

基于多专业领域的团队协作副本一致性维护模型^{*})

姚一成 顾宁

(复旦大学计算机与信息技术系 上海 200433)

摘要 本文针对原有协同系统在解决多用户群协同问题中的缺陷以及原有解决方案的弊端,以 Chord 定位查询^[1]和多版本图形设计协同一致性算法^[2]为基础并改造本地副本策略的通用协同架构,解决了工程设计中多专业领域的团队协作副本一致性问题。

关键词 多领域团队间协同,分布式存储,图形设计协同

Group Collaboration Consistency Model Based on Multi-speciality

YAO Yi-Cheng GU Ning

(Computer Information Technology Department, Fudan University, Shanghai 200433)

Abstract The paper aims that traditional collaboration system cannot apply to multi-group collaboration, whose original solution is not very perfect. It is based on Chord orientation and search^[1], multi-version graph design collaboration consistency algorithm^[2] and rebuilds the tradition collaboration architecture to solve the copy consistency problem in the multi-specialty group collaboration.

Keywords Multi-specialty group collaboration, Distributed storage, Graph design collaboration

计算机支持的协同工程设计是近几年新兴的研究方向。协同副本是副本在分布式对等网络中的一种应用形式。分布式领域内的副本研究通常不考虑用户之间互相感知的问题,仅仅基于文档版本控制来实现副本异步一致性;而实时协同副本的研究基于用户操作,保证用户在协同过程能最大程度获取一致的副本状态。通用的协同模式基于本地副本策略,每个用户拥有设计文档的副本拷贝,实时协同建立在各地副本同步一致性之上。用户所作的操作会作用于本地副本,也会发送给远程用户并通过协同一致性维护作用于其他用户的副本,保证所有用户副本整体一致性,使所有用户及时获取相同的设计文档状态,从而实现实时的协同设计。

然而随着参与协同的用户数量以及协同副本规模的不断增加,通用协同系统中不考虑协同过程中设计者的实际需求的弊端逐渐显现,主要体现在以下两个方面:

一、通用协同对所有用户一视同仁,所有用户需要接收和处理所有操作。而实际应用中由于每个用户关心的操作往往集中在自己的设计的模块或区域。

二、通用的协同不考虑用户的角色,将所有用户看作同等对待。

本文提出简便合理的解决方案力求对于所有协同应用领域适用。第 2 部分给出更为详细的研究背景和相关工作;第 3 部分提出基于 Chord 团队内协同分工模式;第 4 部分给出改进后的多版本图形设计副本一致性控制算法;第 5 部分提出基于多领域团队的协同副本一致性控制模型;第 6 部分给出实验分析;最后总结并提出了未来工作计划。

1 研究背景及相关工作

工程协同往往牵涉到许多的知识领域模块^[4]。如建筑设计领域有结构设计,管道线路设计,室内设计等等;芯片设计

领域也可以分为框架设计,内部线路设计等等。大型的工程对于每一个模块都需要一个设计团队来共同完成,这些团队之间又存在同步协同关系。本文称这种协同为团队协作。

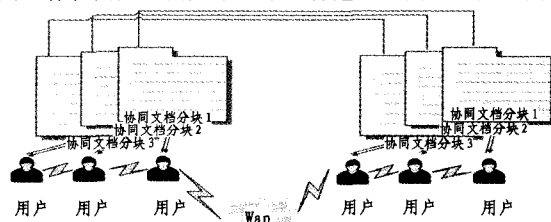


图 1 团队协作模式

图 1 为一个简单的应用场景示例。一个领域的专家必须保证自己的设计不会和其它领域专家的设计产生冲突。有别于通用的协同冲突定义,如果将原来的副本一致性需求称之为时间一致性^[4],那么这种一致性的需求可以称之为空间一致性^[4]。

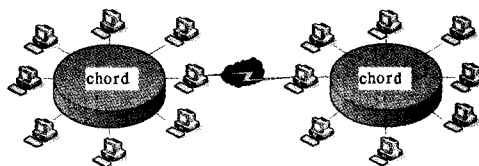


图 2 协同模式的 Chord 架构

原有协同模型强制所有用户接收所有的操作而不是自己感兴趣的操作之外,也不支持团队间和团队内的双层协同模式并且没有考虑副本的空间一致性的维护。因此需要针对新的应用需求设计新的模型架构。

