文章编号: 1006-2475(2007)09-0001-04

网格技术在全球分布计算计划 GIMPS中的应用研究

王 婧[1,2], 曾文华[1,2]

(1.厦门大学软件学院,福建 厦门 361005; 2.智能信息技术福建省重点实验室,福建 厦门 361005)

摘要:旨在寻求新梅森素数的大互联网梅森素数搜寻计划 GIMPS(Great Internet Mersenne Primes Search)[1]在网格技术的协助下已找到第 44个梅森素数。GIMPS是唯一的全球分布计算计划,真正的虚拟组织[2]。梅森素数的计算具有指数复杂性,随着 p达千万级,所需计算时间须以千、万计算机年计。本文基于梅森素数搜索历程中的原理、技术和算法,探讨网格技术给 GIMPS计划带来的突破性进展。

关键词:网格技术; PC网格; 素数; 梅森素数; GIMPS; 虚拟组织; Lucas-Lehmer测试
中图分类号: TP393 文献标识码: A

Research on Application of Grid Technology in Global Distributed Computing Project GIMPS

WANG Jing[1,2], ZENG Wen-hua[1,2]

(1. School of Software, Xiamen University, Xiamen 361005, China;
2. Key Laboratory for Intelligent Information Technology of Fujian Province, Xiamen 361005, China)

Abstract: The Great Internet Mersenne Primes Search (GIMPS)[1] project has found 44 Mersenne Primes up to now with the help of grid computing technology. GIMPS is the only global distributed computing project, the absolute virtual organization (VO)[2]. The computing of Mersenne Primes has exponential complexity. As the index p comes to ten millions level, the needed time should be weighed by thousands or ten thousands computer years. This article further discusses the breakthroughs the grid technology brought in, based on the theories, technologies and algorithms during the Mersenne Primes searching.

Key words: grid technology; PC grid; prime number; Mersenne Primes; GIMPS; virtual organization; Lucas-Lehmer testing

0 引 言

2006年 9月 4日,美国密苏里州立大学数学家库珀和化学家布恩领导的研究小组发现了目前已知的最大梅森素数,表作 $2^{32582657}-1$。此数是 2000多年来人类发现的第 44个梅森素数,达 9808358位。这是一项新的世界记录,是该研究小组使用 GIMPS计划动用 800多台计算机发现的。法国和西班牙的有关专家用了 6天的时间验证了这一发现。该成果标志着人类挑战智力极限的又一次胜利[3]。

梅森素数是指形如 2^p-1的素数(且 p为素数),记为 M_p。对于梅森素数的最初研究要追溯到公元前 300多年,古希腊数学家欧几里德在其著作《几何原本》中讨论完全数时开创了研究 2^p-1型数的先河,由于 2^p-1是偶完全数的一个因子,因此被定义为梅森数的出处[4]。17世纪法国著名数学家,法兰西科学院奠基人马林·梅森(Marin Mersenne, 1588-1648)是最早系统而深入地研究 2^p-1型数的人,为了纪念他,数学界就将其命名为"梅森素数",并以 M_p记之。如果一个数只能被 1和自身整除,那么这个数叫做素数。如果一个梅森数 M_p是素数,那么它称为梅森素数。梅森素数的计算具有指数复杂性,随着指数 p的增大,运算量呈指数增加。每一个梅森素数的产生都艰辛无比,它不仅需要高深的理论、纯熟的技巧,还要进行艰巨的计算。1996年美国数学家 G. Woltman[5]编制了一个梅森素数寻找程序,并把它放在互联网上供数学家和数学爱好者免费使用,这就是著名的"互联网梅森素数大搜索"项目(GIMPS)。该项目在分布计算及网格技术的帮助下取得了突破性进展,用于解决跨实际组织、跨地域分布、动态构成和可控可变的虚拟组织资源共享与合作求解的网格技术[6],通过使用"远处"非本地大量普通计算机的闲置时间来获得相当于超级计算机的能力。研究显示,网格分布计算的最终结果不但与全部问题都在一个超强能力计算机上计算的结果相同,而且当网格资源结点须运行其他更紧要任务时,该搜索程序可方便地以后台方式运行而不影响紧要任

收稿日期: 2007-03-15
基金项目: "985工程"智能化国防信息安全技术科技创新平台项目(0000-X07204)
作者简介: 王婧(1981-),女,新疆乌鲁木齐人,厦门大学软件学院硕士研究生,研究方向:网格资源调度算法;曾文华(1964-),男,教授,博士生导师,博士,研究方向:人工智能,网络计算,嵌入式系统,计算机体系结构,智能控制。

