哈尔滨工业大学

硕士学位论文

基于CDN的服务器放置策略研究

姓名：韩笑

申请学位级别：硕士

专业：计算机科学与技术

指导教师：李斌

20050601

摘　要

随着网络技术的快速发展，网络提供的服务也趋向于多样化，电子商务、网络流媒体等业务的出现使用户对网络的性能要求也越来越高，越来越多的网站遭受到网络拥塞的困扰。为了提高用户访问网站的响应速度、优化现有网络中信息的流动、提高网站的安全性和可用性，可以在现有网络中建立一个完善全面的中间层网络——内容分发网络（CDN）。本文研究了内容分发网络中代理服务器的选择放置问题，希望通过将内容分发到最适合的边缘代理服务器上，来更好地调整内容分发网络结构，降低用户访问延迟。本文将网络拓扑上的多服务器放置问题转为网络节点的聚类问题，分别考虑代理服务器数目的确定和不确定的条件，以节点间的 ping 延迟作为网络结点之间的“距离”，采用经典的 k-means 算法和 ISODATA 算法来进行网络节点聚类和服务器定位，从而使内容分发网络在总体上达到最优配置。针对聚类算法面临的初始点选择问题，本文提出了基于迭代分裂的初始点选择方法，分别与 k-means 和 ISODATA 算法相结合，提高了算法的稳定性，减小了聚类结果对初值的依赖。另外，通过在聚类过程中考察类成员数量，删除较小的类，避免了孤立点对聚类结果的影响，能够较好的均衡 CDN 网络中服务器的负载，合理地配置服务器。本文在 NS 网络模拟条件下，对 CDN 的服务器放置问题进行了模拟，并实现了所提出的算法，获得了很好的结果。由于算法不受网络拓扑结构的限制，所以比较适用于真实的网络环境。

关键词　内容分发网络；服务器放置；k-means；ISODATA；迭代分裂

Abstract

With the fast development of network technology, Internet users are getting better service. But we need a high-performance, reliable vehicle for the appearance of bandwidth-intensive, rich multimedia content such as ecommerce transaction and stream m edia. CDN is the current solution schemes to advance the user's access speed guarantee the network security. This paper studies the placement of proxy server in Content Distribution Network. It aims to place the context to the most compatible fringe proxy server in order to modulate the context placement network's configuration and slow down the delay of user access. This paper transforms the placement of multi-services into the clustering of network node, considers the certain and uncertain qualification of proxy server separately, takes the ping delay between network nodes as the "distance", and adopts the classical k-means algorithm and ISODATA algorithm as to cluster the network and place the services so to place the context to the best network totally. Contraposing the preliminary node in the clustering algorithm, this paper proposes the recursive separating the preliminary node, and combines with k-means algorithm and ISODATA algorithm in order to enhance the stability of the algorithm and decrease the dependence of clustering result on the preliminary value. Furthermore, it reviews the the number of class members in the clustering process, deletes the smaller classes, avoids isolated nodes' influence on clustering results, and equipoises the CDN service load and configures the services reasonably. Under the NS network simulation, this paper simulates the placement of services in CDN, it also realizes the proposed algorithm and gets better results. Because the algorithm isn't confined to the network topology, it is suitable for the real network environment much more.

Keywords CDN; server placement; k-means; ISODATA; recursive separating

第1章 绪论

1.1 课题研究的背景

在网络技术飞速发展的今天，Internet 已经成为信息社会的基础载体。不过恰恰由于 Internet 的日益流行导致网络用户急剧增长，大大超过了现有网络的负担能力。长期以来，网络拥塞、服务器负载过重、网络访问延迟等问题都困扰着用户。

在互联网业界，存在着互联网的“8 秒钟”现象，即 35%的用户在等待 8 秒钟后放弃连接，80%的用户在等待 12 秒后放弃连接。据统计，在 2001 年，由于缓慢的下载和丢包造成的网站交易损失超过$250 亿，也就是说，8 秒钟的等待意味着门户永远失去一个网民。如何确保服务体系结构能满足日益增长的流量和电子商务，如何为成千上万的并发用户提供亚秒级的响应速度，是目前众多网站和.com 企业所面临的巨大挑战。从某种意义上讲，对于 ICP 来说，网站信息传递速度已经成为关系他们生存、发展的关键性问题。

互联网的飞速发展拓宽了人们的视野，丰富了人们的经验，随着技术的发展，人们有理由要求更多，他们希望能够通过网络实时地接收到外面的信息，能够了解科技、政治、生活等方面最前沿的动态，人们还希望通过在线观看等方式接收最及时的报道。要实现这些功能，就必须要解决网站响应速度慢的问题。

1.1.1 传统的 Internet

传统的Internet访问需要用户通过DNS解析域名获得IP地址后直接对网站进行访问。这种模式虽然简单，但访问效率低下，这就无法保证Internet用户的访问质量。Internet是一个开放的网络，在这个开放的网络上，访问和寻址都是通过分布在网络上的众多的路由器来将数据包从一个网段传递到另一个网段上，由于地域分布的广泛，用户对网站的访问必须经过许多路由器的转接，才能最终到达网站的WEB服务器，中间可能要跨过多个ISP和网络，即使在理想情况下，路由器的每一次转接(HOP)，都会造成一个延迟，虽然单个这样的延迟时间较少，但当网络规模很大时，HOP数会大量增加，造成的延迟也会更加明显[1]。而且，任何一个路由器的故障或者拥塞都可能造成

访问的中断或者延迟，严重影响访问的质量。不同网络之间的互联是通过对等点(peering points)[2]连接在一起，这些互联点的带宽比较狭窄，也是导致Internet访问速度慢的一个重要瓶颈。用户对网页的访问是通过HTTP协议向服务器发出请求，这样用户要在浏览器上看到一个完整的页面就需要与原服务器的多次交互访问才行，使由于路由器转接引起的延迟对页面访问过程的影响成倍增加。

由于IP协议中不同物理网络对帧大小的限制不同，使网页中的一个Object可能要被分解到更多个数据包中进行传递，进一步加大了路由器转接延迟引起的总延迟时间。

网页响应时间= 页面大小/网络最小带宽+往返次数×每次往返时间+处理时间

从上面的计算公式可以看到，通过提高网络的带宽可以降低网页的响应时间，但要注意，网络中的瓶颈不一定是用户的接入点，而很可能是不同网络运营商之间的对等点。对于同一个用户下载同一个页面，第一项即页面的大小与网络最小带宽的比值是基本一致的。第三项处理时间是指服务器和用户的PC机的处理能力，该项随着网站服务器的负载而变化。所以网站的服务器处理能力越强，其响应时间相对也会更快，在相同条件下，公式的第二项将会对响应时间产生重要的影响。当用户通过HTTP协议访问网站时，首先必须建立一个TCP连接，然后才能向WEB服务器发送GET请求，同时等待服务器的响应。由于现在页面的设计越来越复杂，页面中的Object的数目往往多达30-50个，使往返次数增加。而每次往返时间则会因为经过的路由器的增加而增加。所以缩短网站与用户之间的距离是提高访问速度的一个有效方法，即将用户要访问的内容推送到离用户最近的地方。

1.1.2 网站镜像

为了能将网站的内容安置到离用户更近的地方，人们提出了网站镜像的解决方案，但它的应用同样存在几个重要的缺陷[3]：

1) 对用户不透明

网站在设立镜像站点后，必须要给每个镜像站点设立相应的域名，由于用户访问通常是直接访问其主站，因此必须在主站的显要位置上放置各镜像站点的链接，这样用户才能访问到相应镜像站点。这样的设置增加了用户访问的复杂性，而且在镜像站点由于故障停机后，链接不能有效及时的修改，

导致部分用户通过链接对镜像站点的访问得不到服务，影响服务质量。

2) 数据同步困难

传统镜像站点使用服务器，将原站点的全部内容通过定时同步的方式拷贝过来，同步周期长，不灵活。

3) 镜像站点较少

由于设立镜像站点的周期长，技术要求高，维护监控费用高，一般只有较大的网站或公司才能在重要城市设立镜像站点，因此服务的范围有限。

通过对传统Internet和网站镜像的分析，可以看出它们对响应速度的提高并不明显，内容分发网[4-7](Content Distribution Network简称CDN)这一概念的提出正好解决了这方面的问题。它可以实现把网站内容高效、稳定地发布到离网民最近的地方，将网站源服务器的内容存储到分布于各地的缓存服务器上，通过网络的动态流量分配控制器，将用户请求自动指向健康可用并且距离用户最近的缓存服务器上，从而为广大客户提供了内容丰富的高质量服务。

CDN技术是近年来在美国首先兴起并迅速发展起来的一种解决互联网性能不佳问题的有效手段。其基本思路就是尽可能避开互联网上有可能影响数据传输速度和稳定性的瓶颈和环节，使内容传输的更快、更稳。通过在网络各处放置节点服务器所构成的在现有的互联网基础之上的一层智能虚拟网络，CDN系统能够实时地根据网络流量和各节点的连接、负载状况以及到用户的距离和响应时间等综合信息将用户的请求重新导向到离用户最近的服务节点上。由此可见，服务器的放置[8,9]以及内容的复制策略对于内容分发网络的性能好坏是至关重要的。

本文基于现存的CDN网络的代理服务器放置算法，提出了一种改进的k均值算法[10]和ISODATA算法[11]用来进行网络节点聚类和服务器定位，解决了CDN网络中如何放置代理服务器的问题。

1.2 课题研究的意义

CDN技术可以有效优化带宽分配，缓解我国互联网信道资源长期紧张的局面。由于它可以将用户经常访问的内容分送到整个CDN中，原先很多需要出国才能访问到的内容，现在从国内就可以快速访问得到。CDN技术的这一特点不仅可以大量节省我国Internet高额的国际信道租用费，而且还可以大幅度降低信道故障带来的风险，这一点在2001年2月9日发生的中美海底光缆受损事件中得到了充分的证明。在大多数ISP/ICP为这次事件叫苦不迭的时候，

一些采用CDN技术的公司则只受到了很小的影响。这次光缆受损事件使CDN技术浮出了水面。

CDN技术的另一个优点是，它可以更好的支持Internet上日益增长的对动态内容(dynamic content)和视频、音频等流媒体(streaming multimedia)内容的访问。这是因为，基于第4层一第7层交换的CDN技术可以更智能的分析用户请求和网络运行状况、更合理地分配网络带宽、调度网络资源、均衡网络负载。CDN技术的这一特点在电子商务、远程教育、视频音频传输和交互式游戏等领域有着广阔的应用前景。

正因为CDN技术可以快速、高效、廉价地提升Internet访问速度与服务质量，它已经成为Internet领域一个新的研究热点和利润增长点。国际权威调查机构HTRC Group的分析预测报告显示，CDN技术符合用户需求，顺应互联网的未来发展趋势，蕴藏着巨大的商机和利益。

CDN技术也可以与目前的城域网技术相结合，进而在全国范围内构成一个国家级的内容分发网络体系，促进我国以中心城市为纽带的区域信息化建设和应用。需要特别指出的是，北京申办2008年奥运会的成功也将会使CDN技术得到更大的优势发挥。它可以将奥运内容快速地通过Internet分送到世界各地，避免北京本地的服务器由于过度访问而导致的服务器过载和网络拥塞，为全世界互联网用户提供高速高质量的网络信息服务，将一个现代、文明、富强的社会主义中国展示在世人面前。

CDN服务器放置的研究具有重要意义，具体表现在以下几个方面：

1. 有效降低服务响应时间，提高服务质量；
2. 将大量的网络用户聚类，每类用户由一个服务器集中提供服务，有利于负载均衡，提高访问效率；
3. 每类用户的连接请求和下载发生的流量只发生在该类用户群的子网中，不对其他子网造成流量，节省了带宽；
4. 有利于合理规划网络结构；
5. 有利于web运营商合理规划服务器资源，降低运营成本。

综上所述，CDN服务器放置的研究不但具有重要的商业价值，同时对合理规划网络和减少网络拥塞有重要意义。

1.3 CDN 研究发展现状

1.3.1 国外 CDN 研究发展现状

CDN 技术是国际互联网界近几年才提出的一个网络加速概念，它的基本技术源自于网络缓存/复制、负载均衡、流量工程、客户端重定向和先进路由等技术。国外同行在网络内容缓存和复制方面开展的研究较早，研究的范围从 Web 服务器、浏览器、代理服务器，到 CacheMesh，从 HTTP 协议本身到 ICP、WCCP 缓存/复制协议，再到 ICAP 等客户端重定向协议[12]，基本上遍布了网络研究领域中与内容有关的各个方面。在此基础上，美国的 NLANR（美国国家应用网络研究实验室）实现了两个专用的缓存代理服务器 Harvest 和 Squid。Harvest 已经商业化，Squid 则仍由美国 NSF 资助，而且目前已经在美国、英国、俄罗斯、新加坡、韩国、日本等世界上许多国家取得了良好的应用效果。流量工程和负载均衡领域的研究主要集中在流量分析[13]、Mirror 站点和基于 NAT、DNS 等技术的服务器集群系统方面 [14]。先进路由技术与常规路由技术的区别在于它可以实时监控网络流量和线路拥塞状况，并对这些情况综合分析，选出一条优化的访问路径，从而提高网络访问速度[15,16]。这些技术都是从某一方面来加速网络的，但它们并不是孤立的。Akamai 首先将这些技术综合运用，在全球建立了第一个庞大的 CDN，将用户要访问的内容分送到网络边缘，使用户就近就可以快速的访问到所需的信息和资源，大大提高了网络的访问速度和服务质量。

