

# 一种分布式吴方法计算模型\*

武永卫<sup>1+</sup>, 杨广文<sup>1</sup>, 杨宏<sup>2</sup>, 郑纬民<sup>1</sup>, 林东岱<sup>3</sup>

<sup>1</sup>(清华大学 计算机科学与技术系,北京 100084)

<sup>2</sup>(北京市计算中心,北京 100005)

<sup>3</sup>(中国科学院 软件研究所,北京 100080)

## A Distributed Computing Model for Wu's Method

WU Yong-Wei<sup>1+</sup>, YANG Guang-Wen<sup>1</sup>, YANG Hong<sup>2</sup>, ZHENG Wei-Min<sup>1</sup>, LIN Dong-Dai<sup>3</sup>

<sup>1</sup>(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

<sup>2</sup>(Beijing Municipal Computing Center, Beijing 100005, China)

<sup>3</sup>(Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: +86-10-62785592, E-mail: wuyw@tsinghua.edu.cn, <http://www.tsinghua.edu.cn>

Received 2004-02-10; Accepted 2004-05-14

Wu YW, Yang GW, Yang H, Zheng WM, Lin DD. A distributed computing model for Wu's method. *Journal of Software*, 2005,16(3):384-391. DOI: 10.1360/jos160384

**Abstract:** As Wu's method, based on symbolic computation, has found applications in more and more fields, it is challenged by more and more complicated calculation problems. The soul of the method, dividing and ruling problem, is very suitable for distributed computation. In this paper, a high performance computing technique is introduced into the symbolic computation and a distributed computing model (DCM) for Wu's method over Internet is put forward. First, the feasibility and requirement for distributed computation of Wu's method is analyzed. Then the data communications for big integers and polynomials are put forward in detail, and finally the design and implementation for DCM based on ELIMINO and Globus Toolkits 3 are presented.

**Key words:** distributed computation; Wu's method; symbolic computation

**摘要:** 吴方法是由我国科学家吴文俊院士开创的一个新兴研究领域.考虑到吴方法“分而治之”的思想非常适合分布式计算,将分布式计算技术引入到该方法的计算过程中,给出一种既可以在集群环境下,也可以在网格环境下实现的分布式吴方法计算框架.首先分析了吴方法分布式计算需求,并以特征列计算为例来说明吴方法分布式计算算法,然后讨论了符号计算基本数据类型:大整数和多项式的消息传递方法,最后简单给出了在网格环境下基于符号计算软件系统 ELIMINO 和网格中件间 Globus Toolkits 3 的分布式吴方法计算环境的设计、实

---

\* Supported by the National Natural Science Foundation of China under Grant Nos.60273007, 60373004, 60373005 (国家自然科学基金)

**作者简介:** 武永卫(1974-),男,甘肃靖远人,博士,主要研究领域为符号计算,并行与分布式计算技术;杨广文(1963-),男,博士,教授,主要研究领域为并行与分布式计算技术;杨宏(1973-),女,工程师,主要研究领域为符号计算,计算机软件;郑纬民(1946-),男,教授,博士生导师,主要研究领域为并行处理与分布计算机系统,面向 AI 体系结构,说明性语言的编译方法,程序开发环境;林东岱(1964-),男,博士,研究员,博士生导师,主要研究领域为密码理论,安全协议,符号计算,软件设计.

现与实验结果.

关键词: 分布式计算;吴方法;符号计算

中图法分类号: TP311 文献标识码: A

吴方法(也称数学机械化思想)<sup>[1,2]</sup>是由我国科学家吴文俊院士于 20 世纪 70 年代末开始倡导的一个新兴研究领域,是脑力劳动机械化在数学科学中的学术实践.它的目的是将数学计算与证明的过程标准化,将数学证明和推理中质的复杂性转化为计算上量的复杂性.由于其简单、刻板,因而可以让计算机来实现.吴方法不仅是数学研究的实质性进展,也为很多高科技问题的解决提供了有力的工具.该方法已在许多高科技领域获得了一批理论成果.包括自动推理、曲面造型、机器人位置分析、几何设计、计算机视觉、智能 CAD、信息安全和数字图像的高速高保真传输等等.

