Integer Factoring Utilizing PC Cluster

Kazumaro Aoki

`maro at isl·ntt·co·jp`

NTT

Contents

- Background
- Integer Factoring Algorithms
- World Records
- On 1024-bit GNFS
- My Experiences

Integer Factoring and Cryptology

until 1977: mostly for recreational purposes

since then, a somewhat better excuse:

to figure out secure RSA key sizes

$\cdots$ **A. Lenstra@SHARCS05**

`http://www.hyperelliptic.org/tanja/SHARCS/talks/ArjenLenstra.ppt`

Integer Factoring Problem (IFP)

Input: composite N

Output: non-trivial factor p **(**$1 < p < N$**,** $p \mid N$**)**

No known algorithm can efficiently find p**.**

Complexity of IF

method	complexity	effective range
TD	$L_p[1, 1]$	$p \leq 2^{28}$
ECM	$L_p[1/2, 1.414]$	$p \leq 2^{130}$
MPQS	$L_N[1/2, 1.020]$	$N \leq 2^{320}$
SNFS	$L_N[1/3, 1.526]$	$N > 2^{320}$
GNFS	$L_N[1/3, 1.923]$	$N > 2^{320}$
MPGNFS	$L_N[1/3, 1.902]$	$N > 2^{2000}$**(?)**

$$L_x[s, c] = \exp((c + o(1))(\log x)^s(\log \log x)^{1-s})$$

Trial Division (TD)

- **Simply divide by 2, 3, 5, . . .**
- **Small divisors can be found by** `factor(N,2^31-1)` **in PARI/GP.** `http://pari.math.u-bordeaux.fr/`

Elliptic Curve Method (ECM)

- **expect $\#E(\mathrm{GF}(p))$ is smooth by changing curves**
- **Excellent implementation in public: GMP-ECM**
  `http://gforge.inria.fr/projects/ecm/`

x **is** y**-smooth** $\Leftrightarrow \forall p \mid x$, p: **prime** $\Rightarrow p \leq y$

Quadratic Sieve (QS)

- **Construct** $x^2 \equiv y^2 \pmod{N}$ **efficiently using index calculus method** **(**$\gcd(x \pm y, N) \mid N$**)**
- **fastest if** N **is less than 100 digits**
- **Good implementation in public: msieve** `http://www.boo.net/~jasonp/qs.html`

Number Field Sieve (NFS)

- **Developed at early 1990s**
- **Similar to MPQS, construct** $x^2 \equiv y^2 \pmod{N}$ **using index calculus method**
- **The asymptotically fastest algorithm known for general-type integer factoring**
- **recent factoring records are done by (G)NFS**
- **an implementation in public: GGNFS** `http://www.math.ttu.edu/~cmonico/software/ggnfs/`

Outline of NFS

find many relations, $(a, b) \in \mathbf{Z}^2$ s.t.

$$\begin{cases} \left|(-b)^{\deg f_1} f_1(-\frac{a}{b})\right| = \prod_{p<B_1} p^{e_p^{(a,b)}} \\ \left|(-b)^{\deg f_2} f_2(-\frac{a}{b})\right| = \prod_{q<B_2} q^{e_q^{(a,b)}} \end{cases}$$

find dependency in $\mathrm{GF}(2)$

$$\{[e_p^{(a,b)}, \ldots, e_q^{(a,b)}, \ldots]\}_{(a,b)}$$

$$\Rightarrow x^2 \equiv y^2 \pmod N$$

Steps of NFS

find $x, y \in \mathbf{Z}$ **s.t.** $x^2 \equiv y^2 \pmod{N}$

1. **polynomial selection**
2. **sieving**
3. **filtering**
4. **linear algebra**
5. **square root**

Polynomial Selection

for given N**,** $d = \deg f$
find $f(X) \in \mathbf{Z}[X]$**,** $M \in \mathbf{Z}$
s.t. $f(M) \equiv 0 \pmod{N}$

