云服务传递网络资源动态分配模型

史佩昌[1)] 王怀民[1)] 尹 刚[1)] 刘雪宁[2)] 袁小群[3)] 史殿习[1)]

[1)](国防科技大学计算机学院并行与分布处理国家重点实验室 长沙 410073)

[2)](清华大学计算机科学与技术系 北京 100084)

[3)](华中科技大学电子与信息工程系 武汉 430074)

摘 要 云服务传递网络(Cloud Services Delivery Networks,CSDN)在 Internet 之上构建了一层分布式服务器网络,以就近和按需的方式向用户提供云传递服务.面对互联网规模化和多样化云服务的资源需求特点,CSDN 形成了针对不同类型云服务传递的逻辑子服务器网络.CSDN 的很大一部分服务器和带宽资源用于流媒体和下载类云服务的传递,该类型云服务传递资源的动态分配问题是该文的研究重点.根据该类型业务内存资源和带宽资源同为瓶颈资源以及该类型热点内容可采用 P2P 机制的两个特点,文中首先将该问题建模为多维设备选址模型.然后在对该建模分析及其 NP 完全性证明后,提出了一种启发式模型求解算法.最后以服务传递开销节省作为性能评价指标,以实际系统的运行数据为输入,全面评估了该模型求解算法的有效性.

关键词 云服务;传递网络;协同;对等网络;动态分配

中图法分类号 TP311 **DOI 号**: 10.3724/SP.J.1016.2011.02305

The Dynamic Allocation Model for the Resources of Cloud Services Delivery Networks

SHI Pei-Chang[1)] WANG Huai-Min[1)] YIN Gang[1)] LIU Xue-Ning[2)] YUAN Xiao-Qun[3)] SHI Dian-Xi[1)]

[1)](*National Laboratory for Parallel and Distributed Processing, School of Computer and Science, National University of Defense Technology, Changsha* 410073)

[2)](*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

[3)](*Department of Electronics and Information Engineering, Huazhong University of Science Technology, Wuhan* 430074)

Abstract Cloud Services Delivery Networks (CSDN) constructs a layer distributed server overlay over the Internet, which uses the way to the nearest and on-demand approach providing services to end users. Facing the scale and diversification of the resource demand characteristics of the Internet cloud services, CSDN forms different logical sub-server overlay for different kinds of cloud services. However, most servers and bandwidth resources of CSDN are used to deliver the streaming and downloading kind of cloud services, and the dynamic allocation of their delivery resource is the main research emphasis in this paper. This paper first models the problem to be a multi-dimensional facility location problem, according to the two characteristics: the memory resource and bandwidth resource of this kind of application are the bottleneck resource; the hot contents of this kind of application can be delivered using the Peer-to-Peer mechanisms. After the model analyzed and its NP-Complete proved, we then propose a heuristic algorithm. Finally, using the service delivery cost savings as the performance metrics, while the actual system's operation trace is as the input, the effectiveness of the algorithm are comprehensively assessed.

Keywords cloud service; delivery networks; cooperation; peer-to-peer; dynamic allocation

收稿日期:2011-07-26;最终修改稿收到日期:2011-10-25.本课题得到国家"九七三"重点基础研究发展规划项目基金(2011CB302600)、国家自然科学基金(90818028,61003226)、国家杰出青年科学基金(60625203)、国家科技重大专项"核心电子器件、高端通用芯片及基础软件"(2009ZX01043-001)资助.**史佩昌**,男,1981 年生,博士研究生,中国计算机学会(CCF)会员,主要研究方向为分布式计算、内容分发网络.E-mail: pcshi.nudt@gmail.com. **王怀民**,男,1962 年生,博士,教授,主要研究领域为分布式计算、信息安全和计算机软件. **尹 刚**,男,1975 年生,博士,讲师,主要研究方向为分布式计算. **刘雪宁**,男,1980 年生,博士,主要研究方向为内容分发网络. **袁小群**,男,1976 年生,博士研究生,主要研究方向为内容分发网络. **史殿习**,男,1966 年生,博士,副教授,主要研究方向为分布式计算、普适计算.

1 引 言

云服务是指以服务的形式在互联网中传递的各种应用[1]. 云服务传递网络(Cloud Service Delivery Networks,CSDN)由内容分发网络(Content Delivery Networks,CDN)演化而来[2]. CSDN 兼具传统 CDN 和云计算的双重优势,即就近服务和按需服务[3-4]. CSDN 在因特网之上以多个独立异构 CDN 协同的方式[2]构建了一层全局虚拟覆盖网,覆盖网中的元素是地理分布的服务器. 用户在请求云服务时,CSDN 将用户请求重定向至最靠近用户且具有足够服务能力的服务器节点,由其就近向用户提供按需服务.

不同类型的云服务所表现出的用户访问行为以及资源需求特点各不相同. CSDN 将物理服务器网络划分为针对缓存类应用、流媒体和下载应用以及动态应用等多个逻辑子服务器网络[3],如图 1 所示. 其中缓存应用网络(Cache Application Networks,CAN)包括静态页面、大小图片等的分发;流媒体和下载应用网络(Streaming and Download Application Networks,SDAN)包括流式和下载式流媒体以及各种软件升级包的下载等;动态应用网络(Dynamic Application Networks,DAN)包括网络购物、网络游戏和社交网站等需要与源服务器交互的网络应用. 图 1 中源服务器负责存储并依据策略向 CSDN 节点传输源内容,逻辑服务器覆盖网根据不同类型云服务需求从物理服务器覆盖网络中动态选择和构建而成. 每个逻辑子服务器网络中服务器位置和数量如何选择,即 CSDN 资源的动态分配,主要依据所传递业务的用户访问请求分布以及 CSDN 系统对性能和开销等的需求. 其本质属于服务器选择问题[5].

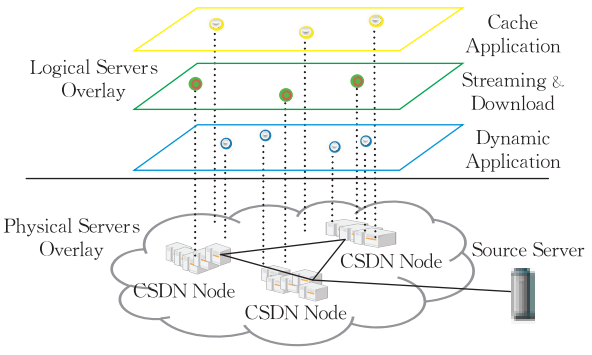


图 1 CSDN 的网络模型[3]

从理论层面看,服务器选择与服务器部署[6-10]的模型类似. 服务器部署一直以来都是学术界和工业界关注的热点和难点,现有的研究也产生了众多有价值的成果. 但随着云服务种类和规模的飞速增长,CSDN 的基础架构、用户访问规模和行为、服务分发模式、性能和开销需求以及 CSDN 的瓶颈问题等都发生了复杂的变化. 传统服务器部署的理论研究成果难以完全满足目前条件下 CSDN 的服务器选择需求. 本文按照问题建模、模型分析、模型求解和评估的思路对服务器选择问题进行了建模并提出了求解算法,同时以服务传递开销节省作为性能评价指标,以实际系统的运行数据作为输入验证了该求解算法在解决 CSDN 中资源动态分配的有效性.

本文主要贡献体现在如下 4 个方面:

(1)通过分析多种互联网传递业务资源需求特点,从与 CSDN 传递业务类型的特点相关联(如 DAN 中计算和带宽是瓶颈资源、SDAN 中内存和带宽是瓶颈资源等)的特殊视角,以 SDAN 为研究对象展开对 CSDN 中 SDAN 类业务服务器资源动态分配问题的研究;

(2)通过分析 SDAN 类热点内容传递中内存资源是瓶颈以及 SDAN 中内容可 P2P 的双重特点,从理论层面将 CSDN 中资源动态分配问题(如不做特别说明,本文的资源动态分配问题主要指面向 CSDN 中 SDAN 类业务的资源动态分配问题),即服务器的动态选择问题建模为多维设备选址模型(Multi-dimensional Facility Location Model,MFLM);

(3)通过对 MFLM 模型的分析和 NP 完全性证明,提出了针对 SDAN 类业务的一种启发式求解算法;

(4)提出了针对 MFLM 的性能分析模型,并对 MFLM 求解算法的有效性进行了全面的评估分析.

因此本文综合了 CSDN 服务器选择问题的本质变化,并以设备选址模型为基础,建立了解决这类问题的模型并给出了启发式求解算法. 本文第 2 节提出和定义服务器选择问题,并对两种代表性理论模型进行综述;第 3 节对问题建模,提出云服务传递网络资源动态配置模型——多维设备选址模型,并对模型复杂性进行分析;第 4 节根据对多维设备选址模型的 NP 完全性分析,提出针对该特定类型云服务传递的启发式求解算法;第 5 节提出针对 MFLM 的性能分析模型,并以实际系统运行数据为输入,对 MFLM 启发式求解算法进行全面的评估分析;最后总结本文并展望下一步的研究工作.

