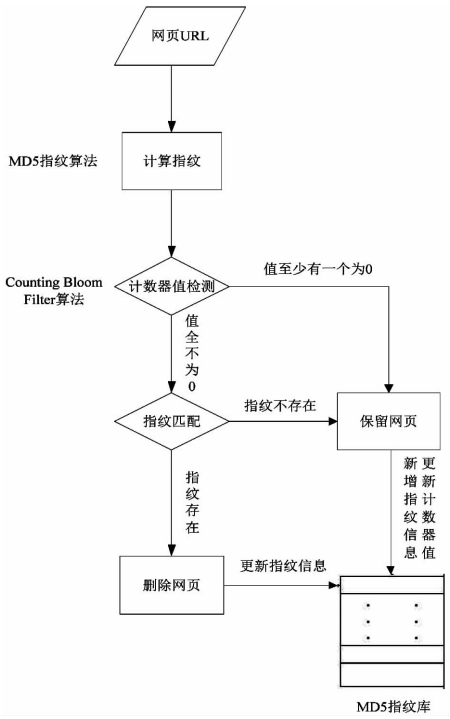

图 2 基于 IMP-CB Filter 算法的海量网页去重流程图

Fig. 2 The flow chart of deduplication based on web-based IMP-CM Filter

假设 Counting Bloom Filter 算法所需哈希函数个数为 k 个,每个计数器占 n 个 bit,计数器个数为 m。首先通过 MD5 指纹算法计算网页 URL 的指纹值 MD5-value,然后通过 Counting Bloom Filter 算法的 k 个哈希函数计算指纹 MD5-value 的值,并记为 $hash_1, hash_2, \cdots\cdots, hash_k$,并取得第 $hash_1$ 计数器,第 $hash_2$ 计数器,……,$hash_k$ 计数器的值,记为 $M_1, M_2, M_3, \cdots\cdots, M_K$,然后判断 $M_1, M_2, M_3, \cdots\cdots, M_K$ 中的值是否都不为 0。

如果 $M_1, M_2, M_3, \cdots\cdots, M_K$ 中的值至少有一个为 0 时,可以确定 MD5 指纹库中不存在和 MD5-value 一样的网页,则将网页保留下来,并分别对 $M_1, M_2, M_3, \cdots\cdots, M_K$ 中的不为 $2^n - 1$ 的值进行加 1 操作。因为网页集中第一次存在该网页,所以该网页被引用次数 Count 为 1。最后,将网页 URL 的指纹 MD5-value,引用次数 Count = 1 以及 $hash_1, hash_2, \cdots\cdots, hash_k$ 添加到 MD5 指纹库中。

如果 $M_1, M_2, M_3, \cdots\cdots, M_K$ 中的值都不为 0 时,由于误判率的存在,不能判断 MD5 指纹库中是否存在和 MD5-value 一样的网页,则在 MD5 指纹库查找是否已经存在指纹 MD5-value。如果 MD5 指纹库中存在和指纹 MD5-value 一样的指纹,说明网页集中已经存在相同的网页,则不需要保留该网页,只需要将 MD5 指纹库中相应指纹 MD5-value 的引用次数 Count 加 1;如果 MD5 指纹库中不存在指纹 MD5-value 时,说明 MD5 指纹库中不存在和 MD5-value 一样的网页,则将网页保留下来,并分别对 $M_1, M_2, M_3, \cdots\cdots, M_K$ 中的不为 $2^n - 1$ 的值进行加 1 操作。因为网页集中第一次存在该网页,所以该网页被引用次数 Count 为 1。最后,将网页 URL 的指纹 MD5-value,引用次数 Count = 1 以及 $hash_1, hash_2, \cdots\cdots, hash_k$ 添加到 MD5 指纹库中,并且分别对 $M_1, M_2, M_3, \cdots\cdots, M_K$ 中的不为 $2^n - 1$ 的值进行加 1 操作。

