视频服务器网络中影像对象映射问题的研究*

周笑波[1,2], 谢 立[1,2], Reinhard Lueling[3]

[1](南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093)

[2](南京大学 计算机科学与技术系,江苏 南京 210093)

[3](德国 Paderborn 大学 计算机并行中心,德国)

E-mail: zbo@uni-paderborn.de; rl@ruleling.de

摘要: 视频服务器网络中的影像对象映射问题是一种新的组合优化问题.服务器网络可以建立在基于局域网的工作站网络之上,也可以建立在广域网之上.基于对用户的服务请求模式、服务器网络的存储容量和通信带宽等因素的综合考虑,研究了服务器网络中影像对象映射问题,利用局部搜索算法给出了一套对该映射问题的解决方案.然后用一套基准集实例对给出的算法集进行验证.结果表明,在较短的计算时间内,该算法可以得到近似最优解的方案.

关键词: 视频点播;映射;服务器网络;比特率;模拟退火

中图法分类号: TP37 **文献标识码**: A

映射问题(mapping)[1]是并行和分布式处理中的一个关键问题.无论是建立一个大规模的并行处理机系统或基于局域网的工作站网络/工作站群簇,还是建立一个基于广域网甚至互联网的分布式计算系统或信息服务系统,都要涉及诸如怎样将顺序通信模式下的大量进程或数据集合理地分配到处理器网络中去的问题.以前研究的映射问题大多着重于怎样将计算密集型的进程或大规模的数据簇映射到处理器网络中,且使得处理器负载的分布尽量地均衡,从而以最有效的方式利用整个并行/分布式系统的资源来提高加速比或者系统流量.

1 视频服务器网络中影像对象映射问题

1.1 影像对象映射问题的产生

随着存储媒质和高速宽带通信网技术的飞速发展,现在已经可以通过高速宽带网络向用户系统提供高质量的影像媒体流.目前的可扩展视频服务器系统,如视频点播 VoD(video on-demand)系统[2,3]、TV-Anytime 系统[4],大多是在一个并行处理系统上实现.这种类型的服务器通常是将多个大容量的存储设备相联接,然后通过专用通信设施将存储设备联接到用户,详细的若干体系结构可见文献[5].

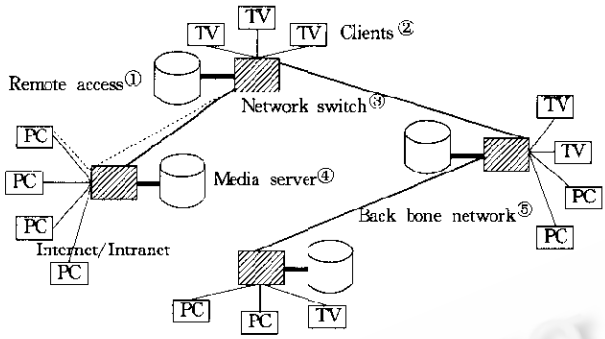
与建立一个大规模的视频服务器不同,本文所研究的视频服务器系统采用的是另一种结构.如图 1 所示,我们通过宽带主干网将一些视频服务器联接成一个分布式服务器网络,服务器网络的每个结点联结一定数量的用户.这种结构具有良好的自适应性和可扩展性,可以在局域网中或广域网上建立一个基于可扩充的视频服务器网络的影像服务系统.

显然,对联结在该服务器网络中某个服务器之上的用户来说,对存放在本地服务器上的影像对象的使用是快速且廉价的.但是,当需要使用存放于远程服务器的影像对象时,就必须付出较大的代价.因为此时,联接服务器的主干网带宽必须用来从远程服务器传输在本地申请的影像对象.若每个用户被允许使用服务器网络上提供

* 收稿日期: 1999-03-16; 修改日期: 2000-06-15

基金项目: 国家 863 高科技项目基金资助项目(863-306-2T02-03-1);德国西门子公司资助项目

作者简介: 周笑波(1973-),男,浙江人,博士生,主要研究领域为分布/并行系统,高性能 Video-on-Demand 系统;谢立(1942-),男,浙江人,教授,博士生导师,主要研究领域为分布/并行系统,高性能多媒体系统;Reinhard Lueling(1962-),男,教授,德国 Paderborn 大学计算机并行中心首席科学家,主要研究领域为分布/并行系统,高性能多媒体系统.



