数理解析研究所講究録 1295 巻 2002 年 29-34

GAPパッケージ ParallelGAP の計算実験

宮本泉 (Izumi Miyamoto)*
山梨大学 工学部 (Yamanashi Univ.)

群計算を中心とする代数的数式処理ソフトウェアパッケージ GAP[5] のシェアパッケージ ParGAP (Parallel GAP/MPI)[4] を使用した並列/分散計算実験の報告です。特に、楕円曲線法による整数の素因数分解で新記録の結果が得られました。実験環境は、筆者の所属するコンピュータ・メディア工学科の教育用パソコン 75 台です。授業・演習で学生が使用するのが用途ですから、重たいプロセスが長時間走っていることは稀で、プロセスの優先度を十分下げても並列計算実験をすることは可能ですが、もちろん本来の使用目的の妨げにならないように注意して使用する必要があります。また、その様な環境ですから、普通の 100BaseLAN の通信網である上に、学生の使用状況によってネットワークが不安定な状態になることも考えられます。実際、本実験では、小さい場合の問題を使った試行実験では計算が完了しても、本来の目的のための実験では、原因を特定できないまま、途中でプロセスが死んでしまった場合が多々ありました。さらに、セキュリティを考慮すると、ネットワークによる接続には障壁を高くしておかなければなりません。しかし、これはネットワークを利用した並列計算には困難をもたらします。コンピュータシステムを設計した業者が設定して、学科のシステム管理者も認識していない障壁も中にはあったようです。

並列計算の場合は、通常、通信が安定していて切れないことが前提条件となりますが、本環境ではそれは保証されません。しかし、多くのコンピュータ設備は、上に述べたような状況になっていると思います。したがって、本実験の目的は、このような一般的な状況で行う並列/分散計算の経過の報告とそのような状況に対処する工夫の紹介、適した並列アルゴリズムの研究にあります。実験にあたっては、計算機システムの目的を考えて、システム管理者に本実験のために便宜を図ることは依頼しないようにしました。

使用しているパソコンは Pentium III, 800MHz, Memory 256MB, OS は Linux です。プロセスの実行には、計算時間の限度が CPU time で 60 分、使用可能メモリ 100MB の制約がありましたが、これはユーザーで変更可能でしたので、計算時間の限度は適宜延長しました。メモリに関しては、X Window System 関係で半分近くが常時使用されていますので、増やすことは実際的に無理なばかりかでなく、むしろ、自粛して使用する必要があると思われます。

ネットワーク接続には ssh を使用しています。並列計算ソフトウェアパッケージのマニュアルで、一番重点が置かれているのが接続に関する注意書です。学科の計算機室の教育用システムでは比較的簡単に並列計算パッケージが動くようになったのですが、研究室のサーバーマシンと計算機室のマシンを使用する設定ではまだ動かないのが現状です。ssh デーモンの設定はシステム管理者の管轄ですから、許される範囲の接続を利用して実験を行っています。特に、子プロセスで計算に長時間かかり返信がないと通信が切れるように設定されています。その時間制限は 10 分程度になっているようです。これに対処するために、ParGAP から本来の計算プログラム以外に 5 分間隔程度に定期的に短い文字列を echo する外部プログラムを起動させています。この文字列はウインドウ画面には表示されますが、ParGAP のプログラムの送受信する内容には含まれないことを利用しています。

花木彰秀氏と筆者は共同でコンピュータによるサイズの小さなアソシエーションスキームの分類をしてきました [6, 7]。これを並列/分散計算で行うことが本実験の動機です。アソシエーションスキームを生成す

*izumi@esi.yamanashi.ac.jp

るプログラムの最新バージョンは花木氏による C-プログラムです。本実験のアソシエーションスキームの計算は、花木氏のプログラムを改造した上で ParGAP をインタフェースとして使用して並列計算を行っています。しかし、現状では通常の計算を越える新しい結果は得られていません。

この実験における新しい計算結果は、楕円曲線法を使用した整数の素因数分解です。アソシエーションスキームの分類は答えをすべてもれなく、あるいは、答えは一つもないことを求めるというタイプの問題です。この様な問題では、並列計算のプロセスが途中で死んでしまった場合、再度の計算を途中の段階から始めるのは困難です。チェックポイントを設定して継続計算を可能にしても、答えの信頼性が低くなることは避けられません。これに比較して、答えを一つでも、あるいは、なるべく多く見つければ良いというタイプの問題では、プロセスの中断に対する管理は楽になります。計算実験を、まず、後者のタイプの問題で行うことを考えて素因数分解を取上げました。楕円曲線法の素因数分解プログラムは P. Zimmermann の GMP-ECM[9] を使用し、ParGAP をインタフェースとして各マシンに計算を実行させました。

