

基于二阶段的 3D 虚拟世界客户分配方法

蒲 斌¹, 崔梦天², 赵海军¹

(1. 西华师范大学计算机学院, 四川 南充 637009;

2. 西南民族大学计算机科学与技术学院, 成都 610041)

摘 要: 针对 3D 虚拟世界中的客户分配问题, 提出一种基于二阶段的分配方法。把客户分配问题分为初始分配阶段和优化分配阶段, 使得具有服务质量的客户数量最大化和分配成本最小化。在初始分配阶段为每个客户寻找一个目标服务器, 在优化分配阶段为每个客户寻找一个触点服务器, 并针对二阶段的客户分配问题提出随机分配区域算法、贪婪分配区域算法、基于虚拟位置的分配客户算法和贪婪分配客户算法。仿真结果表明, 在初始分配阶段中考虑网络时延的算法明显优于不考虑网络时延的算法, 同时能够较好地适应虚拟世界的动态变化特性, 4 种算法在网络平均通信时延和资源消耗方面都优于目前广泛采用的客户分配算法。

关键词: 二阶段; 虚拟世界; 交互性; 客户分配问题; 服务质量

中文引用格式: 蒲 斌, 崔梦天, 赵海军. 基于二阶段的 3D 虚拟世界客户分配方法[J]. 计算机工程, 2016, 42(1): 109-115.

英文引用格式: Pu Bin, Cui Mengtian, Zhao Haijun. Client Allocation Approach in 3D Virtual World Based on Two-stage[J]. Computer Engineering, 2016, 42(1): 109-115.

Client Allocation Approach in 3D Virtual World Based on Two-stage

PU Bin¹, CUI Mengtian², ZHAO Haijun¹

(1. School of Computer, China West Normal University, Nanchong, Sichuan 637009, China;

2. School of Computer Science and Technology, Southwest University for Nationalities, Chengdu 610041, China)

【Abstract】 Aiming at the client allocation problem in 3D Virtual World(VW), an allocation approach based on two-stage is proposed. The concrete implement is that the client allocation problem is divided into the initial allocation stage and refined allocation stage. The objective is that the total number of clients with Quality of Service(QoS) is maximized and the allocation cost is minimized. In the initial allocation stage, a target server is found for each client and then in the refined allocation stage, a contact server is found for each client. In the simulations, four algorithms for the two-stage allocation are simulated, and the results show that the algorithms that take into account network delays in the initial allocation stage significantly outperform those which do not. Moreover, the algorithms can well adapt to the dynamic change characteristics of VW. The performances for the allocation approach based on two-stage proposed are compared with which for the other two algorithms used widely at present, the results show that the allocation approach proposed in the average network communication delay and resource consumption outperforms the other two client allocation algorithms used widely.

【Key words】 two-stage; Virtual World(VW); interactivity; Client Allocation Problem(CAP); Quality of Service(QoS)

DOI: 10.3969/j.issn.1000-3428.2016.01.020

1 概述

近年来, 高速网络技术、计算机图像处理技术、3D 虚拟技术和 CPU 处理能力的提高使得 3D 虚拟世界(Virtual World, VW)得到了快速发展。3D 虚拟世界是一种分布式系统, 它能使众多地理上呈分

布式的客户在一个共享的、由计算机生成的 3D 虚拟世界里实时探讨或互动。在这个虚拟世界中, 客户通过各种不同输入控制其化身行为, 一个化身状态的更新传送到虚拟世界同一个区域中的其他客户, 实现客户间的互动。

本文针对客户分配问题(Client Assignment

基金项目: 国家自然科学基金资助项目(61379019); 中国博士后科学基金资助项目(2013M540704); 西华师范大学基本科研业务费专项基金资助项目(14C002); 四川省科技计划基金资助项目(2015JY0027); 四川省教育厅基金资助项目(16ZB0174); 西华师范大学创新团队基金资助项目(CXTD2015-12)。

作者简介: 蒲 斌(1972-), 男, 副教授、硕士, 主研方向为网络通信、虚拟架构; 崔梦天, 教授、博士; 赵海军, 教授。

收稿日期: 2014-11-27 **修回日期:** 2015-02-11 **E-mail:** lhp2003@126.com

Problem, CAP) 提出一种二阶段分配方法, 由初始分配阶段和优化分配阶段构成。在初始分配阶段, 把虚拟世界的区域分配给服务器; 在优化分配阶段, 把客户分配给合适的 (通常是邻近的) 服务器, 即与管理这个客户所在区域的服务器通信。基于二阶段分配原理, 提出实现客户分配的 4 个具体的分配算法, 并对本文提出的二阶段分配方法与 2 种关于 VW 客户分配的算法进行性能比较。