两个 CDN 技术联盟 Content Alliance 和 Content Bridge Alliance 与 IETF 在 San Diego 举行的 CDN“Brids of Feather”会议上拟组成四个工作小组：Content Delivery Network Peering(CDNP)、Open Proxy Extension Services(OPES)、Contextualization of Resolution 和 Web Replication and Caching，共同研究有关 CDN 的各项关键技术。同时，IETF 资助并举办了五届有关“Web Caching and Content Delivery”的国际学术会议。从会议论文可以看出，整个领域的研究非常活跃，技术创新也是日新月异。为了保持国际领先地位，美国 NSF 继续大力资助 NLANR（National Laboratory for Applied Network Research）进行最新技术的研究，英国的 JANET、德国的 DFN 和新加坡的 SingNet 也都得到了政府的大力支持。

目前国外关于CDN中内容路由、内容缓存和内容智能管理方面的研究比

较多，关于代理服务器放置算法的研究还不是很多。重点还在广义上的设备放置问题上[17,18](FLP，facility location problem)，用的比较多的是各种改进的K均值算法[19,20]。如KAMAL JAIN等将k-meadian通过primal-dual schema和拉格朗日松弛算法进行扩展，应用到FLP上。

Craig W. Cameron等人列举了一个客户端和服务器都比较密集的近似模型，并且提出了一个基于高比率矢量量化理论的简单的服务器分配和放置算法 [21]，该算法是在网络主机数量巨大且分布密集的情况下，考虑了用户对内容的访问频率，通过积分来计算访问延迟代价，选取代价最小的一台或多台服务器放置访问内容。

Moses Charikar等人将网络简化为一种树结构，然后采用动态编程方法实现了将代理服务器放置到最适合的树结点上[22]。在网络呈树形时，该算法取得了较好的效果。

IBM研究中心的Lisa Amini和Henning Schulzrinne提出了一种基于流量探测的服务器放置算法[23]，该算法通过主动和被动测量相结合的方式探测用户群中的流量变化，动态的对网络用户进行聚类，选择放置服务器的位置。

1.3.2 国内 CDN 研究发展现状

CDN 技术在国际上正如火如荼的进行，而国内在这些方面的研究却相对薄弱。目前只在一些专业报纸和杂志上见到过相关的综合性报道，在国际和国内学术刊物上还没有看到中国网络界的相关研究论文。清华大学网络研究中心已经在 Internet CDN 技术领域跟踪国际前沿研究多时，具有较深厚的技术积累和进一步深入研究互联网 CDN 技术的坚实理论基础。

目前国内 CDN 研究大多集中在流媒体的内容传输和负载均衡上，对于内容分发网络服务器的放置也比较少。武汉大学的肖磊，熊大红等在 CDN 代理服务器容量有限以及内容发布者预算有限的情况下，提出了一种基于动态规划和贪心算法组合的代理服务器放置算法[24]。浙江大学的王娟实现了 CDN 内容路由的负载均衡及网络分割算法[25]，将简单算法和分割算法进行了比较，实现了分层网络的服务器放置。

1.4 课题的主要研究内容

本文借鉴国内外学者的研究成果，重点研究了内容分发网络中的代理服务器放置问题，提出了基于改进的 k 均值算法和 ISODATA 算法的代理服务

器放置算法，目标在于最小化用户访问延迟。本文把特定网络拓扑结构上的多服务器放置问题转化为网络节点的聚类问题，并将节点间的延迟代价通过网络工具 PING 实现， 用动态聚类思想解决多服务器放置问题，从而使内容分发网络在总体上达到最优配置。本文并在 NS2 网络仿真条件下，对 CDN 的服务器放置问题进行了模拟，并实现了所提出的算法，获得了很好的结果。

1.5 本文的组织结构

本文的组织包括以下几方面内容：

第一章首先介绍了阐述了内容分发网络服务器放置的背景内容和相关研究，提出了本文的研究内容。

第二章介绍了内容分发网络的结构及工作原理，并通过分析 CDN 的主要技术来探索降低访问延迟的方法。

第三章对服务器放置问题进行了描述，将 CDN 服务器放置问题转化为一般聚类问题，通过求出聚类中心来得出放置 CDN 服务器的最佳节点。通过介绍聚类的主要方法，得出分割聚类中的 k-means 算法是解决 CDN 服务器放置的一个有效办法。

第四章在确定服务器数目的情况下，提出了基于 k-means 算法的服务器放置算法，并用迭代分裂的初始点选择方法对 k-means 算法进行了改进。然后在网络仿真条件下实验模拟，并做出分析比较。

第五章在不确定服务器数目的情况下，将基于迭代分裂初始点选择方法与 ISODATA 算法相结合，得到了合理的代理服务器数目，并找出最佳放置代理服务器的位置。

第2章 内容分发网络

2.1 内容分发网络简介

CDN 的全称是 Content Delivery Network，即内容分发网络。其目的是通过在现有的 Internet 中增加一层新的网络架构，将网站的内容发布到最接近用户的网络"边缘"，使用户可以就近取得所需的内容，解决 Internet 网络拥塞状况，提高用户访问网站的响应速度。从技术上全面解决由于网络带宽小、用户访问量大、网点分布不均等原因，解决用户访问网站的响应速度慢的根本原因。

在CDN技术出现的最初阶段，服务的内容仅仅局限于网站的静态图片和静态内容，但随着Intemet的发展，传统静态内容的CDN服务已经远远不能满足用户和市场的需求。随着流媒体应用的普及，人们发现传统的集中式服务的方式已经完全不适应大规模的流媒体应用，无论是对服务器还是带宽的压力都将导致内容提供商无法为大量用户提供高素质的服务。而CDN网络通过将用户访问有效地分布到各地分节点，实现就近访问，既可以解决带宽的瓶颈又能给用户提供可靠的服务。

由于内容分布网络可以有效地提高网络资源的利用效率，在提高ISP的广域网带宽利用率的同时提高用户的访问速度，增加网站的服务可用性和抵抗黑客攻击的能力，因此受到广泛的重视，是一个正在兴起的并高速发展的技术产业。它提供了一种智能化的解决方案，通过复制源服务器内容到地理位置不同的代理服务器上，同时根据用户请求的特定内容，自动把该请求转发到含有请求内容副本且距离用户最近的代理服务器上去，从而可以避免连接到提供该内容的源服务器。重要的是该过程对用户是透明的，区别于网站镜像服务——需要用户自己选择，无法自动判断最佳站点。简单地说，CDN可使Web内容存储在Internet各个不同地理位置上，当客户访问内容时，CDN将其引导到对客户最有利的代理服务器。特定的内容存放在专门的、经授权的代理服务器里，特定的代理服务器只处理经授权的内容，这样，对存储内容的控制大大提高了访问成功率，并且改善了网络性能。

内容分发网络具有以一下几点显著优点[26]：

1、由于在地理上或者网络拓扑结构中更接近所需内容，客户将获得更快

的相应速度。

2、对各种类型内容的支持，如：HTML、图像、动态内容、经验证内容以及流媒体等。

3、高效传递互联网上内容和服务。

4、安全传输内容。

5、通过集中管理降低管理成本。

2.2 CDN 的网络架构

一般地，我们把CDN看作是一种叠加模型，即CDN可以认为是一种叠加于传统网络之上的用于向分布在各地的用户分发内容的系统。该系统具有一定的扩展性，而且与原有的网络结构无关。CDN和原网络之间的关系可以用图2-1来表示：

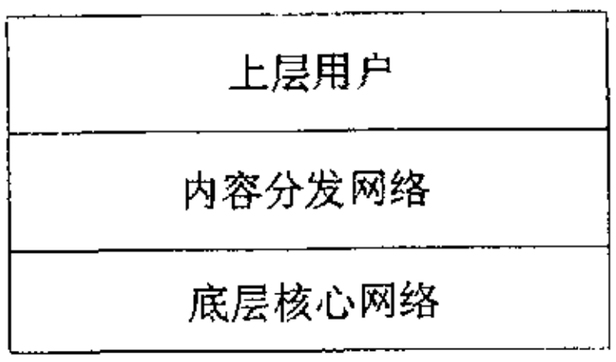

图 2-1 内容分发网络的层次结构

Figure 2-1 Hiberarchy of content distribution network

CDN 网络架构主要由两大部分，分为中心和边缘两部分，如图 2-1 所示。中心指 CDN 网管中心和 DNS 重定向解析中心，负责全局负载均衡，设备系统安装在管理中心机房，边缘主要指异地节点，CDN 分发的载体，主要由高速缓存服务器和负载均衡设备组成。

高速缓存服务器（Cache）负责存储客户网站的大量信息，就像一个靠近用户的网站服务器一样响应本地用户的访问请求。

负载均衡设备负责每个节点中各个 Cache 的负载均衡，保证节点的工作效率；同时，负载均衡设备还负责收集节点与周围环境的信息，保持与全局负载 DNS 的通信，实现整个系统的负载均衡。

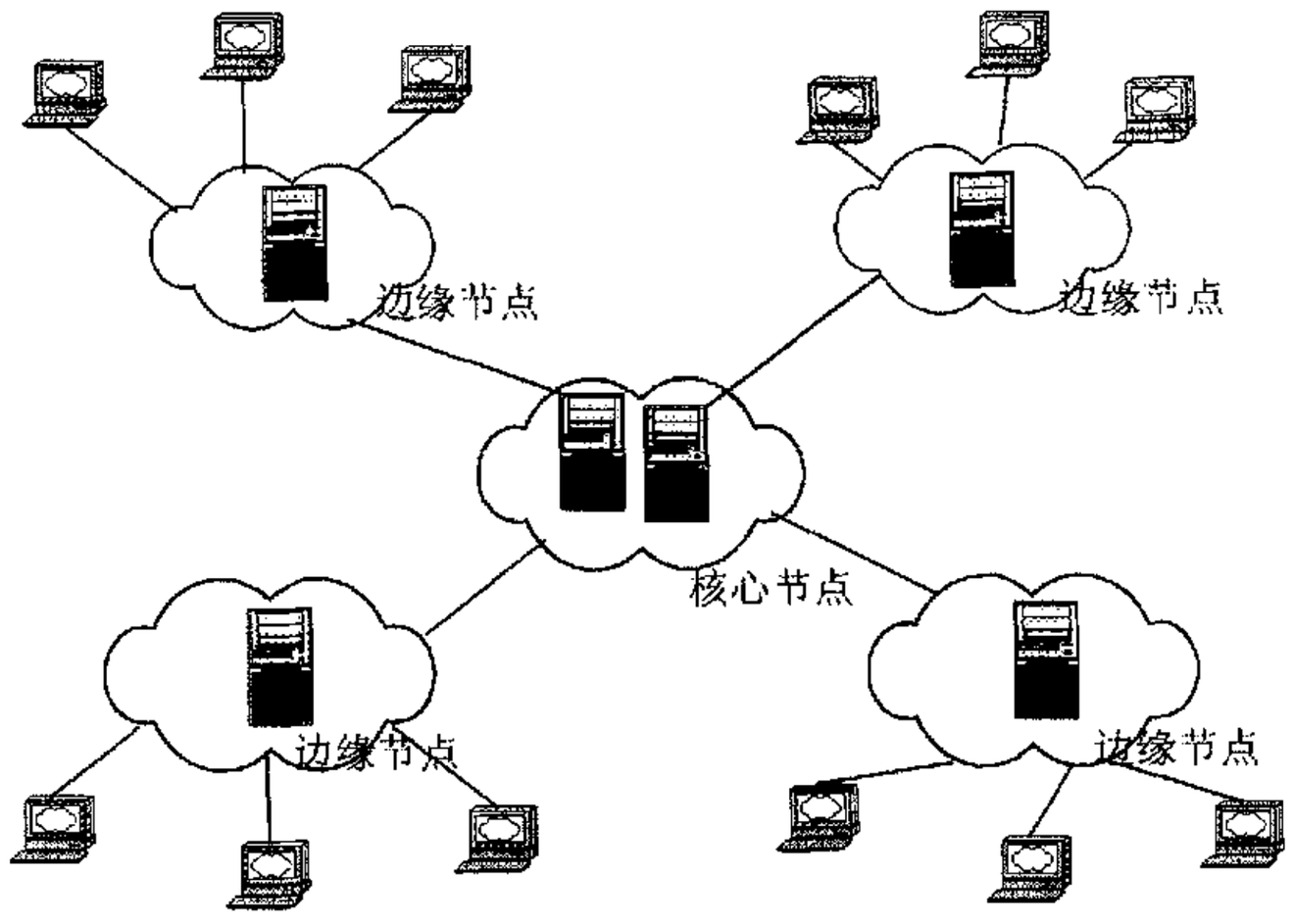

图 2-2 CDN 服务网络结构图

Figure 2-2 CDN service network frame

对于普通的 Internet 用户来讲，每个 CDN 节点就相当于一个放置在它周围的 WEB。通过全局负载均衡 DNS 的控制，用户的请求被透明地指向离他最近的节点，节点中 CDN 服务器会像网站的原始服务器一样，响应用户的请求。由于它离用户更近，因而响应时间必然更快。