GNFS: choose $M \approx N^{1/(d+1)}$**, determine the coefficients of** $f(X)$ **by** $N = \sum_{i=0}^{d} c_i M^i$

SNFS: determined automatically $|c_i| \approx 1$ **from** N

Sieving

find many $(a, b) \in \mathbf{Z}^2$ **(**$\gcd(a, b) = 1$**) s.t.**

$$\begin{aligned} F(a,b) &= \left|(-b)^d f(-a/b)\right| = \prod_{p<B_1} p^{e_p} \\ G(a,b) &= |a + bM| = \prod_{p<B_2} p^{e_p} \end{aligned}$$

choose (a, b) **nearby origin point, because** $[a, b \to \infty] \Rightarrow [F, G \to \infty]$

- **heaviest step in theory and experiments.**
- **sparsely connected distributed computing is possible, but considerably large memory is required.**

Filtering

part of linear algebra step in theory

- removing duplicate relations
- find relation-sets that have non-trivial dependencies
- based on Gaussian elimination keeping sparse

The matrix size is reduced one over tens.

Example (GNFS176):

456M$\times$329M (w: 9G?) $\longrightarrow$ 8.5M$\times$8.5M (w: 1.7G)

Linear Algebra

- **Find linear dependency in sparse and huge $\mathrm{GF}(2)$-matrix ($\approx$ tens of million for WR)**
- **block Lanczos or block Wiedemann algorithm are frequently used.**
- **dominate NFS in theory**

It is not trivial to confirm the intermediate computation as correct.

Square Root

- **Number theoretic knowledges are required only for this step.**
- **Negligible complexity, but long program code.**

Records of GNFS

composite	# of bits	YY/MM	who
RSA-200	663	05/05	Bonn et al.
RSA-640	640	05/11	Bonn et al.
c176 in $11^{281}+1$	582	05/05	NTT et al.
RSA-576	576	03/12	Bonn et al.
c164 in $2^{1826}+1$	545	03/12	NTT et al.
RSA-160	530	03/04	Bonn

From `http://www.crypto-world.com/FactorAnnouncements.html` and others

Records of SNFS

composite	# of bits	YY/MM	who
c274 in $6^{353}-1$	911(913)	06/01	NTT et al.
c248 in $2^{1642}+1$	822	04/03	NTT et al.
$2^{809}-1$	809	03/01	Bonn
c244 in $5^{349}-1$	809(811)	06/04	Kruppa+Bonn
c239 in $2^{811}-1$	793(811)	04/06	NFSNET
c234 in $3^{491}+1$	777(779)	04/09	NFSNET+CWI
c227 in $2^{773}+1$	774(753)	00/11	CWI et al.

From `http://www.crypto-world.com/FactorAnnouncements.html` and others

Records of ECM

composite	$\log_2 p$	YY/MM	who
c214 in $10^{381}+1$	222	06/08	Dodson
c180 in $3^{466}+1$	219	05/04	Dodson
c311 in $10^{311}-1$	212	05/09	Aoki et al.
c175 in $3^{533}+1$	209	05/11	Kruppa
c187 in $2^{2034}+1$	205	05/04	Dodson
c162 in $2^{905}+1$	201	06/09	Dodson
c242 in $2^{1099}+1$	197	05/10	Dodson
c162 in $10^{233}-1$	194	05/02	Dodson

From `http://www.loria.fr/~zimmerma/records/top100.html`

On 1024-bit GNFS

- After proposing the special hardware device, for example, TWINKLE, many estimations were made.
- $o(1) = 0$ approximation in $L_N[1/3, 1.923]$ is very dangerous. We know the complexity increase about 3 times every 10 digits for $N \approx 2^{512}$. It means $o(1) \approx -0.279$.
- People want to know the complexity to factor 1024-bit RSA modulus using simple scale: "X-bit security"

On Pentium 4 [2.53GHz] Platform

RC5-72: 3,549,150 keys/sec (v2.9001-478)