2　服务器选择问题和相关理论模型

2.1　问题的提出和定义

传统服务器部署问题是 CDN 领域的研究热点和难点，也是一个学术界和工业界普遍关注和公知的经典问题. 服务器部署侧重于物理服务器的选址[10-11]研究，其中如何选择地址和如何确定在哪个地址部署多少服务器等是相对静态的问题，因为物理服务器拓扑网络一旦形成，再次改动的复杂度和开销都非常高. 如 CDN 领域最具代表性的 Akamai 网络中服务器规模 2010 年已达 73 000 台，并且跨 70 多个国家和 1000 多个网络[12]，已经有了相对稳定和成熟的拓扑结构，很难大规模改动或重新部署. 因此逻辑层面服务器的动态选择，即服务器资源的动态分配和物理服务器的增量部署更有研究价值. 所谓物理服务器的增量部署是指如何根据现有的物理服务器网络，通过适当增加部分服务器的方式大幅提高分发网络性能. 如文献[2]中提到 Huang 等人认为与 Akamai 具有不同物理服务器覆盖网络的 Limelight 系统，如果在合适的位置增加 9 个大服务器节点，其性能从平均响应时间来看可以与 Akamai 接近. 本文重点研究服务器的动态选择，即以当前的物理服务器网络拓扑为前提约束条件，根据 CSDN 分发业务的动态需求，从物理服务器网络中动态选择部分地理位置分布的服务器形成一个专门针对该服务分发的逻辑子服务器网络. 该子分发网络可依据业务需求的动态变化而演化. 因此服务器选择问题可定义如下.

定义 1（CSDN 的服务器选择问题）. CSDN 根据不同类型云服务的资源需求，以服务性能和开销等为指标，从物理部署的服务器覆盖网络中动态选择部分地理分布的服务器形成逻辑子服务器网络. 服务器选择问题的实质是从何处选多少服务器资源构造逻辑子网络. 难度体现在：服务器的选择如何在性能和开销之间取得一种很好的均衡. 服务器选择也可看成是一个资源动态分配的过程，选择或分配的结果是在物理部署的 CSDN 有限服务器网络资源约束下，针对不同类型的应用构建若干个分布式的逻辑子服务器网络.

如图 1 所示，针对当前各种类型分发业务资源需求的不同特点，物理 CSDN 服务器网络在逻辑层面被划分为多个逻辑子服务器网络. 如对延迟和丢包敏感的缓存应用类；带宽和缓存资源容易成为瓶颈资源的流媒体和文件共享；实时性要求较高的动态应用等. 子逻辑覆盖网的动态构造问题，即服务器选择问题. 其中根据经验，将被请求内容按其热度降序排序，则 Zipf 分布中"非长尾部分"的高热点内容大约是 CSDN 单服务器节点缓存能力的数倍至几十倍，实际系统中存在单个热点文件的大小超过单台服务器内存资源 4 倍的现象①. 因此为了确保不会发生频繁的缓存不命中而影响用户的访问延迟（Patterson 等人[13]认为硬盘的延迟大约在 5 ms～35 ms，内存的延迟大约在 52 ns～225 ns，两者之间的差距高达 5～6 个数量级）和进一步引发磁盘 I\O 瓶颈，我们将缓存资源作为瓶颈参数之一.

2.2　相关理论模型

从理论层面服务器选择问题和服务器部署问题类似. 文献[11]从数学建模角度，将服务器部署模型划分为基于确定信息、基于概率模型和基于博弈论等多种模型. 本节重点综述与本文问题最密切相关的两种基于确定信息的经典模型，以帮助我们理解后面的问题建模、模型分析以及模型求解和验证.

2.2.1　设备选址模型[7,10-11]

设备选址问题定义如下：给定一个地址集合 $L=\{i\}$ 和用户集合 $U=\{j\}$，地址 i 表示设备可能被放置的位置. 在位置 i 处放置一个设备引发的成本为 f_i. 每个用户 j 必定会被指派给离其最近的一个设备，这将引发的成本标记为 $d_j \times c_{ij}$，其中 d_j 表示来自用户 j 的需求，c_{ij} 表示位于地址 i 处的设备和用户 j 之间的网络距离. 目标是：发现一种解决办法（如确定所选择的位置和部署在位置中的设备数），在服务所有用户的情况下使上述成本总和最小，即

$$\text{Min} \sum_{i\in L} f_i + \sum_{j\in U} d_j c_{ij} \tag{1}$$

其中 $\sum_{i\in L} f_i$ 表示设备选址中的一次性投入成本，$\sum_{j\in U} d_j c_{ij}$ 表示边际运营成本，与动态变化的用户需求和采用设备选址模型系统的映射算法[3]等相关.

2.2.2　最小 K 中值模型[7,11]

最小 K 中值问题定义如下：给定 n 个设备，用集合 $F=\{i_1, i_2, \cdots, i_n\}$ 表示，从其中选出 k 个设备组成集合 $S=\{s_1, s_2, \cdots, s_k\}$. 用户用集合 $U=\{j_1, j_2, \cdots, j_m\}$ 表示，然后指派用户 j 到离其最近的设备. 如果用户 j 已经被指派给设备 i，则引发开销 $d_j \times c_{ij}$，其中 d_j 表示来自用户 j 的需求，c_{ij} 表示设

① ChinaCache. http://www.chinacache.com/

备 i 和用户 j 之间的网络距离. 目标是选择 k 个设备,最小化分配代价总和,即找一个集合 S,其中 $S\subseteq F$,且 $|S|<k$,使得

$$\text{Min}\sum_{j\in U} d_j\times c_{ij} \tag{2}$$

上述两个模型之间的区别体现于两点:(1)最小 K 中值模型中设备无初始化成本;(2)最小 K 中值模型对所选择的设备个数增加了一个上限 k.

2.3 其它相关研究

近年来有关服务器选择或部署等服务器资源分配的研究有很多:Li 等人[6]从用户体验角度,提出一种基于结构树的节点部署模型,并设计了一种复杂度为 $O(N^2M)$ 的搜索算法. 其中,M 和 N 分别为选择节点和被选节点数. 该模型只考虑了节点与用户之间的流量和延迟;Qiu 等人[7]将无 Peer 辅助的静态服务器部署问题建模为最小 K 中值问题,并提出了很多简单高效的贪心算法,部署决策的依据主要有用户延迟、请求速率和负载等. 文献[10]将该类问题提炼为经典的设备选址模型,以用户的连接开销作为主要衡量指标并部分地考虑了物理服务器的一次性投入成本. 并在文献[14]中证明了该问题为 NP 完全问题. 此后围绕这类 NP 完全问题的启发式求解算法展开了很多研究. 由于本文解决多维设备选址问题的思路基于 SDAN 特例,因此本文暂不对大量的启发式求解算法进行综述. Presti 等人[15]综合考虑了用户请求重定向问题和服务器动态部署问题,依据分布式和局部机制以较低开销和复杂度实现了服务器的动态部署. 针对服务器数量,服务器的利用率以及最优服务器和新增或新删除服务器间的距离等之间给出了一种有效的折衷方法. 文献[5]提出了一个分布式系统 DONAR,传统服务器选择方法或者基于高可靠和低可扩展的集中式协同方法,或者依赖于次优的甚至不稳定的请求分配的分布式启发式方法. 而在 DONAR 中,通过在分布式节点中运行简单、高效的局部协同决策机制,为客户请求做出服务器选择决策. DONAR 的优势不仅在于稳定和高效,而且综合考虑了用户的性能和服务器的负载.

上述相关研究有两个鲜明的特点:(1)有效解决了不考虑 Peer 辅助情况下的服务器选择问题;(2)不适用于 Peer 辅助下引入新瓶颈参数的服务器选择问题. 有关 P2P 场景中服务器选择问题和考虑计算资源瓶颈参数的相关研究有:文献[16-17]提出了非 CDN 架构下 P2P 系统中服务器的部署算法;Oppenheimer 等人[18]在非 Peer 辅助的真实系统中对服务器部署算法进行验证,指出 CPU 应该作为服务器部署问题中除带宽资源之外的重要瓶颈参数;Relan 等人①对非 Peer 辅助情况下引入 CPU 瓶颈参数的服务器部署模型进行了研究. 在 CSDN 中动态应用类逻辑子服务器网络构造适用于引入 CPU 瓶颈参数. 但 SDAN 的瓶颈更在于内存资源. DONAR[5]基于 CoralCDN 提出将服务器负载考虑在内,但仅考虑了服务器的带宽负载,没有考虑内存成为瓶颈时多参数服务器选择.

综上现有相关研究,尚没有适合针对 SDAN 这种需要 Peer 辅助和引入内存瓶颈参数的 CSDN 服务器部署问题的理论模型和求解算法. 本文剩余部分如不作特别说明,CSDN 资源的动态分配等同于针对 SDAN 类特定云服务的逻辑子服务器网络的动态构造,即服务器选择.