2.3 CDN 的组成

CDN 的基本工作原理就是广泛采用各种 Cache 服务器，将这些 Cache 服务器分布到用户访问相对集中的地区或网络中，并利用全球负载均衡技术（GSLB,Global Server Load Balance）将用户的访问指向离用户最近的工作正常的 Cache 服务器上，由它直接响应用户的请求。如果 Cache 服务器中没有用户要访问的内容，它会根据配置自动到原服务器去抓取相应的页面并提供给用户。可以说，一旦 CDN 网络正常运行后，内容的分布和用户的访问定位全部是自动的[27]。

从概念上看，CDN 并不依赖于非常复杂或难于理解的技术，但要建立一个高效可靠的 CDN，至少需要 5 个基本部分。

1、原始服务器：为了便于管理和配置，CDN 网络中的各个节点一般都使用 Cache 服务器来保存用户频繁访问的网站内容，这些服务器中只保留一

个副本。因此，网站的内容必须要有原始存放的地方，所以要根据网站的大小用一个或多个服务器用来保存原始的 Web 内容。大部分情况下，原始服务器安装在 IDC 中，由内容提供商自己负责维护和管理。

2、GSLB 控制服务器：只有有效地解决了可扩展性和可用性的 CDN 网络才称得上是一个有价值的解决方案。GSLB 控制服务器是专用的具有高可靠和高冗余性的设备，主要用来将流量指向到那些正常工作的 Cache 服务器上，并在它们工作异常时及时将流量转走。这些控制器应该能够直接或间接地取得分布在各地 CDN 节点中的 Cache 服务器的工作状态和性能，并可以判断用户的来源，以保证用户有效地分配到离其“逻辑上”最近、最健康的节点上。

3、CDN 节点—Cache：和本地负载均衡服务器（SLB） 在各个节点上，Cache 服务器扮演 Web 服务器的角色，将保存原始服务器上相应内容的副本，以便及时响应用户的请求。目前有两种方式可以将内容分布到各节点的 Cache 上：一种是“拉”的方式，Cache 在接受用户的请求后才代替用户到原服务器去抓取内容，同时保存一份副本在本地；另一种是“推”的方式，Cache 会将预先指定的内容通过专门的方式传送到 CDN 网络中所有的或指定的 Cache 服务器上。在实际的 CDN 网络环境中，“拉”和“推”这两种方式一般会混合使用。每个节点上一般都会有多于一个 Cache 服务器用于提供不同的服务或提高每个节点的处理能力，例如，有的 Cache 服务器专门用来支持 HTTP 服务，有的专门支持 FTP 服务，有的则提供流媒体服务。因此，还需要一个本地负载均衡控制器来实现流量分配的功能。

4、CDN 管理系统：CDN 网络本质上是一个“内容的网络”，因此，对内容的管理是一个复杂的任务，无论采用“拉”还是“推”的方式，都需要预先做出判断，如哪些内容可以缓存、可以缓存多久或哪些内容根本不能保留在 Cache 服务器中而必须直接回原服务器等，这些都依赖于内容提供商提供的内容和目标客户群。对于 Internet CDN 网络来说，一个完善和稳定的内容管理系统更是网络成功的关键。

5、网络监控管理系统：每个 CDN 网络都需要一个网络监控和管理中心，这和任何一个系统或网络都需要网管中心一样，是保证整个系统正常运转的关键。由于 CDN 网络是一个分布式系统，如何实现集中监控是保证系统可靠、稳定运行的重要因素。网络监控系统必须能够实现分布检测和集中告警，能在第一时间内发现故障并定位故障。而且，由于 CDN 网络是一项内容服务，在监控网络连通的同时还要对应用层的各项服务进行监控。

2.4 CDN 的工作原理

当用户访问已经加入 CDN 服务的网站流媒体内容时，首先通过 DNS 重定向技术确定最接近用户的最佳 CDN 节点，同时将用户的请求指向该节点。当用户的请求到达指定节点时，CDN 的服务器(节点上的高速缓存)负责将用户请求的内容提供给用户，用户访问的基本流程如图 2-3 所示。

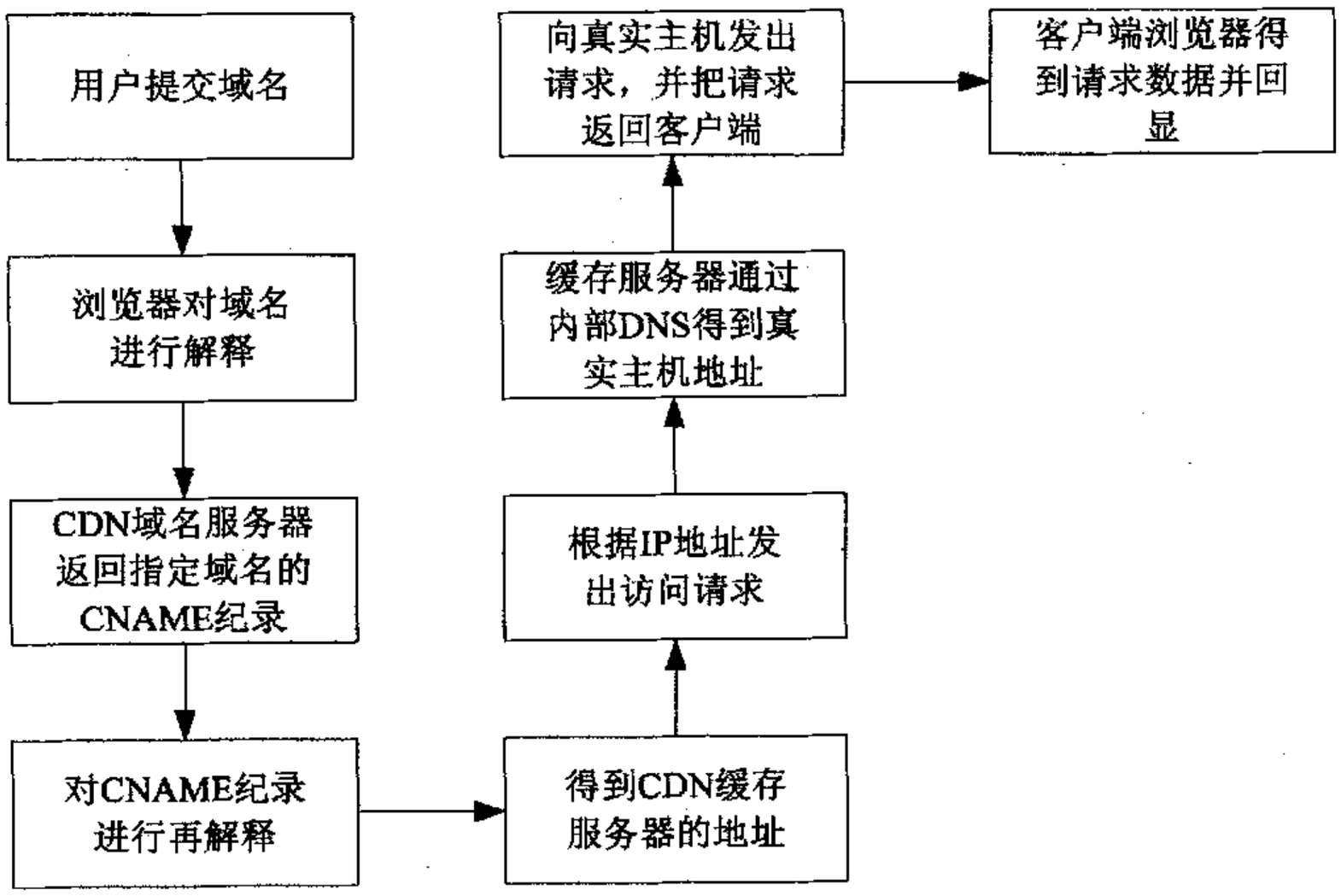

图 2-3 用户访问基本流程

Figure 2-3 Flow chart of user access

用户访问的基本流程如下：

1、用户在自己的浏览器中输入要访问的网站的域名；

2、浏览器向本地 DNS 请求对该域名的解析；

3、本地 DNS 将请求发到网站的主 DNS，主 DNS 再将域名解析请求转发到重定向 DNS；

4、重定向 DNS 根据一系列的策略确定当时最适当的 CDN 节点，并将解析的结果（IP 地址）发给用户。

5、用户向给定的 CDN 节点请求相应网站的内容；

6、CDN 节点中的服务器负责响应用户的请求，提供所需的内容。

用户通过以上的访问流程，在最佳的位置来访问各种内容，同时 CDN 的实现需要依赖多种网络技术的支持[28]：负载均衡技术、动态内容路由、高速缓存机制、动态内容分发与复制、安全服务等。

2.5 CDN 的主要技术

2.5.1 负载均衡技术

负载均衡技术，在网络环境下简而言之就是把网络负载尽量均匀地分配到几个能完成相同任务的服务器或网络节点上去执行和处理。由此来避免部分网络节点过载，而另一部分节点空闲的不利状况，改善网络性能，提高处理效率。

在 CDN 中，负载均衡又分为服务器负载均衡和服务器全局负载均衡[29]。

1、服务器负载均衡

由于网站内容、功能的日益增多，直接导致了支撑网站的服务器数量的不断增加。如何行之有效地根据各台服务器的处理能力动态地分配任务，已成为解决服务器访问速度问题的关键，这里所用到的技术，就是服务器负载均衡技术。利用这种技术，能够在性能不同的服务器之间进行任务分配，既保证性能差的服务器不成为系统的瓶颈，也能保证性能高的服务器资源得到充分利用，从而加快了 Web 服务器的访问速度。

2、服务器全局负载均衡

服务器全局负载平衡允许 Web 网络托管商、门户站点和企业根据地理位置分配内容和服务。分散内容和服务有许多好处，包括可以自动地将用户指引到位于其地理区域中的服务器，从而减少响应时间和对昂贵的国际数据连接的使用时间；引导用户离开拥挤的网络和服务器；通过使用多站点内容和服务来提高容错性和可用性，防止因本地网或区域网络中断、断电或自然灾害而导致的故障。

服务器全局负载均衡一般依照一些标准将用户的请求转到“最佳站点”，从而为其提供更好的服务。这些标准可以是站点的健康状况、站点距离、检索指定内容所需的响应时间等。

2.5.2 动态路由技术

当用户访问加入CDN服务的网站时，域名解析请求将最终由重定向DNS

负责处理。它通过一组预先定义好的策略(如内容类型、地理区域、网负载状况等)，将当时最接近用户的节点地址提供给用户，使用户可以得到快速的服务。同时，它还与分布在世界各地的所有 C D N C (Chinese Domain Name Consortium)节点保持通信，搜集各节点的健康状态，确保不将用户的请求分配到任何一个已经不可用的节点上。它还具有在网络拥塞和失效的情况下，自适应调整路由的能力[30]。

2.5.3 高速缓存机制

缓存服务通过几种方式来改善用户的响应时间。如代理缓存服务、透明代理缓存服务、使用重定向服务的透明代理缓存服务等[31]。

通过缓存服务，用户访问内容时可以将 WAN 的流量降至最低，降低对骨干网的压力。

2.5.4 动态内容分发与复制

内容网站访问响应速度取决于许多因素，如网络的带宽是否有瓶颈、传输途中的路由是否有阻塞和延迟、网站服务器的处理能力、访问距离等。多数情况下，网站响应速度与访问者与网站服务器之间的距离有密切的关系。尽管中国电信计划将骨干网络提速 8 倍，并且增加带宽，但是如果访问者和网站之间的距离太远的话，它们之间的通信一样需要经过重重的路由转发和处理，网络延误不可避免。一个有效的方法就是利用内容分发与复制机制，将占网站主体的大部分静态网页、图像和流媒体数据分发复制到各地的加速节点上。

内容分发网络可以采用智能路由和流量管理技术，及时发现与访问者最近的加速节点，并将访问者的请求转发到该加速节点，由该加速节点提供内容服务。利用内容分发与复制机制[32]，托管客户不需要改动原来的网站结构，只需修改少量的 DNS 配置[33]，就可以加速网络的响应速度。

2.6 CDN 中的代理服务器放置问题

在对于内容分发网络的研究中，目前有很多工作着重讨论了内容路由策略，即怎样高效的重定向用户的请求到离其最近的缓存上去以减小用户的访问延迟和服务器的负载均衡。固然，请求的重定向是内容分发网络的重要技

术，如果重定向策略不好，用户访问的命中率也就随之下降，内容分发网络的性能也就大打折扣。但是，服务器的放置策略对于提升内容分发网络的整体性能也具有不可忽视的作用[34]。

内容分发网络通过将网络服务内容分发推至网络边缘，达到均衡服务器负载，减小用户访问延迟的目的。由此可见，服务器的放置以及内容的复制策略对于内容分发网络的性能好坏是至关重要的。

2.7 本章小结

本章从各方面介绍了内容分发网络，其中包括概念，网络架构，组成，工作原理和主要技术，这其中特别对它的主要技术作了详细得阐述，分析了影响网络访问延迟的多种因素及 CDN 相应的解决方法。本章第六节中指出合理放置代理服务器为降低用户访问延迟、提高 CDN 整体性能的一个重要方面。

第3章 服务器放置策略的研究

内容分发网络的目标是减少用户访问延迟，均衡服务负载，降低网络访问成本。这可分为两个方面，其一为有效重定位用户请求到离用户最近的代理服务器上去，另外一个就是有效放置代理服务器。目前的大部分的工作都集中在有效重定位到代理服务器上，而很少关注 CDN 代理服务器的放置策略，合理的代理放置策略是非常重要的，因为它能减少客户的访问延迟和减少 ISP 的带宽消耗。