2 背景介绍

文献[1-2]基于 3D 开发的虚拟现实技术, 给出实现该技术的场景创建、动画设计和交互设计。文献[3]对 VW 中的学习环境、学习动机、学习心理和行为特点进行详细分析, 给教育教学工作带来新启示。文献[4-7]针对 VW 中的通信环境、信息交换以及多人分组互动技术进行研究。总之, VW 已经应用于许多领域, 如协同设计、电子学习和多人游戏等。

对于实现 VW 应用的技术构架来说, 通常有 3 种主要结构, 即对等网络结构、中心服务器结构和分布式服务器结构。中心服务器结构对于诸如安全性、私密性和业务管理等功能易于实现, 其缺陷在于高时延、可扩展性差和单点故障; 对等网络结构尽管有较低的时延, 但需要很高的带宽, 所以这种结构存在严重的带宽接入问题, 特别是对于上行链路而言。它的另一个不利之处是安全性和匿名性降低, 事实上, 这种结构需要提供游戏玩家的 IP 地址给同一组所有其他玩家, 所以这种结构极少采用。

目前, 关于 VW 中化身动态特性的区域-服务器分配算法的研究并不多见; 文献[8]关于 VW 的静态服务器选择机制, 给出基于客户之间的通信时延和通信频率的服务器选择方法, 但没有讨论当化身移动时服务器选择必须变化的动态情形如何处理; 文献[9]提出的客户分配方法的目标是使 VW 的通信时延上限最小化, 而不能使整个系统最优化; 文献[10]针对协同式虚拟环境, 提出基于任务聚类的虚拟环境固定分区算法, 把参与同一类任务的用户划分给同一个区域服务器进行管理。尽管该算法取得了基于任务聚类的固定分区算法的最优分区效果和平均执行效率, 但无法使用户根据兴趣域进行分组来降低网络通信量, 而且这种基于任务聚类的虚拟环境固定分区算法需要较高的图形压缩技术和网络传输技术支持; 文献[11]提出一种基于交换操作和再分配的算法, 在服务器和所管理的区域之间不断执行交换操作执行区域-服务器的再分配, 这种方法虽然提高了 VW 中用户的交互性, 但明显增加了通信时延, 而且执行交换操作的条件、何时执行交

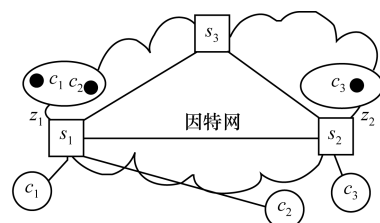
换和再分配都要根据 VW 中多客户运行情况而定, 无疑增加了算法执行成本, 降低了执行效率。

为了解决大规模 VW 中成百上千客户的同时互动, 通常把虚拟世界在空间上划分成若干个不同的区域, 每个区域仅由一个服务器管理, 一个客户只与同一个区域里的其他客户互动, 且可以移动到其他区域。由于一个服务器仅需要处理一个或多个区域而不是整个虚拟世界, 这样系统就是可扩展的, 本文把这种划分方法称为基于区域的方法; 由于 VW 中的客户在地理上呈分布式, 而且互联网也有多种特点, 一个区域中的客户可能对于管理这个区域的服务器有很大的网络时延, 这样, VW 的互动性就可能受到极大影响。因此, 需要一种有效的机制来把实时参与的客户分配给服务器, 以提高 VW 的互动性, 这就是客户分配问题。

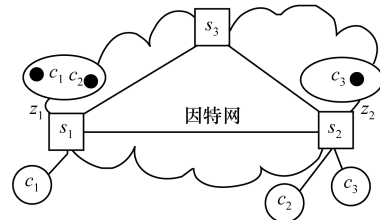
3 虚拟世界结构和客户分配方法

3.1 结构和定义

本文主要针对地理上呈分布式的服务器和基于区域划分的 VW 结构, 如图 1 所示。目前许多内容分发网络都采用这种结构来支持对普通 Web 站点庞大的访问需求。VW 研发人员可以独立地研发自己的服务器结构, 或者从第三方服务器-网络提供商租用服务器和网络资源, 以避免过高的开发和管理成本^[12]。



