

协同虚拟环境中的并发控制机制

符云清, 全文君

(重庆大学计算机学院, 重庆 400044)

摘 要: 针对协同虚拟环境(CVE)并发控制在一致性、响应性和并发性之间存在的矛盾, 提出一种基于客户端的并发控制机制。该机制通过在客户端保存共享物体的状态信息, 使系统的一致性得到提高, 利用基于对象属性的并发控制思想, 提高系统的响应性。实验表明, 该并发控制机制能使 CVE 系统在一致性、响应性和并发性等方面达到较好的平衡。

关键词: 协同虚拟环境; 并发控制; 一致性; 响应性; 属性

Concurrency Control Mechanism in Collaborative Virtual Environment

FU Yun-qing, QUAN Wen-jun

(College of Computer Science, Chongqing University, Chongqing 400044)

【Abstract】 Aiming at the contradiction between consistency, responsiveness and concurrency in concurrency control of Collaborative Virtual Environment(CVE), a client-based concurrency control mechanism is proposed in this paper. The consistency is guaranteed and improved through the state information of sharing objects which is preserved in client. Attribute-based concurrency control is introduced to promote the responsiveness of CVE system. Experiment shows that the consistency, responsiveness and concurrency of CVE system can be well balanced by the control mechanism.

【Key words】 Collaborative Virtual Environment(CVE); concurrency control; consistency; responsiveness; attribute

1 概述

协同虚拟环境(Collaborative Virtual Environment, CVE)是一个计算机支持的、分布式的虚拟空间或虚拟场景的集合。在这个环境中, 人与人之间、人与虚拟物体及智能实体之间可以相互交流。协同虚拟环境具有丰富的表现形式, 可以是 3D 的图形空间或 2.5D、2D 的场景, 也可以是基于文本的环境。协同虚拟环境系统采用的输入输出设备不仅局限于桌面应用设备, 还可以是移动设备、穿戴式设备、公共场合的设备等^[1]。CVE 系统旨在提供共享的、极具真实感的虚拟空间, 使人们如同在现实世界中一样更自然、协调地进行交互和协作。借助该技术, 人们可以真正进入计算机, 通过网络到达任何地方, 甚至进入那些通过仿真创造出来的虚拟世界, 与任何人进行逼真的远程会面, 高效地完成协同工作^[2]。

协同虚拟环境允许多个在地理位置分散的用户同时对一个共享物体进行操作, 这类操作称为并发操作。当 CVE 系统中出现并发操作时, 必须采用一定的控制机制进行协调, 以维护虚拟世界的一致性。本文介绍悲观并发控制和乐观并发控制, 从它们的不足之处入手, 提出一种新的并发控制机制。

2 协同虚拟环境中的并发控制

2.1 设计原则

并发控制的实质是为 CVE 系统提供无冲突的并发操作序列。对于一个 CVE 系统, 一致性、响应性和并发性是实现有效协同的有力保障, 但三者 CVE 的并发行为空间中又是相互冲突的^[3]。

尽管响应性和并发性都是 CVE 系统中非常重要的需求, 但一致性是 CVE 系统中一切协同的基础和最终目标。一个不

一致的虚拟环境不可能实现正确而有效的协作, 因此, 一致性是并发控制策略的首要条件, 牺牲一致性来获取最优响应性和并发性并发控制策略并不可取。但获取最优响应性和并发性而不影响一致性是不可能的, 因此, 设计 CVE 系统并发控制机制的目标是在保证一致性的前提下, 尽可能地优化响应性和改善并发性。

综上所述, 寻找 CVE 系统的一致性、响应性和并发性三者之间的均衡点, 并最终实现 CVE 系统无冲突的并发操作是设计 CVE 系统并发控制机制的关键。

2.2 常用并发控制机制

针对 CVE 系统中的并发操作, 国内外研究人员提出了许多控制算法。根据用户的操作能否得到立即响应, 并发控制机制通常分为悲观并发控制和乐观并发控制两大类^[4-9]。

(1) 悲观并发控制机制以一致性优先, 任一时刻只允许 1 个用户操作某一共享物体, 即当 1 个用户操作共享物体时, 该物体被锁定, 而同一时刻其他所有试图对该物体进行操作的用户必须等待该物体解除锁定后才能对其进行操作。

(2) 乐观并发控制机制以用户响应性优先, 认为并发操作发生冲突的可能性很小, 因此, 在不进行冲突检测的情况下就直接对某共享物体进行操作, 即任何一个用户在获得共享物体的操作令牌之前就可以对该物体直接进行操作, 同时发

作者简介: 符云清(1969 -), 男, 教授、博士, 主研方向: 计算机网络体系结构, 多媒体通信, CSCW, CVE, 现代远程教育; 全文君, 硕士研究生

收稿日期: 2008-07-05 **E-mail:** yqfu@cqu.edu.cn

送物体的操作令牌申请，一旦该申请没有被允许，该用户必须对该操作进行撤销。

3 基于客户端的并发控制模型

3.1 基本思想

本文在现有协同虚拟环境中的并发控制基础上，提出一种基于客户端的并发控制模型，基本思想如下：

(1)基于客户端的并发控制。该模型是针对悲观和乐观并发控制机制的缺点设计的。其基本思想是在客户端保存共享物体的状态信息(共享物体被用户占有的情况)。用户每次操作共享物体之前都要检查该共享物体的状态信息：1)如果没有被其他用户占用，则直接操作，同时向服务器发送该共享物体的令牌请求；2)如果有用户占用，则不操作。这样既在一定程度上避免了乐观并发控制机制的视图回撤，又比悲观并发控制机制提高了响应性。