目前,国内外已有多个基于现有数学计算软件系统的吴方法计算软件实现,例如由中国科学院数学机械化重点实验室开发的基于 MAPLE 和 ELIMINO 的吴方法计算软件包 WSOLVE<sup>[3]</sup>,法国的 D.Wang 实现的基于 SACLIB 和 MAPLE 的吴方法软件包 Charset<sup>[4]</sup>等.但是随着吴方法在越来越多领域的应用和推广,所面临的计算问题也越来越复杂,这些问题在单个计算机资源上通常要花几天、甚至几十天的时间,或者由于内存的限制,问题就根本无法得到求解.因此分布式吴方法计算研究成为一项急迫的任务.

吴方法计算是一种基于符号计算的密集型计算,与普通的数值计算相比,符号计算复杂性比较高、计算强度比较大.20 世纪末,国外的一些研究人员就已经开始将分布式计算技术引入到符号计算的一些关键计算问题当中,以便获得更高的计算效率.比较有名的有美国 Kent State 大学的 Wang<sup>[5-7]</sup>教授和 MIT 的 Rayes<sup>[6]</sup>教授,他们对并行多项式的 GCD、因式分解等符号计算的关键问题进行了研究,并在共享内存的并行计算机上实现了相关算法.Wu<sup>[8]</sup>等人在 MPI 和 ELIMINO 符号计算系统基础上开展了诸如多项式 GCD 符号计算等核心算法的并行计算研究.

尽管吴方法是由我国科学家提出的思想,在国内也已经取得了令世人瞩目的研究成果,但这方面的高性能计算因为存在跨学科等问题而比较滞后.目前,国外有关吴方法核心算法特征列的计算研究已经有了一定进展,例如美国 Kent State 大学 Ajwa<sup>[7]</sup>等人在 PVM 和符号计算自由软件系统 SACLIB 基础上实现了并行特征列计算软件包.在国内,有关并行符号计算的研究还处于起步阶段,Wu<sup>[9]</sup>等人在 MPI 和 ELIMINO 符号计算系统基础上实现了特征列并行计算等.但这些研究均是有关某一具体算法(如特征列)给出的并行计算方法,并没有试图给出一个通用的分布式吴方法计算研究环境,以使用户自己可以利用网格等技术方便、快捷地实现基于广域网的高性能吴方法相关计算.

考虑到数学机械化计算问题分支动态变化,且各个分支之间没有消息传递的模式非常适合在网络上进行计算,本文将给出一种既可以在集群环境下也可以在网格环境下实现的分布式吴方法计算模型,目的是为我国工程技术领域有关数学机械化的实际计算问题提供一个方便、高效、实用的分布式计算和求解环境.本文第 1 节分析吴方法分布式计算环境的必要条件.第 2 节以特征列计算为例讨论吴方法分布式计算算法.第 3 节和第 4 节分别介绍我们自己研制的符号计算软件系统 ELIMINO 和有关吴方法分布式并行计算所必需的基本数据类型:大整数和多项式的消息传递方法.第 5 节简单给出广域网上吴方法分布式计算环境的设计、实现与实验结果.

## 1 分布式吴方法计算分析

考虑到吴方法是一种处理多项式的符号计算方法,它的分布式计算环境至少应该包含以下 4 项内容:

### (1) 基本的符号计算函数库

作为一种符号计算方法,吴方法主要处理多项式,因此大整数(在系统内存允许的条件下可以尽可能地大)运算<sup>[4]</sup>和多项式(例如多项式因式分解,GCD,伪除等等)处理<sup>[10]</sup>是函数库中最基本而且必须的部分.同时,有关吴方法的一些基本运算(例如:基列,升列等)的实现也应当包含在库函数中.

国际上可以提供基本的符号计算函数库的数学计算软件有:Maple,Mathematica,Risa/Asir 等.其中由中国科学院数学机械化研究中心研制的以吴方法为核心计算的符号计算软件系统 ELIMINO 为我们提供了丰富的吴方法计算所必需的基本符号计算函数库.

#### (2) 基本的并行计算消息传递函数库

并行计算就是让不同的计算节点在同一时间内为同一任务服务.显然,这些计算机之间肯定要进行任务分发、接受任务、传送计算结果等各种消息的传递.因此在并行计算环境中,计算节点之间消息的发送、接受、归约、聚合和进程管理等操作是必须的.

目前,可用的并行计算消息传递函数库有 LAM/MPI,MPICH,MPICH-G2 和 PVM 等.