新架构的设计需求就是要在满足时间和空间一致性的基

^{*} 基金项目:此研究受国家自然科学基金(90612008)、国家 973 课题(2005CB321905)和上海市科委基础研究计划重点项目(编号:05JC14006)资助。姚一成 硕士研究生,研究领域:实时协同,分布式计算;顾宁 教授,博导,研究领域:计算机协同,Web services,网格计算,数据流。

础上,实现多领域团队间实时协同并保证处理信息的最小化。

分布式资源的存储方式的是团队内副本组织的理想实现模式,分布式资源查询定位方法能满足对于特定副本资源的定位要求。因此新架构采用分布式资源存储和查询技术作为存储通讯基础。

Chord^[1]能较好地适用于本文提出的架构。图2是Chord实现的简单通信模型。而作为协同副本一致性维护的底层,基于本地副本策略的一致性维护技术在协同领域也已经发展地相当成熟。由于工程设计领域更多地涉及到图形设计领域,因此本文改造GRACE^[2]的图形一致性算法作为底层核心。

2 基于Chord的团队内协同分工

用Chord将团队内所有节点的IP地址散列得到标识符,按大小排列组成了一个环形分布结构。而将散列后关键字为k的副本资源保存在它的后继节点successor(k)中,successor(k)是从k开始顺时针方向距离最近的节点。新架构根据虚拟分割的副本关键字来确定团队内某成员副本的存放位置。

在Chord内一般采用在每个节点维护一张最多m个表项的指针表(finger table)。协同副本定位算法searchsite(K)详见相关论文^[1]。当节点n收到关键字为K的查询请求,寻找Finger[i]=V。如有将S[j](Finger[j]=V)返回给节点n;如没有,在指针表内选最接近V的Finger[r],并将查询请求路由转发给S[r]对应节点;在对应节点重复以上步骤直到找到离V最近的标识符节点并返回。

对团队协同分工和通信有如下定义:

定义1(虚拟副本分割) 虚拟分配文档的某个部分作为工作范围,而实际上文档的副本仍然分配给所有用户。若虚拟分块为 $partition_1, \dots, partition_N$,必须满足 $\forall i, j \in \{1, \dots, N\}$ and $i \neq j, partition_i \cap partition_j = \Phi$ 。而且需要创建并维护一个查询结构Partition-table,定义querypartition(Obj, Partition-table):返回对象的partition号。

定义2(超级节点) 每个团队仍然固定节点作为协同工作的协调者,定义其为超级节点。它可以是团队内负责人所在节点,也可以是物理意义上性能最强的一个节点。

3 多版本图形一致性算法

GRACE^[2,3]提出了完善的基于对象的图形设计协同模型,改进它可以应用于团队协同模式。新系统的主要问题还是时间一致性。在时间一致性冲突得到解决之前应有效地保护各团队用户的操作意图。GRACE提出了多版本对象的解决方法^[2,3],引入一般化兼容操作组集概念,定义NCGS构造算法getNCGS($\{O_1 \dots O_n\}$): NCGS。

在NCGS之上可以定义多版本对象控制算法getNewObjectVersion,输入一个操作和其对应对象的NCGS,可以返回该对象的所有版本和新的NCGS。算法详见相关论文^[2]。

4 多领域团队协同一致性模型

基于多专业领域的团队协同副本一致性维护模型的设计有如下原则:

- 1)相关的操作必须发到群内所有对应站点并进行操作维护。
- 2)只需要将其他领域内产生的操作转发到某个领域内负责该操作对象的用户。
- 3)优先处理群内副本协同,保证每个用户对自己感兴趣的的操作信息的及时获取。

首先从整体上可以将协同过程划分为:协同准备期,工作期和结束期。

协同准备期可以分为如下几步:

- 1)调用createChord模块构建Chord环。
- 2)团队内部分工(虚拟文档分割),标准可以互不相同,调用partitionStorage模块。
- 3)超级节点调用initiateCollaboration模块初始化协同环境。

对应功能模块的细节实现如下:

功能模块1. creatChord:

- 1)统计团队内节点个数M,为节点集合分配一组私有标识 IP_1, \dots, IP_M ,节点标识符 $N_1, \dots, N_M = SHA1(IP_1, \dots, IP_M)$ 。

- 2)按节点标识符 N_i 的大小依次初始化所有节点的后继节点信息和Finger表。

功能模块2. partitionStorage:

- 1)遍历Chord环,查找出团队内的超级节点,每个节点记录该节点信息Super-node。

- 2)根据协商好的分工方案,对文件名为filename的文档做虚拟分割操作,得到partition号 $\{P_1, \dots, P_n\}$,构建Partition-table。

