ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.5, May 2008, pp.1125−1133
DOI: 10.3724/SP.J.1001.2008.01125

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

志愿计算模型形式化方法*

王 宇+, 王志坚

(河海大学 计算机信息及工程学院,江苏 南京 210098)

Formal Models and Methods on Volunteer Computing

WANG Yu+, WANG Zhi-Jian

(College of Computer and Information Engineering, Hohai University, Nanjing 210098, China)

+ Corresponding author: E-mail: won9805@hhu.edu.cn

Wang Y, Wang ZJ. Formal models and methods on volunteer computing. *Journal of Software*, 2008,19(5): 1125−1133. http://www.jos.org.cn/1000-9825/19/1125.htm

Abstract: This paper aims at revealing the essence of volunteer computing from the point of view of formal abstraction. At first, three elements of volunteer computing are analyzed and outlined. It included some concepts of component and computing resource in volunteer computing. One formal model and method concerning volunteers are introduced. Resource roles such as the basic elements of systems are identified in the abstract model, and modeling volunteer systems by basic elements. Relationships among elements and relationship changes are also characterized, based on set theory and operational reduction rules. And take XtremWeb as an example. These elements and relationships are specified in a formal manner. All of the results can help to build a firm foundation for research of formalisms for volunteer computing.

Key words: volunteer computing; component; formal method

摘 要: 旨在从形式化抽象的角度来认识移动计算的本质特点.分析了志愿计算平台的特征,提出并分析了志愿计算中的 3 种角色以及志愿计算中资源和构件的概念,介绍了一种关于志愿计算的形式化模型和方法.对系统中的基本元素和交互关系进行了形式化的描述,并通过集合理论和操作规约,又以志愿计算平台 XtremWeb 为例,描述了构件化的志愿计算形式模型和方法,为系统地研究志愿计算形式化理论打下了基础.

关键词: 志愿计算;构件;形式方法

中图法分类号: TP393 **文献标识码**: A

志愿计算[1]是全局计算的一种,通过使用很多由用户志愿提供的、数量庞大的空闲计算资源来提供无限计算能力,以此满足科学计算的需要.“志愿”体现在灵活、不可预知,同时还继承了分布式系统的一些基本问题,例如可扩展性、容错、异构、安全、编程模型等,在全球计算系统中都必须重新考虑[2].过去的研究项目包括 SETI@Home[3]、商业系统的 P2P 计算[4]、即将到来的网格计算[5].很多研究都是基于桌面网格的,例如

* Supported by the National Natural Science Foundation of China under Grant No.60573098 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z78 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2002CB312002 (国家重点基础研究发展计划(973))

Received 2007-06-10; Accepted 2007-10-15

Charlotte[6],Bayanihan[7],Javelin[8],GUCHA[9],XtremWeb[10],WebCom[11]等等.其中以BOINC(Berkeley open infrastructure for network computing)[12]应用最为广泛.这种计算模式的构成主要分为3个部分:任务请求的客户端(client)、提供计算资源的志愿者(worker)和进行资源控制和调度的志愿管理协调者.目前,在桌面网格中,能够承担这些资源的实体主要是指分布在各处、具有空闲时间的PC或工作站.词典对资源的普遍定义为:在某一特定情况下,任何能够被用来持续、补充、支撑和协助满足某种需要的来源构成.例如,文件、度量测试方法、CPU周期、内存、打印设备、可控设备、信息来源、网络服务等等.

定义(资源的发现). 这是指能够找到满足需求的一系列必要条件的能力.资源的发现包括3个部分:资源提供者、资源用户以及资源的发现管理和控制.这个概念与上述描述的3个部分相对应,即提供计算资源的志愿者、任务请求的客户端(client、进行资源控制和调度的志愿管理协调者.系统中某一实例根据功能或承担的任务不同,可以表现为多重角色.例如,资源提供者可以同时作为资源的使用者或协调相关的资源管理者,所以,从逻辑上看,某一实体是资源复合的结果.

1 形式化描述初步

志愿计算不仅要融合集群等大型、稳定的计算设备以及分布相对固定的PC作为计算资源,还应该融合那些数量庞大的、空闲的或各有特殊功能的计算资源,即普适计算资源,泛指普遍存在的PC和移动式、嵌入式、传感器等实体以及这些设备相对应的软件和计算环境.

