

分 类 号 \_\_\_\_\_

学号 M200771388

学校代码 10487

密级 \_\_\_\_\_

# 华中科技大学

# 硕士学位论文

## CDN 内容分发网络优化方法的研究

学位 申 请 人： 王 玮

学 科 专 业： 通信与信息系统

指 导 教 师： 刘文予 教授

答 辩 日 期： 2009 年 5 月 25 日

**A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Engineering**

**Research on the optimization of content delivery  
strategy in CDN**

**Candidate : Wang Wei**

**Major : Communication and Information System**

**Supervisor : Professor Liu Wenyu**

**Huazhong University of Science & Technology**

**Wuhan 430074, P.R.China**

**June, 2009**

## 独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到，本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

## 学位论文授权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本论文属于 ☐ 保密，在\_\_\_\_\_年解密后适用本授权书。  
☐ 不保密。

(请在以上方框内打“√”)

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

## 摘 要

内容分发网络是为了改善网络负载不均,增加用户满意度而发展起来的热门网络技术。目前最典型的内容分发网络包括点对点网络(P2P)与内容发布网(CDN)两种,文章首先简要介绍了点对点网络与内容发布网的概念和发展现状,并详细给出了内容发布网网络中的几种主要技术:内容路由、内容分发、内容管理。

本文的主要工作围绕商业应用中的PUSH模型展开。它与人研究的PUSH技术相比,有两个特点。一是更高的时效性:至少在一个触发时隙内能完成求解;二是求解的连续性。

与一般的内容分发模型相比,本研究与人文章主要有四点不同:一是简化拓扑到以节点为单位。把一片代理区域内所有用户和代理服务器看做一个统一节点,不考虑具体的边缘服务器到用户的路由,这样做极大的提高了模型规模的可扩展性,而且还把具体的路由技术从内容分发技术中剥离开。二是确立了以内容发布网运营商为服务对象,完全从内容发布网运营商的角度出发考虑问题,不仅使模型的目标函数和限制条件更完备,也更有实际说服力。三是采用内容聚类的思想,把多个热度相似的内容组合成一个频道,在调度时看成一个整体。四是在求解多目标函数的模型时,本文没有把两个目标放在一起加权求和,而是假设两种应用环境,分成两个子模型分别求解。

最小化网络带宽问题的数学模型是一个标准 0-1 规划模型,本文不仅用标准的隐数法求解此模型,还给出了自己的启发式解法,并通过大量实验表明,在线性求解时间内,启发解的性能达到了最优解的 95%。而最小化服务器损耗问题的数学模型属于混合整数规划,变量数目很多时根本无法求最优解。本文用启发式算法求解,并用估计的最优解的上限与之对比,以反映启发解的性能。实际仿真时做出了两个小的改进,通过对比实验表明,性能比以前调高 10%左右。

**关键词:** 内容分发网络, 内容分发, 最优化理论

## Abstract

Content Distribution Network is a hot technology aimed to improve the network load balance, increase customer satisfaction. The most typical content delivery network are P2P and CDN. We have given a brief introduction to both P2P and CDN first, then in detail discussed about three technology in CDN: content delivery, routine, and management.

This thesis mainly studies one type of content delivery strategy for CDN, named realtime-PUSH strategy. Compare to normal PUSH strategy, it has two characters: first, it must be much more efficient in time; second, the environment is a continuing process.

Compared to former research, our model has four differences: 1, We have brought node clustering structure to our model, which means we no longer care about the different behaves between proxy, client, or source, but combine them as a whole node. In this way, the scalability of model is greatly improved. 2, The main purpose of this research is to help CDN provider, so that the model has precise goals and limits, and is much closer to reality. 3, All the contents are clustered as some bucks, called channel. 4, When facing a extremely complicate problem, we depart the problem into two sub-problems, and then gives two different model to fix out each.

The goal of the first model is to minimize the bandwidth consuming. This model has turned out to be a standard 0-1 linear programming. Later in this thesis, we not only give the standard method to solve this problem, but also give a heuristic algorithm. Through a huge number of experiments, we prove that the heuristic algorithm is much more efficient in time and its solution is about 95% close to the optimal one. The goal of the second model is to minimize the exchange of all channels on all proxies. This model proved to be a mixture nonlinear programming problem, and is very hard to get its optimal solution. So we have to give a heuristic algorithm instead. In order to estimate the performance of this heuristic algorithm, also give a estimation on the optimal solution. At last made two small changes on the algorithm, under certurn circumstances, the performance can be improved by 10%, with the price of 2 or 3 times of time consuming.

**Key words:** Content Distribution Network, Content delivery, Optimization

### 中英文对照表:

P2P	----- (Peer to Peer)点对点网络
CDN	----- (Content Distribution Network)内容发布网
Proxy	----- 代理服务器
Content	----- 内容
Replicas	----- 内容镜像
Cache	----- 缓存技术
PUSH	----- 指在CDN网络中，中心服务器将内容主动放入边缘服务器的一种技术
PULL	----- 与PUSH对应，指边缘服务器被动请求中心服务器内容的技术
Backbone	----- 骨干网络
Miss	----- 用户请求无法由本地代理应答

# 目 录

摘 要 .....	I
Abstract .....	II
1 绪论	
1.1 研究背景 .....	(1)
1.2 P2P 网络概述 .....	(2)
1.3 CDN 网络简述 .....	(4)
1.4 (论文结构) .....	(6)
2 CDN 网络技术研究现状	
2.1 CDN 的主要技术 .....	(8)
2.2 最优化理论简述 .....	(10)
2.3 CDN 网络与最优化模型结合现状 .....	(13)
3 实时内容调度模型研究	
3.1 前提条件与说明 .....	(20)
3.2 数学模型的建立 .....	(21)
4 模型求解与仿真	
4.1 模型求解 .....	(28)
4.2 实验与仿真 .....	(33)
5 总结与展望 .....	(43)
致 谢 .....	(45)
参考文献 .....	(46)

# 1 绪论

## 1.1 研究背景

在因特网发展的早期，网络服务以简单的web应用为主，网络流量相对于可用带宽来说很小，当时的内容提供商（ICP: Internet Content Provider）仅采用我们熟知的C/S(Client/Server)模式提供服务就足够了。简单的C/S模型包含两个部分：服务器端和客户端。服务器端由内容服务商提供并配置，一旦运行后专门提供下载等服务。客户端指普通网民，他们提出访问需求后，会被直接定位到相应服务器，收到访问请求的服务器端会根据设定好的简单规则来应答<sup>[1]</sup>。服务器和客户实际上是简单地端到端的模式。图1.1是最简单C/S的体系结构图。

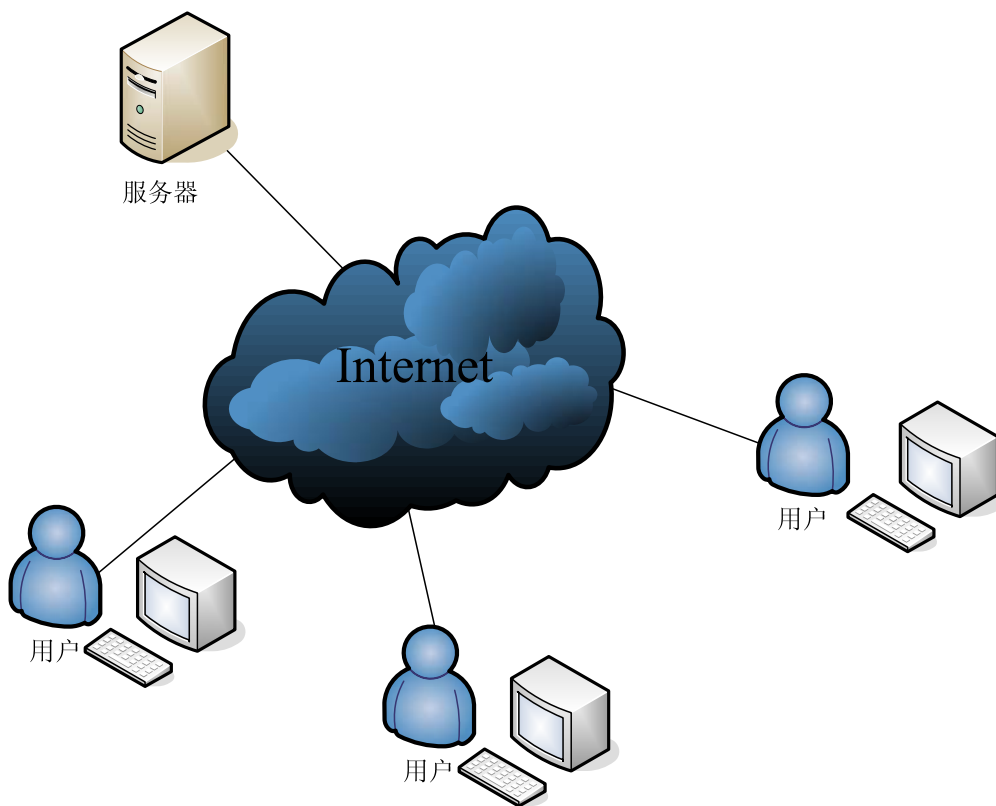


图1.1 简单的C/S结构图



而互联网的发展速度是惊人的，据中国互联网络信息中心发布的《第23次中国互联网络发展状况统计报告》显示<sup>[1]</sup>，截至2008年底，我国网民数达2.98亿，权威的国家部门发布的统计数据上来看，三大门户网站，新浪、搜狐、网易发展势头强劲，以多媒体网络应用为主打的如优酷、土豆等也迅速崛起。随着宽带网络的普及，网络流媒体、网络游戏等新的增值业务日益流行，传统的C/S模型面对如此大的网络流量显得有些乏力。另外某些知名网站的跨区域访问也是日益频繁，当远距离的用户访问这些服务器时，要保证其质量和速度也是相当困难。在原有网络结构体系下，用户访问有时会变得十分缓慢，令人难以接受<sup>[2,3]</sup>。

从客观上分析，访问速度下降的原因并不是整体网络资源已经饱和，而仅仅是服务器端及其接入网处出现了瓶颈。由于C/S模式本身端到端的特性，网络访问对于带宽的要求也呈现端对端的形式。当大量用户同时访问同一台服务器时，单个用户所用的出口带宽并不多，但是服务器端的带宽和负载消耗会变得很高。当有吸引网民关注的事情发生时尤为明显，这让访问量呈现高度集中的趋势，即便整个网络大多数节点的性能良好，而仅被过量访问的少数服务器端这一瓶颈就直接导致了拥塞和质量下降。很显然仅仅依靠增加服务器数量和接入带宽是不合理的，最恰当的方式是将用户集中的访问量分散到整个网络，尽量合理的利用已有网络资源。

为了解决上述问题，尽可能在现有网络资源下满足用户需求，内容分发网络技术便产生了（这里的内容即content，指各类在网络上被用户请求和转发的流媒体，网页等文件）。目前主流的内容分发网络技术有两种：P2P 和 CDN<sup>[4]</sup>。

## 1.2 P2P 与 CDN 对比

P2P（Peer to Peer）和CDN（Content Distribution Network）都是为了解决上述问题而产生的新的网络技术。P2P即点对点网络，点即用户节点<sup>[5]</sup>，其主要特点是用户节点之间的对等性，去掉了服务器的概念。对等网络中每个节点的地位基本都是相同的：既充当服务器，存储相应内容，为其他节点提供转发、查询等服务；同时也是客户，享用其他节点提供的服务。当然为了查询和组织网络的方便，有的P2P网络

中依然存在中心服务器节点，但是它们的功能已经不在于提供内容服务了。

P2P能确实有效地起到增加用户满意度和减轻服务器负担的作用。从服务器端角度看，紧张的访问量得到了分流，以前本来需要提供所有的内容下载服务，现在只需出相应索引帮助用户查询即可。从用户的角度看，P2P使得网络上的用户可以实现面对面沟通交流，而不用经过特定网站和服务器的限制，还可以实现多点多用户交互。除了诸如迅雷看看，PPlive等商业软件的崛起外，P2P网络技术还在很多方面得到了应用：主要体现在分布式系统，协同式工作环境，计算共享等各个领域<sup>[6]</sup>。简单的P2P网络结构图如1.2。