3 CSDN 资源的动态分配问题

本节我们将 CSDN 服务器资源的动态分配问题建模为多维设备选址问题. 如图 1,CSDN 的 3 个逻辑子服务器网络分别负责 3 种不同类型云服务的传递. 其中 SDAN 有 3 个不同于其它逻辑子服务器网络的特点,使得优化其资源动态分配,对整个 CSDN 显得尤为重要. 特点 1,SDAN 所分发的流媒体和大文件下载占据了互联网的大部分流量,一些机构如思科预测此类流量 2014 年在互联网流量中的比例将达到 91%②;特点 2,SDAN 类业务可以采用 P2P 的方式降低服务器的负载,这与其它根据用户需求线性增加服务器资源的分配模式有本质不同;特点 3,SDAN 所分发内容中的热点内容大小远超过某 CSDN 服务器节点中多台服务器的内存总和. 本文将服务器内存使用作为 CSDN 的新瓶颈参数引入到设备选址模型中,该参数的引入改变了传统的 CDN 调度机制:如果用户请求热点内容,但该热点内容并没有被缓存在内存中,而且该节点的内存资源已经不够用,这种情况下,即使带宽资源有剩余,也会将当前的部分用户请求指派给临近缓存有该热点内容的服务器节点,目的是避免频繁的缓存不命中,从而降低 CSDN 的服务质量和效能.

① Relan V, Sonawane B. Multidimensional facility location problem in Content Distribution Networks. http://userpages.umbc.edu/~bhushan1/641.pdf

② http://www.cisco.com/en/US/netsol/ns827/networking solutions sub solution.html

3.1 问题建模

本节我们将 CSDN 的资源动态分配问题建模为多维设备选址模型 MFLM. MFLM 在经典设备选址模型[10]基础上，将内存资源作为 CSDN 资源动态分配的新增参数. 同时鉴于 SDAN 所分发业务可采用 Peer-to-Peer 方式，热点文件对服务器资源的需求并不与用户请求数成正比. MFLM 问题定义如下.

定义 2. 多维设备选址问题. 给定 CSDN 服务器节点集合 $L=\{l_1,l_2,\cdots,l_T\}$ 和服务器集合 $S=\{s_1,s_2,\cdots,s_N\}$. 其中每个服务器节点中包含 S 中的若干台服务器，任一服务器 j 的额定带宽能力为 N_j，额定内存大小为 C_j. 所有地理分布的用户用集合 U 表示. 根据用户的分布和用户的请求，从地理分布的 CSDN 各个服务器节点中自适应地选择数目不等的服务器，来响应用户的请求. CSDN 大文件的传递中，缓存命中率是影响传输性能的一个重要因素，现实中缓存高热点内容的内存资源已经成为瓶颈资源之一. 因此缓存的使用参数 c_j 已经被整合到了服务器选择的成本中，即有时即使服务器带宽资源有剩余，也可能会将用户请求重定向至其它缓存所请求文件的节点. 选择一个服务器的运行成本为 f_j，假设用户 i 被指派给服务器 j，则用 d_{ij} 表示用户与服务器之间的网络距离. MFLM 的目标是：发现一种资源动态分配方法，在服务用户和控制成本开销之间给出一种均衡，即在保证用户服务质量的前提下，最小化总的资源分配成本：

$$\text{Min}\sum_{j\in L}f_j\times|S_j|+\sum_{j\in L}\sum_{i\in U}\gamma_i\times c_i\times b_i\times d_{ij} \quad (3)$$

其中 $|S_j|$ 表示在节点 j 处选择的服务器数目，b_i 表示用户 i 的带宽需求，因采用 P2P 机制引起的用户资源需求削减因子为 $\gamma_i(0\leq\gamma_i\leq1)$. 多维设备选址模型与设备选址模型之间的本质不同在于前者引入了内存参数，并且针对热点文件用户采用 P2P 机制，导致用户对服务器资源需求的下降. P2P 的引入对 MFLM 模型而言没有本质的影响，仅对用户数和资源需求之间关系的影响较大. 引入 P2P 的方法和采用 P2P 后对 CSDN 系统性能和开销等方面产生的影响等细节问题在后续工作中进行深入研究和探讨.

3.2 MFLM 模型分析

定理 1. CSDN 中 MFLM 问题是 NP 完全问题.

证明. 下面分两步来证明 MFLM 问题是 NP 完全的，首先证明 MFLM 是 NP 问题，再证 MFLM 是 NP 难的.

步 1. MFLM 问题属于 NP 问题.

给定一个带权重的图 $G=\langle V,E\rangle$，其中顶点由 n 个服务器和 m 个用户构成. 每个用户 i 的带宽需求为 $band(i)$，内存需求为 $cache(i)$，每个用户 i 都需要被指派给某个服务器 j. 这个指派带来的连接成本为 $band(i)\times cache(i)\times d_{ij}$，其中 d_{ij} 表示用户 i 与服务器 j 之间的网络距离. 每个服务器 j 的内存上限为 C_j，带宽上限为 B_j. 并且部分请求热点文件的用户，由于采用 P2P 机制的原因，需在原有的基础上按比例减少其带宽和内存需求，假设削减因子为 $\gamma(0\leq\gamma\leq1)$，则总成本降低为原来的 γ 倍. A 是将所有用户指派给服务器后的总开销. 问题是给定的指派中是否存在总开销为 A 的方案？假设 $T=$“非确定性图灵机”，输入为 n 个服务器构成的集合 S、m 个用户组成的集合 U 和 A 的猜测函数 f. 函数 f 可确认给定的指派方案的总开销是否为 A. 如果确定成功，则停止；否则继续. 因此至少在给定一个用户到服务器指派的情况下，可以在多项式时间内确定其总开销是否为 A. 因此 MFLM 问题属于 NP 问题.

步 2. MFLM 问题属于 NP 难.

假设存在可求解 MFLM 问题的多项式时间算法 T. 那么一定可将上述算法简化后用以解决 FLP (Facility Location Problem)[14]问题. 即将内存约束去掉，用户到服务器的指派仅依赖于带宽和距离两个限制条件. 因此 FLP 问题能用 MFLM 的多项式时间算法 T 在多项式时间内求解. 然而文献[14]中证明 FLP 是 NP 难的，因此 MFLP 也是 NP 难的.

综上步 1 和步 2 可知，MFLP 是 NP 完全问题.

推论 1. CSDN 中的 MFLP 问题无任何近似算法.

证明. 因为 MFLP 问题是 NP 完全问题，因此 MFLP 有近似算法的前提是 P=NP. 然而 P 是否等于 NP 是非确定性的多项式复杂程度问题. 因此 CSDN 中的 MFLP 问题无任何近似算法.

解决 NP 完全问题的途径有 5 种，本文考虑针对 SDAN 中 peer 有贡献，且内存资源成为继带宽资源之外的瓶颈参数这一特定问题进行启发式求解.

4 MFLM 模型启发式求解算法

MFLM 是 NP 完全问题，因此没有针对该问题的一般性方法. 本节我们通过增加一些前提假设和

约束改变,针对 SDAN 逻辑子服务器网络动态构造的特例,给出相应的求解方法.本节求解算法所涉及的假设,均建立在 CSDN 系统实际运行状况的基础上.

4.1 算法求解的相关基础

为了便于理解问题,本节我们首先介绍 CSDN 网络资源可动态配置的相关基础.

CSDN 物理服务器的分布信息.MFLM 的目的是在当前物理服务器网络中按需动态选择服务器构造逻辑子服务器网络,因此 CSDN 的服务器节点地理位置信息、每个节点中服务器的数量、服务器节点间网络距离等有关信息是解决 MFLM 问题的必要条件之一.

CSDN 用户的分布信息.通过查询用户的 IP 地址,确定用户所在的区域.这是用户请求重定向的基本依据.一般在确定用户所在区域和负责该区域的 CSDN 服务器节点后,理论上重定向机制会将所有属于该区域的用户重定向至该区域的 CSDN 服务器节点.即由最靠近用户的服务器服务用户,符合 CSDN 就近服务的原则.如果该节点有足够的服务能力,则该区域所有用户都由该节点的服务器服务.如果该节点没有足够的服务能力响应该区域中的所有用户请求,则在其它一些调度机制的作用下,该区域的部分用户将被重定向至其它服务器节点.

重定向机制.直观而言,重定向机制就是确保用户请求总是被最靠近用户的服务器响应.实际系统中重定向的依据主要包括用户和服务器之间的网络距离是否最近,服务器可用的带宽资源是否可满足当前的用户需求.在 MFLM 中,内存的使用也成了重定向的依据,即在请求热点文件时,仅有足够的带宽资源和用户与服务器之间的网络距离这两个参数还不足以决定用户会被某服务节点中的服务器服务.提高缓存命中率的内存资源也被作为重定向的一个瓶颈参数.同时在特例 SDAN 中,由于 P2P 机制被引入到高热度文件的分发,因此重定向的依据参数变的更加复杂.

此外还有其它一些不确定变量,如单个服务器的带宽服务能力和内存服务能力、单个用户的需求、热点文件的界定、热点文件的大小等相关信息.在后续算法中会对这些变量进行讨论.

4.2 多内存协同算法

MFLM 在经典设备选址模型中引入了内存使用作为热点文件分发的一个瓶颈参数,因此如何进行多个服务器节点的内存资源协同是 MFLM 求解算法的重要部分.多节点内存资源协同的基本思想是在当前服务器节点的内存资源不足以缓存所有热点内容时,可以临近服务器节点的内存作为补充.