目前,人们对志愿计算的组成构件还缺乏较为完整和深入的认识.“志愿”引起随意性带来的需求期待在新的分布式系统设计中得到满足.资源实体的自治与异质性,使得志愿计算系统设计很难具备实现所需求的特征.比如,数据、程序、构件、操作环境等多种资源实体都要承担3种角色,即成为资源的提供者、资源用户和资源的发现管理及控制;系统设计不得不面对如何描述这些资源实体,如何说明它们何时、怎样转换角色,以及由于不可知的随意性带来的系统重构的问题.这些因素使得志愿计算系统中的资源呈现出多样性与纷乱性,导致可用资源的种类减少和系统规模降低.一些研究工作表明,从抽象层次运用形式方法能够较好地认识某些系统的本质与特征,传统的形式模型有基于指称语义的,如文献[13]表示的actor模型;有基于代数语义的进程代数,如π演算[14]等;有基于状态转换语义的时序逻辑,如TLA[15]等.另外,还有移动Petrinet[16]通过允许场所命名以标记(token)形式出现在场所中来扩充场所转换,支持表示进程间通信通道配置的改变.

随着需求的变化,从传统方法中也有一些扩充,例如Ambient[17],Seal演算[18]是一种同步高阶分布进程演算,可看作是具有层次性位置移动特征和资源访问控制的.如π演算的设计目的是表示与推理分布式系统的安全与移动特性.它们有的是刻画体系结构下的数据移动,刻画移动计算的形式模型是对传统并发和分布式计算形式模型与方法的扩充和修改.还有的是描述智能计算实体间整体转换的,甚至可以刻画包含整个操作环境的硬件平台移动的一类系统[19].这些语言和演算为设计移动系统提供了规范说明和验证方法.由于数学基础、技术目标等因素的不同,还可以用于勾画不同的志愿资源构件模型.

2 资源构件系统模型

志愿计算系统被看作是由多个节点(node)组成的系统,每个节点从物理或逻辑上可以看作包含多个计算位置,它们是资源构件的计算环境,同时也是各种角色资源实体的管理与控制中心.构件之间的交互协作可以用远程通信方式,也可以是构件由于位置改变而进行的局部交互,构件间可以访问本地和远程资源.并且计算环境也可以改变,从而带动其子位置和依赖这个位置的构件一起移动.志愿计算采用志愿者空闲阶段的资源,用户忙闲程度根据职业满足某种规律,对于桌面计算机上的资源提供者——用户来说,这种规律的改变体现为计算位置发生了变更.

如果该模型是对广泛志愿计算系统模型的抽象,则可对应于如图1所示的树型,3种角色分别为计算资源的志愿提供者(resource provide,简称RP)、任务请求的客户端(client,简称CT)、进行资源控制和调度的志愿管理协调者(resource center,简称RC).作为角色的“候选”元素——构件,包括软构件,如数据、程序代码、病毒、进程、

镜像文件等;也包括硬构件,如 PC、PDA、传感器、手机等普适计算设备;甚至包括集群、网络等一些基础设施.而节点(node)是指构件和包纳这些构件的普适环境,这种环境更多地是指(物理或逻辑上的)空间,也可以是普适设备或网络中的物理位置,还可以是逻辑或概念上的,如逻辑网络节点、内存空间、进程空间、虚拟机、IDE、GUI 等.其他概念是构件和位置概念或关系的衍生.“变换”是指构件与位置各自的、相互之间绑定关系的变动.资源访问控制由“位置”界定出访问范围,再由构件身份标识出访问对象;系统配置的变化是由构件相对于“位置”来说发生创建与消亡等因素引起的.因此,该抽象模型实际上是关于构件、位置以及它们之间的关系及关系变化的模型.位置、构件、移动性、资源访问是志愿计算系统的关键概念,其中,位置与构件是核心概念.

图 1、图 2 和图 3 是更直观的等价表示.若将计算资源看作是节点,其中图 1 表示由 3 个角色组成的节点,其中的 *RP*,*CT*,*RC* 分别对应于各自的二元组,例如,节点用(位置 *p*,构件 *c*)表示,每一个(*p*,*c*)都可以看作是一个节点.同时,由这 3 个独立二元组(*p*,*c*)组合成的一个节点 *Nd*1 用虚线框表示,这个单节点也可以用(p_1,c_1)的形式表示.若与 *Nd*1 类似独立的节点共有 3 个,*Nd*2,*Nd*3,如图 2 所示,则它们组成虚线内节点 *Nd*′—(*p*′,*c*′),如图 2 所示,*Nd*′是一个规模更大的节点.