3.1 服务器放置问题描述

对于简单的服务器放置问题，也即网络中的节点放置问题，一般可以分为两类[35]：

1、K中心问题

2、一般设点问题

这两类问题都是NP-Hard问题，即在多项式时间内无解，目前对于这类问题也还没有得到精确解的办法。一般设点问题和K中心问题的根本区别在于：一般设点问题并不知道要设多少个点才是最优的，而且每个中心点的设置都需要一定的费用；而K中心问题事先已经知道要设置K个中心点，同时不考虑中心的设点费用。可以证明，K中心问题的下界可以由一般设点问题得到。

3.1.1 一般设点问题

CDN服务供应商要在互联网中设置一定数目的节点放置代理服务器，但并不知道放置多少个这样的代理服务器才是最好。每个节点的设置需要一定的费用。通常情况下，用户的访问延迟和用户到该节点距离有很大关系。我们的目标是如何放置代理服务器到合适的节点上，使用户的访问延迟最小。

该问题可以看作在二分图上求n条边的集合问题，即对集问题，每一条边表示代理服务器和用户的关系。因此，与这些边有关的参数：各条边的权值，对应于用户访问的延迟，与边关联的顶点的权值，对应于在该点设置的费用，满足边的权值和顶点的权值的和取最小的边集即为问题的解。

这是一个NP-hard问题，有许多近似算法提出，这些算法以常量P为参考，

在多项式时间内找到目标函数的一个近似解。目前为止，Guha 和 Khuller 给出的该问题的最好近似常量为1.728。

3.1.2 K 中心问题

CDN服务提供商要在M个节点中选择K (K<=M)个节点放置代理服务器。不考虑每个节点的设置费用。通常情况下，用户的访问延迟和用户到该节点距离有很大关系。对于被选中作为中心的节点，靠近此中心的所有节点都只能访问它。假如节点j被分配给中心i，所需的时间为d_{ij}，我们的目标是如何放置代理服务器到合适的节点上，使用户的访问总延迟$\sum d_{ij}$最小。

对于有效放置代理服务器，它本身就是一个NP-Hard问题，这本身就不能得到它的最优解，但可以得到它的近似解。它的主要目标是尽量使得所有节点的请求访问总代价最小。现在定义两个节点i和j的的延迟表示它们的请求代价。在已知节点个数和节点间的距离条件，服务器放置中心就可等价为节点的聚类中心，从而把服务器放置问题转化对网络节点进行聚类并求中心的问题。下一章将会详细介绍应用于服务器放置问题的改进的K-means聚类算法。

3.2 聚类算法在服务器放置问题上的引入

本文主要研究的是服务器放置的K中心问题，就是说给N个节点为中心，对于被选中作为中心的节点，靠近此中心的所有节点都只能访问它。假如节点 j 被分配给中心 i，所需花费的代价为 d(i,j)，找出一个最优的选法是，使网络的整体代价最小。在已知节点个数和节点间的距离条件，服务器放置中心就可等价为节点的聚类中心，从而把服务器放置问题转化对网络节点进行聚类并求中心的问题。

聚类分析需要一定的准则函数，才能把真正属于同一类的样本聚合成一个类型的子集，而把不同类的样本分离开来。如果聚类准则选得好，聚类质量就会高。同时，聚类准则函数还可以用来评价一种聚类结果的质量。如果聚类质量不满足要求，就要重复执行聚类过程，以便优化聚类结果。在重复优化中，可以改变相似性度量，若有必要还可选用新的准则函数。

在本文里，根据网络节点聚类的实际特点，借鉴了传统聚类分析里比较通用的误差平方和准则函数，给出了自己的准则函数：

$$J_c^{(j+1)} = \sum_{x \in \omega_j} D_L[X, M_j^{(l)}] \tag{3-1}$$

该准则函数适用于各类节点比较密集且节点数目悬殊不大的样本分布。所以在实际聚类中，我们得对各类的节点数进行一定的调整控制，让各类保证一定的节点数。

在实际的服务器放置问题上，服务器数目可能是固定的，也可能在一定范围内不固定，对此，我们分别采用不同的聚类算法解决这两个问题。当为固定的k个数目的服务器时，我们采用了动态聚类比较流行的K-均值聚类算法，并对起进行了改进；当服务器数目在一定范围内可变(<=C)时,我们采用了ISODATA算法，进行了聚类并寻找聚类中心。下一节简要介绍了几种聚类的方法。

3.3 聚类的主要方法

聚类的主要方法有：分割聚类方法、层次聚类方法、基于密度的聚类方法以及基于网格的聚类方法。

3.3.1 分割聚类方法

分割聚类方法(Partitioning Clustering Method)是发展比较早、也比较基本的一大类聚类分析算法。它是一种基于原型(Prototype)的聚类方法，其基本思路是：首先从数据集中随机地选择几个对象作为聚类的原型，然后将其他对象分别分配到由原型所代表的最相似也就是距离最近的类中。对于分割聚类方法，一般需要一种迭代控制策略，对原型不断地进行调整，从而使得整个聚类得到优化。

k-means是分割聚类方法中的一个代表算法，算法假设由n个对象需要分成k类，那么在k-means算法中，首先随机选择k个对象代表k个类，每一个对象作为一个类的中心,根据距离中心最近的原则将其他对象分配到各个类中。在完成首次对象的分配之后，以每一个类中所有对象的各属性均值(means)作为该类新的中心，进行对象的再分配，重复该过程直到没有变化为止，从而得到最终的k个类。

另外，k-means算法还有一种扩展算法，称为k-modes算法，该算法是将k-means算法的思想应用于分类变量的情况。

总体来讲，在聚类的形状为凸形，大小和密度相似，并且聚类的数目可

以合理估计的情况下，分割聚类算法比较有效，能够形成合理的聚类结果。

3.3.2 层次聚类方法

层次聚类方法(Hierarchical Clustering Method)也是发展比较早、应用也比较广泛的一大类聚类分析方法，它是采用“自顶向下(Top-Down)”或“自底向上(Botom-Up)”方法在不同的层次上对对象进行分组，形成一种树形的聚类结构。如果采用“自顶向下”的方法，则称为分解型层次聚类法(Divisive Hierarchical Clustering)：如果采用“自底向上”的方法，则称为聚结型层次聚类法(Agglomerative Hierarchical Clustering)。

层次聚类方法的基本思路：是按照一定的相似性判断标准，合并最相似的部分，或者分割最不相似的部分。如果合并最相似的部分，那么从每一个对象作为一个类开始，逐层向上进行聚结；如果分割最不相似的两个部分，那么从所有的对象归属在唯一的一个类中开始，逐层向下分解。

比较传统的层次聚类基本算法有AGNES (Agglomerative NESting)算法和DIANA(Divisive ANAlysis)算法，他们分别为聚结型层次聚类算法和分解型层次聚类算法。

近几年，出现了一些新的属于层次聚类方法的聚类算法，一般采用的是聚结型层次聚类策略，例如：BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)算法, CURE (Clustering Using Representatives)算法, ROCK算法和Chameleon算法。

3.3.3 基于密度的聚类方法

基于密度的聚类方法(Density-Based Clustering Method)以局部数据特征作为聚类的判断标准。类被看作是一个数据区域，在该区域内对象是密集的，对象稀疏的区域将各个类分割开来。多数基于密度的聚类算法形成的聚类形状可以是任意的，并且以各类中对象的分布可以是任意的。

基于密度的聚类方法研究是非常活跃的一个领域，近几年提出了许多新的算法，例如：1996年提出的DBSCAN (Density-Based Spatial Clustering of Applications withNoise)算法及其改进算法，1998年提出的WaveCluste算法，DENCLUE(DENsity-based CLUstEring)算法、CLIQUE (Clustering In QUEst)算法和1999年提出的OPTICS (Ordering Point To Identify the Clustering Structure)算法等。WaveCluster算法、DENCLUE算法和CLIQUE算法在基于

密度的同时，也是基于网格的。

3.4 算法比较

传统的层次聚类算法结果受各个类的大小和其中对象分布形状的影响，适用于类的大小相似且对象分布为球形的聚类。在对象分布形状比较特殊的情况下，可能会产生错误的聚类结果。另外，每一次类的聚结或分解都是不可逆的，并直接影响着下一步的聚结或分解。如果某一步的聚结或分解不理想，形成的聚类的质量就可能很低；基于密度的聚类方法适合拥塞的网络模型，对于节点密集的网络效果较好，不适合任意拓扑；在CDN服务器放置问题上，分割聚类方法能够给出最适合的放置服务器的位置，即各类的中心。另外，它简单高效的特点也适用于大规模的网络节点聚类。本文第四章和第五章分别详细介绍了k-means算法和ISODATA算法，并通过改进其初始点的选取来提高其性能。

3.5 本章小结

本章对服务器放置问题进行了描述，将该问题划分为一般设点问题和 K 中心问题两类，给出了解决这两类问题的常用的算法。然后用 ping 延迟作为节点之间的“距离”，将 CDN 服务器放置问题转化为一般聚类问题，通过求出聚类中心来得出放置 CDN 服务器的最佳节点。通过介绍聚类的主要方法，得出分割聚类中的 k-means 算法和 ISODATA 算法是解决 CDN 服务器放置的有效办法。

第4章 确定数目服务器放置算法

CDN 服务提供商要在互联网中设置 K 个节点放置代理服务器，一个简单的方法是将网络中的节点划分为 K 个子网，然后在每个子网的最优节点上放置一个代理服务器，该类中的其他节点均访问该服务器以获得最快的响应，并且整个网络中的总体时间代价为最小，本文采用了经典的 k-means 聚类算法对网络节点进行聚类并寻找中心节点。由于 k-means 聚类算法是局部最优的算法，它的最终结果会受初始点的选择的影响，为了优化聚类性能，本文对传统的 k-means 进行了改进，提出了基于迭代分裂的初始点选择方法。

4.1 k-means 算法简介

假设由N个对象需要分成K类，K-means算法首先随机选取K个点作为初始聚类中心，然后计算各个样本到聚类中心的距离，把样本归到离它最近的那个聚类中心所在的类；对调整后的新类计算新的聚类中心，如果相邻两次的聚类中心没有任何变化，说明样本调整结束，聚类准则函数 J_c 已经收敛。

K-means算法的一个特点是在每次迭代中都要考察每个样本的分类是否正确，若不正确，就要调整。在全部样本调整完后，再修改聚类中心，进入下一次迭代。如果在一次迭代算法中，所有的样本被正确分类，则不会有调整，聚类中心也不会有任何变化，这标志着 J_c 已经收敛，因此算法结束。

K-means算法具体描述如下：

(1)给出 N 个混合样本，从样本中选取 K 个初始聚类中心 $M_1^{(0)}$，$M_2^{(0)}$，…，$M_k^{(0)}$，令迭代次数 $l=0$ ；

(2)依据样本 $\{X_i, i=1,2,...,N\}$ 到聚类中心的距离，将其归入到中心为 $M_j^{(0)}$ 的类 ω_j 中： $x \in \omega_j$ ， $D_L(X,M_j^{(l)}) = \min\{D_L[X,M_j^{(l)}], i=1,\cdots, k\}$ ；

(3)重新调整聚类中心，得到 $M_j^{(l+1)}$，$M_j^{(l+1)}=\frac{1}{N_j^{(l)}}\sum_{x\in\omega_j}X$，其中j = 1，2，…，k；$N_j^{(l)}$是类$\omega_j$中的样本数。并计算准则函数的值 $J_c^{(j+1)}=\sum_{x\in\omega_j}D_L[X,M_j^{(l)}]$，其中j = 1，2，…，k。

(4)如果 $J_c^{(j+1)}=J_c^{(j)}$，则j=j+1，转步骤(2)，继续迭代，否则，停止。

从上面的算法框架可以看出，其特点为调整一个样本后就修改一次聚类中心和准则函数Jc值，当考察完n个样本后，一次迭代运算完成，新的聚类中心和Jc值也计算出来。如果在一次迭代前后，Jc值没有变化，说明算法已经收敛。在这个算法中，准则Jc作为算法是否结束的依据。在迭代过程中，Jc值逐渐减小，直到它的最小值为止。

k-means算法处理大数据集时，该算法是相对可伸缩和高效率的，它的计算复杂度是O(nkt)，其中，n是所有对象的数目，k是类的数目，t是迭代的次数。通常k < n且t<n，这个算法以局部最优结束。但是聚类的个数，k是必须预先指定的参数。

4.2 k-means 算法中初始中心选择对结果的影响

初始划分的产生一般有以下两种方法[37]：

(1) 随机选择k 个数据向量，以它们为类中心，依据某一原则进行初始划分；

(2) 随意地将数据向量划分到k个类中。

这两种方法存在很大盲目性，产生的划分质量比较差。而随机地选择 k 个数据对象用于代表类中心，存在所选的对象本应属于同一类的可能，而依据 k-means 算法的思想，这些本应属于同一类的对象却被硬性地划分到不同的类中去，必然会陷入局部极小。从上边的分析不难看出，k 个初始聚类中心点的选取对聚类结果具有较大的影响，因为在该算法中是随机的选取任意 k 个点作为初始聚类中心。如果有先验知识，可以选取具有代表性的点[38]。

