ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.5, May 2006, pp.1051−1067
DOI: 10.1360/jos171051

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

网页变化与增量搜集技术*

孟 涛+, 王继民, 闫宏飞

(北京大学 计算机科学技术系 网络与分布式系统实验室,北京 100871)

Web Evolution and Incremental Crawling

MENG Tao+, WANG Ji-Min, YAN Hong-Fei

(Laboratory of Computer Networks and Distributed System, Department of Computer Science and Technology, Peking University, Beijing 100871, China)

+ Corresponding author: Phn: +86-10-62758485 ext 23, E-mail: mengtao@net.pku.edu.cn, http://net.pku.edu.cn/~mengtao/

Meng T, Wang JM, Yan HF. Web evolution and incremental crawling. *Journal of Software*, 2006,17(5): 1051−1067. http://www.jos.org.cn/1000-9825/17/1051.htm

Abstract: With the massive and ever increasing pages in the Web, incremental crawling has become a promising method to achieve on-line information. Its main advantage is the resource economization, which comes from the avoidance of downloading unchanged pages. For the precision of change prediction, the evolution of Web is generally studied to find out how pages change. In sum, incremental crawlers often integrate change frequency, change extent, and document quality for each page to determine its relative order as well as its download frequency. In this paper, the researches on Web evolution and incremental crawling in recent years are summarized: First, the change of page is modeled as a Poisson process, and the solutions are given to estimate its parameters, especially the change frequency, and then experimental results are shown. Second, based on the change of pages, three public large-scale incremental crawling systems are introduced, with emphasis on their scheduling policies and strategies to enhance page qualities. Third, theoretical analysis and exploration are performed to find the optimal scheduling policy, three approaches from different points of views are utilized to achieve this object, and a heuristic approximate solution is supplied for the feasibility in practice. Finally, research trends in this area are predicted, and three main issues are listed.

Key words: Web evolution; incremental crawling; scheduling policy; research development

摘 要: 互联网络中信息量的快速增长使得增量搜集技术成为网上信息获取的一种有效手段,它可以避免因重复搜集未曾变化的网页而带来的时间和资源上的浪费.网页变化规律的发现和利用是增量搜集技术的一个关键.它用来预测网页的下次变化时间甚至变化程度;在此基础上,增量搜集系统还需要考虑网页的变化频率、变化程度和重要性,选择一种最优的任务调度算法来决定不同网页的搜集频率和相对搜集次序.针对网页变化和增量搜集技术这一主题,对最近几年的研究成果作总结,并介绍最新的研究进展.首先论述对网页变化规律的

* Supported by the National Natural Science Foundation of China under Grant Nos.60573166, 60435020 (国家自然科学基金); the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No.20030001076 (国家教育部博士点基金)

Received 2005-10-11; Accepted 2006-01-12

建模、模型参数估计和估计效率等问题;然后介绍几个著名的增量搜集系统,着重分析它们的任务调度算法;最后,从理论上分析和总结增量搜集系统的最佳任务调度算法及其一个基于启发式策略的近似解,并预测其将来的研究趋势.该工作对增量搜集系统的设计和 Web 演化规律的研究具有参考意义.

关键词: 网页变化;增量搜集;调度策略;研究进展
中图法分类号: TP393 文献标识码: A

World Wide Web 中网页信息量的指数速度增长给诸如搜索引擎之类的网络应用系统的信息搜集带来了巨大的压力:一方面,网页数量太大,应尽量全面地予以覆盖;另一方面,硬件资源受限,要优先考虑有价值的网页加以搜集.在此背景下,增量搜集系统的产生使得它与采取大规模周期性的遍历方法来搜集和维护信息相比具有更高的效率和时新性.它采取持续的调度策略,避免了因重复搜集未变化的网页带来的资源和时间浪费.

增量搜集系统的工作分为两大部分:一个模块负责根据 URL 列表下载并检查网页是否变化或未曾搜集,并存储其内容,称为下载模块;另一个模块负责预测网页的变化时间并决定搜集检查的频率和次序,然后生成 URL 列表供搜集,称为调度模块.关于下载模块已经有大量成熟的研究工作,例如文献[1–7]中分别介绍的搜集系统.它们都是通过网页间的链接关系来遍历 Web 图[8–10]的成功系统典型.尽管这些工作基本上没有考虑网页的变化带给搜集系统的影响,但它们在系统架构和搜集算法上已经研究得很透彻.除此之外,对诸如噪音消除、URL 消重、镜像发现、节点任务分配、遍历策略和并行搜集等搜集系统的技术细节,已经有大量的研究工作[11–16].

本文的工作主要涉及调度模块.它是增量搜集系统的核心.调度模块工作的理论依据是网页的变化规律和以此为基础的最优化调度策略.网页的变化通常被视为泊松过程,据此可以估计网页的变化周期和下次变化时间.调度策略包括两方面的因素:一方面是网页的搜集频率,由此可计算对每个网页维持一定时新性的平均检查周期;另一方面是网页的搜集先后次序,它对应网页的重要性.二者共同决定搜集系统维护的网页集合的重要性和时新性.在资源允许前提下,通常应尽量提高根据网页的重要性加权后的平均时新性.

本文是对网页变化和增量搜集技术的一个综述.本文第 1 节阐述网页变化的研究进展.第 2 节介绍在网页变化规律已知的基础上,如何实现一个增量搜集系统.第 3 节从理论上分析增量搜集的最优调度策略,包括该调度算法的效率问题和它的一种基于启发式策略的近似解决方法.最后是总结,并预测增量搜集技术的研究趋势.

1 网页变化

对网页变化规律的研究目前主要有两种方法:一种方法是基于实验手段对 Web 中的网页采样,通过搜集和检查来研究样本的变化规律,从而估计整个 Web 的变化规律,如文献[17–24];另一种方法试图从理论上给网页的变化建立数学模型,然后进行分析和论证,并用实验来验证该模型并估计相关参数,以此预测网页的下次变化时间,如文献[25–29].本文将这些工作归纳为以下 5 个主题,分别予以介绍.

1.1 基本模型

尽管某些实验结果[30]显示并非完全如此,但在当前的研究中,网页的变化一般被视为泊松过程[17,18,21,22,31–33]:从某个时刻 0 开始,用 $X(t)$记某个网页在时刻 t 变化的次数,网页的每次变化都是独立同分布的,且变化频率是λ,根据泊松过程的定义,

$$\Pr\{X(s+t)-X(s)=k\}=\frac{(\lambda t)^k}{k!}\mathrm{e}^{-\lambda t},k=0,1,... \tag{1}$$

假定网页下次变化的时刻是 T,则 T 的概率密度函数为

$$f_T(t)=\begin{cases}\lambda\mathrm{e}^{-\lambda t}, & t>0\\ 0, & t\le 0\end{cases} \tag{2}$$

当 k=1 时,由式(1)可知网页两次变化的间隔时间服从指数分布,即式(2).对于搜集系统而言,如果它在某时刻存在本地的某网页的内容与当时该网页在 Web 中的实际内容相同,就称该网页是时新的.搜集系统所维护的

某个网页 e_i 的时新性可以定义如下[25,34]:

$$F(e_i;t)=\begin{cases}1, & \text{if } e_i \text{ is up-to-date at time } t\\ 0, & 0 \text{ otherwise}\end{cases} \tag{3}$$

还有另外一种方式用来定义网页 e_i 的时新程度,那就是网页的年龄[25],定义如下

$$A(e_i;t)=\begin{cases}0, & \text{if } e_i \text{ is up-to-date at time } t\\ t-LMT(e_i), & \text{otherwise}\end{cases} \tag{4}$$

其中,$LMT(e_i)$是指网页 e_i 在 t 之前的最后更新时间.根据式(3)和式(4)的定义,可以进一步定义由 N 个网页组成的集合 S 的平均时新性和年龄如下

$$F(S,t)=\frac{1}{N}\sum_{i=1}^{N}F(e_i,t),\ A(S,t)=\frac{1}{N}\sum_{i=1}^{N}A(e_i,t) \tag{5}$$

增量搜集系统维护的是一个网页集合 S,它所关注的是 S 在某段时间的平均时新性和平均年龄.此时,可以对式(5)在时间上取平均值加以衡量:

$$\overline{F(S)}=\lim_{t\to\infty}\frac{1}{t}\int_0^t F(S,t)\mathrm{d}t,\ \overline{A(S)}=\lim_{t\to\infty}\frac{1}{t}\int_0^t A(S,t)\mathrm{d}t \tag{6}$$

而对于单个网页而言,它在一段时间内的平均时新性以及平均年龄可以通过结合式(1)~式(4)来计算相应的期望值获得.

