# 适应性 Web 缓存的研究*)

李文中　顾铁成　周　俊　陆桑璐　陈道蓄
(南京大学计算机科学与技术系　计算机软件新技术国家重点实验室　南京 210093)

**摘　要**　近年来,由于 Web 缓存技术对缓解因特网上热点现象的有效性,它已迅速得到了研究人员和业界的关注。适应性 Web 缓存(adaptive Web caching)由于能够根据用户的不同访问模式,自适应地调整热点数据在缓存系统中的分布,自动均衡整个缓存系统的负载,因而成为了缓存技术研究的一个新的热点。本文介绍了适应性 Web 缓存领域的研究状况,详细分析了基于组播和基于单播这两种主要的适应性缓存技术。最后,指出了适应性 Web 缓存研究存在的问题和值得进一步改进的方向。

**关键词**　适应性 Web 缓存,请求转发,页面回传,负载均衡

## Research on Adaptive Web Caching

LI Wen-Zhong　GU Tie-Cheng　ZHOU Jun　LU Sang-Lu　CHEN Dao-Xu
(Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

**Abstract**　Recently, in the face of rapid growth of the Internet, Web caching has emerged as a effective way to reduce network traffic and latency. Adaptive Web caching is a new subfield of Web caching, which can adapt to different user access patterns, self-organize system structure and distribution of popular data, hence reducing overall network load. In this paper, we firstly overview the recent research status in adaptive Web caching. Then we analyze two kinds of adaptive Web caching systems, one of witch uses multicast while the other uses unicast. At last we discuss some issues in the research of adaptive Web caching and the future work.

**Keywords**　Adaptive web caching, Request forward, Page retrieval, Load balancing

## 1　引言

随着 Internet 的发展,热点数据分布的不均匀而引起网络拥塞的现象日益严重。Web 缓存(Web caching)是缓解网络拥塞、减少访问延迟的有效方法[6,10~12]。缓存服务器一般放置于用户与源服务器之间,存放用户经常访问的数据,减少用户到源服务器的链路上的数据传输量。当用户请求的数据不在缓存中时,缓存服务器向源服务器请求数据,或将用户请求转发给其他缓存服务器。请求转发的思想导致了协同缓存的出现。单个缓存服务器容易成为瓶颈,且没有扩展性,协同缓存通过一组缓存服务器相互协作,可以达到更好的缓存利用率和命中率,而且各缓存服务器的负载比较均衡,可以避免瓶颈和单点出错等问题[6]。

适应性缓存(adaptive Web caching)是协同缓存的一个研究分支,与其它协同缓存技术不同,适应性缓存系统的配置和组织不需要人工干预,各缓存服务器可以自动根据用户访问情况,自适应地调整缓存的内容,优化数据的分布,对用户频繁访问的数据自动产生多个副本,并将其放置在离用户端较近的缓存上,从而可以节省网络带宽,减少访问延迟。因而,其研究和应用代表着 Web 缓存技术发展的一个方向。适应性缓存的组织存在着若干种方法,它们的思想与做法都有所不同,本文对这些方法进行了分析研究,并对可以进一步改进之处进行了总结。

本文第 2 节描述了适应性缓存的含义及相关研究;第 3 节分析了基于组播的适应性缓存系统;第 4 节分析了基于单播的适应缓存系统;最后比较了两种适应性缓存系统的优缺点,指出适应性缓存的研究方向和未来的工作。

## 2　适应性缓存的含义及相关研究

到目前为止,有关适应性缓存并没有一个很准确的定义,不同的文章对于适应性缓存有不同的理解。对于缓存"适应性"含义的理解,归纳起来有如下几种:对硬件的适应性,即对用户不同的硬件和不同的网络带宽要求,自适应地提供不同质量的数据[8];对内容的适应性,即根据用户的请求中对数据质量的描述和约束,传送不同质量的数据(主要是多媒体数据)[7];对缓存替换算法的适应性,即根据用户访问情况,利用在线学习机制自动选择性能最好的缓存替换策略[9];对用户访问模式的适应性,即使热点数据自动放置在靠近用户的缓存服务器上[1]。

