

# CDN 网络中对象一致性算法研究

陈 妍<sup>1</sup>, 李增智<sup>1</sup>, 王云岚<sup>2</sup>, 廖志刚<sup>1</sup>

(1. 西安交通大学计算机系统结构与网络研究所, 陕西西安 710049; 2. 西北工业大学高性能计算研究与发展中心, 陕西西安 710072)

**摘 要:** 针对内容分发网络设计的基于请求的对象一致性算法 ROCA, 以对象的请求频度为依据, 按照一定的刷新间隔 TTR, 从 web 服务器下载对象的最新版本。但 ROCA 算法存在着对象情况的滞后感知问题, 本文提出了对象模糊聚类算法 OFCA, 并对 ROCA 进行了优化。OFCA 算法以对象名称属性的概念分层为基础, 通过模糊聚类找出同类对象修改的共同规律, 并以此为依据修正 TTR。基于轨迹驱动的仿真结果表明, 算法获得的刷新时间符合对象的修改规律, 模糊聚类算法对 ROCA 有明显的改进。

**关键词:** 内容分发网络 CDN; 对象一致性; 模糊聚类算法; 概念分层

**中图分类号:** TP393 **文献标识码:** A **文章编号:** 0372-2112 (2005) 08-1480-04

## Study of Object Consistency Algorithm in CDN

CHEN Yan<sup>1</sup>, LI Zeng zhi<sup>1</sup>, WANG Yun lan<sup>2</sup>, LIAO Zhi gang<sup>1</sup>

(1. Institute of Computer Architecture & Network, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China;

2. Northwestern Polytechnical University, center for high performance computing, Xi'an, Shaanxi 710072, China)

**Abstract:** Request based Object Consistency Algorithm for Content Distribution Network (CDN)-ROCA was proposed to download modified object from web server by polling web server in Time to Refresh (TTR) interval. In order to make refresh interval more accurate, Object Fuzzy Clustering Algorithm (OFCA) was introduced to find commonness among objects and to optimize ROCA. OFCA adopted concept hierarchy to describe object name property, as well as heuristic method to find initial class mode. Trace driven experiments show that the TTR value obtained by ROCA is according to object modification, and fuzzy clustering algorithm can improve ROCA remarkably.

**Key words:** content distribution network (CDN); object consistency; fuzzy clustering algorithm; concept hierarchy

### 1 引言

内容分发网络(Content Distribution Network, CDN)中的对象一致性问题主要解决复制服务器中的对象与 web 服务器的原始对象的同步。对象一致性策略主要有:

(1) 基于 web 服务器的“推”方式: web 服务器将修改的对象分发给复制服务器<sup>[1]</sup>。

(2) 基于复制服务器的“拉”方式: 复制服务器定期轮询 web 服务器并将修改过的对象“拉”到自己的缓存<sup>[2]</sup>。常用的方法是为每个对象关联一个生存时间<sup>[3]</sup>或者租期<sup>[4]</sup>, 也可以为对象关联一个刷新间隔(Time to Refresh, TTR)<sup>[5]</sup>。第一种方法可以最大限度保证对象的一致性, 但缺乏灵活性和简单性, 并且会对 web 服务器及网络带宽带来很大的负担。因此第二种方法是最常用的策略。但绝大多数的研究工作主要依据对象的修改来确定刷新间隔。这样, 一方面需要复制服务器和 web 服务器交换大量信息, 另一方面这些信息中很多是无意义的(例如那些涉及长时间不被访问对象的信息)。

本文提出了一种以复制服务器中对象请求记录为依据的一致性算法, 一致性工作主要由复制服务器完成, 对 web 服务器和带宽资源的消耗较少。并且, 针对对象请求的处理方式也完全符合 CDN 网络的实际应用需求。

