“Finding a kernel in a haystack”

Analyzing video cards

Candice Quates

BSidesNola 2016

The GPU, our friend.

Video hardware

- Puts stuff on your screen
- GPU = Graphics Processing Unit
- Many have their own dedicated memory
- Manufacturers:
  - NVIDIA, AMD, Intel

Why GPU memory analysis?

- It’s interesting and new
- Possible use in digital forensics
- May be able to combat future malware

Big fancy video cards

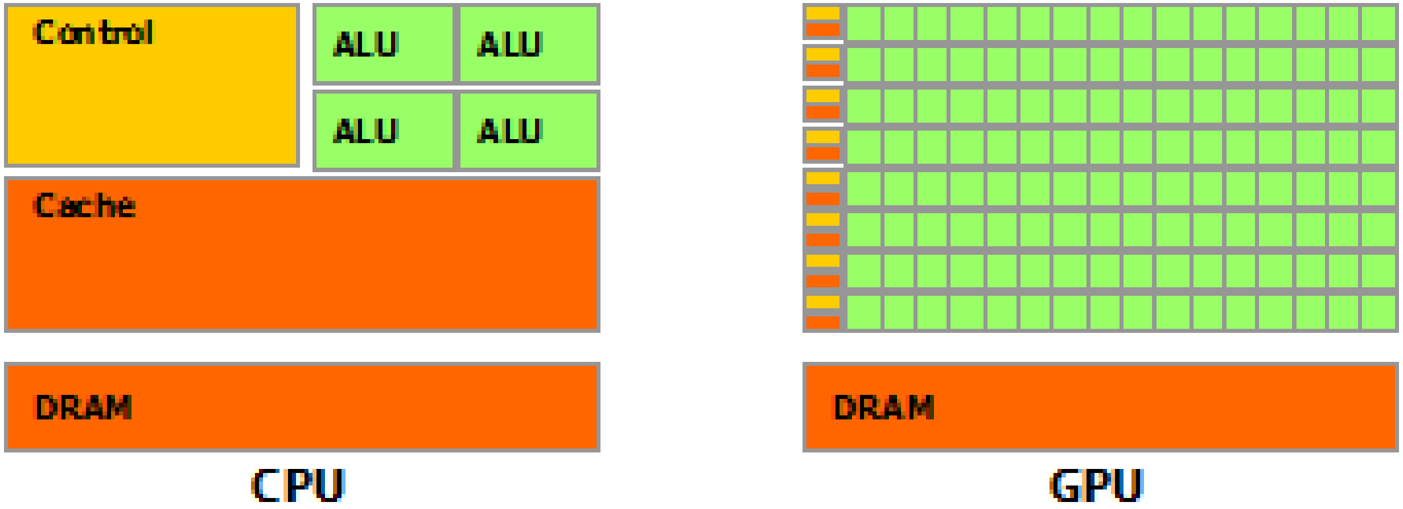
- Ever spent stupid money on a video card?
  - (and not used it?)
- What’s really happening in that 4gb of ram?
- Gaming got us here, but math-based applications push us forward

Video games to Linear Algebra?

- Yes!
- 3D graphics requires a ton of math
- Rendering pretty worlds is faster in parallel

- Result: thousands of cores on a video card, and dedicated memory

Why you should use a GPU for math:



Green = Math

But, math is boring, right?

- Hash cracking
- Bitcoin mining
- Chrome acceleration
- Scientific applications
- Forensics software

GPGPU – How to do Math

- General Purpose GPU computing
- API access to GPU processing and memory
  - (“Ordinary” programming capabilities)

- APIs:
  - CUDA (NVIDIA only)
  - OpenCL (NVIDIA+AMD)

CUDA 10k feet

- API and Programming model for massively multithreaded programming

- Abstracts away thread management in favor of code blocks called kernels

- Everything easy is hard; everything hard is easy.

Basic CUDA application

- Allocate memory on device
- Copy data to device
- Process data on device with kernels
- Copy results from device to host
- Access results

GPU ram is now accessible

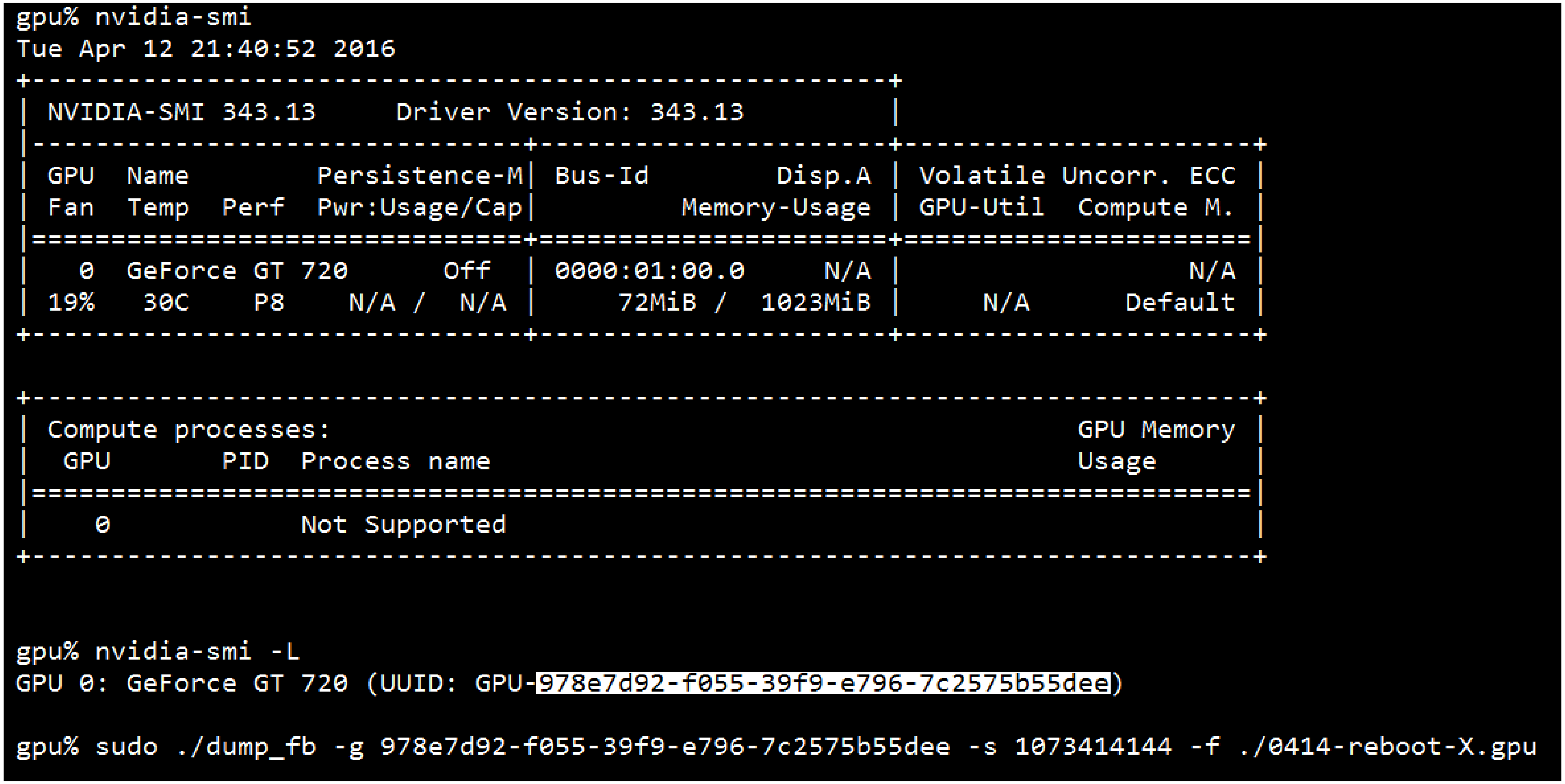
- Dr. Golden Richard’s work with NVIDIA
  - (handling all the hard stuff and the people stuff)
- Driver patch for Linux which allows memory dumps of entire card address space.