综合起来看,在此,我们可以这样来给出适应性缓存的定义:一组自治的、分布的、自组织的缓存服务器协同工作,根据用户的访问情况,自适应地调整缓存的数据,使热点数据靠近用户分布,从而节省网络带宽,减少访问延迟。

在实现自适应缓存系统时,为了达到这一目标,存在着几种不同的作法,下面我们就来讨论两种适应性缓存系统:基于组播的适应性缓存和基于单播的适应性缓存。

*)本文得到国家八六三技术研究发展计划(编号 2001AA113050)资助。李文中　硕士研究生,主要研究方向为分布式计算。顾铁成　博士研究生,主要研究方向为分布与并行计算。周　俊　硕士研究生,主要研究方向为分布式计算。陆桑璐　教授,主要研究领域为分布与并行计算。陈道蓄　教授,博士生导师,主要研究领域为分布计算与并行处理。

## 3 基于组播的适应性缓存

目前，大部分协同缓存系统都是由网络管理员手工配置和组织的，如 SQUID 系统，它依赖于人的经验。这种配置方法是相对比较固定的，它不能很好地适应网络访问突然变化的情况。另一方面，固定配置还会使得系统的可扩展性较差。

Lixia Zhang 等人在文[1]中提出了一种基于组播的适应性 Web 缓存技术。在他们的系统中，Web 服务器与缓存服务器被组织成多个相互交迭的组播组，使用组播技术转发用户请求和回传数据，根据用户的访问情况自动调整缓存数据的分布，用户访问频繁的数据能沿着分布树从源服务器向客户端靠近。通过组管理和维护策略自动调整组播组的大小，从而达到均衡负载的目的。下面就对该系统进行分析。

### 3.1 体系结构

在基于组播的适应性缓存系统中，Web 服务器与缓存服务器被组织成多个相互交迭的组播组，如图 1 所示。G1，G2，G3……是交迭的组播组，C1，C2，C3……是缓存服务器。

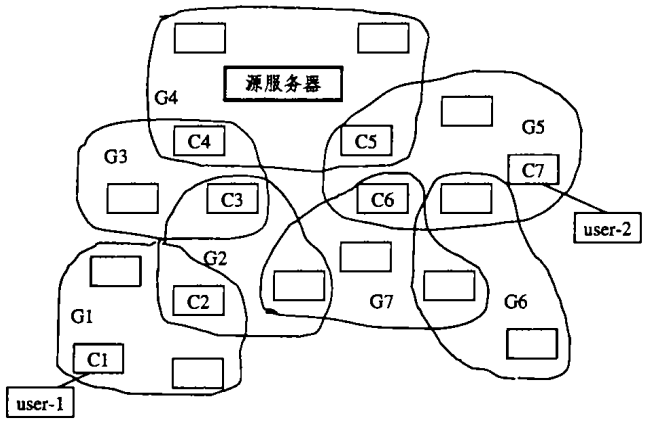


图 1 基于组播的适应性缓存体系结构

在该系统中，组播的作用有两个：其一是信息发现，一个缓存服务器不必确切知道其他缓存服务器上的数据，只需简单地将请求向相关组内组播即可；其二是数据分发，组播是将同样的数据同时发给多个接收者的最有效方法。

### 3.2 请求转发和页面回传机制

适应性缓存系统被组织成多个相互交迭的组播组，一个缓存服务器可以同时属于多个组。当一个组中所有缓存服务器都没有用户所请求的页面时，用户请求应该向邻近的组转发。下面详细讨论页面请求、转发、回传的机制。

(1)当用户 user-1 请求访问一个页面时，该请求被发送到一个附近的代理缓存服务器 C1，如果 C1 的缓存中没有用户请求的页面，C1 就将用户请求向它所在的组播组 G1 中组播，若该组中某一个缓存服务器如 C2 缓存有用户所请求的页面，则 C2 将该页面向 G1 组进行组播，C1 接收 C2 发送的数据，再回传给用户。

(2)若组 G1 中没有用户请求的数据时，G1 中所有缓存服务器将检测自己是否存在离源服务器更短的路径，例如，G1 中的缓存服务器 C2 发现自己所属的另一个组 G2 离源服务器更近，因此，C2 将用户请求向组 G2 组播。

