TMTO[dot]ORG: Hardware Comparison – 8 x GTX580 vs. 4 x HD6990

Author: Jason R. Davis
Site: TMTO[dot]ORG

Table of Contents

Created Date: May, 11th, 2011
Last Edited: May 11th, 2011

Foreword

There has been an emerging class of software making its way into the toolboxes of security auditors and hackers alike – GPU based (OpenCL or CUDA) brute force applications. GPU based brute forcing is an amazing leap forward in raw speed. I naturally took some serious interest in this – because the *Epiphany* portion of TMTO[dot]ORG is driven by GPU based brute forcing. There were many obstacles that had to be overcome, but the benefits outweighed them.

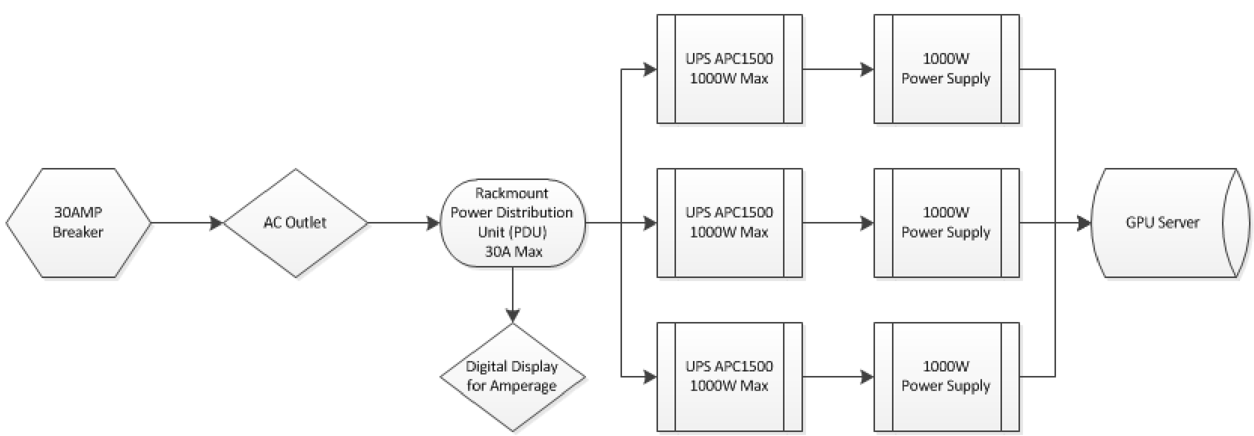
This article is focused on a specific application that has shown much promise and continues to quickly progress with new features and bug fixes. This program is oclHashcat (http://www.hashcat.net) .

I'm not going to get into the semantics of how the program works – it's a GPU accelerated brute force utility. I'm using the "lite" version for the benchmarks as you will later see.

Chapter 1 – Building: *The Machine*

There were two obstacles I would have to overcome before I started building a large scale GPU cracking box. The first one is power and the second was heat.

POWER:
Determining the proper wattage supply and the supporting infrastructure came first. I started out with determining the number of cards in which I wanted to run in parallel. This number actually turned into a range, which was 6-8 GPUs. It turned into a range because I was looking at a couple different motherboards. Some of those motherboards supported six GPUs and others support seven or eight GPUs depending on the profile of the card. Now that I had a range I decided I would just build the underlying power supply structure to support up to eight GPU cards with a max watt requirement of 300W per GPU. This means I needed sufficient power supplies to support at least 2400W (2.4kW) for all eight GPUs. This also meant I needed some head room so three 1kW or 1.2kW power supplies would be more than sufficient. I decided to use 120V, instead of 240V. Next I had to determine how many amps would be drawn while the server was under full utilization. W / V = A, watts divided by the voltage equals the estimated amps. 2400 (W) / 120 (V) = 20 (A). I would need to at least support a 20A draw. This means I needed a 30A breaker, leading to a single 120V wall outlet. From that outlet I would put in a rack mounted power distribution unit (PDU) with a digital amp display, that way I could see the draw without metering. From the PDU I would connect a UPS (APC 1500), which would condition and clean up the power, into the PDU – and from the UPS I would connect to one 1kW or 1.2kW power supply per UPS.