①远程入口,②客户端,③网络交换机,④媒体服务器,⑤主干网络.

Fig. 1 A hierarchical structure of a metropolitan TV-anytime system

图1 一个分布式视频点播服务器网络的模型

的某些影像对象的服务,那么有一个关键问题就是,在哪些服务器上,存放以什么样的比特率进行编码的哪些影像对象以响应请求呢?当考虑怎样将这些大数据量的影像对象,如MPEG压缩的媒体流高效、合理地映射到服务器网络中去的时候,该影像对象映射问题就成为构建一个高效的视频服务器网络的核心工作之一.

1.2 影像对象映射问题的特点

本文将要研究的就是在建立一个分布式的高性能视频服务器系统过程中出现的新的映射问题.影像媒体流有一个很有趣的特性,那就是一个媒体流可以用不同的比特率编码,然后传送给用户.也就是说,当一个影像对象提供给用户时,既可以根据用户期望的服务质量和花费因素来决定影像对象的编码比特率,也可以根据当前系统的存储容量和通信带宽的实际情况的限制来选择合适的编码比特率.所以,该影像对象映射问题最大的特点就是影像对象的大小(取决于编码比特率的大小)不是固定不变的,而是根据系统底层硬件,即存储能力和通信带宽的限制进行优化的.为了解决这个考虑服务质量(QoS)*的影像对象映射问题[5],我们有必要对联接服务器的主干网的通信带宽、服务器的存储容量、影像对象的编码比特率以及用户的服务请求方式(access patterns)作出合理的权衡.

1.3 相关工作

在本文中,我们研究的是通过考虑服务器网络硬件的限制来优化服务质量的影像对象映射问题.该问题与文件分配问题FAP(file allocation problem)[1]有一些类似之处.文件分配问题是一个经典的组合优化问题,与本文的研究内容最大的不同之处在于,映射于文件服务器网络的文件本身是一个长度固定的参数,而不是像本文研究的情况,媒体流的长度是根据编码比特率的大小而变化的.所以,本文研究的映射问题也可看成是文件分配问题的拓展.另外,还有一个类似的映射问题,是在一个大规模并行处理系统中,如何将全局变量指派到处理机网络中去的问题.该问题也已有了广泛的研究,与文件分配问题和本文研究的问题的对象,即文件和影像对象相比,一个根本的不同点就是那些变量的数据集要小得多.

惠普实验室的N. Venk和S. Ram在文献[6]中提及了影像对象映射问题.与本文的工作不同之处在于:(1)他们的底层系统是建立在一个大规模并行处理系统之上;(2)他们并没有对该映射问题的解决方案进行具体的讨论,而仅仅描述了一个在由多个数据源组成的可扩展服务器中复制影像对象的一些原则,着重讨论了用户服务请求的调度算法.本文提出了相应的算法来研究基于视频服务器网络中的影像对象映射的优化问题.

2 影像对象映射问题的形式化定义

2.1 视频服务器网络的模型

一个视频服务器网络可以看成是一个图G,$G=(H,E)$,$|H|=n$,即该网络有n个服务器相连.每个服务器

* 服务质量QoS应该包括代价、速率、服务延迟、抖动、缓冲区等因素,在此我们主要考虑编码速率.

均有存放影像对象的存储量限制,我们用存储量函数 $c:H\to N$ 来表示,其中集合 N 表示正整数集,该函数指派了一个整数来表示该服务器的存储量(MB).图 G 的每一条边 $e\in E$ 表示服务器网络中联接两个服务器的通信联结.我们用函数 $w_e:E\to N$ 来给每一条通信连接赋以 MB/s 为单位的带宽.

应用系统上的用户通常是通过高速网络联接到服务器上,在此,我们假设联接用户和视频服务器的网络能提供足够的带宽来获取媒体流.在服务器网络中,可能会有一些服务器并不与任何用户直接相连,而仅仅用作存储服务器,向其他服务器提供缓存;或者有一些服务器并不提供存储空间来存放影像对象,而仅仅作为服务器网络的一个网关.