#### (3) 复杂数据类型的消息传递函数

既然吴方法是针对多项式处理的一种符号计算方法,那么一般的并行编程语言(如 MPI 和 PVM)所提供的简单的数据类型是不能满足吴方法并行计算需求的.所以新定义的复杂数据类型(如大整数、多项式等)的消息传递函数必须被定义和引进.这一部分的内容将在第 3 节加以描述.

#### (4) 吴方法的分布式计算算法

并行计算环境不仅需要满足并行计算的硬件环境和软件环境,而且需要设计分布式的吴方法计算程序,而程序设计的前提是分布式算法的设计.吴方法计算的算法本身就具有自然的分布式计算特性,这一优势我们将在第 2 节加以描述.

## 2 分布式计算算法

简单地讲,吴方法实质上是用了“分而治之”的思想,将数学证明或计算的过程标准化,也就是说,每完成一步,就可以明确下一步将计算什么.而在整个机械化的过程中,又有大量可以独立运算的模块,因此将分布式计算技术引入吴方法计算的过程中是非常可行的.

吴方法计算中的关键算法,多项式最大公因子计算、多项式因式分解等都可以通过分布式计算来提高效率.下面我们以吴方法计算中的核心算法——特征列计算为例来说明吴方法中各种问题分布式计算的算法.在吴方法的计算中,特征列是一个非常重要的概念,特征列包含了其所对应的多项式方程组零点集的所有信息.在一次性求出零点集太困难时,先求出部分零点,差额部分通过添加一些相关的多项式再求.事实上,得到多项式方程组的特征列后,就可以容易地分析多项式的零点.

对一个给定的多项式方程组  $PS$ ,计算其特征列  $CS$  的算法可以描述如下:

```

Input:  $PS$ ;
Output:  $CS$ ;
 $QS:=PS;RS:=PS$ ;
while  $RS!=NULL$  do
     $CS:=BasicSet(QS);RS:=NULL$ ;
    if nonc( $CS$ ) then
        for all  $Q$  in  $QS-CS$  do
             $R:=prem(Q,CS)$ 
            if  $R!=NULL$  then  $RS:=RS+\{R\}$ ;
         $QS:=QS+RS$ ;

```

其中“+”表示集合的并;“NULL”表示空集合;BasicSet( $QS$ )表示  $QS$  的基列;nonc( $CS$ )表示  $CS$  不是矛盾列;prem( $Q,CS$ )表示  $Q$  关于  $CS$  的余式.

有定理保证上述算法经过有限步之后总会终止,即终止条件  $RS$  在一定时候总会为空.最后返回的基列  $CS$  即为所要计算的特征列.

由上述算法可以看出,伪除运算(求余式)是计算特征列最基本的运算.分析和实践显示,上述计算过程中绝大多数的时间都耗费在第 2 步.

特别地,当中间产生的余式序列  $RS$  中的多项式比较大时,该特点尤为明显.而第 2 步的计算就是伪除运算,这样,余式序列  $RS$  的计算首先应该是被并行的,也是我们在整个吴方法并行算法设计的主要切入点.事实上,余式序列  $RS$  的计算是非常适合并行化的.在上述算法中,为了计算  $RS,PS$  中的每一个多项式被单独拿出,然后独立地求关于  $BS$  余式.在  $RS$  的一次计算过程中, $PS$  和  $BS$  是保持不变的,因此我们可以将  $PS$  分割成若干个互不相交的子集,然后分配到多个计算节点上分别求这些子集关于  $BS$  的余式,最后将它们合并即可.我们相信,使用这种分布式算法实现的吴方法的计算效率将肯定会得到提高.

### 3 符号计算软件系统 ELIMINO

ELIMINO<sup>[11,12]</sup>软件是一个以吴方法为核心算法的面向多项式处理的符号计算软件系统.它主要用于代数 and 微分方程组求解、定理自动证明及其相关问题处理.因此,它首先实现了多项式的各种处理算法,并在此基础上完整地实现了吴方法,然后,将其作为系统的核心,进一步完善成为一个功能强大的解方程器和定理自动证明的软件工具.ELIMINO 的系统结构如图 1 所示.

