Parallelization of Low-Communication Processes

Jörg Waldvogel and Peter Leikauf
Seminar for Applied Mathematics SAM
Swiss Federal Institute of Technology ETH, CH-8092 Zürich

November 2001

1 Project Description

Tapping the large amounts of idle time of numerous workstations and parallelizing large, but simply structured computational efforts is a cheap method of advancing to the limits of computationally feasible tasks. Known systems like PVM are able to function under quite general conditions, however, they achieve considerable complexity. The goal of this project at the Seminar for Applied Mathematics (SAM) of ETH is to parallelize an arbitrary number of workstations for distributed computations with low communication. In particular, we want to exploit the simple situation of a large computational effort that naturally splits into highly independent subtasks producing low data traffic. It was possible to come up with a small but robust system taking advantage of this simple, yet important situation. A typical class of problems of this type are exhaustive-search problems.

1.1 Robustness

The following flexible and fail-safe concept has been implemented on a network of Unix-based workstations, including clusters of workstations like the Beowulf cluster of ETH. There is a varying set of independent processes running on (not necessarily) different machines: One master process and several client processes. The master assigns tasks to any connecting client, keeps track of the assigned tasks and processes the reported results. Each client tries to connect to the master, receives a task, disconnects and begins its computation. After completion of a task the result is reported back to the master, and a new task is immediately assigned to the client.

The only fixed parameter of this system is the master's internet address; the number of client processes and machines taking part can increase or decrease at any time without the need of notifying a central instance. The system can survive almost any failure it may encounter: breakdown of a client process (or machine), breakdown of the entire network, even the breakdown of the master process itself. If a critical breakdown (e.g. the master's machine) lasts less than, say, 2 hours the system will recover without human interaction. In any case, the central mechanism for managing the results is immune against all typical hardware or software breakdown situations and would even survive a (local) media failure on the master's file system.

In addition, much care has been taken of using as little resources as possible. The clients are designed to use no file system at all, so even if a client fails or if its machine is shut down, there will be no trace left. The parallelization software exclusively uses the standard C++ libraries contained in every Unix installation. Network load is kept small since there is no permanent connection between the processes; documentation see [10], `pmp.pdf`.

1.2 Current Application: Clusters of Primes

Currently, we are applying our parallelization concept to an algorithm involving sieving techniques for locating and counting clusters of prime numbers. Whereas the distribution of primes seems to be fairly regular (if the Riemann hypothesis is true), the distribution of twin primes and longer clusters is largely unknown and is characterized by large-scale anomalies. Collecting experimental data on these anomalies is one of the reasons for the interest in clusters of primes.

Another challenge of finding clusters of prime numbers is the unproven prime k-tuple hypothesis, which is concerned with patterns of natural numbers that occur repeatedly with all elements being prime. The hypothesis states that any pattern that is not forbidden by simple divisibility considerations occurs infinitely often in the sequence of primes. An example of a pattern that certainly cannot occur infinitely often in the sequence of primes is the pattern $c = [x, x+2, x+4]$: at least one of its elements is divisible by 3. The only prime instance of c occurs with $x = 3$, when one of its elements is the prime number 3 itself. A proof of the prime k-tuple hypothesis is currently out of reach; not even for the simplest case, the twin prime hypothesis, a proof is in sight, [7]. In contrast, the observed average densities of prime k-tuples in the accessible range are in perfect agreement with the densities $\rho_c(x)$ conjectured by Hardy and Littlewood [5] in 1922:

$$\rho_c(x) = \frac{h_c}{(\log x)^{|c|}}, \tag{1}$$

where h_c is the Hardy-Littlewood constant associated with the pattern c, and $|c|$ is the number of elements in the pattern c.

The basic ingredients in our search algorithm are the Chinese remainder theorem to exclude divisibility by small primes (e.g. $p \leq 53$), sieving techniques to exclude divisibility by intermediate primes ($53 < p \leq 641$), and a probabilistic primality test (Miller-Rabin) for the remaining large primes $p > 641$. Finally, candidates for new clusters of large primes are "hardened“ by rigorous primality proofs. An account of a similar algorithm was given by Tony Forbes in [3].