- http://www.cs.uno.edu/~golden/gpu-malware-research.html

Dumping GPU ram

- Result: Giant single file
- Most GPU ram not used for display persists across reboots
- We can analyze programs running under any OS by rebooting and capture

Dump commands



```
gpu% nvidia-smi
Tue Apr 12 21:40:52 2016
+------------------------------------------------------+
| NVIDIA-SMI 343.13     Driver Version: 343.13         |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GeForce GT 720      Off  | 0000:01:00.0     N/A |                  N/A |
| 19%   30C    P8    N/A /  N/A |     72MiB /  1023MiB |     N/A      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Compute processes:                                               GPU Memory |
|  GPU       PID  Process name                                     Usage      |
|=============================================================================|
|    0            Not Supported                                               |
+-----------------------------------------------------------------------------+


gpu% nvidia-smi -L
GPU 0: GeForce GT 720 (UUID: GPU-978e7d92-f055-39f9-e796-7c2575b55dee)

gpu% sudo ./dump_fb -g 978e7d92-f055-39f9-e796-7c2575b55dee -s 1073414144 -f ./0414-reboot-X.gpu
```

How to find a needle in a haystack?

- Excluding as much as possible
- Magnets
- Burn it down
  - (not advised for video cards, or hay)

Investigative tools

- Large file of /dev/zero
- Program to find zeros in sparse files*
- Large files of text (Hamlet, Macbeth…)
- CUDA API sample programs**
- Hex editor

* https://github.com/candicenonsense/nullfinder
** http://docs.nvidia.com/cuda/cuda-samples/

GPU address space layout

Data	**Size**
Reserved?	tiny
Display + X11	middling – 8-40mb
General Purpose	huge
Kernels?	middling – ~100mb
Reserved	tiny

Useful addresses vary

- Card size – 1gb/2gb/4gb…
- Console type
  - X-windows
  - Frame buffer console
  - Headless
- Screen resolution

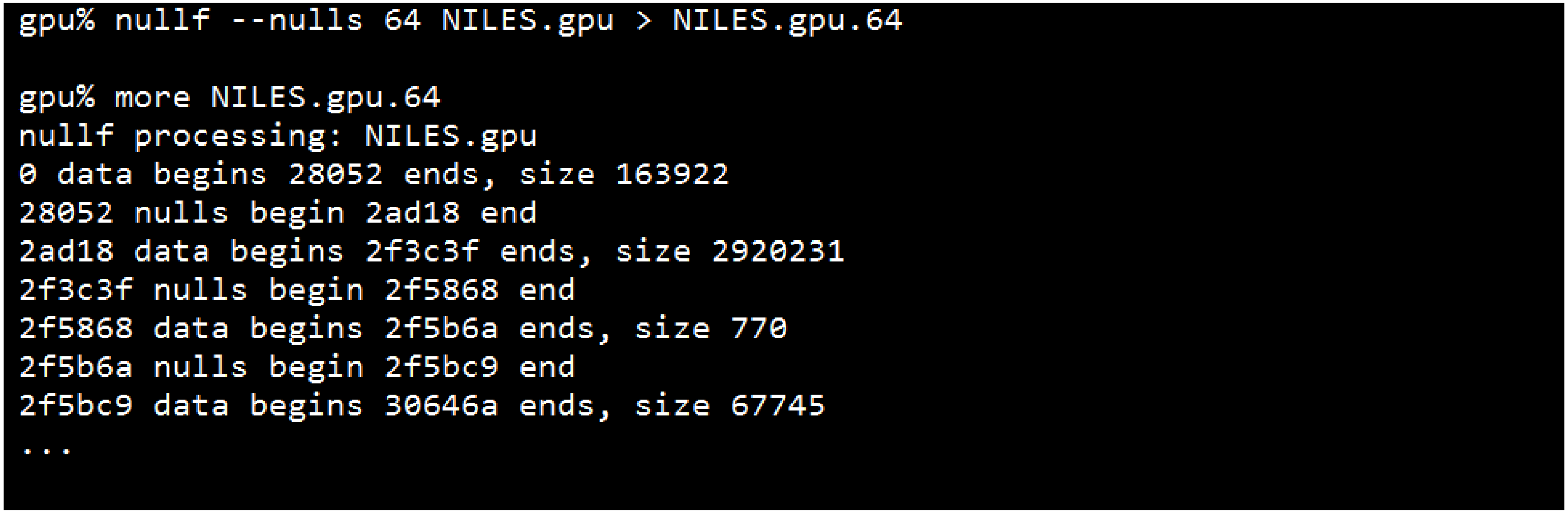
Mapping with nulls

- Baseline: freshly powered on system
  - Image
  - Nulls written to as much space as possible
  - Image
  - Experiments!
  - Image