对于服务器网络中存放的影像对象的集合,我们用 $M=\{a_1,a_2,\ldots,a_m\}$ 来表示.联接在视频服务器网络上的用户对影像对象的使用请求方式也可看成是一个图,$A=(V_A,E_A)$,$V_A=H\cup M$,$E_A=\{(v,a)\mid v\in V_A,a\in M\}$.这样,一条边 $(v,a)\in E_A$ 就表示联接在服务器 v 上的一些用户向服务器网络请求影像对象 a 提供的服务.

影像对象可以通过服务器系统以不同的质量提供给用户,即可以用不同的比特率进行编码.我们用集合 B 表示可能的比特率集.

2.2 映射问题的优化定义

给定一个视频服务器网络 G、影像对象集合 M、一个用户使用影像对象的方式 A、网络带宽集合 w_e 以及影像对象的编码比特率集合 B.现在的问题就是采用什么编码比特率,以将影像对象 M 映射到服务器网络 G 中去,使得每个用户的服务请求均可以得到尽可能最佳的响应.当一个用户向其联接的服务器发出请求时,若请求的影像对象就存放在该服务器上,或者从服务器网络中的某个服务器结点通过网络能以足够大的带宽将该影像对象传送到该服务器上,我们均认为该服务请求是可以响应的.

为了响应用户的请求,我们允许:

(1) 影像对象可以冗余地映射到多个不同的服务器上;

(2) 影像对象的编码比特率是可伸缩的,即其所占存储空间可以根据给定的比特率的变化而变化;

(3) 服务器网络中的影像对象可以通过网络的通信线路进行传输.

现在,综合权衡视频服务器的存储空间限制、服务器网络的带宽限制、用户可获取的影像对象的编码比特率等因素,我们形式化地定义要解决的影像对象映射问题及其优化目标.

给定一个服务器网络 $G=(H,E)$、用户使用方式 $A=(V_A,E_A)$、编码比特率集合 B.

问题是,设 $P(E)$ 是指所有集合 E 的子集,是否存在这样的映射 $\pi_a:A\to P(E)$ 和映射 $\pi_b:A\to B$,使得 $\pi_a(v,a)=\{(v,v_1),(v_1,v_2),\ldots,(v_{l-1},v_l)\}$,$l\in N$,且 $\pi_a(v,a)=v_l$,$\pi_a(v,a)=b$,并满足

$$\sum_{(v,a)\in A,\pi_a((v,a))=h}\pi_b((v,a))\leqslant c(h),\forall\ h\in H$$

且

$$\sum_{(v,a)\in A,(u_1,u_2)\in\pi_a((v,a))}\pi_b((v,a))\leqslant w_e,\forall\ e\in E,e=(u_1,u_2).$$

基于该影像对象映射问题的定义,下面我们给出该优化问题的形式化描述:

$$\mathrm{QoS}:=\sum_{(v,a)\in A}\pi_b((v,a))\to\mathrm{Maximum}.$$

3 影像对象映射问题的局部搜索算法

3.1 局部搜索及模拟退火算法

局部搜索算法是一种解决组合优化问题的有效方法.局部搜索原则不但被用来解决一些经典的图论问题,如旅行商问题 TSP(trevcling salesman problem)和图分解问题 KPP(k-graph partitioning problem)[7],而且也被用来解决一些非常具体的、有许多限制因素的应用问题,如汽车路由问题.局部搜索原则是基于给定的组合优化问题的所有可能的解决方案集上定义的邻结构集,从一个选定的起始方案(可随机选定)开始,在其可行的邻结构集中选择一个结点作为下一次搜索的当前方案.

模拟退火算法[8]是一种传统的局部搜索技术.计算一个起始方案的函数和基于该起始方案计算一个可行的邻方案的函数均是与问题相关的,而算法的终止和接受函数则与具体问题无关.在传统的"爬山算法"中,如果开

销函数的值有所改善,则该新方案将被接受.而若在一定数量的搜索后,开销函数的值没有得到改善,则算法将被终止."爬山算法"的主要缺点是往往掉入局部最优的陷阱.而在"退火算法"中,总过程由一个退火过程的温度参数控制,每一次重复搜索后,温度将根据一个给定的因子略微下调,该因子的值极为接近1.如果开销函数的值有所改善,则该方案将被接受;但当新方案造成开销函数的值变坏时,该方案仍然有$e^{-\Delta/t}$的可能被接受,其中Δ为开销函数的新值与旧值的差.所以,与"爬山算法"算法不同,"退火算法"可以跳出局部最优的限制,从而更有可能找到全局最优的解.接下来,我们将描述用来解决该映射问题的模拟退火算法集.首先,我们描述其算法集的构件以用来构建模拟退火算法.

