

一种基于云存储的 CDN 视频副本部署策略

姚士佳, 朱明, 崔昊旻

(中国科学技术大学自动化系, 安徽合肥 230027)

摘要: 在线视频服务无法离开 CDN (content delivery network) 的支持. 与传统 CDN 相比, 使用云存储节点作为 CDN 的节点对视频内容进行分发可以节省大量运营费用. 在线视频服务的内容文件要尽量预先部署到离用户较近的边缘节点上, 以保证用户的 QoS (quality of service). 现有的 GS (greedy site) 内容预部署算法在有历史记录的前提下, 有着较小的货币开销, 且能满足 QoS 需求. 然而, GS 算法会导致云存储节点的负载均衡情况较差. 为此, 提出了一种称为 GUCP (greedy user core preallocation) 的部署算法, 该算法利用 k -means 算法将负载过重节点上的用户分配到负载较轻的节点以平衡负载, 解决 GS 算法在负载均衡方面存在的不足. 实验表明, GUCP 算法在系统使用费用、QoS 性能方面与 GS 算法十分接近, 在负载均衡性能上远远优于 GS 算法.

关键词: 云存储; CDN; 副本部署; 负载均衡

中图分类号: TP393 **文献标识码:** A doi:10.3969/j.issn.0253-2778.2014.07.003

引用格式: Yao Shijia, Zhu Ming, Cui Haoming. A novel video replica placement strategy for storage cloud-based CDN[J]. Journal of University of Science and Technology of China, 2014, 44(7): 554-562.

姚士佳, 朱明, 崔昊旻. 一种基于云存储的 CDN 视频副本部署策略[J]. 中国科学技术大学学报, 2014, 44(7): 554-562.

A novel video replica placement strategy for storage cloud-based CDN

YAO Shijia, ZHU Ming, CUI Haoming

(Department of Automation, University of Science and Technology of China, Hefei 230027, China)

Abstract: Online video service needs the support of CDN (content delivery network) which can be costly. Using cloud storage nodes to deliver video content can help solve the problem. To guarantee the users' QoS, CDN should pre-deploy the content files of online video service to the edge nodes which are close to the users. The existing GS (greedy site) placement algorithm can satisfy the QoS requirements at a relatively smaller cost when the historical log is provided. However, the GS algorithm will result in bad load balance among cloud storage nodes. A novel replica placement algorithm named GUCP (greedy user core preallocation) was proposed to solve this problem. The algorithm assigned users on overloaded nodes to lightly-loaded ones using the k -means algorithm, in order to balance the load. The numerical experimental results show that the cost and QoS performance of GUCP are very close to those of GS, while its performance of load balance is much better than that of GS.

Key words: cloud storage; CDN; replica placement; load balance

收稿日期: 2013-10-11; 修回日期: 2014-04-29

基金项目: 中国科学院先导专项 (XDA06030900), 国家科技支撑计划 (2012BAH73F01) 资助.

作者简介: 姚士佳, 男, 1985 年生, 博士生. 研究方向: 云存储. E-mail: yaosj@mail.ustc.edu.cn

通讯作者: 朱明, 博士/教授. E-mail: mzhu@mail.ustc.edu.cn

0 引言

在线视频服务的快速发展,出现了大量的中小型在线视频服务企业. 视频服务本身的特点决定了它们无法离开 CDN 服务的支持,像 Akamai 及 Mirror Image 等一些传统的 CDN 提供商,部署了数以万计的数据中心与边缘服务器,用以向全球推送内容. 然而,提供商广泛使用的传统 CDN(如 Akamai)定价远远超出了中小型企业、政府机构、大学以及慈善组织^[1]所能承受范围. 中小型视频服务提供商通过建立新的数据中心或者通过租用 CDN 实现与大型内容提供商的竞争,在财力上难以承受,因此用云存储作为“廉价”的 CDN 是非常具有吸引力的. 云存储提供商提供了快速读写能力并可以方便、廉价的扩展,可以应对突然的网络带宽拥挤以及可预期的增长需求. 从使用成本与性能方面考虑,以现有的“云存储”结构进行 CDN 服务与将大量投资花在建立自己拥有的内容分发平台或者租用类似 Akamai 这样的现有 CDN 相比是十分划算的. 使用云存储作为 CDN 可以有效地减少内容存储、分发与使用的费用. 在本文余下的内容中,基于云存储的 CDN 简称为云 CDN.

由于不同的云存储服务都使用了特定的接口,并且有着不同的特点,因此难以同时直接使用多个云存储进行 CDN 服务. 通常 Web 站点使用独立的云存储分发其内容^[2],最典型的例子是 New York Times^[3]以及 SumgMug^[4]. 大部分“云存储”提供者仅仅负责基本的文件存储以及服务的推送,并没有提供与 CDN 完全一致的功能,如自动复制、故障转移、基于地理信息的双重定向以及负载均衡. MetaCDN 成功地解决了这些问题,MetaCDN 是一种使用许多存储提供商,以创建可以高效率、可靠并且地理冗余分布使用的 CDN 覆盖网^[5].

云提供者使用下面计算模型来向其消费者销售存储及带宽的使用^[5]. 存储费用由每单位时间每 GB 来计算,带宽费用由每 GB 的传输来计算. 带宽费用由为到来的数据支付的上传费用以及为输出的数据支付的下载费用两部分组成,即

$$\text{Cost} = \sum_{i=1}^n S_i T_i + \sum_{i=1}^n P_i T_i + \sum_{k=1}^n D_k T_k \quad (1)$$

作为云存储的消费者,云 CDN 通过不同的云提供者公开价格来减少使用费用. 由于云存储具备可以按需缩放的特点,云 CDN 可方便地根据需要

来调节其存储与带宽. 云 CDN 与传统 CDN 一样可为多用户以及站点提供多种资源,即云 CDN 可以提供与传统 CDN 类似的功能,同时不需要拥有和维护任何基础设施.