同一数据源选取不同的初始类中心运行 k-means 算法结果如图 4-1 所示。

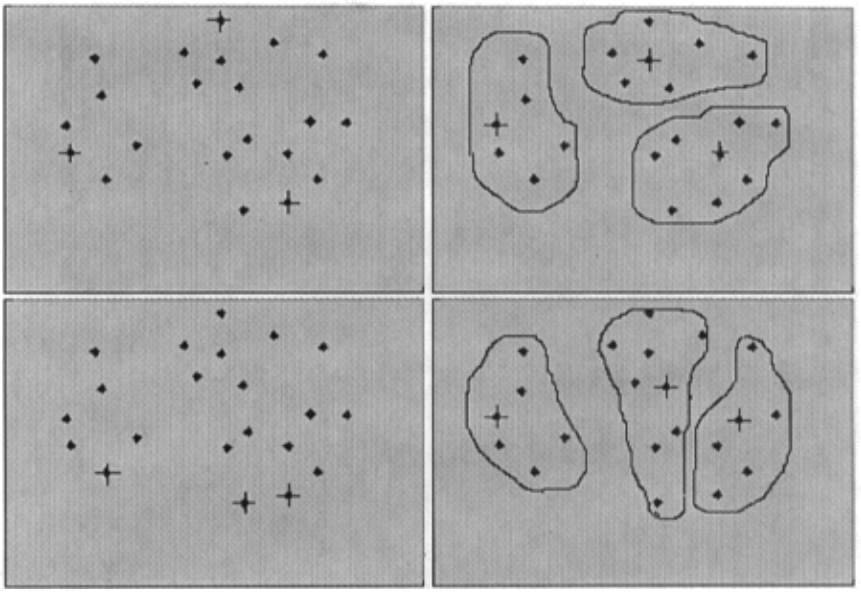

图4-1 同一数据源不同初始聚类中心的聚类结果

Figure 4-1 The clustering result of same data source with different initial partitions

从图4-1 可以看出，同一数据源选取不同的初始类中心运行k-means 算法，产生了不同的划分结果。通过比较可以看出第二次聚类的初始聚类中心选取的是本应属于同两个类的3 个数据向量，这是由于在迭代过程中不能脱离所属的类而陷入了局部极小。

另外，k-means算法对于孤立点和噪声也是敏感的。如果K个初始点中含有孤立点，那么在分配其他节点时，不会有任何节点被该孤立点吸引到这个类中，造成本应属于该类的一些节点被分配到其他类中，使得类之间成员个数差别很大，如果在这个很小的类中放置一台服务器的话，是非常浪费且不利于负载均衡的。

综上所述，初值的选择对聚类结果而言是非常重要的。因此，从随机选择的初始聚类中心开始进行聚类是很难得到一个稳定的聚类结果的。针对这个问题，本文对聚类中心的选取进行了改善，改善聚类算法中选择初值时候的依赖性，提高聚类结果的稳定性，并给出实验结果。

针对聚类初值的选择，在文献[39]中考虑了冗余类中心初始化方法，扩大解空间的搜索范围，减少某些极值点附近无初值的机会，初始聚类中心在数据空间中分布较广，具有多样性。采用适当准则逐步减小类的个数，直到

指定的K的数目，这样得到的聚类结果受随机选择的初始聚类中心的影响较小。初始的聚类中心选的越多，聚类结果受初值的影响就越小。但在这个算法中，需要确定一个合并参数d，即类间距小于d的类就进行合并。实际上，这个参数的确定是很难的，你无法确定一个较为合适的值，而这个参数的大小又直接影响着聚类结果。增加了聚类中心的同时，也增加了计算量，因此相应的聚类效率就大打折扣了。

4.3 选取 K 个初始聚类中心点的现有方法

在K-means算法中，都是随机的选取K个聚类中心。目前初始聚类中心的选择方法有以下的一些[40]：

1、任意的选取K个样本作为初始聚类中心。

2、凭经验选取有代表性的点作为起始聚类中心。根据个体性质，观察数据结构，选出比较合适的代表点。

3、把全部混合样本直观地分成K类，计算各类均值作为初始聚类中心。

4、通过“密度法”选择代表点作为初始聚类中心。所谓密度是指具有统计性质的样本密度。例如，以每个样本为中心，以某个给定正数d_1为半径，在特征空间里划出一个球形邻域，计算落入该邻域里的样本数目作为该点的密度。在计算完每个样本点的密度后，首先选取密度最大的样本作为第一个初始聚类中心，它对应着样本分布密度的最高峰值点；然后，给定一个正数d_2，在离开第一个初始聚类中心距d_2之外选择次大密度点作为第2个代表点，这样可以避免代表点过分集中；依此类推，可以选出K个初始聚类中心。

5、由K-1类聚类问题解出K类问题的代表点。例如：先把全部样本看成一个类，样本总均值点就是第1类的初始聚类中心；然后，由一类的初始聚类中心和离它最远的一个样本作为两类的初始聚类中心；依此类推，由(K-1)类的代表点和离他们最远的一个样本点作为K类问题的初始聚类中心。

6、按最大最小距离聚类法中寻找聚类中心的方法确定初始聚类中心。

7、进行多次初值选择、聚类、找出一组最优的聚类结果。

因此，初始聚类中心的选取方法是很多的，可以随机产生、凭经验知识获取、采用密度方法等等。无论聚类算法采用哪一种选取方法，我们都希望聚类中心越稳定越好、需要先验知识越少越好、需要确定的参数越少越好，而且希望算法能够产生一个较稳定的聚类结果，而不是对初始聚类中心非常敏感，不同的初始聚类中心产生不同的聚类结果。在传统的K-means算法中，

聚类结果对初始聚类中心有较强的依赖性，即不同的初始聚类中心会产生不同的聚类结果，因此聚类结果的有效性直接依赖于初始聚类中心的选择。

4.4 基于迭代分裂的初始点选取

4.4.1 基本思想

在这里，通过迭代分裂的思想寻找初始点。原始的k-means 算法选取k 个点作为聚类中心，依据这k 个点进行聚类。考虑到聚类的目的是目的是使得属于同一类别的个体之间的距离尽可能小，而不同类别上的个体间的距离尽可能的大，因此初始点的选择尽量使k个点比较分散，即k个点之间的距离较远。由此设计一种方法，先将所有样本都放在一个类中，并计算中心和类内距离，此时类的个数为1。然后选择类内距离最大的点进行分裂。分裂的方法是在类中找到距离最远的两个样本点，以这两个点为中心，按照其它样本点距离中心的距离将整个样本集合划分为两个类，类的个数加一。因为聚类的样本中可能会有孤立点，故判断这两个类中成员的个数是否少于设定的类成员个数阈值。如果小于设定的阈值的话，将这个类删除，并将类的个数减一，这样就排除了孤立点对聚类效果的影响。最后再按照类内距离远近找到类的新中心点，这样就完成了一次迭代。按照此方法逐步分裂，直到分裂为k个类，产生k个初始中心为止。

基于迭代分裂的初始点选取能够使k-means算法在更大范围内进行数据移动，充分考虑了k个中心的有效性和孤立点问题，将k-means算法对初始点选择的依赖降到最小。另外，通过调节类成员个数阈值，可以调节类的规模，这在CDN网络中放置代理服务器问题上是有重要意义的，因为为了少数几个用户提供一个服务器从成本来说是非常浪费的。

4.4.2 描述

图 4-2 描述了我们的计算初始聚类中心的过程。

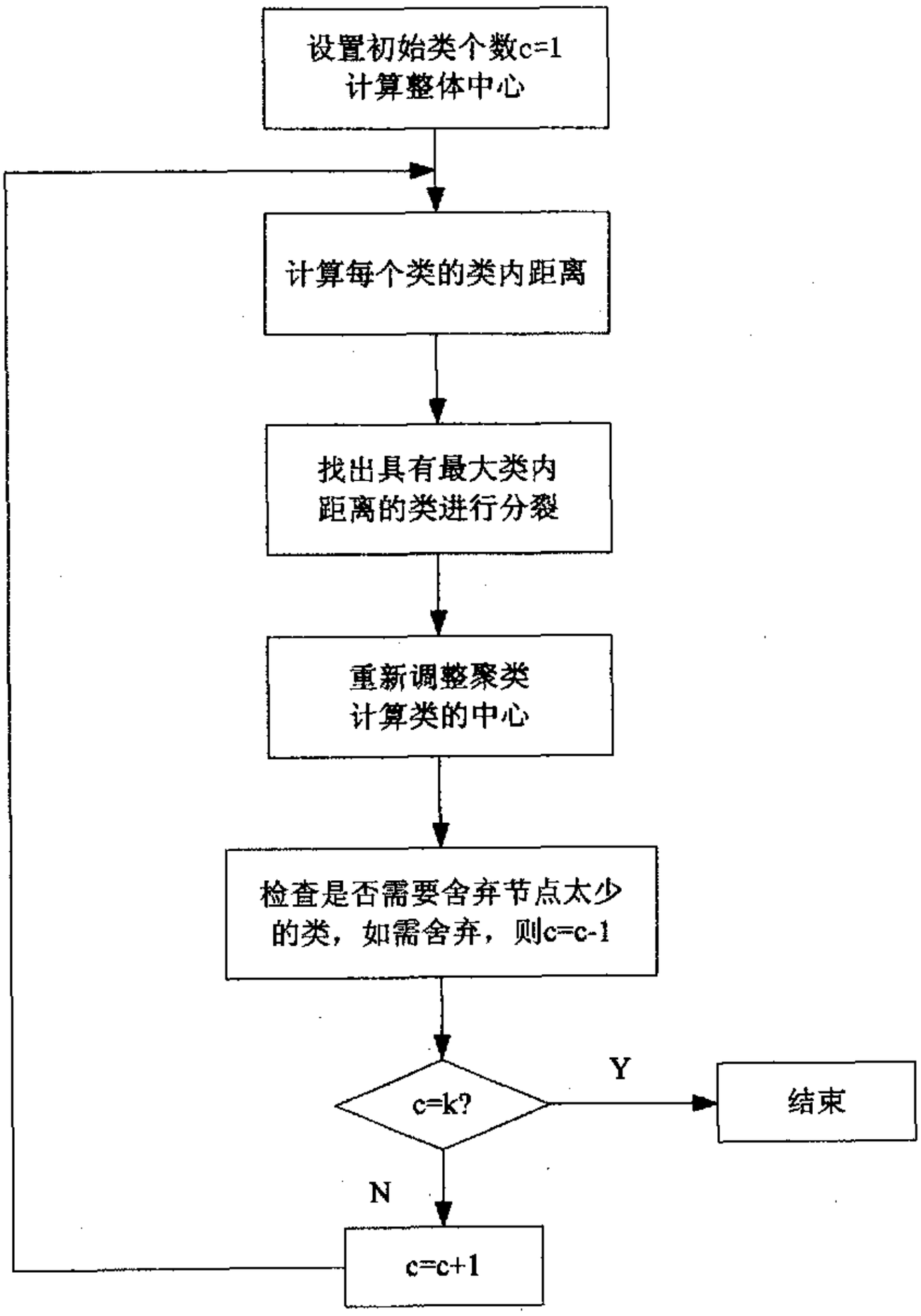

图 4-2 迭代分裂初始点选择流程图

Figure 4-2 Flow chart of selecting the preliminary node based on recursive separating

图中，我们用 c 表示当前迭代中类的个数，$\Gamma_j^{(l)}$表示第l次迭代时 c 个类集合的第 j 个类，$M_j^{(l)}$表示这个类的中心，$N_j^{(l)}$表示其中样本个数，$j=1,2,...,c$。这个算法的基本思想是：逐步分裂。即，c从 1 逐渐增加到 K。图 4-2 中的过程解释如下：

(1) 初始化：令 $c=1$，$\Gamma_1^{(l)}=\Pi$，$N_1^{(l)}=\|\Pi\|=N$，计算整个数据集 Π 的中心 $M_1^{(l)}$。

(2) 计算 $\Gamma_j^{(l)}$ 的总体类内距离 λ_i^{Σ}，对于 $i=1,2,...,c$，

$$\lambda_j^{\Sigma}=\sum_{X_i\in\Gamma_j^{(l)}} D_L[X_i,M_j^{(l)}] \tag{4-1}$$

(3) 找出具有最大总体类内距离的类分成两类。设这个类为 $\Gamma_j^{(l)}$，如果该类成员个数 $N_j^{(l)}\geq 2\times\theta_N$，那么把 $\Gamma_j^{(l)}$ 按照如下方法分裂：在该类中找出两个样本段数据 X_{p1} 和 X_{p2} 使得对任意的该类中的样本对 X_{p3} 和 X_{p4} 满足

$$D_L(X_{p1},X_{p2})\geq D_L(X_{p3},X_{p4}) \tag{4-2}$$

X_{p1} 和 X_{p2} 将作为新的两个聚类中心代替原来的中心 $M_j^{(l)}$，置 $c=c+1$。

(4) 将所有样本按照匹配距离最小原则分到各个类别中去。对所有的 $X_i\in\Pi$，若 $D_L(X_i,M_j^{(l)})<D_L(X_i,M_k^{(l)})$，$i=1,2,...,N$，$j,k=1,2,...,c$，$j\neq k$，则 $X_i\in\Gamma_j^{(l)}$，并记录每个类别中所归属的样本个数 $N_j^{(l)}$，$j=1,2,...,c$。

(5) 检查是否需要舍弃样本太少的类别。若有任何一个 $\Gamma_i^{(l)}$，$i=1,2,...,c$ 其基数 $N_i^{(l)}<\theta_N$，则舍去 $\Gamma_i^{(l)}$，并令 $c=c-1$。