HEAT:
Heat would be easier to address. I didn't want to fool around with water cooling on this build, so I just needed to keep the room at a cool temperature. I decided on 68F for the room temperature. This wasn't frosty, but would be sufficient to keep everything within operating temperatures. I installed an additional free standing AC unit to supplement the rooms central AC. This would lighten the burden on the central AC, and allow me to control the temperature in the room better. I would supplement the internal airflow of the server itself with additional fans if needed. I expected to add a few extra fans in certain places, but in the end I never had to.

I debated many different configurations. In the end I decided on the TYAN FT72-B7015 bare bones server. There are many attractive features of this box:

- 3 x 1000W PSUs @ 120V
- 8 x PCI-E x16 Slots
- 3 x 12cm Fans

Source: http://www.tyan.com/product_SKU_spec.aspx?ProductType=BB&pid=412&SKU=600000150

This server has power to support up to 3000W. It has support for up to 8 GPU cards (dual slot). It also has massive fans for air cooling. It's almost like this box was built for my exact purpose. The company RenderStream (http://www.renderstream.com/) sells the server in a bunch of different configuration. I purchased the barebones version of the VDACTr8 (http://www.renderstream.com/HPC.html#1).

Originally intended for GPU based rendering: http://blog.renderstream.com/?p=833

RenderStream actually made a blog post about my server: http://blog.renderstream.com/?p=1239

Once I received the server I installed some other components:

- Intel(R) Xeon(R) CPU X5690 @ 3.47GHz
- 16GB (2 x 8GB) RAM
- Intel X25-M (120GB) SSD

Then I went into GPU buying mode. I reviewed the specifications of many different video cards. In the end it was a decision between the 8 x GTX580 or 4 x HD6990.

Chapter 2 – 8 x GTX580

The GTX580 cards are EVGA Superclocked. Each with 512 CUDA cores, graphics clock speeds of 772MHz, processor clock speed of 1594MHz, and memory clock speeds of 2004MHz. Stock clock speeds are 772MHz/1544MHz. Each card has a 50MHz bump in processor speed over the stock setting, hence "Superclocked".

Here is a benchmark:

```
gpu-01 x86_64 # ./cudaHashcat-lite64.bin --runtime=60 -n 800 --gpu-loops 1024 -1 ?u?l?d?s 11111111111111111111111111111111 ?1?1?1?1?1?1?1?1
cudaHashcat-lite v0.3 starting...

Platform: NVidia compatible platform found
Watchdog: Temperature limit set to 90c
Device #1: GeForce GTX 580, 1535MB, 1594Mhz, 16MCU
Device #2: GeForce GTX 580, 1535MB, 1594Mhz, 16MCU
Device #3: GeForce GTX 580, 1535MB, 1594Mhz, 16MCU
Device #4: GeForce GTX 580, 1535MB, 1594Mhz, 16MCU
Device #5: GeForce GTX 580, 1535MB, 1594Mhz, 16MCU
Device #6: GeForce GTX 580, 1535MB, 1594Mhz, 16MCU
Device #7: GeForce GTX 580, 1535MB, 1594Mhz, 16MCU
Device #8: GeForce GTX 580, 1535MB, 1594Mhz, 16MCU
[s]tatus [p]ause [r]esume [q]uit =>
NOTE: Runtime limit reached, aborting...

Status....: Aborted
Current...: ***F92e
Monitor1..: 99% GPU, 67c Temp
Monitor2..: 99% GPU, 65c Temp
Monitor3..: 99% GPU, 68c Temp
Monitor4..: 99% GPU, 69c Temp
Monitor5..: 99% GPU, 67c Temp
Monitor6..: 99% GPU, 67c Temp
Monitor7..: 99% GPU, 64c Temp
Monitor8..: 99% GPU, 65c Temp
Speed.GPU*: 18.3G/s
Progress..: 1073741824000/69833729609375 (1.54%)
Running...: 1 min, 1 sec
Estimated.: 1 hour, 4 mins

Started: Thu Apr 28 17:33:09 2011
Stopped: Thu Apr 28 17:35:40 2011
```