Mapping, continued

- Search for large blocks of nulls (1k-1mb) first to find general purpose memory
- Drill downwards with smaller blocks (512 byte, 64-byte) for program data and kernels
- Use results to guide comparisons

Map commands



```
gpu% nullf --nulls 64 NILES.gpu > NILES.gpu.64

gpu% more NILES.gpu.64
nullf processing: NILES.gpu
0 data begins 28052 ends, size 163922
28052 nulls begin 2ad18 end
2ad18 data begins 2f3c3f ends, size 2920231
2f3c3f nulls begin 2f5868 end
2f5868 data begins 2f5b6a ends, size 770
2f5b6a nulls begin 2f5bc9 end
2f5bc9 data begins 30646a ends, size 67745
...
```

After this, convert it into something spreadsheet-like and organize further

1gb card map, blanked, no X

type	**content**	**start**	**end**	**size**
data	Reserved	00000000	000b15fe	726526
data	Reserved	000b1800	000b35fe	7678
data	Reserved	000b3800	000b55fe	7678
data	Reserved	000b5800	000bf1fe	39422
data	Video memory	000bf400	008ca1fe	8433150
data	Video memory	008d0000	01020208	7668232
nulls	**User memory**	**01020208**	**3fa20000**	1050672632
data	Kernels?	3fa20000	3fa361ff	90623
data	Kernels?	3fac8000	3facc1ff	16895
data/nulls	Reserved	…	…	…

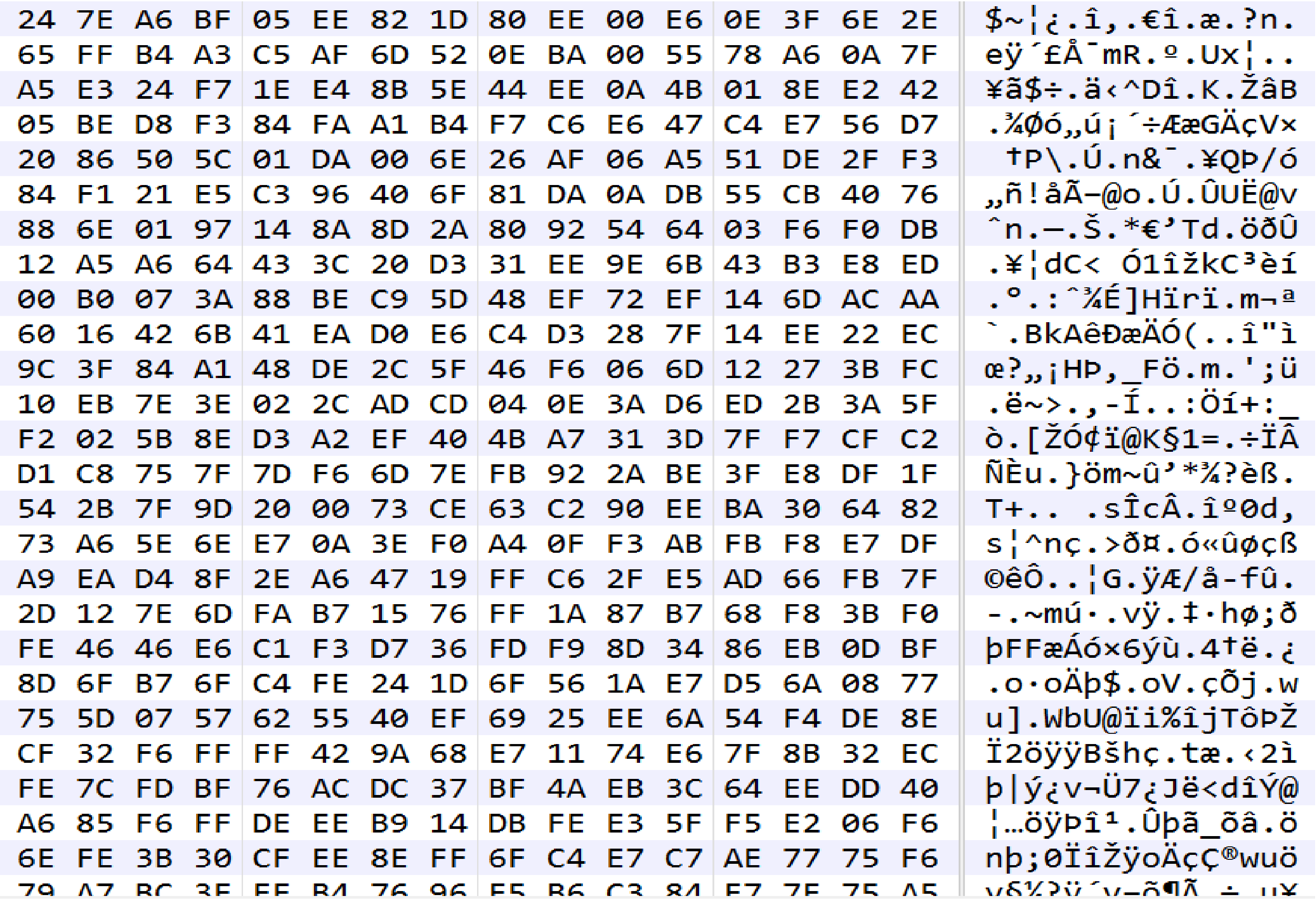
1gb card map, blanked, X

type	likely content	start	end	size
data	Reserved	00000000	008cb0fe	9220350
data	Reserved	008cc000	008cd0fe	4350
data	Reserved	008ce000	008cf0fe	4350
data	Video memory	008d0000	009601ff	590335
data		00960427	0096063f	536
data		00961423	00961643	544
repeat 4-5x				
data	Video memory	009a0004	00a705ff	853499
data	X11?	00a70800	00a715ff	3583
repeat 4-5x				
data	Video memory	00a75800	00a7f1ff	39423
data	Video or X11	00a7f400	010401ff	6032895
data		0105fffc	010601fc	512
data	X11?	01060400	010611ff	3583
data	X11?	01061400	010621ff	3583
data	X11?	01062400	010631ff	3583
data	X11	01063400	019e01ff	9948671
nulls	**User memory**	**019e01ff**	**3b5c000b**	**968752652**
data		3b5c000b	3b5c61fb	25072
data		3b654005	3b68cb3c	232247