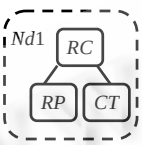


Fig.1 Element of nodes

图 1 元节点

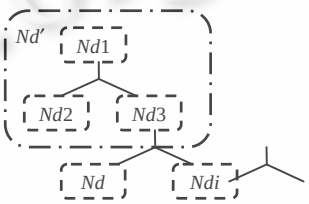
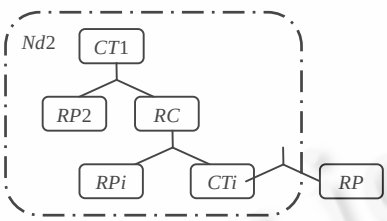


Fig.2 A part of the system

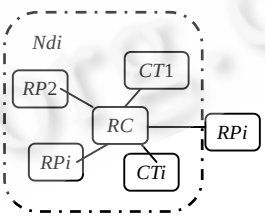
图 2 系统局部图

志愿者的随意性适合使用这种模型来描述,因为从逻辑上看,模型的整体和局部是具有自相似性的,如图 1 和图 2 所示.同理,*Nd*1 同样有可能由 3 个独立的节点构成.图 3 中点线图和实线分别是不同类节点的表现形式,相同线形中的部分表示相同类型的组成成分,便于采用类似于分形动力系统中处理 sierpinski 三角形生成的方法来描述志愿计算模型具有的相似性,把图 1 中的“三元”节点作为系统的吸引子,用于描述系统的变化过程.



(a) The framework of the volunteer computing nodes

(a) 志愿计算节点框架



(b) The equivalent framework chart of the nodes

(b) 节点等价框架图

Fig.3

图 3

图 3(a)表示节点 *Nd*2 由多个构件组成,构件成分与 *Nd*2 表现为被包含与包含关系.图 3(b)为构件的集中式,图 3(a)、图 3(b)只是构件的表达形式不同,都可以用于表达相同功能的节点.其中,*RC* 为资源管理调度者构件,与 *RPi* 为一对多的形式.图 2、图 3 都是包含了构件和位置的演算.这样表示是为了用自相似结构来描述志愿计算系统的组织结构,例如描述现阶段志愿计算网格中资源发现与调度的方法,研究内容主要包括资源集中式(OGSA,UDDI)、资源非集中式(P2P)、混合式(网状结构)等等.由于志愿者的组成和规律的随意性等特点,志愿计算模式下三者并存的状态与联系十分复杂,还有待细化全面的形式化描述.

3 组成关系

从构件与位置的角度来看,志愿计算中角色的转换是构件与位置各自的、有时又是它们相互之间绑定关系的变动,归纳起来一共有两对关系、4 种组合.以 RP-RC 转换为例,既可以理解为资源控制角色 *RC* 构件由位置界定出构件的访问达到的范围发生了变化,从功能上看,同一位置上构件的功能又发生了转换.以位置体现为 *P*.*RP* 变为 *P*.*RC* 为例,过程描述如下:在"位置"*P*,*RP* 消失的同时,新构件 *RC* 出现.例如,对同一个节点或计算设备,志愿资源提供者承担的功能角色由 *RP* 转换为 *RC*,现实中可能只是把原来占用 CPU 内存的 worker 程序挂起、删除或迁移,重新"唤醒"志愿控制程序(cordntor)[20],系统基本配置(OS、硬件设备、支撑软件等)不变.软构件 *RP* 通过"克隆(clone)"、传递、分发等方式被迁移到其他环境或计算设备中去.因此,在形式化的描述系统中,所有模型层次有助于了解普适环境中志愿计算的容错、调度、安全、交互等一系列事件和概念.在系统模型中,必须清楚地区分构件与位置这两个概念,并且标识出它们之间的各种关系和可能的关系变化.这样就能刻画出志愿计算的系统抽象模型以及志愿者角色模式中的其他概念,如节点间的交互、安全控制等.这些在系统设计与实现时非常重要,但在系统的模型层次来看,它们不仅是认识和刻画普适环境本质特征的关键元素,而且由于位置必须以某个构件与其他构件间的关系为参照物,不能单独抽象为一个实体,就像程序和算法不能不依赖与内存空间而单独存在一样.我们只需要考虑把 3 个角色与它们的位置做相互依赖关系,形成志愿者位置、消费者位置和协调者位置,以此来研究构件与构件之间的一系列关系.

构件之间的相互关系有:组合、引用、包含、父子关系(适用于处于同一位置.位置是指网络的同一个层次,例如放在同一个节点 *Nd* 当中的构件).

组合关系:构件 *RP*,*RC*,*CT* 为组合关系.

引用关系:*CT* 通过 *RC* 引用 *RP*.

包含关系:*Nd* 包含 *RP*,*RC*,*CT*.

层次关系:含父子、同辈.如图 2 所示,若 *Nd*1 为资源控制节点,*Nd*2 为资源提供节点,则 *Nd*1.*RC* 与 *Nd*2.*RP* 为父子关系;*Nd*1,*Nd*2 为同辈关系;*Nd*1.*RC* 与 *Nd*2.*RC* 为组合关系;*Nd*1.*CT* 与 *Nd*2.*RP* 为引用关系.