务正常运行, 且停止和重启都很容易[7]。截止目前, GIMPS计划在网格计算技术的帮助下于十年间已连续获得 10个梅森素数(如表 1所示)。现在, 世界上 150多个国家和地区近 12万人参加了这一国际合作项目, 并动用 28万多台计算机联网来进行网格计算, 运算速度已达每秒 250万亿次[3]。本文将基于梅森素数搜寻原理、技术及算法, 对网格计算这一助力作进一步的探讨和研究。

表 1 GIMPS(PrimeNet)项目实现的梅森素数一览表[8-10]

序号	梅森素数	位数	分布式/网格结点数	发现时间, 国别	发现者/验证者
35	$M_{1398269}$	420921		1996. 11. 13, France	Joel Armengaud
36	$M_{2976221}$	895933		1997. 7. 24, U. K.	Gordon Spence
37	$M_{3021377}$	909526		1998. 1. 27, U. S.	Roland Clarkson
38	$M_{6972593}$	2098960	> 21500	1999. 6, U. S.	Nayan Hajratwala
39	$M_{13466917}$	4053946	205000	2001. 11. 14, Canada	Michael Cameron
40	$M_{20996011}$	6320430	211000	2003. 11. 17, U. S.	Michael Shafer
41	$M_{24036583}$	7235733	240000	2004. 5. 15, U. S.	Josh Findley
42	$M_{25964951}$	7816230	> 240000	2005. 2. 18, Germany	Martin Nowak
43	$M_{30402457}$	9152052	260000	2005. 12. 15, U. S.	Cooper, Boone, Woltman, Kurowski, et al
44	$M_{32582657}$	9808358	280000	2006. 9. 4, U. S.	

1 GIMPS的数学理论及算法描述

1.1 相关数学理论[4, 10]

定义 1 一正整数 n叫做完全数, 如果 n等于它的各个正除数之和, 这里的正除数不包括 n自己。

定义 2 若 2^p-1为素数且 p为素数, 则 2^p-1被叫做一个梅森素数。

定理 1 k是偶完全数的充要条件是 $k=2^{p-1}(2^p-1)$, 且 2^p-1为素数。

定理 2 若 2^p-1为素数, 则 p亦然(但逆命题不成立, 举一反例: $M_{11}=23*89$)。

推论 1 令 a与 p是大于 1的整数, 若 a^p-1是素数, 则 $a=2$且 p为素数。

定理 3 令 p和 q为素数, 若 q为梅森素数 2^p-1的一个因子, 那么 q必定是 $2kp+1$的形式, 且 $q=1$或 $7(mod\ 8)$。

定理 4 令 $p=3(mod\ 4)$为素数, 则 $2p+1$也是素数当且仅当 $2p+1$整除 M_p。

1.2 梅森素数分布规律

探索梅森素数的分布规律似乎比寻找新的梅森素数更为困难。英国、法国、印度、美国、德国的数学家在长期的探索中都间或地提出一些猜想, 但都以渐近的表达式给出。而中国数学家及语言学家周海中通过运用联合观察法和不完全归纳法, 从已知的梅森素数出发, 于 1992年首次给出了梅森素数分布的准确表达式[11-13], 即:

当 $2^{2^n}<p<2^{2^{n+1}}$ ($n=0, 1, 2, 3, \cdots$)时, M_p有 $2^{n+1}-1$个数是素数; 并据此做出了当 $p<2^{2^{n+1}}$时, 梅森素数的个数为 $2^{n+2}-n-2$的推论。为数论的发展做出重大贡献, 被国际上称为"周氏猜测"。

1.3 搜索算法

由前述定理 2知, 梅森素数 $M_p=2^p-1$的指数 p集合为素数的子集, 于是寻求梅森素数的计算机算法第一步将从指数着手。即: 先生成一个用于测试的素数指数列表。

方法 1(素性定义法): 从 1到 N逐个判断

```
for( i= 1; i< = N; i++ )
for( j= 2; j< i j++ )
 i逐个与 j求余, 如果都不为 0, 则 i为素数
```

这是最简单、最传统的求解素数算法, 它仅限于 N较小的情况。

方法 2(厄拉托森斯(Eratosthenes)筛法)[4]此法提出于公元前 240年。具体思想为: 删除所有小于等于 N的平方根的素数的倍数, 剩下的即是素数。该方法为目前求解 $N(N<10^7)$以内素数的最有效方法。对于较大 N, 可以使用分段筛选, 算法效率极高, 在时间上和空间上都占有绝对优势。