Unallocated space

- Is not blank at power on
  - (unless you're in the cloud)
- Looks like nonsense
- Varying entropy levels
- Doesn't contain a lot of nulls
  - (less than 8 in a row, over gigabytes of space)

If you get here you’ve gone too far



```
24 7E A6 BF  05 EE 82 1D  80 EE 00 E6  0E 3F 6E 2E   $~¦¿.î‚.€î.æ.?n.
65 FF B4 A3  C5 AF 6D 52  0E BA 00 55  78 A6 0A 7F   eÿ´£Å¯mR.º.Ux¦..
A5 E3 24 F7  1E E4 8B 5E  44 EE 0A 4B  01 8E E2 42   ¥ã$÷.ä‹^Dî.K.ŽâB
05 BE D8 F3  84 FA A1 B4  F7 C6 E6 47  C4 E7 56 D7   .¾Øó„ú¡´÷ÆæGÄçV×
20 86 50 5C  01 DA 00 6E  26 AF 06 A5  51 DE 2F F3    †P\.Ú.n&¯.¥QÞ/ó
84 F1 21 E5  C3 96 40 6F  81 DA 0A DB  55 CB 40 76   „ñ!åÃ–@o.Ú.ÛUË@v
88 6E 01 97  14 8A 8D 2A  80 92 54 64  03 F6 F0 DB   ˆn.—.Š.*€’Td.öðÛ
12 A5 A6 64  43 3C 20 D3  31 EE 9E 6B  43 B3 E8 ED   .¥¦dC< Ó1îžkC³èí
00 B0 07 3A  88 BE C9 5D  48 EF 72 EF  14 6D AC AA   .°.:ˆ¾É]Hïrï.m¬ª
60 16 42 6B  41 EA D0 E6  C4 D3 28 7F  14 EE 22 EC   `.BkAêÐæÄÓ(..î"ì
9C 3F 84 A1  48 DE 2C 5F  46 F6 06 6D  12 27 3B FC   œ?„¡HÞ,_Fö.m.';ü
10 EB 7E 3E  02 2C AD CD  04 0E 3A D6  ED 2B 3A 5F   .ë~>.,-Í..:Öí+:_
F2 02 5B 8E  D3 A2 EF 40  4B A7 31 3D  7F F7 CF C2   ò.[ŽÓ¢ï@K§1=.÷ÏÂ
D1 C8 75 7F  7D F6 6D 7E  FB 92 2A BE  3F E8 DF 1F   ÑÈu.}öm~û’*¾?èß.
54 2B 7F 9D  20 00 73 CE  63 C2 90 EE  BA 30 64 82   T+.. .sÎcÂ.îº0d‚
73 A6 5E 6E  E7 0A 3E F0  A4 0F F3 AB  FB F8 E7 DF   s¦^nç.>ð¤.ó«ûøçß
A9 EA D4 8F  2E A6 47 19  FF C6 2F E5  AD 66 FB 7F   ©êÔ..¦G.ÿÆ/å-fû.
2D 12 7E 6D  FA B7 15 76  FF 1A 87 B7  68 F8 3B F0   -.~mú·.vÿ.‡·hø;ð
FE 46 46 E6  C1 F3 D7 36  FD F9 8D 34  86 EB 0D BF   þFFæÁó×6ýù.4†ë.¿
8D 6F B7 6F  C4 FE 24 1D  6F 56 1A E7  D5 6A 08 77   .o·oÄþ$.oV.çÕj.w
75 5D 07 57  62 55 40 EF  69 25 EE 6A  54 F4 DE 8E   u].WbU@ïi%îjTôÞŽ
CF 32 F6 FF  FF 42 9A 68  E7 11 74 E6  7F 8B 32 EC   Ï2öÿÿBšhç.tæ.‹2ì
FE 7C FD BF  76 AC DC 37  BF 4A EB 3C  64 EE DD 40   þ|ý¿v¬Ü7¿Jë<dîÝ@
A6 85 F6 FF  DE EE B9 14  DB FE E3 5F  F5 E2 06 F6   ¦…öÿÞî¹.Ûþã_õâ.ö
6E FE 3B 30  CF EE 8E FF  6F C4 E7 C7  AE 77 75 F6   nþ;0ÏîŽÿoÄçÇ®wuö
79 A7 BC 3F  FE B4 76 96  F5 B6 C3 84  F7 7E 75 A5   y§¼?þ´v–õ¶Ã„÷~u¥
```

