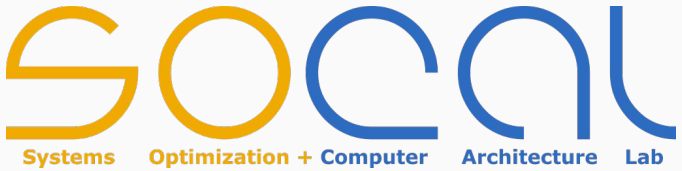
Performance and Power Challenges in Data Center GPUs

Daniel Wong
dwong@ece.ucr.edu
University of California, Riverside
Department of Electrical and Computer Engineering

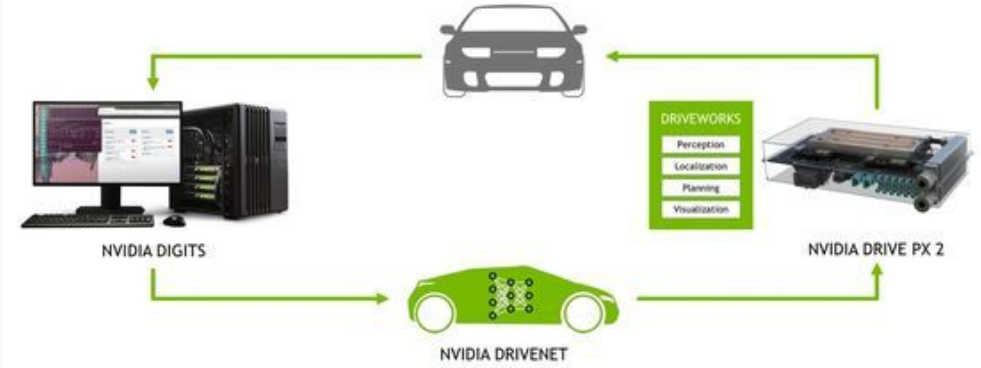


More than just graphics

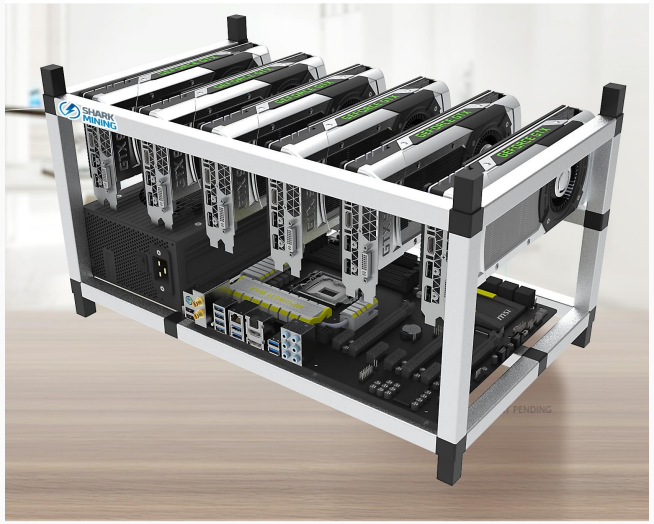
GPUs are everywhere

Data Centers

Embedded Systems



Autonomous Cars