1.3 Software

The current implementation takes advantage of the mathematical software package PARI, written in C++ by the Bordeaux group C. Batut, K. Belabas, D. Bernardi, H. Cohen and M. Olivier, e-mail pari@math.u-bordeaux.fr. Features of the package are arbitrary-precision arithmetics with integer, real and complex numbers, many special functions, general symbolic computations, computational number theory and computational algebra; it may well be useful for other projects and for teaching. It is available as freeware from [1]. Due to the advanced features of PARI the entire algorithm (without the parallelization) can be formulated with less than 30 lines of code ([10], `paricode.gp`). Near-optimum speed is achieved by implementing the innermost loop (the sieve involving only single-precision integers) in C++.

As a prerequisite for our implementation, PARI must be installed on each client machine or cluster (requiring disk space of some 10 MBytes). The processes running on the clients use little memory (< 1 MByte) and are given the lowest possible priority. Tests on 20 workstations of SAM have confirmed that the project will not noticeably disturb other users.

1.4 Implementation

Software development and preliminary experiments were done with the idle time of 20 workstations of the Seminar for Applied Mathematics SAM. On Asgard the project was given the idle time of 432 processors, i.e. the idle time of 90% of Asgard's processors. The authors express their gratitude to the steering committee and to the Asgard operator team for this generous allotment and for the continuous support. Since our project uses little memory and is always running on lowest priority (nice +19), it does not noticeably channel off resources from other users: from a busy processor our project is getting at most 6% CPU time.

During the year 2002 the project was using an estimated portion of 6% to 30% of Asgard's capacity. The highest turn-out was mainly achieved shortly before shut-downs of Asgard, when most users had withdrawn their processes. Due to the robustness of our algorithms we were able to have our processes stopped by the operator – without any loss of data. The restart after the recovery of the system is a matter of a few minutes.

During the upgrading (including renaming of the nodes) of September 2000 and during several periods of hardware problems in 2001 (February, May/June) the robustness was also essential: there were no losses except for the time lost when the machine was down.

2 Results

2.1 Finding an 18-Tuplet of Maximum Density

Our first major goal was to find an 18-tuplet of maximum density in the range of 10^{24}. Consider the pattern

$$c = [13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83] \qquad (2)$$

consisting of the 18 consecutive primes beginning at 13 and ending at 83. For patterns that can be represented by sequences of small primes we will use the shorthand notation involving the initial and final primes, separated by two periods. Accordingly, $c = [13..83]$; $c' = [-83.. - 13]$ represents the mirror image of c, thus introducing the obvious concept of negative primes.

The Hardy-Littlewood constant of the pattern c or c' is found to be $h_c = 6723654.312$ (see [8] for the definition and computation of h_c). Equ. (1)

implies that the expected number of occurrences of the pattern c (as well as of its mirror image) in a large interval of length Δ near a much larger x is given by $h_c \, \Delta \, (\log x)^{-18}$. Integration of this average density yields the following expected frequencies HL(x) (referred to as the Hardy-Littlewood count [5], HL count for short):

x	$1_{10}24$	$2_{10}24$	$3_{10}24$	$4_{10}24$	$5_{10}24$	$1_{10}25$	$1_{10}26$
HL(x)	0.438	0.695	0.912	1.107	1.286	2.056	9.962

Therefore, we could ”hope“ that the above prime pattern c would repeat for some x in the range of $3 \cdot 10^{24}$, and also that its mirror image occurs in the same range. One could even be lucky to have occurrences for smaller values of x, but also, with bad luck, the search might have to be pushed much higher. Estimates based on the idle time of 20 workstations of SAM (80 % idle time) were in the range of 2 to 5 years. Including 6 % idle time of Asgard may result in a speedup of a factor 4. If the search is successful within a reasonable timespan, we could claim a world record that wouldn't be easy to break. Dense clusters of 19 primes in the patterns of [13..89], [37..113] or their mirror images are expected to repeat/occur only in the range of 10^{26}.

Searching the natural numbers up to $3 \cdot 10^{24}$ for clusters of 18 primes among 71 consecutive integers turned out to be at the upper end of currently feasible computations. Locating the two expected occurrences truly amounts to finding the proverbial needle in the haystack. To get an idea of the sheer size of the number $3 \cdot 10^{24}$: It is in the order of magnitude of the total number of cells in all living humans on Earth. Immagine to look for a particular subset of 100 cells within this set!

We began the search for the 18-tuplet c' in August 2000 by searching blocks of size 10^{24}. 432 processors of the Beowulf Cluster and 20 workstations of SAM were involved. We were lucky: the first hit happened in November 2000, the second one (pattern c) at the end of January 2001. The search is now complete up to $2.9999949836 \cdot 10^{24}$. Some details are given in the table below; for more details see [10], `cl18.pdf` as well as [2], [4], [6].