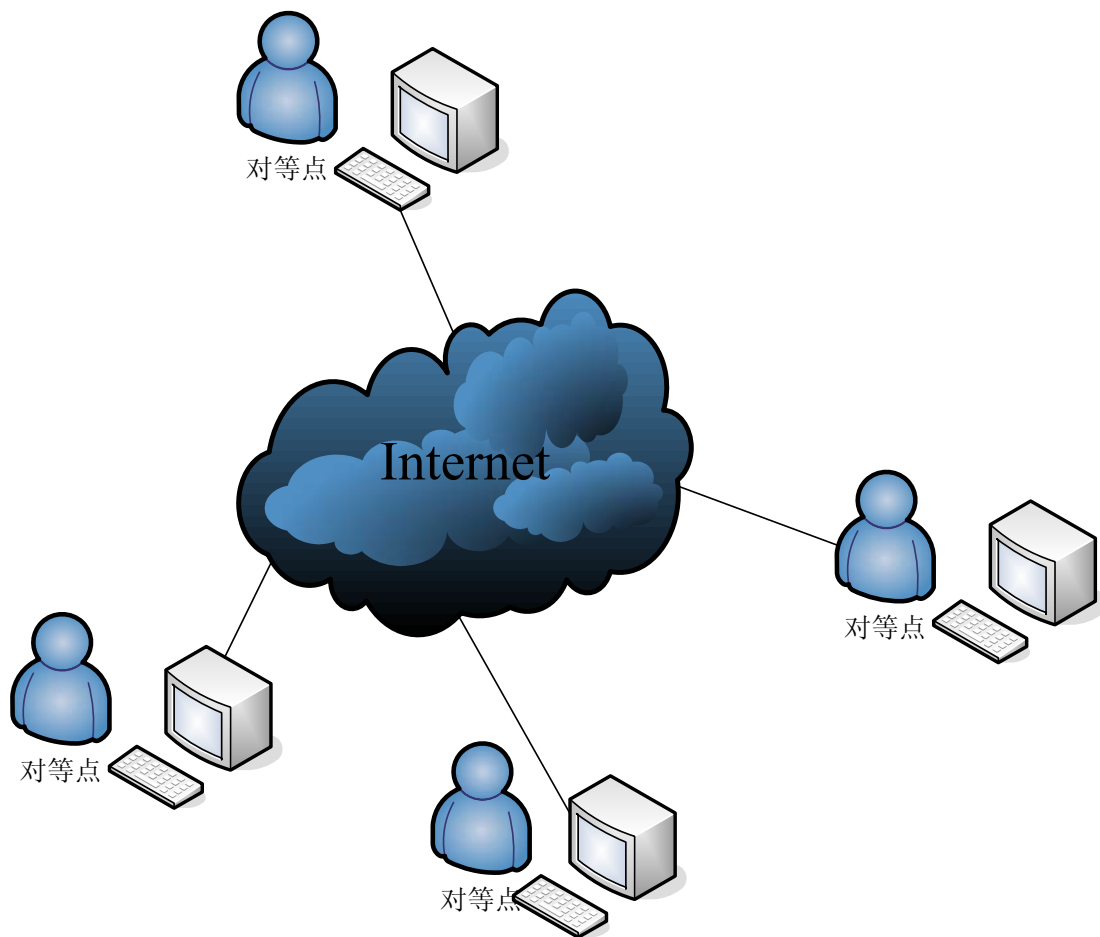


图1.2 简单P2P网络结构

CDN更类似于一个分布式多层次的C/S结构。它通过在用户和源服务器之间加入CDN运营商的代理服务器来解决带宽问题以及服务质量问题。当用户发送关于某内

容的请求时，若此内容已接受CDN运营商的服务，则此请求被域名解析服务器重定向到离该用户最近的CDN代理服务器上，由该服务器对用户进行服务。从服务器端来看，源服务器只要接受CDN服务，就不用再提供服务给用户，唯一要做的工作就是把内容发给各个边缘服务器。从用户端来看，CDN运营商的这些代理服务器离用户更近一点，而且带宽更充足，因此服务质量自然更好。

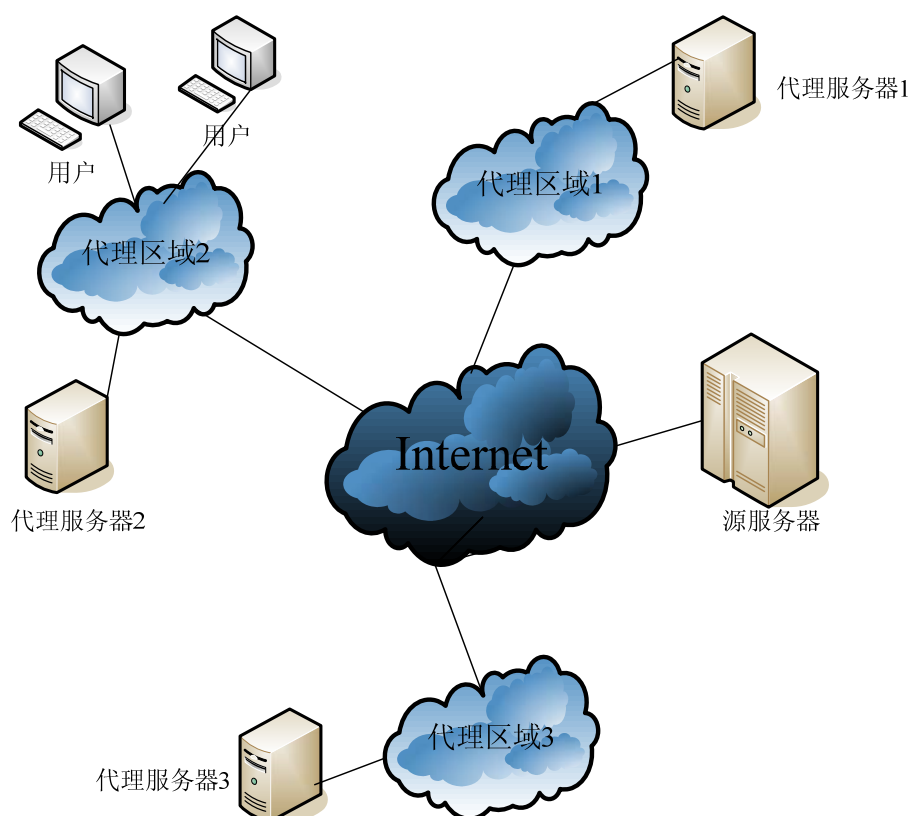


图1.3 CDN网络结构

相比于传统的简单C/S模式，在带来更好服务的同时，它们也或多或少存在一些问题。

- 使用费用

P2P在原始的网络基础上并没有附加任何硬件设施，它只需要用户应用一套新的软件协议，因此用户使用起来是完全免费的。CDN网络是一种纯商业应用，运营商需要的投入是很大的，诸如架设服务器和日常维护的开销等；而从用户角度来看，下载或流媒体的服务和之前费用一样，基本感受不到CDN运营商的存在。

### ● 盗版传播<sup>[1]</sup>

由于P2P服务大多数是提供免费下载的，与版权问题的冲突也是不可避免的。即使没有直接制作盗版，不可否认地是P2P共享软件的繁荣也大大加速了盗版媒体和软件的传播，某种意义上说比小心翼翼发行盗版者更大程度地破坏了知识产权<sup>[7]</sup>。CDN网络则不存在问题，所有资源的使用都还是由内容提供商负责，与CDN运营商没有关系。

### ● 资源浪费

P2P网络中除了正常的下载外，还有相当一部分网络带宽浪费在使用P2P协议时的洪泛资源查询中，这些疯长的网络流量会占用太多的接入网带宽，从而让整个网络拥塞。某种程度上说，ISP提供商是相当害怕P2P的。CDN正好相反，不但没有加重反而减轻了中心设备和骨干网络的压力，这是因为通过CDN可以将内容服务从原来的单一中心结构变为分布式的结构。许多原来必须要跨城域ISP的内容请求，现在由于在本地就有此内容的拷贝，就能在本地解决，减轻骨干网压力，因此CDN是很受ISP商喜爱的。

### ● 安全隐患

网络安全问题随着因特网的发展也变得日益严重。P2P对网络的安全问题的影响尤为严重，主要表现在两方面：网络病毒和不良信息传播。P2P环境下新节点的加入十分自由，这让网络病毒的入侵更加容易；各个逻辑上邻居节点在地理位置上的广泛分布也让病毒的传播更迅速。对等网络中资源不受限制地完全共享传播，没有传统的中心服务器来加以限制，给诸如色情、暴力等不良文化的滋生提供了便利<sup>[7]</sup>。CDN网络没有改变C/S的性质，因而不存在以上问题。而网络的层次性会使安全问题得到更好的控制。可以想象，在CDN边缘代理服务器上层层阻截，会减少病毒的传播可能。

二者对比如表1.1。

表1.1 CDN与P2P对比

二者对比	CDN	P2P
实现费用	耗费大	基本免费
版权问题	不存在	严重
安全问题	易解决	严重
带宽浪费	不存在	严重

通过对比可以看出，CDN相对于P2P在商业上的应用优势是很强劲的。在许多发达国家如美日等，CDN技术的市场占有率已经有80%以上，其商业应用也相当成熟。CDN的产业链主要分由四部分组成<sup>[4]</sup>：内容服务商、CDN运营商、电信运营商以及普通用户。其中CDN运营商为核心，他们自己购置服务器，租赁电信运营商的网络，提供服务给内容服务商，以提高网民的上网质量。

然而，相对于高度发达的国外市场来说，中国的CDN市场发展还远远不够，目前仅有10%的互联网企业和网站接收CDN运营商的服务<sup>[13]</sup>，说明行业的空间还很大。

### 1.3 论文结构

本文主要研究的是目前流行的两种内容分发网中的一种：CDN网络技术。本文主要的研究内容包括：

1. 将数学规划应用于CDN网络，并研究CDN技术中几个普遍问题的数学模型及解法。
2. 从CDN运营商的角度出发，对实际的CDN内容分发问题进行分析 and 建模。
3. 对模型进行求解，并提出自己的启发式算法，尽可能减少求解时间。

文章的结构如下：

第一章首先简要介绍内容分发网络产生的背景以及概述,其中以 CDN 网络为主。

第二章详细介绍 CDN 在实际应用中的主要技术,并简单介绍数学规划理论的基本概念和求解步骤。接着给出前人所研究的关于在 CDN 中结合数学规划理论的几个典型案例。

第三章转入我们自己的研究。首先提出针对商业应用的内容分发问题并加以分析(以一个具体的案例为说明来介绍)。接着说明上一章给出的几种模型的在解决此类问题上的不足。首先是在商业 CDN 应用中,计算的规模相当大,以上模型只是在理论上可解,但求解时间太长。其次,这些理论模型中考虑的影响因素与此类问题有很大不同,需要重新确立目标函数和限制条件。接着给出解决此类问题的实际数学模型,并从减小规模,减少变量,求启发解等方面来优化,得以真正解决日常决策中的内容分发问题。

第四章为仿真设计与结果分析。

第五章总结全文,并做出对未来研究方向的展望。

## 2 CDN 网络技术研究现状

### 2.1 CDN 的主要技术

CDN网络的核心即源服务器与用户之间的代理服务器。因此，如何让这些代理服务器有效地帮助源服务器满足用户访问，也就是CDN运作的核心。为了服务用户，首先代理服务器上必须有用户所需要的内容，而显然一个CDN运营商会为多个内容提供商服务，因而代理上存储着多个内容提供商的内容，必须有选择地规划内容的放置，不是所有内容在所有服务器上都有拷贝，这就涉及到内容分发。从用户端考虑，用户不用从源服务器上请求内容，现在要考虑的是代理服务器有很多个，具体定位到哪个代理上，这就涉及到内容路由的技术。

内容路由技术相对来说比较简单，目前最常用的做法是将用户按地理位置分区，每一个区或者是一个ISP，或几个ISP的管辖范围。所有用户会首先重定向本地代理服务器上，由本地代理服务。而当本地代理上没有该内容时，则重定向到最近的有该内容的邻居代理区域上。在文章[11,12]中有对此有比较详细的研究。

而CDN中技术的难点，也就是本文研究的重点是在于内容分发技术。因此以下具体介绍内容分发技术，当前两种主流的内容分发技术是PULL和PUSH<sup>[16,17]</sup>。

PULL是一种被动分发技术，即按需分发，当有请求到来时才开始内容的分发。基本上是由用户请求来驱动的。当用户请求的内容在本地的边缘Cache上不存在（未命中）时，Cache启动PULL方法从内容源服务器或者其他边缘CDN节点实时获取内容。

PUSH 是一种主动分发技术，在用户请求到来之前预先将内容分发到边缘服务器。PUSH的目的是加快对特定热点内容的响应速度，增加用户满意度。采用PUSH方法比PULL方法还要多一个前提，即需要事先知道或统计出各个内容在不同地区的访问量。这就涉及到内容或者说流媒体流量预测的问题，在本文中不研究此课题，仅采用现有的流量预测结果进行分析。

在实际的CDN系统中，一般是将二者混合使用。根据内容的类型和业务模式的不同，在选择主要的内容分发方式时会有所不同<sup>[16]</sup>。**PUSH的方式适合热点内容如流媒体影音文件等，PULL方式适合相对冷门的内容。**合适的做法是：首先对ICP提供商所需要分发的内容进行大概的历史访问记录进行统计，并预测以后的访问量，对访问量可能较大的内容采用PUSH方法预分发，其他的内容在有请求时再分发。

图2.1给出了CDN网络的具体运营模型。

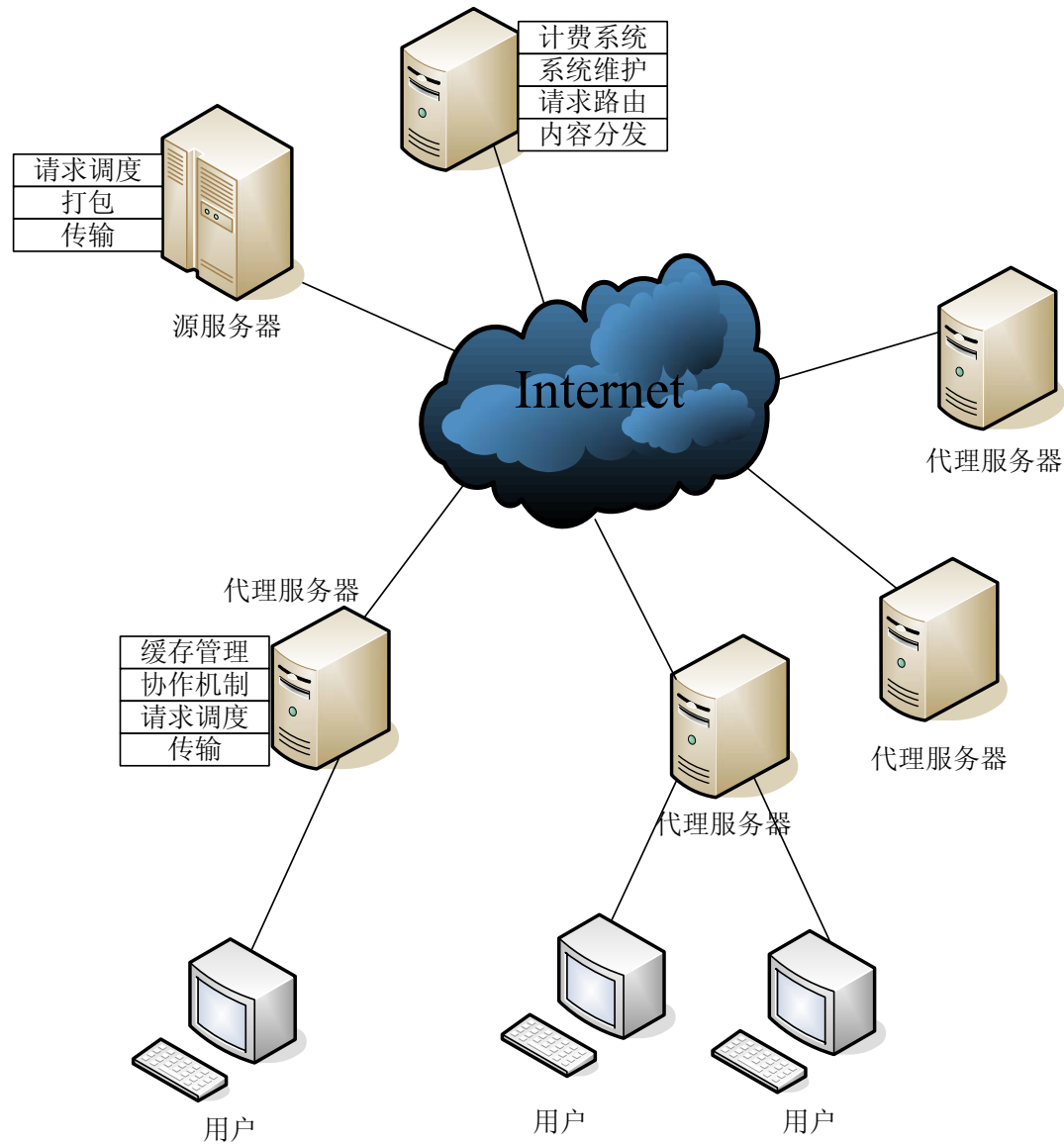


图2.1 CDN网络运营模型



## 2.2 最优化理论简述

最优化理论是一个重要的运筹学分支，它所研究的问题是如何在众多方案中找出可行并且最优的一个，是一门应用十分广泛，实用性很强的学科<sup>[19,20]</sup>。早在18世纪就已经起步。但是当时由于没有计算机，计算能力有限。直到最近，最优化理论才迅速发展，成为一个全新的学科。至今已出现线性规划，整数规划，非线性规划，几何规划，动态规划等许多分支。