假定网页的平均变化频率为λ_i,则它在区间 I=(0,t)内发生变化的概率为

$$\Pr(T\le t)=\int_0^t f_T(t)\mathrm{d}t=\int_0^t\lambda_i\mathrm{e}^{-\lambda_i t}\mathrm{d}t=1-\mathrm{e}^{-\lambda_i t} \tag{7}$$

因此,网页在区间 I 内的平均时新性和平均年龄分别为

$$\left.\begin{aligned}E[F(e_i;t)]&=0\times\Pr\{T\le t\}+1\times(1-\Pr\{T\le t\})=\mathrm{e}^{-\lambda_i t}\\ E[A(e_i;t)]&=t\left(1-\frac{1-\mathrm{e}^{-\lambda_i t}}{\lambda_i t}\right)\end{aligned}\right\} \tag{8}$$

上述定义给网页的变化过程建立了基本的数学模型,并定义了评价增量搜集系统网页质量的时新性和年龄这两个重要指标.对于增量搜集系统所维护的网页集合 S,它在区间 I 内的平均时新性和平均年龄可以根据式(5)、式(6)和式(8)来推出.

1.2 变化估测

基于网页变化模型,需要通过搜集来估计其参数,以获得网页的变化频率并推算下次变化时间.然而,通常没有足够的时间和资源去获得 Web 上每个网页的所有变化时刻,因此要对变化频率进行估计.它一般有两种内涵:(1) 直接估算网页的变化频率;(2) 估计网页变化频率的类别.在增量搜集上,它们都有自己对应的应用场景.

对网页变化频率λ的直接估计,最简单的办法就是用变化的总次数 X 除以时刻 0~时刻 T 的变化间隔 T[17].它有 3 个性能评价准则:是否有偏差,即偏差期望是否为 0;是否具备一致性,即偏差是否会随着搜集检查次数增多而减小;是否有效率,即偏差的方差是否太大.如果用 r 表示λ与搜集检查频率 f 的比值,则 r 最简单的估计器为

$$\hat{r}=\frac{\hat{\lambda}}{f}=\frac{1}{f}\left(\frac{X}{T}\right)=\frac{X}{n} \tag{9}$$

其中,n 是该时间间隔内搜集检查的次数.这种估计方法的期望和方差分别是

$$E[\hat{r}]=\sum_{m=0}^{n}\frac{m}{n}\Pr\left\{\hat{r}=\frac{m}{n}\right\}=\sum_{m=0}^{n}\frac{m}{n}\times\binom{n}{m}(1-q)^m q^{n-m}=1-\mathrm{e}^{-r} \tag{10}$$

$$V[\hat{r}]=E[\hat{r}^2]-E[\hat{r}]^2=\mathrm{e}^{-r}(1-\mathrm{e}^{-r})/n \tag{11}$$

据此可以判断,式(9)的估计方法是有偏差的($r/(1-e^{-r})$只是在 r 很小时才趋于 1),而且效率随着搜集次数的增大而提高.它最大的缺点是不具备一致性,偏差与 n 无关,因为通常都希望搜集次数越多,估测误差越小.

为了达到一致性,文献[29]提出了第 2 种估计方法,它与前面的方法基本的不同在于:前面的方法基于网页

变化的次数来估计,而它则基于网页未变化的次数来估计.通常,网页未变化的可能性大于变化的可能性,因此,该方法应具有较好的性能.

由于网页在某段区间内不变化的可能性即式(10)中的 q,可以推导出 $r=\log q$,则第 2 种估计器为

$$\hat{r}=\hat{\lambda}/f=-\log(\overline{X}/n) \tag{12}$$

为处理网页未发生,变化次数为 0 的情况以及那些每次搜集检查都变化的网页,式(12)可修正如下:

$$\hat{r}=-\log\left(\frac{\overline{X}+a}{n+b}\right) \tag{13}$$

对该式的期望作泰勒展开可得:

$$E[\hat{r}]=E\left[-\log\frac{\overline{X}+a}{n+b}\right]=-\sum_{i=0}^{n}\left(\frac{i+a}{n+b}\right)\binom{n}{i}(1-\mathrm{e}^{-r})^{n-i}(\mathrm{e}^{-r})^{i}\approx\left[-\log\frac{\overline{X}+a}{n+b}\right]+\left[n\log\frac{n+a}{n-1+b}\right]r+\ldots \tag{14}$$

要使上式中的估计器无偏差,应使首项为 0,第 2 项系数为 1,故可得 $a=b=0.5$.此时,对式(13)的估计器计算期望和方差可得:式(13)比式(9)估计的偏差更小,效率更高;尤其重要的是,式(13)的估计器具有一致性,它的偏差随着 n 的增大而快速降低.在实际使用中,还有该估计器的一些变种和优化,如式(27)所示的估计器.

对于第 2 种针对网页变化频率所属类别的估计[29],例如按每周或每月变化一次来区分变化快慢,通常用贝叶斯方法来估计.该方法结合先验概率来进行,首先通过实验得到一个实验结果 E,然后根据 E 来计算变化频率所属类别的后验概率.例如,如果将网页 p 的变化频率只分为两类:一周变化一次(C_W)和一个月(C_M)变化一次,则属于前者的概率为

$$\begin{aligned}P\{p\in C_W\,|\,E\}&=\frac{P\{(p\in C_W)\cap E\}}{P\{E\}}=\frac{P\{(p\in C_W)\cap E\}}{P\{E\cap(p\in C_W)\}+P\{E\cap(p\in C_M)\}}\\&=\frac{P\{E\,|\,p\in C_W\}P\{p\in C_W\}}{P\{E\,|\,p\in C_W\}P\{p\in C_W\}+P\{E\,|\,p\in C_M\}P\{p\in C_M\}}\end{aligned} \tag{15}$$

若实验发现某个网页变化了 5 次,而属于上述两个类别的先验概率分别为 0.5 和 0.5,则属于这两类的后验概率分别为

$$\left.\begin{aligned}P\{p\in C_W\,|\,E\}&=\frac{(1-\mathrm{e}^{-5/7})\times0.5}{(1-\mathrm{e}^{-5/7})\times0.5+(1-\mathrm{e}^{-5/30})\times0.5}\approx0.77\\P\{p\in C_M\,|\,E\}&\approx0.23\end{aligned}\right\} \tag{16}$$

由此可以判断网页更有可能属于“一周变化一次”这一类.如果将该过程视为迭代过程,以前一次的后验概率作为后一次的先验概率,则可能得到更准确的判断结果.

1.3 估测效率

如果一个网页的变化频率已知,则可以很容易估计网页的下次变化时间.然而,实际实现中需要考虑两个因素:(1) 要进行相当长的搜集才能为每个网页获得足以对它的变化频率作有效预测的历史变化轨迹;(2) 如果为每个网页都维护历史变化轨迹,对于数以亿计的网页,很可能会成为系统的性能瓶颈.因此,文献[26]试图避开对网页的变化建模后估计参数的方法,而是直接实验采样以后估计网页的变化频率.

这种基于效率考虑的改进的基本出发点是,绝大多数网页以网站(或其他群体)的形式聚集,不同网站之间的平均变化频率相差极大.例如,商业网站的变化速度大大高于教育类或政府类网站[17].因此,不妨以诸如网站的网页群体为单位进行采样,获得各个网页群体的相对变化速度,从而为增量搜集进行任务分配.

这种变化频率估计的基本过程是:

(1) 对网页进行分组,一般将同一网站内的网页分为一组;

(2) 对每一组网页,选出一部分作为样本,搜集检查样本网页的变化频率;

(3) 参照样本的变化频率,将除去样本之后的剩余待搜集网页数量分配到各个网站.

上述过程中有两点是不明确的:对于第(2)步,由于每天实际搜集的网页数量是有限的,样本容量取多大?对

于第(3)步,通过样本得知各网站的变化频率的快慢后,如何给这些网站分配剩余的搜集量:是采取贪心法策略,全部分配给变化比例较大的网站,还是采取按比例分配的方法分配给所有网站?研究结果是,采取贪心法具有更好的时新性.

而对于最佳样本大小,可以估算它的值为

$$s \approx \sqrt{\frac{Nrf(\rho_t)}{6(\overline{\rho}_r - \overline{\rho})}} \tag{17}$$

其中:N 是所有网站的平均网页数;r 是下载网页数相对于网站总网页数的比值;ρ_i 是网站 i 中变化网页数的比例,$f(\rho_i)$则是所有网站的ρ_i概率密度函数;ρ_t是网站的ρ_i排在前r%的阈值;$\overline{\rho}_r$是所有在阈值之上的ρ_i的平均值,$\overline{\rho}$是所有网站ρ_i的平均值.

在实际估计中经常会出现搜集检查的样本网页量小于实际待检查的网站数量的情况.在这种情况下,为了提高采样的精确性,一般需要将网站聚成数量更少的组.

1.4 实验统计

关于网页的实际变化规律,有大量的实验性研究,例如文献[18–24]等.它们基本上都是通过对 Web 采样,长时间追踪样本的变化,然后统计获得网页的实际变化规律.本文从对网页的采样途径和考察属性上对这些工作分类并进行整理.