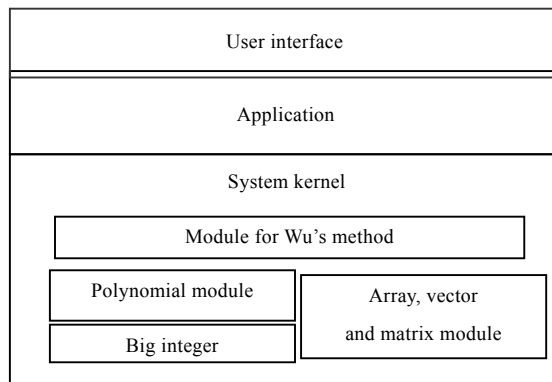


Fig.1 ELIMINO architecture

图 1 ELIMINO 系统结构

ELIMINO 软件实现了以下功能:

- 无穷精度计算(infinite precision computation),包括:大整数、有理数和高精度浮点数的基本运算,例如:+, -, \*, /, >, <, ==, ≤, ≥, >>, <<, GCD, sign, compare 等等;
- 有限域上的基本运算(modular arithmetic);
- 有关线性代数的基本运算(computation for linear algebra),例如:向量、集合、矩阵的基本运算;
- 代数扩域上的基本运算(basic arithmetic in algebraic extensions);
- 在基本运算基础上,实现对多项式的各种处理(polynomial computation),例如:+, -, \*, /, ..., GCD, Factorization, Quotation, Remainder, Substitute, Resultant 等;
- 吴方法的实现,即:完成特征列的计算(characteristic set method).

### 4 消息传递

吴方法的主要处理对象是多项式,而多项式又是以大整数(无穷精度整数)为系数的,因此多项式和大整数作为吴方法计算的核心数据类型,是消息传递的主要考虑对象.

#### 4.1 大整数的数据传输

根据计算机信息单元大小(computer word size)的不同,计算机所能支持的整数范围一般限制在  $65536(2^{16})$  到  $18446744073709551616(2^{64})$  之间.一个大整数可以由一组非负整数  $(a_0, a_1, \dots, a_{n-1})$  表示,大整数  $N$  的值为

$$N = s \sum_{i=0}^{n-1} a_i B^i,$$

其中  $S$  为 1 或 -1, 对应的是大整数  $N$  的正负号,  $B$  是基数,  $a_i (i=0, 1, \dots, n-1)$  小于  $B$ . 为了有效地实现大整数的乘、除运算, 大多数符号计算系统选择  $B$  为 2 的计算机信息单元大小的一半次幂 (例如, 如果计算机信息单元大小为 32, 则  $B=2^{16}$ ).

大整数的传递, 通常有两种方法:

直接法. 传递  $n$  个整数  $a_0, a_1, \dots, a_{n-1}$ , 其中  $a_0$  包含了  $S$  的信息. 也就是说, 对小于  $B$  的大整数的传递需要 4 个字节大小的数据传输来完成; 对大于  $B$  但小于  $B^2$  的大整数的传递需要 8 个字节大小的数据传输来完成, 等等.

转换法. 将大整数  $N$  转换为字符串来完成传递. 也就是说, 对一个有  $l$  位的十进制大整数需要  $l$  个字节大小的数据传输来完成.

直接法简单, 对真正的大整数 ( $>2^{16}$ ) 比较有效. 转换法对小的大整数 ( $>10^5$ ) 比较有效, 但需要传递前后的字符串与符号计算系统内部大整数数据结构之间的数据类型转换.

在文献[9]中, 我们引进了一种新的大整数表示方法, 该方法可以有效地提高大整数的消息传递效率. 我们构造的大整数表示方法是:

$$N' = s \sum_{i=0}^{n-1} a'_i B^i,$$

其中  $B'$  的次幂取与计算机信息单元大小相同, 也就是说  $B' = B^2$ ,  $a'_i (i=0, 1, \dots, n-1)$  仍然是普通整数. 这种表示仅仅为数据传输而设计, 符号计算系统内部仍然采用其原有的数据表示方法. 这样在数据的传输前后需要  $N$  和  $N'$  之间的一个转换过程. 事实上, 这种转换由于它们之间的特殊关系而变得非常容易, 但数据传输却可以取得更好的效率. 我们传递与  $a_i$  相同的  $a'_i$ , 就可以传递更大的大整数. 即传递相同大小的数据, 新方法可以传递更大的大整数, 因此有更高的传递效率.