1 アソシエーションスキームの計算

集合 X 上のアソシエーションスキーム $(X, \{R_i\}_{0\le i\le d})$ (cf. [1]) の隣接行列 $A_0, A_1, \cdots, A_d$ に対して、*relation matrix* を $A = 0A_0 + 1A_1 + 2A_2 + \cdots + dA_d$ で定める. $n = |X|$、すなわち、行列 A_i のサイズをアソシエーションスキームの *order* という。アソシエーションスキームの構成は、可能な relation matrix をバックトラック法により作ることにより行います。A_i の各行各列の 1 の個数は一定で、それを v_i とおくと、relation matrix A は各行各列に v_i 個の i を含む行列になります。アソシエーションスキームの構成プログラムへの入力は $[v_1, v_2, \cdots, v_d]$ と条件 $A_{i^*} = {}^tA_i$ で定まる番号の列 $[1^*, 2^*, \cdots, d^*]$ です。入力の前者は整数 $n-1$ の分割になります。一つの order を固定して考えるとき、代数的組合せ論などにより、可能な入力パターンの個数を減らすように工夫しますが [7]、それでもかなりの個数になります。したがって、ここで考えられる並列/分散計算は

- order を固定して、可能な入力パターン毎に別のマシンで計算する。
- 入力パターンを固定して、バックトラック法の計算を並列化する。

の 2 通りがあります。前者の場合は並列計算として工夫することは取立ててありませんが、通常の計算では時間がかかりすぎて実際上できないときでも答えを得ることができる可能性があります。本実験ではまずこの方法から始めました。入力パターン 1477 個の order 24 の場合の実験経過の概要は下の通りです。

24 次のすべてのパターンの並列計算実験

```
Mon Jul 23 15:25:12 JST 2001
" 1  master -> slave1 ",
" 2  master -> slave2 ",
" 5  master -> slave5 ",
(中略)
" 1422  master -> slave28 ",
" 1379  slave23 -> master  3 ", (実時間で 3 秒かかっている)
" 1424  master -> slave23 ",
" 1351  slave48 -> master  4 ",
" 1384  slave2 -> master  3 ",
 (中略)
" 1472  slave16 -> master  132 ",
```

```
" 1477  slave21 -> master  142 ",
" 1476  slave20 -> master  186 ",
開始 Mon Jul 23 15:25:12 JST 2001   終了 Mon Jul 23 17:33:30 JST 2001
```

今までのアソシエーションスキームの分類計算から、ごく小数の場合で非常に長時間を要していることがわかっています。したがって、第二の方法のバックトラック法の並列化が必要になります。アソシエーションスキームの構成は深さ優先によるバックトラック法により行っています。具体的には、辞書式順序を定めて relation matrix を順に構成していきます。relation matrix の第 1 行は辞書式順序により定まりますので、$(2,3)$ 成分から始めて $(2,n)$ 成分へ、さらに、$(3,4)$ 成分と進んで可能な値のすべてを決めて調べていきます。このことから、途中までの計算データはすべて relation matrix に書かれていることになります。

並列計算するために、上のバックトラック法で適当な行数を決めて、その行までの relation matrix をすべて構成します。これが並列化の前処理になります。次に、構成された relation matrix の残りの行の部分をそれぞれを別々のマシンで計算します。下の計算は入力パターン [11,11],[2,1] の場合に、先に 5 行目までのすべての relation matrix 164 個を計算したのちに並列計算を行った結果です。この場合の通常の計算に要する時間は 40 分弱です。

23 次の一つの入力パターン [11,11],[2,1] の並列計算
(バックトラック法を行列 5 行目までで実行、得られた 164 個の行列を並列計算)

```
"6 master -> slave6 ",
"10 master -> slave10 ",
"2 master -> slave2 ",
中略
"160 slave60 -> master  0",
"152 slave63 -> master  0",
"164 master -> slave48 ",
"163 master -> slave37 ",
"134 slave38 -> master  2",
"81 slave7 -> master  6",
"103 slave58 -> master  6",
"100 slave47 -> master  6",
中略
"31 slave31 -> master  119",
"96 slave43 -> master  155",
"32 slave32 -> master  215",

start : Thu Jul 26 10:53:29 JST 2001
end   : Thu Jul 26 10:57:11 JST 2001
```

実験からわかるように、バックトラック法による探索木のバランスがとれていないため、ここでも、ごく少ない場合に長時間を要しています。これ以上の工夫は通信回数が増えることなどから障害が起こって、成功しませんでした。出力は数十、数百の行列になる場合もありますが、多くの場合は一つも得られません。多くの行列が出力される場合は通信量が問題になり、多くの一つもない場合では短時間計算が終って一斉に返信が来ることが障害になっているかも知れません。

