第 34 卷 第 4 期 计 算 机 工 程 2008 年 2 月
Vol.34 No.4 Computer Engineering February 2008
·软件技术与数据库· 文章编号：1000—3428(2008)04—0071—02 文献标识码：A 中图分类号：TP316.4

并行 BSP 模型在实时集群系统中的应用

薛弘晔[1,2]，李言俊[1]，杜 鸿[3]
(1. 西北工业大学航天学院，西安 710072；2. 西安科技大学计算机科学与技术系，西安 710054；
3. 成都信息工程学院计算机系，成都 610225)

摘 要：分析 BSP 并行计算模型在多源数据处理中的应用特点。构建实时集群计算机系统的并行计算 BSP 模型。对多源任务数据处理的粒度进行了分析设计。给出了实时集群计算机系统中 BSP 模型的实现算法。实际应用验证了算法的有效性。
关键词：并行计算模型；实时集群计算机；BSP 模型

Application of Parallel BSP Model in Real-time Cluster Computing System

XUE Hong-ye[1,2], LI Yan-jun[1], DU Hong[3]
(1. College of Astronautics, Northwestern Polytechnical University, Xi'an 710072; 2. Dept. of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054; 3. Dept. of Computer, Chengdu University of Information Technology, Chengdu 610225)

【Abstract】This paper analyses the characteristic of parallel computing model's application in multi-source data processing, constructs the BSP model of real-time cluster computing system, designs granularity of multi-assignment data processing, and puts forward a kind of new method to realize BSP model in real-time cluster computing system. The appliance proves the validity of the algorithm.
【Key words】parallel computing model; real-time cluster computing system; Bulk Synchronous Parallelism(BSP) model

研究并行计算模型在具体并行计算系统上实现的规律对开发高性能的并行计算软件具有重要意义。当一个应用问题的计算过程需要采用并行技术实现的时候，需要考虑如下几个问题：并行计算机的硬件体系结构，实现并行计算的并行算法模型，针对特定应用问题的并行算法。并行算法模型作为并行计算实现过程的性能评估手段，衍生出针对具体应用问题的有效算法。这里针对一个实时集群应用系统平台，介绍基于 BSP 模型的集群并行算法的实现。在 BSP 模型中，计算过程由一系列用全局同步分开的周期为 L 的超级步(superstep)所组成。在各超级步中，每个处理器均执行局部计算，并通过选路器接收和发送消息；然后进行全局检查，确定该超级步是否已由所有的处理器完成：若是，则前进到下一超级步，否则下个 L 周期被分配给未曾完成的超级步[1-4]。

1 实时集群系统中并行计算模型的构建

在实时集群系统中，采用的并行计算模型是 BSP 模型(同步并行模型)，Oxford BSP 库实现了一个简单而健壮的 BSP 模型。Oxford BSP 库的实现是基于进程级的任务划分，而对实时集群系统而言这种划分的粒度过大，不利于信息的实时处理。由此在对这种并行思想的实现上提出并实现了更低级的方式——以 System V 进程间通信及网络套接字技术为通信基础，在数据级上实现了并行计算模型。BSP 模型使用如下原语定义了一个抽象的并行计算机：(1)执行同步的组件(处理器)；(2)提供组件间点对点通信的路由设施；(3)以一定周期(L)同步所有或部分组件的同步机制，参数 L 表示同步操作之间的最小间隔时间；(4)时间步(time step)是指某些组件对本地数据完成一次操作所需要的时间，它主要用于 BSP 计算性能分析中。

BSP 的计算性能由如下参数描述：处理器数(p)，处理器速度(s)，同步周期(L)和一个指出全局通信平衡计算的参数(g)。处理器速度以每秒可执行的步数计算。L 指成功同步操作之间的最小时间步数。g 是所有处理器每秒钟可以执行的本地操作对通信网络上的总通信字数之比。参数 L 和 g 依赖于处理器数目 p，这种依赖关系由网络结构以及通信同步原语的具体实现所决定。例如，对以固定带宽介质(如以太网)连接的工作站集群而言，大数据量通信将导致各处理器上流入流出的消息呈串行特征。在这种假定之下，g 可以表述为：$g(p)=g_0p$，其中 g_0 为一常数。如果假定同步机制通过软件实现并使用树型结构划分处理器，则 L 可定义为 $L=L_0\log_N(p)$，其中 L_0 为常数。