## 4.2 多项式的数据传输

任意多项式  $p$  可以表示为

$$\sum_{i=1}^n \left( a_i \prod_{j=1}^v x_j^{d_{ji}} \right),$$

其中大整数  $a_i (i=1, \dots, n)$  是  $p$  的系数, 字符串  $x_j (j=1, \dots, v)$  是  $p$  的变元, 普通整数  $d_{ji} (j=1, \dots, v; i=1, \dots, n)$  是变元  $x_j$  在  $p$  的第  $i$  项的次幂. 相对地, 多项式的数据传输也有两种相同的方法:

直接法. 对任意多项式传递下列数据:

- $n$  个大整数  $a_1, \dots, a_n$ .
- $v$  个变元  $x_1, \dots, x_v$ .
- $n \times v$  个普通整数  $d_{11}, \dots, d_{1n}, d_{21}, \dots, d_{2n}, \dots, d_{v1}, \dots, d_{vn}$ .

转换法. 将多项式  $p$  转化为字符串来传输.

直接法对项数比较少、系数比较大的多项式具有更高的性能. 转换法在多数情况下更有效, 原因是多项式的任何一项关于某一变元的次幂总是非常小, 适合转化为字符串传输.

在文献[9]中, 我们提出了一种递归的多项式表示方法:

$$p(x_1, \dots, x_v) = \sum_{i=0}^{\deg(p, x_1)} p_i(x_2, \dots, x_v) x_1^i,$$

其中多项式  $p_i(x_2, \dots, x_v)$  仍然是递归表示的. 在这种表示方法中, 同一变元的同一次幂在整个表达式中只出现一次, 因此传输的数据就会变少. 用我们提出的多项式表示方法, 无论对直接法还是转换法, 对相同的多项式, 所需传输的数据量均变少了. 我们结合直接法与转化法二者的优点, 对多项式的系数, 因其系数是大整数, 我们采用直接法进行传输, 对变元的次幂, 因其总是比较小的整数, 我们采用转化法进行传输.

在吴方法的计算过程中, 需要成千上万次的大整数与多项式的数据传输. 因此, 对单个大整数或多项式传输效率的一点点提高, 对整个计算过程中的数据传输就有了非常明显的改善. 事实上, 当计算问题规模越大时, 我们这种方法的数据传输性能的提高才能更明显地体现出来.

## 5 吴方法分布式计算环境的设计与实现

根据第 1 节分布式吴方法计算的分析和第 3 节符号计算软件系统 ELIMINO 的功能介绍,我们选用 ELIMINO 作为分布式吴方法计算的基本符号计算函数库,而且可以很容易地在 ELIMINO 系统的基础上设计有关吴方法计算的分布式程序.分布式吴方法计算框架如图 2 所示.下面我们来说明符号计算基本数据类型的消息传递方法.

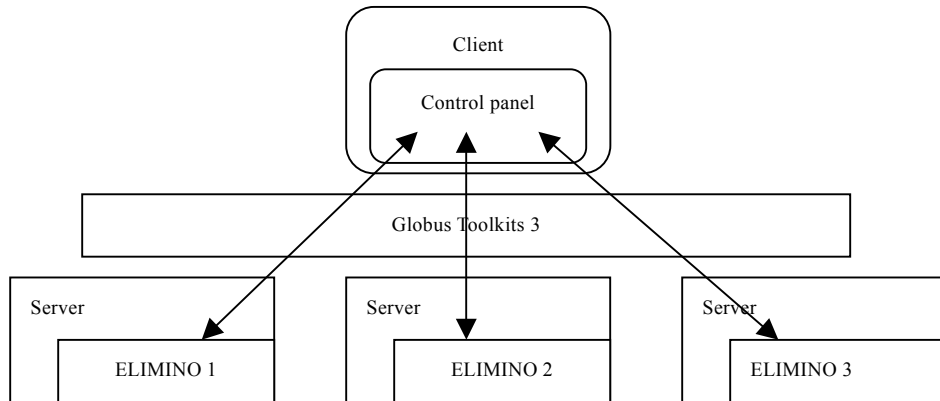


Fig.2 Framework of distributed computing for Wu's method

图 2 分布式吴方法计算框架

由第 3 节可知,大整数和多向式的消息传递可由字符串的传递来完成,采用基于我们提出的新表示方法的转换法,在传输前后完成符号计算系统内部多项式数据类型与压缩之后所产生的字符串之间的相互转换.这样设计有两个主要原因,一是考虑到在互联网上进行数据传输,需要确保通用性;二是考虑到传输的效率,采用我们的新表示方法进行多项式和大整数的数据压缩.