给一个简单案例为例，设某工厂用4种资源生产3种产品，每单位第  $j$  种产品需要第  $i$  种资源的数量是  $a_{ij}$ ，可获利润为  $c_{ij}$ ，第  $i$  种资源总消耗不超过  $b_i$ ，由于市场限制，第  $j$  种产品产量不超过  $d_j$ ，如何安排才能使总利润最大<sup>[20]</sup>？

上述例子中一共有3个变量，即产品的数量，这些我们称为决策变量。要达到的目标是总利润最大，用公式表示即  $\max : f(x) = c_1x_1 + c_2x_2 + c_3x_3$ ，它是一个关于决策变量的函数表达式。而这些决策变量除了要使目标利润最大以外，还必须满足一系列条件，我们称为限制条件的不等式：

$$\text{资源限制 } a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 \leq b_i (i=1,2,3)$$

$$\text{市场销售限制 } x_j \leq d_j (j=1,2,3)$$

$$\text{产量非负限制 } x_j \geq 0 (j=1,2,3)$$

因此，可以把上述问题概括为：在一组约束条件下，确定一个最优生产方案  $x = (x_1, x_2, x_3)$ ，使得目标函数最大，数学模型如下：

$$\begin{aligned}
& \max \sum_{j=1}^3 c_j x_j \\
& s.t. \sum_{j=1}^3 a_{ij} x_j \leq b_i, i=1, \dots, 4 \\
& x_j \leq d_j, j=1, 2, 3 \\
& x_j \geq 0, j=1, 2, 3
\end{aligned}$$

### 2.2.1 线性规划

线性规划是最优化理论里面最基础的内容，占有十分重要的地位。这是因为它可以在多项式时间复杂内求出结果，因此解其它复杂的规划问题也都是以线性规划为元步骤。其特征是：在目标函数中和限制条件的左边，决策变量都是线性的，数学模型用矩阵表示即：

$$\begin{aligned}
& \max \quad CX \\
& s.t. \quad AX \geq b
\end{aligned}$$

线性规划问题是一定有解的，其解法也是不断在被改进中，以减少运算时间。其中最著名的就是单纯型法<sup>[20]</sup>，它是G.B.Dantzig在1947年提出的，后来经过改进，形成许多变种。单纯形法的基本思想是：从一个基本可行解出发，求一个使目标函数值有所改善的基本可行解；通过不断改进基本可行解，最终使解达到最优。其分析和求证过程这里就不讲了，总之用计算机做反复的矩阵变换，在多项式时间内是可以求出最优解的。最近的几个Matlab版本中已经有了求解数学规划的函数，而Gnu也发布了在C环境下求解的函数包。

### 2.2.2 整数规划

由实际问题所建立的数学模型中，常常要求一部分决策变量取整数值，例如人数，机器个数，是否建设服务器等。如果要求所有决策变量均取整数，则称为纯整数规划，若要求部分变量取整数，其它任意，则称为混合整数规划。纯整数规划中还有

一种比较特殊的，在实际中应用比较多的问题，即决策变量只能取0或1，这种问题称为0-1规划。

整数规划也有一种广泛使用的求解方法，即分支定界法。其计算过程主要涉及三个基本的推导概念<sup>[19]</sup>：

- 松弛

去掉整数性约束，得到一个线性规划。

称为原整数规划问题的松弛问题。它与原问题之间有如下关系：

1. 若松弛问题无解，则原问题无解。
2. 松弛问题的解是原问题的下界。

- 分解

设整数规划问题的可行集为  $S(P_0)$ ，子问题的可行集分别为  $S(P_1), \dots, S(P_k)$ ，

它们与原问题有相同的目标函数，且满足：

$$\bigcup_{i=1}^k S(P_i) = S(P_0)$$

则求解原问题可以分解成所有的这些子问题。

- 探测

设整数规划已分解成所有这些子问题之和，各自的松弛问题分别记为  $(\bar{P}_i)$ ，又知道原问题  $P_0$  的一个可行解  $X_0$ ，则可以通过以下规则剪枝：

（1）若松弛问题  $(\bar{P}_i)$  没有可行解，则相应子问题  $(P_i)$  没有可行解，可以减去  $(\bar{P}_i)$  这一分支

（2）若松弛  $(\bar{P}_i)$  问题的最优解还不如  $X_0$ ，则相应的分支  $(\bar{P}_i)$  也可以减去

（3）若解得  $(\bar{P}_i)$  问题的最优解  $X_1$  也是  $(P_i)$  的可行解，则它也是这一分支的最优解，且若此最优解的目标函数优于  $X_0$ ，则将  $X_1$  定为新的上界来剪枝

可以看出，分支定界法的解决方案是把整数规划完备地分解成了许多线性规划来

解，其元步骤也就是解线性规划。但是同时分支定界法的实际应用并没有线性规划让人满意，因为无论剪枝的方法有多少种，剪枝的效率有多高，其思想还是基于枚举法，即在遍历所有可行域的基础上剪枝。因而求解时间也是指数级的，而非多项式时间级。实际上在决策变量很多的时候，解整数规划问题有时候是难以在可容忍的时间内实现的。

混合整数规划比整数规划的解法更复杂，并没有一套公认的好解法。遇到这类问题最著名的求解思路即本德尔分解<sup>[19]</sup>，它采用了梯度下降法和整数规划两个步骤的迭代过程，其求解的基本步骤也是解整数规划，而时间复杂度显然较整数规划更高，在变量数量巨大时更难求解。

## 2.3 CDN 网络与最优化模型结合现状

鉴于CDN网络技术发展的时间不是太久，目前最优化模型在内容分发网上的具体应用还比较少，不像前面介绍的各种工程问题有很完善的模型和解决方案。把最优化理论应用到CDN技术中的研究并不多，虽然已经有不少文章是关于这一课题，但是毕竟没有权威和公认的理论，因而研究的方向也都很杂。而比较经典的研究主要关注两方面：**服务器的放置与内容的拷贝**<sup>[21]</sup>。

**代理服务器的放置**关注的是何如选择放置代理服务器的地理位置，它与CDN整体的架构布置和网络框架相关，早期有很多文章提出了相关算法<sup>[22-27]</sup>，特点是都是集中式求解，而且复杂度很高，并且问题只需要求解一次，无需反复计算。**内容拷贝**是指代理服务器位置已确定，考虑如何将一系列内容拷贝到各个服务器中，以达到**特定的目标函数最小**<sup>[28-31]</sup>。

### 2.3.1 代理服务器的放置

**Kazem**<sup>[22]</sup>最早在它的理论中提到了服务器的放置问题，他认为在因特网中服务器的位置同内容的拷贝的位置一样值得研究。如果CDN网络的目的是将内容边缘化，把内容贴近用户，那么在知道网络拓扑的情况下，代理服务器的位置是首先要考虑的。而不是像很多文章中认为的“理所当然的”仅仅将代理服务器放在局网的路由

器处，或者ISP的网关等特殊位置就可以了。相反，代理服务器存在的原因就是为了分散内容的位置分布。因此，需要根据一系列需求和限制来建立数学模型，理性的考虑代理服务器的位置分布问题。

文章中提到的需要达到的目标有两个：减小全局的网络流量、缩短用户端的时延。它把实际的网络看成是一个树形的拓扑图，整个理论都以此为基础建立。

图2.2为文章中树形给出的树形网络拓扑，其中 $r$ 为根节点，而每个节点既可以是用户也可以是服务器。

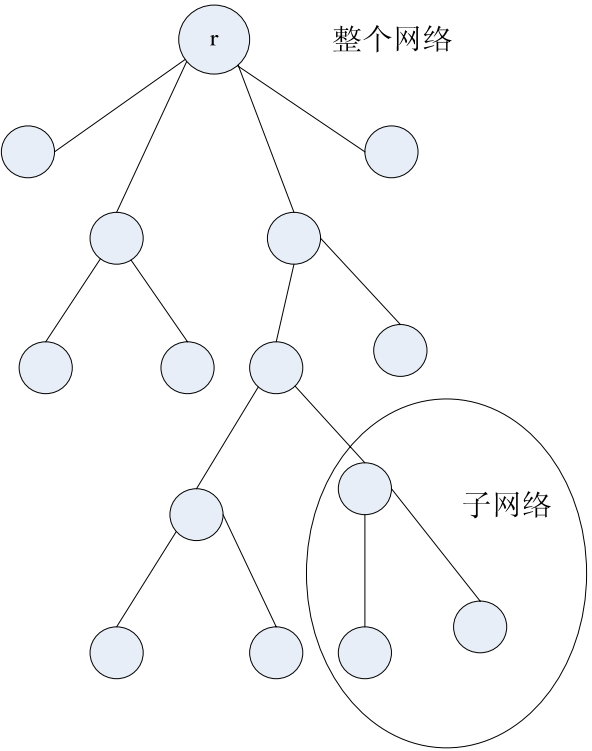


图2.2 树形网络拓扑

设  $T$  是包含  $n$  个节点的树，那么每个节点的子节点可以是任意个。由于每个节点的 Parent 节点是唯一的，则两个节点的最短路径是唯一的，用  $d(x,y)$ 表示其距离。另外设每个节点的权重为  $W(v)$ ，用来表示此节点的网络流量大小。则问题表述为：如何在此图中选择 proxy，以使所有到节点到该 proxy 的代价总和最小。文章还假设所有节点都只会向其 parent 节点请求内容，而不会重指向 child，因此很明显整个树的父节点  $r$  即代表 source 源节点。则目标函数可以表示为：

$$\min \quad \text{cost} = \sum_{v \in T} W(v) d(v, c(v, P))$$

这里  $c(v, P)$  代表在 Proxy 位置确定的情况下, 离节点  $v$  最近的上方的 proxy。(也即在默认只向 parent 节点的方向请求内容的情况下, 也默认了所有内容请求都重定向到最近的 proxy 来服务)。

而文章中给出的限制条件仅有一个:

$$|P| = M$$

Proxy 的总数为  $M$  个。

虽然文章给出了这个最优化模型来分析代理服务器的放置问题, 但是最后实际上并没有用解最优化方程的方式来解决这个问题, 而是根据树形结构的特殊性, 将问题分解为许多子问题求解。这里就不详细介绍了。

单从问题的解决思路上来看是很完美的, 文章最后证明时间复杂度在多项式时间内, 为  $O(N^3 M^2)$ , 其中  $N$  为总节点个数, 而  $M$  为需要布置的 proxy 的个数。但是此文章研究的服务器放置方法在模型建立上有很多问题: 1) 整个理论的基础就是树形的网络拓扑结构, 而 CDN 网络面对的是整个因特网, 显然研究这样简单的拓扑是没有意义的。2) 模型的限制条件过于简单, 仅仅限制了代理服务器的个数恒定, 而每个代理服务器的服务能力, 带宽等都没有提到, 也即默认了所有服务器的服务能力都是无限的, 显然是不合理的。

### 2.3.2 内容调度

服务器放置需要考虑的实际因素太多, 如地理位置上是否可行、是否扰民、建设成本等。根本不能简单的用离用户近这一简单原则来考虑, 也很难用简单的数学模型来分析。随着 CDN 产业的快速发展, CDN 运营商已经建设了大量的服务器群, 布置新服务器的速度也相应变缓, 此类研究的应用变得次要了。相反, 如此多的服务器上各种内容需要如何调度是一个 CDN 服务商每天都必须考虑的问题<sup>[21]</sup>。

Padmanabhan<sup>[30]</sup>早在 2001 年就在 INFOCOM 中发表了一篇在关于在 CDN 网络中合理放置内容源拷贝 (content replicas) 的文章, 其中就用到了最优化理论的数学模型。

它最早地注意到了利用最优化理论可以提高CDN网络的效率与合理性。

其考虑的具体问题是这样的：在一张图中，给出一系列的可以用来放置内容拷贝（实际中即所有临近代理服务器的位置）的点集 $i$ 。在点 $i$ 建设服务器会的代价是 $f_i$ （内容分发和存储的消耗）。每一个客户 $j$ 都指派给其中一个服务器去服务，从而导致的消耗是 $d_j c_{ij}$ ，这里 $d_j$ 代表客户 $j$ 的请求量， $c_{ij}$ 代表的是二者之间的距离。这里用距离作消耗的乘积因子主要是考虑物理距离和响应延时、网络流量成正比。其目标函数即最小化总的消耗，包括放置内容拷贝的消耗和用户端感受。