假设 SDAN 中有 M 个文件,用集合 $F=\{f_1,f_2,\cdots,f_M\}$ 表示,集合中文件按其热度降序排列.其中文件 f_i 的大小为 $size(f_i)$.文件的热度服从 Mandelbrot-zipf(k,N,q,s) 分布[19],即文件热度函数 $hot(f_i)$ 服务如下关系:

$$hot(f_i)=1\Big/\Big[(k+q)^s\times\sum_{i=1}^{N}(i+q)^s\Big],\ 1\leq k\leq N \tag{4}$$

其中 k 表示次序,N 表示文件总数,s 和 q 为分布参数.文件的热度越高,文件被请求访问的次数就越多.设置一个文件热度阈值 hot_thred,热度超过 hot_thred 的文件被称为热点文件,热点文件用集合 $F^*=\{f_1,f_2,\cdots,f_{M'}\}$ 表示,则所有热点文件的总大小用 $size(F^*)$ 表示.

$$size(F^*)=\sum_{i=1}^{M'}size(f_i) \tag{5}$$

CSDN 的内存协同有两种:一是 CSDN 节点内多内存协同,一般情况下由于单个服务器内存缓存热点内容的能力十分有限,因此 CSDN 采用节点内所有内存资源协同的方式提供缓存服务,确保高缓存命中率;二是节点间内存协同,当 $size(F^*)$ 大于一个服务器节点的内存总容量时,为了确保热点文件的缓存命中率,CSDN 将几个邻近 CSDN 节点的内存资源以节点间内存协同的方式提供高缓存命中服务,即利用邻近 CSDN 节点的内存资源缓存当前 CSDN 节点缓存不了的内容,如图 2 所示.

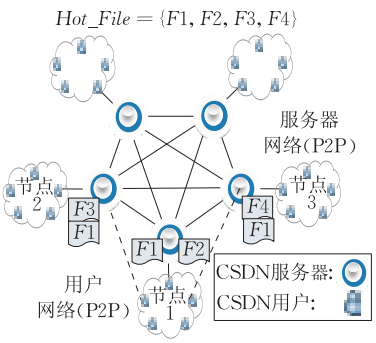


图 2 多服务器节点内存协同

图 2 中 CSDN 节点 1 中所有的内存资源合作缓存的内容包括热点文件集合 $F1$ 和 $F2$.但当前节点 1 中用户访问请求所涉及的热点内容集合为 $F1\cup F2\cup F3\cup F4$,由于节点 1 的内存资源不足以缓存所有的热点文件,而邻近节点 2 缓存热点文件 $F1$ 和

$F3$，邻近节点 3 缓存热点文件 $F1$ 和 $F4$. 因此在均衡缓存不命中开销和非就近访问开销的基础上，节点 1 中请求访问热点文件 $F3$ 和 $F4$ 的用户可被重定向至节点 2 和节点 3，实现多内存协同.

如何均衡缓存不命中开销和非就近访问开销以及多内存协同的细节见算法 1.

算法 1.　多 Cache 协同算法.

1. For $\forall l_j \in L$, while $L=\{l_1, l_2, \cdots, l_T\}$
   \\select one CSDN node
2. Collecting the information about servers F_t inside l_j;
3. Computing the available bandwidth ab_j and cache resource of servers ac_j in l_j;
4. Collecting users information U_t originally served by l_j;
5. Determine the hot file set F;
6. Collecting the distance information between the adjacent server nodes set L_k and l_j;
7. Collecting the hot file index in the cache of L_k and l_j;
8. If $size(F)<$the cache capacity inside l_j then
9. All users' requests from l_j are served by servers within l_j;
10. Else
11. do{
12. Let U' denote the rest users whose request is cache miss;
13. Select some server nodes near to l_j who caching the hot file required by users U'';
14. Pre-evaluating the additional connection cost $add\_con\_cost(U'')$ caused by U'';
15. Computing the missing cost $mis\_cost(U'')$;
16. If $add\_con\_cost(U'')<mis\_cost(U'')$
17. {Redirect U'' to the nearest server nodes whose cache has the hot file;}
18. Else
19. {Reject U'' or using other cache replace algorithm avoiding cache miss;}
20. $U'=U'-U''$;
21. }until ($U'=\varnothing$)
22. End If
23. End For

这是一个分布式算法，运行于每个 CSDN 服务器节点. 算法 1 的第 2～7 行，表示对任一个 CSDN 服务器节点均需要收集其辖域内的服务器资源、用户数量、热点文件、邻近 CSDN 服务器节点与当前节点的距离以及这些节点缓存的热点内容索引等信息，为后续用户请求的重定向和多内存协同等提供依据. 第 8、9 行表示如果当前的服务器节点 l_j 具有足够服务其辖域内所有用户请求的能力，即节点 l_j 有足够的带宽资源响应用户的服务传递需求，同时具有足够的内存资源缓存所有的热点文件以确保高缓存命中率. 第 11～21 行的递归过程表示对于节点 l_j，部分用户 U' 对热点文件的请求在 l_j 中缓存不命中，CSDN 通过预评估将 U' 中部分用户 U'' 重定向至邻近节点的连接开销和 U'' 对 l_j 请求的缓存不命中开销，决定是将 U'' 重定向至其邻近节点还是拒绝用户 U'' 请求或通过采用其它缓存算法[20] 实现热点文件的缓存替换，直到所有用户的请求为空. 鉴于篇幅限制，所涉及的其它问题：如何预评估连接开销、缓存不命中开销、进行内存协同的服务器节点数量和集合的确定等细节暂不详述. 由于该分布式算法的时间复杂度取决于对热点内容缓存不命中的请求处理，显然其时间复杂度与集合 U' 中用户个数 $|U'|$ 相关. 由第 11～21 行可知，多内存协同算法的时间复杂度为 $O(|U'|)$.

4.3　Peer 辅助的服务器选择算法

构成 SDAN 逻辑子网络由被选择的服务器集合 $S^*=\{s_1, s_2, \cdots, s_{N'}\}$ 构成，在任一 CSDN 服务器节点 l_t 处构成 SDAN 网络的子服务器集合记为 $S_t=\{s_1, s_2, \cdots, s_{N''}\}$. 同时根据热点文件的流行性，不失一般性可假设节点 l_t 处理论上所需缓存热点文件的总大小应该和当前网络中所有热点文件总大小 $size(F^*)$ 相同，实际中某个 CSDN 服务器节点所在地区的用户一般也会对所有的高热点内容发出次数不等的访问请求. 但根据经验，l_t 处的内存资源往往不足以缓存所有热点内容，即它们之间有如下关系：

$$size(F^*) \geq \sum_{j=1}^{N''} C_j \tag{6}$$

SDAN 的用户集合 $U=\{u_1, u_2, \cdots, u_P\}$，在 CSDN 节点 l_t 处的用户集合为 $U_t=\{u_1, u_2, \cdots, u_{P'}\}$. 由于用户请求服从 λ 泊松分布[15]，l_t 处用户数占用户总数的百分比为 p_t，则 l_t 处用户请求速率为 $\lambda_t=\lambda \times p_t$，其中 p_t 满足 $\sum_{t=1}^{T} p_t=1$. 用户是分布的，服务器也是分布的. 且位于不同地理位置的 CSDN 服务器节点中的服务器数量并不相同. 考虑到 CSDN 需多家企业协同，因此构成 CSDN 的服务器异构的比重较大. 但从资源分配的角度，所有服务器本身只是个资源实体，不失一般性我们可假设粒度大小不一的 CSDN 服务器节点中的服务器同构. 任一服务器 s_j 的带宽能力用 N_j 表示，内存大小用 C_j 表示. 用

$num(\lambda_t)$表示节点 l_t 处用户数，即 $num(\lambda_t)=|U_t|$，假设平均每个用户的带宽需求为 α，则节点 l_t 处用户对服务器带宽资源的需求 n_j 表示为

$$n_j=num(\lambda_t)\times\alpha \tag{7}$$

假设节点 l_t 处可缓存的热点文件大小为 c_j，其中 $c_j<size(F^*)$，则

$$c_j=C_j\times\lceil n_j/N_j\rceil \tag{8}$$

MFLM 模型中如某服务器未被选中，则称该服务器处于闲置状态，否则任一被选中节点的服务器 j 都有运行成本 f_j^s，该成本由服务器折旧开销、运行开销和购买入网带宽等边际开销构成. 用户 i 与服务器 j 的连接成本表示为用户到为其提供服务的服务器之间的距离 d_{ij}. 模型 MFLM 的优化目标是在保证性能的前提下，最小化系统开销：

$$\text{Min}\sum_{j=1}^{N'}f_j^s+\sum_{i=1}^{P}\sum_{j=1}^{N'}n_j\times c_j\times d_{ij} \tag{9}$$

式中前者表示从 CSDN 物理服务器网络选择服务器构建 SDAN 逻辑子服务器网络引起的运行成本，后者表示 CSDN 系统运行的边际成本. 该目标的物理含义是，给定带宽资源、内存资源和用户需求的情况下，尽量用较近的资源服务用户. MFLM 引入内存作为新的瓶颈参数是指，即使带宽资源仍有剩余，只要现有内存大小比热点文件小，CSDN 服务器节点也会拒绝当前用户访问某文件的请求. 然后选择将用户请求重定向至离用户次近的服务器节点，并将用户所访问的热点内容以推的方式部署到该节点的内存中，即分配服务器带宽资源和内存资源.