(a) 客户目标服务器和触点服务器相同的情形



(b) 客户目标服务器和触点服务器不同的情形

图 1 分布式服务器和基于区域划分的 VW 结构

在构建客户分配问题之前, 先定义下列记号和概念。

c_i : VW 中的一个客户。

$C = \{c_1, c_2, \dots, c_k\}$: 构成 VW 全部客户的集合。

z_i : VW 中的一个区域。

$Z = \{z_1, z_2, \dots, z_n\}$: 构成 VW 全部区域的集合。

s_i : VW 中的一个服务器。

$S = \{s_1, s_2, \dots, s_m\}$: 构成 VW 全部服务器的集合。

触点服务器: 一个客户的触点服务器是指与这个客户直接连接的服务器。客户仅发送输入给它们的触点服务器, 如果触点服务器正管理客户所在区域, 则触点服务器可以处理输入并发送更新给客户, 或者把客户输入转交给另一个管理客户所在区域的服务器, 如在图 1(a) 中, s_1 是 c_1 和 c_2 的触点服务器。

目标服务器: 一个客户的目标服务器是指管理这个客户所在区域的服务器。客户的输入可以转交给它的目标服务器, 如果目标服务器也是客户的触点服务器, 则目标服务器也可以把更新直接发送给客户, 或者通过客户的触点服务器间接发送。一个区域中的全部客户有相同的目标服务器, 但全部客户可以有不同的触点服务器。

如在图 1(a) 中, s_1 既是 c_1 和 c_2 的触点服务器, 又是它们的目标服务器。在图 1(b) 中, 把 c_2 调换到服务器 s_2 (但 c_2 的化身仍然在区域 z_1 中), c_1 和 c_2 的目标服务器仍然是 s_1 , c_1 的触点服务器是 s_1 , 但 c_2 的触点服务器现在是 s_2 , 来自 c_2 的输入由 s_2 转发给 s_1 , 由于 s_1 和 s_2 之间的网络连接配置良好几乎无阻塞, 因此 c_2 和 s_1 之间的通信可以有更小的时延, 但这将导致服务器 s_1 和 s_2 之间的内部通信消耗额外的资源。

R_{s_i} : 服务器 s_i 的资源消耗, 可以通过对 CPU 和网络带宽的使用测量得到。一般服务器的 CPU 不存在瓶颈, 故以网络带宽使用作为资源消耗计量。

$R_{c_i}^T$: 客户 c_i 在它的目标服务器上使用的网络带宽, $R_{c_i}^T > 0, \forall c_i$ 。

$R_{c_i}^C$: 客户 c_i 在它的触点服务器上使用的网络带宽。如果 c_i 的触点服务器和目标服务器是同一个, 则 $R_{c_i}^C = 0$, 否则 $R_{c_i}^C = 2R_{c_i}^T$, 因为 c_i 和它的目标服务器之间的全部通信都由它的触点服务器发送。

R_{z_i} : 区域 z_i 在它的目标服务器上使用的网络带宽, $R_{z_i} = \sum_{c_j \in z_i} R_{c_j}^T$ 。

C_{s_i} : 服务器 s_i 的资源容量。

$d_{c_i s_j}$: 客户 c_i 和服务器 s_j 之间的往返网络时延。

D : VW 的时延界限。时延界限表示一个客户和它的目标服务器之间的往返通信时延必需的上限, 以保证 VW 的互动性。对不同种类的 VW 来说, 有不同的时延界限要求, 如多人实时战略游戏通常需要 500 ms, 单人射击游戏需要 250 ms。一个客户和它的目标服务器之间的通信时延与这个

客户和它的目标服务器之间的网络时延是不同的, 通信时延是网络时延和在目标服务器上处理时延的总和, 由于本文假设服务器的 CPU 不存在瓶颈, 因此客户-服务器之间的通信时延可以由客户-服务器之间的网络时延决定, 以下的网络时延和通信时延含义相同。

对像 VW 这种互动应用来说, 通信时延是系统提供给客户最重要的 QoS 参数。在本文中, 如果客户和它的目标服务器之间的通信时延小于或大于时延界限, 表示这个客户具有 QoS 或不具有 QoS。

3.2 客户分配问题

客户分配问题就是如何把参与到 VW 中的客户分配给服务器, 以使具有 QoS 的客户数量最大化, 构建客户分配问题如下:

定义 1 客户分配问题

对 VW 中的每个客户 c_j 来说, 寻找它的目标服务器 s_k 和触点服务器 s_l , 以最大化:

$$|\{c_j \in z_i : d_{c_j s_k} = (d_{c_j s_l} + d_{s_l s_k}) \leq D, \forall z_i \in Z\}| \quad (1)$$

并满足:

$$R_{s_i} \leq C_{s_i}, \forall s_i \in S \quad (2)$$

其中, $|\cdot|$ 表示集合的基数。

在定义 1 中, 如果 s_l 和 s_k 为同一个服务器, 则 $d_{s_l s_k} = 0$ 。

3.3 二阶段客户分配方法

在初始分配阶段, 把区域分配给服务器, 即为每个客户寻找一个目标服务器。在优化分配阶段, 把每个客户分配给一个合适的服务器与它的目标服务器通信, 即为每个客户寻找一个触点服务器。

事实上, 初始分配和优化分配本身就是复杂的最优化问题, 为了得到 CAP 的满意解, 必须使这 2 个分配阶段的分配成本最优化。在初始分配阶段, 把一个区域 z_j 分配给一个服务器 s_i 的量化成本为:

$$C_{ij}^I = |\{c_k \in z_j : d_{c_k s_i} > D\}| \quad (3)$$

C_{ij}^I 是一个区域 z_j 中的客户数量, 但并不满足时延界限 D , 即不具有 QoS。因此当全部区域被分配时, 通过使总的成本最小化, 则 VW 中具有 QoS 的总的客户数量将最大化, 于是初始分配问题就可以构建如下:

定义 2 初始分配问题

令 $I = \{1, 2, \dots, m\}$, $J = \{1, 2, \dots, n\}$ 分别为系统中服务器和区域的指标集合, 对每个 $i \in I, j \in J$, 把区域 z_j 分配给服务器 s_i 的成本 C_{ij} 按式 (3) 定义, 寻找一个分配矩阵 $x = \{x_{ij}\}$ 且如果把区域 z_j 分配给服务

器 s_i , 则 $x_{ij} = 1$, 否则 $x_{ij} = 0$, 使得总成本最小化。

$$C^I(x) = \sum_{i=1}^m \sum_{j=1}^n C_{ij}^I x_{ij} \quad (4)$$

满足:

$$\sum_{j=1}^n R_{z_j} x_{ij} \leq C_{s_i}, \forall i \in I \quad (5)$$

$$\sum_{i=1}^m x_{ij} = 1, \forall j \in J \quad (6)$$

$$x_{ij} \in \{0, 1\}, \forall i \in I, \forall j \in J \quad (7)$$

在初始分配阶段后, 一些客户可能仍然不具有 QoS, 优化分配阶段将尝试进一步增加具有 QoS 的客户数量。式(8)则为为客户 c_j 选择服务器 s_k 为触点服务器的量化成本, 这里 s_i 为 c_j 的目标服务器。

$$C_{ij}^R = \begin{cases} (d_{c_j s_k} + d_{s_k s_i}) - D(d_{c_j s_k} + d_{s_k s_i}) > D \\ 0 & (d_{c_j s_k} + d_{s_k s_i}) \leq D \end{cases} \quad (8)$$

因此, 当全部客户被分配后, 通过使总的成本最小化, VW 中具有 QoS 的客户总数量将最大化, 则优化分配问题可构建如下:

定义 3 优化分配问题

令 $I = \{1, 2, \dots, m\}$, $J = \{1, 2, \dots, n\}$ 分别为系统中服务器和客户的指标集合, 对每个 $i \in I, j \in J$, 设选择服务器 s_i 作为客户 c_j 的触点服务器的成本为 C_{ijR} , 寻找一个分配矩阵 $x = \{x_{ij}\}$ 且如果客户 c_j 把服务器 s_k 作为它的触点服务器, 则 $x_{ij} = 1$, 否则 $x_{ij} = 0$, 使得总成本最小化。

$$C^R(x) = \sum_{i=1}^m \sum_{j=1}^n C_{ij}^R x_{ij} \quad (9)$$

满足:

$$\sum_{j=1}^n R_{c_j}^C x_{ij} \leq (C_{s_i} - R_{s_i}), \forall i \in I \quad (10)$$

$$\sum_{i=1}^m x_{ij} = 1, \forall j \in J \quad (11)$$

$$x_{ij} \in \{0, 1\}, \forall i \in I, \forall j \in J \quad (12)$$

4 分配算法

为了求解客户分配问题, 提出用于初始分配和优化分配的具体算法。对于初始分配, 把区域分配给服务器, 即为客户决定目标服务器, 对此采用 2 个算法: 第 1 个算法是随机分配区域; 第 2 个算法是贪婪启发式分配区域, 以使系统中不具有 QoS 的客户数量最小化。在客户的目标服务器确定后, 即优化分配阶段, 为每个客户寻找一个合适的触点服务器, 也采用 2 个算法, 第 1 个算法根据客户的虚拟位置把客户分配给触点服务器, 第 2 个算法也类似于求解初始分配的贪婪启发式算法^[11]; 把这些算法组合