在线视频服务的内容文件较大,用户需要尽量短的响应时间,因此,视频内容的副本要尽量预先部署到离用户较近的 CDN 边缘节点之上. 当前在大部分线视频服务网站都会将内容预先部署到边缘 CDN 结点.

本文提出了一种针对在线视频的云 CDN 的副本部署算法. 这种算法针对云 CDN 副本部署算法负载不均的问题,提出了一种部署策略,可以有效实现 CDN 各节点负载均衡.

1 相关工作

为了有效部署 CDN 中的副本,许多学者已经做了大量的研究. 为了使内容检索代价最小化, Li 等^[6]以及 Krishnan 等^[7]在一般的网络拓扑结构中,将内容部署问题规划为一种 NP 问题,并针对树形拓扑结构提出了最佳解决方案,使检索成本最小化. Qiu 等^[8]对一些启发式算法进行评估并发现一种贪心算法可以提供最佳性能. Radoslav 等^[9]以及 Jamin 等^[10]提出一种基于扇出的启发式的算法,这种算法中,副本以最高的扇出放置到节点上. 上述方法中,一旦一组副本站点被选定,路径的选择问题也将隐含其中.

除了检索成本, Xu 等^[11]和 Jia 等^[12]进一步考虑到更新开销,而 Cidon 等^[13]考虑到存储成本. 此外, Kalpakis 等^[14]综合考虑了所有三项成本(上传,下载和存储),并只提供了对于树拓扑结构的解决方案. 复制站点之间的供应成本与复制方向相关的问题,目前仍有待研究.

大量工作已经考虑到服务质量,要求所有的用户请求在一定网络距离之内到达副本服务器. Tang 等^[15]和 Wang 等^[16]提出了优化全部存储和更新成本的算法. 他们假定请求可以从任何节点发出,并且将检索成本忽略. Rodolakis 等^[17]在优化存储和检索成本时将服务器容量限制加入到规划中.

随着云计算的快速发展,云服务被广泛地应用到在线视频服务当中. 文献[18]提出了一种优化的云协助的视频分发服务的部署方案,并对运营开销以及用户体验之间权衡问题做了深入的研究.

云计算的出现,为 QoS 提供了一种保障的多媒体服务了较为经济的方法^[19],虚拟化技术可用于细

粒度的提供内容分发和多媒体处理服务中的资源分配^[20-21]. 文献[22]提出了一种使用云协助的流媒体直播架构. 可见云计算已经广泛应用于在线视频服务当中.

Broberg 等^[5]提出的 MetaCDN 是一种使用云存储资源的低成本 CDN. 该系统提供了一种机制将内容放在不同的云存储网络中, 而后重定向用户的请求到相应的副本, 但 MetaCDN 没有给出副本放置以及请求重定向的算法.

Chen 等^[23]针对 MetaCDN 提出了一种预部署算法 GS(greedy site), 然而 GS 算法并没有考虑云存储节点的负载均衡问题.

2 云 CDN 及其性能参数

2.1 云 CDN

由于云 CDN 的使用费用是带宽与存储的费用之和, 带宽与存储的费用会随着用户负载变化而改变. 为了有效地操作云 CDN, 智能的副本放置与用户重定向策略是必要的. 在传统 CDN 中, “副本放置”问题已经被深入地研究. 然而, 已有的研究结果却因为以下的原因不能直接用于云 CDN:

很多传统 CDN 中的工作假定网络拓扑是给定的, 如树的根在源服务器. 在使用云存储的情况下, CDN 构建者有着建立所有潜在副本站点任意拓扑结构的自由. 这个拓扑可能不同于基本的网络拓扑. 因此, 云 CDN 中的副本放置是一个建立分布式路径与复制的综合问题, 与传统 CDN 的副本放置问题有着一定区别.

定义从其他站点 u 复制内容到 v 的费用为 $d(u; v)$. 传统 CDN 使用无向图的边, 即 $d(u; v) = d(v; u)$. 然而, 云存储上传下载的费用是不同的, 边是有向的. 这表明只选定副本站点是不够的, 需要对副本方向进行指定.

图 1 展示了一个在云 CDN 中副本放置的问题. 在图 1(a) 中, 实线表示潜在的副本从 C_0 开始的分发路径, 以及潜在的副本站点 $C_1 \sim C_4$, 假定每个站点都有到每个用户 U_k 的路由路径. 虚线表示能够满足用户请求 QoS 的路径. 图 1(b) 展示了副本放置问题的一种解决方案, C_1, C_3 被选定分别服务来自于 U_1, U_4 的请求, C_4 被选定服务来自于 U_2 和 U_3 的请求. C_1 被选定服务来自于 U_4 的请求, 同时副本从 C_0 通过 C_1 被转发副本到 C_3 和 C_4 .

