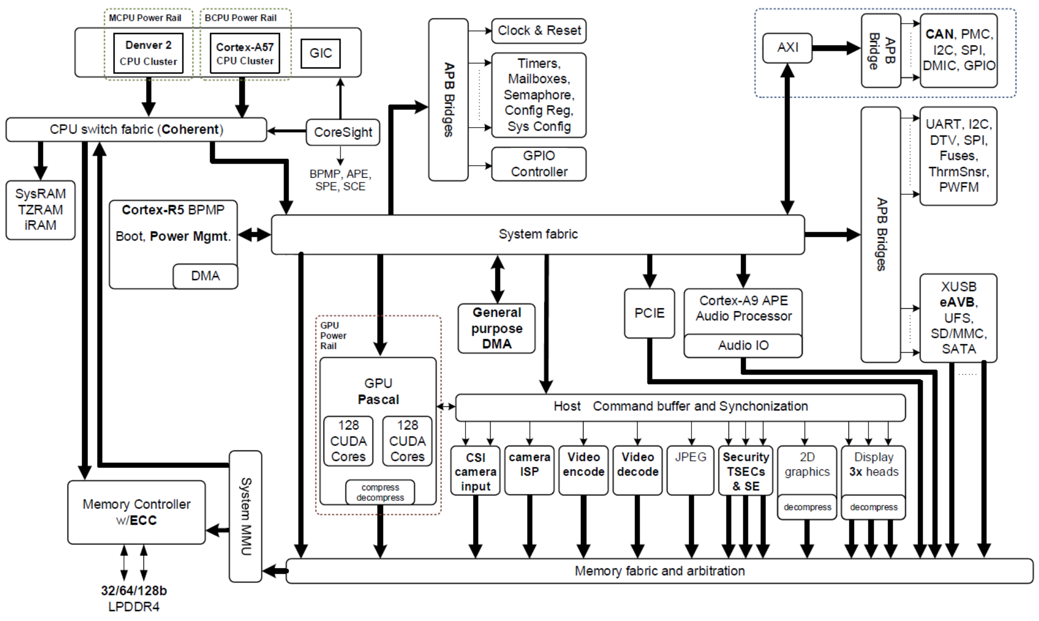
Progress after 4/13 project check-in meeting

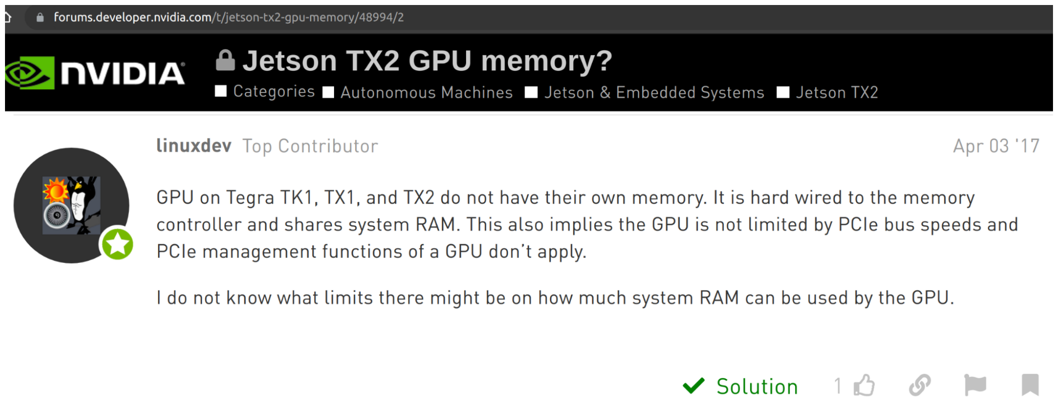
Rajesh:

- Logistics
    - Resolved storage issue with cloudlab: Unmounted drive had to be used
    - Resolved unresponsive issue with c4130: Ubuntu 18.04 works as expected
    - Resolved SegmentationFault issue for large matrix generation: Malloc/write needed update