起来, 从而得到求解 CAP 的算法, 这样, 对于求解 CAP, 总共有 4 个算法。

4.1 求解初始分配问题的算法

4.1.1 随机分配区域算法

在随机分配区域 (Random assignment of Zones, RanZ) 算法中, 把区域随机分配给已选择的未过载服务器。算法重复下列过程直至全部区域被分配: 首先选择具有最大客户数量的区域 z_j , 其次随机选择一个有足够容量的服务器 s_i 来管理 z_j , 即区域 z_j 中全部客户的目标服务器是 s_i 。

4.1.2 贪婪分配区域算法

由于 RanZ 在把区域分配给服务器时没考虑客户-服务器网络时延, 所得到的具有 QoS 的客户数量可能不理想, 也就是由式(4)定义的分配成本可能很高, 因此, 贪婪分配区域 (Greedy assignment of Zones, GreZ) 算法采用贪婪启发式使得系统中不具有 QoS 的客户总数量最小化。

令 $\mu_{ij} = -C_{ijR}$ (C_{ijR} 为区域 z_j 中不具有 QoS 的客户数量, 如果 z_j 被分配给 s_i) 为把区域 z_j 分配给服务器 s_i 的启发式期望值, μ_{ij} 越大, 成本 C_{ij} 就越小。算法考虑了全部未分配区域并挑选出区域 z_j , 在最大期望值 μ_{ij} 和次极大期望值 μ_{sj} 之间有最大差值 ρ_j , 把 z_j 分配给具有最大 μ_{ij} 值和足够资源容量的服务器 s_i , 实现这个分配过程的伪代码如下:

```

begin
  for each  $z_j \in Z$  do
    find  $\mu_{ij} = -C_{ij}^I, \forall s_i \in S$ ;
    sort the list  $L_j^M$  of values  $\mu_{ij}$  descendingly;
    find  $\rho_j = \max_{s \neq i} \mu_{sj} - \mu_{ij}$ ;
  end
  sort the list  $L^R$  of values  $\rho_j$  descendingly;
  while  $Z \neq \emptyset$  do
    pick zone  $z_j$  with highest  $\rho_j$ ;
    while  $L_j^M \neq \emptyset$  do
      pick the highest desirability  $\mu_{ij} \in L_j^M$ ;
      pick server  $s_i$ ;
      if  $R_{s_i} + R_{z_j} \leq C_{s_i}$  then
        assign  $z_j$  to  $s_i$ ;
         $R_{s_i} = R_{s_i} + R_{z_j}$ ;
        remove  $z_j$  from  $Z$ ;
        break;
      end
      remove  $\mu_{ij}$  from  $L_j^M$ ;
    end
  end
end

```

4.2 求解优化分配问题的算法

4.2.1 基于虚拟位置的分配客户算法

基于虚拟位置的分配客户 (Virtual location based assignment of Clients, VirC) 算法采用最自然的方法把 VW 中的客户分配给服务器。当为 c_j 确定触点服务器时,算法仅考虑每个客户 c_j 的虚拟位置, c_j 将连接到管理 c_j 所在区域的同一个服务器上,即 c_j 的触点服务器与它的目标服务器相同。这样就不会导致服务器内部链接的通信成本,但并未考虑每个客户到它的目标服务器的网络时延,则具有 QoS 的客户数量与初始分配相比可能不会有提高。