经过基于跟踪的研究发现, 在同等条件云 CDN

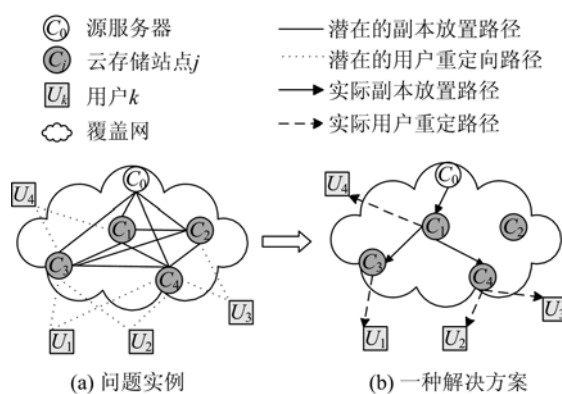


图 1 云 CDN 中的副本放置问题

Fig. 1 The problem of replica placement in cloud CDN

有效地减少了站点每个月需要支付给 CDN 服务商的费用, 从传统 CDN 的每个月 99 美元降低到每个月 2.62 美元. 文献[23]提出了一种适用于云 CDN 的副本部署 GS 算法. 由于 GS 算法并没有考虑云存储节点的负载均衡问题, 将导致资源无法得到高效使用, 因此本文提出 GUCP (greedy user core preallocation) 部署算法, 可解决 GS 算法在负载均衡方面所存在的不足.

2.2 云 CDN 使用费用描述

多个存储节点从源服务器 C_0 复制副本, 而后对终端用户的请求进行应答. 实际上, 云提供者拥有并操作多个存储站点或者数据中心, 但是, 每个存储节点仅属于一个云提供者. 属于不同云提供者的云存储节点在地理上可以位于同一区域, 但这些属于不同的云提供者的云存储节点可能会有不同的价格.

本文假定共有 m 个终端用户, $U = \{U_1, U_2, \dots, U_m\}$ 由 k 索引, n 个云存储节点 $C = \{C_1, C_2, \dots, C_n\}$ 由 i 或 j 索引. 假定视频副本大小为 T . 云提供者会对副本的存储以及输入和输出的传输进行收费. 云存储节点 C_j 对副本进行存储, 该站将在每个月收取以 GB 为单位的存储费用 S_j ; 副本上传到站点 C_j 时, 将被收取以 GB 为单位的传输费用 P_j ; 为了分发副本到其他站点或推送内容到终端用户, 将被收取以 GB 为单位的输出传输费用 D_j .

定义 V_{uv} 为从 u 到 v 的复制开销. V_{uv} 的计算方式由节点 v 作为一个云存储节点或作为一个用户来决定. V_{uv} 在这两种情况下有两种不同的含义. ① v 是云存储节点, V_{uv} 是节点打开的费用, 即节点 $v = C_j$ 从有副本的节点 $u = C_i$ 下载副本的费用. 平均来说, 本文假定在单位时间内, $F \cdot T$ 为每次更新的内容部分, F 代表更新的频率, $V_{uv} = V_{ij} = (S_j + P_j F + D_j F) T$, 一般

假定 $V_{ij} > 0$. 第一个有副本的节点 C_0 作为源服务器. 副本分发路径将会以 C_0 作为根形成一个类似于图 1(b) 的树形结构. ② v 在 V_w 中代表一个用户, 即 $v = U_k$, c 是用户 U_k 通过站点 $u = C_j$ 的存取费用. 一个终端用户 U_k 请求大小为 T_k 并且 $V_w = V_{ik} = T_k D_j$.

综上, 云 CDN 的运营费用是将内容部署到云存储节点(打开节点)的费用以及服务用户请求的费用之和, 即 $Cost = OpenCost + UserCost$. 本文通过随机方式定义了节点的上传、下载以及存储费用批量, 本文中云存储节点的费用以抽象的货币值(并非是像美元这样的具体货币)进行计量. 即在本文中, 使用费用仅用于衡量不同算法的性能, 不代表实际情况下的开销值.

2.3 云 CDN 中的服务质量(QoS)

本文中, 以 R_{jk} 表示从 C_j 到用户 U_k 的路由距离, 路由距离表示用户到服务节点的通信质量, R_{jk} 越小表示两个节点(用户与节点)之间的通信质量越好, 用户所能获得的服务质量也越好. 由于本文使用了云存储节点作为副本服务节点, 可认为所有副本服务节点的带宽、可靠性以及并发处理等能力是一致的. 将 QoS 距离视为路由距离, 距离越小, QoS 越好. 因此可将系统的 QoS 需求定义为每个用户与所分配到节点之间的路由距离 R_{jk} 不大于某个阈值 Q , 即为了满足用户对于 QoS 的要求, 需要保证用户的路由距离 $R_{jk} \leq Q$, R_{jk} 越大, 表示用户的 QoS 性能越差. GS 算法即是将用户分配到路由距离最小的存储节点.

由于无法获得用户与服务节点之间的路由、带宽信息, 本文定义用户与服务节点之间的路由距离为用户与云存储节点之间地理距离, 以用户与节点坐标之间欧氏距离表示路由距离. 云 CDN 用户数量较大, 随着用户量的增多, 路由距离总值会持续增加. 为便于对比, 本文中的测量值为所有用户与其所对应的云存储节点欧氏距离的平均值, 即系统平均路由距离. 平均路由距离 \tilde{R} 的计算公式为:

$$\tilde{R} = \sum_k^m \sqrt{(X_{\text{node}}^j - X_{\text{user}}^k)^2 + (Y_{\text{node}}^j - Y_{\text{user}}^k)^2} / m \quad (2)$$

式中, $(X_{\text{node}}^j, Y_{\text{node}}^j)$ 为用户 k 所对应节点的坐标, $(X_{\text{user}}^k, Y_{\text{user}}^k)$ 为用户 k 的坐标, m 为用户的总数. 因用户数量较多, 可认为 $\tilde{R} \leq Q$ 时, 系统中的用户 QoS 得到满足.