(6) 重新调整聚类中心：重新计算 $\Gamma_j^{(l)}$ 的聚类中心 $M_j^{(l)}$，$j=1,2,...,c$。重复(2)~(6)步直至 c 达到 K 为止。

从上面的算法框架可以看出初始聚类中心不再是随机的选取，而是以一定的搜索准则进行多次搜索、选择。由于进行了多次类中心的选取，因此和原来算法比较起来更稳定了。

4.5 K-means 算法在 CDN 服务器放置问题上的应用

4.5.1 问题定义

Internet的拓扑结构用连通图 $G=(V,E)$ 表示，其中 $V=\{X_i,i=1,2,...,N\}$ 代表节点的集合，E代表网络链路的集合。对于链路 $(u,v)\in E$， $d(u,v)$ 表示节点u到节点v的延迟。CDN服务商需要在这N个Internet路由器节点上选择K个节点来放置代理服务器，这K个节点用集合 $C=\{M_i,i=1,2,...,K\}$，$C\subset V$ 表示。对于任意节点 $X_i\in V$， M_j 是集合C中与 X_i 延迟最小的服务器节点，即对C中的任意服务器节点 M_p， $1\le p\le K$， $p\ne j$，有 $d(X_i,M_j)<d(X_i,M_p)$ 令 $D_L[X_i,M_j]$ 表示从节点 X_i 到最近服务器 M_j 的延迟，我们的目的是找到这个集合C，使得网络中节点访问服务器的总延迟 $J_c=\sum_{X_i\in V} D_L[X_i,M_j]$ 最小。

4.5.2 算法描述

CDN的一个重要目标就是减少用户访问服务器的等待时间。我们使用从用户到服务器的延迟作为度量，一般情况下，节点会访问网络中距离自己最近的服务器。我们的目的是将网络中的若干节点分成若干子网，在每个子网中找到一个最佳位置放置一个CDN代理服务器，每个子网中的其他节点均就近访问这个CDN代理服务器以获得最佳响应。

确定服务器数目的CDN服务器放置问题可以用k-means算法很好的解决，并且出于成本考虑，可以采用改进的k-means算法适当调整每个子网的节点数目。

算法的完整描述为：

1、初始子网中心选择：

用 c 表示当前迭代中子网的个数， $\Gamma_j^{(l)}$ 表示第 l 次迭代时 c 个子网集合的

第 j 个子网，$M_j^{(l)}$ 表示这个子网的中心，即放置代理服务器的路由器，$N_j^{(l)}$ 表示子网中节点个数，$j=1,2,...,c$。

(1.1) 初始化：令 $c=1$，$\Gamma_1^{(1)}=V$，$N_1^{(1)}=\|V\|=N$，计算整个数据集 V 的中心 $M_1^{(1)}$。

(1.2) 计算子网 $\Gamma_j^{(l)}$ 中每个节点 X_i 到中心 $M_j^{(l)}$ 的延迟的总和 λ_i^{Σ}，对于 $i=1,2,...,c$，

$$\lambda_j^{\Sigma}=\sum_{X_i\in\Gamma_j^{(l)}} D_L[X_i,M_j^{(l)}] \tag{4-3}$$

(1.3) 将具有最大延迟总和且满足分裂条件的子网分成两个子网。设这个子网为 $\Gamma_j^{(l)}$，如果该子网节点个数 $N_j^{(l)}\geq 2\times\theta_N$，那么把 $\Gamma_j^{(l)}$ 按照如下方法分裂：

在该子网中找出两个节点数据 X_{p1} 和 X_{p2} 使得对任意的该子网中的节点对 X_{p3} 和 X_{p4} 满足

$$D_L(X_{p1},X_{p2})\geq D_L(X_{p3},X_{p4}) \tag{4-4}$$

X_{p1} 和 X_{p2} 将作为新的两个子网中心代替原来的中心 $M_j^{(l)}$，置 $c=c+1$。

(1.4) 将所有节点按照匹配距离最小原则分到各个子网中去。对所有的 $X_i\in V$，若 $D_L(X_i,M_j^{(l)})<D_L(X_i,M_k^{(l)})$，$i=1,2,...,N$，$j,k=1,2,...,c$，$j\neq k$，则 $X_i\in\Gamma_j^{(l)}$，并记录每个子网中所归属的节点个数 $N_j^{(l)}$，$j=1,2,...,c$。

(1.5) 检查是否需要舍弃样本太少的类别。若有任何一个 $\Gamma_i^{(l)}$，$i=1,2,...,c$ 其基数 $N_i^{(l)}<\theta_N$，则舍去 $\Gamma_i^{(l)}$，并令 $c=c-1$。

(1.6) 重新调整聚类中心：重新计算$\Gamma_j^{(l)}$的聚类中心$M_j^{(l)}$，$j=1,2,...,c$。重复(1.2)~(1.6)步直至c达到 K 为止。

至此，选出 K 个初始子网中心$M_1^{(0)}$，$M_2^{(0)}$，…，$M_k^{(0)}$，设置初始迭代次数$l=0$，然后进入 k-means 迭代。

2、依据节点$\{X_i, i=1,2,...,N\}$到子网中心的距离，将其归入到中心为$M_j^{(0)}$的子网ω_j中：$x\in\omega_j$，$D_L(X,M_j^{(l)})=\min\{D_L[X,M_j^{(l)}],i=1,\cdots,k\}$；

3、重新调整子网中心，得到$M_j^{(l+1)}$，$M_j^{(l+1)}=\frac{1}{N_j^{(l)}}\sum_{x\in\omega_j}X$，其中j = 1，2，…，k；$N_j^{(l)}$是子网$\omega_j$中的节点数。并计算准则函数的值$J_c^{(j+1)}=\sum_{j=1}^{K}\sum_{X\in\omega_j}D_L[X,M_j^{(l)}]$。

(4)如果$J_c^{(j+1)}=J_c^{(j)}$，则j=j+1，转步骤(2)，继续迭代，否则，停止。

D(N，N)表示一个N×N的矩阵，用来存放N个节点中任意两个节点之间的延迟，这个延迟是用ping来得到的。

4.5.3 实验模拟

nem[41](network manipulator)是法国ULPS大学(University of Louis Pasteur Strasbourg)于2002年开发的。其特点是具有通用性，并对拓扑图在最短路径长度和树结构方面的度量进行优化。nem实现了除早期3种算法与GLP之外的全部5种拓扑生成算法。nem将拓扑特征分为5类：树、距离、连通性、节点对最短路径数以及类别(class，包括叶上节点数、环上节点数等)，可对这5类特征进行分析。nem为网络模拟软件ns-2，GloMoSim和OPNet提供接口，以批处理文本文件作为用户接口。

本文通过 nem 生成一个 1000 个节点的随机拓扑模型，然后通过 NS2 网络模拟工具模拟出这个拓扑。NS[42]是由 Lawrece Berkeley National Laboratory

于 1989 年开发出的免费的网络模拟软件。NS 是一种可扩展、易配置和编程的事件驱动网络模拟工具，是从 S.Keshav's REAL 模拟器发展而来的。后来 NS 在 Virtual InterNetwork Testbed（VINT）项目的支持下由南加州大学、施乐公司、加州大学与 Lawrece Berkeley National Laboratory 协作发展 NS 软件，目前的版本是 NS2。

Nem产生拓扑结构分如下四个步骤：

1. 在平面上放置节点。
2. 连接节点。
3. 给每个节点设置属性，如连接的延迟，带宽，id等。
4. 以特定格式输出拓扑结构。

本文采用 nem 产生放置于网络中的节点，使用 waxman 的随机模型[43]连接节点构成边，即：

$$P(u,v)=\beta\exp\frac{-d(u,v)}{L\alpha} \tag{4-5}$$

其中，d(u，v)表示节点 u 与 v 之间的距离，L 为节点间最长距离，α与β取值范围是(0，1)。

在模拟中，我们假设如下条件成立：

1. 每个路由器节点均可以连接服务器。
2. 网络中任意两个节点是连通的，如果非连通，则设置这两个节点间的延迟为设置的最大值。
3. 每个服务器的处理能力能够满足子网中用户的访问需求。

模拟步骤如下：

1. 用nem生成1000个路由器节点，采用waxman模型随机放置节点，其中有1290条边，随机生成每条边上的带宽和延迟。
2. 编写tcl脚本，用ns2模拟这个网络拓扑。
3. 将每个节点绑定ping服务，设置起始时间和终止时间获得1000个节点之间的ping访问延迟，存储在一个文件中。
4. 用C++语言实现改进的k-means算法，读取文件中的ping访问延迟作为节点之间的距离，设定待放置的服务器数目k，然后运行程序，获得放置k个服务器的节点号和整体总延迟。
5. 随机选择初始点，采用k-means算法获得放置k个服务器的节点号和整体总延迟，与改进的k-means算法进行比较。

通过多次实验，我们得到如下结果。

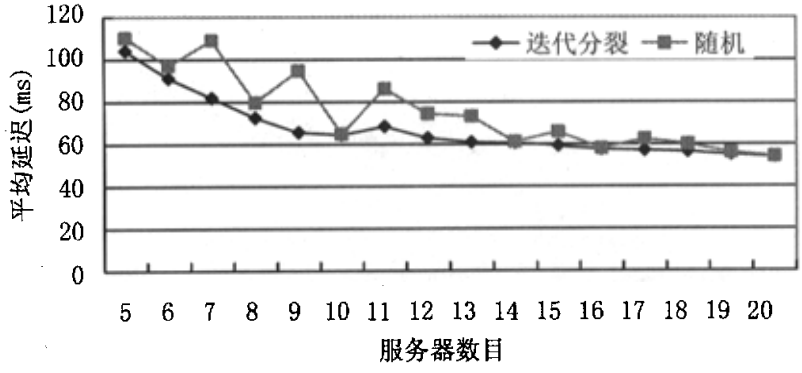

图 4-3 平均延迟对比图

Figure 4-3 Contrast of average delay

从实验结果可看出在使用迭代分裂初始点选择方法的 k-means 算法在网络中设置代理服务器节点后，用户的平均访问延迟得到改善，并且随着服务器数目的增大，用户的平均延迟平滑减少，这说明了基于迭代分裂的初始点选择的 k-means 算法具有良好的稳定性，更接近整体最优值。但我们同时也可以从图中看出，在服务器数量增加到一定程度的时候，平均延迟下降得不再明显，这说明此时再增加服务器数量来降低整体延迟已经不划算。

从图中看出在放置服务器个数较少时，随机选择初始点方法的曲线波动较大，说明随机方法的稳定性较差，很容易陷入局部极小而导致迭代停止。而服务器个数较多的时候，初始点选择对整体结果的影响已经不大，但是 k-mesns 算法得到的平均延迟依旧较小，说明得到的子网划分比较合理。

改进的 k-means 算法与随机选初试点的 k-means 算法聚类后每个类的节点个数如图所示，服务器数量是 10 个，横坐标表示第几个类，纵坐标表示类中节点数目。

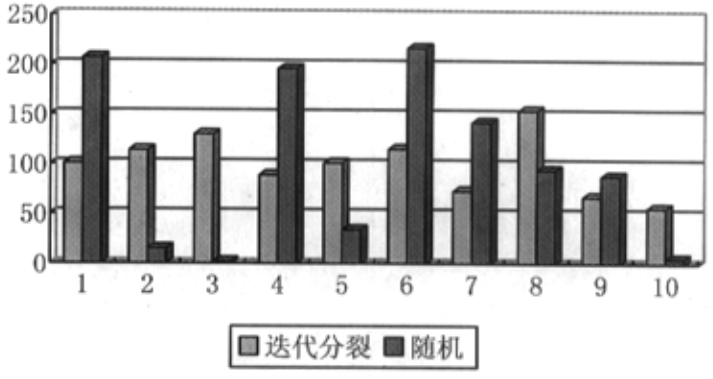

图 4-4 类间节点个数对比图

Figure 4-4 Contrast of node number in the clusters

出于成本考虑，CDN 服务供应商希望各服务器提供服务的节点数量比较接近，即每个类的规模大小接近。由于 k-means 算法对于孤立点是敏感的，如果初始点选择不好，很容易造成孤立点自成一类，而其他类节点很多的状况，这对于 CDN 的负载均衡是很不利的。从图中可以看出，随机选取初始点，聚类后各类节点个数波动较大，随机情况的第一个类中只有 2 个节点，如果在此类中放置一个服务器，将造成很大的浪费，而第 5 个类中节点多达 215 个，此类中的服务器将出于异常繁忙的状态，很难满足访问需求。用迭代分裂初始点选择方法的 k-means 算法由于对于较小的类进行了处理，所以 10 个类的节点个数较为平均，可以均衡网络负载，充分利用服务器资源。

在采用相同的服务器数目时，对比随机选择初始点的 k-means 算法，采用本文改进的 k-means，系统的平均延迟得到了降低。从而证明改进的 k-means 算法在实践上是可行的，基于迭代分裂的初始点选择方法是有效的。

4.6 本章小结

本章针对经典的 k-means 算法中初始点选择所面临的问题，提出了基于迭代分裂的初始点选择算法。在 NS 网络仿真条件下，对 CDN 的服务器放置问题进行了模拟，并将改进的 k-means 算法与随机选择初始点的 k-means 算法进行了比较，获得了较好的效果。