2 楕円曲線法による素因数分解

楕円曲線 $by^2 = x^3 + ax^2 + x$ を利用した整数の素因数分解法で、曲線の係数を変えて何回も素因数分解を試みる方法です。アルゴリズムはもう少し工夫されていますが、まだわからない素因数を法として考えた曲線上の点の成す群の位数が適当に bound として定めた数以下の素数および素数ベキの積であることを期待します。この bound を大きくとれば分解の成功する確率は高くなりますが、計算時間およびメモリー使用量も増えます。bound はこのことを考慮して適当に定めれば、毎回変える必要はありません。曲線の係数はプログラムが乱数で定めますので、毎回同じ入力で計算をすることになります。

したがって、並列/分散計算をするまでもなく、それぞれのマシンで繰返し計算するシェルプログラムをバックグラウンドで動かしておけば良いのですが、(75 台のマシンでこの様に実行することが現実的かどうかは別として) 一般的計算機システムにおける並列計算の実験用問題として、ParGAP[4] を使って各マシンに GMP-ECM[9] の計算実行の指令を出す、各マシンから計算結果を受取る、素因数分解が成功していれば計算は終了、失敗の場合は再度、計算実行の指令を出す、という方法で行いました。また、出力も簡単な計算データのメッセージに、成功したときには因数が追加されるだけですので、通信への負荷は非常に軽くなっています。

素因数分解した整数に関するデータは、下にまとめた通りです。楕円曲線法では、分解の困難さは素因数の大きさによります。楕円曲線法による素因数分解記録では、ここで得られた 10 進 55 桁の素因数は、約 2 年ぶりに前の記録を越える新記録になりました (cf. [2])。

```
素因数分解した整数 (112 桁)
(629^59 - 1)/(628 * 365377296628421249503829 71 * 13274814114538692574828847)
素因数 (55 桁)
72308801275268216939250595089720829527021330045523462 81
商 (57 桁)
599055788512257114593036852972837566170071937294830361923
実験時間
trial * time : 約 28000 曲線 * 55 CPU-min = 1000 CPU-days
                                        (1000/75 = 約 2 週間)(合計)
Bound1=45,000,000
sigma=267937500 (係数のための乱数の値)
Group order = 72308801275268216939250595125167554207112832071145584 64
            = 2^10*3*13*103*151*1123*12619*15649*26249*404197*4742809
              *25268183*41286269351 ( by P. Zimmermann )
```

注: 素因数分解した整数は三島さんのホームページ [8] の cyclotomic number からです。
what's new / history / errata (August 19, 2001)
(59 629 (13274814114538692574828847) (C 138))By Masaki Ukai (August03,2001)

なを、この方面の記録としては、一番強力と考えられている Special Number Field Sieve 法を使った Cabal グループによる 233 桁の分解 [3] で、2000 年 11 月に、総計約 2 万 CPU-time の計算で得られた結果があります。楕円曲線法による伊豆さんの結果もあります (cf.[2])。

毎回の計算の概要は下のようになりましたが、まとめると、計算時間や使用メモリーに制限のあることを知らずに bound の設定を大きくしすぎた失敗計算約 40,000 回の後に、約 28,000 回の計算で素因数分解に成功しました。この計算回数は並列計算で行った楕円曲線法素因数分解の計算回数の合計であって、下に報告するように、並列計算は何回も途中で止まってしまい、正常に終了することは稀でした。

```
        開始時間                              終了時間
Thu Aug 23 14:19:54 JST 2001 Thu Aug 23 16:44:08 JST (約1時間20分)
Limit1(=Bound1):=100,000,000; Curves:=150(計算する曲線数); (正常終了)
Curve no. 34 failed  UNIX_time : 3601 (秒  実時間)
Curve no. 5 failed  UNIX_time : 3602
Curve no. 11 failed  UNIX_time : 3601
.......
Curve no. 149 failed  UNIX_time : 3713
Curve no. 150 failed  UNIX_time : 3624 (最後の曲線)
Curve no. 146 failed  UNIX_time : 4903

Thu Aug 23 20:01:23 JST 2001  Aug 25 00:19 (約24時間)
Limit1:=100,000,000; Curves:=15,000;
.......
Curve no. 2071 failed  UNIX_time : 3602
Curve no. 2072 failed  UNIX_time : 3602  (約2000曲線の計算で途中終了)

以下、同様に、Limit1:=100,000,000; Curves:=15,000; の入力に途中終了。
Sat Aug 25 10:35:18 JST 2001  Aug 27 12:53 (約50時間20分)
Tue Aug 28 10:46:50 JST 2001  Sep  3 16:04 (約6日5時間20分)
Mon Sep  3 19:32:56 JST 2001  Sep  4 05:46 (約9時間)
(出力テスト)Tue Sep  4 18:56:21 JST 2001  Tue Sep  4 19:56:24 JST 2001
  その後の実験も、Limit1:=120,000,000など大きすぎて失敗計算
Tue Sep  4 20:27:27 JST 2001  Wed Sep 12 15:57:35 JST 2001
.......                    (計算トラブル?、正常終了)      (約7日19時間30分)
Thu Sep 13 10:24:35 JST 2001  Sep 17 21:56 (約4日11時間20分)
Tue Sep 18 19:31:39 JST 2001   Sep 18 19:55 ?? (約25分)
Tue Sep 18 20:09:38 JST 2001 Thu Sep 20 16:06:03 JST 2001 (約1日20時間)
.......                                          (計算トラブル、正常終了)
Curve no. 24000 failed by 57 crux10 UNIX_time : 0 14:38:25
Thu Sep 20 20:07:50 JST 2001   Sep 21 11:13 (約15時間)
```