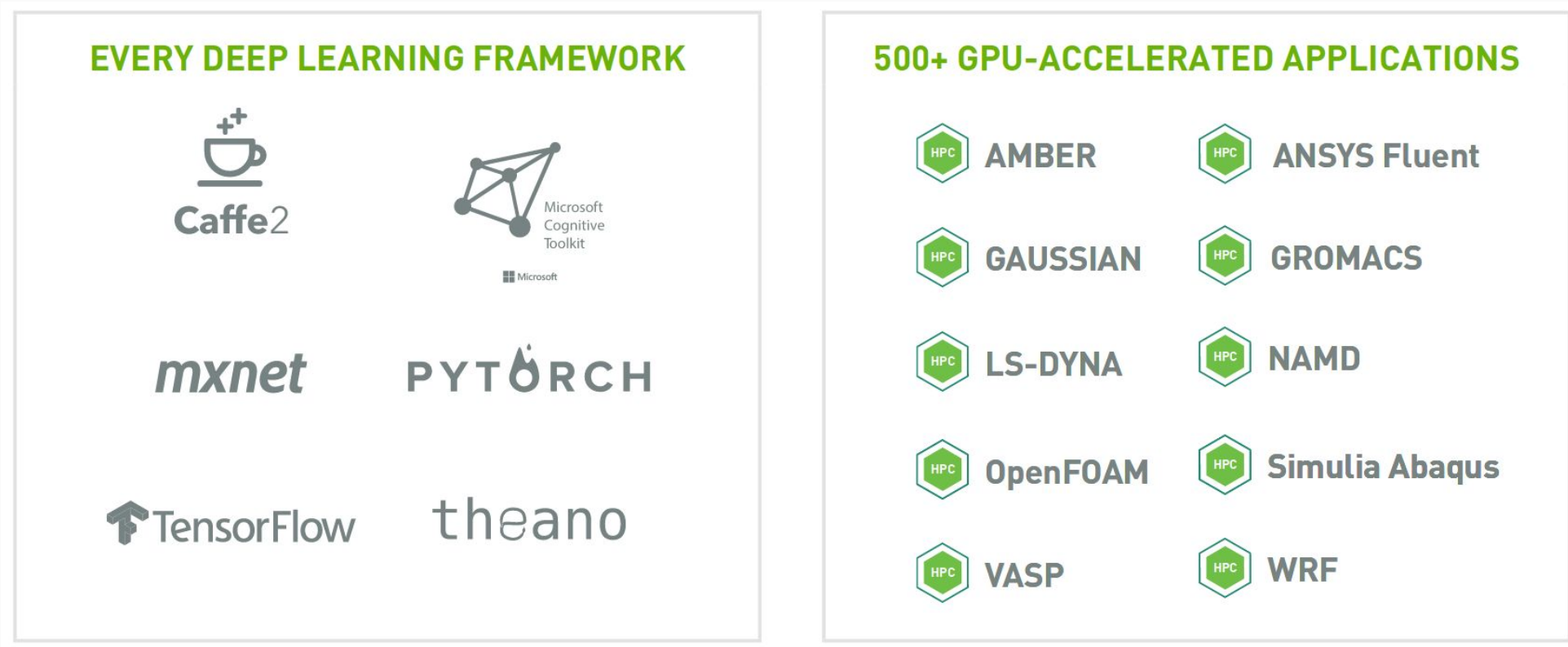


Cryptocurrency Mining

Picture sources: Nvidia

GPUs can do (almost) everything

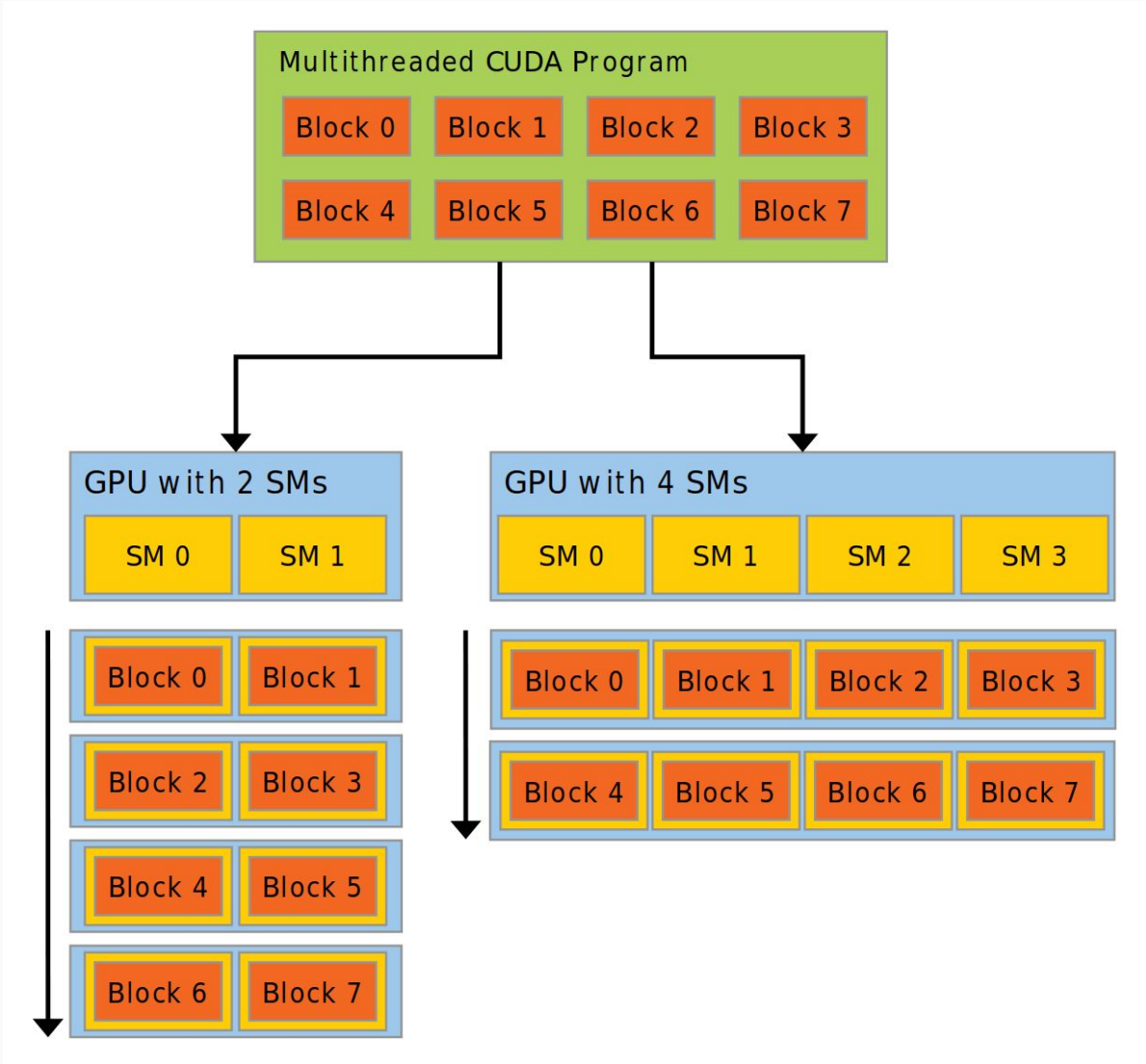
- Deep Learning, Mining, Graphics, HPC, Database, Network



GPUs are massively parallel accelerators

GPUs leverages *Data-level Parallelism*

- CUDA program is partitioned into a grid of blocks
- Each block is then scheduled to an SM within the GPU