### 2 基于请求的对象一致性算法 ROCA

由于网络延迟的存在, 不可能保证复制服务器与 web 服务器对象版本完全、实时的一致。一致性策略只能保证  $\Delta$  一致性, 即对象同步时间差不超过  $\Delta$ , 如公式(1)所示。

$$\forall t, \exists r, 0 \leq r \leq \Delta, P_{t+r}^a = S_t^a \quad (1)$$

其中,  $S_t^a, P_t^a$  为对象  $a$  在时刻  $t$  在 web 服务器和复制服务器上的版本。

基于请求的对象一致性算法 ROCA<sup>[6]</sup>的基本思想是将对象按照请求频度分为两级兴趣度: 0 和 1。算法以对象为单位执行, 初始时兴趣度为 0。当对象有两次间隔小于阈值  $R_{Threshold}$  的连续请求, 则将其兴趣度调整为 1, 并设置刷新间隔  $TTR = R_{Threshold}$ ; 对于兴趣度为 1 的对象, 动态修正 TTR 取值使其符

合对象的修改规律. 如果  $TTR$  已经达到最大值, 且刷新间隔内无请求, 则重新设置兴趣度为 0.

ROCA 算法的核心是动态修正兴趣度为 1 的对象的刷新间隔  $TTR$ , 使其符合对象修改规律.  $TTR$  调整算法采用了基于动态滑动时间窗的处理方式, 通过对上一个时间窗内历史数据的学习, 获取对象请求、修改等相关测度; 并以此为基础不断地自我调整、逼近, 实现  $TTR$  与对象行为特征的高度吻合.

设:  $TTR_{max}$  和  $TTR_{min}$  为  $TTR$  的上界和下界.  $TTR_{min} = \Delta$  (即算法提供  $\Delta$  一致性);  $TTR_i$  为第  $i$  个时间窗的  $TTR$  值.  $Pc^i$  为对象在第  $i$  个时间窗内被请求的总次数;  $Ps^i$  表示对象在第  $i$  个时间窗内是否被修改过. 算法根据  $Ps^{i-1}$  和  $Pc^{i-1}$  动态调整  $TTR_i$  的取值, 基本算法如下:

$AdjustTTR()$

```

{
    if  $Pc^{i-1} = 0$  // 上一个时间窗内无请求
         $TTR_i = TTR_{i-1} * pow(2, w)$ ;
    else
        if  $Ps^{i-1} = 0$  // 上一个时间窗未被修改过
             $TTR_i = TTR_{i-1} + \epsilon$ ;
        else
             $TTR_i = \max(TTR_{i-1} * m, TTR_{min})$ ;
 $TTR_i = \text{Max}(TTR_{min} * a * \text{Min}(TTR_{max}, TTR_i) + (1 - a) * TTR_{i-1})$ ;
//  $0 \leq a \leq 1$ , 用来确定算法在调节  $TTR$  时, 保持旧值的概率
}

```

### 3 聚类算法对 ROCA 的优化

ROCA 能以较低的代价保证复制服务器中对象的新鲜程度, 但是算法存在着一定的滞后问题. 例如, 当对象的兴趣度从 0 变为 1 时,  $TTR$  的初始取值为  $R_{Threshold}$ , ROCA 算法需要经过若干个时间窗后才能将其调整到恰当的值. 本文提出了对象模糊聚类算法 OFCA 对 ROCA 算法进行了优化, 通过对复制服务器中保存的对象进行聚类分析, 找到对象修改的共同规律. OFCA 属于一种划分方法, 算法借鉴了  $k$ -中心点算法 PAM 的某些思想, 不同之处在于 PAM 只可用于属性值类型为数值数据的聚类, 而 OFCA 对聚类数据对象的属性值类型没有特殊要求; PAM 初始的中心点是随机选择的, 而 OFCA 通过启发式方法给出较优的初始类模式. OFCA 中引入模糊技术是为了解决由于对象属性选取相对简单而带来的聚类不准确问题.

OFCA 中采用了概念分层的思想<sup>[7]</sup>, 将对象的名称属性按照对象所属 web 站点的 URL、路径和名称定义为图 1 所示的概念分层结构. 为方便研究对该属性做如下定义:

定义 1 (属性取值) 假设  $\{O_1, O_2, \dots, O_m\}$  为对象属性的集合,  $O_i$  是一个属性,  $dom(O_i)$

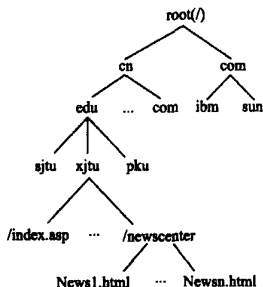


图 1 对象名称属性的概念层次

为属性  $O_i$  可能的取值集合, 对象  $o$  在属性  $O_i$  上的投影  $o[O_i]$  为对象  $o$  的属性  $O_i$  的值.