第5章 不确定数目服务器放置算法

5.1 引言

在 k-means 算法中 K 是事先给定的，这个 K 值的选定是很难估计的。很多时候，我们事先并不知道给定的数据集应该分成多少个类别才最合适。这也是 k-means 算法的一个不足。在实际的服务器放置问题里，事先确定服务器数目，往往因为和网络情况不相匹配，从而达不到最优性能。因此，可以在一定的服务器数目上限下，根据网络本身特点，进行动态聚类，根据聚类个数确定服务器数目，能更好的优化整个系统。在这里我们采用了 ISODATA 算法，作为网络节点的聚类算法。

5.2 迭代分裂初始点选择的 ISODATA 算法

5.2.1 基本思想

k-均值算法比较简单，但它的自我调整能力也比较差。这主要表现在类别数不能改变，受代表点初始选择的影响也比较大。ISODATA 算法的功能与 k-均值算法相比，在下列几方面有改进。

1. 考虑了类别的合并与分裂，因而有了自我调整类别数的能力。合并主要发生在某一类内样本个数太少的情况，或两类聚类中心之间距离太小的情况。为此设有最小类内样本数限制θ_n，以及类间中心距离参数θ_c。若出现两类聚类中心距离小于θ_c的情况，可考虑将此两类合并。

分裂则主要发生在某一类别的某分量出现类内方差过大的现象，因而宜分裂成两个类别，以维持合理的类内方差。给出一个对类内距离的限制参数θ_s，用以决定是否需要将某一类分裂成两类。

2. 由于算法有自我调整的能力，因而需要设置若干个控制用参数，如聚类数期望值 C、允许迭代次数$t_{\max}$等。

这节用第四章中描述的迭代分裂的初始点选择方法与 ISODATA 算法相结合，代替原来的随机选择初始点的方法，使得算法的结果对初始点依赖更小。

5.2.2 算法描述

整个聚类算法过程的具体描述如下，总体的流程图如图 5-1 所示：

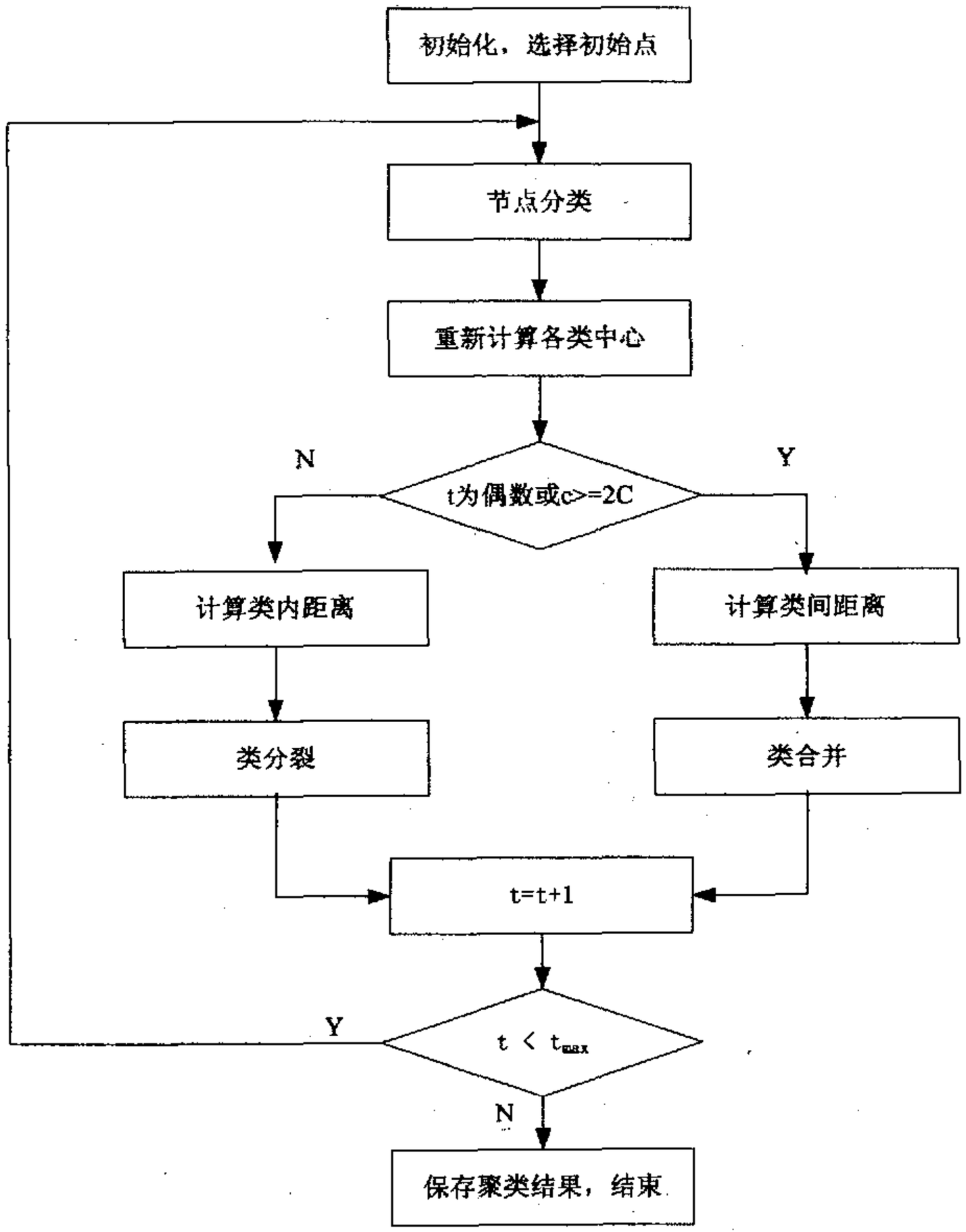

图 5-1 ISODATA 算法流程图

Figure 5-1 Flow chart of ISODATA algorithm

1. 初始化

设 $V=\{X_i,i=1,2,...,N\}$ 是全部节点集合，每两个节点之间的距离仍旧使用二者之间的延迟，设为 $d(X_i,X_j)$，从而形成一个距离矩阵 $D_{V\times V}$，后面在使用时可以直接用查表的方法得到距离。设 c 表示类的个数，C 为期望得到的聚类数。θ_C 表示进行类合并的类间距离阈值。θ_D 表示进行类分裂的类内平均距离阈值。θ_N 表示每个类中的最少样本数。t 表示迭代次数，$t_{\max}$ 为最大迭代的次数。

2. 初始化聚类中心

算法以上章描述的迭代分裂初始点选择算法作为计算初始聚类中心。该算法的基本思想是：对训练样本逐步分裂，一直达到所设定的类别数目或者达到设置的循环最大值。

3. 样本分类

将网络节点按照距离最小原则分到各个类别中，并记录每个类别的节点个数。对任意 $X_i\in V$，若 $j=\arg\min_p d(X_i,M_p)$，$p=1,2,...,c$，则 $X_i\in\Gamma_j$。其中 M_j 表示类 Γ_j 的中心。同时检查是否舍弃样本太少的类，若有任何类的样本个数小于 θ_N，则舍去该类，令 $c=c-1$，并将该类样本重新分到新的类别中。

4. 重新计算聚类中心

重新计算每个类的聚类中心 M_j，$j=1,2,...,c$。聚类中心求法如下，先以类内任意一点为伪中心，求出相对于该伪中心的总类内距离，总类内距离最小所对应的伪中心就是实际要求的中心。

5. 若这是偶数次迭代或者 $c\geq 2C$，则转向步骤 8 进行合并；否则继续。

6. 计算类内距离

计算 Γ_j 的总体类内距离 λ_j^{Σ} 和平均类内距离 $\overline{\lambda_j}$，$\lambda_j^{\Sigma}=\sum_{X_i\in\Gamma_j^{(l)}} D_L[X_i,M_j^{(l)}]$，

$\overline{\lambda_j} = \lambda_j^{\Sigma} / \|\Gamma_j\|$，$j = 1,2,..,c$。本文主要根据网络聚类实际情况，采用了平均类内距离 $\overline{\lambda_j}$。

7. 类分裂

将具有最大类内距离的类分裂成两类。最大类内距离可以有两种选择：总体类内距离 λ_j^{Σ} 和平均类内距离 $\overline{\lambda_j}$，这里选择总体类内距离 λ_j^{Σ}。设选择的类为 $\Gamma_{j\max}$，如果 $\|\Gamma_{j\max}\| \geq 2\theta_N$ 并且 $\lambda_j^{\Sigma} > \theta_D$，或者 $c \leq C/2$，$\Gamma_{j\max}$ 按照如下方法分裂：在该子网中找出两个节点数据 X_{p1} 和 X_{p2} 使得对任意的该子网中的节点对 X_{p3} 和 X_{p4} 满足

$$D_L(X_{p1}, X_{p2}) \geq D_L(X_{p3}, X_{p4}) \tag{5-1}$$

X_{p1} 和 X_{p2} 将作为新的两个子网中心代替原来的中心 $M_j^{(l)}$，置 $c = c+1$，转向步骤 10。

8. 计算类间距离

利用延迟计算所有的聚类中心两两之间距离：$d(M_i, M_j)$，$1 \leq i, j \leq c$。

9. 类合并

找出所有 $d(M_i, M_j)$ 中的最小值 $d(M_p, M_q)$，如果 $d(M_p, M_q) < \theta_C$，那么将类 Γ_p 和类 Γ_q 合并，新的聚类中心 $\overline{m}$ 可用合并的二类用步骤 4 所提到的方法求出。

10. $t = t+1$，若 $t < t_{\max}$，则转向步骤 3；否则，保存聚类有关数据：聚类数目 c，聚类中心以及与这个聚类中心最近的 T_N 个样本，结束。

5.3 实验模拟

本章的实验采用第四章的NS网络模拟环境，在1000个节点的随机网络中，用C++语言实现迭代选择初始点的ISODATA算法，读取文件中的ping访问延迟作为节点之间的距离，设置期望得到的聚类数C，进行类合并的类间距离阈值θ_C，进行类分裂的类内平均距离阈值θ_D，每个类中的最少样本数θ_N和最大迭代的次数$t_{\max}$，然后运行程序，根据经验适当调整上述参数，获得实际放置的服务器数目c和相应的节点号以及整体访问平均延迟。

为了将改进的 k-means 算法与 ISODATA 算法相比较，先运行改进的 k-means 算法，得到结果如图 5-2 所示。

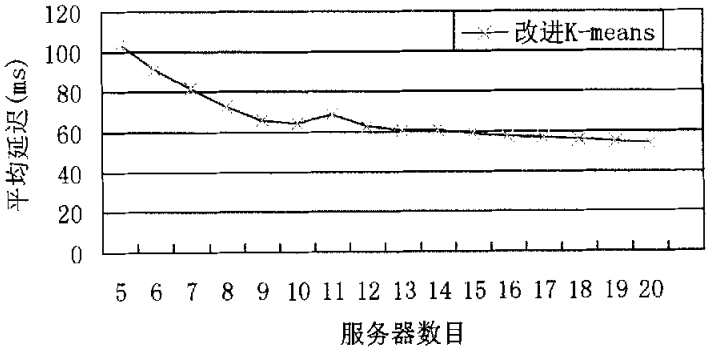

图 5-2 改进的 k-means 算法结果图

Figure 5-2 Result of improved k-means algorithm

从图 5-2 可看出在使用改进的 k-means 算法在网络中设置代理服务器节点后，用户的平均访问延迟得到改善，并且随着服务器数目的增大，用户的平均延迟呈下降趋势，但下降的幅度不一样，所以存在拐点，拐点对应的服务器数目应该是比较合理的。 ISODATA 聚类的一大特点就是聚出合理的服务器数目，就是找出所谓的拐点。

ISODATA 的聚类数情况如图 5-3。

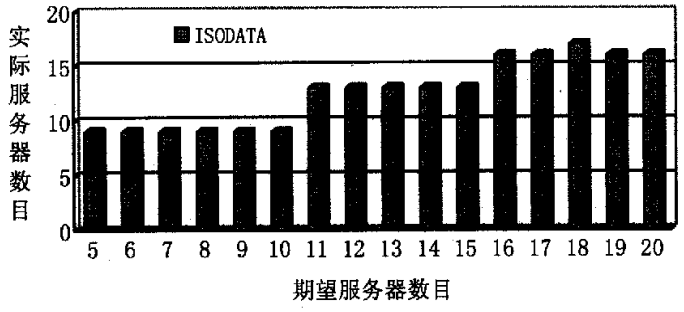

图 5-3 ISODATA 算法实验结果图

Figure 5-3 Results of ISODATA algorithm experiment

由图可知，当期望服务器数 C 从 5 到 20，实际聚类得到服务器数目集中在 9，13，16 三个值上。由改进的上图可知当服务器数在 9，13，16 上时，平均延迟下降明显，选择这样的服务器数目显然比较合理。