上述海量网页去重 IMP-CB Filter 算法不仅可以提高海量网页去重的效率,还保证了 Counting Bloom Filter 的计数器不是多次记录被引用的网页,而是只记录一次被引用的网页,缓解了计数器的溢出,也进一步提高了海量网页去重的效率。

3 实验结果与分析

本实验数据来源于数据堂。根据文献[12]选取 Counting Bloom Filter 参数 k 和 m/n,分别为 8 和 20。在不同的数据量下(500 000,1 000 000,1 500 000,2 000 000)进行测试实验并进行分析,具体过程如下:分别在上述四组数据量下,测试并记录基于 MD5 指纹库的网页去重处理时间和基于标准 IMP-CM Filter 算法的网页去重处理时间。根据上述测试方案,得到的测试结果如图 3 所示。

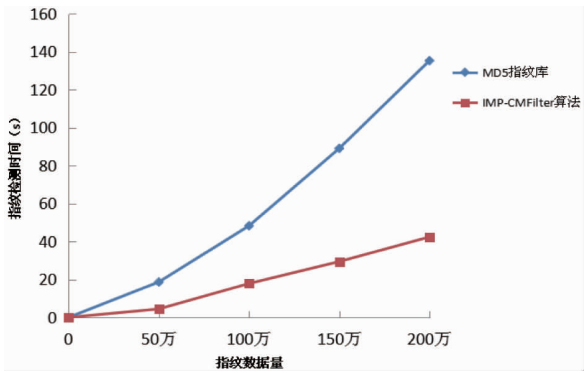

图 3 网页去重技术数据对比图

Fig. 3 The comparison of web page deduplication technologies

由图 3 可知,随着网页数据量的增加,MD5 指

纹库查询去重技术指纹检测时间延长越严重,这是因为主要遇到 I/O 操作瓶颈,而本文提出 IMP-CM Filter 算法在进行指纹 MD5-value 检测时利用 Counting Bloom Filter 算法减少了 I/O 操作,从而提高了指纹 MD5-value 检测的速率,最终提高海量网页去重的效率。

4　总结与展望

针对加快海量网页去重的需求[16-17],本文在基于 MD5 指纹库的基础上,结合 Counting Bloom Filter 算法的特性,提出 IMP-CM Filter 算法。并且分析和解决了其中的关键技术问题,主要包括 Counting Bloom Filter 算法的计数器溢出问题,Counting Bloom Filter 算法的计数器大小问题,删除操作时引起 Counting Bloom Filter 算法的计数器值更新问题。在此基础上,详细阐述了海量网页去重过程。最后通过进一步的模拟实验,证明了本文提出的 IMP-CM Filter 算法的有效性。

参考文献:

[1] 阎亚杰. 网页去重方法研究[J]. 电脑开发与应用,2008,21(08):60-62.
YAN Yajie. Study on method on deletion of duplicated web pages [J]. Computer Development & Application, 2008, 21 (08): 60-62.

[2] LI ZHIYI, LIANG SHIJIN. National research on deleting duplicated web pages: status and summary [J]. Library and Information Service. 2010, (07): 118-121.

[3] NAM GW, PARK JH, KIM TY. Dynamic management of URL based on object-oriented paradigm [C]// Proceedings of the International Conference on Parallel and Distributed Systems. Taiwan, China: IEEE Computer Society Press, 1998: 226-230.

[4] M NAJORK, A HEYDON. High-performance web crawling[M]. Handbook of Massive Data Sets, Kluwer Academic Publishers Inc, 2001: 25~45.

[5] Gulli A, Signorini A. The indexable web is more than 11.5 billion pages [C]//Special Interest Tracks and Posters of the 14th International Conference on World Wide Web WWW. 05. ACM Press 2005: 902-903.

[6] YE YUNMING, YU SHUI, MA FANYUAN, et al. On distributed web crawler: architecture, algorithms and strategy [J]. China Academic Journal Electronic Publishing House. 2002, 30(S1): 2008-2011.