4 构件组成范性

志愿计算模式的本质是构件随时间序列的演变充当不同角色的生命周期.模型中的实体有两个刻面:软构件的实体是包括了数据、代码和状态的封装;另一个刻面是系统配置实体间的结构关系,也就是角色和位置的封装.在模型中,构件可充当的实体有代码和数据,包含计算代码、数据状态或执行状态和志愿计算环境,计算环境是一个复杂的概念,包含了志愿者的属性和能力.

(1) 迁移——数据(信息和代码)在 3 个角色的位置上迁移.例如,计算需要的数据分割后,随计算任务(可执行代码)从 *CT* 通过 *RC* 分配到不同的 *RP* 上运算.

(2) 转换——是包含和引用关系的改变,伴随发生大规模的迁移.例如,节点由 *RC* 转换为 *RP*,则 *RC* 上的数据、应用关系等构件实体都要迁移.构件的组织形式如图 4 所示.

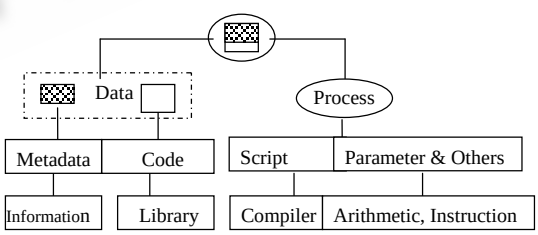


Fig.4 The structure of component model

图 4 构件模型的结构

例如,可以把每个任务定义为包含两个部分的内容,即"数据"和"过程","数据"又包含计算所用的信息数据

和相关的函数库.“计算过程”就是可执行代码的推演过程.

定义 *agent.part*={*inf*,*cd*,*sp*,*pa*},分别表示构件的数据信息、代码、脚本、参数.

(3) 如图 4 所示,数据部分(元数据和代码)迁移称为弱迁移;数据和过程(包括状态和实时参数)同时移动称为强迁移.

5 构件系统的形式模型

资源系统形式模型描述有很多方法,我们用集合理论来定义各种成分和它们之间的关系.

G 代表所有成分的集合,这里描述的成分主要有资源 *R*、数据 *D*、代码 *Cd*、节点 *Nd*、参数 *Pa*、角色 *Ac*、位置 *P*、构件 *C*、*ag*(表示 agent)、变量 *V*、地址 *ad* 和系统 *S*,实体定义在一个名字空间 *Name* 中.多元组 $G\in\langle R,D,Cd,Nd,Pa,Ac,P,C,ag,ad,V,S\rangle$;模型中各基本成分的定义如下:

Def. 1. *R* 为志愿者提供的资源实体,$R\in\{Rp,Ct,Rc\}$这 3 个角色,标示为 *R.N=Rxi*;所在位置用 *R.p=Nd* 来表示;角色转换意味着配置发生了变化,用 *Ri.cg* 表示.

Def. 2. 资源构件 $Ci\in\{D,Cd,ag,Pa\}$,操作为 *Ci.cmd*,引用资源列表 *Ci.reflist*.

Def. 3. 若 *Ci=D* 或 *Ci=Cd*,则 *Ci.cmd*={*copy*,*run*,*moveto*,*Updata*,*Takeway*}.

Def. 4. 若 *Ci=ag*,则:构件的状态 $Ci.s\in\{active,Nactive\}$;活动列表 $Ci.act(Ci.s)\in\{moveto,Clone,updata,Ref,Deref,Send,Receive,\ldots\}$;*Ci.s*(*pa*)表示构件某状态下带参数 *pa*;*Ci.part*={*inf*,*cd*,*sp*,*pa*}.

Def. 5. 节点 $Nd_i.Ac\in\{Rp,Ct,Rc\}$,*Nd.in*(*Ci*)*c* 表示节点的包含构件.

Def. 6. 节点 *Pi*,抽象包含名字 *Pi.name=pi*,子列表 *Pi.chlist*.

Def. 7. 设构件集合为 *Ci*,节点集合为 *Ndi*,则:

(1) 构件之间的组合用 *gup*(*Ci*)表示,*i*={1,2,3,…};

(2) 引用关系用 *Ref*(*Ci*)表示,*i*∈{1,2,3,…};

(3) 节点之间的层次关系用 *child*(*Ndi*)表示,*i*={1,2,3,…}表示父子;

(4) 包含关系由 *Ndi=icd*(*Cj*)表示,*i*={1,2,3,…},*j*∈{1,2,3,…}.

资源角色与构件,角色与节点之间的关系由 $role(Ci)\in\{RP,RC,CT\}$表示.