Pattern	Block	Begin/end of search	Date	Initial element	HLcount
[-83..-13]	1	8/03 - 9/19/00			0.438
[-83..-13]	2	9/19 -10/26/00			0.695
[-83..-13]	3	10/29 -11/20/00	11/13/00	2845372542509911868266807	0.880
[13.. 83]	1	12/19 - 1/23/01			0.438
[13.. 83]	2	1/23 - 2/27/01	1/31/01	1906230835046648293290043	0.673
[13.. 83]	3	2/27 - 3/26/01			0.912

2.2 Four Kinds of Dense Patterns of Seventeen Primes

Minimal clusters of 17 primes among 67 consecutive integers can exist in the patterns [13..79], [17..83] or in the mirror images of these patterns. We searched the range up to $3.333545 \cdot 10^{23}$ for all four patterns. As the table below shows, the distribution is very uneven, e.g., the pattern [17..83] has 8 instances, whereas [-83..-17] does not occur at all in this interval. For this pattern the search was extended to 10^{24}, and a pair of rather close instances, near $7.3 \cdot 10^{23}$ and near $7.5 \cdot 10^{23}$, turned up; the Hardy-Littlewood count for this interval is 11.2. In contrast to this discrepancy, the total number, 18, of tuplets present up to $\frac{1}{3} \cdot 10^{23}$ agrees very well with the expected number (Hardy-Littlewood) of 19.256. Among these clusters (listed below by their initial elements), 14 are new, 4 have been discovered earlier (see the comments in the table), and 2 are in the range of small primes.

```
Pattern [ 13.. 79],   3/26/01 - 4/30/01,  6 tuplets,  HLcount = 4.316
                       13
  47624415490498763963983
  78314167738064529047713
  83405687980406998933663
 110885131130067570042703
 163027495131423420474913

Pattern [-79..-13],   5/02/01 - 6/29/01,  4 tuplets,  HLcount = 4.316
   1620784518619319025971   J. Waldvogel 1997, Tony Forbes  1998
   2639154464612254121531   Tony Forbes  1998, J. Waldvogel 1998
   3259125690557440336631   Tony Forbes  1998, J. Waldvogel 1998
 124211857692162527019731

Pattern [ 17.. 83],   8/17/01 -10/25/01,  8 tuplets,  HLcount = 5.312
                       17
  37630850994954402655487
  53947453971035573715707   Tony Forbes 1998
 174856263959258260646207
 176964638100452596444067
 207068890313310815346497
 247620555224812786876877
 322237784423505559739147

Pattern [-83..-17],   7/02/01 - 8/17/01,  0 tuplets,  HLcount = 5.312
Pattern [-83..-17],   Interval (0,1e24),  2 tuplets,  HLcount = 11.2
 7349755347933245127179 47
```

2.3 The Largest Known 15-Tuplet

In order to demonstrate the capability of our algorithm to handle even larger numbers, we screened the interval

$$[10^{30} - 10^{20},\ 10^{30} + 99 \cdot 10^{20}]$$

for 14-tuplets in the pattern of $c = [11..61]$. With the corresponding HL constant $h_c = 50975.35252$ we expect about

$$50975 \cdot 10^{22} \cdot \log(10^{30})^{-14} = 9.05$$

such 14-tuplets in the above interval; actually 12 are present, where the first one happens to be the largest known 15-tuplet of Pattern [11..67] (no larger one was found up to 2006). In the table below we also indicate the sequences of differences of consecutive primes.

```
      999999999990000000000000000000 = 1e30 - 1e20, lower search limit
 1    999999999962618227626700812281   114 2 4 2 4 6 2 6 4 2 4 6 6 2 6 30
 2   1000000001044178961179268851051    22 2 4 2 4 6 2 6 4 2 4 6 6 2 18
 3   1000000001544051614464292419601   162 2 4 2 4 6 2 6 4 2 4 6 6 2 126
 4   1000000001553601074663653211311    52 2 4 2 4 6 2 6 4 2 4 6 6 2 18
 5   1000000001772437688818681781011    48 2 4 2 4 6 2 6 4 2 4 6 6 2 82
 6   1000000003068759599025980926181    54 2 4 2 4 6 2 6 4 2 4 6 6 2 112
 7   1000000004930964950164522054901   112 2 4 2 4 6 2 6 4 2 4 6 6 2 60
 8   1000000005644941246959007679801    22 2 4 2 4 6 2 6 4 2 4 6 6 2 106
 9   1000000005832631360266813468481    52 2 4 2 4 6 2 6 4 2 4 6 6 2 78
10   1000000006672161724368529625351    82 2 4 2 4 6 2 6 4 2 4 6 6 2 112
11   1000000007541367760266886291861   150 2 4 2 4 6 2 6 4 2 4 6 6 2 286
12   1000000008282508019026959814211   240 2 4 2 4 6 2 6 4 2 4 6 6 2 148
     1000000009900000000000000000000 = 1e30 + 99e20, upper search limit
```