GPU becoming more specialized

Modern GPU “Processing Block”

- 32 Threads
- 16 INT
- 16 single-precision FP
- 8 double-precision FP
- 4 SFU (sin, cos, log)
- 2 Tensor units for DNN
- 64KB RF

GPU Streaming Multiprocessor

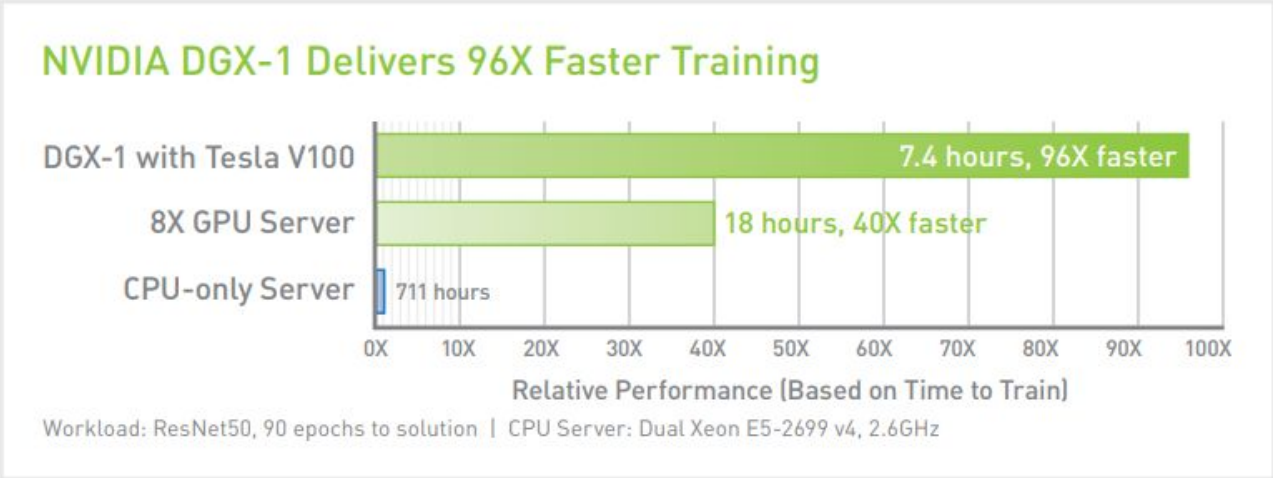
- Contains 4 “Processing Blocks”
- Each independently schedules a set of 32 threads called a warp
- Share L1 Cache between blocks