以上の約40,000 curvesでは、時間超過で毎回の楕円曲線法の計算が途中1時間で打ち切りになっていたLimit1(=B1)が大き過ぎた。その他の障害もあり、計算が完了できたのは、下の約28,000 curves。

```
Fri Sep 21 11:27:22 JST 2001  Sep 25 18:04 (約4日6時間30分)
Limit1:=40,000,000;
Tue Sep 25 18:14:43 JST 2001  Oct  6 05:22 (約10日11時間)
Limit1:=45,000,000;
Sat Oct  6 11:54:11 JST 2001  Sat Oct  6 16:52:25 JST 2001 (約5時間)
.......                                                  (正常終了)
[ 278 (Curve no.),
"GMP-ECM 4c-m0, by P. Zimmermann (Inria), 16 Dec 1999, with contributions \
from T. Granlund, P. Leyland, C. Curry, A. Stuebinger, G. Woltman, JC. Mey\
rignac, A. Yamasaki, and the invaluable help from P.L. Montgomery. Input n\
```

```
umber is 433170059643319045108479137066914324703799675942540484239636962666\
36518589015401914361387760450667957626053058363 (112 digits) Using B1=4500\
0000, B2=43295692440, polynomial x^1, sigma=267937500 Step 1 took 2603480m\
s for 589347784 muls, 3 gcdexts Step 2 took 973400ms for 269164894 muls, 1\
86285 gcdexts ********** Factor found in step 2: 72308801275268216939250595\
08972082952702133004552346281Found probable prime factor of 55 digits: 72\
30880127526821693925059508972082952702133004552346281 Report your potentia\
l champion to Richard Brent <rpb@comlab.ox.ac.uk> (see ftp://ftp.comlab.ox\
.ac.uk/pub/Documents/techpapers/Richardo.Brent/champs.txt) Probable prime \
cofactor 599055788512257114593036852972837566170071937294830361923 has 57 \
digits " ]
```

参 考 文 献

[1] E. Bannai and T. Ito, Algebraic Combinatorics I: Association Schemes, Menlo Park, CA: Benjamin/Cummings, 1984.

[2] R. Brent, http://www.comlab.ox.ac.uk/oucl/work/richard.brent/factors.html, ftp://ftp.comlab.ox.ac.uk/pub/Documents/techpapers/Richard.Brent/champs.txt

[3] "The Cabal", ftp://ftp.cwi.nl/pub/herman/SNFSrecords/SNFS-233

[4] G. Cooperman, http://www.ccs.neu.edu/home/gene/pargap.html

[5] The GAP Group, *GAP – Groups, Algorithms, and Programming, Version4*, Lehrstuhl D f Mathematik, Rheinisch Westfälische Technische Hochschule, Aachen, Germany and School of Mathematical and Computational Sciences, Univ. St. Andrews, Scotland, 1997.

[6] A. Hanaki. Data of association schemes, published at WWW (1999〜): http://kissme.shinshu-u.ac.jp/as/.

[7] A. Hanaki and I. Miyamoto, *Classification of association schemes with* 16 *and* 17 *vertices*, Kyushu J. Math. **52** (1998) 383-395. *Classification of association schemes with* 18 *and* 19 *vertices*, Korean J. Comp. App. Math. **5** (1998) 543-551. *Classification of association schemes of order up to* 22, Kyushu J. Math. **54** (2000) 81-86.

[8] H. Mishima, http://www.asahi-net.or.jp/ KC2H-MSM/cn/index.htm

[9] P. Zimmermann, http://www.loria.fr/ zimmerma/records/ecmnet.html