(3)若 G2 也没有所请求的数据，再使用同样的策略，由 G2 中的缓存服务器如 C3 将请求向组 G3 转发。

(4)持续上述转发过程，直到请求到达一个存有用户请求数据的缓存服务器或到达源服务器为止。

当请求到达一个拥有该数据的缓存服务器或源服务器时，页面回传的机制如下：

(1)拥有请求页面的服务器将页面组播，同一组内的缓存服务器都接收并缓存该页面。图 1 中，假设只有源服务器拥有用户请求的页面，则源服务器将页面向 G4 组播，G4 中所有缓存服务器(如 C4，C5)都接收并缓存该页面，下一次如果另一个用户 user-2 向缓存服务器 C7 再次请求该页面时，请求只需转发到 C5，由 C5 向 G5 组播该页面。

(2)沿着路径上的其他组里的缓存服务器只需负责将数据沿原路回传，不作组播或缓存工作。如图，当 C4 获得所请求的页面后，沿路径上的 C3，C2，C1 只需简单地将页面沿路回送给客户即可。

回传机制可以实现让热点数据靠近用户分布：同一个页面的访问次数每增加一次，页面就向用户移近一组，访问的次数越多，页面越靠近用户。

### 3.3 请求转发的若干问题

请求转发希望沿着一条较短的路径向源服务器转发，最坏情况下，沿路径上的缓存服务器都没有所请求的数据，则用户请求最后到达源服务器，再由源服务器沿原路径返回给用户。

动态确定请求转发路径是适应性缓存研究的一个核心问题。对于同一组中的每个缓存服务器，应该有一种机制来自动决定是否转发请求。由于缓存运行在主机而不是路由器上，每个缓存服务器中并没有网络拓扑结构的信息。对于一个缓存服务器 C，它可以得到的信息有：发出组播的缓存服务器 N 的地址和用户请求页面的源服务器 S 的地址。C 应该使用某种策略来判断自己是否比 N 更靠近 S，从而决定是否转发该请求。相关的解决方案有如下几种：(1)通过比较网络 ID，若发现 S 和 C 接在同一个网络上，则由 C 转发用户请求；(2)利用用户请求中的域名信息，域名中一般包含有国家、地区等表示地域范围的信息，C 可以根据这些信息，判断自己是否比 N 在地理上更靠近 S；(3)利用历史经验，C 可以根据历史访问情况，采用一种学习机制，确定请求转发的方向；(4)增加邻近组的信息，对于一个组内的缓存服务器，不仅可以知道本组内缓存服务器的地址，还可以知道邻近组的缓存服务器地址，这样，就可以用广度优先的遍历算法找到一条到 S 的最短路径；(5)为 IP 路由协议开发一种新的标准接口，这样，C 可以通过查询它邻近的路由器来判断自己是否比 N 更靠近 S。

这 5 个解决方案各自还有不完善的地方，并不能完全解决请求转发的问题。值得提出的是，请求转发不一定要沿着向源服务器的最短路径进行，还应该考虑到各服务器的负载情况，如果最短路径上的缓存服务器负载较重时，可以沿着一条较长的、但负载较轻的路径转发，这样响应时间可能更快。

一个组内各缓存服务器是各自决定是否转发请求的，理想的情况下，只有一个缓存服务器转发请求，否则会产生以下情况：

(1)有多个缓存服务器都转发请求，会引起“冲突”，有可能两个缓存服务器的转发路径有重合，而且同一个页面可能被取回多次。

(2)若所有缓存服务器都不能转发请求，则发出广播的缓存服务器应该直接向源服务器请求数据，或者从组里随机选择一个缓存服务器来转发请求。

### 3.4 组的创建和管理

当适应性缓存系统被组织成多个相互交迭的组播组，这些组的结构并不是固定的，可以随着用户访问情况的变化动态调整。系统根据用户数量和负载情况，来自动创建新的组或合并相邻的组。下面详细介绍组播组的创建和管理方法。

3.4.1 *组创建* 当一个新的缓存服务器如 C2 要加入缓存系统时，它可以加入已经存在的一个组，或者创建一个新的组，其步骤如下：

(1)当 C2 要加入缓存系统，C2 向系统中现有的组组播一个 Group Join 请求；

(2)C2 等待 TTL(Time To Live)长的一段时间，如果在 TTL 时间内，C2 接收到某个组 G2 中的缓存服务器 C3 的应答信息，则 C2 加入 G2 组；