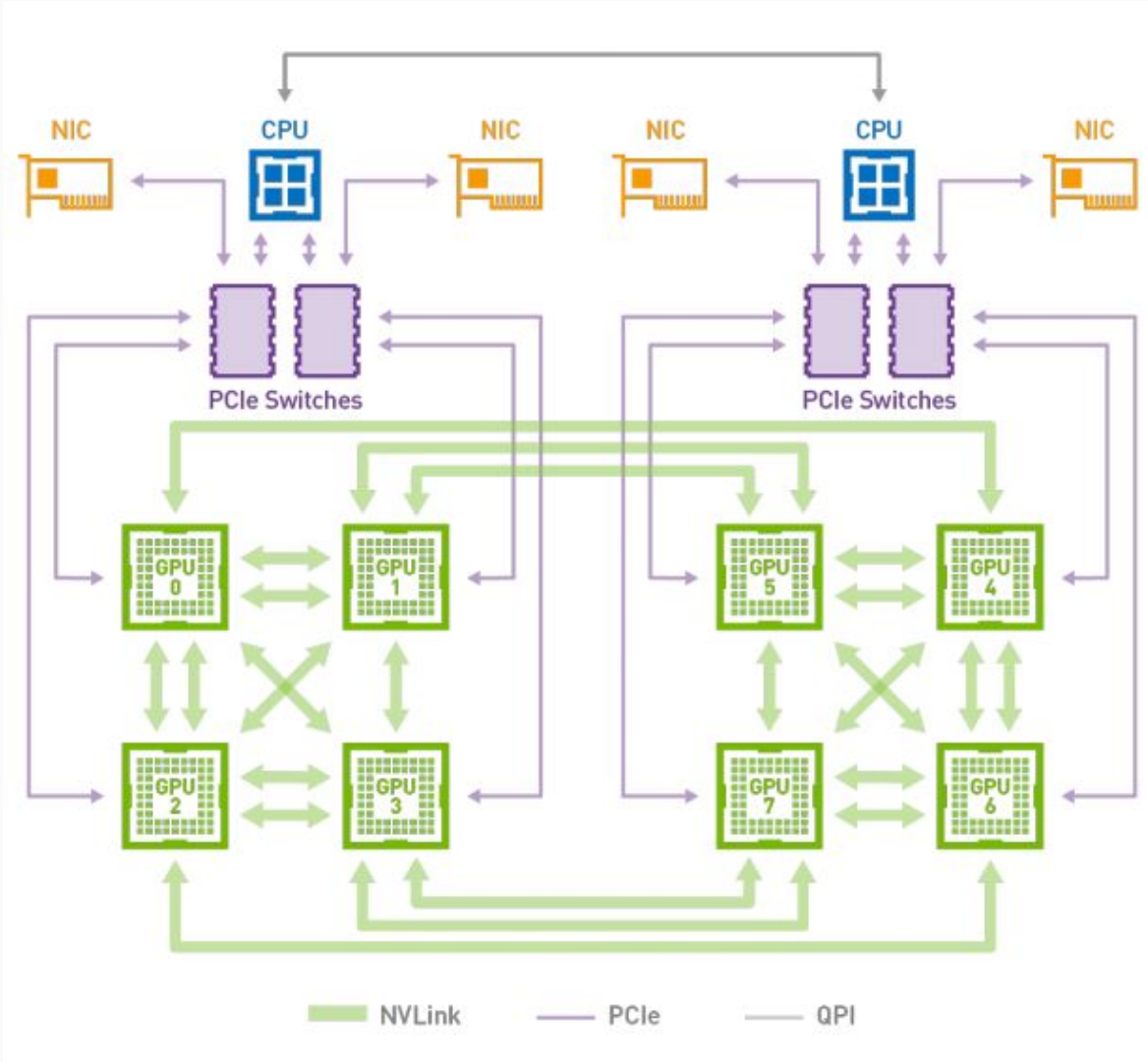
GPU Hardware

- V100 has 80 SM
- 5376 FPU
- Peak 15.7 TFLOPS

GPU “Data center in a box”

- DGX
  - A Multi-GPU “Node”
  - 300GB/s NVlink 2.0 cube mesh
  - 1 PFLOPS
  - Faster Machine Learning