定义 2 (概念分层) 在  $dom(O_i)$  的元素上形成的树  $T_i$  被称为是概念分层. 设树的深度为  $h$ , 从根起各层的层号为  $1, 2, \dots, h$ . 定义第  $i$  层与第  $i+1$  层的相异性为  $W_{i,i+1}$ . 对于属性值  $x \in dom(O_i)$ , 其所在层次记为  $l(x)$ .

定义 3(祖先) 对于  $x, x' \in dom(O_i)$ , 如果在  $T_i$  中存在一条从  $x$  到  $x'$  的路径  $p$ , 则称  $x$  为  $x'$  的祖先.  $x$  的祖先的集合记为  $ance(x)$ . 若  $x$  为  $x'$  的祖先, 且  $x$  在  $l$  层,  $x'$  在  $l'$  层, 则  $x$  到其祖先  $x'$  的相异性定义为:

$$\delta(x, x') = \delta(x', x) = \sum_{j=1}^{l'-1} W_{j,j+1}$$

定义 4 (最小公共祖先) 对于  $y, y' \in dom(O_i)$ ,  $y, y'$  最小公共祖先的定义为:

$$\min-ance(y, y') = \begin{cases} y, y = y'; \\ y, \text{若 } y \text{ 为 } y' \text{ 的祖先}; \\ y', \text{若 } y' \text{ 为 } y \text{ 的祖先}; \\ x, \text{满足 } l(x) = \min\{l(z), z \in \text{ance}(y) \cap \text{ance}(y')\} \end{cases}$$

定义 5 (属性值相异性) 对于  $y, y' \in dom(O_i)$ ,  $y$  与  $y'$  之间的相异性定义为:  $\delta(y, y') = \delta_{\min-ance}((y, y'), y) + \delta_{\min-ance}((y, y'), y')$

定义 6 (对象的相异性) OFCA 中主要选取了对象的两个属性: 名称属性和类别属性. 其中, 类别属性主要用于对名称属性进行约束. 数据对象  $o, o'$  的相异性定义为:

$$d(o, o') = \begin{cases} \sum_{i=1}^m \delta(o[O_i], o'[O_i]), C = C' \\ 1, C \neq C' \end{cases} \quad (2)$$

其中:  $C, C'$  表示对象  $o, o'$  的类别. 从文献[8]的分析中可看出, 对于不同类别的对象有不同的修改规律. 对象的类别主要分为三类: Image 类对象很少被修改, application 类对象经常被修改, 对于 text/html 类对象一半以上对象很少被修改, 其余对象第一次访问后被修改.

定义 7 (类模式) 假设  $X$  为一组分类对象的集合,  $X$  的一个模式为向量  $Q = [q_1, q_2, \dots, q_m] \in \Omega$ , 使得  $D(X, Q) = \sum_{i=1}^n d(Q_i, Q)$  极小.

定义 8 (对象聚类问题) 假设  $D$  为对象请求数据库中需要进行聚类分析的数据集合, 把数据聚为  $k$  类, 即要求  $D$  的一个分划, 使得  $D = \cup_{i=1}^k D_i$ , 且  $D_i \cap D_j = \phi, i \neq j$ , 若  $D_i$  类的模式为  $M_i$ . 则对象聚类的目标函数为:  $f = \sum_{i=1}^k D(D_i, M_i)$ .

设对象  $o_i$  属于  $D_j$  的隶属度为  $P_{ij}$ , 且  $\sum_{j=1}^k P_{ij} = 1, i = 1, \dots, m$ , 则对象模糊聚类的目标函数修正为:  $f_F = \sum_{j=1}^k \sum_{i=1}^m [P_{ij}]^b D(D_i, M_i)$ , 其中:  $b > 1$  为可控的聚类结果模糊程度常量.