样本网页的获取方式一般有以下几种:

(1) 选取 Web 中的某些著名网站或从 Google Directory 等目录中选择有代表性的网站作为监控对象,研究这些网站内部的网页的变化,如文献[17,24,27,28,35].

(2) 选取某一次大规模搜集的网页作为监控对象,研究这个庞大的网页集合在一段时期内的变化,或与下一次搜集的网页相对比,如文献[19–22,36].

(3) 选取某次大规模搜集的网页中一部分作为监控对象,一般通过随机算法产生伪随机数来选择,例如文献[19,37,38];或是手工辅助挑选样本后搜集监控[39].

(4) 选取 Web 中具有相同主题的一部分网页作为监控对象,例如某些主题网站目录下链接的所有网页,参见文献[23,32].

(5) 从一些其他的角度进行采样,例如采取随机产生 IP 的方法进行采样,或者是采取对网关数据进行复制的方法来采样,见文献[18].

这些采样方法各有优点.一般来说,规模越大的样本对真实 Web 的反映越准确,但平均采样的间隔越长;而规模越小的样本对变化次数的估计越准.因此,既要考虑样本的全面性,还要考虑样本搜集的时新性,使得搜集间隔尽可能地小.

通过搜集和监控网页,可以获得如下几类网页属性数据:

(1) 不同类型和地点的网页的变化频率.例如,文献[17]统计得到商业网站的变化频率远大于教育网站;文献[19]统计得出中国的网页变化很慢,而德国的网页变化很快.

(2) 网页的变化频率和年龄的分布.例如,文献[22]观察到变化间隔服从明显的指数分布;文献[24]观察到变化间隔的指数分布并推导出变化的时间局部性规律.

(3) 网页的内容及相关属性的变化.例如,文献[28]和文献[19]从不同角度研究了网页内容的变化程度及规律;文献[23]将网页分成内容、结构和表现 3 部分分别进行研究;而文献[40]则研究了网页内容与其所属目录的主题的偏差的演化.

(4) 网页的链接分析属性的变化.例如,文献[41,42]得到网页的重要性随时间变化的情况;文献[24]得到网页的出度、入度等在短时期内基本不变的规律;文献[27]的实验得到了网站链接结构的演化规律及变化程度.

总的来说,不同的采样方法和不同的观察属性的任何一种组合都可以通过实验的方法来获得统计结果.这

些实验对增量搜集具有重要意义,尽管有些工作缺乏深入的理论分析.

1.5 价值演化

网页的价值(或称重要性)在增量搜集中意义重大,它既决定着网页的相对搜集次序,又影响着该网页时新性的价值.关于重要性与时间的相互影响,文献[42]通过实验考察了位于 Web 图中不同部位[8]的网页的链接分析权值与其年龄的相关性;文献[43]通过实验研究了网站的入度与时间的关系;更深入的工作[41,44]则从理论上进行了探讨.

从用户的角度考虑,网页的重要性可以通过以下几个因素来衡量:

- 流行度.用 $P(p,t)$表示网页在某个时刻的流行度.
- 价值.$Q(p)=P(L_p|A_p)$,网页的价值与时间无关,定义为一旦用户注意到该网页便喜欢该网页的概率,它是搜集系统所关心的网页的重要程度.
- 访问度.用 $V(p,t)$表示单位时间内访问该网页的用户数.
- 注意度.用 $A(p,t)$表示在某时刻注意到该网页的用户数.

上述 4 个指标不是独立的,下面将讨论它们之间的联系.根据定义,首先有

$$P(p,t)=A(p,t)Q(p) \tag{18}$$

在两个基本的假设(所有用户以相等的概率访问所有网页,以及 $V(p,t)=rP(p,t)$)下,注意度与流行度存在以下关系:

$$1-A(p,t)=\left(1-\frac{1}{n}\right)^{\int_0^t V(p,t)\mathrm{d}t}=\left[\left(1-\frac{1}{n}\right)^{-n}\right]^{-\frac{r}{n}\int_0^t P(p,t)\mathrm{d}t}\to \mathrm{e}^{-\frac{r}{n}\int_0^t P(p,t)\mathrm{d}t} \tag{19}$$

根据上式可得到网页的价值和它的流行度之间的关系

$$P(p,t)=\frac{Q(p)}{1+\left[\frac{Q(p)}{P(p,0)}-1\right]\mathrm{e}^{-\left[\frac{r}{n}Q(p)\right]t}},\ Q(p)=\left(\frac{n}{r}\right)\left(\frac{\mathrm{d}P(p,t)/\mathrm{d}t}{P(p,t)}\right)+P(p,t) \tag{20}$$

通常将网页的权值作为网页的流行度来计算.例如,通过链接分析和计算网页内容与查询的匹配度来计算流行度 $P(p,t)$.计算网页在某个时刻的权值的方法主要有:

(1) 基于全局链接分析的 PageRank 算法[45];

(2) 基于局部链接分析的 HITS(hypertext induced topic selection)算法[9,46];

(3) 基于网页内容与用户查询的相似度计算的 IR(information retrieval)方法,例如向量余弦夹角[47].

由于网页的搜集期限较长,一般只能获得这些流行度在某些时刻(若干次搜集后)的权值.可以通过式(20)从这些离散的流行度估计网页价值的演化过程[44]如下:

$$\hat{Q}(p,t_i)=\left(\frac{n}{r}\right)\left(\frac{P(p,t_i)/t_i}{P(p,t_i)}\right)+P(p,t_i) \tag{21}$$

其中,$P(p,t_i)$是网页在被搜集保存的时刻 t_i 的权值;$\hat{Q}(p,t_i)$ 是网页的价值.

如果考虑到其他因素对网页权值的影响[41],即当 V 与 P 并不成比例,例如当考虑搜索引擎系统对网页流行度的影响(权值高的网页通常排在查询结果前面,从而使得流行度越来越大)时,需要改变上述用户模型中的假设如下:

$$V(p,t)=r'P(p,t)^{9/4} \tag{22}$$

此时,在搜索引擎的影响下,基于式(22),网页的流行度按下式进行演化[41]:

$$\sum_{i=1}^{\infty}\frac{[P(p,t)]^{(i-9/4)}-[P(p,0)]^{(i-9/4)}}{(i-9/4)Q(p)^i}=\frac{r'}{n}t \tag{23}$$

据此,可以在某一次大规模的搜集后采取不同的算法计算出网页的权值(即流行度),然后,根据它估算网页在将来某个时刻的价值(即重要性),并决定其在搜集中的优先程度.

2 增量搜集的系统实现

上一节阐述了如何给网页的变化建立数学模型,并估计该模型的参数,除此之外,还介绍了网页的重要性如何随着时间流逝而演化.在这一节中,将假定网页的变化频率和网页价值已知,介绍当前发表的几个以此为基础的著名增量搜集系统及其技术;进一步地,将分析系统实现中的主要问题.

2.1 Web Fountain Crawler

IBM Almaden 研究中心开发的 Web Fountain Crawler 是一个强大的增量搜集系统.它的系统模型和工作原理见文献[48],其简单描述如下:

(1) 将网页按照它们的变化频率分为若干组,不同组的网页搜集代价(诸如下载速度之类的耗费)不同;

(2) 将搜集的过程分成一个个搜集周期,不同的搜集周期可以分配不同的搜集任务,而每个搜集周期内的网络带宽资源有限;

(3) 搜集的总目标是使每个周期中的旧网页、该周期内新出现的网页中,未来得及搜集部分的总过期时间尽可能地小(即时新性尽可能地高).

该模型的目标是使下面两种网页的过期时间最少:已经过期但未搜集的网页;当前周期中新出现但未搜集,从而过期的网页.以此为基础,确定该增量搜集系统的总目标

$$\text{Minimize}\left\{\sum_{t=1}^{T}\left[\left(\sum_{i=1}^{B} oldwt_{it}\times y_{it}\right)+newwt_{it}\times n_t\right]\right\} \tag{24}$$

其中,y_{it}是第 i 组在第 t 周期结束时过期的网页数,而 n_t是第 t 个周期结束时所有组中未搜集新网页数之和.式(24)的约束条件为

$$\text{在每个周期 } t \text{ 内满足总带宽约束: } Cconbt_t \geq \sum_{i=1}^{B} cconst_{it}x_{it} + dconst_t z_t,$$

即在 t 内的总带宽不小于各组内搜集旧网页所耗的带宽之和加上 t 内搜集新网页所耗带宽以后的总和.除此之外,与式(24)有关的约束关系还包括由 17 个变量构成的另外 3 个等式和 3 个不等式(例如下一个周期的网页数与上一个周期网页数的关系).

式(24)虽然明确刻画了增量搜集的目标,但它太多的系数和约束使得这个非线性约束条件下的非线性目标优化问题非常难解.在文献[48]中,使用 NEOS 公共服务器系统和标准 NLP 包 MINOS,根据模拟的网页变化数据来求解该问题.