其最优化理论的数学模型如下：

$$\begin{aligned} \text{Min} \quad & \sum_{i,j \in N} d_j c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{i \in N} x_{ij} = 1 \quad \text{for each } j \in N \\ & x_{ij} < y_i \quad \text{for each } i, j \in N \\ & \sum_{i \in N} y_i < k, \\ & x_{ij}, y_i \in \{0,1\}, \quad \text{for each } i, j \in N \end{aligned}$$

这里的 $y_i$ 和 $x_{ij}$ 都是0-1整数变量， $y_i$ 表示是否在 $i$ 点放置内容， $x_{ij}$ 表示是否将用户 $j$ 的请求重定向到服务器 $i$ 。前三个限制条件的含义分别是：对于每一个用户请求，都会发送给并只发送给其中一个服务器；若 $i$ 点并没有放置该内容拷贝，则用户请求不会定向过去；总的内容拷贝份数会有一个上限。

显然这里的决策变量都是0-1，所以是一个整数规划。可以用拉格朗日松弛将第一个限制条件转化到目标函数里面，再迭代求解。

$$L(x, y, z) = \sum_{i,j \in N} d_j c_{ij} x_{ij} - \sum_{j \in N} u_j (\sum_{i \in N} x_{ij} - 1)$$

然后用文献<sup>[32]</sup>中的梯度下降法来迭代求解次优解。从初始的乘积因子 $u_0$ 开始，先在这种情况下解出一个最优解，那么此解必然是原问题的上界。这时根据求出的解和 $u_0$ 的值得到下一个探测乘积因子 $u_1$ ，然后继续求解，反复迭代。每一次迭代都会以一定梯度得到一个更好的上界。一定次数后便可以得到误差容许内的解。

文章的作者自己也说了，在变量数目很大时，这类问题只能用近似解法，并不能

得到精确解。而且在结果的仿真中，其总节点数只有200（包括所有的client和proxy）。

这篇论文虽然是最早将最优化理论应用到CDN网络中，但它更偏向于求解数学模型部分，而在本身模型的建立上，并没有从实际CDN网络的问题出发。1) 目标函数和限制条件考虑不明确。关于目标函数，其文章中只是简单的带过说是减少用户到proxy的距离，即便此距离能代表时延，显然考虑也是不完善的，最基本的负载均衡都没有考虑。而限制条件更少，就只有总存储份数的上限，关于各个proxy的存储空间，带宽这些最起码的基本限制条件都没有提到。2) 问题的出发点不合实际，在CDN网络中，并不能明确的预先知道哪几个用户会对何种内容提出请求，更别说预先就指派给其服务的proxy。因此关于这个内容放置的理论模型，其对类似NP问题的求解方式是可以借鉴的，但应用到实际CDN网络中还远远不够。

2002年Jussi<sup>[28]</sup>在此基础上改进了此模型，更加注意到了在实际Internet上的应用。它声称其仿真是建立在实际Internet拓扑数据上，主要根据是其生成的结点图的连接引用了一个经典的模型NLNR<sup>[32]</sup>。在建立具体的数学模型时，它考虑的问题是这样的：

网络中共有I个Proxy，每一个的最大存储量为 $S_i$ 。需要存储的内容有j个种类，第j种内容的大小为 $b_j$ ，而且还对应有一个热度 $q_j$ ，对应于此内容被用户请求的概率，这里有一个用来简化模型的假设，即：同一内容相对于所有Proxy来说，其请求热度是相同的。其目标则是在满足所有内容的用户请求时，最小化所有Proxy之间的转发次数，因为它默认所有用户都由离自己最近的Proxy来服务，而如果最近的Proxy没有所需内容，则必须缓存转发较远Proxy的包，因此增加了用户端的时延，最小化这个时延能增加用户满意度。

给出的数学模型如下：

$$\begin{aligned} \text{Min} \quad & c(x) = \sum_i \sum_j q_j d_{ij}(x) \\ \text{s.t.} \quad & \sum_j b_j x_{ij} \leq S_i \quad i = 1, \dots, I \end{aligned}$$



这里出现了一个新的变量  $d_{ij}(x)$ ，它表示在所有Proxy中存储内容都确定，即在每一个  $x_{ij}$  值都给出的情况下，第  $i$  个Proxy到离它最近的含有内容  $j$  的Proxy的跳数，显然，若第  $i$  个Proxy存储了内容  $j$ ， $d_{ij}(x)$  就是0。这里首先是假设了一种简单的转发机制：每个用户都直接重定向到离它最近的Proxy，如果有用户所需内容则直接服务，没有则由离它最近的含该内容的服务器来转发。

显然这个模型更加贴近现实，其目标函数明确定义了最小化Proxy之间的总转发跳数，这比纯粹的“距离”更有说服力。而且限制条件也更合理，显然把内容拷贝到各个Proxy中，最基本的限制即各个Proxy的存储量，而非内容的总存储份数这么飘渺的解释。

虽然这一研究在CDN技术中更合理和深入，但还是有一定的不足。首先它的简化假设就难以成立，很显然每一种内容在不同地域的热度是不同的，不论流媒体或web页面均是如此。那么  $q_j$  会变成  $q_{ij}$ ，数学公式会复杂许多。其次，它仍然没考虑服务器端的带宽限制，这一限制条件会在后面提到，并且理服务器的服务能力也是有限的。举个简单的例子，如果某份内容只在一个Proxy中存储一份，那么在此文章中也是有可行解的，但是实际中如果所有用户都由此Proxy来服务，那它肯定会当机。第三，文章中提到的数学模型求解过于困难，因为只有在  $x_{ij}$  确定的情况下， $d_{ij}(x)$  才有明确的值。以至于在最小的变量下也很难求解，最后作者不得不假设所有内容的热度相同（实际上等同于只有一种内容），才能迭代出次优解，并在文章的末尾给出了几种其它几种启发式解法。由于没有求出最优解，这些启发算法的性能也无从考证。

以往的内容调度模型在建模上都存在着或多或少的问题。如第一章所述，CDN网络的主要手段是将内容放置到尽量离用户近的地方，而主要目的是减少用户端时延，达到网络负载均衡。早期的文章主要都围绕着减少时延这一目标展开，而忽略了负载均衡的问题<sup>[22,23,24,28]</sup>。因而在建立CDN模型时，很少考虑到如代理服务器的最大负载，带宽等限制，得出的模型均比较简单而不实际。极端一点假设，所有服务

器的服务能力都是无限的，那么将所有的内容在所有的服务器上全都拷贝一份，则用户端的响应时延肯定最小。

在前面分析的文章中，所有的模型和仿真都建立在具体的网络拓扑图中，都有一个相同点：每一个client和proxy是一个单独的节点，每一个用户对每一种内容的需求都是给定的，作为输入。而CDN网络的规模是很大的，得到其所服务地区的网络拓扑图已经很困难，而得到每个具体用户的内容请求基本是不可能的。文献<sup>[30]</sup>中将用户请求改为概率的形式，即每个用户有一定概率请求某一内容。这种说法似乎更合理，但是单个用户请求本来就是随机现象，而且变化很快，也无法有效统计，估计每个用户的请求概率也是难以实现的。而且在实际的网络拓扑图中分析问题还会带来另外一个问题，即用户请求重定向的问题，大多数文章都仅将用户重定向到最近的边缘proxy，而没考虑到本地proxy负载过重无法服务的问题，这在实际中是经常发生的。因此本文尝试将用户聚类，某一片用户的请求作为一个整体，不用考虑边缘proxy到终端用户的复杂拓扑，这使得模型更关注于全局的内容调度，而不是针对单个地区单个用户，客观上模型也变的更简单，输入更现实合理。

### 2.3.3 本章小结

本章介绍了前人针对CDN网络中的内容分布模型的一些研究，并分别举例说明。由于CDN本身就是一种商业应用，对其研究也更偏向实际商业运作，而非纯理论。这也是服务器的放置问题越来越销声匿迹的根本原因。2002年以后，研究服务器的放置的文章逐渐变少，取代的是研究如何在代理服务器上拷贝热点内容，即智能的PUSH策略。

## 3 商用内容分发模型研究

### 3.1 前提条件与说明

由于CDN本身就是一种商业应用,因此我们认为关于内容分发的最优化算法也应该从商业考虑出发,且需要能实时在线使用。本文的目标就是研究在商业应用中,使用怎样的内容分发技术能让CDN运营商最满意。与前文所述的内容分发算法不同,首先它的目标和限制条件都完全从CDN运营商角度出发考虑,而并不在乎整个网络性能和用户满意度等;其次它需要有更高的时效性,至少在一个触发时隙(例如一天更新一次代理服务器的存储内容,则时隙为一天)内能完成求解;最后该算法研究的是处于运行中的CDN系统中的一个时隙内的行为,因此下一时隙的初始状态即上一时隙的结束状态,具体到模型中即每个代理服务器都有初始存储内容,需要做的是根据下一个时刻的流量预测做出调整,而不是单纯的把热点内容PUSH到空的代理服务器中去。

在研究内容分发算法时,还必须先假设整个CDN网络运作的环境,最重要的就是内容路由方法的选择。在文献[22,25,27,28,29,30,31]中,所有用户的请求都重定向到本地代理服务器,默认全部由其服务,而不管其负载能力。文献[21]对此提出了异议,把代理服务器的负载放入目标函数中,但是没有给出清楚的证明。因此下面先给出完整的CDN网络运行方式,包括网络拓扑描述和详细的用户重定向方式。

我们不考虑具体到各个用户的路由与请求应答,而是将**所有位置上临近某个代理服务器的所有用户以及此服务器看成是一个整体节点**。因为本地代理如果能满足临近用户的请求,则直接服务不用重定向。我们重点考虑的是在本地代理因为内容缺失或负载过重时的调度,因为这样会产生冗余的流量,所以将本地代理和用户聚类考虑是完全合理的。在图 3.1 中一个代理节点表示一个拓扑区域,包括其代理服务器及其下属的所有用户。每个节点内的所有用户产生的总的请求若都能由本地代理应答,则不产生到骨干网的流量,否则将所有的未应答的看成一个整体,流出到骨

干网内。以节点为单位来考虑问题，大大减小了模型的规模，使问题的注意力集中到整体的内容分布规划上。

图 3.1 表示网络拓扑，有两种节点，即源服务器节点和代理节点。

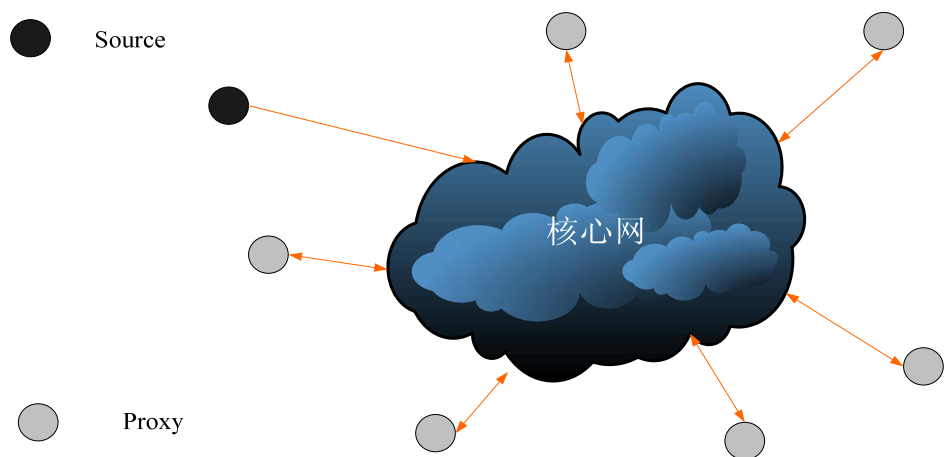


图3.1 实时内容调度的节点网络模型

### 3.2 数学模型的建立

建立最优化数学模型需要首先确定目标函数与限制条件。

首先分析目标函数，如上所述，我们建立的模型完全是从 CDN 运营商的角度出发的，所以考虑的目标也不外乎两方面：最大化运营收入与最小化运营成本。

一般来说，CDN 运营商是靠服务量来收费的，也就是用户从 CDN 运营商下载多少内容，就向相应的内容提供商缴多少费。而用户的请求是自发的，请求量是 CDN 运营商无法左右的。因此 CDN 运营商为了提高收入，唯一能做的就是尽量满足用户的所有请求，而不要因为网络拥塞等原因放弃用户的请求。实际上，一般的 CDN 运营商都是以服务所有用户请求为目标的，这样不仅最大化了收入，而且也是公司声誉的保证。因而我们把服务所有客户的请求作为限制条件之一。

不考虑服务器扩建等特殊事件，CDN 运营商日运营成本有两方面：从 ISP 处购买带宽费用、更新服务器内容带来的硬件损耗。前者比较好理解，消费与购买带宽理论上成正比。后者的实际情况是这样：以蓝汛公司为例，将内容以频道（一个频

道 1.5TB) 为单位划分, 一台服务器仅存储并服务一个频道, 如果要更换服务器上内容, 就必须整个频道全部删除然后重新下载, 这样对服务器带来的损耗是巨大的, 所以要尽量少更换服务器上存储的内容。骨干网的带宽消耗又分三种情况: 用户请求重定向到远端代理、内容同步、内容更换。如第一章所述, 用户请求原则上就近由本地代理服务, 而在本地代理无法服务时, 就会重定向到远端代理, 消耗的骨干网带宽即应答内容的大小。内容同步指的是代理服务器上存储的内容需要实时和源服务器上的内容保持一致, 会周期性的使用少量骨干网带宽。代理服务器更换存储内容时, 是从源服务器或其它代理上下载的, 显然会消耗与内容大小相同的带宽。

