Detecting very large sets of referenced files at 40/100 GbE, especially MP4 files

A. Larbanet, J. Lerebours, J.P. David

DFRWS 2015, Philadelphia PA, 2015 august 11th
Internet growth

- 2.5 G users and 51.2 EB/mo in 2013.
- Estimated 3.9 G users and 131.6 EB/mo in 2018 (Cisco).
- +50% growth/yr for high-end home connection (Nielsen's law).
- ISP and Internet backbones w/ 10, 40 & 100 Gbps technologies.
Illegal behaviors on the Internet

- Downloading copyright infringing materials.
  - Movies, music, books, video games, etc.
- Illegal intrusions and stealings of confidential documents.
- Sharing child pornography related materials.
What Google, Facebook and Twitter can do

Google, Facebook and Twitter to block 'hash lists' of child abuse

© 10 August 2015 | UK

Web giants Google, Facebook and Twitter have joined forces with a British
Detection difficulties on the network

- Link speed: up to 10, 40 and 100 Gbps.
- With Ethernet, IP & TCP/UDP protocols:
  - Fragmentation in packets.
  - Packet order is not known.
  - Route is not predetermined.
- Millions of clean packets/s (noise).
  → Reconstruction is difficult (~impossible).
  → Fast detection of known files based on a small subset...

Adrien Larbanet
adrien.larbanet@polymtl.ca
A solution: the max-hashing algorithm

- Presented by David in 2013 [1].
- Focuses on small windows (16B) w/ a local property:
  - Local maximum hash.

\[
\text{Final fingerprint: } F = \max_{0 \leq n \leq N} (H(W_n)) \quad \text{with} \quad N = S_B - S_W
\]
# Max-hashing – Principle

## Basic referencing
- Fragment the file in blocks.
- Compute a hash for each 16B window on the blocks.
- Store every local maximum into database.

## Detection
- Capture network packets
- Compute a hash for each 16B window on the payloads.
- Compare the local maxima with the referenced fingerprints.

→ Packets are independent: task is parallelizable.
Basic referencing with Max-Hashing

Once upon a midnight dreary, while I pondered weak and weary, Over many a quaint and curious volume of forgotten lore, While I nodded, nearly napping, suddenly there came a tapping, As of some one gently rapping, rapping at my chamber door. "Tis some visitor,' I muttered, `tapping at my chamber door - Only this, and nothing more.' Ah, distinctly I remember it was in the bleak December, And each separate dying ember wrought its ghost upon the floor. Eagerly I wished the morrow; - vainly I had sought to borrow From my books surcease of sorrow - sorrow for the lost Lenore - For the rare and radiant maiden whom the angels name Lenore - Nameless here for evermore. And the silken sad uncertain rustling of each purple curtain Thrilled me - filled me with fantastic terrors never felt before; So that now, to still the beating of my heart, I stood repeating "Tis some visitor entreating entrance at my chamber door - Some late visitor entreating entrance at my chamber door; - This it is, and nothing more,'

Edgar A. Poe, The Raven
Basic referencing with Max-Hashing

- Fragment document into blocks.

- Once upon a midnight dreary, while I pondered weak and weary, Over many a quaint and curious volume of forgotten lore, While I nodded, nearly napping, suddenly there came a tapping, As of some one gently rapping, rapping at my chamber door. `Tis some visitor,' I muttered, `tapping at my chamber door -Only this, and nothing more.'

- Ah, distinctly I remember it was in the bleak December, And each separate dying ember wrought its ghost upon the floor. Eagerly I wished the morrow; - vainly I had sought to borrow From my books surcease of sorrow - sorrow for the lost Lenore - For the rare and radiant maiden whom the angels name Lenore - Nameless here for evermore.

- And the silken sad uncertain rustling of each purple curtain Thrilled me - filled me with fantastic terrors never felt before; So that now, to still the beating of my heart, I stood repeating "Tis some visitor entreatling entrance at my chamber door - Some late visitor entreatling entrance at my chamber door; - This it is, and nothing more,'

Edgar A. Poe, The Raven
Basic referencing with Max-Hashing

- Fragment document into blocks.
- Hash every 16-character wide windows.
- Keep local maximum in every block.
  → Are all 16-character strings equi-probable in language?
- Some are common.
- Some are unique.

- Once upon a midnight dreary, while I pondered weak and weary, 
  Over many a quaint and curious volume of forgotten lore, 
  While I nodded, nearly napping, suddenly there came a tapping, 
  As of some one gently rapping, rapping at my chamber door. 
  'Tis some visitor,' I muttered, 'tapping at my chamber door - Only this, and nothing more.'

- Ah, distinctly I remember it was in the bleak December, 
  And each separate dying ember wrought its ghost upon the floor. 
  Eagerly I wished the morrow; - vainly I had sought to borrow 
  From my books surcease of sorrow - sorrow for the lost Lenore - 
  For the rare and radiant maiden whom the angels name Lenore - 
  Nameless here for evermore.

