面向应用的分布式多机联合计算的系统设计问题*

凌云翔　党岗　史扬　金士尧

(国防科技大学计算机系　长沙　410073)

摘　要　随着计算机应用向分布和异构发展,如何建立一个具有良好性能价格比的分布式多机联合计算系统是当前计算机应用开发者遇到的重要课题。文中结合分布式仿真应用实例,对多计算机联合计算的系统划分、数据通讯机制和高性能并行计算问题进行了论述。

关键词　多计算机系统, 通讯机制, 并行计算

分类号　TP302.1

Basic Principles of Application System Design of Collocative Computing of Distributed Multi-computers

Ling　Yunxiang　Dang Gang　Shi Yang　Jin Shiyao

(Department of Computer Science, NUDT, Changsha, 410073)

Abstract　Base on a case of collocating computing system design, this paper discusses the principles of system partitioning, communication mechanism, and high-performance parallel computing of the distributed multi- computers system.

Key words　multi-computers, communication mechanism, parallel computing

1　系统的划分

系统划分是将一个系统从功能及数据方面逻辑地分成片,并把它们分配给那些可用的资源。划分没有一般的综合性原则,往往是遵循应用问题本身固有的自然边界范围[3]。例如,从事多种武器对抗综合仿真所需的分布式仿真环境(软、硬件平台)[2],在体系结构上可采用分布交互仿真(DIS)或高层体系结构(HLA),由于参与仿真的各实体所充当的角色和效能参数不同,对各自仿真平台的需求也有所不同,飞机和导弹的帧时间为毫秒级,而雷达和火炮要求在秒级,前者可由高性能计算机完成,后者在一般的微机上即可仿真;而且各实体并非孤立节点,相互之间还有协同或者对抗的关系。

由此可见,由多计算机联合组成的分布式系统的目的是充分发挥各个计算机的优势,并通过互联网络来支持计算实体间的有效协作,从而获得整个系统良好的性能价格比。为了叙述方便起见,以图 1 所示的体系结构为例来讲述这种典型的异构多计算机联合系统:①系统的硬件实现基于总线(如 PCI)的开放式体系结构,总线对应用模型的数据传输延迟必须满足系统中最苛刻的部分;②控制计算机与网上的工作站通讯以便进行交互控制和显示,包括程序加载、运行时间交互、选择变量显示和参数调整等;③通讯计算机是系统中的调度和通讯中心,负责全部数据的交割;④各种档次的计算机平台构成了多个并行计算资源,它归并了计算能力和外部硬件接口为一体,能满足应用问题对计算和 I/O 的要求。

2　联合计算系统的通讯机制

美国国防部建模与仿真办公室(DMSO)于 1996 年 8 月提出了一个全新的高级体系结构 HLA (High Level Architecture)[1]。HLA 的一个重要特征是将仿真应用与底层的通信和基本功能相分离,由

* 国家部委基金项目资助
1998 年 10 月 27 日收稿
第一作者:凌云翔,男,1972 年生,博士生

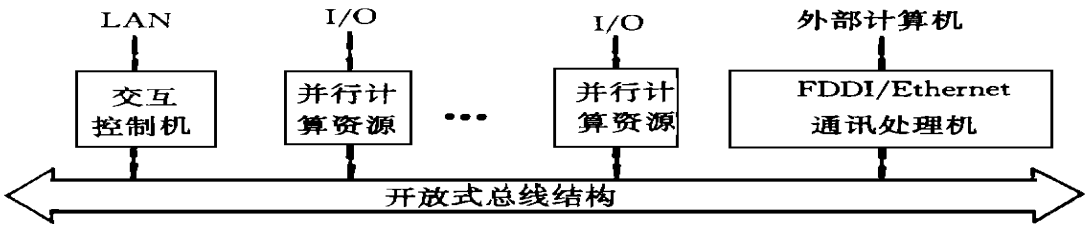

图 1 多计算机联合计算系统的组织

Fig. 1 Distributed multi- computers collacative computing architecture

RTI(Run Time Infrastructure)提供的服务来实现底层的通信和基本功能,即在一个联邦的执行过程中,所有的联邦成员按照 HLA 的接口规范说明所要求的方式同 RTI 进行数据交换,实现成员间的互操作。RTI 提供的功能相对于联邦成员是透明的,联邦成员不必涉及底层的网络编程,因而可将精力放在应用领域有关的仿真应用开发上,同时遵循共同 RTI 接口的仿真应用可灵活组成功能各异的联邦,有利于仿真构件的重用以满足不同的需要。RTI 作为联邦构建与运行的核心,是分布仿真系统的通讯中心,各实体之间的通讯主要通过时间管理(TM)和数据分配管理(DDM)来实现。