定理 1. 已知(*Nd*1.*Ac=RC*) && (*Nd*2.*Ac=RP*) && (*Nd*3.*Ac=CT*),则:

- 组合:*gup*(*Nd*1.*RC*)=*gup*(*Nd*2.*RC*).由 Def. 7(1)可证.
- 层次:if [*Nd*1=*icd*(*Rci*) && *Nd*2=*icd*(*Rpi*)],则 *Nd*2.*Rpi=child*(*Nd*1),即 *Nd*2.*Rpi* 与 *Nd*1 为父子关系.由 Def. 5 和 Def. 7(2)可证.
- 引用:if *Nd*2=*icd*(*RP*2)&&*Nd*3=*icd*(*CT*3),则 *Nd*3.*CT=Ref*(*Nd*2.*RP*).由 Def. 5~Def. 7 可证,其中,*Nd*2.*Rpi=Nd*1.*Pi.Chilst*.

如果满足以上关系,则 *RC=role*(*Nd*1)||*RP=Role*(*Nd*2)||*CT=role*(*Nd*3).

证明:由已知,且 *Nd*2.*Rpi=child*(*Nd*1) && *Nd*2.*Rpi=Nd*1.*Pi.Chilst*.

*Nd*1 与 *Nd*2 为组合关系,由 Def. 1(见图 3)和 Def. 7 可证:*RC=role*(*Nd*1)||*RP=Role*(*Nd*2).

定理前件:已知引用关系和 Def. 7(4),可证:*RC=role*(*Nd*1)||*RP=Role*(*Nd*2)||*CT=role*(*Nd*3). □

志愿模式中的计算资源存在于普适环境中,适合采用 Ambient 环境算法的思想.在基本模型中,环境的概念为抽象有界的计算场所,即计算资源以软、硬件的形式重叠交互出现,出现形式从基本规律和概念上满足广域分布式计算的本质.如果把资源构件看作是独立运算的单元,则 *RP*,*RC*,*CT* 为各个计算节点上的构件与功能的耦合关系.承担同一角色的节点由不同构件组成,构件之间为平等的组合关系,例如 P2P 系统.在节点内部,这些构件承担不同的角色,并且相互之间紧耦合.

Def. 8. 用 *C@R* 表示构件 *C* 承担角色 $R,R\in\{Rp,Ct,Rc\}$.

(1) 用 *Nd@R* 表示节点 *Nd* 承担的角色为 $R,R\in\{Rp,Ct,Rc\}$;

(2) 志愿者间紧耦合的计算表示为 *Ci@RP* && *Cj@RP* && *Ck@RP*;

(3) 分布式计算中,对于任务来说,节点间关系为松耦合的并行计算表示为

$$Ndi@RP||Ndj@RP||Ndk@RP.$$

定理 2. 由系统的自相似性,容易证明志愿计算系统为

$$S_{VOL}=\left|\left|_{i,j}ag@_{i,j}\&\&E.\right.\right.$$

其中,$C@E$ 表示资源构件 C 在志愿计算环境 S_{vol} 中,上式表示构件 C 的构成结构为 agent,记为 ag.如果采用如图 4 所示的构件框架,则构件的计算实体为 agent,计算实体的描述转换为 agent 在实体节点上,则 $ag@Nd$ 节点代表了一系列状态转换之间的规约.

Def. 9. 设 e 为构件在志愿计算环境 E 中对应的配置信息,表现为 $e\in\{(ad1,v1),(ad2,v2),\ldots,(adi,vi),\ldots\}$,语义为某位置 ad 上的取值 v,包含构件对环境感知的所有上下文,则构件 C 与环境 E 的关系反映为实体对偶(e,ag),则$[e@Nd,ag]$表示 agent 在节点 Nd 上的配置.

Def. 10. 角色 $RP.RC.CT$ 由$[e@Nd]=\{(ad1,v1)\ \&\&\ (ad2,v2)\ \&\&\ldots(adi,vi)\}$判定,$Nd$ 可以为构件或节点,(ad,v)表示节点 Nd 是否具备满足 $RP||RC||CT$ 的软、硬件环境.

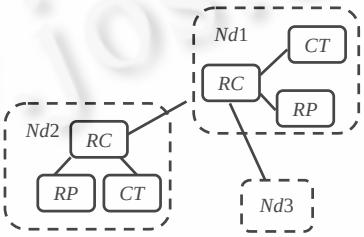


Fig.5 The relationship between the volunteer computing nodes

图 5 志愿计算节点间相互关系

6 语义和规约

6.1 系统语义

由 Def. 2 可知,当构件的成分为 agent 时,系统语义描述如下("→"表示状态的转换):

(1) ag 的操作使节点 Nd 改变角色,即$[e,ag@Nd]\rightarrow[e',ag@Nd']$,其中,$e'=ag.s(pa)\ \&\&\ ag.s=active$,表示构件对上下文的作用.