而限制条件除了前面说的应答所有用户请求, 还需要考虑以下几点: 首先每个代理服务器的存储容量有限, 所存储的内容大小总和不能超过这个值。第二, 每种内容至少在代理服务器上有一份拷贝, 即不论多么不热门的内容, 只要从 CDN 运营商处购买了服务, 就不能完全丢弃。正因为有了这一限制条件, 才能保证所有在本地无法得到满足的用户请求都能从骨干网处得到应答。第三, 每个节点的代理服务器到骨干网的带宽是有限的, 这也一定程度上保证了负载均衡。

上述即 CDN 内容调度中所有的目标和限制条件。以此建立数学模型会非常复杂, 难以在要求时间内求解, 并且在用数学公式表达目标函数时, 还会出现与文献[31,35]相同的问题: 当目标含两种不同单位的变量时, 如何将二者放在一个表达式内? 文献[35]中的目标函数与我们较类似, 是最小化存储份数与带宽消耗(我们的目标是最小化带宽消耗和机器折损), 在建立模型时, 分别给二者乘以权值然后相加, 认为此权值代表了二者的重要性比值。而我们认为这个加权值是很难说清楚的, 至少无法在理论上证明。我们的想法是将二者分开考虑, 即建立两个简化的模型, 分别考虑不同实际情况下的问题。还是以北京蓝汛公司为例, 由于业务庞大而稳定, 为了确保能满足所有用户请求, 其每年购买的骨干网带宽一般都是恒定的, 比较充足; 而服务器数量众多, 日常最大的开销就是内容更换带来的折损。因此它更注重减少服务器内容更换频率, 而骨干网带宽只要不超出上限即可。可以想象, 相对于业务发展比较快, 骨干网带宽购买需要不断增加的公司, 会希望通过策划尽量减少这方面的消耗。因此下面我们以两种假设情况将目标函数分离, 并分别建立各自模型求解。

### 3.2.1 最小化网络带宽模型

在给出数学模型前，先定义各类参数：

定义周期 $t$ ，即每隔时隙 $t$ 进行一次内容调度。定义集合符号如下： $I$ 表示节点， $J$ 表示内容。定义常量如下： $K$ 表示内容大小，仅与 $J$ 相关； $M$ 表示维持内容的消耗，仅与 $J$ 相关。 $S$ 表示服务器存储上限，仅与 $I$ 相关； $B$ 表示使用的骨干网带宽，仅与 $I$ 相关； $P$ 表示预测访问量，比文章[8]更合理是，这里的预测访问量不仅与内容相关，与节点号也是有关的，即同一个内容的热度在不同节点处是不同的。定义模型的决策变量：

$$x_{ij} = \begin{cases} 1 \\ 0 \end{cases}$$

此变量表示内容 $j$ 在下一时段是否在节点 $i$ 上存储，1表示存储，0表示不存储。

定义与决策变量形式类似的内容存储初始值 $C_{ij}^{init}$ ：

$$C_{ij}^{init} = \begin{cases} 1 \\ 0 \end{cases}$$

它是个常量，表示上一时刻服务器存储内容的状况。它的值与决策变量的取值相反，0存储，1表示没存储。下面会提到这样设的原因。

定义目标函数为：最小化各节点消耗的骨干网流量总和：

$$\sum_i \sum_j \left( p_{ij} k_j t (1 - x_{ij}) + x_{ij} M_j + x_{ij} C_{ij}^{init} \right) \quad (3.1)$$

目标函数的三项分别代表了上述的三种骨干网带宽消耗。第一项表明如果该内容不在本节点内存储，则所有本节点针对该内容的请求都将由其它节点服务，从而产生入口流量 $p_{ij} k_j t$ 。第二项表明如果该内容在本节点内存储，则要承担相应的维持内容一致性流量。实际仿真中这一项的大小可以忽略，因为它的流量相比访问量来说太小，在整个目标函数中占份额少于1%。第三项是内容更换所消耗带宽。这里有一个小技巧：对于每一个 $(i, j)$ 对， $x_{ij} C_{ij}^{init}$ 有四种情况，即 $x_{ij}$ 和 $C_{ij}^{init}$ 分别取0和1，

但是只有二者同时取 1 的时候乘积才为 1，否则为 0。而其物理含义也正是如此， $C_{ij}^{init}$  取 1 表示上一时刻在节点  $i$  上没存储内容  $j$ ， $x_{ij}$  表示下一时段在节点  $i$  上存储内容  $j$ ，即内容  $j$  被分发到了节点  $i$  上。而其余三种情况要么表示维持原状态，要么表示删除内容，都不会消耗骨干网带宽。这样做的好处是线性化目标函数，方便求解。

由于  $p_{ij}$  和  $k_j$  都是常量，令

$$k_{ij} = k_j \quad \text{for each } (i, j)$$

则有：

$$\sum_i \sum_j \left( p_{ij} k_{ij} t (1 - x_{ij}) + x_{ij} M_j + x_{ij} C_{ij}^{init} \right) \quad (3.2)$$

将  $p_{ij} k_{ij} t$  整合成一项： $m_{ij}$ ，则目标函数变为：

$$\sum_i \sum_j \left( m_{ij} (1 - x_{ij}) + x_{ij} M_j + x_{ij} C_{ij}^{init} \right) \quad (3.3)$$

限制条件转为公式如下：

$$\sum_j k_j x_{ij} \leq S_i \quad (3.4)$$

式子左边表示节点  $i$  上存储的内容大小总和，右边是节点  $i$  上的存储限制，即每个代理服务器的存储量不得超过其上限。

$$\sum_i x_{ij} \geq 1 \quad (3.5)$$

将变量  $x_{ij}$  对  $i$  求和，得到的是存储内容  $j$  的服务器的总数，即内容被存储的总份数。这里我们没有考虑 miss 情况下的具体路由问题，因此为了保证能满足所有用户的需求，必须保证所有内容都至少有一个备份。

$$\sum_j k_j x_{ij} \leq S_i \quad (3.6)$$

为简化模型，忽略掉几乎没有影响的维持消耗  $x_{ij} M_j$ 。总结整个模型如下：

$$\begin{aligned}
\min \quad & \sum_i \sum_j \left( m_{ij}(1-x_{ij}) + x_{ij}C_{ij}^{init} \right) \\
s.t. \quad & \sum_j k_j x_{ij} \leq S_i \\
& \sum_i x_{ij} \geq 1 \\
& \sum_j \left[ m_{ij}(1-x_{ij}) + x_{ij}M_j \right] \leq B_i, \forall i
\end{aligned} \tag{3.7}$$

观察整个数学模型，决策变量只有  $x_{ij}$ ，为 0-1 整数变量。目标函数和限制条件都是关于此变量组的线性表达式，因此该模型是标准的 0-1 规划模型，可以用前面介绍的分支定界法或隐数法求解。求解复杂度与变量个数有关，这里是  $i$ 、 $j$ 。虽然看上去表示  $i$  个服务器上的存储内容， $i$  个变量已足够，可以减少变量个数，但这样目标函数就很难用线性函数表示，也就不是整数规划问题了。

### 3.2.2 最小化服务器损耗模型

同上一节，先给出参数定义：

同上，令决策周期为  $t$ ，即每隔时隙  $t$  进行一次内容调度。定义节点集合如下： $N$  表示节点， $I$  表示节点内服务器， $J$  表示频道。定义所有常量如下：骨干网带宽上限  $B$ ，与节点序号相关；预测频道访问量大小  $P$ ，与节点和频道序号均相关；所有频道大小相等为  $S$ ，一个服务器存且仅存储一份频道；所有代理服务器服务能力上限相同，为  $M$ 。

定义决策变量组  $\sum_{i,j} C_{ij}^{init} x_{ij}$  表示服务器  $i$  对频道  $j$  的实际服务流量。

定义决策变量组：

$$x_{ij} = \begin{cases} 1 \\ 0 \end{cases}$$

此变量表示频道  $j$  在下一时段是否在节点  $i$  上存储，1 表示存储，0 表示不存储。



定义与决策变量形式类似的内容存储初始值  $C_{ij}^{init}$  :

$$C_{ij}^{init} = \begin{cases} 1 \\ 0 \end{cases}$$

它是个常量，表示上一时刻服务器存储内容的状况。它的值与决策变量的取值相反，0 存储，1 表示没存储。这样表示的原因与上一节相同。

目标函数为最小化所有服务器上频道更换的总份数：

$$\text{Min} \quad \sum_{i,j} C_{ij}^{init} x_{ij} \quad (3.8)$$

上一节解释了这样表示的原因，这里就不多说了。

限制条件有以下几点：

$$\forall j, \sum_i v_{ij} \geq p_j \quad (3.9)$$

左边是频道  $j$  的所有服务流量，右边频道  $j$  的预测访问量。它的含义是：频道的访问量必需全部得到服务。

$$\forall n, \sum_j \sum_{i \in n} v_{ij} \leq B_n \quad (3.10)$$

不等式左边是节点  $n$  上所有的服务流量总和，注意这里并不是对全部  $i$  求和，只是对从属于节点  $n$  的所有服务器求和。此限制条件的含义是：节点的实际服务流量不超过带宽限制。

$$\forall i, \sum x_{ij} \leq 1 \quad (3.11)$$

每个服务器上最多存储一份频道。

$$\forall i, j \quad v_{ij} \leq M \cdot x_{ij} \quad (3.12)$$

每个服务器上的服务能力上限为  $M$ 。

模型总结如下：

$$\begin{aligned}
& \min \sum_{i,j} C_{ij}^{init} x_{ij} \\
& s.t. \quad \forall j, \sum_i v_{ij} \geq p_j \\
& \quad \forall n, \sum_j \sum_{i \in n} v_{ij} \leq B_n \\
& \quad \forall i, \sum_j x_{ij} \leq 1 \\
& \quad \forall i, j \quad v_{ij} \leq M \cdot x_{ij}
\end{aligned} \tag{3.13}$$

与上一节的模型不同，这里的决策变量有两组， $x_{ij}$  和  $v_{ij}$ ，即存储何种频道与实际服务流量多少。虽然目标函数与限制条件都是关于决策变量的线性表达式，但  $x_{ij}$  是整数变量而  $v_{ij}$  没有要求，这是一个混合整数规划问题，如第二章所述，混合整数规划问题的求解是 NP 难问题，极难求解。本文在下一章用基于整数规划的启发算法来求解，而不会给出最优解。

### 3.3.3 本章小结

本章从 CDN 网络运营商面临的实时调度问题出发，全面地分析了运营商所想达到的目标和面临的条件限制。然后分解模型，将原问题化成两个问题来分析，并说明二者分别适用的情况。然后给出了两个问题的具体数学模型，即最小化网络带宽与最小化服务器损耗。前一个问题的数学模型恰好是标准的 0-1 规划，很方便求解；而后一个问题为多变量的混合整数规划，要复杂的多，并无标准解法。因此在下一章着重研究第二个问题的求解。

## 4 模型求解与仿真

### 4.1 模型求解

#### 4.1.1 最小化网络带宽问题求解

我们采用 0-1 整数规划的分支定界方法求解此问题，并为了缩短解决时间，使其在时隙内能运算完，还在此基础上给出了简化的启发式算法，并在最后给出比较。

- 标准的 0-1 整数规划解法

求解 0-1 规划有两种标准方法，即分支定界和隐数法，二者原理类似，在第二章已经有简要介绍，很多最优化理论书籍(如文献[19, 20])均有详细说明。因此下面直接给出用隐数法的求解步骤，不作详细介绍了。

步骤 1：松弛：把决策变量  $X$  的整数限制去掉。则原整数规划问题  $P$  松弛成线性规划问题  $\bar{P}$ ，求此松弛问题的解  $S(\bar{P})$ 。

步骤 2：分解：0-1 规划的分解比较简单，仅需将变量分为两个分支，即分别取 0 和 1。将  $S(\bar{P})$  中离整数最远的  $x_{ij}$  找出来，分解为两个分支继续。

步骤 3：探测：用 2.2.2 小节中的标准的 3 步法探测。

步骤 4：判断是否所有分支求解结束。是则问题结束，最终解为所有分支求解结果中最小的那个。否则继续步骤 2。

上述解法用图表示如 4.1。

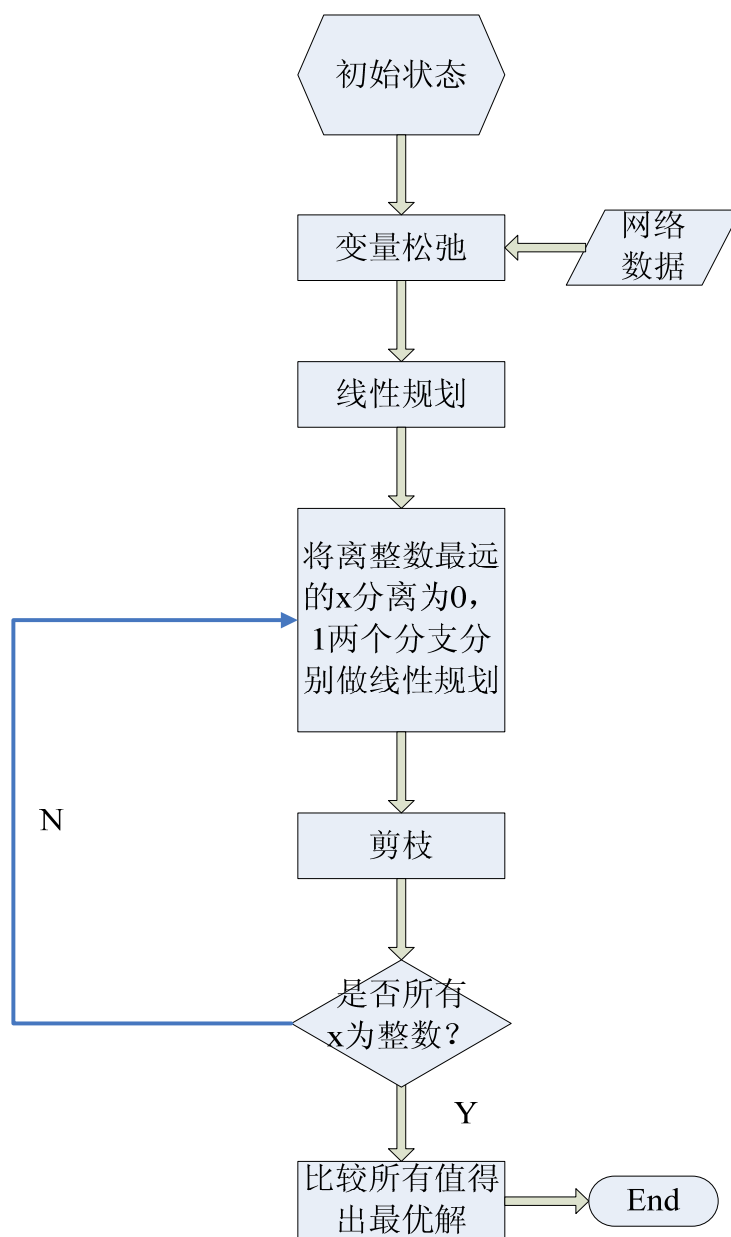


图 4.1 标准 0-1 规划解法步骤

#### ● 快速解法

我们提出了一种快速算法，其原理是在原算法基础上加入局部贪婪思想，不考虑所有情况，只考虑局部最优，让剪枝更快速。在得到多个中间分支时，我们并不像原解法那样对每个分支继续求解，而是只找出中间结果最好的一支，迅速剪掉其它分支。时间上由指数级变成了线性级，性能上陷入了局部最优解。启发式算法的步骤如下：