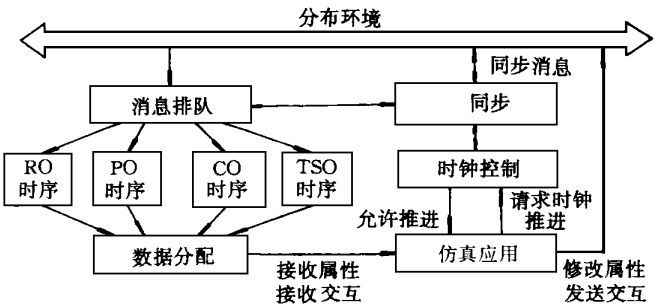

图 2 时间管理框架

Fig. 2 Time management architecture

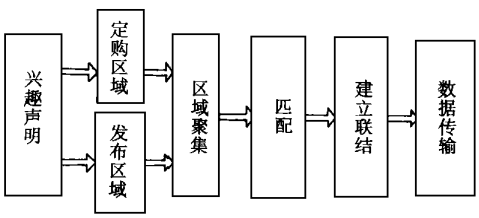

图 3 数据分配管理框架

Fig. 3 Data distribution management architecture

2.1 同步方式: 时间管理

从 HLA 高层看来,联邦相对于 RTI 是一组相互交换带时戳事件的成员集合,而 RTI 是事件的交换机。时间管理的目的是保证 RTI 能在适当的时间以适当的方式和顺序将来自成员的事件转发给适当的其它成员。时间管理是综合集成框架,如图 2 所示,可以支持联邦内多个成员之间不同的时间机制。

①联邦成员之间可以有不同的时序要求,时间管理支持按四种时序处理消息,包括接收顺序(RO)、优先级顺序(PO)、因果序(CO)和时戳顺序(TSO)。

②联邦成员的时钟推进方式不同,通常用以下方式之一进行: 独立时间推进,步进的协调的时钟推进,事件驱动的时钟推进。

③联邦成员之间的时间制约机制不同。时间约束(Time Constrained)成员受到其它成员的时钟制约(如实时应用),时间规划(Time Regulating)成员则会影响其它成员的时钟。

④联邦成员之间的同步机制不同。保守同步(Conservative Synchronization)要求消息或事件的处理遵循 TSO,优化同步(Optimistic Synchronization)在一定程度上不受 TSO 的限制。

2.2 通讯方式: 数据分配管理

数据分配的目的是限制一个大规模联邦中成员接受的信息范围,这样一方面可以减少成员处理的数据量,另一方面可减少网络上传输的数据量。支持数据分配的基本概念是路径空间(Routing Space),路径空间是一个多维的关键空间,联邦成员可以用它来描述其希望接受与发送的数据。路径空间的子

集称为区域(Region)。联邦成员可以指定一个订购区域(Subscription Region), 这意味着通知 RTI 只有落入该区域的数据才发送给它。联邦成员也可以为一个对象指定一个更新区域(Update Region), 这表示成员将保证当它更新对象属性时, 如属性值进入了更新区域, 则将更新的属性值发出。

实现 DDM 的框架如图 3 所示。①兴趣声明: 每个邦元向 RTI 声明自己欲接收的数据和欲发送的数据; ②聚集: 归并定购区域、发布区域, 减少要处理的区域数; ③匹配: 将归并后的定购区域和发布区域进行比较, 以确定重叠情况, 保证定购成员能接收到应该接收到的数据; ④建立联结: 根据匹配的结果, 建立网络联结, 一个联结即为一个组播组; ⑤传输数据。

3 联合计算系统的高性能计算

3.1 多任务生成

联合计算系统的高性能计算与是否充分挖掘并行资源的并行处理能力有密切关系。实现高性能并行计算的技术难点可归结到两处: 一是如何确定一个好的并行划分模式和调度算法以获得最大的程序并行性; 二是如何进行同步与通信的控制以保证并行程序的正确性、高效性。多任务生成的解决方案如下所示, 其实质就是如何安排各个子任务的执行时刻表的调度问题[5]。

①根据应用模型构造出表示计算任务先后次序的数据流图;

②基于并行计算机系统采用一种求解数据流图的调度算法, 用于分配一组具有先后关系、数据通讯约束的并行有序任务集;

③采用信号量控制并行任务之间的同步与通信, 确保应用程序正确、高效地运行。

3.2 高性能仿真计算实例

雷达电子对抗综合仿真是典型的具有多种仿真应用模型参与的多计算机联合计算系统。从 HLA 的角度来看, 各模型是系统中的一个联邦成员, 遵循 HLA 接口规范, 通过 RTI 与其它成员交互[4]。各种应用模型中, 飞行器仿真对并行计算资源的实时计算能力提出了较高的要求。飞行器属空气动力学系统仿真问题, 我们选择 SMP 结构的工作站作为飞行器成员的并行计算平台。空气动力学连续系统仿真问题具有确定性计算模型, 适合以右函数计算为单位的任务一级并行性开发。