3/25

- Strategy
    - Resource allocation concerns
    - Seek advice on experiment space (Matrix sizes, device configurations, mem to matsize)
        - Previously 24kx24k
        - Constants: Duration is immaterial. Contention is fine. In fact, necessary for practical modeling
        - Find parameters first: Power? Recreate noise (n-1) This is where Jetson might help?
        - Locality: Time and space separately; Experiment: Influence of (n-1) GPUs with base (as adversarial as possible, memcpy intensive) vs test-workload. Cross of GEMM, RESNET->less compute intensive relatively, .. etc
        - Incorporate profiling into the plan: sgemm-cuda/prof_sgemm.sh
        - Beyond: Vary things like power utilizing sudo
    - Previous results/learnings
- Repository
    - SGEMM source double check?
- JetsonTX2
    - How do we determine matrix size?
        - Memory stress but activity too: 8G/3? 3 -> A, B, result. On datacenter GPU, more power draw expected here due to DRAM accesses
        - Compute stress 256 SP’s
    - On-chip interconnect, direct 8G DRAM access https://developer.nvidia.com/blog/jetson-tx2-delivers-twice-intelligence-edge/




forums.developer.nvidia.com/t/jetson-tx2-gpu-memory/48994/2

Jetson TX2 GPU memory?

Categories · Autonomous Machines · Jetson & Embedded Systems · Jetson TX2

linuxdev Top Contributor Apr 03 '17

GPU on Tegra TK1, TX1, and TX2 do not have their own memory. It is hard wired to the memory controller and shares system RAM. This also implies the GPU is not limited by PCIe bus speeds and PCIe management functions of a GPU don't apply.

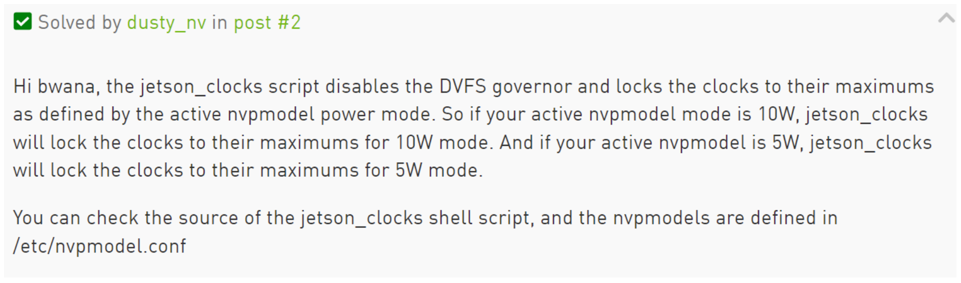
I do not know what limits there might be on how much system RAM can be used by the GPU.

Solution 1

3/11

Meeting before Spring Break

- Cloudlab resource usage question?
  - Reservation attempted for 3/14
- Jetson TX2
  - Install CUDA
    - Ethernet does not seem to work? No ssh yet
      - Reflash through SDK manager? Cable. **WiFi suggested**
  - Get sgemm-cuda working (sizes tailored to 2SM, 128SPs each)
  - Vary frequency through kernel to simulate GPUboost. Check for observable variations temperature, power, utilization through sweep across simulator space
    - Ashwin wants to share some of his previous work with us tonight
    - Fine-grained power limit control Source. Also, fan pwm control. Things are surprisingly easy to configure and flexible /usr/bin/jetson_clocks


✅ Solved by dusty_nv in post #2

Hi bwana, the jetson_clocks script disables the DVFS governor and locks the clocks to their maximums as defined by the active nvpmodel power mode. So if your active nvpmodel mode is 10W, jetson_clocks will lock the clocks to their maximums for 10W mode. And if your active nvpmodel is 5W, jetson_clocks will lock the clocks to their maximums for 5W mode.

You can check the source of the jetson_clocks shell script, and the nvpmodels are defined in /etc/nvpmodel.conf