4.2.2 贪婪分配客户算法

贪婪分配客户 (Greedy assignment of Clients, GreC) 算法在为每个客户选择触点服务器时,考虑了每个客户到它的目标服务器的网络时延。这个算法类似于求解初始分配问题的 GreZ 算法,2 个算法的主要区别在于采用的成本计量,GreC 采用的成本计量为式(8)。GreC 从考虑系统中每个客户到目标服务器的往返时延开始,如果客户 c_j 到它的目标服务器 s_i 的网络时延小于时延界限,则算法选择同一个服务器 s_i 作为 c_j 的触点服务器;否则,把 c_j 加入到序表 L^E 中,这个序表仅包括那些网络时延大于时延界限的客户,算法接下来尝试为 L^E 中的每个客户分配一个触点服务器,以增加系统中具有 QoS 的客户数量。实现这个分配过程的伪代码如下:

```
begin
  for each  $c_j \in C$  do
    get target server  $s_i$  of  $c_j$ ;
    if  $d_{c_j s_i} \leq D$  then
      select  $s_i$  as the contact server of  $c_j$ ;
    else
      add  $c_j$  to the list  $L^E$ ;
    end
  end
  for each  $c_j \in L^E$  do
    find  $\mu_{ij} = -C_{ij}^R, \forall s_i \in S$ ;
    sort the list  $L_j^M$  of values  $\mu_{ij}$  descendingly;
    find  $\rho_j = \max_{s_i \neq i} \mu_{sj} - \mu_{ij}$ ;
  end
  sort the list  $L^R$  of values  $\rho_j$  descendingly;
  while  $L^E \neq \emptyset$  do
    pick client  $c_j$  with the highest  $\rho_j$ ;
    while  $L_j^M \neq \emptyset$  do
      pick the highest desirability  $\mu_{ij} \in L_j^M$ ;
      pick server  $s_i$ ;
      if  $R_{s_i} + R_{c_j}^C \leq C_{s_i}$  then
        select  $s_i$  as the contact server of  $c_j$ ;
```

```
 $R_{s_i} = R_{s_i} + R_{c_j}^C$ 
    remove  $c_j$  from  $L^E$ ;
    break;
  end
  remove  $\mu_{ij}$  from  $L_j^M$ ;
end
end
end
```

把上述求解初始分配问题和优化分配问题的算法进行组合,就得到总共 4 个不同的求解 CAP 的二阶段算法,即 RanZ-VirC, RanZ-GreC, GreZ-VirC 和 GreZ-GreC。

5 仿真建模与结果分析

5.1 仿真模型

分配算法的输入数据包括客户-服务器和服务-服务器的往返网络时延,以及每个客户对服务器的资源需求。网络时延采用可扩展网络测量工具 King^[13] 得到,King 用递归 DNS 询问来精确估算任意 2 个网络终端主机之间的往返网络时延。

把每个客户对服务器的资源需求称为带宽需求,因此,预先估算出一个区域中每个客户的带宽需求^[14]。为了估算带宽需求,每个客户的输入发送频率(帧频)设置为每秒 25 个报文,每个输入或更新的大小为 100 Byte,互动要求即 VW 时延界限 D 设置为 250 ms。

仿真采用拓扑生成器 BRITE Internet 来生成类似于因特网层级拓扑的虚构拓扑。拓扑由 500 个节点构成,共有 20 个自治域,客户和服务器的位置在 500 个节点中随机选取,任意 2 个节点间的最大往返时延设置为 500 ms。

仿真中采用 2 个指标评价算法的性能,一是把系统中具有 QoS 的客户百分比作为系统交互性的度量,记为 $pQoS$;二是把服务器资源消耗作为算法成本的计量,记为 R ,仿真给出的结果是对 50 个运行求平均值。

为了验证和比较本文算法的性能,在仿真中,一方面对算法在不同 VW 结构以及 VW 动态变化情况下的性能进行验证分析;另一方面,也对算法和目前广泛采用的 2 种分配算法^[10-11] 在通信时延和资源消耗性能方面进行仿真比较分析。

5.2 结果分析

5.2.1 不同 VW 结构对算法性能的影响

一个特定的 VW 结构由服务器数量、区域数量、客户数量和系统总的资源容量决定,用记号服务器

数量-区域数量-客户数量-容量来表示一个 VW 结构,如 20s-80z-1000c-500 cp 表示这个 VW 有 20 个服务器、80 个区域、1 000 个客户和总共 500 Mb/s 服

务器带宽。

相关性参数 δ 取 0.5 时,对 3 种不同 VW 结构进行仿真得到的 $pQoS$ 和 R 值如表 1 所示。

表 1 不同 VW 结构的 $pQoS$ 和 R

算法	5s-15z-200c-100cp		10s-30z-400c-200cp		20s-80z-1000c-500cp	
	$pQoS$	R	$pQoS$	R	$pQoS$	R
RanZ-VirC	0.57	0.60	0.57	0.61	0.61	0.58
RanZ-GreC	0.66	0.77	0.69	0.84	0.75	0.88
GreZ-VirC	0.79	0.60	0.83	0.61	0.89	0.58
GreZ-GreC	0.82	0.66	0.88	0.69	0.94	0.66