IBM WebFountain Crawler 的系统模型(即式(24))最核心的参数在于各个周期的系数如 $oldwt_{it}$,但由于求解的复杂性,文献[48]只选取了一些具有代表性的系数组合进行实验,得到的结论是:为了得到较好的时新性,最后一个周期的 $oldwt_{it}$ 应最大,往前依次递减.

2.2 Univ. Chile Crawler

由智利大学开发的一个增量搜集系统[49](本文称为 Univ. Chile Crawler)主要从搜索引擎的网页索引的时新性来考虑增量搜集的调度.它将网页各方面的重要因素尤其是网页内容与用户查询的近似度等,作为网页的价值考虑进来影响搜集次序,同时,在搜集过程中,随着下次变化时刻的临近计算最近的变化可能性,从而选择价值较高的网页优先搜集.

Univ. Chile Crawler 所使用的网页搜集优先度的计算公式如下:

$$V(o_i) = q_i^a \times r_i^b \times p_i^c \tag{25}$$

在该式中,a,b,c 是动态可调整的参数,网页 o_i 的权值被分为 3 个部分:

q_i:表示网页的内在价值,根据如下因素计算:如 PageRank 和 HITS 算法之类的链接分析结果,与给定查询的相似度;在索引中被用户访问的次数;以及根据它的 URL 地址属性计算得到的一些参数.

r_i:表示网页的表示价值,用来衡量下载和保存该网页需要的存储资源,由以下几部分组成:URL 长度;AnchorText 大小;全文索引大小;文本摘要大小;全文长度.

p_i:表示网页的时新性,在间隔上次访问时间某一间隔 H_i 后仍然时新的可能性为

$$p_i = \mathrm{e}^{-\lambda_i H_i} \tag{26}$$

文献[49]使用的网页变化这一泊松过程的频率λ_i的估计公式为

$$\lambda_i \approx \frac{(X_i - 1) - \dfrac{X_i}{N_i \log(1 - X_i / N_i)}}{S_i T_i} \tag{27}$$

其中,N_i 是对网页的访问次数,S_i 是从网页第一次访问到当前的时间跨度,X_i 是该网页被发现变化的时间,T_i 是网页被发现没有变化的次数.

在系统实现上,Univ. Chile Crawler 有两个调度器:一个负责在较长时期内计算各网页的重要性,选取排在前面的网页以便在下一段时间搜集;另一个负责在短期内调度网页的搜集次序,以便控制从不同服务器上的网页下载量,保证对 Web 服务器访问的友好性.这个短期调度器的功能,实际上相当于 Mercator[6,50]中的 URL Frontier.

2.3 天网增量搜集系统

天网增量搜集系统由北京大学开发,旨在搜集中国 Web,它的原型可参见文献[5].关于天网系统网页搜集的增量调度策略及实现,文献[24,37,51]进行了详细论述.

设时刻 T_1 所有网页集为 $S(T_1)$,对它的每个元素,天网搜集系统要处理下列数据:

(1) 确定网页内容是否变化的摘要信息,记 *CheckPoints*;

(2) 网页最后变化时间 *LMT*,即 *CheckPoints* 变化的时间;

(3) 网页的变化间隔 *ChangePeriod*;

(4) 网页下次搜集时间 *NextCrawlTime*,即当前时间加上 *ChangePeriod*;

(5) 结果状态 *DisappearTimes*,是指连续发现网页不存在的次数.

天网系统将搜集过程分为一个个周期,并在每个周期都完成“搜集任务分配-搜集检查网页-提取新 URL”这 3 个步骤.搜集中需要操作的上述信息被组织成 5 块,分别是:

(1) 网页信息表 *PageInfoTable*:所有成功搜集的网页的信息.

(2) 新 URL 队列 *NewURLQueue*:未曾搜集的 URL.

(3) 未完成的 URL 列表 *RemainedURLs*:由资源限制当天未能搜集的 URL.

(4) *CheckPoints* 列表:所有网页的 *CheckPoint*.

(5) *ParsedURLs*:所有曾经被提取出来的 URL.

在上述 3 个步骤的每一步,都要对这些数据进行相应的数据读写.若 T_1 到 T_2 为一个搜集周期,假定系统的搜集能力为 N_C,该模型的详细流程如下:

第 1 步,搜集任务分配:若 I%的能力用于新网页,从 *NewURLQueue* 中选择 $I\%\times N_C$ 作为任务的一部分;再从 *PageInfoTable* 中找出 *NextCrawlTime* 已到的 F 个:

如果 $F<N_C-I\%\times N_C$,再从 *RemainedURLs* 中选 $N_C-I\%\times N_C-F$ 个;

否则,从 F 中选择 $N_C-I\%\times N_C$,个,其余的存入 *RemainedURLs*.

第 2 步,搜集检查网页:对分配的每个 URL 获得网页内容,根据返回状态和 *CheckPoints* 决定是否存储该网页和更新对应的信息.

第 3 步,提取新 URL:T_1 到 T_2 之间的网页搜集结束后,对提取出的 URL,检查它是否在 *ParsedURLs* 出现.若没有,则把它存入 *ParsedURLs* 和 *NewURLQueue* 中.

天网系统选择式(13)估算网页的变化频率.为缓解对大量网页历史轨迹的维护导致的性能瓶颈,它应用网页变化的时间局部性规律[37],在短时期内直接搜集多次变化的网页;而为了尽快获得新网页,它利用 Index 型网页[24]快速追踪新出现的网页.

这 3 个系统是目前公开发表且实际大规模运行的系统.除此之外,还有一些性能很好但未公开的增量搜集

系统,例如 Google 和百度(http://www.baidu.com/)的搜集系统等;以及很多实验性较强的小规模系统,例如文献[23]和文献[33]所使用的搜集系统.上述 3 个系统对大规模增量搜集系统的建立具有参考价值.

2.4 系统实现中的问题

以上主要介绍了几个实际系统的增量调度策略.对于搜集系统的系统架构和数据结构,文献[17,52]和文献[51]分别给出了一种实现.本节将分析这些系统涉及到的另外一些技术难题及解决方案.最主要的是网页质量问题:由于 URL[53]是无穷尽的,对它的选择应该在排序后有所取舍.由此导致以下两个重要的问题:

- 网页排序策略.网页质量主要受搜集系统采取的 URL 排序策略的影响,应对 URL 排序,以优先得到重要网页.文献[13]证明了广度优先策略会优先获得重要网页;文献[54]证明了 PageRank 优先策略强于网页入度优先策略或广度优先策略;文献[55]则证明了 OPIC(online page importance computation)方法[56]和大站点优先策略尽管不如 PageRank 优先策略,但具有比广度优先策略更好的效果.也可根据网页的内容来排序,例如利用网上信息的主题局部性[57]优先搜集同主题下的网页.尽管存在不少排序方法,但通过维护一个巨大的堆来选择当前最值得搜集的网页,实现起来非常复杂.
- 网页噪音消除.网页 Spam 对增量搜集影响恶劣,是搜索引擎最大的挑战之一[58].除了搜索接口[59]和站点重复[14,60,61]情况之外,主要的干扰来源于所谓的搜索引擎优化者[62],它们企图通过大量自动产生链接关系很强的网页集合来提高在索引结果中的排名.当前主要通过链接分析来识别 Spam,例如文献[12]给出 Spam 的特殊链接统计属性;文献[63]给出一种寻找链接关系强的异常网页群体的方法;文献[64]提出了一种 TrustRank 算法来判断网页的可信度;而文献[65]则提出了一种比文献[64]更准确的网页可信度计算方法.除此之外,从网页内容分析角度,文献[66]给出了通过统计网页内容中的高频词来识别 Spam 的方法.

除网页质量以外的其他问题主要与系统效率相关:如何调度搜集器以便能在下载速度[16,67]、站点友好性[68]和资源耗费[69,70]这 3 者中找到一个合适的折衷;如何寻找比 Shingling 方法[71]更好的根据网页内容计算出能够蕴含其变化程度[72]的摘要字符串[71].这些问题在当前还未见公开发表的完美解决方案.

3 增量搜集的理论分析

上一节介绍了 3 个大规模的增量搜集系统,它们都是在网页变化频率和网页重要性已知的前提下,给出实际的搜集调度策略来维护所搜集网页集合,使得该集合的平均时新性尽可能地高.本节将针对这些系统,进一步分析其性能,研究为了获得所维护网页集合的最佳时新性,如何制定最优的增量调度策略.

3.1 理论模型

增量搜集的理论模型可从两方面来分析:一是如何制定搜集性能的最优目标;二是为了达到该目标所采取的解决方案.