3.2 算法集的构件

3.2.1 邻结构算法——第1阶段

基于对服务器网络的系统结构是一个全连接(clique)网络的假设,给定一个影像对象映射问题的起始解决方案,计算一个邻结构的工作可分两步进行.

第1阶段,随机选择服务器网络中的一个结点,对已映射在该服务器上的一个影像对象的编码比特率的值进行增加,或者映射上新的影像对象.该操作将可能引起已映射在该服务器之上的某些影像对象的编码比特率值的减小,或者导致某些以最小比特率映射的影像对象从该服务器上被删除,以满足该服务器的存储量限制*.以上这些对该服务器上映射方案的局部修改,还将影响到那些远程使用该服务器之上的影像对象的其他服务器,这是第2阶段的邻结构算法需要解决的问题.

我们给出3个不同的计算邻结构第1阶段的算法.算法1相对来说比较简单,先随机地选择一个服务器结点和一个影像对象,若该影像对象已经被映射到该服务器上,则增加其比特率到下一个值;否则,以最小的比特率映射到该服务器上.若该操作引起了服务器的存储空间的不足,则随机选择该结点上的影像对象,降低其编码的比特率或删除该影像对象,直到服务器的存储量限制得到满足为止.

算法1. Neighbor_1.1

a node $h\in\{1,\ldots,n\}$ and an asset $a\in M$ are choosen at random (uniform distribution);

if a has been mapped onto h **then** its bit-rate is increased to the next possible bit-rate;

if a has not been mapped onto h **then** map it with the minimal bit-rate onto h;

while (current storage load of node h exceeds $c(h)$) **do**

an asset b is choosen randomly with uniform distribution;

if b has been mapped onto h and b encoded with the minimal bit rate **then** delete b from h;

if b has been mapped onto h **then** decrease the bit-rate of b to the next possible bit-rate.

在下面的算法2中,服务器结点的选择也是随机的,但影像对象的选择则是根据该结点的用户请求方式A来随机选择的.并且比特率的增加幅度也加大了.当由于上述操作带来的服务器结点的存储空间不足时,我们同样选取服务器上某个或某几个影像对象,对其比特率进行降低或删除那些影像对象.但是,这些比特率将被降低的影像对象的选择是根据指数分布来选择,那些当前以较低比特率编码的影像对象将被给予更高的优先级.也就是说,当前以较低比特率编码的影像对象被删除的可能性将更大.

算法2. Neighbor_1.2

a node $h\in\{1,\ldots,n\}$ is choosen at random with uniform distribution;

an asset $a\in M$ and $(h,a)\in A$ is choosen at random with uniform distribution;

if a has been mapped onto h with bit-rate x **then** choose its bit-rate randomly from $[x,$ maximum$]$;

if a has not been mapped onto h **then** map it with the randomly choosen bit-rate onto h;

---

* 当然,这种算法中的比特率的改变并不是真正意义上的改变.要物理地改变一个影像对象的编码比特率,一般说来有两种途径.一种途径是,根据该影像对象的原始数据,用新的编码比特率重新进行压缩,该方法的缺点是必须保存有影像对象的原始数据.另一种途径是,先将现有的影像对象解码,然后用新的编码比特率重新进行压缩.该方法的缺点是,在解码和编码的时候会丢失一些数据,并且计算量较大.

while (current storage load of node h exceeds $c(h)$) **do**

an asset b is choosen randomly with exponential distribution according to the bit-rate;

if b has been mapped onto h and b is encoded with the minimal big-rate **then** delete b from h;

if b has been mapped onto h and b is not encoded with the minimal bit-rate **then**

decrease the bit-rate of b to the next possible bit-rate;

算法起始阶段延用算法 2 的同样策略,但当存储空间不足时,对比特率将被降低的影像对象的选择根据的不是当前编码的比特率大小,而是当前各影像对象被其他服务器远程使用的数量大小.对于当前正被较多的远程服务器所使用的影像对象来说,其被降低编码比特率,或者被删除的可能性将较小.由此,修改算法 2 中的第 6 行,我们可以得到算法 Neighbor_ Ⅰ.3(略).