RSA-150(496-bit) sieve: 20,597,260 seconds

$\longrightarrow$ "46-bit security"

- **3 times every 10 digits**

  $\cdots$ 72-bit security $\approx$ 1024-bit IF

"at least a factor 200 gap between 1024-bit RSA and 80-bit security"

$\cdots$ A. Lenstra@SHARCS05

My Experiences

- **Big factorings: GNFS164, SNFS248, GNFS176, ECM311, SNFS274**
- **Joint work with Kida, Shimoyama, Sonoda, and Ueda**
- **Partly supported by CRYPTREC project.**

How to choose candidate composites?

- **RSA challenge: 576, 640, 702, . . . bits**
- **old RSA challenge: every 10 digits**
- **Cunningham project: $b^e \pm 1$ ($2 \leq b \leq 12$) (described as b, $e\pm$)**
- **partition number, near repunit, . . .**

- **ECM (removing small factor)**
- **GNFS vs SNFS (special type composite)**

GNFS164 (1) — c164 in 2,1826L

- **Our first attempt to make a world record. At that time, the world record is 160 digits.**
- **The polynomial selection step was started mid-Oct 2003, in parallel with GMP-ECM with B1=43M. A candidate, c165 in 2,2030L, was factored by ECM (44 digits factor).**
- **Franke team already finished sieving for RSA-576 at Sep 2003.**

GNFS164 (2)

- **Sieving: late Oct to early Dec**
- **Filtering: late Nov to early Dec**
- **Lenstra announced at Asiacrypt (Nov 30 - Dec 4): a workshop for IF will be held Dec 12**
- **Linear algebra: Dec 3 to Dec 15**

GNFS164 (3)

- **RSA-576 factoring announcement was posted on `sci.crypt` Dec 4.**
- **Our factoring was completed Dec 18.**

SNFS248: c248 in 2,1642M (1)

- **We change the target from GNFS to SNFS. At that time, the world record is 244 digits.**
- **We found 56 digits factor by ECM (3rd largest at that time) Dec 17 in the first candidate (ECM started Dec 5, 2003).**
- **Sieving: mid-Dec 2003 to early Feb 2004 in parallel with GMP-ECM with B1=43M (finished Jan 10).**
- **NFSNET was already started sieving for c239 in 2,811-.**

SNFS248 (2)

- **Filtering: early Feb 2004**
- **Linear algebra (CRYPTREC cluster): Feb 11 to Feb 24**

SNFS248 (3)

- **Square root: Feb 25, but failed**
- **Failure reason: 324 relations with $\gcd(a, b) \neq 1$ are included**
- **Go back to filtering step**

Feb 28: CRYPTREC cluster deadline

- **2nd Filtering: late Feb 2004**
- **2nd Linear algebra (Rikkyo Univ): Mar 1 to Mar 20 (including HW trouble, and manual operation mistakes)**