(3)若在 TTL 时间内，C2 接收到多个组的应答信息，C2 可以根据这些组的负载状况，距离远近等信息选择加入其中一个组；

(4)若在 TTL 时间内 C2 接收不到任何组的应答信息，则 C2 独自创建一个新的组，并且为该组设置一个计时器，C2 开始等待其他新的缓存服务器加入该组。如果计时器超过一定域值后仍然没有新的缓存服务器加入该组，C2 将增加 TTL 的值，然后重复(1)-(4)的步骤，再次尝试加入到现有的一个组中。

3.4.2 *组管理* 组播组根据当前负载情况和用户数量自动调整组的大小，主要有组的分裂和合并两种情况：

(1)组分裂：一个组中缓存服务器数量越多，组内的负载和网络拥塞也会增加。当一个组的负载过高时，该组会根据各缓存服务器当前的负载状况和它们之间的距离自动分裂成两个较小的组。

(2)组合并：当一个组的负载和缓存命中率都很低时，它可以根据邻近各组的负载情况，考虑与相邻的某一组合并。

组的分裂与合并机制可起到负载均衡的作用。当一个组比较大时，该组的负载和网络拥塞都比较重，分裂成两个较小的组后，各自的负载都会减轻。相反，当一个组负载较轻，有较多闲置资源时，与邻近的组合并后，可使闲置的资源得到充分利用。

## 4 基于单播的适应性缓存

组播需要特殊的设备和协议的支持，当前 Internet 上并不是所有路由器都支持组播的，因此，基于组播的适应性缓存实现起来比较困难。在 M. J. Kaiser 等人的研究中，提出了一种基于单播的适应性缓存[3~5]，它比较易于在当前的 Internet 环境下实现。

### 4.1 体系结构

从逻辑上看，基于单播的适应性缓存系统，可以分为三个层次：用户层，代理缓存服务器层，源服务器层，如图 2 所示。缓存服务器层之间的互连是任意的，每个缓存服务器维护一个列表，表示与之相邻的缓存服务器。各缓存服务器协同工作，用户请求可以发送给任一个缓存代理服务器，如果该服务器上不存在用户请求的内容，则通过请求转发机制将请求转发到邻近的缓存服务器或源服务器。缓存服务器根据用户访问情况，自适应地调整缓存数据的分布，对于热点数据，可以生成多个副本。

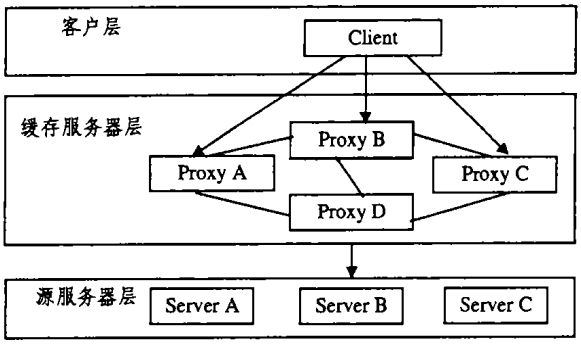


图 2 基于单播的适应性缓存体系结构

### 4.2 映射表

适应性缓存是运行在主机而非路由器上的，因此，请求转发时需要维护相应的路由信息，这里使用一个称为映射表的数据结构，如图 3 所示。映射表主要用于目标定位和请求转发过程。映射表初始为空表，适应性缓存通过学习逐步建立路由信息，每个缓存服务器维护自己的映射表，不同的缓存服务器的映射表是不同的。

| OBJ-ID | PROXY | $T_{LAST}$ | $T_{AVG}$ | HITS |
|---|---|---|---|---|
| www.abc | [8] | 2356 | 60 | 3 |
| www.xyz | THIS | 4158 | 73 | 25 |
| www.pq | [3] | 3562 | 85 | 8 |