NVIDIA

DATA CENTER PRODUCTS SOLUTIONS APPS FOR DEVELOPERS TECHNOLOGIES

DGX-1 OVERVIEW TOUR SPECS ORDER NOW

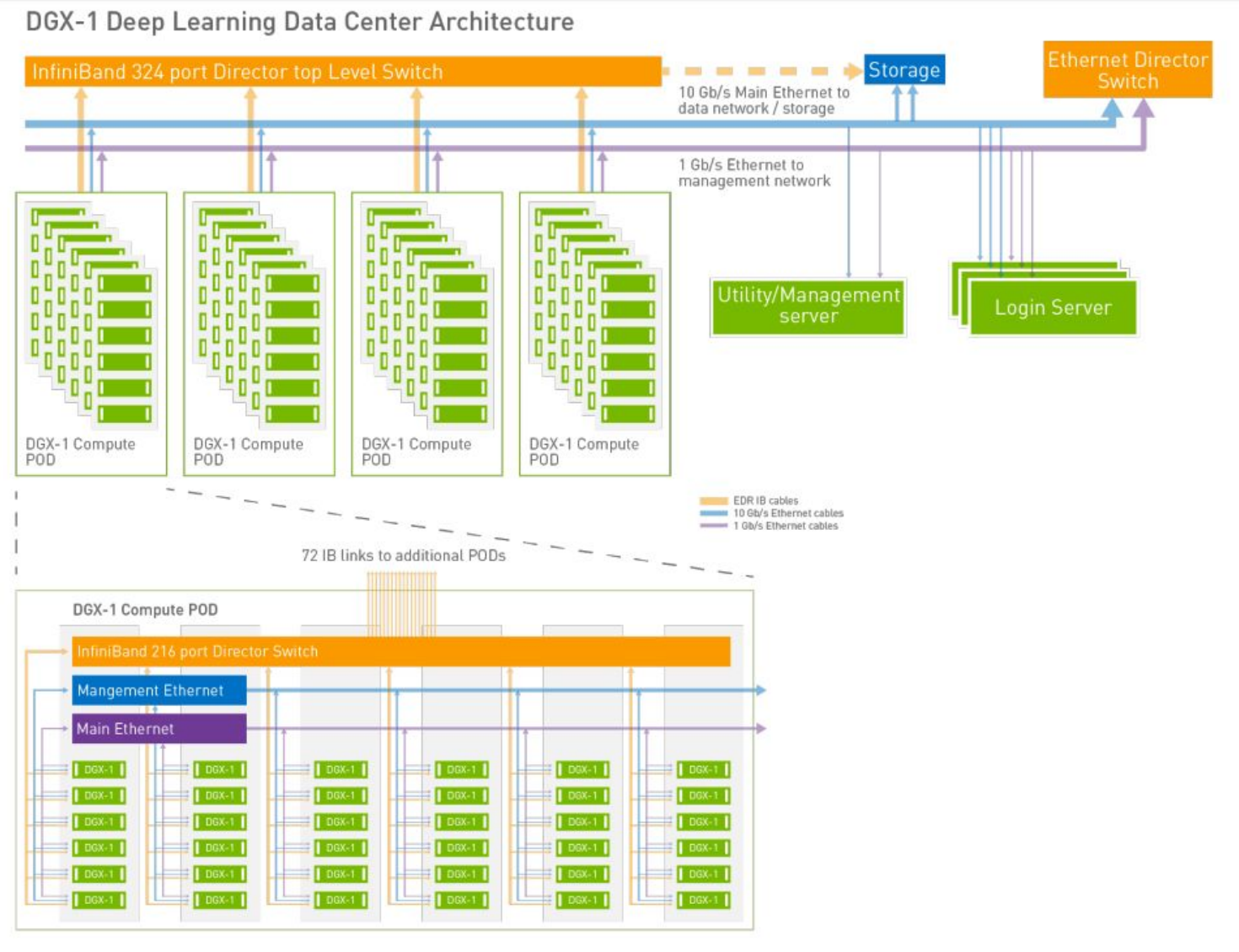
NVIDIA DGX-1

Essential Instrument of AI Research

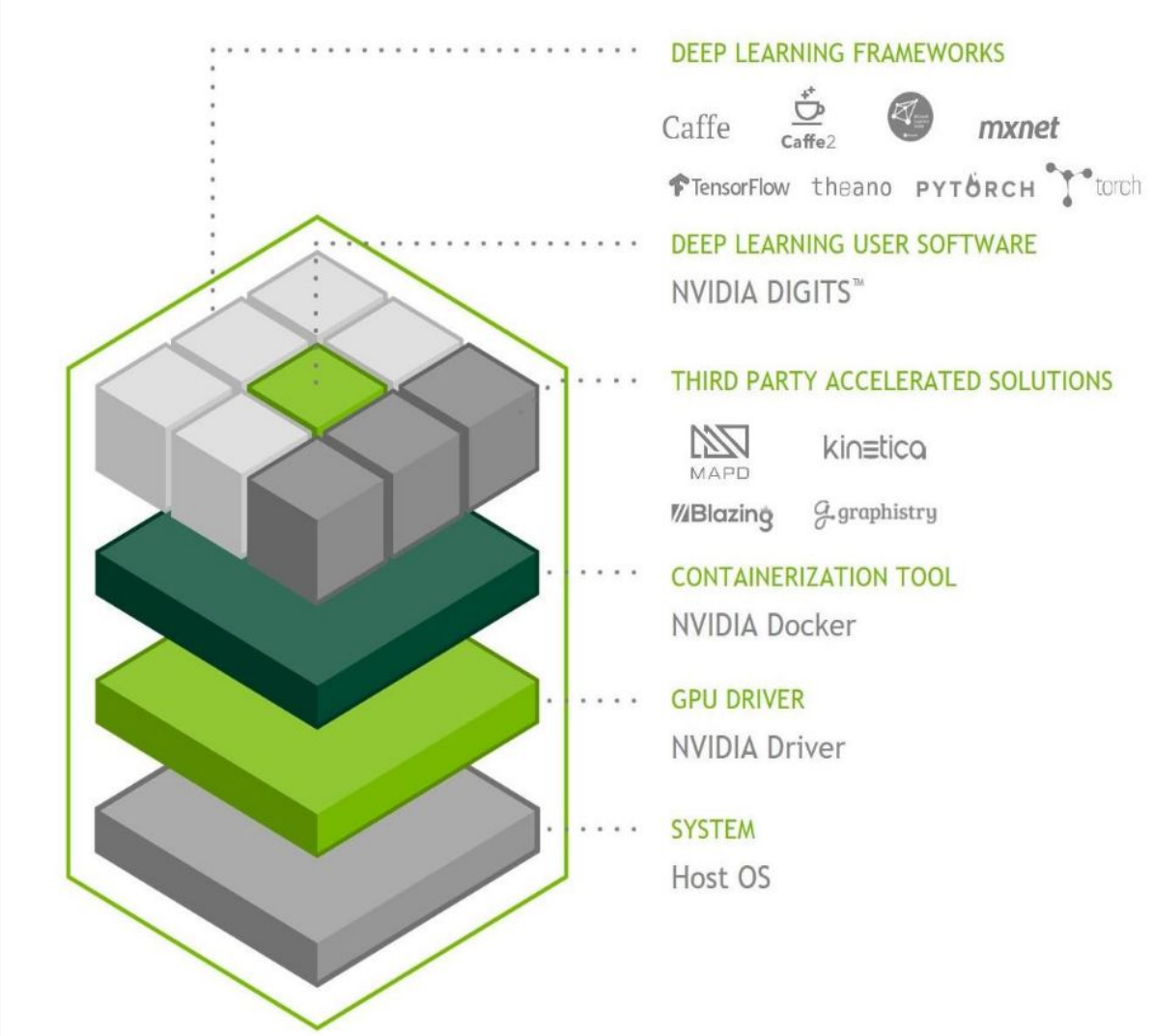
THE FASTEST PATH TO DEEP LEARNING

Building a platform for deep learning goes well beyond selecting a server and GPUs. A commitment to implementing AI in your business involves carefully selecting and integrating complex software with hardware. NVIDIA® DGX-1™ fast-tracks your initiative with a solution that works right out of the box, so you can gain insights in hours instead of weeks or months.