(2) ag 在节点间迁移,即$[e,ag@Nd]\rightarrow[e,ag'@Nd']$,其中,$Nd'=gup(Nd),ag'=ag.act(ag.s)$.

(3) 构件与节点同时转变,这个过程比较复杂,即节点在志愿计算中角色和节点上的计算同时转换.这种情况往往发生在计算资源和计算任务同时迁移的时候,例如软、硬件同时重构.可以认为是(1)、(2)两种情况同时发生.具体基本操作规约如下:

更新:$[e@Nd,ag.act=\{Ci.s(pa),updata\}]\rightarrow ag'=ag.act(ag.s)$;

$(ad,v)\rightarrow(ad,v'),Ci.inf=\{\cup Ci'.inf\}$.

发送:$[e@Nd,ag.act=\{Ci.s(pa),Send\}]\rightarrow\{[e,ag@Ci]\rightarrow[e,ag'@Ci']\}$;

$\{(ad,v)\rightarrow(ad',v)\}\ \&\&\ child(Ci.ad)=\{\cup Ci'.adi'\}$.

引用:$[e@Ci,ag.act=\{Ci.s(pa),ref\}]\rightarrow[e@ci\cup\{cj.ad\}],Ci.reflist\cup\{Cj.v\}$.

删除:$Del[e@Ci,ag.act=\{Ci.s(pa),del\}]\rightarrow[e@ci—\{cj.ad\}],Ci.reflist—\{Cj.v\}]$.

接收:$[e@Nd,ag.act=\{Ci.s(pa),receive\}]\rightarrow\{[e,ag@Nd]\rightarrow[e',ag@Nd']\}$;

$[(ad,v)\rightarrow(ad',v)]\ \&\&\ child(Ci.inf)=\{\cup Ci'.inf\}$.

6.2 系统操作规约

Agent 作为系统中的组件,节点作为计算资源,则志愿计算各个资源的操作 *R.OP* 的场景规约表示如下:

(1) *RC.OP*∈{*updata*,*receive*,*ref*,*send*};任务在 *RC* 上进行分割,通过 *RC* 的更新操作分割为 agent,再通过引用操作获得分布在各处的资源提供者 *RP* 参与计算,接下来通过发送操作把任务分发到各个 *RP*,计算完成后主动从 *RP* 接受到任务结果,使用接受操作.

更新:[*e*@*Nd*,*ag.act*={*Ci.s*(*pa*),*updata*}]→*ag*′=*ag.act*(*ag.s*);{(*ad*,*v*)→(*ad*,*v*′)} && {*Ci.inf*={∪*Ci*′.*inf*}}.

引用:[*e*@*Ndi*,*ag.act*={*Ci.s*(*pa*),*ref*}]→[*e*@*Ndi*∪{*cj.ad*},*Ci*[*reflist*∪{*Cj.v*}].

例如,[*e*@*Nd*,*ag.act*={*Ci.s*(*pa*),*ref*}]→[*e*,*ag*@*Ndi*∪{*Ndj.CT*}∪{*Ndk.RP*}].

发送:[*e*@*Nd*,*ag.act*={*Ci.s*(*pa*),*Send*}]→[*e*,*ag*@*Nd*]→[*e*′,*ag*′@*Nd*′];

(*ad*,*v*)→(*ad*′,*v*) && *child*(*Nd.ad*)={∪*Nd*′.*adi*′}.

(2) *CT.OP*∈{*send*,*receive*}资源客户端用发送提交计算任务,用接受获得计算结果.

发送:[*e*@*Nd*,*ag.act*={*Ci.s*(*pa*),*Send*}]→{[*e*,*ag*@*Nd*]→[*e*′,*ag*′@*Nd*′]};

{(*ad*,*v*)>(*ad*′,*v*)} && {*child*(*Nd.ad*)={∪*Nd*′.*adi*′}}.

接受:[*e*@*Nd*,*ag.act*={*Ci.s*(*pa*),*receive*}]→{[*e*,*ag*@*Nd*]→[*e*,*ag*@*Nd*′]};

{(*ad*,*v*)→(*ad*′,*v*)} && {*child*(*Ci.inf*)={∪*Ci*′.*inf*}}.

(3) *RP.OP*∈{*receive*,*updata*},*RP* 从 *RC* 接受任务 agent,完成运算更新后等待 *RC*.

更新:[*e*@*Nd*,*ag.act*={*Ci.s*(*pa*),*updata*}]→[*ag*′=*ag.act*(*ag.s*)];{(*ad*,*v*)→(*ad*,*v*′)} && {*Ci.inf*={∪*Ci*′.*inf*}}.