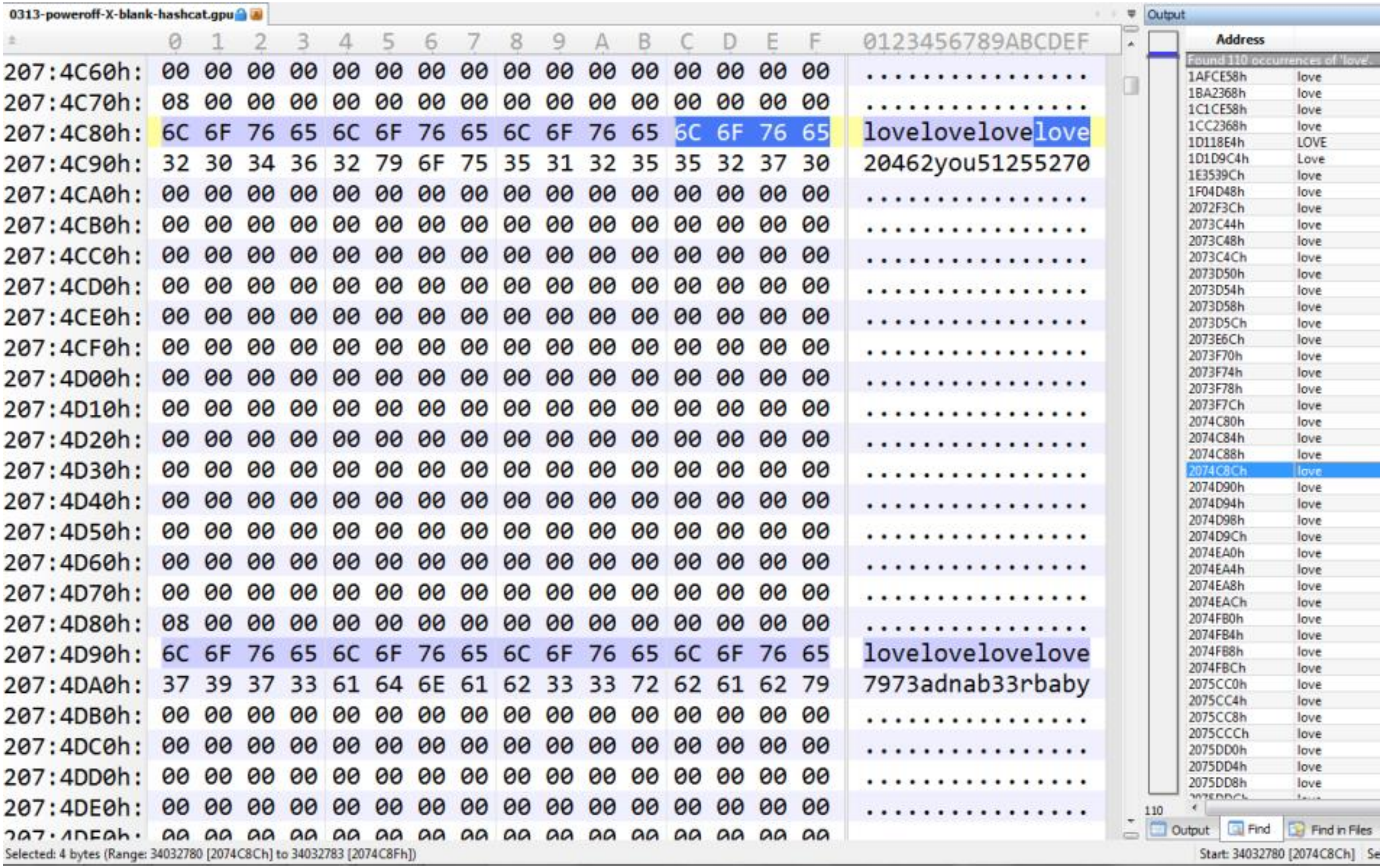
1gb GeForce GTX 720

cudaHashcat

- If you are cracking passwords on the GPU, what should be in the card's memory?

- Words!
- Take GPU memory dump
- Open in hex editor, search 'pass', 'love', #(*&#$(&%, etc.

cudaHashcat word data

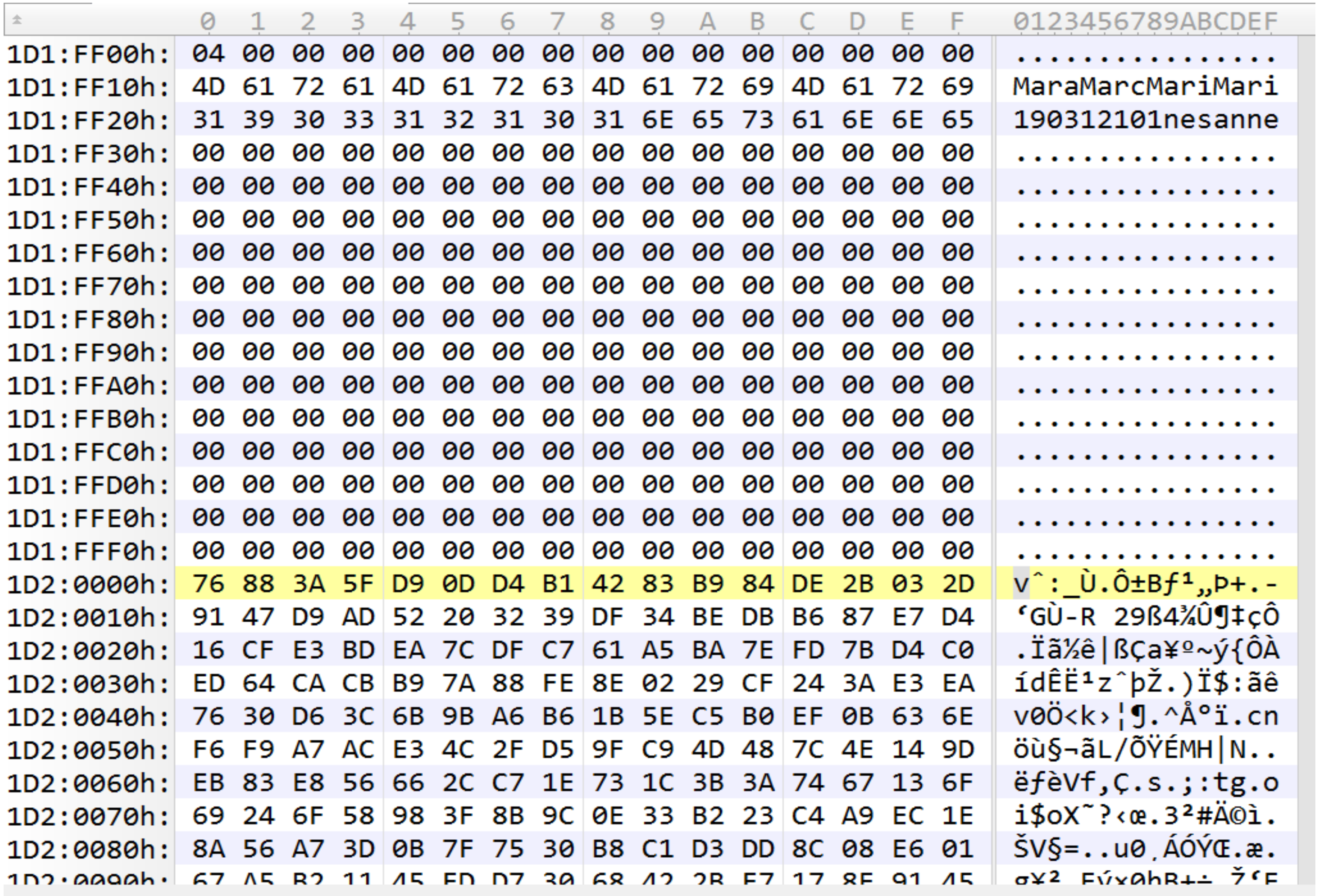


```
           0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
207:4C60h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4C70h: 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4C80h: 6C 6F 76 65 6C 6F 76 65 6C 6F 76 65 6C 6F 76 65   lovelovelovelove
207:4C90h: 32 30 34 36 32 79 6F 75 35 31 32 35 35 32 37 30   20462you51255270
207:4CA0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4CB0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4CC0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4CD0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4CE0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4CF0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4D00h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4D10h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4D20h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4D30h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4D40h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4D50h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4D60h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4D70h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4D80h: 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4D90h: 6C 6F 76 65 6C 6F 76 65 6C 6F 76 65 6C 6F 76 65   lovelovelovelove
207:4DA0h: 37 39 37 33 61 64 6E 61 62 33 33 72 62 61 62 79   7973adnab33rbaby
207:4DB0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4DC0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4DD0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4DE0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
207:4DF0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Selected: 4 bytes (Range: 34032780 [2074C8Ch] to 34032783 [2074C8Fh])
```

```
Address
Found 110 occurrences of 'love'..
1AFCE58h   love
1BA2368h   love
1C1CE58h   love
1CC2368h   love
1D118E4h   LOVE
1D1D9C4h   Love
1E3539Ch   love
1F04D48h   love
2072F3Ch   love
2073C44h   love
2073C48h   love
2073C4Ch   love
2073D50h   love
2073D54h   love
2073D58h   love
2073D5Ch   love
2073E6Ch   love
2073F70h   love
2073F74h   love
2073F78h   love
2073F7Ch   love
2074C80h   love
2074C84h   love
2074C88h   love
2074C8Ch   love
2074D90h   love
2074D94h   love
2074D98h   love
2074D9Ch   love
2074EA0h   love
2074EA4h   love
2074EA8h   love
2074EACh   love
2074FB0h   love
2074FB4h   love
2074FB8h   love
2074FBCh   love
2075CC0h   love
2075CC4h   love
2075CC8h   love
2075CCCh   love
2075DD0h   love
2075DD4h   love
2075DD8h   love
2075DDCh   love
110
Start: 34032780 [2074C8Ch]   Se
```