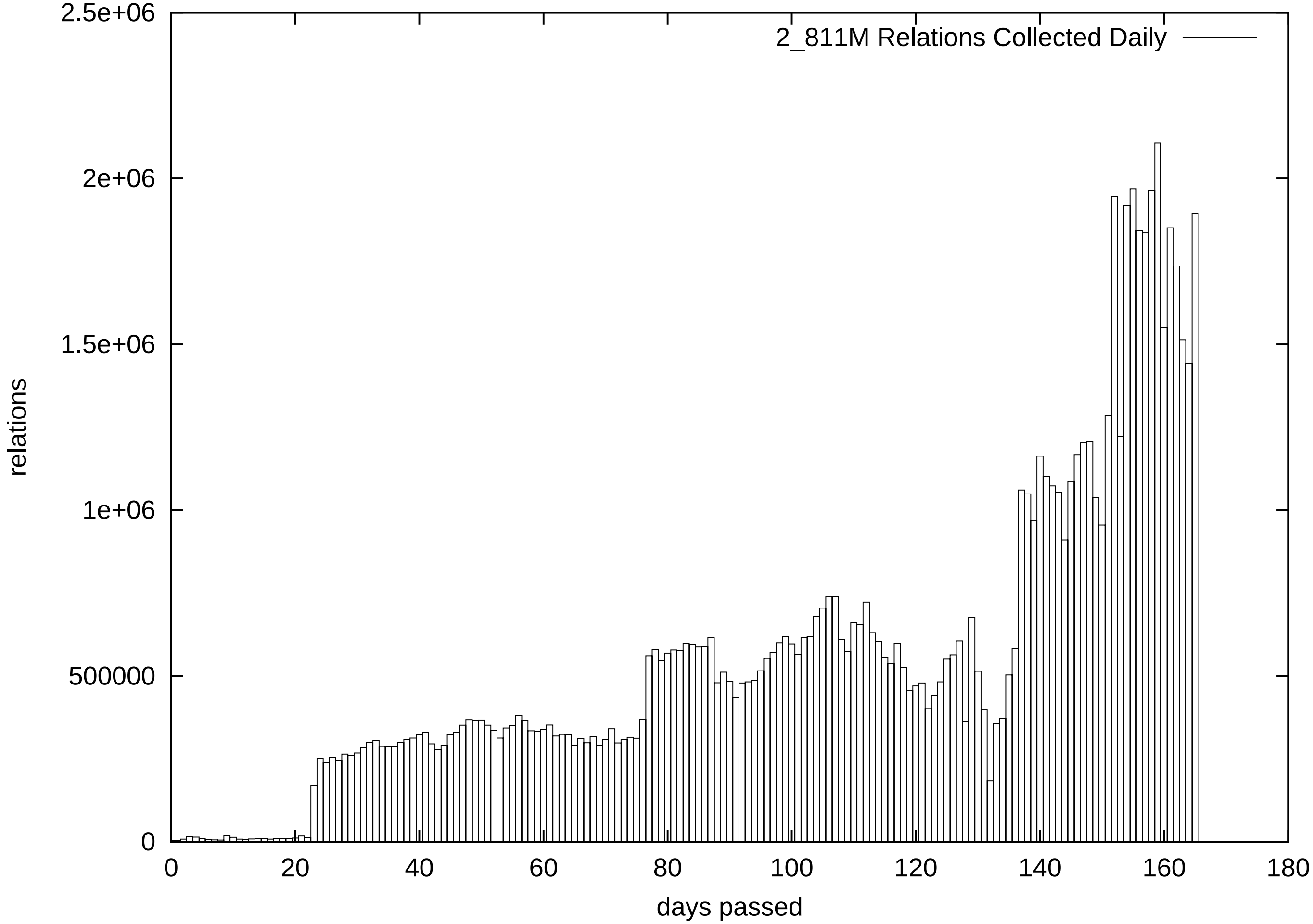
SNFS248 (4)

- **Our linear algebra code said:** $\mathrm{rank} >$ **# of rows**
- **half day examination RAM using** `memtest86`
- **3rd Linear algebra (Rikkyo Univ): Mar 16 to Mar 25**

SNFS248 (5)

- **Our linear algebra code said:**
  $\mathrm{rank} >$ **# of rows**
- **4th Linear algebra (NTT):**
  Mar 19 to Mar 29 (estimation)

NFSNET 2_811M Daily Reports



From `http://www.nfsnet.org/stats2/statsreporter.cgi?template=relations.html&project=2_811M`

SNFS248 (6)

- **Mar 27 (Sat): one of PC crashes (disk trouble)**
- **4th Linear algebra (NTT):**
  Mar 29 (restart) to Apr 2 (estimation)

SNFS248 (7)

- **Apr 2 (Fri) 1:20am: power stop by lightening strike**
- **4th Linear algebra (NTT):**
  Apr 3 (restart) to Apr 3 midnight
- **33 dependencies are found**
- **Square root: Apr 3 to Apr 4 (midnight)**
- **1st solution:**
  $\gcd(N, x+y) = \gcd(N, x-y) = 1$

SNFS248 (8)

- **When computing square root using 2nd dependencies, we found a factor by** $\gcd(N, x - y/2)$
- **after factoring we found the reason (a parameter is doubled)**

Hardware failures in 3 years

40 servers including 32 2U P4[2.53GHz] servers.