2.4 云 CDN 中负载均衡

在集群系统中, 负载均衡是系统的一项重要性

能. 集群系统负载均衡的目标是依据服务器的性能来分配与其相称的任务, 以确保应用程序执行时间的最小化.

在线视频服务过程中, 用户与服务节点的连接是一个长期的过程. 为保证有限的节点可同时服务更多用户, 在单个节点能力一定的情况下, 系统的负载均衡性能显得十分重要.

本文假定负载资源为服务节点所服务的用户数量, 其他资源(如存储、带宽或 CPU 等)因本文研究对象(使用云存储节点提供针对视频的 CDN 服务)的特点不予考虑, 只通过服务节点所服务的用户数量来衡量其负载情况. 本文采用负载权值的方式进行描述, 下面对负载权值^[24]进行定义.

依据节点所承担的负载与负载能力需要相匹配的原则, 假定云 CDN 系统中, 共使用了 n 台云存储服务器作为 CDN 服务节点. 其中云存储节点 $C_i (i=1, \dots, n)$ 的固有负载能力为 $\omega_i (i=1, \dots, n)$, 在云存储环境下, 可认为云存储节点的负载能力相同, 即

$$\omega_1 = \omega_2 = \dots = \omega_n = \omega \quad (3)$$

设节点 i 的负载情况为 L_i , 定义某服务节点的当前负载与其固有负载能力的比值为该服务节点的负载权值, 记为:

$$W_i = \frac{L_i}{\omega_i} = \frac{L_i}{\omega} \quad (4)$$

CDN 服务节点达到负载均衡时有:

$$W_i = W_j, i \neq j; i, j = 1, 2, \dots, n \quad (5)$$

定义云 CDN 集群的平均负载权值为:

$$\bar{W}_L = \frac{1}{n} \sum_{i=1}^n W_i = \frac{1}{n\omega} \sum_{i=1}^n L_i \quad (6)$$

那么云 CDN 集群负载均衡时, 负载权值为:

$$W_1 = W_2 = \dots = \bar{W}_L = \frac{1}{n} \sum_{i=1}^n W_i = \frac{1}{n\omega} \sum_{i=1}^n L_i \quad (7)$$

由于 n 与 ω 均可确定, 云 CDN 集群负载完全平衡时, 可近似的表达为:

$$L_1 = L_2 = \dots = L_n = \bar{W}_L \quad (8)$$

即负载情况仅由存储节点所服务用户数量体现. 定义集群负载系数, 用以衡量系统负载的平衡情况. 本文使用负载均方差来表示系统中服务节点的负载状况, 定义负载状况值 η 为:

$$\eta = \frac{\sum_{i=1}^n (L_i - \bar{W}_L)^2}{n} \quad (9)$$

3 云 CDN 中的视频副本部署策略

本文研究的 CDN 针对在线视频, 在线视频内容本身具有文件长度大、内容变化小的特点, 并且用户对内容的请求需要有快速的响应, 所以副本预先部署到云存储节点上, 可以快速响应用户对视频请求, 当前的大部视频 CDN 均采用预先部署的策略. 在有用户历史记录的前提下, 将视频内容预先“推”到边缘节点效果良好. GS 算法和 GUCP 算法均为有历史记录条件下, 将用户分配到指定的云存储节点, 同时将副本部署到节点中的算法.

实际环境下, 用户的请求分布情况随时会发生变化. 当新的用户进入系统请求 CDN 中的视频内容, 用户会依据 GS 算法被分配到与其最近的节点, 然后用户会被周期性通过 GUCP 的负载均衡算法重新分配到相应的节点.

3.1 GS 算法

文献[23]提出了 GS(greedy site)算法, GS 算法迭代的决定打开一个关闭中的站点最大效用并向其分配所有潜在的用户. 一个站点的潜在用户是在 QoS 距离较小且尚没有被分配的用户. 用户被分配到与其 QoS 距离最小的存储节点. 打开一个云存储节点, 然后寻找下一个需要打开的站点, 直到所有用户都被分配到一个节点. 定义 O_j 为打开云存储节点 C_j 的费用:

$$O_j = \begin{cases} 0, & \text{if } C_j \text{ is OPEN} \\ \min_{i \in \{i | C_i \text{ is open}\}} V_{ij}, & \text{otherwise} \end{cases} \quad (10)$$

算法 3.1 为 GS 算法的伪代码.

算法 3.1 Greedy site^[23]

```

E is the set of users who have not been assigned
Ej is the current set of users who can be assigned to Cj
while E ≠ ∅ do
    Wj ← ∑k wk, Uk ∈ Ej
    j* ← argmaxj ∈ {Cj is CLOSED}  $\frac{W_j}{W_j D_j + O_j}$ 
    Assing all users in Ej* to Cj*
    Open Cj*
    E ← E - Ej*
end while

```

3.2 问题提出

从 3.1 节可以看出, GS 算法对用户的分配只考虑 QoS 距离, 这样会导致某些时刻, 某些存储节点被分配的节点过多, 发生负载不均衡, 从而会减弱整

个系统的并行服务能力.

假定视频副本需要部署的区域共有 20 个云节点和 600 个用户. 使用 GS 算法将用户分配到节点后, 会出现负载不均衡. 此时, 负载状况值 $\eta = 1\ 411.71$, 负载最大的节点负载值 $L_{\text{Max}} = 155$, 负载最小的节点负载值 $L_{\text{Min}} = 54$.