得到素数指数列表后, GIMPS将通过对 M_p执行不同途径的操作以得到梅森素数。采用的均为大范围分布式网格计算。

途径 1(梅森素数判定法——Lucas-Lehmer素性测试)[10]该方法是迄今为止已知的检测梅森素数素性的最好方法。由 Lucas于 1878年发现, 并由 Lehmer于 1930年作了改进而得名。该方法基于循环数列的计算, 其原理为: 对素数 p, 2^p-1为素数当且仅当 2^p-1整除 $S(p-1)$, 这里 $S(1)=4$, $S(p+1)=S^2(p)-2$。

以下是用 C语言实现的实用版 Lucas-Lehmer素性测试算法[2]。

```
Lucas-Lehmer( int p)
{
 int s= 4;
 int i, sl;
 sl= 2** p- 1;
 for( i= 3; i< = p; i++ )
 s= ( s** 2)% sl;
 return( s= = 0? 1: 0);
}
```

例如: 测试 2^7-1是素数的过程如下($2^7-1=127$)

S0= 4

S1= (4* 4- 2) mod 127= 14

S2= (14* 14- 2) mod 127= 67

S3= (67* 67- 2) mod 127- 42

S4= (42* 42- 2) mod 127= 111

S5= (111* 111- 2) mod 127= 0

Lucas-Lehmer素性测试方法通过对第一步生成的素数指数列表中的 p逐个进行测试, 找到新的梅森素数。

途径 2(用修正的 Eratosthenes筛法来试验分解因子)[14-15]根据前述定理 3, 利用一个二进位表示一个可能的 2kp+1形式的因子。这种筛法可排除大约 40000以下的素因子 2kp+1。同样, 表示除以 8的余数为 3或 5的素因子 2kp+

1所代表的二进位被清除,这个过程排除大约 95%可能的因子。而剩下的可能因子,使用下述高效的算法进行测试,用于判定一个数是否能整除 2^p-1,从而找到新的梅森素数。

例如,测试素数 q= 89是否能整除 $2^{11}-1$。11的二进制表示为 1011,从最左边的二进位 1开始,重复以下步骤:平方→删除该二进位→如果该位为 1,将平方后的值乘以 2→将结果与 47求余,如果最终结果余数为 1,则 q为 M_p的一个因子(如表 2所示)。

表 2　测试素数 **q= 89**是否能整除 $2^{11}-1$

平方	删除最左边二进位	若需要即乘以 2	结果与 47求余
1* 1= 1	1…… 011	1* 2= 2	2% 89= 2
2* 2= 4	0…… 11	no	4% 89= 4
4* 4= 16	1…… 1	16* 2= 32	32% 89= 32
32* 32= 1024	1……	1024* 2= 2048	2048% 89= 1

途径 **3**(波拉德(Pollard) p-1法分解因子)[14-16]这一方法被 GIMPS程序用来搜索因子,避免了进行素性测试的花费,用于搜索梅森素数甚至更高效。根据前述定理 3,如果 q是某数的一个因子,则 q只能是 2kp+ 1的形式,如果 q-1是高度复合的(即 q-1只有小因子),即 2kp+ 1中的 k是高度复合的,p-1方法就可以找到因子 q。该方法的执行步骤是,在第一阶段挑选一个边界 B1,只要 k的所有因子都小于 B1(称 k为 B1平滑,B1-smooth),p-1方法就能找到 q。首先计算 E= ∏(小于 B1的所有素数),然后计算 x= 3E* 2* p,最后,检查 x-1和 2p-1的最大公约数就可以知道是否找到一个因子。第二阶段挑选第二边界 B2,如果 k在 B1到 B2之间刚好有一个因子,而其它因子都小于 B1,就可在第二阶段找到 q因子。不过这个阶段要使用大量的内存。用于 GIMPS程序中寻找一些给人印象深刻的因子。

2　网格技术协助实现 GIMPS计划

跨实际组织、跨地域分布、动态建立的个人或团体联盟叫做虚拟组织[17],GIMPS即是这样的虚拟组织[2]。它使用加州软件科学家 Scott Kurowski[18]开发的分布计算平台 PrimeNet[19-20](全球分布互联网研究计算系统)及其核心技术和资源进行 GIMPS计划的网格计算实施。GartnerGroup的研究显示,当今 PC机能力的 95%被浪费[2],CPU性能的改进使 CPU时钟频率稳步升高也促成了其不断膨胀的闲置处理器时间。而另一方面,在实验科学、数学、密码学及其它领域,个人或团体需要大量的计算能力来解决复杂问题。一边是大量闲置和浪费的资源,另一边是计算能力的严重不足。网格计算技术就是应这种需求而诞生的,它是分布计算发展的高级阶段,它构建容易,操作经济,工作效率高,将互联网从通信和信息交互的平台提升到资源共享和协同工作的平台[21]。使巨型超高速并行计算机难于处理的问题变得可能解决或容易解决。网格系统总体性能的自动改进(通过提升结点性能)也使得曾一味专注于提高大型超级计算机性能的专家们将重心放在更亟待解决的问题上[7]。