DGX Data Center



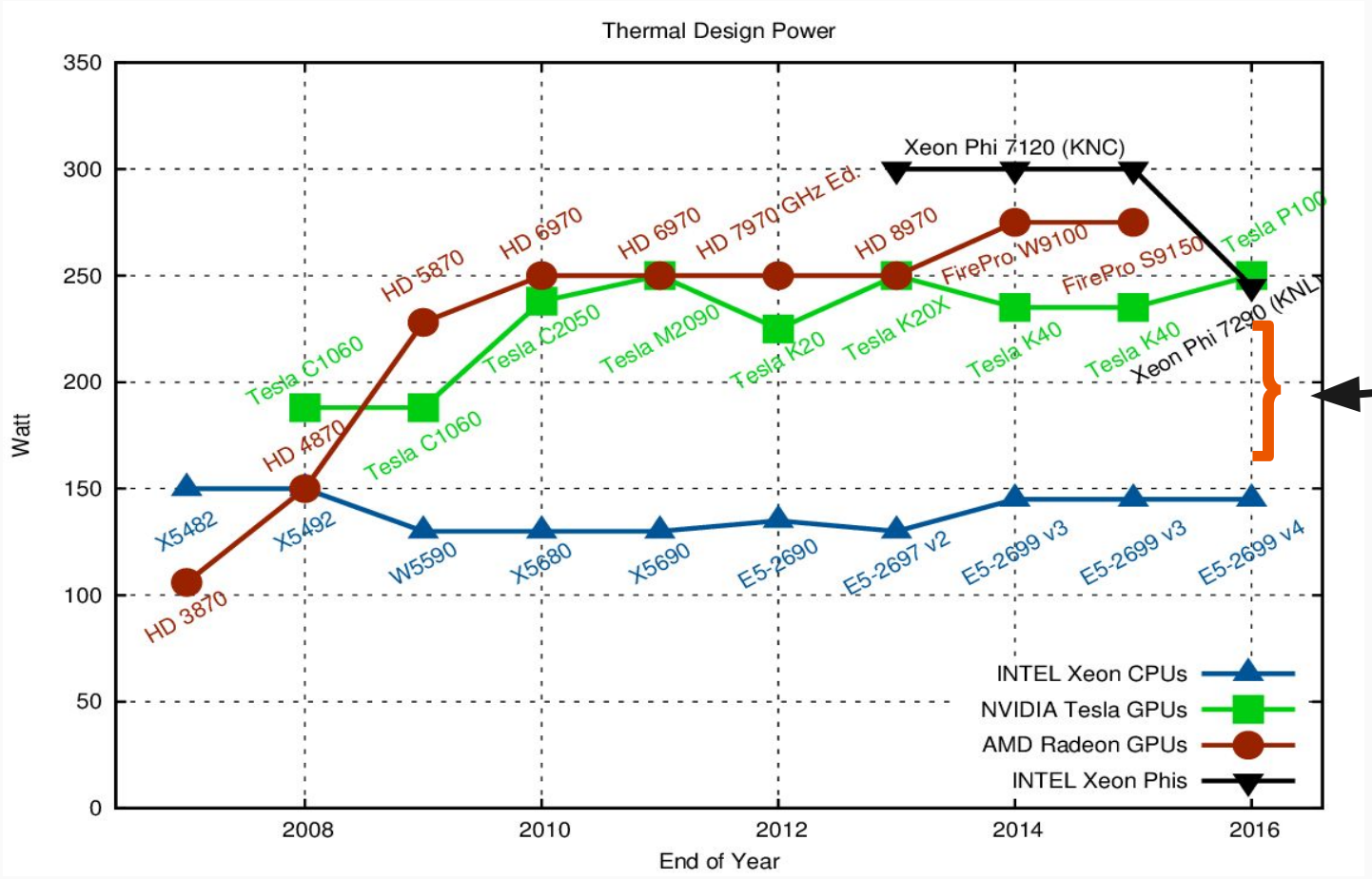
GPU Support in Cloud Computing Stack



GPUs in the Cloud