在实时集群系统中，同步机制是由时统信号来实现的。GPS 产生周期可调的 TTL 脉冲，由智能时统终端接收中断信号并取得 GPS 精密时间然后向集群内部广播含有时间信息的短消息。集群各节点收到时统信号后开始一个新的工作周期并向控制中心应答一短消息，即心跳信号(用来表征节点是否在线同时附有节点资源使用情况和负载情况)。

在系统中，超级步在不同的节点中有不同的内容并且完成时间不能超过时统周期(两时统信号之间隔)。在每一超级

基金项目：国家科技部创新基金资助项目(01c26226111003)；国家火炬计划基金资助项目(2003EB011441)
作者简介：薛弘晔(1960－)，男，副教授、在职博士研究生，主研方向：实时计算机控制；李言俊，教授、博士生导师；杜 鸿，高级工程师、博士
收稿日期：2007-02-21 E-mail：xuehy60@163.com

步内，集群各节点都要完成本节点信息收集并向控制中心发出心跳信号(主控中心和备控中心互发心跳信号)。另外，主控中心要收集集群各节点资源使用和负载情况、统计数据通道信息、完成任务分配；备控中心监听主控中心的心跳信号，若未能收到，备控中心上升为主控；通信服务器要接收测量数据、向控制中心注册数据通道信息并向计算节点广播原始数据；数据库服务器汇总各种数据并广播至指挥显示中心；计算节点要接收原始数据、任务分配表、完成计算任务、向控制中心汇报计算情况、向数据库服务器输出计算结果。系统计算任务处理结构如图 1 所示。

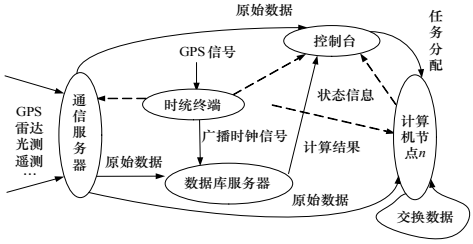


图 1　并行计算任务处理结构

2　任务粒度的选择

集群计算其实质就是将计算任务划分为若干可并行执行的元素，并将其分配到由网络互连起来的集群节点中，同时执行这些计算元素以获得更高的计算性能。为了实现集群并行计算，开发人员总是需要把任务分解为并行分量 w_p 和串行分量 w_s，并使用任务粒度 G_w 和通信粒度 G_c 两种概念来衡量对任务的分解程度，其定义为

$$\text{任务粒度 } G_w=\frac{w_p}{p\times w},\ \text{通信粒度 } G_c=\frac{t_{\text{comm}}}{t_{\text{comp}}}$$

其中，p 为计算任务 w 的并行分量 w_p 的并行度，衡量任务并行分量最多能够同时在多少处理器上被执行；t_{comp} 为并行程序进行数据通信之前的计算时间开销;t_{comm} 为通信时间开销。针对每一个应用，研究人员总是在不断寻求最佳的并行算法，也就是确定合理的计算粒度，以获得更好的计算加速比和计算效率。采用细粒度算法，分解后的计算任务中的串行分量 w_s 可能减少，有利于任务分配，但是又带来消息传递开销 t_{comm} 的增加，会影响系统加速比和效率。采用大粒度算法，任务额外开销 w_o 可能减少，但是又带来串行分量 w_s 的增加，同样影响系统加速比和效率。由于任务串行分量 w_s 和并行计算额外开销 w_o 的存在，使得寻求最佳集群并行算法往往是非常专业和困难的事情，总是需要根据应用特征做出适当折中。

对于实时应用而言，计算由时间或事件激活并执行相应的数据处理任务。系统中一般存在多个可并行执行的计算模块，为此采用模块级的并行计算粒度，并将其称之为中粒度集群计算。采用中粒度的另一个好处是每一个计算模块核心仍然是串行程序，甚至是原有的标准数据处理子程序，能够减少开发工作量。