文献[25]中定义负载阈值为 $Th = (1 + \bar{\mu})/2$, 其中 $\bar{\mu}$ 为当前系统的平均负载权值. 经实验验证, 如此定义并不符合本文的系统环境. 根据具体情况, 本文重新定义动态负载阈值为:

$$Th = \overline{W}_L^T * \alpha \quad (11)$$

变量 α 为取值范围为 (0, 1) 的负载平衡因子. 在 4.1 节中将通过实验测定, 给出 α 的取值.

当节点负载不均时, 有 $L_p > L_q$, 且 $L_p - L_q \geq Th$, $p \neq q$; $p, q = 1, 2, \dots, n$. 此时需要对用户节点的分配作出调整, 以保证负载均衡.

为了解决 GS 算法引起的负载不均问题, 本文提出了 GUCP(greedy user core preallocation) 预分配算法.

3.3 GUCP 算法

GUCP 算法根据云存储节点与用户之间的路由距离, 将用户分配到与其路由距离最小的节点后, 若出现负载不均的情况 ($L_p > L_q$), 则需要将分配到节点 p (用户量较多的节点) 的一部分用户调整到节点 q 上, 这部分用户在需要获取云 CDN 服务时, 将被重定向到节点 q . 本文假定在指定区域内的用户与节点坐标呈均匀分布.

当分配到 p, q 两个节点上的用户数量不均时, 需要从分配到节点 p 的用户中选出约 $(L_p - L_q)/2$ 个用户迁移到节点 q , 以使两节点上所分配的用户数量趋于平衡. 由于在指定区域内的用户坐标是均匀分布的, 故可利用 k -means 算法对分配到 p 的用户进行聚类, 并将节点 p 上与节点 q 最近的用户集合批量迁移到节点 q 上.

k -means 算法是一种常见的基于样本间相似性度量的间接聚类方法, 属于非监督学习的分类算法. 该算法是基于统计的分类方法, 依据测试样本在特征空间中 k 个最近邻样本中的多数样本的类别来进行分类, 因此具有直观、无需先验统计知识等特点. k -means 算法的一个明显的缺点是对不同的数据集难以给出合适的 k 值, 需要迁移大约 $(L_p - L_q)/2$ 个用户以使得 p, q 上分配的用户数量趋于接近, k -means 算法核的数量 N_k 由下式得到:

$$N_k = 2L_p / (L_p - L_q) \quad (12)$$

在经 k -means 算法所划分出来的 N_k 个用户核中, 选出一个与节点 q (用户数量较少的节点) 距离最近的核 $Core_k$, 而后将核 $Core_k$ 所对应的用户迁移到节点 q 上. 然后重新检查是否存在负载不均衡, 若存在, 重复进行节点用户的负载调整操作, 直至所有云存储节点达到负载均衡. GUCP 算法的核心思想是通过通过对过载节点对应的用户使用 k -means 算法进行聚类, 利用用户核对用户进行重新批量调整, 使集群能够在较少调整次数的前提下, 快速达到用户分配平衡. 算法 3.2 为 GUCP 算法的伪代码:

算法 3.2 Greedy user core preallocation

```

E is the set of users who have not been assigned
Ej is the current set of users who can be assigned to Cj
while Lp - Lq ≥ Th(p ≠ q; p, q = 1, 2, ..., n) do
  Nk cores ← k-means (Up)
  Find min distance Core_num between Cq and cores
  Assign users Ucore_num to Cq
end while
while E ≠ ∅ do
  Wj ← ∑k wk, Uk ∈ Ej
  j* ← argmaxj ∈ {Cj is CLOSED}  $\frac{W_j}{W_j D_j + O_j}$ 
  Assign all users in Ej* to Cj*
  Open Cj*
  E ← E - Ej*
end while

```

3.4 GUCP 算法复杂度分析

GUCP 算法实现负载均衡的主要方式是通过 k -means 算法对用户所分配的节点进行重新划分.

通过 Mathematica 软件进行数值实验分析, 可知 GUCP 算法中, 对调用 k -means 算法的次数进行最小二乘法拟合, 得出基为 $\{1, n^{\frac{1}{2}}, n^{\lambda}\}$, 其中 $0 < \lambda < 1/2$ 时, 误差平方和较小, 且所拟合的曲线能够比较准确地描述调用次数的走向趋势, 因此调用 k -means 算法是关于用户数量的增函数, 但是增的趋势越来越缓慢.

4 实验与结果分析

4.1 仿真参数

本文采用 Matlab 进行仿真实验, 假定共有 20 个云存储节点供云 CDN 使用, 节点的上传、下载、存储费用均为 0~10 之前的随机值, 且每个节点的

三种费用值均不相同. 节点可提供带宽为 10 M, 用户数的上限为 1 000, 节点随机分布在半径为 10 的圆内, 用户随机分布在外半径为 20、内半径为 10 的圆环内部. 设每个用户观看视频时占用的带宽平均为 100 kb, 20 个节点最多可同时服务 2 000 个用户, 所以选用 1 000 个用户进行仿真实验不会超过系统的负载上限. 选用 20 这个比较小的数字反映当前云存储提供商的数量并不多的现状; 而且使用 100 个以上的云存储节点进行仿真实验, 对本文的结论不会产生本质影响.

假定云 CDN 对 100 部视频内容提供副本缓存服务, 内容文件大小为 0~10 Gb 之间的随机值, 每个用户请求的内容数量为 0~10 个.