图 3 映射表的结构图

在映射表中，每一行表示用户访问的一个对象。第一列 OBJ-ID 表示用户访问的对象的 ID，这里用 URL 表示；第二列 PROXY 表示对象所在的缓存服务器，若对象缓存在本机上，则为 THIS；第三列 $T_{LAST}$ 表示该对象最近被访问的时间，这里使用相对时间(用户访问的序号)来表示；第四列 $T_{AVG}$ 表示对象被访问的平均时间间隔，反映了对象的访问频率，当对象第一次被访问时，$T_{AVG}$ 的值被设为 0，以后对象每一次被访问，使用公式(1)来更新 $T_{AVG}$ 的值；第五列(HITS)表示对象被访问的次数。

$$T_{AVG}=\frac{T_{AVG}+(T_{NOW}-T_{LAST})}{2} \quad (1)$$

基于单播的适应性缓存系统中，每个缓存服务器需要维护三张映射表，单表(Single-table)、多表(Multiple-table)和缓存表(Caching-Table)。每当用户访问一个对象，系统就会为该对象生成一个记录项，该记录项根据一定的替换策略在这三张映射表之间移动，表 1 表示了这三张映射表的作用，以及记录项在三张表中移动和替换的策略。

利用式(1)计算的 $T_{AVG}$ 只反映了对象两次被访问的平均时间间隔，有可能某个对象在一段时间内被频繁访问，它的 $T_{AVG}$ 值很小，但是之后该对象不再被访问，它的 $T_{AVG}$ 也不被更新，这样，该对象虽然不再被访问，但仍然一直留在缓存中。为了避免这种情况，对于映射表每个对象，引入一个 $T_{AGE}$ 值，$T_{AGE}$ 按照公式(2)计算：

$$T_{AGE}=\frac{T_{AVG}+(T_{NOW}-T_{LAST})}{2} \quad (2)$$

每次更新映射表时，重新计算每个对象的 $T_{AGE}$ 值，若某对象的 $T_{AGE}$ 值超过了系统预定的阈值，则直接将该对象从映射表删除。

### 4.3 请求转发

当缓存服务器接收到一个用户请求时，它首先检查用户

请求的数据是否在本地缓存中，如果数据在缓存中，则直接将数据返回给用户，否则，将请求转发给其它缓存服务器或源服务器。

表1 单表、多表、缓存表的作用及替换策略

| 映射表 | 作用 | 记录项的排序 | 移动和替换策略 |
|---|---|---|---|
| 单表 | 表示最近一段时间用户访问的对象 | 按对象最近被访问的时间 $T_{LAST}$ 排序 | 对每个用户访问的新对象，都在单表中建立一个记录项，插入到单表的顶部，替换掉 $T_{LAST}$ 最小的一项。 |
| 多表 | 存放被用户访问两次或两次以上的对象 | 按照对象被访问的平均时间间隔 $T_{AVG}$ 排序 | 单表移入多表的情况：<br>当单表中的一个对象被访问两次以上时，系统更新它的访问平均时间间隔 $T_{AVG}$ 值，若该对象的访问平均时间间隔低于多表中平均时间间隔最长的对象时，该对象从单表移入多表，同时从多表中移出的数据被移入到单表中，等待下一次被命中的机会。 |
| 缓存表 | 表示已经被系统缓存的对象 | 按照对象被访问的平均时间间隔 $T_{AVG}$ 排序 | 多表移入缓存表的情况：<br>当多表中的一个对象的访问平均时间间隔低于缓存表中平均时间间隔最长的对象时，该对象从多表移入缓存表，同时从缓存表中移出的数据被移入到多表中 |

请求转发要避免无限制转发和产生环路两种情况。为此，缓存服务器在转发的用户请求中附加一个转发的地址列表，每个转发请求的缓存服务器都将自己的地址附加到地址列表中，当转发地址列表的长度超过了系统规定的最大转发次数，或同一地址在列表中出现两次（产生环路）时，请求将被直接转发给源服务器。同时，页面回传时，也根据地址列表信息使数据沿着原路径返回给用户。

由于请求转发是根据历史经验数据来转发的，有可能当一次用户请求建立起一条转发路径后，下一次同样的请求沿着该路径转发时，原来缓存的数据已经被替换或缓存服务器发生了故障，因此，等待相应的服务器接收到“请求失败”信息或接收不到任何应答，则该服务器应该随机选择另外的缓存服务器来转发用户请求。