GIMPS项目中网格技术的应用表现为 PC网格计算。PC网格基于其成本低、广泛的可用性以及工作独立性等优点被成功地用于 Internet上任何大粒度分布式超级计算应用。它的主要特点在于:通过置分配给网格 PC结点的子任务在其任务执行集合中优先级最低而不致影响该资源结点执行更紧要的计算任务,并且它会有效采用负载均衡来减少为保证处理能力而引起的高负载,强容错能力带来了可靠性,传输数据量少使得测试容易[7]。PC网格的分布式计算结果与大型超高速计算机运行的结果相同。在对第 42个梅森素数素性测试时,PrimeNet峰值计算等价于 472台 CrayT16超级计算机,或 236台 Cray的顶级 T932,成为 GIMPS成功的根本保证[2]。

前文所述梅森素数搜寻的三个途径都须基于 PC网格分布式计算实现。PC网格上分布计算流程如图 1所示。

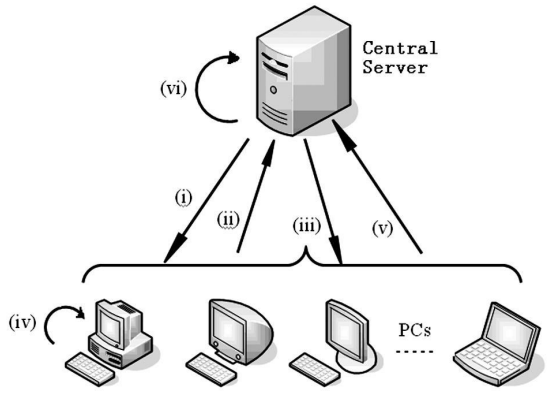

图 1　PC网格计算的运作机制[7]

(1)参与计算的 PC结点从 Web服务器上下载专用软件并安装于本地;

(2)本地 PC结点向中央服务器请求发送程序和数据;

(3)中央服务器将大型计算应用问题按合理的大小分解成多个小的任务(即由并行处理的程序和数据组成的工作单元)发送给不同的网格 PC结点;

(4)网格 PC结点将接收到的子任务作为优先级最低的任务在其 CPU空闲时被执行;

(5)当处理完毕,网格 PC结点上的专用软件将结果送回中央服务器并请求新的数据(第(3)到第(5)步将循环执行直到整个计算应用结束);

(6)中央服务器收集并整理,保存从参与计算的 PC结点上获得的处理结果,必要时作一些处理。

在设计时,必须保证网格 PC结点需下载和上传的数据量尽可能少,一方面减轻了用户的负担,另一方面不但可以节省网格并行计算的总时间,而且结果的正确性也容易验证。

现将 GIMPS计划中寻找指数为 4000到 5000之间可能的梅森素数的 PC网格计算过程在中央控制器(主程序)中安排示例如下[4]:

第一步:求出 4000~ 5000内的 N个素数(可用前述筛法计算,从数据库获得);N= 119(即有 119个素数),将其存放在 p[i](i= 1, 2, …, 119)中;

第二步:将 p[i]中素数分别发送到 N= 119个网格结点分布执行;

/* 注:第 i个外围网格结点验证 $2^{p[i]}-1$是否为素数 (可用 Lucas-Lehmer方法),将验证结果发送回主程序的 r[i]中; * /

第三步:若 r[i]

为 0,则 $2^{p[i]}-1$不是梅森素数;

为 1,则 $2^{p[i]}-1$是梅森素数。

本例结果: p[33] = 4253为第 18个梅森素数 (由返回值 r[33] = 1); p[52] = 4423为第 19个梅森素数 (由返回值 r[52] = 1)。

3 搜索梅森素数的意义

早期探寻梅森素数似乎只是为探寻完全数[22],而当今已有了丰富的意义。

古希腊数学家欧几里德证明了素数是无限的,而是否有无穷多个梅森素数还是个不解之谜,探寻梅森素数是发现已知最大素数的最有效途径。

探寻梅森素数是测试计算机运算速度及其它功能的有力手段(第 34个梅森素数 $M_{12257787}$就是 1996年 9月美国克雷公司在测试其最新超级计算机的运算速度时得到的)。