1gb GeForce GTX 720

cudaHashcat hashes



```
             0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
1D1:FF00h:  04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
1D1:FF10h:  4D 61 72 61 4D 61 72 63 4D 61 72 69 4D 61 72 69   MaraMarcMariMari
1D1:FF20h:  31 39 30 33 31 32 31 30 31 6E 65 73 61 6E 6E 65   190312101nesanne
1D1:FF30h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
1D1:FF40h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
1D1:FF50h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
1D1:FF60h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
1D1:FF70h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
1D1:FF80h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
1D1:FF90h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
1D1:FFA0h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
1D1:FFB0h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
1D1:FFC0h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
1D1:FFD0h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
1D1:FFE0h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
1D1:FFF0h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
1D2:0000h:  76 88 3A 5F D9 0D D4 B1 42 83 B9 84 DE 2B 03 2D   vˆ:_Ù.Ô±Bƒ¹„Þ+.-
1D2:0010h:  91 47 D9 AD 52 20 32 39 DF 34 BE DB B6 87 E7 D4   ‘GÙ-R 29ß4¾Û¶‡çÔ
1D2:0020h:  16 CF E3 BD EA 7C DF C7 61 A5 BA 7E FD 7B D4 C0   .Ïã½ê|ßÇa¥º~ý{ÔÀ
1D2:0030h:  ED 64 CA CB B9 7A 88 FE 8E 02 29 CF 24 3A E3 EA   ídÊË¹zˆþŽ.)Ï$:ãê
1D2:0040h:  76 30 D6 3C 6B 9B A6 B6 1B 5E C5 B0 EF 0B 63 6E   v0Ö<k›¦¶.^Å°ï.cn
1D2:0050h:  F6 F9 A7 AC E3 4C 2F D5 9F C9 4D 48 7C 4E 14 9D   öù§¬ãL/ÕŸÉMH|N..
1D2:0060h:  EB 83 E8 56 66 2C C7 1E 73 1C 3B 3A 74 67 13 6F   ëƒèVf,Ç.s.;:tg.o
1D2:0070h:  69 24 6F 58 98 3F 8B 9C 0E 33 B2 23 C4 A9 EC 1E   i$oX˜?‹œ.3²#Ä©ì.
1D2:0080h:  8A 56 A7 3D 0B 7F 75 30 B8 C1 D3 DD 8C 08 E6 01   ŠV§=..u0¸ÁÓÝŒ.æ.
1D2:0090h:  67 A5 B2 11 45 FD D7 30 68 42 2B F7 17 8E 91 45   g¥².Eý×0hB+÷.Ž‘E
```

128kb of high-entropy hashes

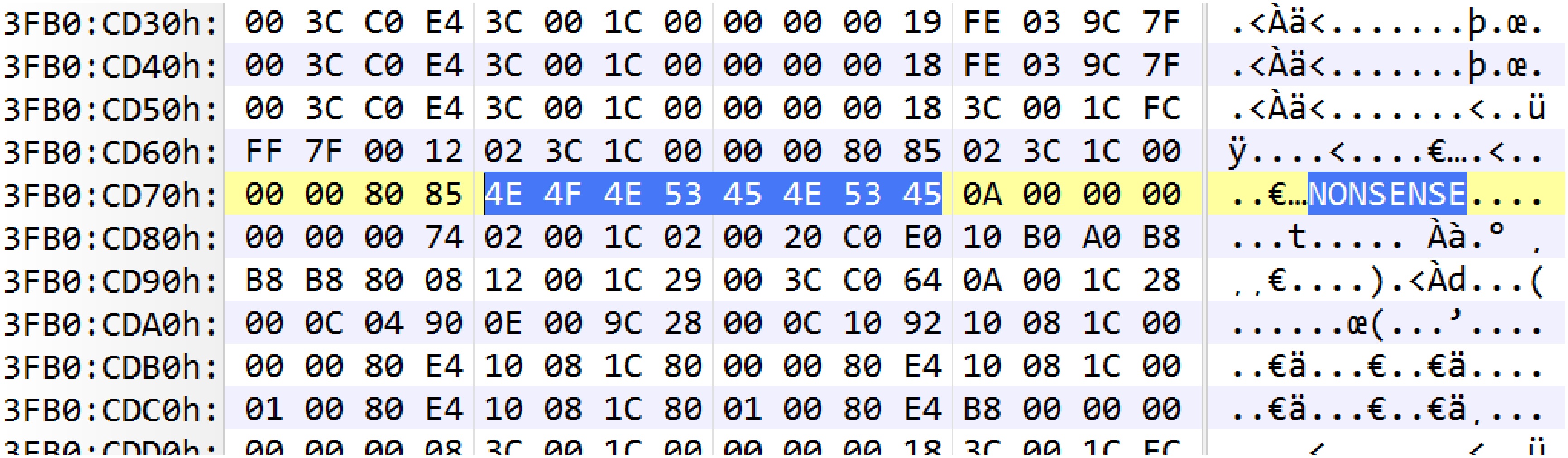
CUDA kernels

- small c functions
- execute many times in parallel
- kernel code is a single thread's work
- use primitive types/long arrays of data
- fastest memory access aligned to powers of 2
- thread groups (warps) of 32 threads