3) $\{N_0, \dots, N_n\} = SHA1(filename + \{P_1, \dots, P_n\})$

- 4)searchsite(N_0, \dots, N_n)找到所有对应节点标识符并得到对应私有地址 $\{pr-IP_1, \dots, pr-IP_n\}$,为每个分块的实际处理节点地址 $\{pu-IP_1, \dots, pu-IP_n\}$ 分配对应的私有地址,如节点 $pu-IP_i$ 不存在该文件副本则传输文件transferfile(filename, $pu-IP_i$)。

- 5)把Partition-table发送给团队内节点,每个节点记录自己处理的partition号 P_{local} ($local \in \{1, \dots, n\}$)。

- 6)完成上述步骤之后在超级节点处标识collaboration-ready状态表示团队已具备协同条件。

功能模块3. initiateCollaboration:

- 1)超级节点处于collaboration-ready状态后广播协同请求信息使每个团队都知道需要参加协同团队个数为N个。

- 2)向CHORD环内内部节点广播该节点地址。任意节点都应该有N-1个广播地址,当团队内的一个节点都得到N-1个广播地址后,标识site_ready向超级节点发出就绪消息。超级节点标识cluster_ready。

- 3)当超级节点标识cluster_ready,向其它团队的超级节点发送就绪消息。

- 4)当超级节点收到其它N-1个团队超级节点cluster_ready的时候,说明所有其它团队就绪,那它也就可以开始协同工作。

协同工作期中有2个功能模块:

- 1.当某个节点i产生操作O之后,首先查询该操作是否发送到其他群内用户,然后将该操作广播至其他N-1个超级节点:

功能模块4. exec&sendoperation(O):

$p = \text{querypartition}(O, \text{Obj}, \text{Partition-table})$

$N = \text{SHA1}(filename + p)$

send(O, searchsite(N))

broadcast(O, {Super-node_{1-N-1}})

- 2.对于节点来说还要接收并维护执行远程操作。但是对于超级节点和普通节点来说收到一个远程操作O之后的处理不同:

功能模块5. receive&excuteoperation:

```

if (node is super-node){
    N= querypartition(O, Obj, Partition-table)
    If Plocal = N (该操作由超级节点本身负责)
        getnewObjectVersion (Oi, NCGSi-1)
    else if (O 来自于自身的群)
        else (该操作由团队内其它节点负责)
            Ni=SHA1(filename+ N)
            send(O, searchsite(Ni))
    }else{
        If Plocal = querypartition(O, Obj, Partition-table)
            getnewObjectVersion (Oi, NCGSi-1)
        else
            return;
    }
}

```

协同结束期的工作可以由以下功能模块实现:

功能模块 6. endCollaboration;

1) 某个节点结束工作后,向超级节点发送退出工作消息 send(site_exit, super-node)。

2) 超级节点收到团队内所有其它节点 site_exit 消息并且本身也完工后,向其它超级节点发出团队退出协同工作消息。

3) 当超级节点收到所有其它超级节点退出协同工作消息并且自己所在团队也结束所有工作时,在本团队内收集所有的文档对象,进行 merge 得到完整的文档对象 filename。

5 架构性能分析及实验

新架构原来一个站点顺序发送的操作可以同时被多个站点发送;而需要在同一站点处理的远程操作也可以分布式地发送到许多站点并发处理,类似多线程机制。由于底层机制相同,可以屏蔽底层一致性而仅仅考虑两种协同模式之间的差异。实验实现了 CHORD 环构造和查询功能,并模拟 2 种协同的交互模式。假设协同参与站点平均 5 秒产生一个本地操作,而每个站点维护远程操作处理时间为 2 秒。理论上 N 个成员的团队效率是单个站点的 N 倍,实际效率要差一点。

可以发现其它一些特性:

1) 从图 4 可以看出,团队内成员数量对于团队协作是很重要的,参与成员的多少对于团队处理操作的总输出有着决定意义。

2) 从图 5 可以看出,在操作基本均匀分布的情况下,通用模式的站点与团队模式的站点操作负荷差不多。新系统在不断增加站点工作开销使整体工作效率得到了提高。

结论及未来工作展望 基于多专业领域的团队协同一致性模型架构是有别于通用协同模式的,它基于 Chord 和图形设计协同一致性维护算法,完善地兼顾领域内和跨领域协同,