3.2.2 邻结构算法——第 2 阶段

在计算邻结构的第 2 个阶段,对于属于服务器网络结点集 H 上的每一个服务器 v,根据连接在其上的用户的请求模式,对于所有的 $(v,a)\in A$,若影像对象 a 还没有映射到服务器 v 上,那么就必须考虑怎样从某个存放有影像对象 a 的远程服务器上将 a 路由到 v 上,即必须将这些边 (v,a) 映射到服务器网络中去.接下来,我们给出该路由算法 Neighbor_Ⅱ.

算法 3. Neighbor_Ⅱ

for $i:=1$ to n **do** $bw\_in(i):=0$;

for $h:=1$ to $n(h\in H,n=|H|)$ **do**

for all $a\in M$ with $(h,a)\in A$, and a has not been mapped onto h;

let K be the set of all nodes that store a copy of asset a;

while not successful and $K\neq\varphi$ **do**

choose a node $h'\in K$ with bit-rate of asset a on h' is the maximum;

$K:=K-\{h'\}$;

if $bw\_in(h')+$ bit-rate of a on node $h'\leqslant w_c(h',h)$ **then**

successful:=true; asset a is streamed from node h' to h;

$bw\_in(h'):bw\_in(h')+$ bit-rate of a on node h';

if not successful **then** access to asset a from node h is not possible temporally.

3.2.3 起始方案算法

为了方便下面的讨论,我们假设服务器网络中的存储容量和网络带宽均足以根据请求模式 A,将每个影像对象以最小的编码比特率随机地映射到服务器网络中.接下来,我们给出两个计算初始方案的算法.

算法 4. 起始阶段算法 Initial_1

for $a:=1$ to $m(a\in M,m=|M|)$ **do**

map asset a onto a random node h with $(h,a)\in A$ and h provides sufficient storage capacity to store a with the minimal bit-rate;

if no such node h exists **then** the initialization fails;

use algorithm Neighbor_Ⅱ to assign the access graph onto the server network.

显然,使用算法 4 给出的算法 Initial_1,各服务器结点的存储容量和网络的带宽有可能没有被充分利用.在接下来的算法 5 所给出的算法 Initial_2 中,我们通过提高已经被映射到服务器网络上的影像对象的比特率来解决该问题.

算法 5. 起始阶段算法 Initial_2

use algorithm Initial_1 to compute the initial assignment;

for $h:=1$ to $n(h\in H,n=|H|)$ **do**

for $a:=1$ to $m(a\in M,m=|M|)$ **do**

if $(h,a)\in A$ and a has not been mapped on h yet **then**

map a onto h with minimal bit-rate if sufficient storage capacity is available on h;

while (not all assets mapped on h are encoded with the maximal bit-rate) and

(storage capacity of h is not exceeded) **do**

choose a random asset a mapped on h and increase its bit-rate to next higher one.

3.3 一个解决映射问题的模拟退火算法

现在,我们已经有两个计算起始方案的算法,3个计算邻结构第1阶段的算法和1个计算邻结构第2阶段的算法.所以,实际上我们有6个不同的算法可以用来构建模拟退火算法.在下面的算法6中,我们用参数 t_0,t_∞,a 来控制该退火算法的过程.其中函数 cost()用来计算该影像对象映射方案所能得到的服务质量.

算法 6. 一个解决影像对象映射问题的模拟退火算法

$t:=t_0$; $s:=$ initial_x, $s_{opt}:=s$;

while $t>t_\infty$ **do**

$s':=$Neighbor_I.$x(s)$;

$s'':=$Neighbor_II.$x(s')$;

if $cost(s'')>cost(s)$ **then** $s:=s''$;

if $cost(s'')<cost(s)$ **then** $s:=s''$ with probability $e^{(cost(s)-cost(s''))/t}$;

if $cost(s)>cost(s_{opt})$ **then** $s_{opt}:=s$;

$t:=t\cdot a$.

4 性能评价

4.1 质量上限(upper bound)

服务器网络中任意一个服务器能得到的服务质量取决于该结点的存储容量和相连的通信线路的带宽限制.若该通信带宽或服务器的存储容量不能被该服务器上的用户请求方式所全部利用,则一个服务器结点可得到的服务质量是由该结点上的影像对象的请求数量决定的.