- And the silken sad uncertain rustling of each purple curtain 
  Thrilled me - filled me with fantastic terrors never felt before; 
  So that now, to still the beating of my heart, I stood repeating 
  'Tis some visitor entreatizing entrance at my chamber door - 
  Some late visitor entreatizing entrance at my chamber door; - This it is, and nothing more,'

Edgar A. Poe, The Raven
Referencing files – Problems with MP4+H.264

• Using CCV video DB [2]: 8073 MP4 files.

1. Keep 512 local maxima per file:
   - Over 4 Millions fingerprints.
   - 371 redundancies.
   - 243 files implied.

2. With detection setup:
   - Over 500 Millions comparisons with fingerprints in DB.
   - 18000 false-positives.
Referencing files – MP4/H.264 file structure

MP4

H.264

Adrien Larbanet
adrien.larbanet@polymtl.ca
Advanced referencing for MP4/H.264 files

- Modified free and open-source software Ffmpeg:
  - Get the position of the “high-entropy” data field for each Macro-Blocks.
  - When referencing, only keep local maximum computed on these segments.

1. Referencing 8073 files:
   - No redundancies.

2. Detection of half the DB:
   - No false-positive.

Adrien Larbanet
adrien.larbanet@polymtl.ca
Let's detect!
Detection – GPU architecture

- **SIMD** based.
  - Parallelizable task on independent data.

- Greater computation power (than CPU):
  > 4Tflops (Single-precision) for high-end GPU.

- Cheaper and easier to program than FPGA.

NVIDIA GK110 Multiprocessor
Max-Hashing with GPU, system overview

- **Goals:**
  - Detect Ethernet packets carrying known content.
  - Flag all the packets belonging to a suspicious stream.
Transfer Ethernet packets over PCIe 2.0

- From DDR3 main memory.
- To onboard GDDR5.
- Through PCIe 2.0 (16x).
  - 64 Gbps in theory.
- Using CUDA DMA.

→ Up to 50 Gbps.
Parsing headers on GPU

- **Kernel** tasks for each packet:
  - Getting payload offset and length (store these in memory).
  - Extracting TCP/UDP-IP information for latter analysis.

400 Millions packets per second
Max-Hashing on GPU

• *Kernel* tasks:
  - Get payload position and length.
  - Hash payload and extract 4 local maximum hashes.

• Benchmark setup:
  - 768 MB of random data hashed by 524,288 threads.

→ 119.9 Gbps, producing 40 millions fingerprints per second.
Fingerprints and TCP/UDP flows databases

- Hashtable:
  - $2^p$ rows and $c$ columns.
  - $p$ LSB of fingerprint as index in the hashtable.
  - For each fingerprint to compare, $c$ threads are launched.

- Example with 8-bit fingerprints ($p=4$):
  - Is 0x22 in the DB?

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>...</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0xA0</td>
<td>0x30</td>
<td>0</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0x21</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0xE2</td>
<td>0x22</td>
<td>0xF2</td>
<td>...</td>
<td>0x32</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0x54</td>
<td>0x24</td>
<td>0xB4</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0xC5</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>15</td>
<td>0xEF</td>
<td>0x0F</td>
<td>0</td>
<td>...</td>
<td>0</td>
</tr>
</tbody>
</table>
Fingerprint Database embedded in GPU mem.

• Hashtable configuration:
  – Contains referenced fingerprints (64 bits each).
  – $2^{21}$ rows, $c_1$ columns.

• Benchmark setup
  – Over 2 millions random fingerprints to search in the DB.
  – $c_1$ is variable.
TCP/UDP-IP flows database in GPU mem.

• Hashtable configuration:
  - Contains IP@ and TCP/UDP port boundaries.
  - $2^{16}$ rows, $c_2$ columns.
  - 16-bit index computed from IP@.

• Benchmark setup
  - More than half a million random TCP/UDP-IP flow informations.
  - $c_2$ is variable.
Chained *kernels*

Transfers through PCIe 2.0: *50 Gbps.*

Chained *kernels*: *63 Gbps.*

Sequentially: ~25 Gbps...

Pipelining computation and transfer ?
Pipelining – Implementation
Conclusion

Referencing MP4 files

• Basic approach not suitable for heavily formatted files.
• Quick format study reveals position of high-entropy segments.
  - Good candidates for hashing!

Detection on GPU

• Up to 50 Gbps.
• Suitable for 40 GbE.
• Limited by PCIe 2.0...
Discussion - Future work

• Switch to a PCI-e 3.0 x16 compatible GPU:
  – 128 Gbps in theory.

• Multiple GPU setups.
  – Mirror setup.
  – Database dedicated GPU.
  – ...

• Extend dedicated referencing to other video and audio formats:
  – H.265, VP9, etc.
References and acknowledgments


The authors are grateful to NetClean and the Government of Quebec.
Thank you!
Any question?