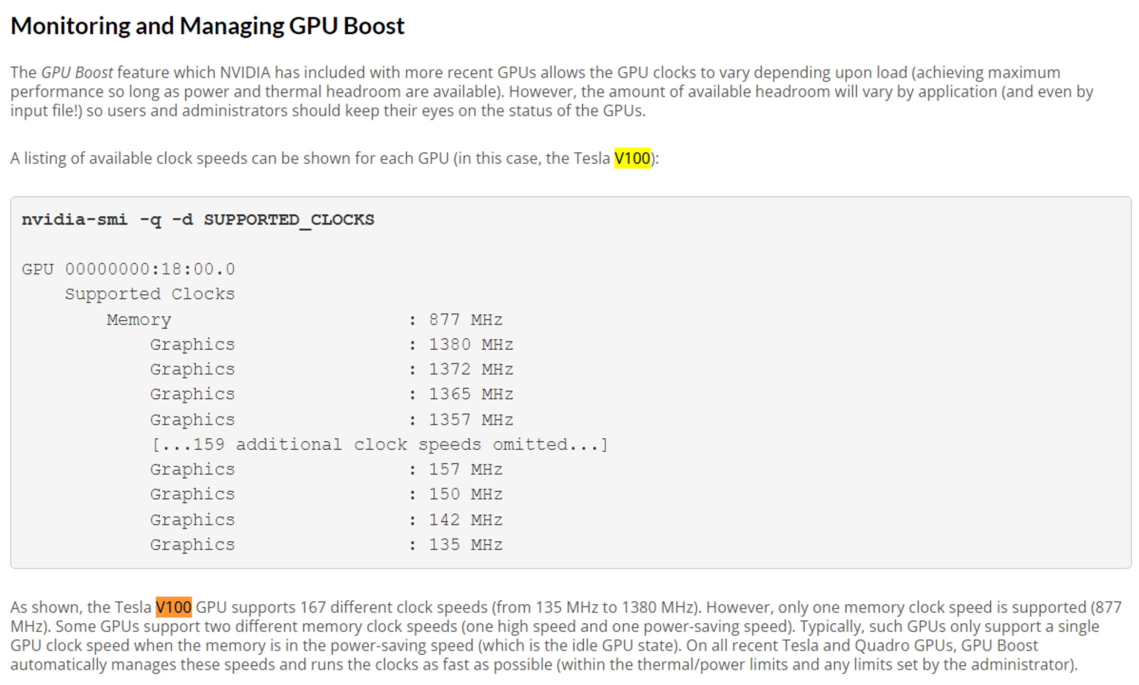
3/4

Minutes:

- Given the somewhat contradictory notion of autoboost and that other work(such as tail at scale) frequently notes this, use the first task to understand GPUboost and effects on variability
  - Study GPU Boost on SGEMM on CloudLab machines with 4-6 GPUs. This will also serve as a ramp up on scripts/setup used by Akhil. Also rule out possibility of variability being an artifact of boost
  - Jetson board as a side task for fine-grained measurements with the objective of outlining effects of boost. NeuOS may have some details on this
- (tail) Latency dimension is interesting for inference workloads. Most existing work is throughput focussed. Determine latency sensitive workloads. Compare compute vs latency sensitive temporally
- Don’t worry about mitigation for now. Characterization/profiling/identification is more important.
- Most prior work does not focus on GPU only systems.

Updates from previous:

- Autoboost (aka GPU boost) is enabled by default



Monitoring and Managing GPU Boost

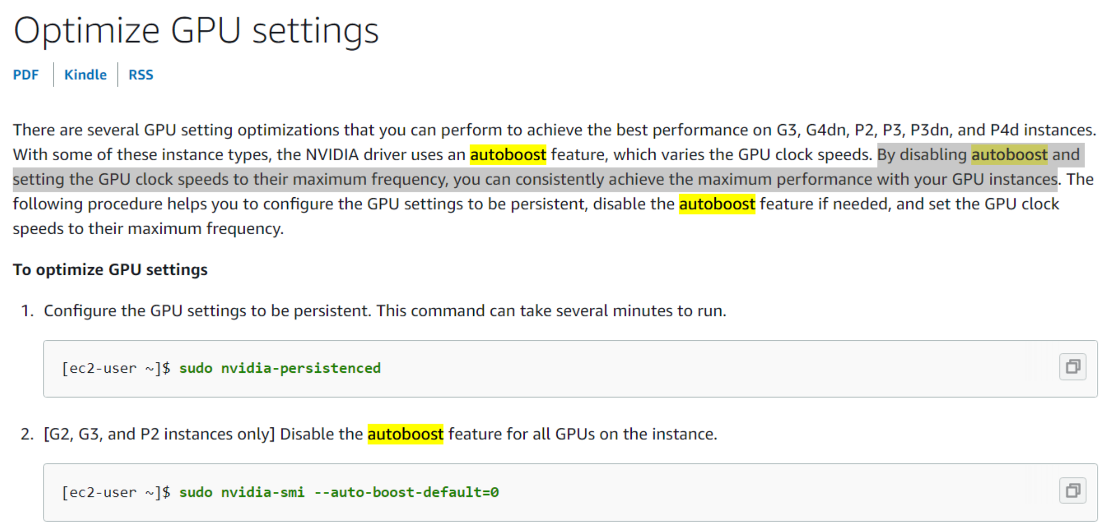
The *GPU Boost* feature which NVIDIA has included with more recent GPUs allows the GPU clocks to vary depending upon load (achieving maximum performance so long as power and thermal headroom are available). However, the amount of available headroom will vary by application (and even by input file!) so users and administrators should keep their eyes on the status of the GPUs.