当聚类服务器数目分别为 9，13，16 时，改进的 k-means 和 ISODATA 算法得到的平均延迟比较如图（ISODATA 的平均延迟是均值）。

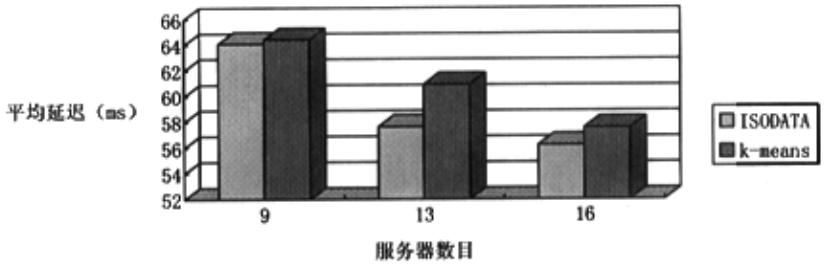

图 5-4 ISODATA 与改进的 k-means 实验结果对比图

Figure 5-4 Contrast between results of ISODATA and improved k-means algorithm

图 5-4 可以看出，当聚出的服务器数目相同时，ISODATA 算法的平均延迟比改进的 k-means 还要好，这说明 ISODATA 算法通过合并和分裂方法，对节点重组的幅度更大，从而使整体性能受初始点选择影响更小，更能接近全局最优。

综上所述，当服务器数目在一定范围内不确定的条件下，基于迭代分裂初始点选择的 ISODATA 算法能确定合理的服务器数目，并找出最佳的位置。

5.4 本章小结

本章主要讨论了不确定服务器数目下的服务器放置问题，将基于迭代分裂初始点选择方法与 ISODATA 算法相结合，得到了合理的代理服务器数目，并找出最佳放置代理服务器的位置，在优化网络结构，降低用户访问延迟方面达到了很好的效果。

结　论

本文在介绍了内容分发网络的概念、网络架构和组成以及工作原理和主要技术之后，针对内容分发网络的代理服务器放置策略进行了研究，分别在代理服务器数量确定和不确定的条件下，将基于迭代的初始点选择方法与 k-means 算法和 ISODATA 算法相结合，在 NS 网络模拟条件下，对 CDN 的服务器放置问题进行了模拟实现，获得了很好的结果。得出的结论如下：

1、基于迭代分裂的初始点选择方法能够降低 k-means 算法的聚类结果对初始点的依赖，避免了算法陷入局部极小而不能收敛于全局最优。并且，由于加入了类成员个数限制，有效降低了类之间成员数量差别，对全局负载均衡和降低成本有重要作用。

2、ISODATA 算法能够根据网络状况动态的对网络中节点进行聚类，找出合理的放置服务器的数目和最佳放置节点，具有效率高，所需信息量较少的优点。通过将迭代分裂初始点选择方法与 ISODATA 算法相结合，使算法对节点重组的幅度更大，从而使整体性能受初始点选择影响更小，更能接近全局最优。

3、通过优化代理服务器放置位置，能够有效地降低用户的访问延迟，提高 CDN 的整体性能，对于合理规划网络结构和调整信息流动有着重要意义。

4、在得到代理服务器放置位置的同时，还将访问节点进行了聚类，在某个节点访问该网站时，可以通过直接查找该节点所属的类来直接定位距离该节点“最近”的代理服务器，即该类的中心。

尽管本文在 CDN 代理服务器放置策略的研究中取得了一些成果，但仍需在如下方面做进一步工作：

1、作为聚类依据的“距离”可以考虑更多的因素，如带宽，费用和用户对内容的访问频率等。

2、代理服务器放置在考虑用户访问延迟的同时，还可以考虑从原始服务器向代理服务器分发内容的延迟，以更好地优化内容路由。

参考文献

1 毕经平，吴起，李忠诚, Internet 延迟瓶颈的测量与分析. 计算机学报,2003, 26(4):406-416.

2 Singh M P, Peering at Peer-to-Peer Computing. IEEE Internet Computing, 2001-01-02.

3 E.Cronin, S.Jamin, Cheng Jin,"Constarined Mirror Placement on the Internet", IEEE 2002.2.

4 Kirk L Johnson, John F Carr, Mark S Dayetal. The measured performance of content distribution networks [C]. 5th International Web Caching and Content Delivery Work shop, May 2000.

5 Day, B.Cain, G.Tomlinson and P.Rzewski, A model for content Internet working (CDI), IETF Internet Draft.

6 Krishnamurthy B, Wills CE, Zhang Y .On the use and performance of content distribution networks. In P roc. Of SIGCOMMIMW 2001, Nov. 2001.

7 Syam Gadde, Jeff Chase, Michael Rabinovich. Web caching and content distribution:A view from the interior. Computer Communications, 24(2), February 2001.

8 Lili Qiu,V.N. Padmanabhan and G.M.Voelker, On the Placement of Web Server Replicas, IEEE 2001.1.

9 S. Jamin, C. Jin, D. Raz, Y. Shavitt, and L. Zhang, On the placement of internet instrumentation, in Proc. IEEE INFOCOM, Mar. 2000.

10 M. Charikar, S.Guha, A constant-factor approximation algorithm for the k-median problem, Proceedings of the 31st Annual ACM Symposium on Theory of Computing (1999).

11 Bezdek JC. A convergence theorem for the fuzzy ISODATA clustering algorithm. IEEE Trans Pattern Anal Mach Intell. 1980; PAMI-2:1-8.

12 Duane Wessels, K. Clafy, the Squid Web Cache. IEEE Journal on Selected Areas in Communications. 1998, 16(3).

13 Feldmann Anja, Gilbert Anna C, Huang Polly. A study of the role of variability and the impact of control. Computer Communication Review. 1999, 29(4): 24-36.

14 Feldmann Anja, Gilbert Anna C, Huang Polly. A study of the role of variability and the impact of control. Computer Communication Review. 1999, 29(4): 24-36.

15 Goldszmidt, G.Hunt. Scaling Internet services by dynamic allocation of connections. Proc. of the 6th IFIP/IEEE Integrated Management, Boston MA. May 2002.

16 V.Cardellini, M.Colajanni. Redirection algorithms for load sharing in distributed Web server systems. IEEE Computer Soc. Press. May.2001.

17 R. Ravi, A. Sinha, Multicommodity facility location, Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms, January 11-14, 2004.

18 David B. Shmoys, Eva Tardos , Karen Aardal, Approximation algorithms for facility location problems, Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, p.265-274, May 04-06, 1997, El Paso, Texas, United States.

19 Moses Charikar, Sudipto Guha, David B. Shmoys. A constant-factor approximation algorithm for the k-median problem. Journal of Computer and System Sciences. v.65 n.1, p.129-149, August 2002.

20 Adam Meyerson, Serge Plotkin, A k-Median Algorithm with Running Time Independent of Data Size, Machine Learning, v.56 n.1-3, p.61-87.

21 Craig W. Cameron, Steven H. Low, David X. Wei. High-Density Model for Server Allocation and Placement, Joint International Conference on Measurement and Modeling of Computer Systems, 2002 p.152-159.

22 Moses Charikar, Chandra Chekuri, Ashish Goel, Sudipto Guha. Rounding via Trees: Deterministic Approximation Algorithms for Group Steiner Trees and k-median, In Proc. of IEEE Infocom, volume 2, pages 670-678, March 2000.

23 Lisa Amini and Henning Schulzrinne. Client Clustering for Traffic and Location Estimation. IEEE International Conference on Distributed Computing Systems (ICDCS), March 2004.

24 熊大红，黄传河，贾小华，肖磊，蔡莉，李桓，CDN 中的代理服务器放置算法. 计算机应用研究. 2004 Vol.21 No.5 P.250-251.

25 王娟，赵问道，基于网络拓扑的 CDN 内容路由技术研究. 浙江大学学报（工学版）. 2004 Vol.38 No.4 P.414-419.

26 Content Delivery Networks:An Introduction (white paper), Available on-line at htp://cdn.hcltech.com.

27 J. Kangasharju, J. Roberts, and K. W. Ross. Object Replication Strategies in Content Distribution Networks. In Proceedings of WCW'01: Web Caching and Content Distribution Workshop, Boston, MA, June 2001.

28 A. Shaikh, R. Tewari, and M. Agrawal. On the effectiveness of DNS-based server selection. Proceedings of the IEEE Infocom 2001 Conference, (Anchorage, Alaska USA), Apr. 2001.

29 Anees Shaikh, Renu Tewari, Mukesh Agrawal. On the Effectiveness of DNS-based Server Selection. In Proceedings of IEEE INFOCOM'2001 [C]. Anchorage,Alaska: IEEE, 2001:1801-1810.

30 Hirokazu M, Miki Y. Content routing with network support using passive measurement in content distribution networks [A]. The 11th International Conference on Computer Communications and Networks [C]. Chicago, I L: IEEE, 2002: 96-101.

31 Gadde, J. Chase, M. Rabinovich. Web caching and content distribution: A view from the interior. In Proc. of the 5th Int. Web Caching and Content Delivery Workshop. May 2000

32 J. Kangasharju, J. Roberts and K.W. Ross. "Object Replication Strategies in Content Distribution Networks", Computer Communications, Vol.25 (4), pp. 376-383, March 2002.

33 Cardellini, M.Calajanni. DNS dispatching algorithms with state estimators for scalable Web server clusters. World Wide Web J. Baltzer Science, Bussum, Netherlands. 1999, 2(2)

34 X.D.Hu, Optimal. Placement of Web Proxies for Replicated Web Servers in the Internet. The computer Journal, Vol.44, No.5, p329-339

35 B.Li, M.J.Golin, F.Italiano, X. Deng, K.Sohraby, On the optimal placement of Web proxies in the Internet, Proc. IEEE INFOCOM, 1999, pp.1282-1290.

36 Lili Qiu, V.N, Padmanabhan and UM.Voelker. On the Placement of Web Server Replicas. IEEE 2001.1.

37 史忠植. 知识发现. 北京:清华大学出版社,2002. 356 - 368.

38 D. Pelleg. A. Moore. Accelerating Exact K-means Algorithms with Geometric Reasoning. January 2000, CMU-CS-00-105.

39 徐燕，单波，王颖. 对一种矢量量化聚类算法的改进及应用. 华北电力大学学报. 2001, 28(3): 62-65

40 边肇棋，张学工. 第二版.模式识别聚类技术.北京:清华大学出版社,1999:235-247.

41 Kevin Fall. NS Manual[R]. LBL, UC Berkeley, USC/ISI, 2002.

42 Magoni D. nem: A software for network topology analysis and modeling. In: Proc. of the MASCOTS 2002. IEEE Computer Society, 2002. 364 -371.

43 Waxman BM. Routing of multipoint connections. IEEE Journal on Selected Areas in Communications, 1988, 6(9):1617-1622.

攻读学位期间发表的学术论文

1　韩笑，李斌，田志宏. 一种大规模网络环境下高可扩展的信息监测系统. 计算机工程. 录用待发表于 2005.10

哈尔滨工业大学硕士学位论文原创性声明

本人郑重声明：此处所提交的硕士学位论文《基于 CDN 的服务器放置策略研究》，是本人在导师指导下，在哈尔滨工业大学攻读硕士学位期间独立进行研究工作所取得的成果。据本人所知，论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

作者签字 [illegible]　　日期：2005 年 6月26日

哈尔滨工业大学硕士学位论文使用授权书

《基于 CDN 的服务器放置策略研究》系本人在哈尔滨工业大学攻读硕士学位期间在导师指导下完成的硕士学位论文。本论文的研究成果归哈尔滨工业大学所有，本论文的研究内容不得以其它单位的名义发表。本人完全了解哈尔滨工业大学关于保存、使用学位论文的规定，同意学校保留并向有关部门送交论文的复印件和电子版本，允许论文被查阅和借阅。本人授权哈尔滨工业大学，可以采用影印、缩印或其他复制手段保存论文，可以公布论文的全部或部分内容。

保密□，在　　年解密后适用本授权书。

本学位论文属于

不保密☑。

（请在以上相应方框内打“√”）

作者签名：[illegible]　　日期：2005 年 6 月26日

导师签名：[illegible]　　日期：06年6月27日

致　谢

在论文完成之际，谨向给予我无私帮助的老师和同学们致以诚挚的谢意！

首先向我的导师李斌副教授表示深深的感谢。感谢李老师的关心、指导和教诲。在学习的道路上李老师不断给我鼓励，帮我把握前进的方向，在我遇到困难、挫折而感到迷茫的时候热心的帮助我、支持我，教给我许多做人的道理，使我终生难忘。

感谢胡铭曾教授在生活、学习、科研等方面的关怀和指导。胡老师开阔的视野、渊博的知识、敏锐的思维、民主而严谨的作风使学生受益匪浅，并终生难忘。

感谢张宏莉教授在进行课题工作中所给予的悉心帮助和耐心指导，张老师严谨的学风、严以律已、宽以待人的崇高品质对学生将是永远的鞭策。

感谢方滨兴教授，他在学术上和工作中孜孜不倦、锐意进取。方老师的敬业精神值得我们每一个人学习。

感谢云晓春教授，云老师能够始终把握所研究域的技术前沿，具有敏锐的洞察力和理解力，他的钻研精神令我十分敬佩。

感谢我的师兄王垚博士，他对我的毕业设计始终给以密切的关注和具体、直接的指导，提出了许多建设性的建议。

感谢何慧老师，包秀国博士，田志宏老师、张伟哲老师，王东滨老师，杜阿宁博士、时金桥博士在进行课题工作中所给予的悉心指导和无私帮助。

感谢刘妍、王威、刘乙璇、关晓巍、王蕾、张永、王福亮、刘鹏、沈志杰、李少敏同学在进行课题工作和生活中对我无微不至的关怀和帮助。

感谢 PACT 实验室，PACT 团结进取的奋斗精神，是我一生的精神财富。

感谢我的家人多年来对我的支持和帮助，是他们的无私的爱让我完成学业。

再次感谢所有关心、支持和帮助过我的人。