(2)基于对象属性的并发控制。多个用户同时对同一个对象的操作可能产生冲突，也可能不冲突。比如一个用户改变对象 O 的颜色，同时另一个用户改变对象 O 的位置，虽然两者都同时对同一个对象进行操作，但操作本身并不冲突。因此，针对同一对象不同属性的并发操作不会产生冲突。

(3)操作分类。CVE 系统主要包含两大类操作：1)共享物体的观察类操作；2)共享物体的属性改变类操作。其中，对共享物体的观察类操作不会产生冲突。因此，并发控制主要是针对共享物体的属性改变类操作。

基于客户端的并发控制模型结合了基于对象属性的并发控制和操作分类的思想。先将基于对象属性的并发控制的思想引入到的并发控制模型中，这样对共享物体的操作就细化为对共享物体属性的操作。因为对共享物体不同属性的操作是彼此独立的，所以对共享物体的每个属性都采取基于客户端的并发控制机制，这样可以大大提高并发性。

新并发控制模型的基本流程如图 1 所示。

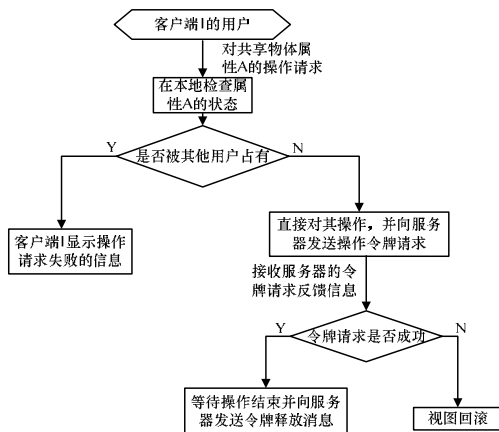


图 1 新并发控制模型的基本流程

(1)当有多个用户对共享物体属性 A 进行并发操作时，服务器根据先来先服务的原则把操作令牌分配给消息最先到达的用户。

(2)当用户 U 拥有了共享物体属性 A 的操作令牌时，服务器会每隔一定的时间间隔给用户 U 发送探测消息，用户 U 收到该消息则立即给服务器回复，表示自己还在线。如果服务器端在连续发送 3 次探测消息都没收到回复的话，便默认地认为用户 U 已经掉线，于是将用户 U 拥有的令牌释放。

3.2 操作步骤

(1)当用户 U 要对某个共享物体 O 的属性 A 进行操作时，

首先需要在本地判断共享物体 O 的属性 A 的当前状态，其状态有以下几种：1)没有用户占用；2)该用户自己占用；3)其他用户占有。

(2)如果共享物体 O 的属性 A 的当前状态为没有用户占用或被用户 U 自己占用，则用户 U 在享物体 O 的属性 A 上的操作立即被执行，并且给服务器发送对操作请求。

(3)当服务器接收到对用户 U 对共享物体 O 的属性 A 的操作请求时，根据共享物体 O 的属性 A 的当前状态进行下列处理：

1)如果没有用户占用共享物体 O 的属性 A ，则把对物体 O 的属性 A 的操作令牌分配给用户 U 。

2)如果共享物体 O 的属性 A 被用户 U 自己占用，则服务器向用户 U 发送已获取令牌的信息。

3)如果共享物体 O 的属性 A 被其他用户占用，则服务器向用户 U 发送请求令牌失败的信息。

(4)用户 U 接收到服务器的响应信息后，若请求令牌失败的信息，则立即从服务器重新更新共享物体 O 的属性 A 信息，即撤销用户 U 的操作。

3.3 优势分析

本文提出的新并发控制机制以保证最终一致性、优化响应性和并发性为目标，同时找到了 CVE 系统一致性、并发性、响应性三者各自的平衡点，该并发控制机制具有以下优点：

(1)全面保证了 CVE 系统的一致性。

(2)通过客户端的控制在一定程度上优化了 CVE 系统的并发性。

(3)基于属性的思想有效地提高了系统响应性。

(4)简单且易于实现。

3.4 实验分析

本实验所用的并发控制模型采用了 Java 的 Socket 通信技术，运用多线程和灵活的消息机制。它分为服务器端和客户端。服务器端程序用来启动并发控制服务，处理客户端的各种请求。客户端程序负责向服务器发出请求，请求预处理、更新数据等。本系统实施简单、易于维护，具有较强的集成性、可扩展性和灵活性。实验数据和结果如图 2 所示。

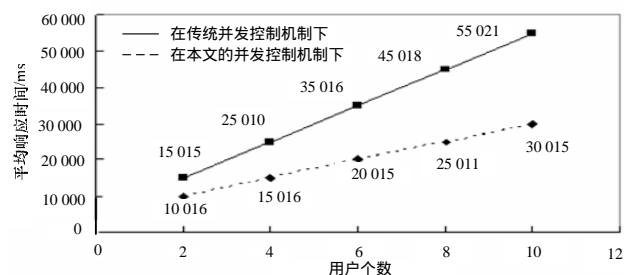


图 2 用户个数和平均响应时间的关系

在图 2 中，实线表示在传统并发控制机制下用户个数和平均响应时间的关系。用户针对某一共享物体的并发操作，用户对共享物体的操作时间为 10 000 ms。虚线表示本文提出的并发控制机制下用户个数和平均响应时间的关系。50%的用户对某一共享物体的属性 A 并发操作，50%的用户对某一共享物体的属性 B 并发操作，用户对共享物体属性的操作时间为 10 000 ms。从图 2 可以看出，本文提出的并发控制机制在保证一致性的前提下，较好地加快了并发操作的平均响应时间，即间接地改善了用户对共享物体操作的并发性，从而使

(下转第 55 页)