A listing of available clock speeds can be shown for each GPU (in this case, the Tesla V100):

```
nvidia-smi -q -d SUPPORTED_CLOCKS

GPU 00000000:18:00.0
    Supported Clocks
        Memory                      : 877 MHz
            Graphics                : 1380 MHz
            Graphics                : 1372 MHz
            Graphics                : 1365 MHz
            Graphics                : 1357 MHz
            [...159 additional clock speeds omitted...]
            Graphics                : 157 MHz
            Graphics                : 150 MHz
            Graphics                : 142 MHz
            Graphics                : 135 MHz
```

As shown, the Tesla V100 GPU supports 167 different clock speeds (from 135 MHz to 1380 MHz). However, only one memory clock speed is supported (877 MHz). Some GPUs support two different memory clock speeds (one high speed and one power-saving speed). Typically, such GPUs only support a single GPU clock speed when the memory is in the power-saving speed (which is the idle GPU state). On all recent Tesla and Quadro GPUs, GPU Boost automatically manages these speeds and runs the clocks as fast as possible (within the thermal/power limits and any limits set by the administrator).



Optimize GPU settings

PDF | Kindle | RSS

There are several GPU setting optimizations that you can perform to achieve the best performance on G3, G4dn, P2, P3, P3dn, and P4d instances. With some of these instance types, the NVIDIA driver uses an autoboost feature, which varies the GPU clock speeds. By disabling autoboost and setting the GPU clock speeds to their maximum frequency, you can consistently achieve the maximum performance with your GPU instances. The following procedure helps you to configure the GPU settings to be persistent, disable the autoboost feature if needed, and set the GPU clock speeds to their maximum frequency.

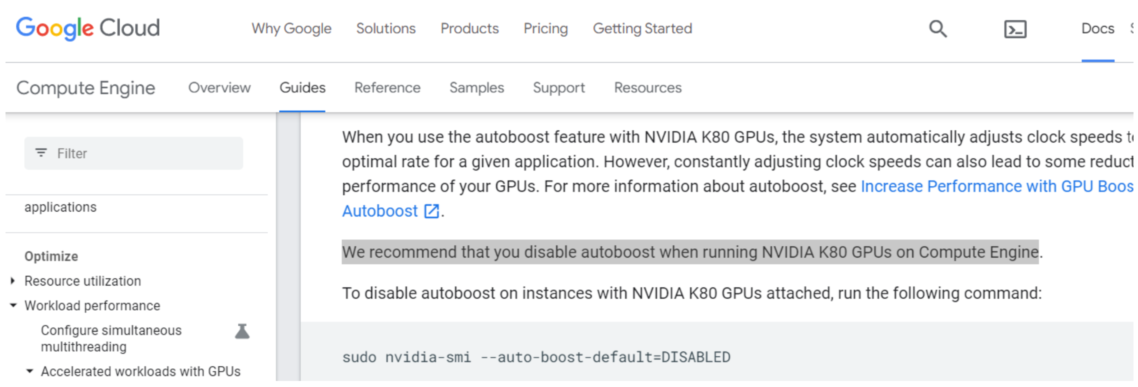
To optimize GPU settings

1. Configure the GPU settings to be persistent. This command can take several minutes to run.

```
[ec2-user ~]$ sudo nvidia-persistenced
```

2. [G2, G3, and P2 instances only] Disable the autoboost feature for all GPUs on the instance.

```
[ec2-user ~]$ sudo nvidia-smi --auto-boost-default=0
```



When you use the autoboost feature with NVIDIA K80 GPUs, the system automatically adjusts clock speeds t optimal rate for a given application. However, constantly adjusting clock speeds can also lead to some reduct performance of your GPUs. For more information about autoboost, see Increase Performance with GPU Boos Autoboost.

We recommend that you disable autoboost when running NVIDIA K80 GPUs on Compute Engine.