综合以上两个因素,对于一个给定的服务器网络、用户的请求模式、存储器容量和网络带宽,我们给出该服务质量的上限为

$$\mathrm{QoS}\leqslant\sum_{i=1}^{n}\min\left\{c_i+\sum_{j=1,j\neq i}^{n}w_e((j,i),\max_{b\in B}b\cdot|(i,j)\in E_A|\right\}.$$

4.2 基准集的定义

基于服务器网络的系统结构是一个全连接网络,每个服务器结点有固定的存储量限制,每两个服务器结点间的通信联接有固定的通信带宽的假设.我们现在定义一个由服务器结点数 n、服务器的存储量 c、每个通信联接的通信带宽 w_e、影像对象的个数 $m=|M|$、用户的请求模式 A(以一致分布或指数分布)以及可用来编码的比特率集合 B 等因素构成的基准集.

表1给出了具体的基准集,所有的请求模式都是随机产生的.在以 R 开头的基准集中,每个服务器上的用户请求影像对象的方式是根据一个给定的一致分布来决定的.在以 E 开头的基准集中,每个服务器上的用户请求影像对象的方式是根据一个指数分布来决定的.在实现一个视频媒体服务所的时候,后者很好地反映了新旧影片对用户来说因兴趣度不同而导致的服务请求模式不一致的情况.

Table 1 Definition of Benchmark Instances in a Hierarchical Server Network

表1 一个全连接网络的基准集实例的定义

Benchmark class①	n	m	c	w_e	B	Access pattern②
R_32_256_750_100_0.5	32	256	750	100	{5,10,15,20,50}	Random③ $p=1/2$
R_32_256_1000_100_0.5	32	256	1000	100	{5,10,15,20,50}	Random $p=1/2$
R_32_256_1500_100_0.5	32	256	1500	100	{5,10,15,20,50}	Random $p=1/2$
E_32_256_250_100_0.5	32	256	250	100	{5,10,15,20,50}	Exponential④

①基准集,②访问模式,③随机值,④指数级.

4.3 性能分析与讨论

下面,我们对不同的算法集和基准集的性能做详细的分析及比较.表2给出了由模拟退火算法得出的服务质量和第4.1节给出的各个基准集的服务质量上限.表中给出了4个不同的基准集和第3.2节中给出的6个算法组合而成的6个不同的退火算法,并比较了得到的服务质量及服务质量上限之间的差异程度(百分比).因为退火算法是随机性的,所以对于每个基准集,我们产生5个实例,并用模拟退火算法进行了10次计算,并取其平均值.所有退火的参数设定是一样的,$t_0=10$,$t_\infty=0.0001$,$a=0.9995$.

Table 2 The performance of algorithm

表2 算法的性能结果

	Amulated annealing algorithms①						Bound②
Initial_Neighbor	1	1	1	2	2	2	
phrase I③	1	2	3	1	2	3	
*R*_32_256_750_100c0.5	108 690	112 553	113 425	109 772	116 980	118 224	123 200
	13.34%	9.45%	8.61%	12.23%	5.31%	4.20%	
*R*_32_256_1000_100_0.5	118 265	122 540	123 348	119 329	124 992	124 873	131 200
	10.93%	7.06%	6.36%	9.95%	4.97%	5.07%	
*R*_32_256_1500_100_0.5	132 883	140 195	140 856	134 331	141 552	140 565	147 200
	10.77%	4.99%	4.50%	9.58%	3.99%	4.72%	
*E*_32_256_250_100_0.5	12 390	12 563	12 433	12 690	12 930	12 754	13 300
	7.34%	5.86%	6.97%	4.8?%	2.86%	4.28%	

①模拟退火算法,②上限,③邻结点阶段I.

从上述性能分析中我们可以看到,服务质量的上限和模拟退火算法得到的服务质量之间的差别很小,变化范围从大约10%~3%不等.因为这是将算法得到的服务质量与其上限相比,而不是与最优方案相比,所以我们认为,该算法可以提供很好的解决方案.