SDAN 逻辑子网络有其特殊性，与另外两类业务相比，该类业务的多用户间可采取 P2P 的方式降低对服务器的资源需求. 如图 3 所示，SDAN 可采用用户辅助的模式进行内容分发，而 CAN 和 DAN 的服务分发则多采用 C/S 模式.

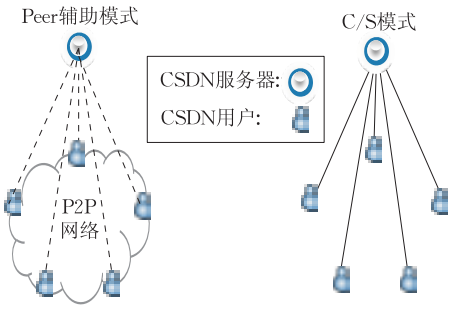


图 3 Peer 辅助模式和 C/S 模式

考虑到 SDAN 进行内容分发可采用 P2P 模式的特殊性，我们假设某 CSDN 节点 l_t 处 k 个用户中有 t 个用户，其中 $0<t<k+1$，分别具有不同的上载能力 $\rho_1,\rho_2,\cdots,\rho_t$，则在大量用户发出请求时，用户对服务器的带宽资源需求并不会随着用户需求的上升而上升. 假设此时内存资源和带宽资源一样按比例降低，则 l_t 处的需求则变为

$$n_j'=\left(1-\sum_{r=1}^{t}\rho_r\Big/k\times\alpha\right)n_j \tag{10}$$

鉴于文献[21-22]等认为资源满足用户需求即代表云服务传递质量有保障，因此我们可得到如下有关性能和开销的一种折衷，即 MFLM 的目标函数：

$$\text{Min}\sum_{j=1}^{N'}f_j^s+\sum_{i=1}^{P}\sum_{j=1}^{N'}n_j'\times c_j\times d_{ij} \tag{11}$$

如何采用 Peer 辅助的方式进行服务器选择和 Peer 辅助方式对多内存协同的影响等见算法 2.

算法 2. Peer 辅助服务选择算法.

1. For $\forall l_j\in L$, while $L=\{l_1,l_2,\cdots,l_T\}$
2. Analyze and rank the hot file F;
3. For each hot file f_j, $i=1,2,\cdots,|F|$ with descending order;
4. Given $hot(f_j)$, $size(f_j)$ and hot_thred;
5. If $(hot(f_j)>hot\_thred)$
6. The users accessing f_j with number of $num(f_j)$ forming P2P overlay;
7. Computing the bandwidth reduced by $\Delta n_j^i=num(f_j)\times\bar{\rho}_i$;
8. The cache reduced by $\Delta c_j^i=size(f_j)$;
9. $n_j=n_j-\Delta n_j^i$; $c_j=c_j-\triangle c_j^i$;
10. Else
11. Stop and check whether multi-cache coordination needed in current time window;
12. End If
13. Observe the parameters of hot_thred, ρ and λ according to different CSDN nodes.
14. End For
15. End For

算法 2 的第 2 行首先对热点文件进行了分析，以当前 CSDN 服务器节点 l_j 为研究对象，统计出某个时间窗口内不同热点文件的访问用户数. 第 3 行将所有热点文件按其访问用户数(热度)降序排列，第 4 行给出了其中某热点文件的热度、该文件的大小以及当前文件热度的阈值等. 第 5、6 行首先判断热点文件的热度是否超过热度阈值，然后使热点文件 f_j 的所有用户相互共享彼此拥有的内容形成 P2P 覆盖网. 第 7、8 行计算所有请求访问文件 f_j 的用户在采用 P2P 方式后对带宽资源需求的降低量 Δn_j^i 和缓存资源需求的降低量 Δc_j^i，其中 $\bar{\rho}_i$ 表示访问热点文件 f_j 所有用户的平均上载能力. 算法 2 的第

9 行和参数 hot_thred 的调整可影响算法 1 中多内存协同的临界条件. 同算法 1 一致，该分布式算法的时间复杂度取决于对针对热点文件构造 P2P 覆盖网络的处理，显然其时间复杂度与集合 F 中的文件数 $|F|$ 相关. 因此算法 2 的时间复杂度为 $O(|F|\times|F'_u|)$，其中 $|F'_u|$ 表示需要进行内存协同的低热度文件的用户数.

5　性能分析

本节，我们提出了评估 MFLM 启发式算法的性能指标和性能分析模型，并对 MFLM 求解算法的性能进行了全面的评估分析.

5.1　性能指标和分析模型

为了建立 MFLM 性能分析，我们需要建立包含 CSDN 服务器节点集合、每个节点内存资源、可进行内存资源协同的邻近 CSDN 节点集合、本地缓存命中访问连接延迟、本地缓存不命中访问延迟、访问邻近 CSDN 节点缓存命中内容的延迟和文件热度分布等多参数的分析模型. 模型假设与当前节点邻近的可进行内存协同的 CSDN 服务器节点有 $m-1$ 个. 由于 P2P 分发系统一般将待分发文件分割为大小相等的文件片段，不失一般性本文假设此类待传递的热点内容由 N 个热度服从 Mandelbrot Zipf 分布 $f(i)$[23] 的文件片段组成. 并且对于任意 CSDN 节点而言，热点文件的热度分布均一致. 每个节点内存资源的缓存能力为 k 个文件片段. 其中一个用户从其所属本地 CSDN 节点内存中获取单位热点内容的访问延迟和缓存不命中的访问延迟分别为 l_{cache} 和 $l_{\text{cache}}+l_{\text{disk}}$. 如果用户所访问的热点内容本地缓存不命中，则依据 MFLM 算法用户以 $1/m$ 的概率从其邻近 $m-1$ 个 CSDN 节点中任意一个的缓存中下载，且从邻近 CSDN 节点缓存中下载内容的延迟统一设置为 $l_{\text{cache}}+e_{\text{cache}}$，同时，不失一般性，本文假设 $l_{\text{disk}}>e_{\text{cache}}$. MFLM 启用 P2P 机制后，分析模型中做如下假设：首先，可进行内存协同的 m 个 CSDN节点中热度排名前 M 的文件的所有用户进行 P2P，根据前面的假设每个 CSDN 中可进行 P2P 的热点文件的概率同样也为 $1/m$；其次，在高热度文件采用 P2P 后，热度排名次序紧邻在 mk 之后的 M 个热点文件同理也将以 $1/m$ 的概率分别缓存在 m 个 CSDN 服务器节点的内存中，同时 $M+mk<N$；

文献[21-22]等研究表明云服务的传递质量与用户访问延迟之间具有一定的负相关关系. 因此本文将获取云服务传递中各种因素产生的用户访问延迟作为衡量服务质量的指标. 性能分析的目的是分析采用 MFLM 算法后对云服务传递性能的提高，即采用 MFLM 算法后所降低的用户访问延迟. 用 D_{single} 表示未使用多 CSDN 节点内存资源协同的用户访问总延迟，由于单个 CSDN 节点的缓存能力是 k 个文件片段，则

$$D_{\text{single}}=l_{\text{cache}}\sum_{i=1}^{N}f(i)+l_{\text{disk}}\sum_{i=k+1}^{N}f(i) \tag{12}$$

即所有超过本地 CSDN 节点缓存能力的文件均为缓存不命中，与所有访问均缓存命中的情况相比，会额外多引入 $l_{\text{disk}}\sum_{i=k+1}^{N}f(i)$ 的访问延迟. 在未启用 P2P 机制的情况下采用多 CSDN 节点内存协同意味着，mk 个热点文件本地缓存命中的概率为 $1/m$，从其它 $m-1$ 个 CSDN 节点中缓存命中的概率为 $m-1/m$. 用 D_{multi} 表示此种情况下所需的用户访问总延迟，则

$$D_{\text{multi}}=l_{\text{cache}}\sum_{i=1}^{N}f(i)+e_{\text{cache}}\frac{m-1}{m}\sum_{i=1}^{mk}f(i)+l_{\text{disk}}\sum_{i=mk+1}^{N}f(i) \tag{13}$$

由于 MFLM 包含多 CSDN 节点内存协同和 P2P 机制，且定义文件热度阈值 f_{thred}，在文件热度高于该热度阈值时，系统启动 P2P 机制. 假设采用 P2P 的热点文件为 M 个，且这 M 个文件原来均匀分布于 m 个 CSDN 节点. 在 M 个文件采用 P2P 之后，另外 M 个未在任何内存中缓存的剩余热点文件将以 $1/m$ 的概率分别缓存于 m 个内存中，用 $D_{\text{multi-P2P}}$ 表示采用 P2P 机制后所节省的总延迟，则

$$D_{\text{multi-P2P}}=l_{\text{cache}}\sum_{i=M+1}^{N}f(i)+e_{\text{cache}}\frac{m-1}{m}\sum_{i=M+1}^{mk+M}f(i)+l_{\text{disk}}\sum_{i=mk+1+M}^{N}f(i) \tag{14}$$

当 $M=0$ 时，$D_{\text{multi-P2P}}=D_{\text{multi}}$. 根据系统实际的运行情况和经验，可知 $M<N-mk$.