[7] Kasom Koht-arsa, Surasak Sanguanpong. In-memory URL compression [J]. Chiangmai: National Computer Science and Engineering Conference(NCSEC-2001), 2001: 425~428.

[8] JIANG ZL, ZHAO Q, XIAO H, et al. High performance parallel crawler [J]. Computer Engineering & Design, 2006, 27(24): 4762-4766.

[9] Bloom B. Space-time Tradeoffs in Hash Coding with Allowable Errors [J]. Communication of the ACM, 1970, 13(7): 422-426.

[10] NASRE R, RAJAN K, GOVINDARAJAN R, et al. Scalable context-sensitive points-to analysis using multi-dimensional bloom filters [C]//Programming Languages& Systems. Asian Symposium, Aplas, Seoul, Korea, . 2009, 5904: 47-62.

[11] LIU W, GUO YB, HUANG P. Pattern matching engine based on multi-dimensional bloom filters [J]. Journal Of Computer Applications. 2011, 31(01): 107-114.

[12] PEI Cao. Bloom Filters-the math[EB/DL]. http://pages.cs.wisc.edu/~cao/papers/summary-cache/node8.html.

[13] GAO K, WANG YC, XIAO J. The strategy on processing replicated web collections[J]. Journal of Shanghai Jiaotong University. 2006, 5(05): 775-778.

[14] 吴丽辉,白硕,张刚,等. Web 信息采集中的哈希函数比较[J]. 小型微型计算机系统,2006,27(04):673-676.
WU Lihui, BAI Shuo, ZHANG Gang, et al. Hashing comparison in Web crawling [J]. Mini-micro Systems, 2006, 27 (04): 673-676.

[15] 肖明忠,代亚非. Bloom Filter 及其应用综述[J]. 计算机科学,2004,31(04):180-183.
XIAO Mingzhong, DAI Yafei. A survey on Bloom Filters and its applications[J]. Computer Science, 2004, 31(04): 180-183.

[16] 鄢斌,陈宾,杨春麟,等. 浅谈大数据管理在基层供电企业的应用与发展[J]. 电力大数据,2018,21(01):32-34.
YAN Bin, CHEN Bin, YANG Chunlin, et al. Talking about the application and development of big data management in grassroots power supply enterprises[J]. Power Systems and Big Data, 2018, 21(01): 32-34.

[17] 张嵩,刘洋,许芳,等. 配电网中大数据的挖掘应用[J]. 电力大数据,2018,21(02):8-12.
ZHANG Song, LIU Yang, XU Fang, et al. Application of big data mining in power distribution network[J]. Power Systems and Big Data, 2018, 21(02): 8-12.

收稿日期:2018-10-19

作者简介:

吴家奇(1988),男,硕士,工程师。主要从事电力网络信息安全、信息化建设的研究工作。

(本文责任编辑:王　燕)

Research of massive web rapidly filter base on Counting Bloom Filter

WU Jiaqi, LIU Nianguo, LI Xue, XIE Xiang, WANG Tao

(State Grid Huainan Power Supply Company, Huainan 232007 Anhui, China)

Abstract: Web deduplication is a process which detected duplicate content pages from a given amount of data collection, and then removed from the copy of the collection, It can effectively reduced the pressure of retrieval and storage. Which research of web deduplication based on the URL filter、the structure and characteristics 、the contents and homology, has achieved great development, But it is no good solution to the problem of running time and stored space in the massive web pages filter. Based on web-based MD5 fingerprint deduplication algorithm, and using Counting Bloom filter algorithm, this essay proposed a menthod that is space-saving large-scale data and fast de-duplication, and implemented a algorithm for rapidly deduplication called IMP-CM Filter, which could improve the efficiency of mass web pages filter by reducing the frequent operation of I/O. On the fact that the IMP-CMFilter algorithm had higher performance.

Key words: web pages deduplication; MD5 fingerprint database; Counting Bloom Filter; IMP-CM Filter algorithm;