步骤 S1: 松弛, 同原步骤 1。

步骤 S2: 近似: 设定某门限值, 将松弛问题的解根据此门限作近似。即离整数的距离小于此门限的都看做整数。我们实际中设为 0.01。

步骤 S3: 分解, 同原步骤 2。

步骤 S4: 我们不用探测法来剪枝, 而是只保留中间结果好的一条分支, 直接去掉另一条。

步骤 S5: 由于只有一个分支存在, 因此只要得到一组可行解, 则问题结束。否则继续步骤 2。

整个算法与标准隐数法区别主要在于步骤 S2、S4。S2 是我们在原隐数法程序运行时发现的经验, 迭代过程中有许多变量值十分接近 0, 将这些变量看成整数 0 会方便快速很多。因目标函数是关于变量的线性函数, 将这些变量做稍微调整对目标函数影响也十分微小, 加入 S2 这一步对最优解结果影响是很微小的。加入 S4 步骤的结果是直接剪掉了“中间步骤表现不好的分支”, 使求解陷入了贪婪的局部最优, 也因此让原问题的复杂度由指数级下降到线性级。

#### 4.1.2 最小化服务器损耗问题求解

$$\begin{aligned} \min & \sum_{i,j} C_{ij}^{init} x_{ij} \\ \text{s.t.} & \quad \forall j, \sum_i v_{ij} \geq p_j \\ & \quad \forall n, \sum_j \sum_{i \in n} v_{ij} \leq B_n \\ & \quad \forall i, \sum_j x_{ij} \leq 1 \\ & \quad \forall i, j \quad v_{ij} \leq M \cdot x_{ij} \end{aligned} \quad (4.1)$$

如上所述, 此模型由于同时含整数变量  $x_{ij}$  和非整数变量  $v_{ij}$ , 称为混合整数规划, 无标准解法。我们的想法是先固定一组变量的值, 然后求解只含另一组变量的整数或线性规划问题, 然后根据求解结果反推被固定的变量, 如此反复迭代逼近。毕竟求解线性规划是多项式时间复杂度, 而整数规划是指数级时间复杂度, 我们决定固

定变量  $x_{ij}$ ，剩余的问题就只含  $v_{ij}$ ，是一个线性规划问题。与文献[40，41]类似，虽然都无法给出最优解，但我们会给出最优解的下限估计，以其做对比来估计启发式解的性能。

先给出算法物理含义：由初始条件开始，即不改变上一时刻服务器存储内容，试图通过调节各个服务器的服务流量来满足各个频道的访问量。如果能满足，则求解结束，最优解即 0，不需要改变任何服务器的存储频道。如果不能，则尝试只改变一个服务器上的频道，如果能满足所有访问量则结束。否则选择挑选改变其中一个服务器上的频道，使得“能满足尽可能多的访问量”。如此反复，直至最终所有频道访问量都能被满足。

接着给出求解步骤。

步骤 1：固定  $x_{ij}$  为上一时刻初始值，即不改变任何存储频道。

步骤 2：在  $x_{ij}$  确定的情况下，求解子问题：

$$\begin{aligned}
 \max \quad & \sum_i \sum_j v_{ij} - \sum_j p_j \\
 s.t. \quad & \forall n, \sum_j \sum_{i \in n} v_{ij} \leq B_n \\
 & \forall i, j \quad v_{ij} \leq M \cdot x_{ij} \\
 & \forall j, \sum_i v_{ij} \leq p_j
 \end{aligned} \tag{4.2}$$

这里的两个限制条件分别与原问题（4.1）的限制条件 2，4 相同，并把（4.1）中的限制条件 3 变到目标函数中去，并多出一个限制条件：

$$\forall j, \sum_i v_{ij} \leq p_j \tag{4.3}$$

这个限制条件含义是：对于每个频道，服务量不超过访问量。如果某频道访问量已经满足，追加服务量只会浪费节点带宽而没有任何好处，因此加入这个限制不会使求得的最优解变差。而这里加入这个限制的原因是很明显的：问题（4.2）的目标是尽量满足尽量多的“未被满足的”频道访问量，而  $\sum_i \sum_j v_{ij} - \sum_j p_j$  这个式子中计

算的是所有频道访问量，加入限制（4.3）后才可。另外  $\sum_j p_j$  是常量，因此这里要

求解的线性规划问题可以变成：

$$\begin{aligned} \max \quad & \sum_i \sum_j v_{ij} \\ \text{s.t.} \quad & \forall n, \sum_j \sum_{i \in n} v_{ij} \leq B_n \\ & \forall i, j \quad v_{ij} \leq M \cdot x_{ij} \end{aligned} \quad (4.4)$$

步骤 3：如果所有访问量被满足，即求出的  $\sum_i \sum_j v_{ij} = \sum_j p_j$ ，则问题结束，得

出最优解。否则继续。

步骤 4：取任意一个代理服务器，令其存储未被满足的任意一个频道。即改变一个  $x_{ij}$  的值，然后在此基础上求解问题（4.4），找出目标函数最大的一个，继续步骤 3。

算法解混合整数规划的思想与文章[42，43]相似，都采用先固定整数变量再求线性规划的迭代方法。但是由于我们模型的特殊性，可以用物理意义更鲜明的启发式算法求解，算法的时间复杂度分析如下：步骤 1 与步骤 2 只执行一次，步骤 3 仅是一个判断过程，其运行时间可以忽略。步骤 4 为迭代过程，理论上它需要遍历所有服务器对于所有未满足频道的存储。但实际上如果一个服务器没有服务，或者服务量很小，更换存储频道的概率会更大。根据经验，步骤 4 中我们仅挑出 10 个服务量最小的服务器作比较，由于频道总数为  $J$ ，则每执行一次步骤 4 的试探的总数最多为  $10 \cdot J$ ，每试探一次需要解变量数为  $IJ$  的线性规划问题。而步骤 4 执行的次数就是改变存储频道的服务器个数，不会超过  $I$ 。因此最多需要解  $10 \cdot IJ$  次变量数为  $IK$  的线性规划即可。第二章已说过，无法给出单纯形法解线性规划的具体时间复杂度，只是公认为多项式时间内。所以我们的算法也可以说是以  $IJ$  为变量的多项式时间复杂度内。

## 4.2 实验与仿真

### 4.2.1 网络带宽模型仿真

我们通过实验比较隐数法和启发式算法解 0-1 规划的时效性,并统计次优解与最优解的平均差距。

设定的仿真实验参数如下:

节点数量 5—— $I$ , 内容数量 10 个—— $J$ 。为简化计算,不妨设所有节点硬件设施相同:存储容量设为 20(单位:TB)—— $S$ ;每次实验中各个内容的大小取 1 到 10 间的随机数(单位:TB)—— $K$ ;各个内容的访问量则取 1 到 100 间的随机数—— $P$ ;为了体现实时调度实验的时间连续性,整个实验在一个连续的时段中完成,即上一时段求解出的结果(这里结果指决策变量组取值,即服务器上存储的内容)就是下一时段的初始条件。为了简化模型,这里我们并没有给出维持内容的消耗  $M$ 。

Matlab 和 C 环境下均有求解线性规划单纯形法的工具包,但都没有解整数规划的工具包。我们选择用 C 语言编写隐数法和启发式算法的程序,其中使用了 GNU 发布的求解单纯形法的工具包<sup>[20]</sup>。

为了隐数法求最优解的正确无误,我们用暴力搜索法作为验证。暴力搜索法遍历了所有的情况,所以得到的解一定是最优的。

在进行的 100 个重复实验中,启发算法消耗时间平均 50~70 秒;0-1 规划方法平均 2~6 分钟;而简单枚举法的结果平均是 15 分钟。表 4.1 为从 100 次实验中随机找出的 5 次实验结果的对比。这里的结果即消耗的骨干网带宽总和,单位 MB。第一列为最优解,是枚举法与隐数法的求解结果,即消耗的总骨干网带宽。第二列是用启发式算法求出的结果。



表 4.1 随机 5 次实验的消耗对比(单位: MB)

最优解	启发解
4653	5098
5643	5809
5109	5109
5273	5817
4908	5162

表 4.2 为选取的 5 次实验的求解时间对比。每一行都是一次条件相同的随机实验，数据的单位为秒。

表 4.2 随机 5 次实验所用时间(秒)

启发算法	隐数法	枚举法
49	154	911
57	200	905
52	190	907
54	254	905
68	265	920

图 4.2 中两条曲线分别表示了最优解和次优解求得的骨干网带宽消耗（单位: MB），可以看出，次优解比最优解要差一些，但是相对于基数来说相差百分比并不大。并且二者还有重合点。