负载阈值影响着系统的负载状况值 η 以及出现负载不均后, 系统所需要进行的调整次数. 合理的阈值将保证负载状况值较低同时确保调整次数较少. 为了确定动态负载因子 α 的取值, 假定有 600 个用户使用云 CDN. 图 2、图 3 分别为用户数量为 600 的

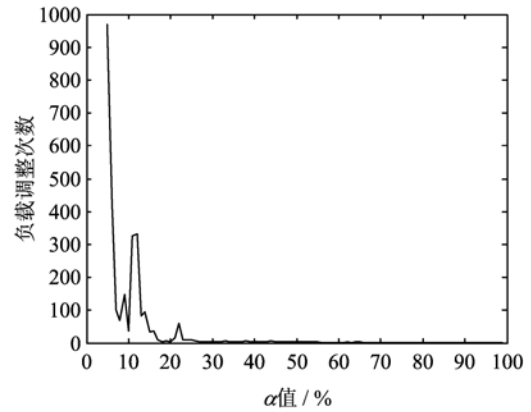


图 2 α 与负载调整次数对应关系

Fig. 2 The relationship between α and numbers of adjustments

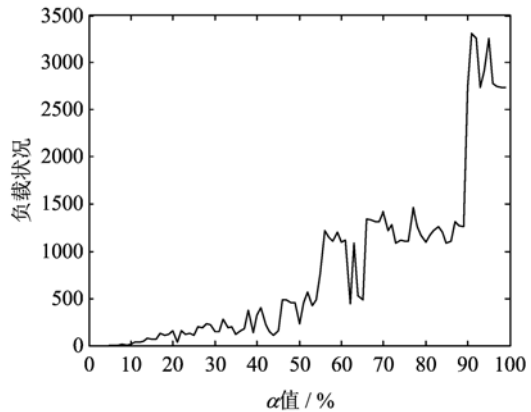


图 3 α 与负载状况对应关系

Fig. 3 The relationship between α and load situation

情况下,变量 α 与负载调整次数和负载状况值的对应关系.由图 2、图 3 可见,随着 α 的增加,负载的调整次数递减,而负载状况值保持递增.并且从 $\alpha=25\%$ 开始,系统进行负载调整次数变得较为平稳,负载状况值相对较小,因此动态负载因子 α 可取为 25%,负载阈值可定义为 $Th = \overline{W}_t * 0.25$.依据节点与用户的分布情况,假定用户与节点之间平均距离为 10,所以将 QoS 阈值 Q 定为 10,即平均路由距离的上限为 10.表 1 为仿真实验参数.

表 1 实验参数

Tab. 1 The experimental parameters

节点数	用户上限	内容数量	Q	α
20	1 000	100	10	25%

4.2 GUCP 算法耗时分析实验

图 4 中,红色点集为调用 k -means 的次数,蓝线为对调用次数进行最小二乘拟合后的函数.由图 4 可见,基为 $\{1, n^{\frac{1}{2}}, n^{\lambda}\}$, $0 < \lambda < 1/2$ 时,所得函数可较为准确地描述次数变化的趋势.最小二乘拟合的结果表明,随着用户数量增多,GUCP 算法调用 k -means 算法的次数也随之递增,但是增速趋于平稳.同时,本文还对调用 k -means 算法所耗时间进行了统计,所使用的服务器处理器为 Intel Xeon(R) 4 核,内存大小为 3 G, Linux 内核版本为 2.6.18. 经

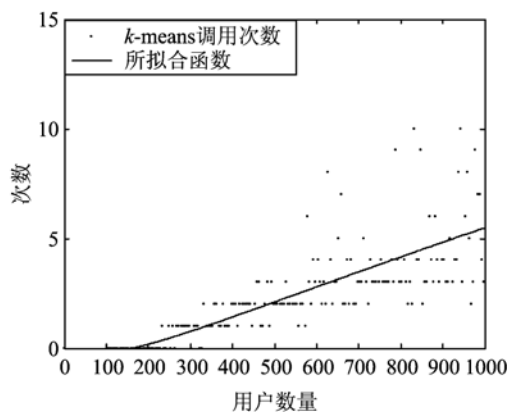


图 4 最小二乘法拟合

Fig. 4 The least squares fitting

实验发现,当用户量为 1 000 时,调用一次 k -means 算法所耗时间约为 3 ms,单次运行时间较短.实验结果表明,GUCP 算法进行负载均衡的开销是可以接受的.

4.3 对比实验结果

本文进行了三组对比实验,三组实验均采用用

户数量作为变量,对比用户数量从 100 增加到 1 000 的过程中,GS 与 GUCP 算法的负载、费用以及平均路由距离三种性能的情况.

图 5 为 GS 与 GUCP 两种算法负载情况的对比.由图 5 可见,GS 算法与 GUCP 的负载状况值 η 随用户数量的增多不断增大,但 GS 算法的负载状况值增幅远远大于 GUCP 算法,在用户数量接近 1 000 时, η 已经在 4 000 左右,可见 GS 算法的负载状况很差.而 GUCP 算法的负载一直维持在较低的水平,保持着较慢的增长.在用户数量不到 1 000 时, η 没有超过 400,表明 GUCP 算法的负载状况良好.由此可见,GUCP 算法的负载均衡性能远远好于 GS 算法.