用 $Gain_{\text{multi-P2P}}=(D_{\text{single}}-D_{\text{multi-P2P}})/D_{\text{single}}$ 表示采用多内存协同和 P2P 机制后所节省的用户访问总延迟比率，则

$$Gain_{\text{multi-P2P}}=\frac{l_{\text{cache}}\sum_{i=1}^{M}f(i)+l_{\text{disk}}\sum_{i=k+1}^{mk+M}f(i)-e_{\text{cache}}\frac{m-1}{m}\sum_{i=M+1}^{mk+M}f(i)}{l_{\text{cache}}\sum_{i=1}^{N}f(i)+l_{\text{disk}}\sum_{i=k+1}^{N}f(i)} \tag{15}$$

与其它延迟 l_{disk} 和 e_{cache} 相比，l_{cache} 的值相对较小，因此为了简化上述方程式，我们将 l_{cache} 设置为

0,定义 $\theta=e_{\text{cache}}/l_{\text{disk}}$,其中有关 f_i 的定义可参见式(4)中 $hot(f_i)$,则

$$Gain_{\text{multi-P2P}}=\frac{\sum_{i=mk+1}^{mk+M}f(i)-\theta\frac{m-1}{m}\sum_{i=M+1}^{mk+M}f(i)}{\sum_{i=k+1}^{N}f(i)} \tag{16}$$

进一步定义 $F(x,y)=\sum_{i=x}^{y}f(i)$,则上式可变为

$$Gain_{\text{multi-P2P}}=\frac{F(k+1,mk+M)-\theta\frac{m-1}{m}F(M+1,mk+M)}{F(k+1,N)} \tag{17}$$

5.2 性能评估

5.2.1 实际系统的日志数据

根据相关研究[19,23],我们认为 SDAN 逻辑子服务器网络中可 P2P 的大文件和视频等热点文件服从 Mandelbrot Zipf 分布. 文献[19]以被动测量的方法针对 Gnutella 文件共享系统进行了为期 8 个月的测量. Gnutella 中有两类结点:超结点和叶结点. 测量中部署了一个超级结点,并使其并发连接数达到 500,其中约 350 个连接为超级结点间的连接. 同时 350 个超级结点中每个结点平均可连接 10 个超级结点. 以此类推,收集 7 跳以内路由所覆盖的所有用户的流量. 并从数千自治域中挑选出 9 个自治域作为研究对象,分析了 P2P 系统中文件热度的概率密度分布服从 Mandelbrot Zipf 分布. 本文以某代表性自治域中热点文件的概率密度分布 Mzipf(4,0.78)(分布参数 $q=4$,$s=0.74$)为后续性能评估分析的依据,如图 4 所示. 并给出了在 q 取不同值时 Mandelbrot Zipf 分布的变化趋势. 后续算法评估分析中将以 Gnutella 该实际数据为输入,分析式(17)中采用多 CSDN 节点内存协同和采用 P2P 对访问开销的影响.

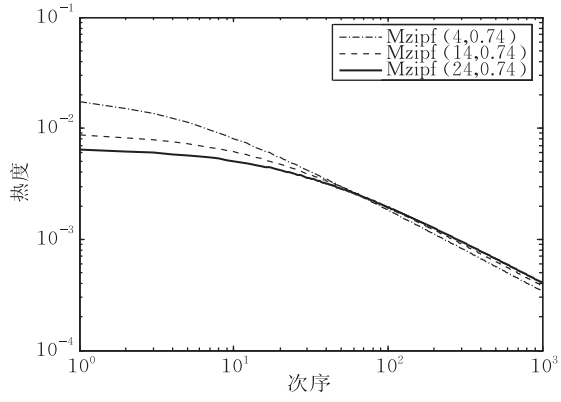


图 4 不同热点文件的热度分布

对于不同 CSDN 节点间延迟,CDN 实验室①通过 Net Clust 工具,对约 6 万~12 万在线用户(电信用户占约 60%,教育网用户约 33%,其它网络运营商用户约为 7%)进行了抽样测量. 测量时间为 2010 年 5 月 18 日~5 月 26 日,并从 5GB 监控数据分析了周末和非周末时间某 CSDN 节点到其它 CSDN 节点间的网络延迟. 图 5 为一次覆盖全中国 31 个省、直辖市和自治区的 CSDN 节点间延迟的测量,其中某 CSDN 节点与其它节点间延迟小于 120 ms 的仅为个位数.

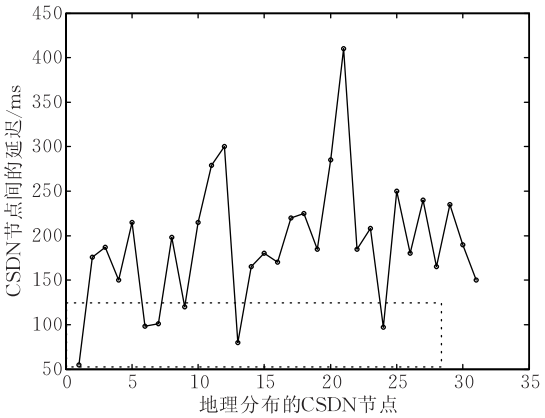


图 5 CSDN 中某节点与其它节点间延迟

5.2.2 参数设置和性能分析

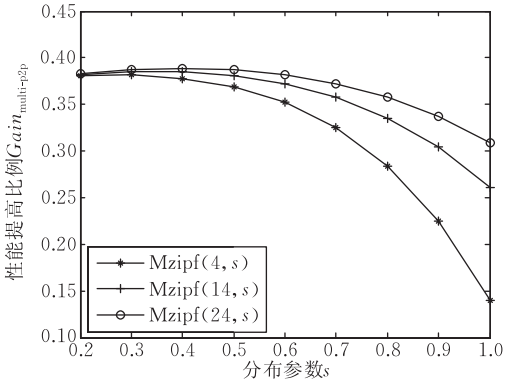
在如下性能评估分析中,CSDN 节点数设置为 300. 依据中国互联网中不同 CSDN 节点间的网络延迟,将可进行内存协同的节点数设置为 4. 根据文献[13],外部缓存命中访问延迟和本地缓存不命中访问延迟的比例 θ 为 0.3,并可在合理范围变动. 热点文件总个数为 N,而每个节点的内存可缓存的内容占热点文件总数的比例为 0.2. 我们在基于实际系统运行数据对 MFLM 启发式求解算法进行全面评估分析时,考虑一次只变动一个参数,假设其它参数均已设置了合理值.

图 6(a)为不使用 P2P 机制且 q 分别取值 4,14,24,s 在区间[0.2,1]内变化时节约的访问延迟比率. 其中 $Gain_{\text{multi-P2P}}$ 随 s 增大而降低,且 q 取值越大降低的速度越慢;图 6(b)为不使用 P2P 机制且 s 分别取值 0.2,0.4,0.6,q 在区间[0,100]内变化时节约的访问延迟比率. 其中 $Gain_{\text{multi-P2P}}$ 随 q 增大而增大,且 s 越大变化越剧烈. 图 6(c)为使用 P2P 机制且 q 分别取值 4,14,24,s 在区间[0.2,1]内变化时节约的访问延迟比率. 与图 6(a)相比,图 6(c)为使用 P2P 机制(M 取值为 0.1N)后,$Gain_{\text{multi-P2P}}$ 增加幅度特别大,随着 s 值的增大而增大,对 q 值的影响

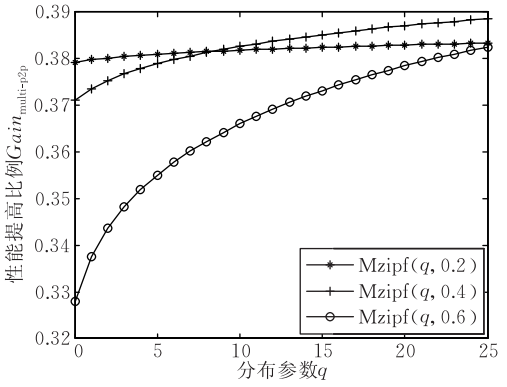
① CDN Research Institute. http://lab.cdnlab.org/

不是非常敏感，并且系统的整体性能在采用 P2P 机制后有较大幅度提高的访问延迟比率. 图 6(d)为使用 P2P 机制且 s 分别取值 0.2，0.4，0.6，q 在区间[0,24]内变化时节约的访问延迟比率. 图中采用 P2P 机制后 s 取值越大，系统性能越好，即 s 取值为 0.6 时比 s 取值为 0.2 时系统的性能约高出 4 个百分点，并且与图 6(c)反映出的规律一致：系统性能的提高受 q 值的变化影响不大. 同理与图 6(b)相比，图 6(d)使用 P2P 机制后，系统性能约有 20 个百分点的提高.