接受:[*e*@*Nd*,*ag.act*={*Ci.s*(*pa*),*receive*}]→{[*e*,*ag*@*Nd*]→[*e*,*ag*@*Nd*′]};

{(*ad*,*v*)→(*ad*′,*v*)} && {*child*(*Ci.inf*)={∪*Ci*′.*inf*}}.

(4) *RP*,*RC*,*CT* 与系统中的软、硬件实体资源对应,对某一实体角色的转换可以由引用 ref 过程,更新 *updata* 过程和 del 删除过程这 3 个过程交替演算来描述.

删除引用:[*e*@*Ndi*,*ag.act*={*Ci.s*(*pa*),*del*}]→[*e*@*Ndi*—{*Cj.ad*}];*Ci.reflist*—{*Cj.v*}.

可以用操作规约简化的描述自愿计算构件间的各种关系,例如,XtremWeb[21]图六网格系统[22]中,任务管理是协调员-工作者(dispatcher-worker)模式.XtremWeb 由 3 个实体组成——Dispatcher,Worker 和 Client,分别承担 *RC*,*RP*,*CT* 的功能.协调员负责任务管理进程.志愿资源的提供者是那些贡献出空闲 CPU 时间来执行由协调员提供的计算任务的志愿计算设备.图 6 表示一个简化的志愿计算的处理过程:1) 表示用户通过 *CT* 提交计算任务,把计算任务提交给 *RC* Dispatcher;2) Dispatcher 引用志愿者提供的空闲计算资源,任务被更新操作分割为各个副本;3) *RC* 发送任务到 *RP* 即 Worker 计算,*RP* 接受计算任务;4) 表示 *RP* 通过计算更新任务包,*RC* 获得计算结果后通过更新各个计算结果的副本;5) *RC* 把更新后的最终结算结果发送回 *CT*,*CT* 接受结果后反馈给用户.

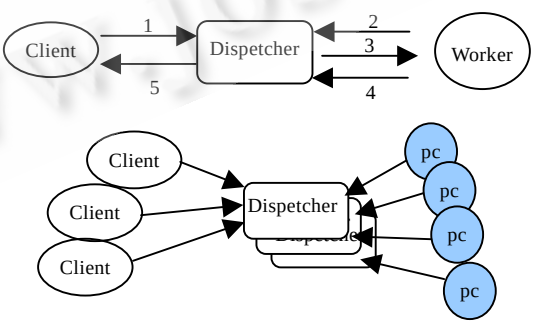


Fig.6 The framework of the XtremWeb

图 6 XtremWeb 框架结构

例如,操作过程形式化描述如下:

1) *CT.send*,*RC.receive*

2) *RC.ref,RC.updata*

3) *RC.send,RP.receive*

4) *RP.updata,RC.receive,RC.updata*

5) *RC.send,CT.receive*

Xtremweb 的实验平台为实验室中各个自愿提供资源的计算节点,包括 1 个 Dispatcher 节点和 52 个 Worker 节点,所选 Worker 节点都为桌面台式 PC,硬件、软件配置大致相同.在模拟实验中,志愿者的加入过程描述为 *RC* 引用新的 *RP*,*RP* 更新计算任务副本(*RC.ref*,*RP.updata*);志愿者的离开过程描述为 *RP* 通过计算更新任务包,获得计算结果后被 *RC* 获得计算结果,*RC* 删除(*RP.RP.updata*,*RC.receive*,*RC.del*).将实验室局域网内提供网格服务的计算机拟定为网格计算中的计算单元,每个单元既能作为独立的服务提供者,又能与其他单元一起通过 MPI 协同工作,实现并行计算.

7 总 结

本文分析和定义了志愿计算系统中的基本概念和它们之间的关系,这些志愿提供计算资源的实体在软件构件等技术支持下,以开放、自主的方式存在于 Internet 的各个节点上,同时具有一定的自相似性和自适应性.如果使用构件和语义的规约规则描述,这样一些软件实体以各种协同方式与其他软件实体进行跨网络的互连、互通、协作和联盟.文中以 Ambient[23]模型为基础,刻画了并行分布式计算平台的紧耦合系统模型,但同时又强调了资源与角色概念的结合,形式地描述了关系变化表现的各种提供资源的志愿者的概念和关系.志愿计算模式作为网格技术的一种分支,形式化地描述它的核心概念和问题,可作为系统结构设计和实现的基础.例如上面所说的 XtremWeb,其中 Client/Worker/Dispatcher 这 3 个组件的功能可以分别对应于 CT/RP/RC,虽然严格按照志愿计算形式化构件设计的系统还需要完善.进一步的工作是基于更精确的规约,设计一个充分支持 Web 计算资源系统模型特点的资源规范描述语言.

References:

[1] Anderson DP, McLeod VII J. Local scheduling for volunteer computing. IEEE 1-4244-0910-1/0, 2007.