3　并行计算的实现

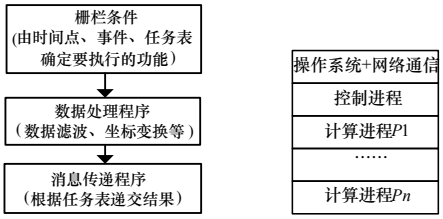
集群并行计算将多个计算任务分配到计算节点中的处理器上同时执行，以提高处理速度。在实时集群系统中，用任务表定义数据处理过程。如表 1 所示。其中，w_a, w_b, w_c, w_d 依次为数据滤波、目标航线管理、坐标变换、控制功能程序块；$D(a, i)$为计算块 w_a 的第 i 个输入数据。

计算程序块 w_a 的第 i 个数据 $D(a, i)$在集群中的处理位置由 $C(w, p, n)$定义，计算结果递交目的由 $O(n, p, w)$定义，这样就能够在程序中使用一个数据结构来生成和维护任务表。对于大多数实时应用来说，数据处理过程总是由时间点或外部事件触发和同步的。在实时集群系统中，采用了基于任务表的 BSP 并行计算模型。

表 1　任务表结构

	栅栏条件	程序模块	计算定位	结果派发
原始数据	$D(a, i)$	Wa	$C(w, p, n)$	$O(n, p, w)$
	$D(b, i)$	Wb	$C(w, p, n)$	$O(n, p, w)$
	$D(c, i)$	Wc	$C(w, p, n)$	$O(n, p, w)$
	$D(d, i)$	Wd	$C(w, p, n)$	$O(n, p, w)$

图 2 表现了集群并行计算模型和并行计算实现方法。图 2(a)是一个 BSP 模型计算进程，每一个计算进程由 3 部分构成：栅栏条件，数据处理程序块和消息传递程序。栅栏条件根据时间点、事件和任务表确定该进程在某一个时间段内要执行的功能；数据处理程序块执行具体计算；消息传递程序根据任务表的定义将计算结果递交给后续程序块。



(a)计算进程结构　　(b)计算进程 2 倍于节点数

图 2　BSP 并行模型和实现方法

当计算节点启动时集群控制进程收集到节点内处理器数量为 P，随即在本节点内复制 P 个相同计算子进程，如图 2(b)所示，然后等待主控制节点通过任务表分配计算任务。集群主控制节点收集系统内在线节点以及处理器资源，根据在线资源生成任务表并将任务表广播到各个计算节点触发计算节点执行计算任务。

在实时集群系统中，并行计算环境中的任务为中粒度，按照 BSP 进程模型修改原有的数据处理程序接口，将数据处理程序嵌入 BSP 进程模型中，并根据数据处理流程生成任务表，即可实现集群并行计算。

4　结束语

实时集群计算机系统的构建主要由硬件和软件两方面的实时性来考虑，对于硬件的构建，可以通过选择较高性能的设备产品来支持；但对于软件方面来说需要操作系统、应用平台软件、应用软件等的实时性来保证[1]。尤其必须重视的是为应用任务选择一个并行性适合的并行计算模型。本文针对一个实时集群计算机系统应用平台在某国防重点工程中的应用，主要介绍了基于 BSP 模型的实时集群并行算法的设计与实现。其实时性通过现场应用得到了验证。但随着科技发展，现场的实际任务可能会对系统提出更高要求，自然需要对系统的实时性作进一步改进和提高。

参考文献

[1] 杜　鸿, 薛弘晔. 一种基于任务表方法的实时集群平台[J]. 计算机工程, 2005, 31(18): 76-78.

[2] 郑纬民, 石　威. 高性能集群计算[M]. 北京: 电子工业出版社, 2001.

[3] 杜　鸿, 赵越超, 薛弘晔. 实时集群计算技术在相控阵雷达中的研究[J]. 现代雷达, 2004, 26(6): 22-25.

[4] Konchady M. Parallel Computing Using Linux[EB/OL]. [2006-12-28]. http://www.linuxjournal.com/article.