吴方法计算的特点是计算量大,分支多,任务平衡分配困难.因此我们选择了主从式星型计算模式作为吴方法并行计算的编程模式.其中的主进程负责控制整个计算的流程,分割并发送计算任务给从进程,并从这些进程收集计算结果.

在主从模式中,一个主要的问题就是从进程的负载平衡问题.理想的状态是,每个从进程一直忙于计算,直到整个计算任务完成.当然这种状态是很难达到的,一个主要原因就是不同的进程在不同的节点上计算花费的时间是不相等的,导致尽管每个进程的程序代码相同,还是会出现一个或几个进程已经完成了计算任务,而其他进程还在忙于计算的现象.为了提高并行计算的效率,必须采取一个高效的负载平衡的方法.

现在我们来讨论有关吴方法分布式计算的负载平衡问题.一般而言,有两种基本的方法可以采纳,一种是动态的,一种是静态的.动态是指当从进程处于就绪状态时,再分配任务给它.这种方法可以减少所有从进程的闲置时间,但却是以增加通信量为代价的,而且有可能出现多个从进程同时就绪的状态,此时也就有可能出现进程闲置的现象.静态是指主进程在一开始就“平均”分配任务给各个从进程.但这种“平均”是不太可能预先知道的.这种方法减少了通信的时间,但是很有可能导致多个进程闲置的现象.在目前的吴方法并行计算环境中,我们采取了动静结合的方法,初始按静态法则将大部分计算任务进行分配,然后采用动态方式分配剩余任务.

在该并行计算环境中,用户既可以使用/调用已经实现的吴方法高性能计算函数,也可以利用该环境提供的丰富函数自己编写相应的分布式计算程序,以达到高性能计算的目的.文献[11]给出了在该环境中进行多项式特征列计算的实验结果.

我们在基于 Globus Toolkits 3 的网格环境下也实现了一个分布式符号计算系统,用户通过我们提供的一套类似 MPI 的并行编程接口及其实现,可以利用多个广域网上的 ELIMINO 进行分布式吴方法计算.

该环境提供的基本接口函数有以下几类:

### (1) 节点信息接口

- getnodeid() //获取节点编号
- (2) 与节点进行数据传输接口
- upload(localfile,remotefile) //上传数据文件
- download(localfile,remotefile) //下载数据文件
- localfile:本地文件绝对路径;remotefile:远端目标文件相对路径
- (3) 任务执行接口
- execute(commandline,type) //阻塞方式执行
- synexecute(commandline,type) //非阻塞方式执行
- isjobover() //非阻塞方式下查询任务是否结束
- waitexecute() //非阻塞方式下等待任务执行结束
- String getresult() //获取任务执行结果
- commandline:命令字符串;type:字符串类型
- (4) 节点交互接口
- sendfile(sorucefile,targetid,targetfile) //向另一个节点发送文件
- receivefile(targeted) //从另一个节点接收文件
- sendvalue(targeted,message) //向另一个节点发送数据
- receivevalue(targeted) //从另一个节点接收数据
- sourcefile:源节点文件相对路径;targeted:目的节点号;targetfile:目的节点文件相对路径
- message:发送的消息
- (5) 回报信息接口
- reportstatus(message) //向客户端报告任务执行情况
- message:回报的信息

表 1 给出了利用我们实现的编程接口在由 4 个普通 PIII 800 的 PC 机组成的网格环境中进行多项式特征列计算的实验结果.

**Table 1** Experimental times for distributed characteristic set computation over grid

表 1 网格环境下多项式特征列计算实验数据

No.	Serial time	2 CPU		3 CPU		4 CPU	
		Time	Speed-Up	Time	Speed-Up	Time	Speed-Up
1	4.906	3.337	1.47	2.441	2.01	2.681	1.83
2	9.034	6.546	1.38	5.377	1.68	3.980	2.27
3	64.32	42.32	1.52	37.61	1.71	25.42	2.53

我们发现,在网格环境下,当计算节点增加时,加速比并不能保证稳定增加(例如计算实例 1),这是因为计算问题的特殊性,计算分支只有 3 个,用 4 个节点进行计算当然不会得到更高的加速比.另外,网络与计算资源的负载都是动态变化的,即使同一个计算问题、同样的计算节点和计算代码,所测得的计算时间差距也还是非常大的.但是当问题规模不断增加时,仍然会得到一个比较合理的随节点数目增加的加速比.

