Distributed GPU password cracking

Alexander Kasabov
&
Jochem van Kerkwijk

University van Amsterdam
System and Network Engineering

{akasabov | jkerkwijk}@os3.nl

February 2, 2011
• Introduction
• Password cracking
• Graphics processing unit
• Distributed architectures
• Evaluation
• Conclusions
Research Question

What is the best possible way to do password cracking with GPU processing power in a distributed environment?
KPMG

- MPI Cluster
  - Patched version of John the Ripper

- Super GPU machine
  - 2x NVIDIA
  - 1x ATI

- A lot of unused GPUs
Passwords

- Symbol sequence
- Used for authentication
- Hashing
  - One way
  - Avoids plain text
  - Prone to interception and replay
Password Strength

![Diagram showing the relationship between password length and entropy for different types of passwords.](image-url)
Attack Methods

• Brute force
  • Computational intensive
  • Simple to implement

• Dictionary attack
  • Smart dictionaries
  • I/O intensive

• Pre computation
  • Rainbow tables
  • I/O intensive
Graphics Processing Unit

- GPU vs CPU

  - Suitable for “embarrassingly parallel” tasks
  - Bottleneck is bandwidth

\[^1\] NVIDIA CUDA Programming guide v2.0 - 2008
Speed Comparison

- GT200 = GeForce GTX 280
- G71 = GeForce 7900 GTX
- NV35 = GeForce FX 5950 Ultra
- G92 = GeForce 9800 GTX
- G70 = GeForce 7800 GTX
- NV30 = GeForce FX 5800
- G80 = GeForce 8800 GTX
- NV40 = GeForce 6800 Ultra
- G80 Ultra
- G92
- GT200

\[^2\]NVIDIA CUDA Programming guide v2.0 - 2008
GPGPU

- General processing for GPU (GPGPU)
  - Starts in 2003 with NVIDIA and ATI
  - Support for integer function
- GPGPU APIs
  - Suitable for
    - linear algebra, scientific simulations, pattern recognition, video encoding, image scaling and . . .
    - password cracking
  - CUDA, Stream SDK, OpenCL
  - Support for multiple GPUs on one host machine
Distributed GPU architectures

- Approaches for distributed GPU password cracking
  - Process distribution for CPU and GPU by software
  - Combination with GPGPU API
  - Existing software for password cracking on GPU
Architecture overview

- CLara
- BOINC
- MPI
- oclHashCat
- IGHASHGPU
- OpenCL
- Stream
- CUDA

Distributed architectures

Zonenberg DHC
Elcomsoft

GPU
Criteria

- Distributing the key space
- GPU support
- Recovery and error handling
- Different hash types (extensible)
- Current KPMG cluster
## Evaluation

<table>
<thead>
<tr>
<th></th>
<th>BOINC</th>
<th>MPI</th>
<th>CLara</th>
<th>IGHASH GPU</th>
<th>oclHashCat</th>
<th>DHC</th>
<th>Elcomsoft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributing key space</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>GPU support</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Recovery &amp; error handling</td>
<td>+</td>
<td>+/-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Different hash types (extensible)</td>
<td>c</td>
<td>c</td>
<td>c</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+/-</td>
</tr>
<tr>
<td>API, Documentation &amp; support</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+/-</td>
</tr>
<tr>
<td>KPMG cluster</td>
<td>+</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>

C = custom application development required; ? = unknown
Conclusions

• Practical solutions
  • An open-source password cracking tool which supports distributed GPUs
  • MPI + OpenCL

• For the long term - CLara
  • Custom application development allows for tweaks
  • OpenCL is open source implemented by NVIDIA & ATI cards
  • Support for heterogeneous systems including Cell, FPGA, Playstation...
A & Q

• Acknowledgements
  • Michiel van Veen & Marc Smeets
  • Marcus Bakker & Martijn Sprengers

• Questions?