1001 ways to fail record computations

F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé, P. Zimmermann

Dec. 2nd, 2019
Polynomial Selection

\[ p = \text{RSA-240} + 49204 \]

\[ f = 39x^4 + 126x^3 + x^2 + 62x + 120 \]

\[ g = 286512172700675411986966846394359924874576536408786368056x^3 \]
\[ + 24908820300715766136475115982439735516581888603817255539890x^2 \]
\[ - 18763697560013016564403953928327121035580409459944854652737x \]
\[ - 236610408827000256250190838220824122997878994595785432202599 \]

\[ \text{Res}(f, g) = -540p \]
Polynomial Selection

\[ p = \text{RSA-240} + 49204 \]

\[ f = 39x^4 + 126x^3 + x^2 + 62x + 120 \]

\[ g = 286512172700675411986966846394359924874576536408786368056x^3 \]
\[ + 24908820300715766136475115982439735516581888603817255539890x^2 \]
\[ - 18763697560013016564403953928327121035580409459944854652737x \]
\[ - 236610408827000256250190838220824122997878994595785432202599 \]

\[ \text{Res}(f, g) = -540p \]

Use the wrong polynomials
Polynomial Selection

\[ p = \text{RSA-240} + 49204 \]

\[ f = 39x^4 + 126x^3 + x^2 + 62x + 120 \]

\[ g = 286512172700675411986966846394359924874576536408786368056x^3 + 24908820300715766136475115982439735516581888603817255539890x^2 - 18763697560013016564403953928327121035580409459944854652737x - 236610408827000256250190838220824122997878994595785432202599 \]

\[ \text{Res}(f, g) = -540p \]

Use the wrong polynomials

dlp240test.sh:

```bash
if [ "\$sha1sum $poly" != 0423a08c3b518fb5788300caf2cd01c3c4fbda03 ]
then
    echo "You are using the wrong polynomial !!!" >&2
    exit 1
fi
```

Collect millions of relations between $f$-side and $g$-side with cado-nfs' binary las.

Use wrong las options and parameters about 70 parameters or options -lambda0, -lambda1, -bkthresh, -bkthresh1, -bkmult

Too many files ls impossible server crashed re-organize files, move .tgz relation files, etc...

RSA-240 and DLP-240 3/7
Collect millions of relations between $f$-side and $g$-side with cado-nfs’ binary las.

**Use wrong las options and parameters**

- About 70 parameters or options
- `-lambda0`, `-lambda1`, `-bkthresh`, `-bkthresh1`, `-bkmult`
Collect millions of relations between $f$-side and $g$-side with cado-nfs’ binary las.

**Use wrong las options and parameters**

- about 70 parameters or options
- `-lambda0`, `-lambda1`, `-bkthresh`, `-bkthresh1`, `-bkmult`

**Too many files**

- `ls` impossible
- server crashed
- re-organize files, move `.tgz` relation files, etc...
Core binding

Bind each mono-thread job to a virtual core
Bind each double-thread job \(-t 2\) to a physical core

```
p=`grep "processor" /proc/cpuinfo | tail -1 | cut -d " " -f 2`
let N=$(((p-1)/2))
for i in `seq 0 $N`; do
   b="core:${i}"
   hwloc-bind --membind $b --cpubind $b $cmd &
   sleep 0.2
done
```
Core binding

Bind each mono-thread job to a virtual core
Bind each double-thread job \texttt{-t 2} to a physical core

\begin{verbatim}
p=`grep "processor" /proc/cpuinfo | tail -1 | cut -d " " -f 2`
let N=$(((p-1)/2))
for i in `seq 0 $N`; do
  b="core:${i}"
  hwloc-bind --membind $b --cpubind $b $cmd &
sleep 0.2
done
\end{verbatim}

Use a physicists’ cluster with heterogeneous CPU

binding error, server crashed

Blindly trust auto CPU binding

can be 30\% slower in some cases
Linear algebra

Block Wiedemann stores intermediate computation as big vectors. Gigabytes each.
Block Wiedemann stores intermediate computation as big vectors. Gigabytes each.

**Transient storage fault for checkpoints**

- Counter-measure: verify the checkpoints.
- Counter-failure: verifier was incomplete!
- Iterate.
Block Wiedemann stores intermediate computation as big vectors. Gigabytes each.

**Transient storage fault for checkpoints**

- Counter-measure: verify the checkpoints.
- Counter-failure: verifier was incomplete!
- Iterate.

**Battle silent failures**

- MPI-send data.
- Hit a silent 4G limitation in message size in some cases.
Finally, everything went fine!

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>17</td>
<td>33</td>
<td>49</td>
<td>2</td>
<td>18</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>19</td>
<td>35</td>
<td>50</td>
<td>3</td>
<td>20</td>
<td>36</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>37</td>
<td>51</td>
<td>4</td>
<td>22</td>
<td>38</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>39</td>
<td>52</td>
<td>5</td>
<td>24</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>40</td>
<td>53</td>
<td>6</td>
<td>26</td>
<td>41</td>
</tr>
<tr>
<td>6</td>
<td>27</td>
<td>42</td>
<td>54</td>
<td>7</td>
<td>28</td>
<td>43</td>
</tr>
<tr>
<td>7</td>
<td>29</td>
<td>43</td>
<td>55</td>
<td>8</td>
<td>30</td>
<td>44</td>
</tr>
<tr>
<td>8</td>
<td>31</td>
<td>45</td>
<td>56</td>
<td>9</td>
<td>32</td>
<td>46</td>
</tr>
<tr>
<td>9</td>
<td>32</td>
<td>46</td>
<td>57</td>
<td>10</td>
<td>Mem[</td>
<td>170G/188G</td>
</tr>
<tr>
<td>10</td>
<td>33</td>
<td>47</td>
<td>58</td>
<td>11</td>
<td>Swp[</td>
<td>0K/3.72G</td>
</tr>
<tr>
<td>11</td>
<td>34</td>
<td>48</td>
<td>59</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>35</td>
<td></td>
<td>60</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>36</td>
<td></td>
<td>61</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>37</td>
<td></td>
<td>62</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>38</td>
<td></td>
<td>63</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>39</td>
<td></td>
<td>64</td>
<td>Mem</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Swp</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tasks: 365, 119 thr; 65 running
Load average: 65.01 64.26 52.02
Uptime: 00:42:24

RSA-240 and DLP-240
Finally, everything went fine!

RSA-240 =
509435952285839914555051023580843714132648382024111473186660
296521821206469746700620316443478873837606252372049619334517
* 
244624208838318150567813139024002896653802092578931401452041
221336558477095178155258218897735030590669041302045908071447

$p = RSA-240 + 49204$

target = hex("The magic words are still Squeamish Ossifrage")

log_5(target) =
926031359281441953630949553317328555029610991914376116167294
204758987445623653667881005480990720934875482587528029233264
473672441500961216292648092075981950622133668898591866811269
28982506005127728321426751244111412371767375547225045851716

https://caramba.loria.fr/dlp240-rsa240.txt