Chapter 3 – 4 x HD6990

The HD6990 cards are XFX. Each has 3072 stream processors (cores), graphics clock speeds of 880MHz, and memory clock speeds of 1250MHz. Stock clock speeds are 830MHz/1250MHz. Each card has a 50MHz bump in processor speed over the stock setting. The overclock is easily achieved with the XFX version of the card by simply flipping a switch that causes the card to boot in a "high performance" mode. The voltage supplied to the card is also increased slightly to support the additional speed. It should also be noted that each HD6990 card is actually dual GPU.

Here is a benchmark:

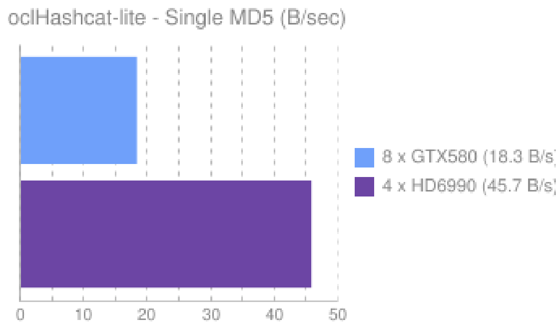
```
gpu-01 x86_64 # ./oclHashcat-lite64.bin -1 ?l?d?s?u --pw-min 8 --runtime 60 -n 800 -m 0 ffffffffffffffffffffffffffffffff ?1?1?1?1?1?1?1?1
oclHashcat-lite v0.3 starting...

Platform: AMD compatible platform found
Watchdog: Temperature limit set to 90c
Device #1: Cayman, 1024MB, 0Mhz, 24MCU
Device #2: Cayman, 1024MB, 0Mhz, 24MCU
Device #3: Cayman, 1024MB, 0Mhz, 24MCU
Device #4: Cayman, 1024MB, 0Mhz, 24MCU
Device #5: Cayman, 1024MB, 0Mhz, 24MCU
Device #6: Cayman, 1024MB, 0Mhz, 24MCU
Device #7: Cayman, 1024MB, 0Mhz, 24MCU
Device #8: Cayman, 1024MB, 0Mhz, 24MCU
[s]tatus [p]ause [r]esume [q]uit =>
NOTE: Runtime limit reached, aborting...

Status....: Aborted
Current...: ***Qw(la
Monitor1..: 99% GPU, 82c Temp
Monitor2..: 99% GPU, 81c Temp
Monitor3..: 99% GPU, 81c Temp
Monitor4..: 99% GPU, 80c Temp
Monitor5..: 99% GPU, 73c Temp
Monitor6..: 99% GPU, 78c Temp
Monitor7..: 98% GPU, 81c Temp
Monitor8..: 98% GPU, 75c Temp
Speed.GPU*:   45.7G/s
Progress..: 2748107980800/6634204312890625 (0.04%)
Running...: 1 min, 0 secs
Estimated.: 1 day, 16 hours

Started: Fri Apr 29 18:12:06 2011
Stopped: Fri Apr 29 18:13:11 2011
```

Chapter 4 – Conclusion



There are a few things to take into consideration. The OpenCL version of OCLHashcat Lite (oclHashcat-lite64.bin) utilizes "bit align" to gain 4x (even 5x) MD5 generation speeds. The CUDA version (cudaHashcat-lite64.bin) does not. Now this isn't the developers fault. NVIDIA simply doesn't provide a "bit align" function in the CUDA SDK at this time. I'm willing to speculate that the performance gap observed will decrease when/if NVIDIA adds that functionality to their SDK/GPUs. The other thing to take into consideration is that these speeds were recorded against a single MD5 hash. The performance of both cards drops as the list of hashes being tested increases.

Taking Advantage of Bit Align: http://www.golubev.com/blog/?tag=bfi_int