请求转发的算法可以描述如下：

```
begin
    存储用户请求
    if（请求转发超过系统规定的最大转发次数）or（出现循环）
            直接将请求转发到源服务器
    else
            if(对象在缓存表中)
                  if (PROXY＝THIS)//对象在本地缓存
                        将数据返回给客户；
                  else          //对象在其他缓存服务器中
                        将请求转发给 PROXY 对应的缓存服务器；
                  endif
            else
                 if(对象在单表或多表中)
                    if(PROXY＝THIS) //对象不在本地缓存，且不存在往后转发路径返回“请求失败”信息；
                    else          //存在往后转发路径
                        将请求转发给 PROXY 对应的缓存服务器；
                    endif
                 else
                    随机选择一个缓存服务器执行请求转发
                 endif
            endif
    endif
    更新映射表（单表、多表、映射表）；
end
```

### 4.4 页面回传

当某一服务器响应用户请求后，用户请求的页面将沿着原路径返回给用户。沿途每经过一个缓存服务器，该缓存服务器根据单表和多表的访问记录情况，各自独立决定是否缓存该页面。

页面返回算法描述如下：

```
begin
if(返回的对象在多表中)
    if(该对象的访问平均时间间隔 T_AVG 值小于缓存表中 T_AVG 值最大的对象)
        缓存该对象，将该对象的访问记录项移入缓存表，相应
            PROXY 值置 THIS；
            原来缓存表中 T_AVG 值最大的对象被替换，移进多表；
    endif
else if（返回的对象在单表中）
        if（该对象的访问平均时间间隔 T_AVG 值小于多表中 T_AVG 值最大的对象）
            将该对象的访问记录项移入多表；
            原来多表中 T_AVG 值最大的对象被替换，移进单表；
        endif
endif
将对象沿原路返回；
end
```

通过请求转发和页面回传机制使得系统可以自适应地分散热点数据和均衡负载。当一个对象被频繁访问时，经过若干次映射表的更新，它会被移入到缓存表，被系统缓存。一个对象被访问越频繁，它的副本就越多，它离用户端越近。这样，热点数据就被分散到多个缓存服务器中，起到了负载均衡的作用。

**总结** 适应性缓存是协同缓存的一个研究分支，目前，协同缓存大多数依赖于管理人员的经验和手工调整，来达到优化配置代理缓存服务器的目的。适应性缓存则可以脱离人工干预，自适应地调整和优化缓存数据的分布，达到均衡负载，防止由于突发热点数据造成的网络拥塞等效果，因而具有较好的研究和应用前景。

本文详细介绍了基于组播和基于单播这两种适应性缓存系统，这两种系统各有其优缺点。基于组播的适应性缓存将缓存服务器组织成相互交迭的组播组，不需要维护路由信息；组播组可以根据系统用户的数量和负载情况自适应地组织和调整。但是基于组播的适应性缓存在当前的 Internet 环境中较难实现。基于单播的适应性缓存实现起来比较简单，但需要利用映射表维护路由信息，路由信息是根据历史访问情况生成的，不一定是最佳的传播路径。同时，基于单播的适应性缓存虽然考虑了整体系统的热点数据自动复制和负载均衡，但是却未考虑单个缓存服务器的负载状况，仍然可能出现部分服务器负载很重而另一部分服务器很空闲的情况。

目前，关于适应性缓存的理论研究和实践工作都不完善。从研究的角度看，适应性缓存的研究有以下一些方向：(1)适应性缓存的体系结构：使用分布式还是集中式的体系结构来构建缓存系统[13,14]；(2)自适应的缓存数据分布算法：前面讨论的两种适应性缓存系统都有各自的缓存数据分布策略，但未必是最优的，采用更好的缓存数据分布策略可以达到更高的缓存命中率[15]；(3)适应性缓存的负载均衡问题：主要研究在适应性缓存系统中，如何调度和管理各独立的代理缓存服务器，使得系统有更大的吞吐量[16,17]；(4)多媒体数据的缓存和复制问题：多媒体文件与普通 web 页面文件不同，一般比