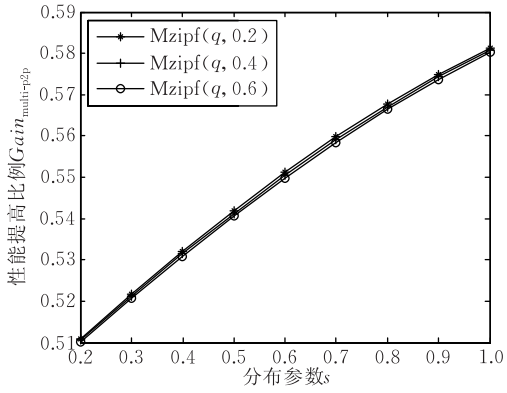
图 7(a)为未使用 P2P 机制且 q 和 s 分别取特定值时节约的访问延迟比率与 θ 的关系. 图 7(b)为使用 P2P 机制且 q 和 s 分别取特定值时节约的访问延迟比率与 θ 的关系. 通过分析图 7(a)中 $Gain_{\text{multi-P2P}}$ 随 θ 变化的规律可知，外部缓存命中访问延迟和本地缓存不命中访问延迟 θ 越大，多内存协同带来的延迟降低就越小，尤其是 θ 在超过 0.5 之后多内存协同几乎不会给系统性能带来任何提升. 与图 7(a)相比，图 7(b)在此基础上启用了 P2P 机制，两者主要的不同在于所节约的访问延迟比率的程度不同. 后者在 q 和 s 取特定值时，带来的系统性能提升在 20%～80%之间，并且 s 和 q 值的微小变化对系统性能的提升几乎没有影响.



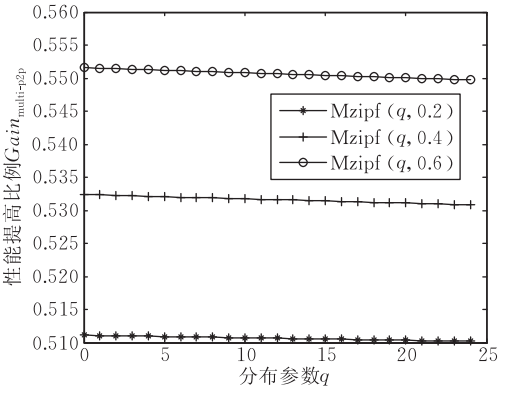
(a) 不使用P2P且s取值区间为[0.2,1]



(b) 不使用P2P且q取值区间为[0,24]

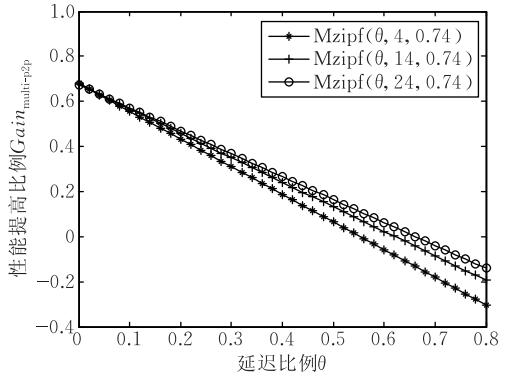


(c) 使用P2P且s取值区间为[0.2,1]

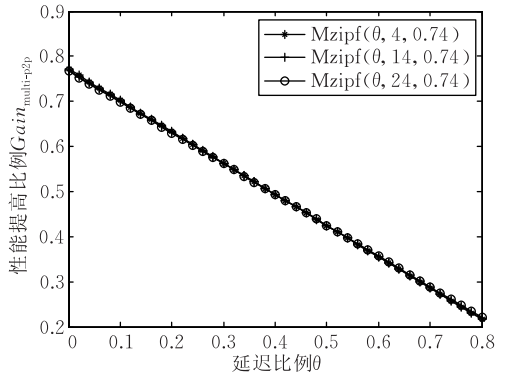


(d) 使用P2P且q取值区间为[0,24]

图 6　在 q 和 s 变化时使用和不使用 P2P 所节约的访问延迟比率



(a) 不使用P2P且θ取不同值



(b) 使用P2P且θ取不同值

图 7　在 θ 变化时使用和不使用 P2P 所节约的访问延迟比率

图 8(a)为未使用 P2P 机制，$s=0.74$ 和 θ 分别取特定值时节约的访问延迟比率与 θ 的关系. 图 8(a)中 $Gain_{\text{multi-P2P}}$ 随着 q 值的增大而略有增大，且 q 越大 $Gain_{\text{multi-P2P}}$ 越小. 图 8(b)为使用 P2P 机制，$s=0.74$ 和 θ 分别取特定值时节约的访问延迟比率与 θ 的关系. 图 8(b)在图 9(a)基础上启用了 P2P 机制，整体上对 $Gain_{\text{multi-P2P}}$ 的贡献增大了，但在 P2P 机制启用后基本对 q 的变化没有影响，且 θ 值越小 $Gain_{\text{multi-P2P}}$ 越大. 图 8(c)为未使用 P2P 机制，$q=4$ 和 θ 分别取特定值时节约的访问延迟比率与 s 的关系. 其中 $Gain_{\text{multi-P2P}}$ 随着 s 值的增大而略有减小，且 θ 越大 $Gain_{\text{multi-P2P}}$ 越小. 图 8(d)为使用 P2P 机制且 θ 和 q 分别取特定值时节约的访问延迟比率与 s 的关系. 图 8(d)在图 8(c)基础上启用了 P2P 机制，整体上对 $Gain_{\text{multi-P2P}}$ 的贡献增大了；并且在 P2P 机制启用后对 s 变化敏感度随 θ 值增大而增大.

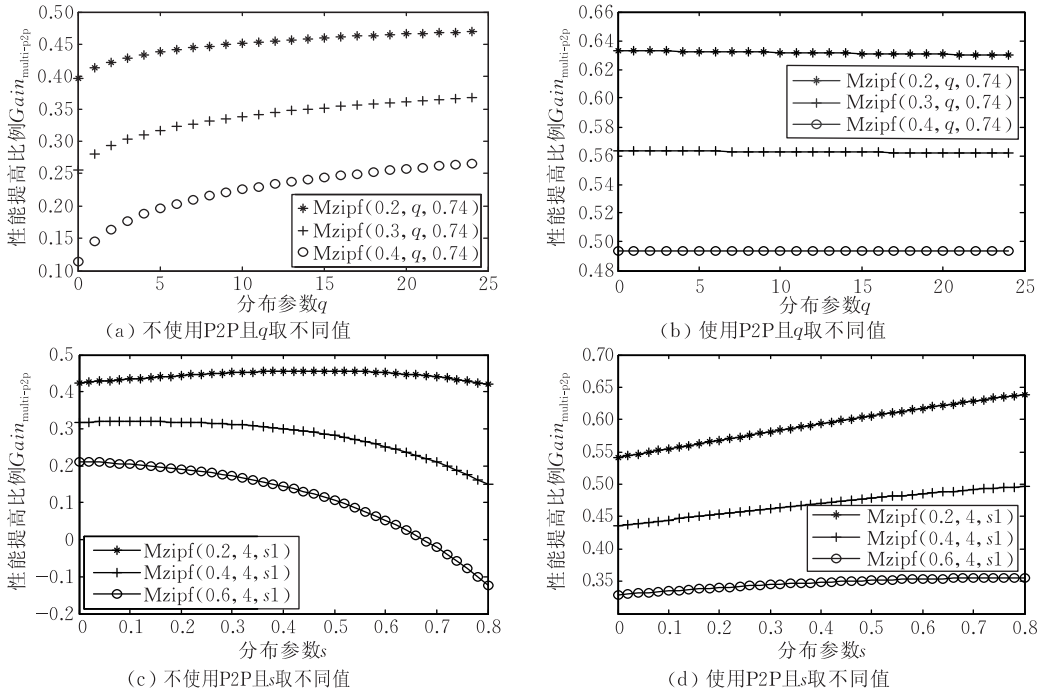


图 8 在 q 和 s 变化时使用和不使用 P2P 所节约的访问延迟比率

通过上述评估分析可知：在仅采用多 CSDN 节点内存协同方法的前提下，Gainmulti-P2P 与参数 s 具有一定的反比关系和高相关度，与参数 q 具有一定的正比关系和低相关度；采用 P2P 机制对解决内存瓶颈方面效果十分明显，但与参数 θ 成反比；同时，外部缓存命中访问延迟和本地缓存不命中访问延迟之比越大，多内存协同效果就越小，而且对 s 变化敏感度随 θ 值增大而增大. 该评估分析结果验证了 MFLM 算法对 CSDN 中 SDAN 逻辑子服务器网络即资源动态分配的效果，并且多邻近 CSDN 节点内存协同和 P2P 机制是 SDAN 整体性能提升的两个主要因素. 也是针对 SDAN 特例求解 MFLM 模型算法中的两个核心问题. 同时性能评估分析通过评估 q，s 和 θ 等参数变化(即需求变化)情况下，MFLM 求解方法的有效性，从一定程度上间接说明了多邻近 CSDN 节点内存协同和 P2P 机制对于解决复杂的 MFLM 问题具有一定的普适性.