增量搜集的性能目标一般都被抽象为一个优化问题:

$$\left.\begin{array}{l}\text{Maximize}\sum_{i=1}^{N}\xi_i F(f_i,\lambda_i)\\ \text{Satisfying: 带宽或其他限制}\end{array}\right\} \tag{28}$$

即在满足带宽或其他资源的限制下,最大化所维护网页集合的平均时新性 $F(f_i,\lambda_i)$(或对其他对应尺度求极值,例如,像文献[48]那样最小化平均过期时间).其中,f_i 和 λ_i 分别是网页 i 的搜集频率和变化频率,ξ_i 是它的重要性.在不同的系统中,ξ_i 根据系统目标的差异而存在不同的定义:文献[73]中定义为用户访问该网页的可能性;文献[74]中定义为网页的 PageRank 值等重要尺度;文献[75]中定义为搜索引擎将该网页返回给用户的概率、该网页在索引结果中处于某位置的概率以及用户看到后点击该网页的概率这三者共同决定的一个参数;文献[33]中定义为用户查询与网页内容的匹配程度;文献[76]中则提出用数据挖掘的方法来确定该网页的权值,通过索引中点击频率、URL 位置和网页种类等学习决定.

对每个网页及其时新性的重要程度作出定义之后,可以如求解式(28)所示的优化问题,来确定每个网页的

最优搜集频率和最优搜集间隔(不同的搜集间隔对应着不同的搜集次序).对此问题,目前有 3 种解决思路:文献[25,52,77]将网页的变化视为泊松过程,对搜集频率和搜集间隔分开研究;文献[78]基于排队理论中的轮询系统(polling system)给增量搜集过程建立数学模型,并对优化问题求解;文献[75]将增量搜集视为标值点过程(marked point process)[79]来求解优化问题.分别介绍如下:

(1) 基于泊松过程的简单分析

文献[25,52,77]对搜集频率和搜集间隔的研究分开进行:在研究最优搜集间隔时,假定每个网页都有相同的搜集频率;在研究搜集频率时,又假定以均匀的间隔来搜集网页.这样处理虽然不够彻底,但仍然得到了很多有价值的结论.

首先,可以证明均匀的搜集间隔将获得最大的时新性.网页的搜集间隔可被简化为 3 种方式:A) 每周期内的网页次序固定,即均匀间隔;B) 每周期内的网页次序随机;C) 整个搜集期间内网页次序随机.对于方式 A),网页的平均时新性为

$$\overline{F(e_i)}=\frac{1}{I}\int_0^I E[F(e_i;t)]\mathrm{d}t \tag{29}$$

对于方式 B),首先考虑搜集间隔的概率函数为

$$W:\text{Interval of } e_i,\ f_W(w)=\begin{cases} w/I^2, & 0\le w\le I \\ (2I-w)/I^2, & I\le w\le 2I \end{cases} \tag{30}$$

因而可求得网页的平均时新性为

$$\overline{F(e_i)}=\lim_{t\to\infty}\frac{1}{t}\int_0^I F(e_i;t)\mathrm{d}t\frac{\int_0^{2I} f_W(w)\left(\int_0^w E[F(e_i;t)]\mathrm{d}t\right)\mathrm{d}w}{\int_0^{2I} f_W(w)w\mathrm{d}w} \tag{31}$$

对于方式 C),不失一般性,假定 $f_W(w)$的分布服从 Zipf 定律[80],则可以简化式(31).这 3 种不同的搜集次序对应的网页平均时新性和平均年龄见表 1.

Table 1 Average freshness and age under different URL-ordering policies

表 **1** 不同搜集次序(间隔)下的网页平均时新性和平均年龄

Policy	Freshness $\overline{F}(S)$	Age $\overline{A}(S)$
Fixed-Order	$\frac{1-\mathrm{e}^{-r}}{r}$	$\frac{1}{f}\left(\frac{1}{2}-\frac{1}{r}+\frac{1-\mathrm{e}^{-r}}{r^2}\right)$
Random-Order	$\frac{1}{r}\left(1-\left(\frac{1-\mathrm{e}^{-r}}{r}\right)^2\right)$	$\frac{1}{f}\left(\frac{1}{3}+\left(\frac{1}{2}-\frac{1}{r}\right)^2-\left(\frac{1-\mathrm{e}^{-r}}{r^2}\right)^2\right)$
Purely-Random	$\frac{1}{1+r}$	$\frac{1}{f}\left(\frac{r}{1+r}\right)$

由此可以比较它们的平均时新性,并得到均匀间隔最好的结论.采用类似的方法,文献[25,52,77]还证明了单一的搜集频率与正比于变化频率的搜集频率相比,具有更好的时新性.

(2) 基于轮询系统的分析

假定需要维护的网页数量固定,增量搜集系统可以看作一个轮询系统:下载进程对应于服务机器,网页对应于服务请求,下载时间对应于队列的切换时间(服务时间为 0),网页变化对应于服务请求到达,最佳的网页搜集序列对应于轮询系统中的一个最优的服务队列.基于排队理论中的轮询系统模型[81,82],文献[78]对增量调度进行了深入的分析.

对于任何一个调度策略π(每个网页的搜集频率和搜集间隔),式(28)被转化为

$$\text{Minimize}Cost = C(\pi) = \sum_{i=1}^{N} c_i r_i$$
$$r_i = \lim_{n\to\infty} \frac{\sum_{j=1}^{m_n^i} Z_j^i}{\sum_{j=1}^{m_n^i} X_j^i} = \frac{1}{E[X]} \lim_{n\to\infty} \frac{\sum_{j=1}^{m_n^i} E[Z_j^i]}{n} = \frac{\sum_{j=1}^{m_K^i} E[Z_j^i]}{K \times E[X]} \tag{32}$$

其中 c_i 为网页的权值(即重要性),r_i 为网页的平均过期程度.在对 r_i 的定义中, X_j^i 为网页 i 第 j 次下载的等待间隔; Z_j^i 为在第 j 次等待间隔中的过期时间; m_n^i 是前 n 次下载中访问网页 i 的次数,当 n 趋于无穷时,它就是访问频率;K 是以网页个数表示的循环周期的长度;π是网页的搜集序号函数,也就是调度策略.

基于轮询系统的研究结果[81,82],容易证明,为了减小平均过期程度,搜集间隔越均匀越好.当以频率 f_i 搜集变化频率为μ_i 的网页时,该网页的平均过期程度具有下界:

$$r_i = \frac{1}{E[X]} \left(E[X] - \frac{f_i}{\mu_i} + \frac{f_i}{\mu_i} h_i^{1/f} \right) \tag{33}$$

此时, $h_i = E[\mathrm{e}^{-\mu_i X}]$,为 X 在μ_i处的拉普拉斯变换.对于式(33),假定μ_i正比于网页权值 c_i(即变化越快的网页越重要),用拉格朗日算子解式(33)中的下界(作为式(32)的近似),得到最优的搜集频率 f_i 为

$$x_i = \frac{\ln h_i}{\sum_{i=1}^{N} \ln h_i} = \frac{\ln (h_i)^{-1}}{\sum_{i=1}^{N} \ln (h_i)^{-1}} \tag{34}$$

由此可知,仅当 X 为常量时,搜集频率 f_i 才正比于网页的变化频率μ_i.可见,即便是变化越快的网页越重要,为了达到最好的时新性,搜集频率也不一定正比于变化频率.

(3) 基于标值点过程的分析

文献[75]将标值点过程[79]引入增量搜集过程的建模,定义增量搜集中的若干个重要时刻如图 1 所示.

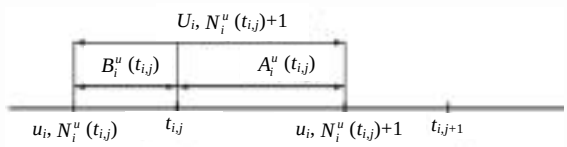

Fig.1 Important points in an incremental-crawling model based on marked point process

图 1 基于标值点过程的增量搜集模型中的重要时刻

在图 1 中,$t_{i,j}$ 是对第 i 个网页第 j 次搜集的时间;$u_{i,j}$ 是第 i 个网页第 j 次发生变化的时间; $N_i^u(t)$ 是第 i 个网页在时刻 t 之前(包括 t)变化的次数;$U_{i,j}=u_{i,j}-u_{i,j-1}$;而 A 和 B 则是图中所示的相应时刻的差.

对于第 i 个网页,它在上述搜集中的平均过期程度可由下式进行描述:

$$a_i(t_{i,1}, t_{i,2}, ..., t_{i,x_i}) = \frac{1}{T} \sum_{j=0}^{x_j} \int_{t_{i,j}}^{t_{i,j+1}} \left(\int_0^{\infty} \frac{P[A,B]}{P[B]} g_{B_i^u}(v) \mathrm{d}v \right) \mathrm{d}t \tag{35}$$

其中, $A = \{U_{i,n_{i,j+1}} \le t - t_{i,j} + v\}; B = \{U_{i,n_{i,j+1}} > v\}$; $n_{i,j} = N_i^u(t_{i,j})$.由式(35),假定网页的变化为强度为λ_i 的更新过程,$U_{i,j}$ 服从的分布为 G_i,上式可以推导如下:

$$a_i(t_{i,1}, t_{i,2}, ..., t_{i,x_i}) = \frac{1}{T} \sum_{j=0}^{x_j} \int_{t_{i,j}}^{t_{i,j+1}} \left(1 - \lambda_i \int_0^{\infty} (1 - G_i(t - t_{i,j} + v)) \mathrm{d}v \right) \mathrm{d}t \tag{36}$$

基于式(36),对更新过程的强度λ_i和更新间隔的分布函数 G_i 任取一个实例(例如泊松过程),都可以求出单个网页的最优平均过期程度 $A_i(x_i)$.将此结果代入式(28),并用网页被用户从索引中访问的概率作为网页重要性的衡量,可以求解式(28),获得一个网页集合中每个网页的最优搜集频率和最优搜集间隔.文献[75]直接引用了文献[83]在资源分配算法上的研究成果,用一种基于动态规划的算法对上述优化问题进行求解.