探寻梅森素数还需要素数判别和数值计算的理论与方法以及高效的程序设计技术,于是它还推动了数论的研究及计算数学和程序设计技术的发展。

大的梅森素数还可以用于现代密码设计领域,素数越大,密码被破译的可能性就越小。

梅森素数搜索计划为分布计算和网格技术的研究提供了理想模型。

研究梅森素数还可以检验人们的智慧和运算能力,从某种意义上说,它标志着一个国家的科技发展水平。

4 结束语

网格技术用于执行 GIMPS计划,其搜索算法和调救机制上还存在很多方面须改进,下一步的工作将是根据上文所给出的梅森素数分布规律,进一步优化算法,以及合理配置网格资源,将网格任务调度策略[23]应用于计算当中。

参考文献:

[1] George Woltman. The Great Internet Mersenne Primes Search[DB/OL]. http://www.mersenne.org/, 2006-09-11.

[2] 高全泉. “大互联网梅森素数寻求 (GIMPS)”研究计划进展[J]. 数学的实践与认识, 2005, 35(10): 166-171.

[3] 钟勇, 杨玲. 美科学家发现迄今最大梅森素数[DB/OL]. http://210.34.4.20/news/detail.asp?serial=43444&keys=网格&key=计算机, 2006-09-13.

[4] 高全泉. 梅森素数研究的若干基本理论及意义[J]. 数学的实践与认识, 2006, 36(1): 232-238.

[5] George Woltman. George Woltman's Personal Information and His Primes[DB/OL]. http://primes.utm.edu/bios/page.php?lastname=woltman, 2006-03-28.

[6] [美] Ian Foster, Carl Kesselman. 网格计算: Blueprint for a new computing infrastructure[M]. 金海, 袁平鹏, 石柯译. 北京: 电子工业出版社, 2004: 275-293.

[7] Makoto Tachikawa. PC grid computing——Using increasingly common and powerful PCs to supply society with ample computing resources[J]. Science & Technology Trends Quarterly Review, 2006(18): 45-53.

[8] 方程. 魅力无穷的梅森素数[J]. 世界科学, 2004(7).

[9] 曾小宁, 齐德昱, 李同武. 一种计算网格环境的研究与实现[D]. 华南理工大学硕士学位论文, 2005: 53, 60.

[10] Chris Caldwell. Mersenne Primes: History, Theorems and Lists; The Lucas-Lehmer Test and Recent History[DB/OL]. http://primes.utm.edu/mersenne/index.html, 2006-09-20.

[11] 周海中. 梅森素数的分布[J]. 科技导报 (粤版), 1992(3): 68.

[12] 周海中. 梅森素数的分布规律[J]. 中山大学学报: 自然科学版, 1992, 31(4): 121-122.

[13] 岑成德. 关于梅森素数分布性质的猜想[J]. 中山大学学报, 1999, 38(3): 107-108.

[14] Chris Caldwell. Selected Theorems and Their Proofs[DB/OL]. http://primes.utm.edu/notes/proofs/, 2006-09-25.

[15] Chris Caldwell. Sieve of Eratosthenes[DB/OL]. http://primes.utm.edu/glossary/page.php?sort=SieveOfEratosthenes, 2006-09-15.

[16] Chris Caldwell. 波拉德 (Pollard) (P-1)方法[DB/OL]. http://www.frenchfries.net/paul/factoring/theory/pollard_pr1.html, 2006-09-20.

[17] 高全泉. 网格: 面向虚拟组织的资源共享技术[J]. 计算机科学, 2003, 30(1): 1-5.

[18] Scott Kurowski. Scott Kurowski's Home Page[DB/OL]. http://www.scottkurowski.com/, 2006-10-19.

[19] 高全泉. 关于网格及其它分布计算技术的若干问题的讨论[J]. 计算机科学, 2003, 30(2): 17-21.

[20] Scott Kurowski. PrimeNet V5 Server Web API[DB/OL]. http://www.scottkurowski.com/v5/v5webAPI.html, 2004-01-13.

[21] University of California. SETI@Home Project[DB/OL]. http://setiathome.berkeley.edu/, 2007-01-17.

[22] 方成. 魅人的梅森素数[J]. Knowledge Is Power, 2006(12).

[23] G Allen, T Dramlitsch, I Foster, et al. Supporting efficient execution in heterogeneous distributed computing environments with cactus and globus[C]//Proceedings of SC 2001, November 2001.