对大规模工程协同应用有其借鉴意义。

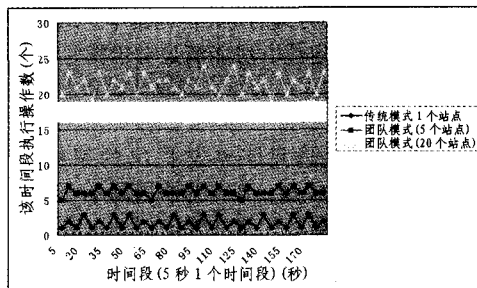


图 4 协同过程中操作输出对比(考虑通用模式,以及有 5 个和 20 个成员的团队模式)

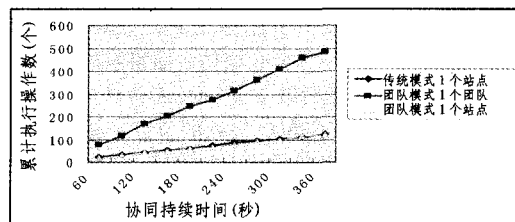


图 5 两种模式操作负荷量对比

在未来的工作中可以作如下改进:

1) 可以考虑提高架构的通用性或者提供更好的封装性,使其适用于多种协同。

2) 进一步研究在现有技术如何改进从而支持群内用户的动态加入和退出。

参考文献

- 1 Stolica I, Morris R, Karger D. Chord: A scalable peer-to-peer lookup service for Internet applications. In: Proc. Of the ACM SIGCOMM. San Diego, 2001
- 2 Sun C Z, Chen D. Consistency Maintenance in Real-Time Collaboration Graphics Editing Systems. ACM Transactions on Computer-Human Interaction, March 2002
- 3 Sun C Z, Chen D. A multi-version approach to conflict resolution in distributed groupware system. In: Proceedings of International Conference on Distributed Computing Systems
- 4 Gao Liping, Shao Bin, Gu Ning. Separating Data and View: Support View-wandering Between Different Trades. CSCWD'07

(上接第 279 页)

不具有时限的访问控制模型是不完备的。本文在 RBDM0 和 RDM2000 两种基于角色的委托模型基础上,提出了支持有效时间的委托授权模型,并对带时限的角色完全授权和部分授权以及增加有效时间对相应委托树结构的影响进行了讨论,并形式化地描述了他们的授权判断规则。我们还实现了基于 JAVA 的带时限的委托模型的原型,该原型充分体现了委托模型中的时效性。

在现实中对委托授权模型的时限性要求是多种多样的,进一步完善与时间相关的约束规则的定义,以及他们在委托模型中的应用是下一步研究的方向。

参考文献

- 1 Barka E, Sandhu R. Framework for Role-based Delegation Models. In: Proceedings of 16th Annual Computer Security Application Conference, 2000. 168~176
- 2 Zhang Xinwen, Oh Sejong, Sandhu RS. PBDM: A Flexible Delegation Model in RBAC. In: Proceedings of the 8th ACM Symposium on Access Control Models and Technologies, 2003. 149~157

- 1 Stolica I, Morris R, Karger D. Chord: A scalable peer-to-peer lookup service for Internet applications. In: Proc. Of the ACM SIGCOMM. San Diego, 2001
- 2 Sun C Z, Chen D. Consistency Maintenance in Real-Time Collaboration Graphics Editing Systems. ACM Transactions on Computer-Human Interaction, March 2002
- 3 Sun C Z, Chen D. A multi-version approach to conflict resolution in distributed groupware system. In: Proceedings of International Conference on Distributed Computing Systems
- 4 Gao Liping, Shao Bin, Gu Ning. Separating Data and View: Support View-wandering Between Different Trades. CSCWD'07
- 5 Zhang Longhua, Ahn G-J, Chu Bei-tseng. A Rule-based Framework for Role-based Delegation. In: Proceedings of the 6th ACM Symposium on Access Control Models and Technologies, 2001. 153~162
- 6 徐震,李澜,冯登国. 基于角色的受限委托模型. 软件学报, 2005, 16(5):970~978
- 7 孙波,赵庆波,孙玉芳. TRDM——具有时限的基于角色的转授权模型. 计算机研究与发展, 41(7):1104~1109
- 8 汤庸. 时态数据库导论. 北京:北京大学出版社
- 9 Zhang Xinwen, Oh Sejong, Sandhu R S. PBDM: A Flexible Delegation Model in RBAC. In: Proceedings of the 8th ACM Symposium on Access Control Models and Technologies, 2003. 149~157