2.4 Comments

Prime numbers have been one of the favorite objects of research of classical mathematics, and various computations involving primes have been done already by Euclid. With the advent of modern computing machines the size of the accessible numbers has dramatically increased, and ever more impressive – though mostly futile – results have been obtained. Only with the rise of modern cryptography [9] primes had grown up from objects mainly existing in the brains of mathematicians to real-world objects with eminently important applications.

This is only a partial explanation for the ongoing quest for all kinds of prime number records. *The New Book of Prime Number Records* by Paulo Ribenboim [7] has 541 (the 100th prime!) pages and is in its third edition. Among hundreds of records there are the largest known prime, the largest known value of $\pi(x)$, the largest known maximal gap, etc., and last not least, the longest known dense cluster of large primes.

The discovery of a dense cluster of 18 primes in the range of $3 \cdot 10^{24}$ on November 13, 2000 received immediate coverage in the web journal of ETH (ETH Life, December 6, 2000, [2]). The news about this computation, being about 50 times harder than previous computations in the same field [3], [10], were also announced in the number theory press, e.g. [6]. Dense clusters of primes receive particular attention on the website [4], continuously actualized by Tony Forbes. The successes of our implementation bear the danger of monopolizing this site and taking away all the fun!

3 Future Projects

A typical property of computations involving prime numbers is that they can go on forever. This is not our goal, however. It was mentioned that finding a large cluster of 19 primes (among 77 consecutive integers) is by far harder than the 18er, perhaps 50 times the effort. Therefore, this is out of reach for the current implementation.

If the idle time of Asgard is still available to our project in the current rate, we plan to further exploit the excellent performance of our system in the directions mentioned below. The computation time for each item is expected to be in the order of months.

1. Generate a fair number (30...100) of densest 16ers, Patterns [13..73] and [-73..-13], in order to investigate the regularity/irregularities of the distribution of this instance of a long dense cluster.

2. Search for exotic long patterns, such as 9 twin primes within 105 consecutive integers, in order to support the prime k-tuple hypothesis [5].

3. Search for exotic shorter patterns, e.g. 10 consecutive primes with mutual difference 210 to support the hypothesis that in some range of the natural numbers 210 is the most abundant difference of consecutive primes.

References

[1] C. Batut, K. Belabas, D. Bernardi, H. Cohen, M. Olivier: *The software package PARI* (freeware). http://pari.math.u-bordeaux.fr/

[2] ETH Life, Die tägliche Webzeitung der ETH, 6. Dezember 2000, Archiv. *Prozessoren malochten 100 Tage.* http://www.ethlife.ethz.ch/articles/tages/Waldvogel.html

[3] Tony Forbes: *Prime clusters and Cunningham chains.* Math. of Comp. **68** (1999) 1739-1747.

[4] Tony Forbes: *Prime* k*-tuplets.* http://www.ltkz.demon.co.uk/ktuplets.htm

[5] G.H. Hardy and J.E. Littlewood: *Some problems of Partitio Numerorum III.* Acta Math. **44**, 1922, 1-70.

[6] *Number theory news.* http://www.utm.edu/research/primes/

[7] Paulo Ribenboim: *The New Book of Prime Number Records*, 3rd ed. Springer 1996, 541 pp.

[8] Hans Riesel: *Prime Numbers and Computer Methods for Factorization*, 2nd ed. Birkhäuser 1994, 464 pp.

[9] R.L. Rivest, A. Shamir, L. Adleman: *A method for obtaining digital signatures and public-key cryptosystems.* Communications of the ACM **21** (1978) 121-126.

[10] Jörg Waldvogel: *Homepage.* http://www.math.ethz.ch/˜waldvoge/Projects/