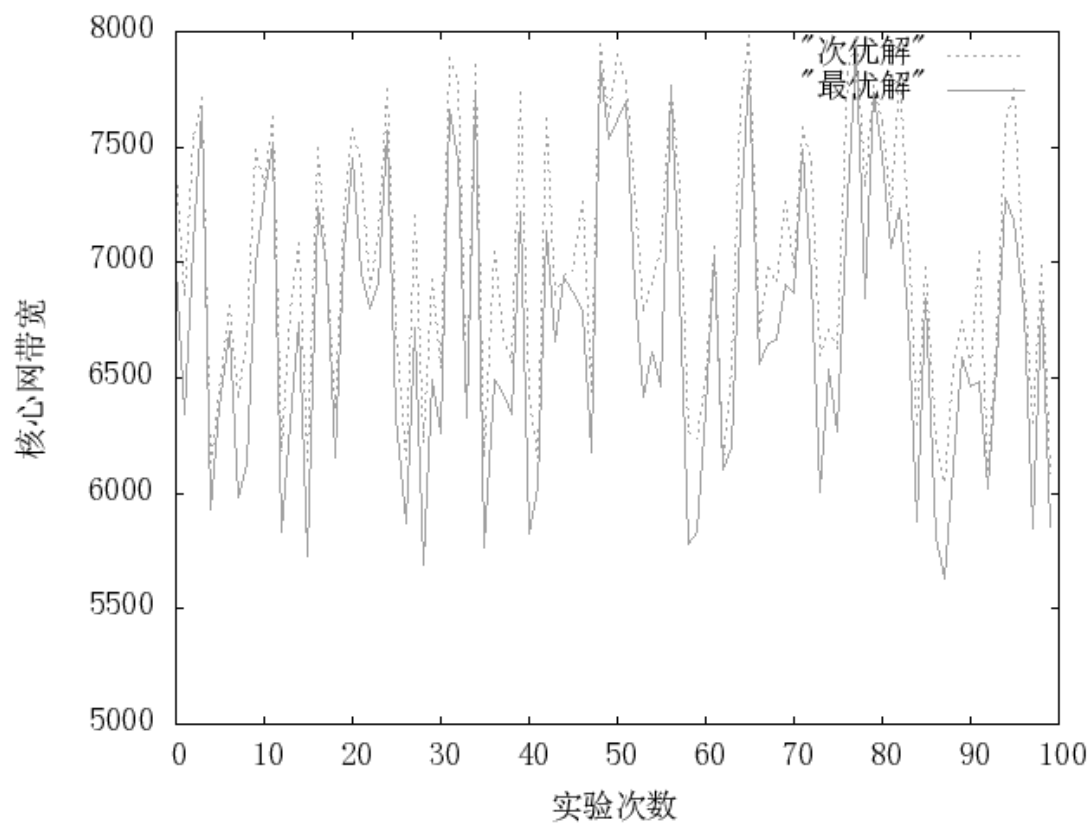


图 4.2 最优解和次优解结果对比（单位：TB）

图 4.3 所示的是三种解法的时间曲线。单位为秒。

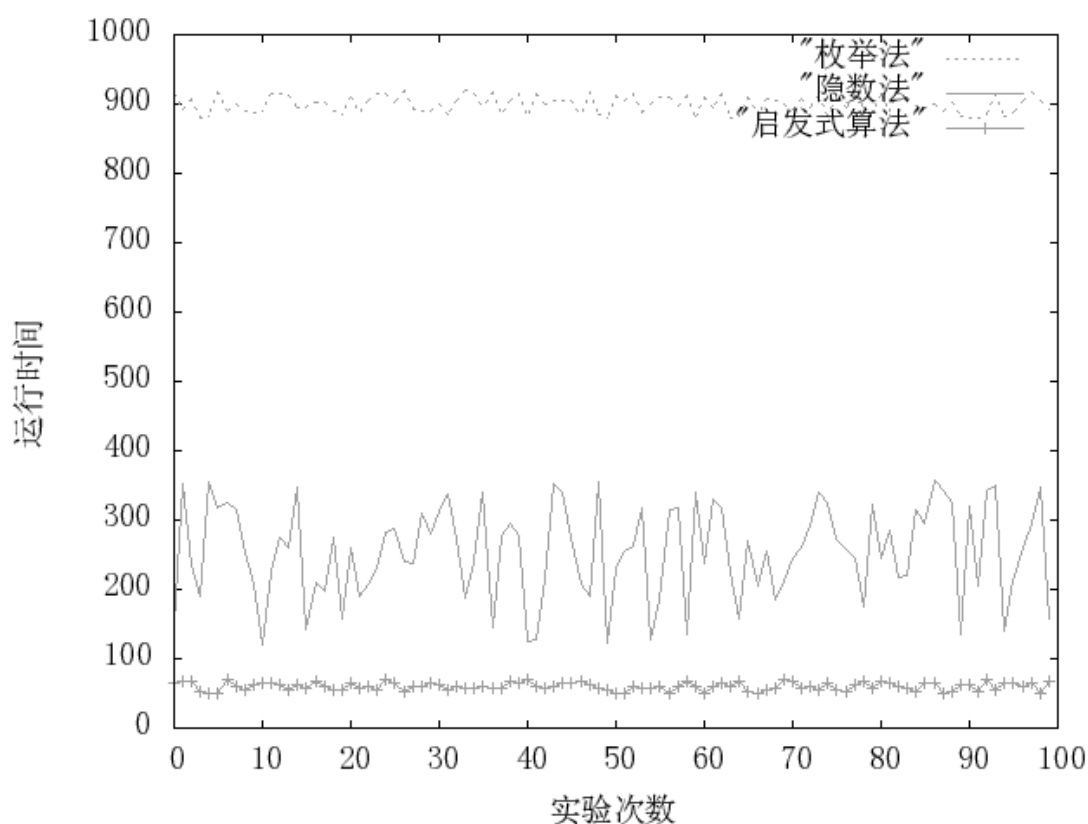


图 4.3 三种算法的运行时间对比（单位：秒）

由于枚举法求解做法是遍历决策变量组的所有取值情况，每种情况都仅将变量带入目标函数求值，且 100 次实验中决策变量数量是一定的，因此每次实验用时也基本是相同的。启发式解法运算时间很快而且也比较稳定，求解时间稳定是因为它的求解步骤是每解一次线性规划剪枝一次，而多数情况下剪枝次数只比变量个数略少。又变量数量相同时，求解一次线性规划的时间相差也不大，因此每次随机实验启发式算法的运行时间也相差不大。隐数法求解时间介于二者间，且变动比较大，这是因为隐数法剪枝速度主要取决于剪枝快慢，这与很多偶然因素相关，比如第一个可行解的好坏。当第一个可行解就很接近最优解时，后面的剪枝会非常快，反之则基本不剪枝，完全遍历。

为了反映启发算法的性能，我们统计了 100 次实验中启发解和最优解的比值。启发式算法得到的解与最优解百分比平均值是 90 %，证明求解效果是很好的。

从求解时间上看，启发算法用时仅为隐数法的 1/4。实际上从时间复杂度上分析，

已有的解整数规划问题的解法同隐数法原理相同，都基于枚举的思想，其时间复杂度是随着变量的增加呈几何级数增长的。而我们的启发式算法最多只求解  $n$ (变量数) 次单纯形法，而普遍认为单纯形法也是多项式时间复杂度的，因此整个算法也是多项式时间复杂度。本次实验给出的变量数目较少仅为 50，而实际的 CDN 网络应用中变量数目远不止如此，比如 100 个服务器和 50 个内容的情况下，变量数目为 5000，用时间几何级数增长的隐数法根本无法在需要的时间内解出最优解，这时就只能退而求其次采用启发式算法。

#### 4.2.2 服务器损耗模型仿真

本仿真实验验证了在数据规模极大的应用环境下，我们提出的启发式算法仍然能很快的求出模型的次优解；并与模型最优解的下限作比较，验证解法的可靠性。

这里的仿真数据直接采用 4.1.2 节中提到的蓝汛公司实际案例的数据。案例内容如下：

公司一个大区内共有 100 多个节点，每个节点内部有 1~14 个服务器，我们在仿真实验中取平均值，即规定节点数为 100，每个节点内服务器数为 7。每个节点和各个 ISP 骨干网连接的总带宽为 1GB，各个频道的大小固定为 1.5TB，每个服务器的服务能力为 200MB。频道总数为 5，一周内每个频道的流量预测如图 4.4。明显在周一到周五各个频道访问量都比较稳定，周六和周日的访问量会剧增。由于这里提供的流量预测以天为单位，所以我们实验中的时隙也是一天。即每天进行一次调度，上一天的求解结果为下一天的初始条件。

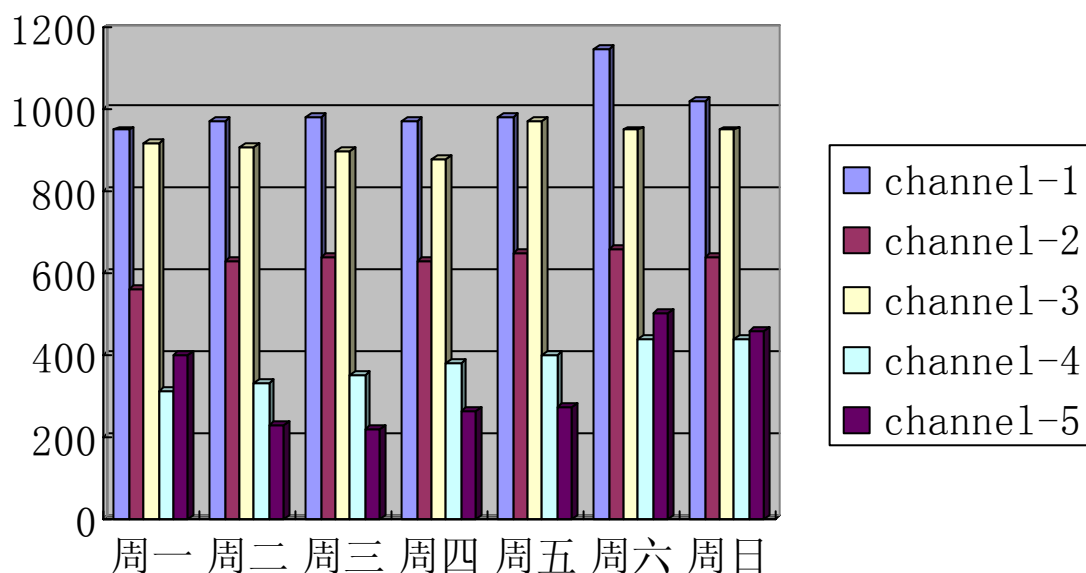


图 4.4 一周内频道的访问量预测（单位：TB）

我们进行了 10 组实验，每组实验先随机地给出各个服务器上的初始存储频道号，然后用启发式算法连续求解周一到周日 7 天内每天的频道更换个数。

为了验证启发算的性能，我们还将给出最优解的下限来做对比，这里我们给出两种方法来估计最优解的下限，在每次实验中取其中较大也即是较接近的一个为准。

估计 1：首先将问题（4.1）完全松弛为线性规划，即去掉决策变量组  $X$  的整数限制。然后求解松弛问题得  $C_1$ ，则  $C_1$  为问题（4.1）最优解的一个下限。由于（4.1）的目标函数是整数，因此取不小于  $C_1$  的整数  $[C_1]$ ，也一定是最优解的下限，显然这个下限更接近一些，因此取  $[C_1]$  为最优解的下限。

估计 2：提取出（见 4.1.2 小节）步骤 2 的求解结果  $C_2$ 。这个解的物理含义是：“保持初始条件不变，尽可能的用现有配置，最多还有多少访问量不能被满足？”然后将  $C_2$  除以每个服务器的最大服务量 200MB 然后向上取整，其结果  $[C_2 / 200]$  也必然是最优解的下限。这个很容易想通：更换一个服务器上的存储频道，肯定不如加入一个可以满载的服务器更能提高服务能力。

表 4.3 给出了两组典型实验的结果。这里的结果指所以服务器上改变的频道数总和。第一列是用启发算法求出的解，二、三列为上述两个方法分别估计出的最优解下限，第四列为根据二、三列结果得出的最终估计。

表 4.3 两组典型实验的结果（单位：个）

（实验一）	启发解	最优解估计 1	最优解估计 2	最终估计
周一	2	2	2	2
周二	5	4	4	4
周三	4	2	1	2
周四	3	3	2	3
周五	12	4	5	5
周六	17	5	4	5
周日	9	4	4	4

（实验二）	启发解	最优解估计 1	最优解估计 2	最终估计
周一	0	0	0	0
周二	2	1	1	1
周三	0	0	0	0
周四	4	3	2	3
周五	6	4	5	5
周六	7	3	4	4
周日	4	3	2	3

与文章[28,42]的做法类似：在无法给出最优解时，我们直接用启发解和最优解下限估计的比值作为解法性能的评价标准。上面两组实验是表现最差和最好的两组。实验一在周六计算出的结果最差，最优解的上限为 5，但是启发解是 17，按百分比计算性能只有 29 %；而实验一的平均性能也只有 50%。实验二中同样是在周六，性能

有 57%；平均性能达到 70%。

接着我们分析二者差异的原因。进行的所有 10 组实验除了初始条件（即一开始各个服务器存储的频道）不同以外，其余条件均相同。观察两组实验的初始条件，发现实验一的初始条件中，频道存储比较集中：例如节点 1 中 7 个服务器都存储频道 1，节点 2 中 7 个服务器都存储频道 2。而实验二的初始条件中，频道的存储在各个节点中分布相对更平均，即每个节点中都存储多种频道。设想如果一个频道只集中存储在少数几个节点中，那么在该频道访问量不断增加时，这几个节点的访问量终究会超过带宽限制；反之如果分布比较平均，当此频道访问量增加时，可以由存储此频道的多个节点都满载服务该频道，而忽略这些节点上存储的其它冷门频道。让频道分布更平均也是负载均衡的一种措施，用户访问量会平均地分配到各个节点上。因此我们将解法稍作修改，以解决频道分布不平均的情况：当步骤 4 中出现子问题目标函数相同的情况时，优先选取存储频道集中的节点。

实际实验时还发现程序运行时间非常短，在所有实验中最长的一次运算时间也不超过 15 分钟，相对于时隙为一天的调度模型来说，似乎太短了一点。因此我们还在原算法基础上加入一个扩展参数，用时间复杂度换取实验精度：还是在原解法的步骤 4 中，我们保留  $n$  个目标函数最好的中间结果，而不是只保留一个。显然这样运算时间会和  $n$  成正比，但是求解结果会更好。

表 4.4 10 组实验耗时统计（单位：秒）

日期	平均耗时	最大耗时	最小耗时
周一	52	215	10
周二	125	236	64
周三	87	190	12
周四	192	312	111
周五	376	634	257
周六	460	893	342
周日	265	430	210

我们把程序加上以上两点小改进后（ $n$  取 3），在实验一的初始条件下重做一次，然后与原实验结果作对比，如图 4.5，在周五周六两天的求解中都体现了出了较好的结果，平均性能比原算法提高了 10%。

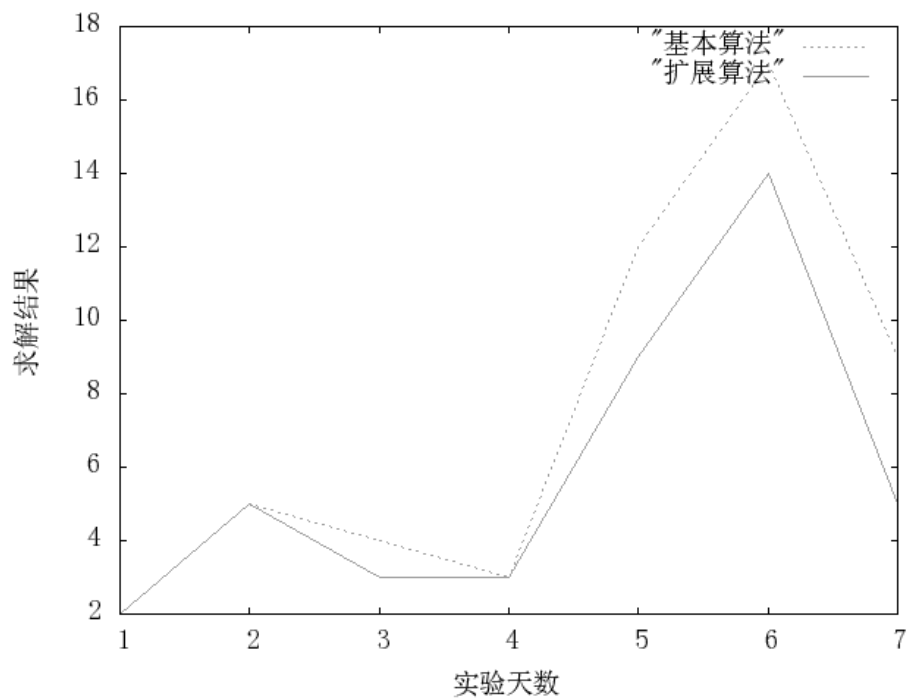


图 4.5 扩展后算法和原算法结果比较（单位：个）

二者耗时对比如图 4.6。



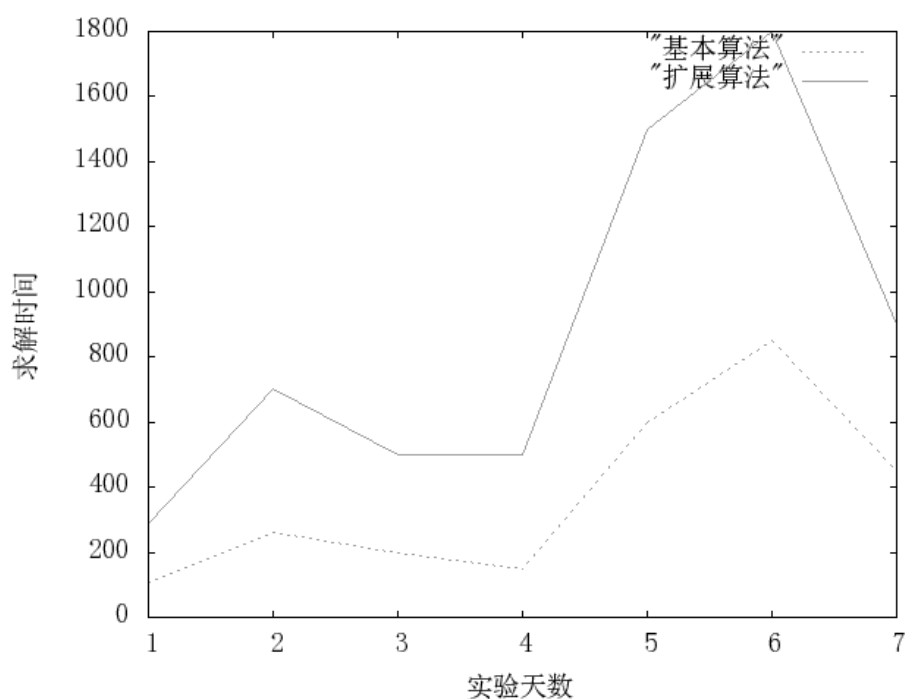


图 4.6 扩展算法和原算法耗时对比（单位：秒）

由于改进算法中每次迭代保留了 3 个中间结果，所消耗时间也大增大了许多。如果继续增加中间结果数目  $n$ ，则运算时间也会成正比增加，而求解结果的优化速度显然会放缓。我们在多次实验中发现  $n$  取 2 或 3 比较合适。