上述 3 种方案还不够完美:第 1 个模型将搜集频率和搜集间隔分开研究,虽然简单,但对二者之间的关系研究得不够透彻;第 2 个模型实际使用最佳调度方案的平均时新性的一个上界作为它的极值的近似,因而解决得

还不够完美;第 3 个模型比前两者要好,它不仅建立了理论模型,还给出了基于动态规划的求解策略,尽管求解代价通常很高.

3.2 可行的近似求解方法

从上可以看出,最优调度策略的产生通常对应着具有大量参数的非线性约束条件下的非线性目标函数优化问题的解决,求解复杂度太大.当网页的数量 n 达到数亿时,该问题的求解不具备实际意义.文献[73]提供了一种基于启发式策略的近似求解方法.

该方法以实际可行的局部最优解来替换不具有实际意义的全局最优解.算法如下:

(1) 将所维护的网页集合按照若干规则分成组;

(2) 为每组选择一个代表性的网页,用它的变化频率λ_i来代替该组内所有网页的变化频率,并用它的搜集频率 f_i 作为组内所有网页的搜集频率;

(3) 对所有的代表性网页,将原优化问题转换成一个规模小得多的近似的非线性目标函数优化问题,并对其求最优解;

(4) 将近似目标函数的解传递给原问题,并作用到各组内其他网页.

通过该算法,将式(28)所描述的优化问题

$$\left.\begin{array}{l}\text{Maximize}\overline{PF}=\sum_{i=1}^{N}p_i\overline{F}(f_i,\lambda_i)\\ \text{Satisfying}\sum_{i=1}^{N}f_i=TotalBandWidth\end{array}\right\} \tag{37}$$

转化为一个近似的实际较易解决的问题:

$$p_i=\sum_{e_j\in partition\ i}p_j/T_i,\lambda_i=\sum_{e_j\in partition\ i}\lambda_j/T_i$$

$$\left.\begin{array}{l}\text{Maximize}\overline{PF(S)}=\sum_{i=1}^{G}p_i\overline{F}(f_i,\lambda_i)T_i\\ \text{Satisfying}\sum_{i=1}^{G}T_i f_i=TotalBandWidth\end{array}\right\} \tag{38}$$

其中,p_i 为系统的搜集频率,一般对应着网页的重要性;λ_i 是网页的变化频率;N 是网页的数量;G 是对网页进行分组之后的组数;e_j 表示分组.

关于网页的分组方式,文献[73]采取了如下 4 种方法:

(1) 根据 p_i 分组,即 p_i 邻近的分为一组;

(2) 根据λ_i 分组,即λ_i 邻近的分为一组;

(3) 根据 p_i/λ_i 分组,即 p_i/λ_i 邻近的分为一组;

(4) 根据网页在某个特定 p_i 下的平均时新性分组.

实验结果显示,第 4 种分组方式具有最接近于最优理论值的时新性.更进一步地,可以将上述分组作为初始类别,用 N-neighbours 聚类算法[84]将具有较小欧式距离 $E_{dist}(e_1,e_2)=((p_1-p_2)^2+(\lambda_1-\lambda_2)^2)^{1/2}$ 的网页聚为一类,然后对聚好的类再次用近似的目标函数(式(35))作规划,以求近似解.这种经过聚类改进后的方法具有更好的效率和准确性.

4 总结和预测

本文对最近几年内网页变化和增量搜集技术相关的研究工作进行了总结.由于这项工作的基础——大规模的非增量式网页搜集技术已经很成熟[84−86],本文着重对网页的变化以及如何针对这种变化规律制定增量搜集的策略进行了论述,并介绍了系统实例.

对于网页变化,本文第 1 节详细阐述了网页变化的基本模型,如何估计该模型中的参数,以及该模型带来的性能瓶颈问题的一种基于采样的近似解决方法;特别地,分别对基于实验方法和基于理论分析研究不同种类的网页变化的规律进行了总结.

以网页的变化规律为基础,本文进一步介绍了迄今公开发表的 3 个大规模的实际持续运行的增量搜集系

统,它们分别是 IBM WebFountain Crawler,Univ. Chile Crawler 和天网增量搜集系统,特别是详细介绍了它们的系统模型和所使用的增量搜集策略.

对增量搜集的最佳调度算法,本文总结了相关研究工作.用多种数学方法给增量搜集系统建立模型后,将最优调度算法转化为一个非线性约束条件下的非线性目标函数规划问题,分析其最优解,并从实际出发介绍了它的一个基于启发式策略的近似最优解.

对网页变化和增量搜集技术在近期的研究发展,我们认为有以下 3 方面的趋势:

- 增量搜集模型的完善.大规模的增量搜集还没有公开的、完美的系统方案.本文介绍的 3 个系统[48,49,51]都使用了很多未经严格理论证明的近似解决办法[49,51],或是发表时仍未实现的理论方案[48].如何制定接近最优解的可行搜集策略,仍有待解决.
- 对网页质量的追求.既然 URL 搜集不完[53],且存在噪音,硬件资源也有限,就需要采取措施优先搜集质量高的网页.在不同的搜集策略(或其他条件)下,如何提高对不同质量的网页的覆盖率[87−89],在以后一段时期内仍然值得研究.
- 系统效率的提高.网页总数的高速增长,给搜集带来巨大的压力.寻找合适的数据结构,以适应对大规模网页的高效和快速搜集,对系统效率非常关键.在网页及相关信息的压缩(如文献[90])、索引和分析等处理上,其效率有待改善.

致谢 本文的工作以实现北京大学天网搜索引擎中的增量搜集子系统为背景,相关内容作为项目中开发该系统的技术基础.感谢李晓明教授在论文研究期间的指导以及对论文的修改所提出的宝贵意见,同时也感谢课题组的彭波老师和其他同学提供的帮助.

References:

[1] Shkapenyuk V, Suel T. Design and implementation of a high-performance distributed Web crawler. In: Proc. of the 18th Int'l Conf. on Data Engineering. San Jose: IEEE Press, 2002. 357−368.

[2] Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine. Computer Networks, 1998,30(1-7):107−117.

[3] Burner M. Crawling towards eternity: Building an archive of the World Wide Web. Web Techniques Magazine, 1997,2(5):37−40.

[4] Boldi P, Codenotti B, Santini M, Vigna S. Ubicrawler: A scalable fully distributed Web crawler. Software: Practice & Experience, 2004,34(8):711−726.

[5] Yan HF, Wang JY, Li XM, Guo L. Architectural design and evaluation of an efficient Web-crawling system. Journal of Systems and Software, 2002,60(3):185−193.

[6] Najork M, Heydon A. High-Performance Web crawling. Research Report, 173, Compaq Systems Research Center, 2001.

[7] Hafri Y, Djeraba C. High performance crawling system. In: Proc. of the 6th ACM SIGMM Int'l Workshop on Multimedia Information Retrieval. New York: ACM Press, 2004. 299−306.

[8] Broder A, Kumar R, Maghoul F, Raghavan P, Rajagopalan S, Stata R, Tomkins A, Wiener J. Graph structure in the Web: Experiments and models. In: Proc. of the 9th WWW Conf. North-Holland: Elsevier Science Publishers, 2000. 309−320.

[9] Kleinberg J, Kumar R, Raghavan P, Rajagopalan S, Tompkins A. The Web as a graph: Measurements, models, and methods. Lecture Notes in Computer Science, 1999,1627:1−18.

[10] Cooper C, Frieze A. Crawling on simple models of Web graphs. Internet Mathematics, 2003,1(1):57−90.

[11] Broder AZ, Najork M, Wiener JL. Efficient URL caching for World Wide Web crawling. In: Proc. of the 12th Int'l World Wide Web Conf. New York: ACM Press, 2003. 679−689.

[12] Fetterly D, Manasse M, Najork M. Spam, damn Spam, and statistics: Using statistical analysis to locate Spam Web pages. In: Amer-Yahia S, Gravano L, eds. Proc. of the 7th Int'l Workshop on the Web and Databases (WebDB 2004). New York: ACM Press, 2004. 1−6.