较大，因而不适宜将整个文件缓存，一般采用分片缓存的解决办法[18]。

以上列举的是适应性缓存比较有代表性的几个研究方向，在更进一步的工作中，适应性缓存的研究还可以考虑结合目前流行的 P2P，Grid 等体系结构，构建更高效的适应性缓存系统。

## 参 考 文 献


1 Zhang L，Floyd S，Jacobson V. Adaptice Web Caching. In：Proc. of the NLANR Web Cache Workshop，1997
2 Michel Scoot，Nguyen Khoi，Rosenstein Adam，Zhang Lixia. Adaptive Web Caching：Towards a New Global Caching Architecture. In：Third Intl. Cachinig Workshop，June 1998
3 Kaiser M J，Tsui K C，Liu J. Adaptive distributed caching. In：Proc. of the 2002 Congress on Evolutionary Computation，2002
4 Kaiser M J，Tsui K C，Liu J. Adaptive distributed caching with minimal memory usage. In：Proc. of Simulated Evolution and Automated Learning Conf. 2002
5 Kaiser M J，Tsui K C，Liu J. Self-organized Autonomous Web Proxies. AAMAS，2002
6 Mohammad，Salimullah，Raunak. A Survey of Cooperative Caching：[Technical report]. December 1999
7 Buchholz S，Schill A. Adaptation-Aware Web Caching：Caching in the Future Pervasive Web. In：Proc. of the GI/ITG Fachtagung Kommunikation in Verteilten Systemen (KiVS 2003)/ Leipzig，Feb 26～28，2003
8 Fox A，Gribble S D，Brewer E A，Amir E. Adaptation to Network and Client Variability via On-Demand Dynamic Distillation. In：Sixth Intl. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS VI)，Cambridge，MA，Oct. 1996. ACM
9 Gramacy R B，Warmuth M K，Brandt S A，Ari I. Adaptive caching by Refetching. In Advances in Neural Information Processing Systems (NIPS) 2002
10 Danzig P B，Hall R S，Worrell K J. A case for caching file objects inside internetwork. In ACM SIGCOMM，Sep. 1993
11 Abrams M，Standridge C R，et al. Caching proxies：Limitations and potentials. In：4th Intl. World Wide Web Conf. Boston，USA，www.w3.org/Conference/www4/Papers/155，1995
12 Bestavros A，carter R L，Crovella m E. Application-level document caching in the internet. In：Second Intl. Workshop on Services in Distributed and networked Environments，1995
13 Mic B C，Danzig P B，et al. The harvest information discovery and access system. In：Second Intl. World Wide Web Conf. October 1994
14 Wang Zheng. Cachemesh：A distributed cache system for world wide web. In：2nd NLANR Web Caching Workshop，June 1997
15 Calvert B K，Zegura E. Self-Organizing Wide-Area Network Caches. In：Proc. of IEEE Infocorn'98，San Francisco，CA，March 1998
16 Kaiser M J，Tsui K C，Liu J. Self-organized autonomous web proxies. In：Proc. of the First Intl. Joint Conference on Autonomous Agents and Multiagent Systems，July 2002. 1397～1404
17 Tsui K C，Liu J，Liu H L. Autonomy Oriented Load Balancing in Proxy Cache Servers. Web Intelligence：Research and Development，First Asia-Pacific Conf.，WI 2001. 115～124
18 Tran D A，Hua K A，Sheu S. A new caching architecture for efficient video services on the internet. In：IEEE Symposium on Applications and the Internet，Orlando，FL，USA，2003. 29


（上接第5页）

解码和 DAB(Digital Audio Broadcasting)标准相结合，首次实现了快速移动交通工具上无扭曲接收电视节目。但是，无线带宽非常有限(即使是3G)，通过无线广播提供多媒体信息仍是一个新课题，需要我们深入研究相应的调度策略，从同步的调度、同步的反馈技术、基于网络的模式、基于缓存的模式四个方面来处理网络抖动、终端系统抖动、时钟漂移、网络条件变化，以及帧内和帧间同步，设计3G 无线移动网络以及数字化广播电视网络环境中的同步协议，以充分利用有限的无线带宽，在可接受的访问和调谐时间范围内提供优质的多媒体服务。