## 5 总结与展望

内容分发网络是为了改善网络负载不均, 增加用户满意度的网络技术。最典型的内容分发网络的有 P2P 与 CDN 两种。本文分别介绍了二者的特点与发展现状, 并指出 CDN 更适合于商业化应用, 所以其发展前景和空间更大。论文简要介绍了 CDN 网络的具体内容, 实现整个网络运作的两个核心技术。内容路由是将用户的请求导向整个 CDN 网络中的最佳节点。而内容分发又分 PUSH 和 PULL 两种, PULL 是比较简单而被动的办法, 即各个边缘服务器根据本地内容 miss 的情况决定内容分发, 因而目前比较热门的研究方向就是更高效的 PUSH 方法。文中给出了几个比较典型的调度模型, 分服务器放置与内容调度两种, 二者均用最优化理论建模, 而且由于求解最优解的复杂度太大, 各个文献中一般都会给出启发式解法。

本文提出了一个商用环境下的实际内容分发模型, 模型分析完全从 CDN 运营商视角出发, 其目标归结为两点: 占用骨干网总带宽最小、服务器损耗最小。以往的类似文章在研究多目标问题时一般采用乘权值再相加的方式。而本文采用的方式是先假设两种应用环境, 在每种应用环境中都分别有各自的主要目标, 从而忽略掉另外的一个次要目标。这样便衍生出两个不同的子模型, 分别以最小化骨干网带宽和服务器损耗为目标。不仅物理意义更明确, 也使求解更简单。

两个子问题均采用节点聚类的方式, 即把临近用户和代理服务器聚类为单个节点考虑, 使问题的重心从边缘服务器转移到整体网络规划上, 同时也大大简化了模型规模。在此基础上, 我们分别给出了二者的最优化数学模型并给出求解。最小化骨干网带宽问题最终化了一个标准 0-1 整数规划模型, 我们不仅用标准的隐数法求解此模型, 还给出了自己的启发式解法, 并通过大量实验表明, 在线性求解时间内, 启发解的性能达到了最优解的 90%。最小化服务器问题的数学模型属于混合整数规划, 变量数目很多时根本无法求最优解。因此我们只能用自己的启发式算法求解, 并用估计的最优解的上限与之对比, 以反映启发解的性能。我们还在实际仿真时发现了两个可以改进的小点, 通过对比实验表明, 以消耗求解时间为代价, 算法性能

可以提高 10%左右。

本文提出的实时内容调度方法在几个方面还有一些不足的地方，希望在以下几个地方能有所改进：

- 最小化骨干网带宽问题建立的模型中。限制条件中只考虑了骨干网到节点的入带宽限制，而忽略了节点到骨干网的出带宽限制。这个限制条件不好加到模型中去，因为在决策变量还没确定，即还不知道哪个节点存哪些内容时，根本无法指明某内容 miss 时，具体由谁帮忙服务。文章[8]在建模时出现了与我们类似的情况，最后模型变得十分复杂，没有给出解法。
- 最小化服务器损耗问题是实际存在的，即蓝汛公司给出的案例，有实测数据，所以仿真结果更能说明问题。反观最小化骨干网带宽模型更多是从文献[5-10]中总结出的，仿真用数据都是自己随机设的，说服力不太够。
- 如 4.2.2 小节中所说，我们在仿真过程中发现了两个可以改进的小点：令频道在各个节点中更均衡地存储；用运算时间的增加换取更好的性能。然后做对比实验发现结果确有提高。实际在做仿真的过程中发现的问题并不止这两点，但因为时间关系我们只挑选了影响最大的做改进。例如在极苛刻的边缘条件下，还应加入求基本的可行解的方法，不至于让程序陷入死循环。
- 虽然文中给定的是实时调度模型，但是都仅研究了一个时隙内的静态模型，即每个时隙的目标都是最小化当前代价。更完美的方法应该是能根据整体的流量预测达到整个时段的目标代价最小。

## 致 谢

马上就要结束学校生活踏入工作岗位了，在此回忆六年来的大学时光，不禁感慨万分，学到的东西很多，也包括一些不该学的；没学到的也很多，包括很多应该学的。在此临别之际，特别感谢帮我一同走过的长辈、伙伴们。

导师刘文予教授是我亲生经历中最佩服的人之一，感谢刘老师给我的教育和做出的榜样。刘老师在工作上的严谨态度让我记忆最深。记得一天傍晚刚下车到珠海，他便与张江山老师一同来跟我们开会讨论珠海工作的进展，激烈的会议一直持续到凌晨两点。当时我个人观察，一旁的张老师已经撑不住快睡着了。刘老师同时为人十分包容大度，这一点全实验室公认。我几次因为个人的一点小事找刘老师帮忙，百忙中的刘老师都热心相助。祝刘老师工作顺利，生活顺心，身体健康，阖家幸福。

在这里也非常感谢张江山老师在珠海时给我们的指导，以及玩 CS 时无情地虐我和张奎。

衷心感谢带我做毕业设计的张帆师兄，谢谢你对我不耐其烦的教导，让我感受到了实验室的温暖。感谢老大田臣，不仅教会我知识，也教会我学会许多做人和打牌的技巧。感谢蒋洪波老师为光电实验室带来积极的工作环境和掏腰包赞助的许多娱乐活动。感谢光电实验室与我一同上班吃饭调侃的刘春晓、熊雄、陈金华、叶小琼、舒乐同学，与你们在实验室度过的两年，让我感觉很快乐很充实，祝大家以后前程似锦，万事如意。

感谢两年来和我关系最好的牛人张奎，很遗憾不仅没有从你那里学到很多知识，还将你带坏玩 dota；感谢慷慨大方的郝未请我在西门外和群光吃的 n 顿饭；感谢周彦为人处事的积极乐观态度对我的影响；感谢心宽体胖的徐汉为实验室带来的笑声；感谢和我一同去过珠海莲花路的刘瑜、焦夫；感谢珠海项目组的全体兄弟姐妹，祝大家在日后的学习工作中不断进步，早日实现自己梦想。

感谢我的父母，亲情最无私。

值此毕业时节，再次感谢所有对我给过直接和间接帮助的人们，虽然以后各奔东西，美好的祝愿常在。

## 参考文献

- [1] 钟瑛, 我国互联网发展现状及其竞争格局, 新闻与传播研究, 2007, 13(4): 29-33
- [2] 夏淑华, 浅谈 P2P 技术应用及存在的问题, 警官论坛, 2007, 6(4): 178-184
- [3] 金世杰, 赵问道, CDN 网络路由技术, 计算机应用研究, 2003, 5(5): 156-159
- [4] 杨明川, CDN 中的负载均衡技术, 信息网络, 2003, 8(2): 45-49
- [5] 王樟, 柳健, CDN 网络中的内容分发技术研究, 网络与应用, 2004, 7(2): 87-98
- [6] 姜文颖, CDN 网络中八种负载均衡实现技术的探讨, 中国数据通信, 2004, 8(3): 134-145
- [7] 董梅, 肖民, 宽带流媒体领域的 CDN 技术, 广播电视信息, 2005, 3(2): 121-125
- [8] 徐卫东, 王康, 适用于内容分发网络的动态负载均衡策略, 计算机科学, 2005, 10(2): 48-51
- [9] 宋家友, 桑红涛, CDN 技术的发展及应用, 宽带网络与传输, 2005, 7(1): 25-30
- [10] 马明霞, 张学军, CDN 技术及其在图书馆局域网中的应用, 现代图书情报技术, 2004(4): 13-22.
- [11] 聂秀英, 网络内容分发技术, 电信科学, 2004, 3(2): 36-50.
- [12] Y.Chen, R.H.Katzdynamic, Replica placement for scalable content delivery, in: Proc of 1st Int. Workshop on Peer-to-Peer Systems, 2002, 5(3): 124-126
- [13] M.Chen, J.P.Singh and A.LaPaugh, Subscription-enhanced content delivery, in: Proc of WCW'03, 2003, 3: 56-78
- [14] C.Huang and T.Abdelzaher, Towards content distribution networks with latency guarantees, in: Proc of 12th IWQoS, 2004, 12(4): 123-127
- [15] J.Kangasharju, J.Roberts, and K.W.Ross, Object replication strategies in content distribution networks, in: Proc of WCW'01, 2001, 1: 43-67
- [16] Yannis Matalas, Nikolaos D, A Scalable Framework for Content Replication in Content Distribution Networks, Journal of Algorithms, 2001, 11(2): 110-115.

- [17] M.Korupolu,G.Plaxton,and R.Rajaraman, Placement algorithms for hierarchical cooperative caching, *Journal of Algorithms*, 2001,4(2):79-95
- [18] A.Bestavros, WWW Traffic Reduction and Load Balancing Through Server-based Caching, *IEEE Concurrency*, 1997,6(4)104-135
- [19] 陈宝林, 最优化理论与算法, 清华大学出版社, 2005.10
- [20] Luenberger D G. *Linear and nonlinear programming*. Addison-Wesley, 1984
- [21] Lili Qiu, Venkata N.Padmanabhan, On the Placement of Web Server Replicas, *IEEE INFOCOM*, 2001,3(5)145-167
- [22] Kazem,and S.Guha, Improved Combinatorial Algorithms for the Facility Location and K-Median Problems, in: *Proc of the 40th Annual IEEE Conference on Foundations of Computer Science*, 1999,6:89-102
- [23] L.Qiu,V.N.Padmanabhan,and G.M.Voelker, On the placement of Web server replica, in: *Proceedings of IEEE INFOCOM*, 2001,7:78-98
- [24] S.Jamin,C.Jin,Y.Jin,D.Raz,Y.Shavitt,L.Zhang, On the Placement of Internet Instrumentation, in: *Proc.of INFOCOM'2000*, 2000,6:56-63
- [25] B.Krishnamurthy and J.Wang, On Network-Aware Clustering of Web Clients, In: *Proc of SIGCOMM'2000*, 2000,4:25-50
- [26] B.Li,M.J.Golin,G.F.Ialiano,and X.Deng, On the Optimal Placement of Web Proxies in the Internet, in: *Proc.of INFOCOM*, 1999,8:84-92
- [27] E.W.Zegura,K.L.Calvert,and S.Bhattacharjee, How to Model an Inter- netWork, in: *Proc of INFOCOM*, 1996,7:87-97
- [28] Jussi Kangasharju, James Roberts, Object replication strategies in content distribution networks, *Computer Communications*, 2002,7(2):43-77.
- [29] Yan Chen, Lili Qiu, Efficient and Adaptive Web Replication using Content Clustering, *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, 2003,10(2):56-72.

- [30] Padmanabhan,S.Kutten,R.Soffer, Optimal allocation of electronic Content, in: Proceedings of IEEE Infocom, 2001,2:124-143
- [31] S.Jamin,C.Jin, Constrained mirror placement on the Internet, in:Proceedings of IEEE INFOCOM, 2001,4:32-56
- [32] NLANR measurement and operations analysis team. <http://moat.nlanr.net>.
- [33] A.Luotonen and K.Altis, World-wide web proxies, in: Proc of the
- [34] First International Conference, 1994,12:65-86
- [35] T.P.Kelly,Y.-M.Chan,S.Jamin,and J.K.MacKie-Mason, Biased replacement policies for web caches:Differential quality-of-service and aggregate user value, in: Proc of the International Web Caching Workshop, 1999,6:123-134
- [36] V.N.Padmanabhan and J.C.Mogul, Using predictive prefetching to improve world wide web latency, in: Proc of ACM SIGCOMM Computer, 1996,6:45-59
- [37] P.Wolfe, and H. P. Crowder, Validation of Subgradient Optimization, Math Programming, 1974,16(2):62 - 88
- [38] Tolga Bektas,Jean-Fran, Exact algorithms for the joint object placement and request routing problem in content distribution networks, Computers&Operations Research, 2007,2:43-58
- [39] uanping Z,Weidong W,Xiaopeng T,Yonghu Z, Data replication at web proxies in content distribution network, Lecture notes in computer Science, 2003 ,10(5):560–569
- [40] Kangasharju J,Roberts J,Ross KW, Object replication strategies in content distribution networks, Computer Communications, 2002,3(4)156-178
- [41] Tassiulas L, Market-based resource allocation for content delivery in the Internet, IEEE Transactions on Computers, 2003,4(2):78-96
- [42] Datta A,Dutta K,Thomas H,VanderMeer D, World wide wait:a study of Internet scalability and cache-based approaches to alleviate it, Management Science, 2003,4(3):32-60

- [43] Radoslavov P, Govindan R, Estrin D, Topology informed Internet replica placement, Computer Communications, 2002, 5(5)113-130