[13] Najork M, Wiener JL. Breadth-First crawling yields high-quality pages. In: Proc. of the 10th Int'l World Wide Web Conf. North-Holland: Elsevier Science Publishers, 2001. 114−118.

[14] Bharat K, Broder A. Mirror, mirror on the Web: A study of host pairs with replicated content. In: Proc. of the 8th Int'l World-Wide Web Conf. North-Holland: Elsevier Science Publishers, 1999. 501−512.

[15] Li XM, Feng WS. Two effective functions on hashing URL. Journal of Software, 2004,15(2):179−184 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/179.htm

[16] Cho J, Garcia-Molina H. Parallel crawlers. In: Proc. of the 11th World Wide Web Conf. New York: ACM Press, 2002. 124−135.

[17] Cho J, Garcia-Molina H. The evolution of the Web and implications for an incremental crawler. In: Proc. of the 26th Int'l Conf. on Very Large Databases. San Francisco: Morgan Kaufmann Publishers, 2000. 200−209.

[18] Douglis F, Feldmann A, Krishnamurthy B. Rate of change and other metrics: A live study of the World Wide Web. In: Proc. of the USENIX Symp. on Internet Technologies and Systems. 1997. 147−158.

[19] Fetterly D, Manasse M, Najork M, Wiener J. A large-scale study of the evolution of Web pages. In: Proc. of the 12th Int'l World Wide Web Conf. New York: ACM Press, 2003. 669−678.

[20] Fetterly D, Manasse M, Najork M. On the evolution of clusters of near-duplicate Web pages. In: Proc. of the 1st Latin American Web Congress. 2003. 37−45.

[21] Brewington BE, Cybenko G. How dynamic is the Web? In: Proc. of the 9th Int'l World Wide Web Conf. North-Holland: Elsevier Science Publishers, 2000. 257−276.

[22] Brewington BE, Cybenko G. Keeping up with the changing Web. Computer, 2000,33(5):52−58.

[23] Luis FR, Shipman F, Karadkar U, Furuta R, Arora A. Perception of content, structure, and presentation changes in Web-based hypertext. In: Proc. of the ACM Conf. on Hypertext. New York: ACM Press, 2001. 205−214.

[24] Meng T, Yan HF, Wang JM, Li XM. The evolution of link-attributes for pages and its implications on Web crawling. In: Proc. of the 2004 IEEE/WIC/ACM Int'l Conf. on Web Intelligence. Washington: IEEE Computer Society Press, 2004. 578−581.

[25] Cho J, Garcia-Molina H. Synchronizing a database to improve freshness. In: Proc. of the 2000 ACM Int'l Conf. on Management of Data. New York: ACM Press, 2000. 117−128.

[26] Cho J, Ntoulas A. Effective change detection using sampling. In: Proc. of the 28th Int'l Conf. on Very Large Databases. San Francisco: Morgan Kaufmann Publishers, 2002. 514−525.

[27] Ntoulas A, Cho J, Olston C. What's new on the Web? The evolution of the Web from a search engine perspective. In: Proc. of the 13th World-Wide Web Conf. New York: ACM Press, 2004. 1−12.

[28] Ipeirotis PG, Ntoulas A, Cho J, Gravano L. Modeling and managing content changes in text databases. In: Proc. of the Int'l Conf. on Data Engineering. Washington: Computer Society Press of the IEEE, 2005. 606−617.

[29] Cho J, Garcia-Molina H. Estimating frequency of change. ACM Trans. on Internet Technology, 2003,3(3):256−290.

[30] Padmanabhan VN, Qiu L. The content and access dynamics of a busy Web site: Findings and implications. In: Proc. of the ACM SIGCOMM Conf. New York: ACM, 2000. 111−123.

[31] Li XM. An estimation of the quantity of Web pages ever in China. Journal of Peking University (Science and Technology), 2003, 39(3):394−398 (in Chinese with English abstract).

[32] Bar-Ilan J, Peritz BD. Evolution, continuity, and disappearance of documents on a specific topic on the Web: A longitudinal study of "informetrics". Journal of the American Society for Information Science and Technology, 2004,55(11):980−990.

[33] Pandey S, Olston C. User-Centric Web crawling. In: Proc. of the 14th Int'l Conf. on World Wide Web. New York: ACM Press, 2005. 401−411.

[34] Arasu A, Cho J, Garcia-Molina H, Paepcke A, Raghavan S. Searching the Web. ACM Trans. on Internet Technology, 2001,1(1): 2−43.

[35] Ceaparu I, Shneiderman B. Finding governmental statistical data on the Web: A study of categorically organized links for the FedStats topics page. Journal of the American Society for Information Science and Technology, 2004,55(11):1008−1015.

[36] Toyoda M, Kitsuregawa M. Extracting evolution of Web communities from a series of Web archives. In: Proc. of the 14th ACM Conf. on Hypertext and hypermedia. New York: ACM Press, 2003. 28−37.

[37] Meng T, Yan HF, Wang JM. Characterizing temporal locality in changes of Web documents. Journal of the China Society for Scientific and Technical Information, 2005,24(4):398−406 (in Chinese with English abstract).

[38] Koehler W. Web page change and persistence—A four-year longitudinal study. Journal of the American Society for Information Science and Technology, 2002,53(2):162−171.

[39] Chiang WTM, Hagenbuchner M, Tsoi AC. The WT10G dataset and the evolution of the Web. In: Proc. of the 14th Int'l Conf. on World Wide Web. New York: ACM Press, 2005. 938−939.

[40] Dalal Z, Dash S, Dave P, Francisco-Revilla L, Furuta R, Karadkar U, Shipman F. Managing distributed collections: evaluating Web page changes, movement, and replacement. In: Proc. of the 4th ACM/IEEE-CS Joint Conf. on Digital Libraries. New York: ACM Press, 2004. 160−168.

[41] Cho J, Roy S. Impact of Web search engines on page popularity. In: Proc. of the 13th World-Wide Web Conf. New York: ACM Press, 2004. 20−29.

[42] Baeza-Yates RA, Saint-Jean F, Castillo C. Web structure, dynamics and page quality. Lecture Notes in Computer Science, 2002,2476:117−130.

[43] Adamic LA, Huberman BA. Power-Law distribution of the World Wide Web. Science, 2000,287:2115.

[44] Cho J, Roy S, Adams RE. Page quality: In search of an unbiased Web ranking. In: Proc. of the 2005 ACM Int'l Conf. on Management of Data. New York: ACM Press, 2005.

[45] Page L, Brin S, Motwani R, Winograd T. The PageRank citation ranking: Bringing order to the Web. Technology Report, 1998. http://www-db.stanford.edu/~backrub/pageranksub.ps

[46] Kleinberg JM. Authoritative sources in a hyperlinked environment. Journal of the ACM, 1999,46(5):604−632.

[47] Raghavan VV, Wong SKM. A critical analysis of vector space model for information retrieval. Journal of the American Society for Information Science, 1986,37(5):279−287.

[48] Edwards J, McCurley K, Tomlin J. An adaptive model for optimizing performance of an incremental Web crawler. In: Proc. of the 10th Int'l Conf. on World Wide Web. New York: ACM Press, 2001. 106−113.

[49] Castillo C, Baeza-Yates R. A new model for Web crawling. In: Proc. of the 11th World Wide Web Conf. New York: ACM Press, 2002.

[50] Heydon A, Najork M. Mercator: A scalable, extensible Web crawler. World Wide Web, 1999,2(4):219−229.

[51] Meng T, Yan HF, Wang JM. A model of efficient incremental spider for the Chinese Web and its implementation. Journal of Tsinghua University (Science and Technology), 2005,45(S1):1882−1886 (in Chinese with English abstract).

[52] Cho J. Crawling the Web: Discovery and maintenance of large-scale Web data [Ph.D. Thesis]. Stanford: Stanford University, 2001.

[53] Baeza-Yates R, Castillo C. Crawling the infinite Web: five levels are enough. In: Proc. of the 3rd Workshop on Web Graphs. Berlin: Springer-Verlag, 2004. 156−167.

[54] Cho J, Garcia-Molina H, Page L. Efficient crawling through URL ordering. Computer Networks, 1998,30(1-7):161−172.

[55] Baeza-Yates R, Castillo C, Marin M, Rodriguez A. Crawling a country: Better strategies than breadth-first for Web page ordering. In: Proc. of the 14th Int'l Conf. on World Wide Web. New York: ACM Press, 2005. 864−872.

[56] Abiteboul S, Preda M, Cobena G. Adaptive on-line page importance computation. In: Proc. of the 12th Int'l Conf. on World Wide Web. New York: ACM Press, 2003. 280−290.

[57] Davison BD. Topical locality in the Web. In: Proc. of the 23rd Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM Press, 2000. 272−279.

[58] Henzinger M, Motwani R, Silverstein C. Challenges in Web search engines. SIGIR Forum, 2002,36(2):1−8.

[59] Cope J, Craswell N, Hawking D. Automatic discovery of search interfaces on the Web. In: Proc. of the 14th Australasian Database Conf. 2003. 181−189.