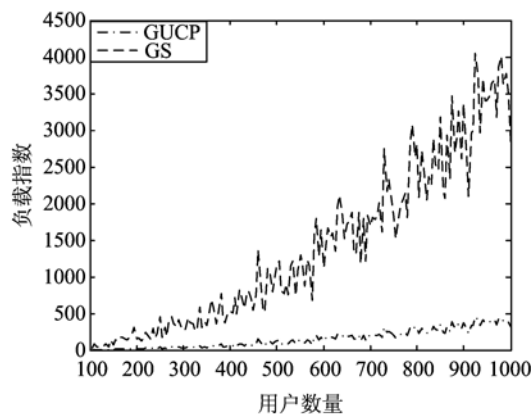


图 5 负载对比

Fig. 5 The comparison of load

图 6 为 GS 与 GUCP 两种算法费用的对比.从实验结果可以看出,GS 算法与 GUCP 算法在用户量为 250 左右的时候,系统使用费用基本一致.随着用户量的增加,GUCP 算法的使用费用开始稍高于 GS 算法,但仍小于上限 5 500.当用户量增加到一定程度

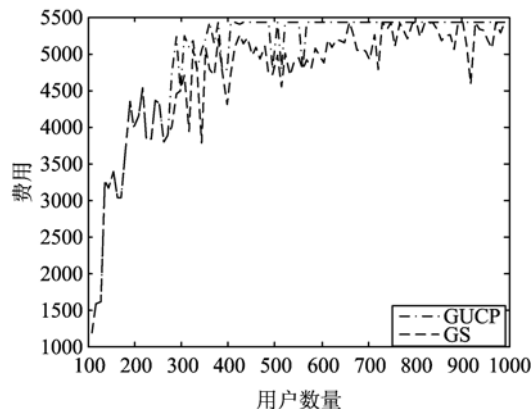


图 6 费用对比

Fig. 6 The comparison of cost

时,GS 算法与 GUCP 算法的系统费用十分接近.

图 7 为 GS 与 GUCP 两种算法平均路由距离的对比. 依据 2.3 小节中对路由距离的定义,路由距离为云存储节点与用户之间的欧氏距离. 即路由距离越大,用户的请求得到响应的时间也就越长,得到的服务质量也越差.

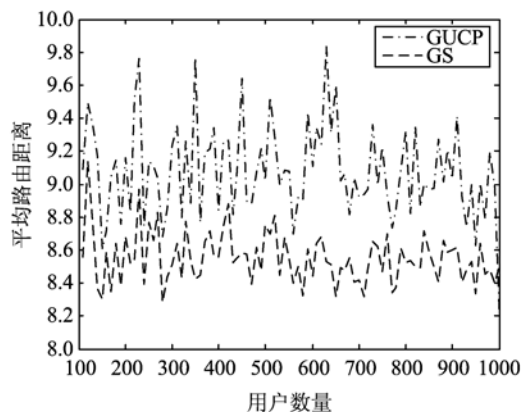


图 7 平均路由距离对比

Fig. 7 The comparison of average route distance

从实验结果可以看出,从用户数量为 100 开始,随着用户数量的增加,GUCP 算法的平均路由距离高于 GS 算法,但是二者之差维持在 1 左右. 同时平均路由距离均小于 Q ,这是符合系统要求的,从全局性能角度考虑,是可以接受的.

5 结论

本文提出了一种针对视频内容分发的云 CDN 的内容预部署算法——GUCP 算法. 算法通过使用 k -means 算法对用户进行重新划分,将负载过重节点所负担的用户批量分配给负载较轻的节点,以保证云 CDN 中云存储节点的负载均衡. 实验表明,GUCP 算法与 GS 算法相比,在费用以及 QoS 性能方面与 GS 算法十分接近,负载均衡性能有着极大提升. 可以认为 GUCP 算法是一种有效的针对视频云 CDN 的副本预部署算法.

参考文献 (References)

- [1] Rayburn D. Content delivery networking pricing, costs for outsourced video delivery[EB/OL]. <http://www.scribemedial.org/2007/12/12/cdn-pricing-video/>.
- [2] Elson J, Howell J. Handling flash crowds from your garage[C]// Proceedings of the USENIX Annual Technical Conference. Santa Clara, USA: USENIX Association, 2008: 171-184.
- [3] Gottfrid D. Self-service, prorated super computing fun! [EB/OL]. <http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-com-putting-fun/>.
- [4] MacAskill D. Scalability: Set Amazon's servers on fire, not yours[EB/OL]. <http://blogs.smugmug.com/don/files/ETech-SmugMug-Amazon-2007.pdf> (0).
- [5] Broberg J, Buyya R, Tari Z. MetaCDN: Harnessing storage clouds' for high performance content delivery [J]. Journal of Network and Computer Applications, 2009, 32(5): 1 012-1 022.
- [6] Li B, Golin M J, Italiano G F, et al. On the optimal placement of web proxies in the internet [C]// Proceedings of the 8th Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE Press, 1999, 3: 1 282-1 290.
- [7] Krishnan P, Raz D, Shavitt Y. The cache location problem [J]. IEEE/ACM Transactions on Networking, 2000, 8(5): 568-582.
- [8] Qiu L, Padmanabhan V N, Voelker G M. On the placement of web server replicas[C]// Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies. Anchorage, USA: IEEE Press, 2001: 1 587-1 596.
- [9] Radoslavov P, Govindan R, Estrin D. Topology-informed internet replica placement [J]. Computer Communications, 2002, 25(4): 384-392.
- [10] Cronin E, Jamin S, Cheng J, et al. Constrained mirror placement on the internet[J]. IEEE Journal on Selected Areas in Communications, 2002, 20(7): 1 369-1 382.
- [11] Xu J, Li B, Lee D L. Placement problems for transparent data replication proxy services[J]. IEEE Journal on Selected Areas in Communications, 2002, 20(7): 1 383-1 398.
- [12] Jia X, Li D, Hu X, et al. Placement of web-server proxies with consideration of read and update operations on the internet[J]. The Computer Journal, 2003, 46(4): 378-390.
- [13] Cidon I, Kuten S, Soffer R. Optimal allocation of electronic content [C]// Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies. Anchorage, USA: IEEE Press, 2001, 3: 1 773-1 780.
- [14] Kalpakis K, Dasgupta K, Wolfson O. Optimal placement of replicas in trees with read, write, and storage costs[J]. IEEE Transactions on Parallel and Distributed Systems, 2001, 12(6): 628-637.
- [15] Tang X Y, Xu J L. QoS-aware replica placement for content distribution[J]. IEEE Transactions on Parallel and Distributed Systems, 2005, 16(10): 921-932.