6 总结及展望

云服务传递网络 CSDN 以就近和按需的方式向用户提供云传递服务. 由于 CSDN 覆盖不同类型业务的传递，因此其物理服务器网络被划分为针对缓存类应用、流媒体和下载应用以及动态应用等多种不同类型业务的逻辑子服务器网络. 不同类型业务资源需求特点各不相同，其中用于流媒体和下载类云服务传递的逻辑子服务器网络 SDAN，消耗了 CSDN 的大部分资源. 与动态应用类业务不同：SDAN 中内存资源是除带宽资源之外的另一种瓶颈资源，而动态应用逻辑子服务器网络 DAN 中计算和带宽资源是瓶颈；与 DAN 中内容需要回源[2]不同，SDAN 中内容具有可 P2P 的特性. 本文重点研究和解决该类型云服务传递的资源动态分配问题. 因此根据该类型业务内存资源和带宽资源同为

瓶颈资源以及该类型热点内容可采用 P2P 机制的两个特点，我们将该问题建模为多维设备选址模型.在对该模型 NP 完全性分析的基础上，提出了一种针对 SDAN 特例的启发式求解算法，并对 MFLM 求解算法的有效性进行了全面的评估分析.

服务器资源动态分配问题一直都是学术界研究的重点和难点，未来围绕该问题我们将对如下几个基本问题展开研究：(1)互联网业务用户访问请求分布和流量行为统计特点；(2)互联网多种类型业务的分类方法和分类结果；(3)在上述两点研究结果的基础上，研究针对不同类型业务间服务器资源的动态分配问题.

致　谢　作者在此感谢为本文贡献有价值思路、讨论和数据的各位专家及工程师：清华大学 CDN 研究所所长尹浩教授，ChinaCache 公司运维部宋裔智副总裁，富媒体下载组的李政工程师和 P2P 部张玮经理等！

参　考　文　献

[1] Armbrust M, Fox A, Griffith R, Joseph R D, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M. A view of cloud computing. Communications of the ACM, 2010, 53(5): 862-876

[2] Yin H, Liu X N, Min G Y, Lin C. Content delivery networks: A bridge between emerging applications and future IP networks. IEEE Network Magazine, 2010, 24(4): 52-56

[3] Nygren E, Sitaraman R K, Sun J. The Akamai network: A platform for high-performance internet applications. ACM SIGOPS Operating Systems Review, 2010, 44(3): 2-19

[4] Albanese F, Carra D, Michiardi P, Bestavros A. Cloud-based content distribution on a budget. Boston University, Boston: Technical Report BUCS-TR-2010-022, 2010

[5] Wendell P, Jiang J W, Freedman M J, Rexford J. DONAR decentralized server selection for cloud services//Proceedings of the SIGCOMM. New Delhi, India, 2010: 231-242

[6] Li B, Deng X, Go L M, Sohraby K. On the optimal placement of web proxies in the internet: Linertopology//Proceedings of the 8th IFIP Conference on High Performance Networking. Networking. Vienna, Austria, 1998: 485-495

[7] Qiu L L, Padmanabhan N V, Voelker G M. On the placement of web server replicas//Proceedings of the IEEE INFOCOM. Alaska, USA, 2001: 1587-1596

[8] Wang Z, Jiang H, Sun Y, Li J, Liu J, Eryk D. A K-coordinated decentralized replica placement algorithm//Proceedings of the ISCC. Riccione, Italy, 2010: 811-816

[9] Jiang H, Wang Z, Wong A K, Li Jun, Li Zhongcheng. A replica placement algorithm for hybrid CDN-P2P architecture//Proceedings of the ICPADS. Shenzhen, China, 2009: 758-763

[10] Drezner Z, Hamacher H W. Facility Location: Applications and Theory. New York: Springer, 2004: 132-141

[11] Yin Hao, Yuan Xiao-Qun, Lin Chuang, Zhang Fa, Pang Shan-Chen, Liu Zhing-Yong. The survey of service nodes placement theories for content networks. Chinese Journal of Computers, 2010, 33(9): 1611-1620(in Chinese)
(尹浩，袁小群，林闯，张法，庞善臣，刘志勇. 内容网络服务节点部署理论综述. 计算机学报，2010，33(11)：1611-1620)

[12] Cohen J, Repantis T, McDermott S, Smith S, Wein J. Keeping track of 70000+servers: The Akamai query system//Proceedings of the 24th USENIX Large Installation System Administration Conference. 2010: 223-238

[13] Patterson D A. Latency lags bandwidth. Communications of the ACM, 2004, 47(10): 71-75

[14] Averbakh I. Complexity of robust single facility location problems on networks with uncertain edge lengths. Journal of Discrete Applied Mathematics, 2003, 127(3): 505-522

[15] Presti F L, Bartolini N, Petrioli C. Dynamic replica placement and user request redirection in content delivery networks//Proceedings of the ICC. Seoul, Korea, 2005: 1495-1501

[16] Shen H. EAD: An efficient and adaptive decentralized file replication algorithm in P2P file sharing systems//Proceedings of the P2P. Aachen, Germany, 2008: 99-108

[17] Shorfuzzaman M, Graham P, Eskicioglu R. Popularity-driven dynamic replica placement in hierarchical data grids//Proceedings of the PDCAT. Dunedin, New Zealand, 2008: 524-531

[18] Oppenheimer D, Chun B. Patterson D, Snoeren A C, Vahdat A. Service placement in a shared wide-area platform//Proceedings of the ATEC. Dallas, USA, 2006: 26-39

[19] Hefeeda M, Saleh O. Traffic modeling and proportional partial caching for Peer-to-Peer systems. IEEE/ACM Transactions on Networking, 2008, 16(6): 1447-1460

[20] Borst S, Gupta V, Walid A. Distributed caching algorithms for content distribution networks//Proceedings of the INFOCOM. Shanghai, China, 2011: 1-9

[21] Leighton T. Improving performance on the Internet. ACM Queue, 2008, 6(6): 20-29

[22] Chen K T, Huang C Y, Huang P, Lei C L. Quantifying skype user satisfaction//Proceedings of the SIGCOMM. 2006

[23] Hefeeda M, Noorizadeh B. On the benefits of cooperative proxy caching for peer-to-peer traffic. IEEE Transactions on Parallel and Distributed Systems, 2010, 21(7): 998-1010

[24] Huang C, Wang A, Li J, Ross K W. Measuring and evaluating large-scale CDNs. Microsoft Research, Washington State: Technical Repor MSR-TR-2008-106, 2008

[25] Xu D Y, Kulkarni S S, Rosenberg C, Chai H K. Analysis of a CDN-P2P hybrid architecture for cost-effective streaming distribution. Journal of Multimedia Systems, 2006, 11(1): 383-399

SHI Pei-Chang, born in 1981, Ph. D. candidate. His current research interests include distributed computing.

WANG Huai-Min, born in 1962, Ph. D., professor, Ph. D. supervisor. His research interests include distributed computing, information security and computer software.

YIN Gang, born in 1975, Ph. D., lecturer. His research interests include distributed computing and information security.

LIU Xue-Ning, born in 1980, Ph. D.. His research interests include live streaming and content delivery network.

YUAN Xiao-Qun, born in 1976, Ph. D. candidate. His research interests include server placement and content delivery network.

SHI Dian-Xi, born in 1966, Ph. D. , associate professor. His research interests include distributed computing and pervasive computing.

Background

Today the Internet traffic is increased rapidly which is produced by many kinds of applications such as Static Web, Streaming, Downloading and SNS applications etc. CSDN is an effective method to solve internet congestion problem in delivering these cloud services to geo-distributed end users. But the delivery resource allocation among different kinds of applications in CSDN is a great challenge. Fortunately, most servers and bandwidth resources of CSDN are used to deliver the streaming and downloading cloud services, and the dynamic resource allocation of their delivery resource is the main problem in CSDN. So we mainly focus on streaming and downloading applications resource allocation. In this kinds of applications, the bandwidth and cache resources have become the bottleneck resources nearly at the same level. So an important and available way to solve this problem is dynamic configuring bandwidth and cache resources.

In this paper, we mainly present a dynamic resource configuration method for CSDN and validate our method effectiveness through massive simulation evaluations. We hope that they can help and inspire the researchers and practitioners who concern the resource allocations among different kinds of applications in CSDN. This work is supported by the National Natural Science Foundation of China under Grant No. 90818028: "Behavior Monitoring and Trustworthiness-Oriented Evolution of Large-Scale Distributed Software Systems". This project aims at the online evolution methodology and a set of necessary techniques for software services like Software as a Service (SaaS) in geo-distributed cloud system. This work is also supported by the National Basic Research Program (973 Program) of China under Grant No. 2011CB302600: "Basic Research on Effective and Trustworthy Internet-Based Virtual Computing Environment (iVCE)", whose purpose is to design basic models and mechanisms for effective and trustworthy virtual computing environment which can be thought of as hybrid cloud consists of multi-scale resources. This work is also supported by the National Science Fund for Distinguished Young Scholars under Grant No. 60625203: "Computer Software" whose purpose is to design basic method and mechanisms for trustworthy software and the National "Core Electronic Devices, High-End General Purpose Chips and Fundamental Software" Major Project under Grant No. 2009ZX01043-001: "Domestic Middleware Reference Implementation and Platform" whose purpose is to design reference implementation and platform for domestic middleware.