对象模糊聚类算法 OFCA 描述如下:

(1) 为每个类选择一个初始类模式;

Repeat

(2) 对于对象  $o_i$ , 按照公式(3) 计算对象的隶属度函数, 从而得出隶属度矩阵  $P$ .

$$P_{ij} = \frac{1}{\sum_{r=1}^k \left( \frac{d(o_i, D_j)}{d(o_i, D_r)} \right)^{1/(b-1)}} \quad (3)$$

(3) 根据当前的隶属度函数更新每个类的类模式.

Until 迭代次数达到预定义的值, 或没有对象被置入新的类中.

初始类模式的选择应满足两个目标: (1) 类模式中每个属性的值尽可能处于较高的概念层; (2) 不同类的类模式之间的相异性较大. 采用以下启发式方法选择初始类模式:

(1) 选择聚类对象集  $D$  的一个样本  $S$ .

(2) 对  $S$  中的每个对象  $o$ , 计算  $g(o) = \sum_{o_i \in O} (o[O_i])$ .

(3) 从  $S$  中取函数  $g(o)$  值最小的前  $N$  个对象组成对象集  $S'$ , 按照公式(2) 计算这  $N$  组对象两两间的相异度.

(4) 取  $S'$  中  $g(o)$  最小的一个对象作为第一个类模式  $Q_1$ ,  $N = N - \{Q_1\}$ .

(5) 假设已选定  $i$  个类模式, 选择对象  $o \in S'$  作为第  $(i+1)$  个类模式时,  $\sum_{1 \leq j \leq i} \delta(o, Q_j)$  最小,  $S' = S' - \{Q_{i+1}\}$ .

(6) 重复(5), 直到选定  $k$  个类模式.

聚类算法定期执行, 并按公式(4)对  $TTR$  进行修正.

$$TTR_i^{new} = u \times \sum_{j=1}^k (P_{ij} \times TTR_j^{ave}) + ((1-u) \times TTR_i^{old}) \quad (4)$$

其中:  $TTR_i^{new}$  和  $TTR_i^{old}$  表示对象  $o_i$  修正前后的  $TTR$ ,  $TTR_j^{ave}$  为第  $j$  类对象的平均  $TTR$ ;  $0 \leq u \leq 1$ , 用来确定同类对象刷新时间的相似度.

### 4 仿真实验

选用公共轨迹文件对算法进行了实验. 实验数据来自美国应用网络实验室 NLNR 的 web 代理服务服务器轨迹<sup>[9]</sup> (proxy trace), 详细信息如表 1 所示. 考虑到轨迹文件中未包含对象的修改信息, 采用文献[10]中的实验方法, 在 trace 数据中为对象增加了基于经验驱动模拟修改数据. 限于篇幅, 以下实验结果都是以 text/html 类对象为例说明的, 并且对象采用定期修改方式.

表 1 实验轨迹数据

持续时间	请求数	模拟修改数	轨迹文件大小/kB
2004 10 05T 00:00:00- 24:00:00	410, 167	7, 867	70, 530
2004 10 06T 00:00:00- 24:00:00	410, 881	9, 382	62, 465

图 2 中给出了 ROCA 算法  $TTR$  的调整曲线以及不同  $TTR$  区间内对象的请求量(图 2 中的原点). 算法参数为  $TTR_{max} = 1800s$ ,  $TTR_{min} = 300s$ ,  $m = 0.8$ ,  $\epsilon = 200$ . 对象修改间隔为 1800s.

OFCA 的  $TTR$  调整实验的结果如图 3 所示. 图 3 中过滤了因长时间没有请求导致  $TTR$  大于  $TTR_{max}$  的情况. 这是因为, 一方面优化算法不对这种情况进行修正, 另一方面会影响图表的清晰度. 从图 3 中可以看出优化后的  $TTR$  与实际的

对象修改规律更加接近.