- **15% HD were broken, but 90% were repaired by automatic reallocation of bad sectors.**
- **2 power units were broken.**
- **4 memory modules were broken.**
- **8 CPUs sometimes produced incorrect result.**
- **2 CPU fans were stopped.**
- **1 motherboard was broken.**
- **1 of 4 HUBs was broken.**

GNFS176: c176 in 11,281+

- **Our first world record of GNFS**
- **Feb 2, 2005 to Apr 22, 2005**

poly sel	**3.5 year @ P4[3.2GHz]**
sieving	**9.7 year @ P4[3.2GHz]**
linear alg	**5 day @ 36 P4[2.8GHz-3.2GHz] w/ GbE**

- **The record was only kept in a week.**
- **RSA-200 factoring was announced May 2005.**

of PCs Used in Each Step

	Step	distributed computing	# of PCs for GNFS176
1	poly. sel.	easy	52
2	sieve	easy	400
3	filtering	rel. easy	2
4	linear alg.	tight conn.	36
5	square root	rel. easy	36

Details of Our Program Running

	time spent GNFS176	
poly. sel.	20d	pol51m0b $\longrightarrow$ pol51opt
	2h	mkprime
sieve	27d	ltsieve
filtering	4h	classifyRel $\longrightarrow$ uniqRel, 32to64
	3h	getLP $\longrightarrow$ countLP $\longrightarrow$ lptxt2bin
	2h	sfctr
	8h	scmpi
	1h	compff $\longrightarrow$ mkprematrixbin
	2d	splitpm + smerge
lin. alg.	1h	shufflematrix $\longrightarrow$ mkmatrixbin
	1h	cut224mat $\longrightarrow$ splitmatrix
	5d	planczos256
	1h	solve224mat $\longrightarrow$ rff $\longrightarrow$ gaussext
$\sqrt{}$	1h	anneal
	1h	papprox
	1h	pcouveignes, rsqrt

Program Lines

Step	# of lines	ratio
polynomial selection	5626	10%
sieve	16943	30%
filtering	17607	32%
linear algebra	7352	13%
square root	8150	15%
total	55678	100%

as of October 2005

ECM311: 10,311-

- **kilo-bit SNFS candidate**
- **2nd largest factor found by ECM at that time: R311 = p64 $\times$ p247**
- **We call the idle CPU time in NTT for Step 1, and Step 2 was done by our occupied PCs.**
- **7.91 year @ Opteron[2.0GHz] w/ 4GB RAM (89 calendar days)**

SNFS274: c274 in 6,353-

- SNFS record
- 911 bits number
- sieving tried to start Sep 11, 2005 (actually started Sep 10)
- factoring expected to complete Jan 19, 2006 (actually Jan 23)

sieving	16.6 year @ P4[3.2GHz] (=17.3 year @ A64[2.0GHz])
linear alg	34.64 day @ 25 P4[3.2GHz] w/ GbE