数据流图产生器为每一个仿真计算输出量所产生的结点, 将是多任务生成的主要依据, 定义结点的数据结构(记录) 如下:

NAME	T	Ec	Es	Lc	Ls	I_DG	O_DG	I_LK	O_LK	FLAGS

其中, NAME——输出变量名　　T——运算量

Ec——最早完成时间　　Es——最早启动时间(Es= Ec-T)

Lc——最迟完成时间　　Ls——最迟启动时间(Ls= Lc-T)

I_DG——结点的入度(即输入量个数, 这些输入量又称为源结点)

O_DG——结点的出度(即引用该结点的结点数, 又称为目的结点)

I_LK——指向诸源结点的链表表头　　O_LK——指向诸目的结点的链表表头

FLAGS——一组表征结点状态的标志

我们采用了Scheduler 调度算法, 其基本思想是 "最迟启动时间最小者优先调度"[5]。Scheduler 的时间复杂度为 n 的三次多项式, 其中 n 为调度执行的子任务数目。算法的流程如下:

①求出数据流图中各结点的 Ls, 并置所有的结点为未执行状态。

idle_cpu_cnt= m , scheduling_timer= 0

cpu_state_tab[i] 初始化:

cpu_timer= 0, running_node= 0, idle= true

②将 0 号结点的后继纳入 ready_queue 就绪队列。

③将就绪队列 ready_queue 的前 min{idle_cpu_cnt, ready_cnt} 个结点依次分配到下标最小的空闲处理机; 如果 running node 与刚分配的结点之间没有弧, 则引入一条 "附加弧"。修改这些处理机的

cpu_state_tab:

idle= false , running_node= 刚分配的结点号

cpu_timer= cpu_timer+ ti(ti 为该结点的处理时间)

④对所有的处理机求 next_timer= min{cpu_timer cpu_timer> scheduling_timer}, 如果 next_timer> scheduling timer 则转入下一步, 否则调度结束。

⑤scheduling_ timer= next_ timer, 如果 cpu_timer≤scheduling_timer, 则把该 cpu 的 cpu_state_tab 中的 idle 置为 true, 并求 idle_ cpu_ cnt。

⑥将 cpu_timer= scheduling_timer 的 cpu 上的 running_ node 置为已执行, 并将其直接后继中可执行的结点纳入 ready-queue 队列, 如果 ready_ cnt> 0 则转到③, 否则转至④。

其中: ready_ cnt: 可执行结点个数

ready_ queue: 按 Ls 从小到大排列的可执行结点队列

scheduling_ timer: 指示调度进度的时钟

cpu_state_tab[1∶m]: 关于 m 个处理机的状态表, 每个表包括三项: 处理机忙闲标志 idle, 处理机时钟 cpu_ timer, 处理机上最近运行的结点编号 running_ rode。

idle_ cpu_ cnt: 空闲处理机计数

next_ timer 为临时变量, 0 号结点为入口结点。

以某飞航式导弹仿真模型(步长 h= 5ms) 为应用实例, 在 SMP 结构的SGI Challenge 工作站(四个 MIPS 处理机) 上进行并行计算, 得到如表 1 所示的结果。表 1 说明通过合理的多任务生成, 多计算机联合系统中的并行计算资源完全可以达到高性能计算的目的, 促使整个系统性价比的提高。

表 1 仿真计算实例数据

Tabel 1 Example: simulation computing data

	SGI(串行计算)	SGI(并行计算)	YH-F2(银河仿真 II)
帧时间(ms)	0.47	0.18	0.24
加速比(h/帧时间)	10.67	27.8	21.10

4 结束语

计算机应用领域向广度和深度的发展, 极大地推动了具有良好性能价格比的分布式多计算机联合系统的研究、设计和实现。本文结合基于HLA 的雷达电子对抗仿真实例, 对多机联合计算的系统划分、数据通讯和高性能并行计算问题进行了讨论, 希望有益于读者。对于多计算机环境下的运行控制和分布式任务调度, 作者将在以后的论文中论述。

参考文献

1 DMSO, High Level Architecture Rules: Version 1.0. August 15 1996, http://www.dmso.mil

2 金士尧, 凌云翔, 毛羽刚. 现代化战争中武器对抗仿真平台的研究. 国防科技参考, 1997(4): 117～124

3 苏金树, 吴纯青, 卢锡城. 多计算机联合计算的系统设计基本问题. 计算机研究与发展, 1995, 32(10)

4 凌云翔, 王召福, 刘晓建, 金士尧. 基于 HLA/RTI 的仿真系统设计. 国防科技大学学报, 1999(2)

5 凌云翔, 杜铁塔, 金士尧. 基于 SMP 结构的连续系统并行仿真. 计算机学报, 1997, 20(增刊): 77～82