To disable autoboost on instances with NVIDIA K80 GPUs attached, run the following command:

```
sudo nvidia-smi --auto-boost-default=DISABLED
```

- Paper ashwin referenced
    - Latency vs energy efficiency in autonomous driving DNNs

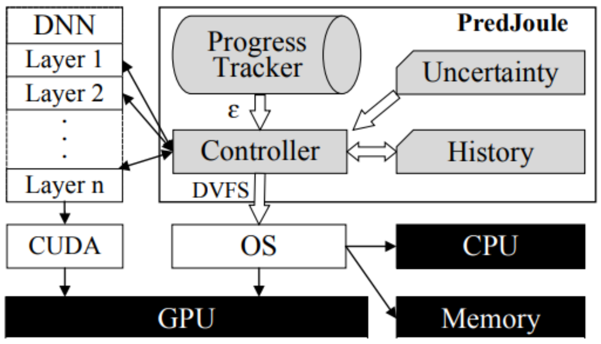


Fig. 3: Design overview of PredJoule.

as NVIDIA TX2 and PX2 [36]), there is at least 5 differen
DVFS adjustments:

- Small Core Activated Mask (M_s)
- Small Core Frequency (F_s)
- Big Core Activated Mask (M_b)
- Big Core Frequency (F_b)
- GPU Frequency (F_{GPU})
- Memory frequency (G_m)

- Bilge Acun's research in MLSys efficiency
  - Variability: Formalization in thesis
  - Interesting research utilization plot
  - Things that it affects like tail latency,
  - Load balancing
  - Formalizing HPC efficiency
- Need some advice on direction/start point
- References presented in this meeting:
  - https://www.usenix.org/system/files/atc20-paper332-slides-bateni.pdf
  - http://charm.cs.illinois.edu/newPapers/18-06/thesis.pdf
  - https://www.youtube.com/watch?v=LPrFL2gWmTY
  - https://scholar.google.com/citations?user=RMTfWQkAAAAJ&hl=en
  - https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9407135
  - https://cacm.acm.org/magazines/2013/2/160173-the-tail-at-scale/fulltext
  - https://dl.acm.org/doi/pdf/10.1145/2925426.2926289
  - https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7598172
  - https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6702627
  - https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7529982

2/25

- Project proposal draft submitted. No meeting as we used Office Hours to discuss 839 assignment

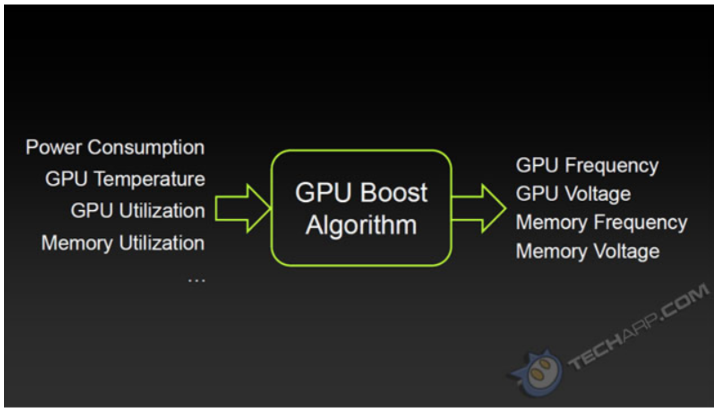
2/18

- First meeting with Ashwin

- Two possible approaches:
- Take a single machine and try effects of spatial and temporal locality on the same board
  - Prove manual placement of workload is better than random
  - Objective: Be able to ascertain ideal placement factoring in power variability on real case machines
  - Reference to Ashwin's work on the Jetson boards
  - Can use dP/dt to assume variability as long as it is within the envelope
  - We might not have access
- Explaining GPU boost
  - Check if previous work had this enabled by default
- Identify parameters to quantify power variability via profiling + those that can be treated as constants during analysis
- Identify data parallel workloads and patterns in ML workloads that tend to exhibit such variations

Questions from previous meeting:

- Power draw over PCIe
- cuBLAS SGEMM reflecting realism?
- Runtime duration primary component in measurements
- Summit Analysis for spatial locality effects
- Distinguish DRAM/MC power useful for irregular workloads and attribution? TGP vs TDP
- Similar to di/dt should we try dP/dt to first experimentally determine effects such as throttling etc? Testbench of sorts using something like PowerNetics https://www.tomshardware.com/reviews/power-consumption-measurement-cpu-gpu-components-powenetics,5481.html
- Is there value in studying GPU Boost since it is within the power envelope?



- Can we attempt data parallel algorithms like BytePS or Horovod to measure this new dimension to have a holistic analysis?

2/11

- Initial discussion. Overview of project and reading resources

References

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/optimize_gpu.html
https://www.microway.com/hpc-tech-tips/nvidia-smi_control-your-gpus/