Our contributed optimization

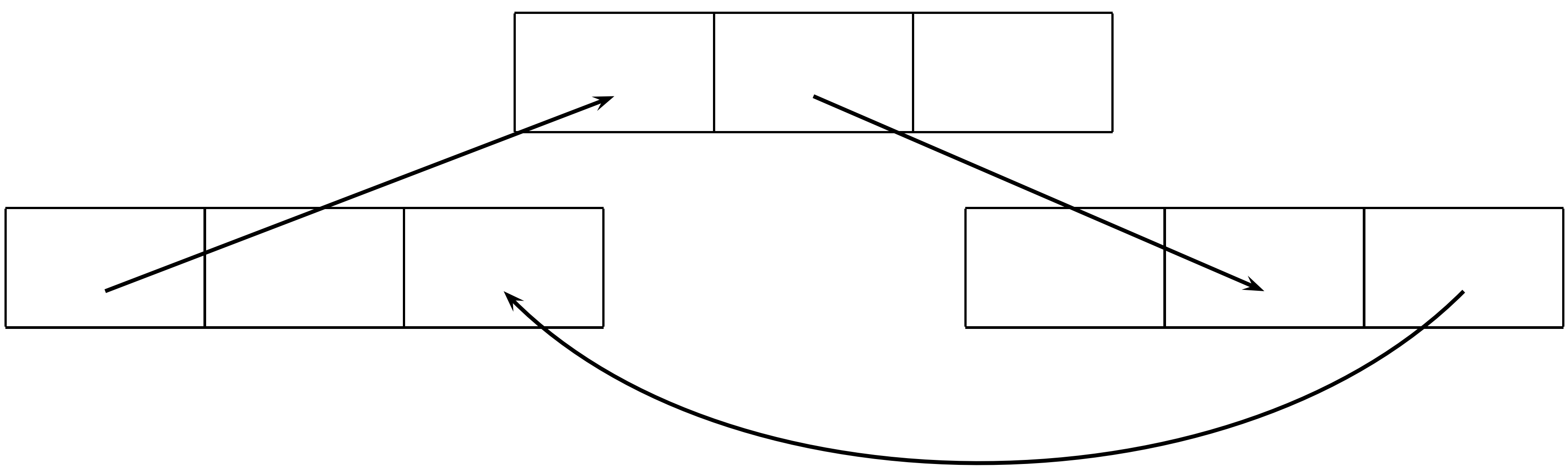
- **Use of bucket sort for sieving step (Asiacrypt 2004)**
- **Variable sieving range for lattice sieve**
- **Sum share algorithm for linear algebra step (reinvention of wheel?)**
- **Network construction for PC cluster (reinvention of wheel?)**

Sum Sharing

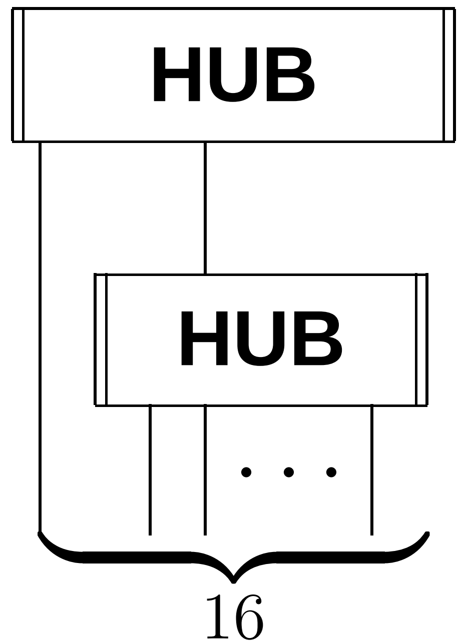
before: length l vector in n nodes

after: sum of all vectors shared in all nodes

A full-duplex ring network can realize in $2(n-1)\left\lceil \frac{l}{n} \right\rceil$, where length 1 vector can transfer in time 1.



Network Construction: 16 nodes



with 16-port HUB. each node has 1 NIC.

Network Construction: 36 nodes

1-	1-	12	12	13	13
1-	1-	12	12	13	13
12	12	2-	2-	23	23
12	12	2-	2-	23	23
13	13	23	23	3-	3-
13	13	23	23	3-	3-

using 3 20-port HUBs. each node has 2 NICs.

Final Remarks

- **I feel that PC cluster is the best solution to factor big integer for $<\approx$500,000 USD budget (not including human resources).**
- **It is very difficult to keep all nodes available.**

Keep the factors coming!

$\cdots$Sam Wagstaff (Cunningham table maintainer)