### 3.3 用户界面

目前，传统数字图书馆系统通常都采用 B/S 结构，用户通过 Web 浏览器访问数字图书馆资源。但是，对手持移动计算机来说，未来发展趋势是更小、更易于携带和使用，屏幕尺寸也会进一步减小，使用传统 Web 浏览器作界面显然不合适。目前，手持移动计算机的输入方式主要有笔输入、语音输入、特制键盘输入等三种。随着移动设备输入技术的不断发展，针对主流输入方式，在非常有限的显示区域内设计类似于 Web 浏览器、非常友好的用户界面是我们面临的又一个新挑战。可在现有语义 Web 和个性化服务领域研究成果的基础上，进一步研究动态自适应、设备无关及个性化的 Web 处理机制。首先提出一种普遍适应各种移动设备的自适应处理框架；在 CC/PP 的基础上研究设备 Profile 在无线移动网络上的传输机制；基于语义 Web 方法，研究动态自适应的 Web 处理算法，并通过建立服务质量模型，进一步提高自适应处理的有效性。

### 3.4 缓存

传统数字图书馆系统中常使用缓存来改进对静态用户的响应时间[8]。与传统分布式计算环境不同，移动计算机频繁处于断接状态，这使得当移动用户访问数字图书馆时数据缓存显得更重要，需要用缓存来支持断接时的数据连续使用。断接时间经常是可预知的，这样就可以将断接期间需要显示的数据量计算出来，并预取到缓存。比如在视频点播技术支持下，用户欣赏数字图书馆中的视频数据。当静态服务器给用户发送视频文件块时，可以一边传输视频数据，一边将收到的数据展示给用户。若在断接前已经预取了足够的数据并存于缓存中，那么断接期间的用户就可以连续欣赏而感觉不到断接的存在[5]。这就需要我们深入研究存储容量非常有限的移动计算机适合于多媒体数据传输和播放的缓存机制。

**小结** 随着移动通讯和移动计算机技术的飞速发展，移动计算已经成为现实。另一方面，各种基于海量分布式多媒体数据集成、管理与通讯的数字图书馆系统，在未来信息社会中将扮演愈来愈重要的角色。尽管数字图书馆的固有目标就是支持用户随时随地访问大量在线信息，数字图书馆的研究已十余年，但是数字图书馆移动服务系统的研究仍是国内外信息科学研究领域的全新课题。本文在讨论移动计算环境特点的基础上，给出了移动数字图书馆系统的典型体系结构，阐述了将数字图书馆服务延伸到移动计算环境时面临的关键问题以及研究进展情况，为进一步开展这方面的研究指明了方向。

## 参 考 文 献


1 Marshall C，Golovchinsky G，Price M N. Digital libraries and mobility. Communications of the ACM，May 2001
2 Madria S K，Mohania M，Bhowmick S S，Bhargava B. Mobile data and transaction management. Information Sciences，2002，141：279～309
3 Imielinski T，Badrinath B R. Mobile wireless computing：challenges in data management. Communications of ACM，1994，37(10)：18～28
4 Imielinski T，Viswanathan S，Badrianth B R. Data on air：organization and access. IEEE Transactions on Knowledge and Data Engineering，1997，9(3)：353～372
5 Bhargava B，Annamalai M，Pitoura E. Digital library services in mobile computing. ACM SIGMOD Record，1999，24(4)：34～39
6 Barbara D. Mobile computing and databases-A survey. IEEE Transactions on Knowledge and Data Engineering，1999，11(1)：108～117
7 贾焰，李霖，等. 分布式数据库技术. 国防工业出版社，2000
8 Li Chao，Xing Chunxiao，Zhou Lizhu. Using a Light-weighted Cache Pool to Leverage the Performance of Web Picture Databases，2003
9 Pitoura E，Samaras G. Locating objects in mobile computing. IEEE Transactions on Knowledge and Data Engineering，2001，13(4)：571～592
10 马洪超，李德仁. 一种定义影像相似性的新方法及其在影像检索中的应用. 武汉大学学报，2001，26(5)：405～411
11 段文娟，高文，林守勋，马继勇. 图像检索中的动态相似性度量方法. 计算机学报，2001，24(11)：1156～1162