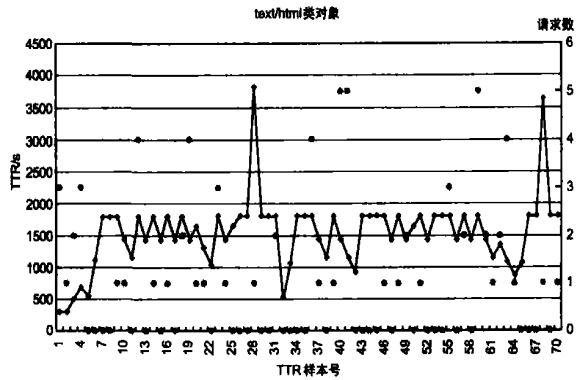


图 2 ROCA 的 TTR 调整结果

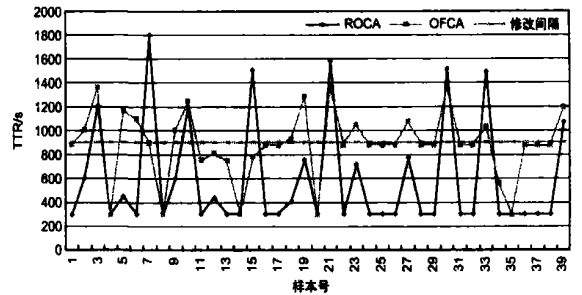


图 3 OFCA 算法的优化结果

### 5 结束语

本文提出了一种基于请求的 CDN 网络的对象一致性算法. 该算法主要由复制服务器完成, 不会给 web 服务器带来额外的负荷. 基于滑动时间窗的方式只对最近时间段的信息进行处理, 这不仅大大减轻了处理的复杂度, 而且也符合 CDN 网络的请求规律. 模糊聚类算法的引入, 避免了算法的滞后问题. 进一步的研究工作将集中在根据 web 服务器和网络的负载情况调整刷新时间和策略, 以及算法效率的提高.

### 参考文献:

[ 1 ] Balachander Krishnamurthy, Craig E. Wills. Study of piggyback cache validation for proxy caches in the world wide web[ A ]. USENIX Symposium on Internet Technologies and Systems[ C ]. California, USA, December, 1997. 1- 13.

[ 2 ] James Gwertzman, Margo Seltzer. World Wide Web cache consistency [ A ]. Usenix Technical Conference[ C ]. San Diego, CA, Jan. 1996. 141 - 152.

[ 3 ] L Breslau, P Cao, L Fan, G Phillips, S Shenker. Web caching, and zipf like distributions: Evidence, and implications[ A ]. IEEE INFOCOM' 99 [ C ]. New York, March, 1999. 126- 134.

[ 4 ] V Duvvri. Adaptive leases: A strong consistency mechanism for the world wide web[D]. Master's thesis, Univ. of Massachusetts, 1999.

[ 5 ] B Urgaonkar, A Ninan, M Raunak, P Shenoy, K Ramamritham. Maintaining mutual consistency for cached web objects[ A ]. The 21st International Conference on Distributed Computing Systems[ C ]. Mesa, AZ, April, 2001. 371- 380.

[ 6 ] 陈妍, 李增智, 廖志刚. 内容分发网络中基于请求的对象一致

性算法[J]. 西安交通大学学报, 2005, 39(8): 836- 839.

- [ 7 ] Jiawei Han, Micheline Kamber. 数据挖掘概念与技术[M]. 范明, 孟小峰, 等. 译. 北京: 机械工业出版社, 2001.
- [ 8 ] F Douglas, A Feldmann, B Krishnamurthy, J Mogul. Rate of change and other metrics: A live study of the World Wide Web[ A ]. USENIX Symposium on Internet Technologies and Systems [ C ]. Monterey, Calif

omia, USA, December 1997.

- [ 9 ] NLNR, Weekly access logs at NLNR' s proxy caches[ EB/OL]. ftp://ftp.ircache.net/Traces/. 2004-10-08.
- [ 10 ] A Ninan. Maintaining Cache Consistency in Content Distribution Networks[D]. Master' s thesis, Department of Computer Science, Univ. of Massachusetts, June 2001.

#### 作者简介:



陈 妍 女, 1972 年出生于西安, 西安交通大学计算机系统结构与网络所讲师, 博士, 主要研究方向为内容分发网络、主动网络及网络应用. E-mail: chenyan@mail.xjtu.edu.cn.



李增智 男, 西安交通大学教授、博士生导师, 中国计算机学会网络与数据通信专业委员会委员, 主要研究方向为计算机网络及应用.