Kernel locations

- Kernels just after user-accessible memory
  - (@ 3b710000+ on 1gb card)
- Based on before/after differencing with hashcat having run
  - (hashcat blew up the top end of the card with differences)
- And memory stuffing kernels

Memory stuffing kernel



```
3FB0:CD30h: 00 3C C0 E4 3C 00 1C 00 00 00 00 19 FE 03 9C 7F  .<Àä<.......þ.œ.
3FB0:CD40h: 00 3C C0 E4 3C 00 1C 00 00 00 00 18 FE 03 9C 7F  .<Àä<.......þ.œ.
3FB0:CD50h: 00 3C C0 E4 3C 00 1C 00 00 00 00 18 3C 00 1C FC  .<Àä<.......<..ü
3FB0:CD60h: FF 7F 00 12 02 3C 1C 00 00 00 80 85 02 3C 1C 00  ÿ....<....€….<..
3FB0:CD70h: 00 00 80 85 4E 4F 4E 53 45 4E 53 45 0A 00 00 00  ..€…NONSENSE....
3FB0:CD80h: 00 00 00 74 02 00 1C 02 00 20 C0 E0 10 B0 A0 B8  ...t..... Àà.° ¸
3FB0:CD90h: B8 B8 80 08 12 00 1C 29 00 3C C0 64 0A 00 1C 28  ¸¸€....).<Àd...(
3FB0:CDA0h: 00 0C 04 90 0E 00 9C 28 00 0C 10 92 10 08 1C 00  ......œ(...’....
3FB0:CDB0h: 00 00 80 E4 10 08 1C 80 00 00 80 E4 10 08 1C 00  ..€ä...€..€ä....
3FB0:CDC0h: 01 00 80 E4 10 08 1C 80 01 00 80 E4 B8 00 00 00  ..€ä...€..€ä¸...
3FB0:CDD0h: 00 00 00 08 3C 00 1C 00 00 00 00 18 3C 00 1C FC  ....<.......<..ü
```

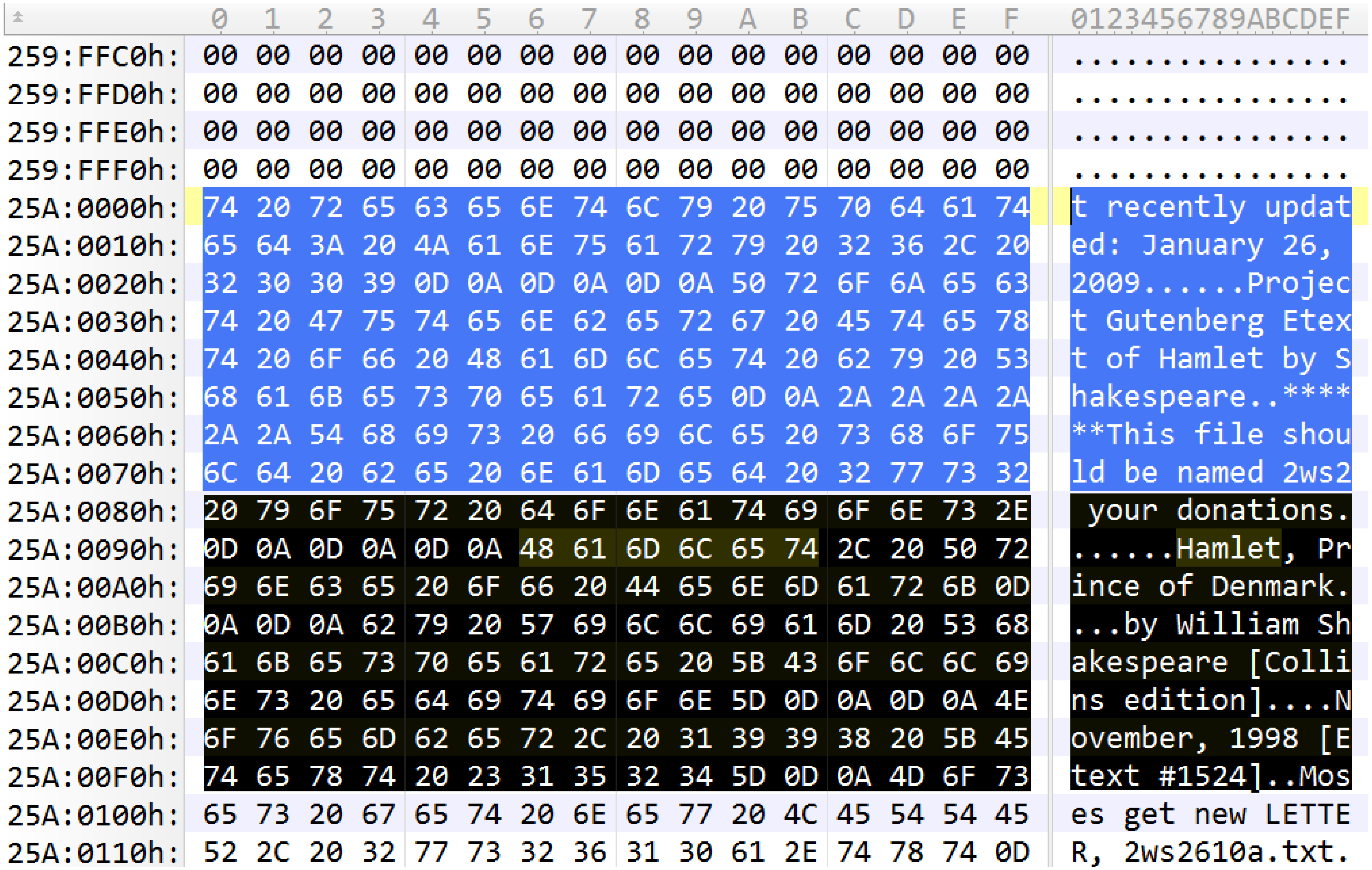
- 3 occurrences of marker phrase NONSENSE at lower end of ram in 1gb card
- At least one is likely the memory loading kernel (cudaMalloc)

Allocation blocking

- Sometimes the cards fill memory “in order”
- But oftentimes not.