[2] Choi SJ, Baik MS. Volunteer availability based fault tolerant scheduling mechanism in desktop grid computing environment. In: Proc. of the 3rd IEEE Int'l Symp. on Network Computing and Applications. Cambridge: IEEE Computer Society, 2004. 366−371.

[3] Distributed.net. 2007. http://distributed.net

[4] SETI@home. 2006. http://setiathome.ssl.berkeley.edu

[5] Berman F, Fox GC, Hey AJG. Grid Computing: Making the Global Infrastructure a Reality. Chichester: John Wiley & Sons, Ltd., 2003.

[6] Baratloo A, Karaul M, Kedem Z, Wyckoff P. Charlotte: Metacomputing on the Web. In: Proc. of the 9th ICPDCS Int'l Conf. on Parallel and Distributed Computing and Systems. Chicago: Elsevier Science Publishers, 1996. 559−570.

[7] Sarmenta LFG, Hirano S. Bayanihan: Building and studying volunteer computing systems using Java++. Future Generation Computer Systems Special Issue on Metacomputing, 1999,15(5-6):727−735.

[8] Neary MO, Brydon SP, Kmiec P, Rollins S. Javelin++: Scalability issues in global computing. In: Geoffrey F, ed. Proc. of the Scalability Issues in Global Computing, Concurrency: Parctice and Experience. John Wiley & Sons, Ltd., 2000. 727−735.

[9] Lau LF, Ananda AL, Tan G, Wong WF. Gucha: Internet-Based parallel computing using Java. In: Parhami B, ed. Proc. of the Int'l Conf. on Algorithms and Architectures for Parallel Processing. 2000. 397−408.

[10] Fedak G, Germain C, Neri V, Cappello F. XtremWeb: A generic global computing system. In: Proc. of the Int'l Conf. on Grid and Cooperative Computing, Workshop on Global Computing on Personal Devices. Brisbane: IEEE/ACM, 2001. 582−587.

[11] Morrison JP, Kennedy JJ, Power DA. WebCom: A Web based volunteer computer. The Journal of Supercomputing, 2001,18(1): 47−61.

[12] Anderson DP. BOINC: A system for public-resource computing and storage. In: Baker M, ed. Proc. of the 5th IEEE/ACM Int'l Workshop on Grid Computing. Pittsburgh: Rajkumar Buyya, 2004. 365−372.

[13] Actors AG. A Model of Concurrent Computation in Distributed Systems. Cambridge: The MIT Press, 1986. 28−40.

[14] Milner R, Parrow J, Walker D. A calculus of mobile processes parts I and II. Journal of Information and Computation, 1992,100(1): 1−77.

[15] Lamport L. The temporal logic of actions. ACM Trans. on Programming Languages Systems, 1994,16(3):872−923.

[16] Asperti A, Busi N. Mobile Petri nets. Technical Report, UBLCS 10, Bologna: University of Bologna, 1996. 1−13.

[17] Merro M, Nardelli FZ. Behavioral theory for mobile ambients. Journal of the ACM, 2005,52(6):961−1023.

[18] Cardelli L. Abstractions for mobile computations. In: Vitek J, ed. Proc. of the Microsoft Research Secure Internet Programming: Security Issues for Mobile and Distributed Objects. London: Springer-Verlag, 1999. 1−47.

[19] Saha D, Mukherjee A. Pervasive computing: A paradigm for the 21st century. IEEE Computer, 2003,36(3):25−31.

[20] Vitek J, Castagna G. A calculus of secure mobile computations. In: Proc. of the IEEE Workshop on Internet Programming Languages. Boston: Springer-Verlag, 1998. 47−77.

[21] Lodygensky O, Cordier A, Fedak G, Nen V, Cappello F. Auger & XtremWeb: Monte Carlo omputation on a global computing platform. In: Branson J, ed. Proc. of the Computing in High Energy and Nuclear Physics 2003. La Jolla, 2003. 24−28.

[22] Fedak G, Germain C, Neri V, Cappello F. XtremWeb: A generic global computing platform. In: Proc. of the IEEE/ACM Special Session Global Computing on Personal Devices. IEEE Press, 2001. 582−587.

[23] Wei J, Zhou H. Specifying and analyzing model for mobile component systems. Journal of Software, 2001,12(1):56−64 (in Chinese with English abstract).

附中文参考文献:

[23] 魏峻,周桓.移动组件系统模型的分析与描述.软件学报,2001,12(1):56−64.

王宇 (1979－),男,云南昆明人,博士生,助教,CCF 学生会员,主要研究领域为移动计算,水信息学,嵌入式软件环境.

王志坚(1958－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为软件复用技术,构件技术,分布式计算.