## 6 进一步工作

吴方法是由我国科学家吴文俊院士开创的一个新兴研究领域,它的目的是将数学计算与证明的过程标准化、规范化,可以让计算机来实现.随着吴方法在越来越多领域的应用和推广,所面临的计算问题也越来越复杂.考虑到吴方法“分而治之”的思想非常适合分布式计算,本文将高性能计算技术引入到吴方法的计算过程中,给出一种既可以在集群环境下也可以在网格环境下实现的分布式吴方法计算模型.文章在分析了吴方法分布式计算需求和吴方法分布式计算算法的基础上,给出了符号计算基本数据类型:大整数和多项式的消息传递方法,并简单给出了一个集群环境下基于符号计算软件系统 ELIMINO 和消息传递标准规范 MPI 的并行吴方法计算环境和网格环境下基于 Globus Toolkits 3 的分布式吴方法计算环境的设计与实现.

分布式吴方法计算的研究为有关吴方法的实际应用问题给出了一个高性能求解的环境.该模型也可以用到其他的复杂符号计算问题的求解当中去.进一步的工作主要包括符号计算问题的负载平衡计算研究,特别是有关吴方法的符号计算问题的负载平衡研究;吴方法在网格环境下的计算研究等.

#### References:

- [1] Wu WT. On the decision problem and the mechanization of theorem in elementary geometry. *Scientia Sinica*, 1978,21:159-172.
- [2] Wu WT. *Basic Principle of Mechanical Theorem Proving in Geometries (Part on Elementary Geometries)*. Beijing: Science Press, 1984 (in Chinese).
- [3] Wang DK, Zhi LH. Software development in MMRC. In: Yang W, Wang D, eds. *Proc. of the ATCM'95*. Singapore: Academic Press, 1995. 234-243.
- [4] Wang D. An implementation of the characteristic set method in Maple. In: Pfalzgraf J, Wang D, eds. *Automated Practical Reasoning: Algebraic Approaches*. New York: Springer-Verlag, 1998. 187-201.
- [5] Wang P. Parallel polynomial operations on SMPs: An overview. *Journal of Symbolic Computation*, 2001,11(1):377-396.
- [6] Rayes M, Wang P. Parallel GCD for sparse multivariate polynomials on shared memory multiple processors. In: *Proc. of the PASCO'96*. Washington: IEEE Press, 1996. 326-335.
- [7] Ajwa I, Wang P. Applying parallel/distributed computing to advanced algebraic computations. In: *Proc. of the 1997 IEEE National Aerospace and Electronics Conf.* Washington: IEEE Press, 1997. 156-164.
- [8] Lin DD, Wu YW, Yang H. Parallel computation for polynomial GCD. In: Zheng WM, Yang GW, Wu YW, eds. *Proc. of the China National Computer Conf.* Beijing: Tsinghua Press, 2003. 900-905 (in Chinese with English abstract).
- [9] Wu YW, Yang GW, Lin DD. On the parallel computation for characteristic set method. *Chinese Journal of Electronics*, 2004,18(3):383-388.
- [10] Wang P. Parallel univariate polynomial factorization on shared-memory multiple processors. In: Caviness B, Johnson J, eds. *Proc. of the ISSAC'99*. New York: ACM Press, 1999. 145-151.
- [11] Lin DD, Liu J, Liu ZJ. Mathematical research software: ELIMINO. In: Li ZB, ed. *Proc. of the ACM'98*. Lanzhou: Lanzhou University Press, 1998. 107-114.
- [12] Yang H, Liu Z, Lin D. Development of an object-oriented number system. *Mathematics Mechanization Research*, 2000,18: 212-219.

#### 附中文参考文献:

- [2] 吴文俊. 定理机器证明基本原理(初等几何部分). 北京: 科学出版社, 1984.
- [8] 林东岱, 武永卫, 杨宏. 并行多项式最大公因子计算. 见: 郑纬民, 杨广文, 武永卫, 编. 2003 年中国计算机大会论文集. 北京: 清华大学出版社, 2003. 900-905.