- Blocking in 1024 byte blocks
  - 256-byte pairs
  - Second half first
- Best guess is sized to thread groups
  - (warps of 32)

To be or not to be



```
             0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
259:FFC0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
259:FFD0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
259:FFE0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
259:FFF0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
25A:0000h: 74 20 72 65 63 65 6E 74 6C 79 20 75 70 64 61 74  t recently updat
25A:0010h: 65 64 3A 20 4A 61 6E 75 61 72 79 20 32 36 2C 20  ed: January 26, 
25A:0020h: 32 30 30 39 0D 0A 0D 0A 0D 0A 50 72 6F 6A 65 63  2009......Projec
25A:0030h: 74 20 47 75 74 65 6E 62 65 72 67 20 45 74 65 78  t Gutenberg Etex
25A:0040h: 74 20 6F 66 20 48 61 6D 6C 65 74 20 62 79 20 53  t of Hamlet by S
25A:0050h: 68 61 6B 65 73 70 65 61 72 65 0D 0A 2A 2A 2A 2A  hakespeare..****
25A:0060h: 2A 2A 54 68 69 73 20 66 69 6C 65 20 73 68 6F 75  **This file shou
25A:0070h: 6C 64 20 62 65 20 6E 61 6D 65 64 20 32 77 73 32  ld be named 2ws2
25A:0080h: 20 79 6F 75 72 20 64 6F 6E 61 74 69 6F 6E 73 2E   your donations.
25A:0090h: 0D 0A 0D 0A 0D 0A 48 61 6D 6C 65 74 2C 20 50 72  ......Hamlet, Pr
25A:00A0h: 69 6E 63 65 20 6F 66 20 44 65 6E 6D 61 72 6B 0D  ince of Denmark.
25A:00B0h: 0A 0D 0A 62 79 20 57 69 6C 6C 69 61 6D 20 53 68  ...by William Sh
25A:00C0h: 61 6B 65 73 70 65 61 72 65 20 5B 43 6F 6C 6C 69  akespeare [Colli
25A:00D0h: 6E 73 20 65 64 69 74 69 6F 6E 5D 0D 0A 0D 0A 4E  ns edition]....N
25A:00E0h: 6F 76 65 6D 62 65 72 2C 20 31 39 39 38 20 5B 45  ovember, 1998 [E
25A:00F0h: 74 65 78 74 20 23 31 35 32 34 5D 0D 0A 4D 6F 73  text #1524]..Mos
25A:0100h: 65 73 20 67 65 74 20 6E 65 77 20 4C 45 54 54 45  es get new LETTE
25A:0110h: 52 2C 20 32 77 73 32 36 31 30 61 2E 74 78 74 0D  R, 2ws2610a.txt.
```

4gb GRID GPU, EC2

DFRWS 2015 Challenge

- 2GB GPU dump available for anyone to analyze
- Fun in the pulling-out-hair sort of way
- http://www.cs.uno.edu/~golden/gpu-malware-research.html

Card Images

- 1gb NVIDIA GT 720 (mine)
- 2gb NVIDIA GTX 750Ti (DFRWS Challenge)
- 4gb NVIDIA GRID K520 (Amazon)

2gb card map

type	content	start	end	size
data	Reserved?	0	2801a	163866
data		2ad18	2f3c07	2920175
data		307c64	316448	59364
data		316c20	500007	2003943
data		5004ae	699b07	1676889
data		699b08	700007	419071
data	Video/X	722800	294001f	35772447
nulls	**User memory**	**294001f**	**2940023**	
data		2940079	39c93e9	17339248
data		39c93eb	41b3060	8297589
repeat many		41b3062	499ccd7	8297589
data	**Unallocated**	**41ed967d**	**7f6b8008**	**1031661963**
data	footer	7f6b8009	7f6b8068	95
data		7f6cc100	7f6dc104	65540
data	Kernels?	7f6e9000	7f70914a	131402
data		7f709400	7f723002	105474

2gb NVIDIA GTX 750Ti (DFRWS Challenge)

4gb card map

type	content	start	end	size
data		0	1ff	511
data		3db180	3db3bf	575
repeat	noise			512+
data		3dedc0	3e017f	5055
nulls		3e017f	25a0000	
data	User	25a0000	325a01ff	805306879
nulls	Unallocated	325a01ff	ff413005	
data	footer	ff413005	ff415a31	10796
data		ff419a0c	ff4233da	39374
data		ff4d9a0c	ff4e33da	39374
data		ff8c0000	ff8d21fc	74236

4gb NVIDIA GRID K520 (Amazon)
with programs running

Next steps

- Collecting and borrowing video cards of assorted sizes
- Making a profile out of the maps for auto extraction of features
- Profit?

Resources

- NVIDIA CUDA docs
  - https://docs.nvidia.com/cuda/
- GPU Memory Dump tools and samples
  - http://www.cs.uno.edu/~golden/gpu-malware-research.html
- GPU in the cloud
  - http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using_cluster_computing.html
- Little mapper program
  - https://github.com/candicenonsense/nullfinder

Thank you!

- Contact: candice@egobsd.org
- @candicenonsense
- https://github.com/candicenonsense

details

“…array of multithreaded *Streaming Multiprocessors* (*SMs*).

When a CUDA program invokes a kernel, the blocks of the grid are enumerated and distributed to multiprocessors with available capacity.

The threads of a thread block execute concurrently on one multiprocessor, and multiple thread blocks can execute concurrently on one multiprocessor.

As thread blocks terminate, new blocks are launched on the vacated multiprocessors.”

http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#hardware-implementation

vectorAdd kernel

```
__global__ void
vectorAdd(const float *A, const float *B, float *C, int
numElements)
{
    int i = blockDim.x * blockIdx.x + threadIdx.x;

    if (i < numElements)
    {
        C[i] = A[i] + B[i];
    }
}
```

Source: CUDA samples http://docs.nvidia.com/cuda/cuda-samples/