- [16] Wang H, Liu P, Wu J J. A QoS-aware heuristic algorithm for replica placement[C]// Proceedings of the 7th IEEE/ACM International Conference on Grid Computing. Bcelona, Spain: IEEE Computer Society, 2006: 96-103.
- [17] Rodolakis G, Siachalou S, Georgiadis L. Replicated server placement with QoS constraints [J]. IEEE Transactions on Parallel and Distributed Systems, 2006, 17(10): 1 151-1 162.
- [18] He J, Wu D, Zeng Y P, et al. Towards optimal deployment of cloud-assisted video distribution services [J]. IEEE Transactions on Circuit and System for Video Technologies, 2013, 23(10): 1 717-1 728.
- [19] Zhu W, Luo C, Wang J, et al. Multimedia cloud computing: Directions and applications [J]. Signal Processing Magazine, 2011, 28(3): 59-69.
- [20] Wu Y, Wu C, Li B, et al. CloudMedia: When cloud on demand meets video on demand[C]// Proceedings of the 31st International Conference on Distributed Computing Systems. Minneapolis, USA: IEEE Press, 2011: 268-277.
- [21] Li H T, Zhong L L, Liu J C, et al. Cost-effective partial migration of VoD services to content clouds [C]// Proceedings of the International Conference on Cloud Computing. Washington, USA: IEEE Press, 2011: 203-210.
- [22] Wang F, Liu J C, Chen M H. CALMS: Cloud-assisted live media streaming for globalized demands with time/region diversities [C]// Proceedings IEEE INFOCOM. Orlando, USA: IEEE Press, 2012: 199-207.
- [23] Chen F F, Guo K, Lin J, et al. Intra-cloud lightning: Building CDNs in the cloud[C]// Proceedings IEEE INFOCOM. Orlando, USA: IEEE Press, 2012: 433-441.
- [24] 郭成城, 晏蒲柳. 一种异构 Web 服务器集群动态负载均衡算法[J]. 计算机学报, 2005, 28(2): 179-184.
- [25] Godfrey B, Lakshminarayanan K, Surana S, et al. Load balancing in dynamic structured P2P systems [C]// Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies. Hong Kong, China: IEEE Press, 2004, 4: 2 253-2 262.

(上接第 553 页)

- [11] Gkoulalas-Divanis A, Verykios V S, Mokbel M F. Identifying unsafe routes for network-based trajectory privacy[C]// Proceedings of the SIAM International Conference on Data Mining. Sparks, USA: SIAM Press, 2009: 942-953.
- [12] Chow C Y, Mokbel M F, Liu X. A peer-to-peer spatial cloaking algorithm for anonymous location-based service[C]// Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems. New York: ACM Press, 2006: 171-178.
- [13] Chow C Y, Mokbel M F. Enabling private continuous queries for revealed user locations[C]// Proceedings of the 10th International Symposium on Advances in Spatial and Temporal Databases. Boston, USA: Springer, 2007: 258-273.
- [14] Ghinita G, Kalnis P, Skiadopoulos S. MobiHide: A mobile peer-to-peer system for location-based queries [C]// Proceedings of the 10th International Symposium on Advances in Spatial and Temporal Databases. Boston, USA: Springer, 2007, 221-238.
- [15] Gkoulalas-Divanis A, Verykios V S. A free terrain model for trajectory k-anonymity[C]// Proceedings of the 19th International Conference on Database and Expert Systems Applications. Turin, Italy: Springer, 2008: 49-56.
- [16] Mokbel M F, Chow C Y, Aref W G. The new Casper: Query processing for location services without compromising privacy[C]// Proceedings of the 32nd International Conference on Very Large Data Bases. Seoul, Korea: ACM Press, 2006: 763-774.
- [17] Gedik B, Liu L. Location privacy in mobile systems: A personalized anonymization model[C]// Proceedings of the 25th International Conference on Distributed Computing Systems. Columbus, USA: IEEE Press, 2005, 620-629.
- [18] Pan X, Xu Ji L, Meng X F. Protecting location privacy against location-dependent attacks in mobile services [J]. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(8): 1 506-1 519.
- [19] 潘晓, 郝兴, 孟小峰. 基于位置服务中的连续查询隐私保护研究[J]. 计算机研究与发展, 2010, 47(1): 121-129.
- [20] 杨晓春, 王雅哲, 王斌, 等. 数据发布中面向多敏感属性的隐私保护方法[J]. 计算机学报, 2008, 31(4): 574-586.