从表 1 可见,在 4 个二阶段算法中, GreZ-GreC 有最好的 $pQoS$ 性能,尤其对大规模 VW 结构来说, GreZ-GreC 算法是最优的;还可看到,在初始分配阶段考虑了网络时延的 2 种算法,即 GreZ-VirC 和 GreZ-GreC,比另外 2 种在初始分配阶段不考虑网络时延的算法能提供更好的交互性,这说明贪婪区域分配算法 GreZ 更有效。

从表 1 还可看到, RanZ-VirC 和 GreZ-VirC 有最低的资源消耗,因为在这 2 种算法中,每个客户 c_i 的目标服务器和触点服务器是相同的,这样就不会导致额外的带宽需求 $R_{c_i}^C$;考虑到 VW 更重要的问题是交互性而非带宽,所以 GreZ-GreC 算法在所有算法中是最好的选择,如果带宽需求变得更加重要,则 GreZ-VirC 算法也是一个好的选择。

5.2.2 算法对 VW 动态变化的适应性

VW 的动态性是指在虚拟世界互动过程中,客户可能从一个区域移动到另一个区域,新的客户可能加入虚拟世界,而现有的客户也可能离开虚拟世界,获得分配的客户在一段时间之后可能又变得不理想,这样,就必须考虑算法对 VW 动态变化的适应性。

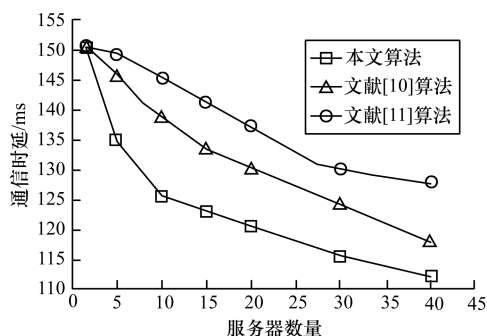
仿真时,首先得到 VW 结构为 20s-80z-1000c-500cp 时的客户分配。为了仿真 VW 的动态特性,让 200 个新客户随机加入虚拟世界,200 个现有客户随机离开虚拟世界,200 个客户随机移动到另一个区域,仿真结果如表 2 所示。“变化前”列数据表示让客户加入、离开和移动前全部算法的 $pQoS$,“变化后”列数据表示刚好让客户加入、离开和移动后,全部算法仍执行原来的 VW 结构时的 $pQoS$,可见,除 RanZ-VirC 外,其他 3 种算法的交互性都降低了,因此必须针对客户加入、离开和移动后新的 VW 结构,重新执行算法来获得更好的性能,所以“重新执行”列数据表示针对新的 VW 结构重新执行全部算法后的性能,可见,本文算法能够很好地适应 VW 的动态特性。

表 2 VW 动态变化时的 $pQoS$

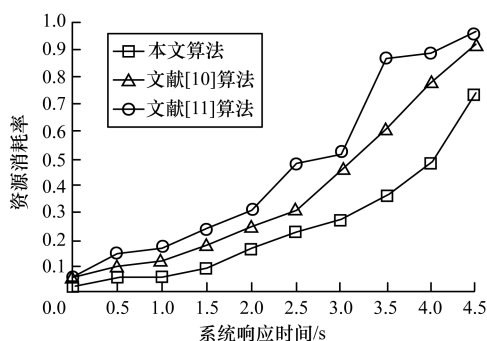
算法	变化前	变化后	重新执行
Ranz-Virc	0.58	0.58	0.58
Ranz-Grec	0.72	0.68	0.70
GreZ-Virc	0.82	0.79	0.81
GreZ-Grec	0.89	0.82	0.89

5.2.3 不同算法的性能仿真比较分析

为了对本文算法和文献[10-11]算法性能进行比较,仿真中,设定客户数量为 200, D 设置为 300 ms。对 3 种算法在不同服务器数量情况下的通信时延和在相同服务器数量情况下的资源消耗率进行仿真,结果如图 2(a)、图 2(b)所示。