[60] Bharat K, Broder A, Henzinger MR. A comparison of techniques to find mirrored hosts on the WWW. Journal of the American Society and Information Science, 2000,51(12):1114−1122.

[61] Sivasubramanian S, Szymaniak M, Pierre G, van Steen M. Replication for Web hosting systems. ACM Computing Surveys, 2004, 36(3):291−334.

[62] Perkins A. White paper: The classification of search engine spam. 2001. http://www.silverdisc.co.uk/articles/ spam-classification/

[63] Davison B. Recognizing nepotistic links on the Web. In: Proc. of the AAAI 2000 Workshop on Artificial Intelligence for Web Search. San Jose: AAAI Press, 2000. 23−28.

[64] Gyongyi Z, Garcia-Molina H, Pedersen J. Combatting Web spam with Trustrank. In: Proc. of the 30th VLDB Conf. San Francisco: Morgan Kaufmann Publishers, 2004. 576−587.

[65] Wu BN, Davison BD. Identifying link farm spam pages. In: Proc. of the 14th Int'l Conf. on World Wide Web. New York: ACM Press, 2005. 820−829.

[66] Fetterly D, Manasse M, Najork M. Detecting phrase-level duplication on the World Wide Web. In: Proc. of the SIGIR 2005. New York: ACM Press, 2005. 170−177.

[67] Czumaj A, Finch I, Gasieniec L, Gibbons A, Leng PH, Rytter W, Zito M. Efficient Web searching using temporal factors. Theoretical Computer Science, 2001,262(1-2):569−582.

[68] Koster M. Robots in the Web: Threat or treat? ConneXions, 1995,9(4):2−12.

[69] Craswell N, Crimmins F, Hawking D, Moffat A. Performance and cost tradeoffs in Web search. In: Proc. of the 15th Australasian Conf. on Database. 2004. 161−169.

[70] Talim J, Liu Z, Nain P, Coffman E. Optimizing the number of robots for Web search engines. Telecommunication Systems Journal, 2001,17(1-2):234−243.

[71] Broder A, Glassman S, Manasse M, Zweig G. Syntactic clustering of the Web. In: Proc. of the 6th Int'l World Wide Web Conf. New York: ACM, 1997. 1157−1166.

[72] Dyreson CE, Lin HL, Wang YX. Managing versions of Web documents in a transaction-time Web server. In: Proc. of the 13th Int'l Conf. on World Wide Web. New York: ACM Press, 2004. 422−432.

[73] Carney D, Lee S, Zdonik S. Scalable application-aware data freshening. In: Proc. of the 18th Int'l Conf. on Data Engineering. California: ACM Press, 2003. 481−492.

[74] Han JC, Cercone N, Hu XH. A weighted freshness metric for maintaining search engine local repository. In: Proc. of the 2004 IEEE/WIC/ACM Int'l Conf. on Web Intelligence. Washington: IEEE Press, 2004. 677−680.

[75] Wolf JL, Squillante MS, Yu PS, Sethuraman J, Ozsen L. Optimal crawling strategies for Web search engines. In: Proc. of the 11th Int'l World Wide Web Conf. New York: ACM Press, 2002. 136−147.

[76] Bullot H, Gupta SK, Mohania MK. A data-mining approach for optimizing performance of an incremental crawler. In: Proc. of the IEEE/WIC Int'l Conf. on Web Intelligence. Washington: IEEE Press, 2003. 610−613.

[77] Cho J, Garcia-Molina H. Effective page refresh policies for Web crawlers. ACM Trans. on Database Systems, 2003,28(4): 390−426.

[78] Jr Coffman EG, Liu Z, Weber RR. Optimal robot scheduling for Web search engines. Journal of Scheduling, 1998,1(1):15−29.

[79] Sigman K. Stationary Marked Point Process: An Intuitive Approach. New York: Chapman and Hall, 1995.

[80] Taylor HM, Karlin S. An Introduction To Stochastic Modeling. 3rd ed., New York: Academic Press, 1998.

[81] Borst S, Boxma OJ, Harink JHA, Huitema GB. Optimization of Fixed time polling schemes. Telecommunications Systems, 1994, 3(1):31−59.

[82] Boxma OJ, Levy H, Weststrate JA. Efficient visit orders for polling systems. Performance Evaluation, 1993,18(2):103−123.

[83] Ibaraki T, Katoh N. Resource Allocation Problems: Algorithmic Approaches. Cambridge: MIT Press, 1988.

[84] Chakrabarti S. Mining the Web: Discovering Knowledge from Hypertext Data. San Francisco: Morgan-Kauffman Publishers, 2002.

[85] Xu BW, Zhang WF. Search Engines and Information Retrieval Technology. Beijing: Tsinghua University Press, 2003 (in Chinese).

[86] Li XM, Yan HF, Wang JM. Search Engine—Principle, Technology and System. Beijing: Science Press, 2005 (in Chinese).

[87] Meng T, Yan HF, Li XM. An evaluation model on information coverage of search engines. Chinese Journal of Electronics, 2003, 31(8):1168−1172 (in Chinese with English abstract).

[88] Naumann F, Freytag JC, Leser U. Completeness of integrated information sources. Information Systems, 2004,29(7):583−615.

[89] Vaughan L, Thelwall M. Search engine coverage bias: Evidence and possible causes. Information Processing and Management: An Int'l Journal, 2004,40(4):693−707.

[90] Boldi P, Vigna S. The Webgraph framework I: Compression techniques. In: Proc. of the 13th World Wide Web Conf. New York: ACM Press, 2004. 595−601.

附中文参考文献:

[15] 李晓明,凤旺森.两种对 URL 的散列效果很好的函数.软件学报,2004,15(2):179−184. http://www.jos.org.cn/1000-9825/15/179.htm

[31] 李晓明.对中国曾有过静态网页数的一种估计.北京大学学报(自然科学版),2003,39(3):394−398.

[37] 孟涛,闫宏飞,王继民.Web 网页信息变化的时间局部性规律及其验证.情报学报,2005,24(4):398−406.

[51] 孟涛,闫宏飞,王继民.一个增量搜集中国 Web 的系统模型及其实现.清华大学学报(自然科学版),2005,45(S1):1882−1886.

[85] 徐宝文,张卫丰.搜索引擎与信息获取技术.北京:清华大学出版社,2003.

[86] 李晓明,闫宏飞,王继民.搜索引擎——原理、技术与系统.北京:科学出版社,2005.

[87] 孟涛,闫宏飞,李晓明.一个评价搜索引擎信息覆盖率的模型及其验证.电子学报,2003,31(8):1168−1172.

孟涛(1980－),男,湖北公安人,博士生,主要研究领域为 Web 搜索.

闫宏飞(1973－),男,博士,副教授,主要研究领域为 Web 搜索,信息检索.

王继民(1966－),男,博士,副教授,主要研究领域为 Web 搜索.

第 2 届智能 CAD 与数字娱乐学术会议(CIDE 2006)

征 文 通 知

由中国图象图形学学会计算机动画与数字娱乐专业委员会和中国人工智能学会智能 CAD 与数字艺术专业委员会联合主办，山东大学承办的第二届智能 CAD 与数字娱乐学术会议(CIDE2006)，将于 2006 年 10 月在美丽的泉城济南举行。本次会议将为智能 CAD、数字娱乐及相关领域的学者提供一个交流最新研究成果、进行广泛学术讨论的平台，届时邀请智能 CAD 与数字娱乐领域的知名学者和业界精英做精彩报告。会议期间，将举办数字艺术作品比赛和产品展示。会议录用的部分优秀论文将推荐至《计算机学报》、《软件学报》、《计算机辅助设计与图形学学报》、《系统仿真学报》、《中国图像图形学报》发表。大会论文集将由山东大学出版。

征稿范围（不仅限于如下主题）

● 智能 CAD	● 数字艺术	● 计算机动画	● 数字内容管理	● 虚拟现实
● 可视化技术	● E-hom	● 模式识别	● 人机交互	● 计算机图形学
● 图像处理	● 信息融合	● 人脸表情跟踪与识别	● 多媒体技术	● 计算机视觉
● 人工智能	● 交互式玩具	● 运动捕获动画	● 数字博物馆	● 网络游戏

投稿要求

论文必须未公开发表，中英文均可，不超过 10 页。论文包括题目、作者姓名、作者单位、摘要、关键字、正文、参考文献，具体格式请访问 http://cide2006.sdu.edu.cn

电子投稿，请将 WORD 格式的文件发到：cide2006@sdu.edu.cn

网络投稿：http://cide2006.sdu.edu.cn

重要日期

全文投稿：2006 年 6 月 10 日　　录用通知：2006 年 6 月 30 日　　修改定稿：2006 年 7 月 20 日

联 系 人：王璐，黄月珠　　**联系电话：**0531-88364810　　E-mail: cide2006@sdu.edu.cn