将文中给出的不同算法进行比较,我们还可以发现,具有智能性的邻结构阶段1的算法Neighbr_1.2和Neighbr_1.3与津单的邻结构阶段1算法Neighbr_1.1相比较,更有可能得到较佳的结果.即对那些比特率将被降低的影像对象的选择,根据当前比特率编码值大小的指数分布进行选择,或根据当前各影像对象被其他服务器远程使用的数量大小的指数分布进行选择,比纯粹的随机选择要好得多.通过对起始方案算法的性能比较,我们发现,充分利用服务器存储量和网络带宽的算法Initial_2能比Initial_1提供更好的解决方案.

5 结论和今后的工作

作为视频服务器网络系统的设计和实现的关键性问题之一,影像对象的映射问题是一个必须权衡多个因素的组合优化问题.本文详细描述了该映射问题的模型,给出了解决该问题的模拟退火算法的构件.基于解决组合优化问题的一种有效途径——局部搜索算法,通过对基准集的性能进行模拟,我们在较短时间内就得到近似最优的映射方案,并可以预计,基于服务器网络的并行计算能力,该算法集的运行时间还可以大大缩短.性能分析的结果还表明,局部搜索算法的邻结点结构的选择对算法的性能有很大的影响.

本文给出的算法集静态地将影像对象映射到视频服务器网络中,同时考虑了网络通信带宽、存储器容量限制、用户的服务请求方式等因素的权衡来优化服务质量.在实际系统中,用户请求方式动态地改变,影像对象动态重映射的工作也就必不可少.影像对象的动态映射算法必须作出对影像对象迁移/上载的代价以及附加的通信开销与得到的更价服务质量之间的权衡,这些将是我们下一步的工作.

References:

[1] Dowdy, L.W., Foster, D.V. Comparative models of the file assignment problem. Computing Surveys, 1982,14(2):34~56.

[2] Anderson, D.P., Osawa, Y., Govindan, R. A file system for continuous media. ACM Transactions on Computer Systems, 1992,10(4):311~337.

[3] Dan, A., Kienzle, M., Sitaram, D. Dynamic policy of segment replication for load-balancing in video-on-demand servers. ACM Multimedia Systems, 1996,4(3):112~121.

[4] Zhou Xiao-bo, Reinhard Lueling, Xie Li. Heuristic solutions for a mapping problem in a TV-anytime server network, parallel and distributed processing. In: Rolim, J. ed. Proceedings of IPDPS 2000 Workshops, Vol 1800. Lecture Notes in Computer Science, Berlin: Springer-Verlag, 2000. 210~217.

[5] Gomez, F. C., Lueling, R. A parallel continuous media server for internet environments. In: Peter, S. ed. Proceedings of the International Conferences on High-Performance Computing and Networking (HPCN Europe'98), Vol 1401. Lecture Notes in Computer Science, Berlin: Springer-Verlag, 1998. 78~86.

[6] Venkatasubramanian, N., Ramanathan, S. Load management in distributed video servers. In: Simon, S. ed. Proceedings of the International Conference on Distributed Computing Systems (ICDCS). Baltimore, MD: IEEE Computer Society, 1997. 31~39.

[7] Diekmann, R., Lueling, R., Monien, B. Combing helpful sets and parallel simulated annealing for the graph-partitioning problem. Parallel Algorithms and Applications, 1996,8(1):61~84.

[8] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P. Optimization by simulated annealing. Science, 1983,220(4598):671~680.

Media Mapping in Video Server Networks

ZHOU Xiao-bo[1,2], XIE Li[1,2], Reinhard Lueling[3]

[1](*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210093, *China*);

[2](*Department of Computer Science and Technology, Nanjing University, Nanjing* 210093, *China*);

[3](*Pader Center for Parallel Computing, University of Paderborn, Germany*)

E-mail: zbo@uni-paderborn.de; rl@ruleling.de

Received March 16, 2000; accepted June 15, 2000

Abstract: Media mapping in video-on-demand server networks is a new combinatorial optimization problem. The network of video servers can be implemented on the top of a closely connected network of workstations or on a wide area network of servers. This paper addresses the media mapping problem, taking the user access patterns, overall storage capacity and communication bandwidth limitations of the server network into account. A number of methods based on local search algorithms for the solution of the mapping problem are proposed and verified by use of a set of benchmark problems. The simulations show that these heuristic solutions can achieve nearly optimal solutions in a short computational time.

Key words: video-on-demand; mapping; server network; bit-rate; simulated annealing