(a)3种算法的通信时延比较



(b)3种算法的资源消耗率比较

图 2 3 种算法的性能比较

从图 2(a)可见,本文提出的分配算法的通信时延分别比文献[10-11]的算法平均减小了约 13 ms 和 18 ms,这对于大规模运行的 VW 应用来说,能够

极大地提高客户的实时交互性;而且对3种算法来说,服务器越多,效果越好。从图2(b)可见,本文提出的算法在资源消耗率上比文献[10-11]的算法平均分别降低了约12%和22%;从图2(b)还可看到,在资源达到饱和利用状态时,本文算法的资源消耗率大约在0.7左右,与表1中得到的仿真结果基本一致,同时,文献[11]算法的资源消耗率在某些状态时发生了突变,主要原因是该算法在这些状态时为了寻找到合适的交换条件导致交换操作频繁发生。总之,从图2可见,无论对系统的通信时延还是资源消耗率来说,本文算法都优于其他2种算法。

6 结束语

本文研究了呈分布式服务器结构的VW。在这种结构中,把一个庞大的虚拟世界划分成不同的区域在服务器之间分配负荷,把参与客户分配给服务器以提高VW的交互性。仿真结果表明,提出的算法能够很好地适应VW的动态变化特性,而且在网络平均通信时延和资源消耗方面都优于目前广泛采用的2种算法。

由于在虚拟世界互动过程中,客户可能从一个区域移动到另一个区域,新的客户可能加入虚拟世界,现有的客户也可能离开虚拟世界,获得分配的客户在一段时间之后又可能变得不理想,因此VW是一个动态复杂的系统,下一步工作将继续研究有关VW的热点问题。

参考文献

- [1] 陈宏敏,卢宇,张杰,等. 基于Java3D的虚拟网络电力监控环境设计[J]. 计算机与网络,2008,34(19): 50-52.
- [2] 张乐. 网络3D虚拟技术在大学图书馆中的应用研究[J]. 图书与情报,2010,(6):76-78.
- [3] 樊晓红. 虚拟世界的学习行为研究——教育研究的一个新领域[J]. 电化教育研究,2010,(11):24-27.
- [4] Bosch-Sijtsema P M, Sivunen A. Professional Virtual Worlds Supporting Computer-mediated Communication, Collaboration and Learning in Geographically Distributed Contexts [J]. IEEE Transactions on Professional Communication,2013,56(2):160-175.
- [5] Brendel D, Williams R. Methods of Effective Information Exchange in the Virtual World [C]//Proceedings of International Conference on Collaboration Technologies and Systems. Washington D. C., USA: IEEE Press,2013:553-560.
- [6] Mauve M, Fischer S, Widmer J. A Generic Proxy System for Networked Computer Games [C]//Proceedings of the 1st Workshop on Network and System Support for Games. New York, USA: ACM Press,2002:25-28.
- [7] Almashor M, Khalil I. Load-balancing Properties of 3D Voronoi Diagrams in Peer-to-Peer Virtual Environments [C]//Proceedings of the 16th International Conference on Parallel and Distributed Systems. Washington D. C., USA: IEEE Press,2010:839-844.
- [8] Fujikawa K, Hori M, Shimojo S, et al. A Server Selection Method Based on Communication Delay and Communication Frequency Among Users for Networked Virtual Environments [C]//Proceedings of the 7th Asian Computing Science Conference on Advances in Computing Science. Berlin, Germany: Springer,2002:125-139.
- [9] Lee K W, Ko B J, Calo S. Adaptive Server Selection for Large Scale Interactive Online Games [J]. International Journal of Computer and Telecommunications Networking, 2005,49(1):84-102.
- [10] 胡小梅. 协同虚拟环境系统的可伸缩性研究 [D]. 西安:西北工业大学,2007.
- [11] Ta D N B, Zhou Shuiping, Shen Haifeng. Greedy Algorithms for Client Assignment in Large-scale Distributed Virtual Environments [C]//Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation. Washington D. C., USA: IEEE Press,2006:103-110.
- [12] Duan Zhenhai, Zhang Zhili, Hou Yiwei. Service Overlay Networks: SLAs, QoS and Bandwidth Provisioning [J]. IEEE Transaction on Networking, 2003, 11 (6): 870-883.
- [13] Gummadu K P, Saroiu S, Gribble S D. King: Estimating Latency Between Arbitrary Internet End Hosts [C]//Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement. New York, USA: ACM Press, 2002:5-18.
- [14] Pellegrino J D, Dovrolis C. Bandwidth Requirement and State Consistency in Three Multiplayer Game Architectures [C]//Proceedings of the 2nd Workshop on Network and System Support for Games. New York, USA: ACM Press,2003:52-59.

编辑 顾逸斐