- Exponential demand for more compute power

GPUs are power hungry

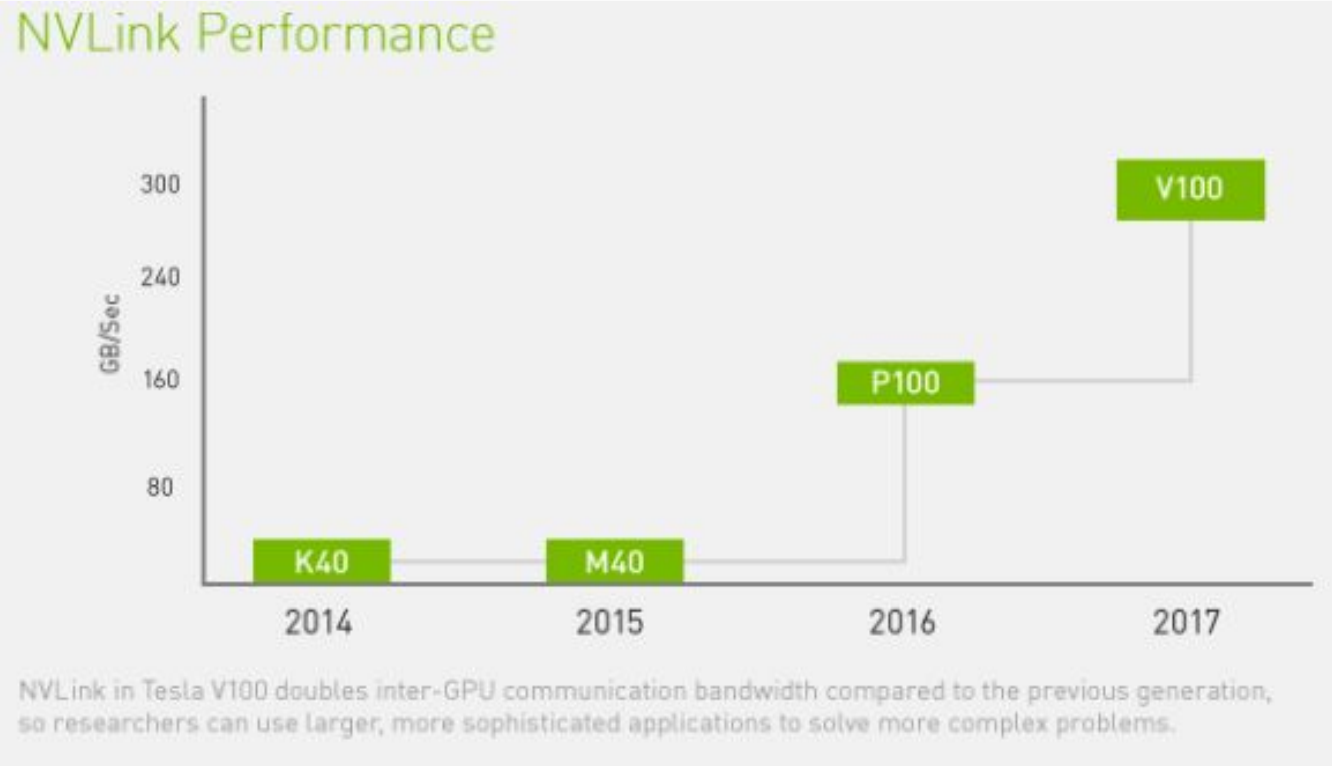


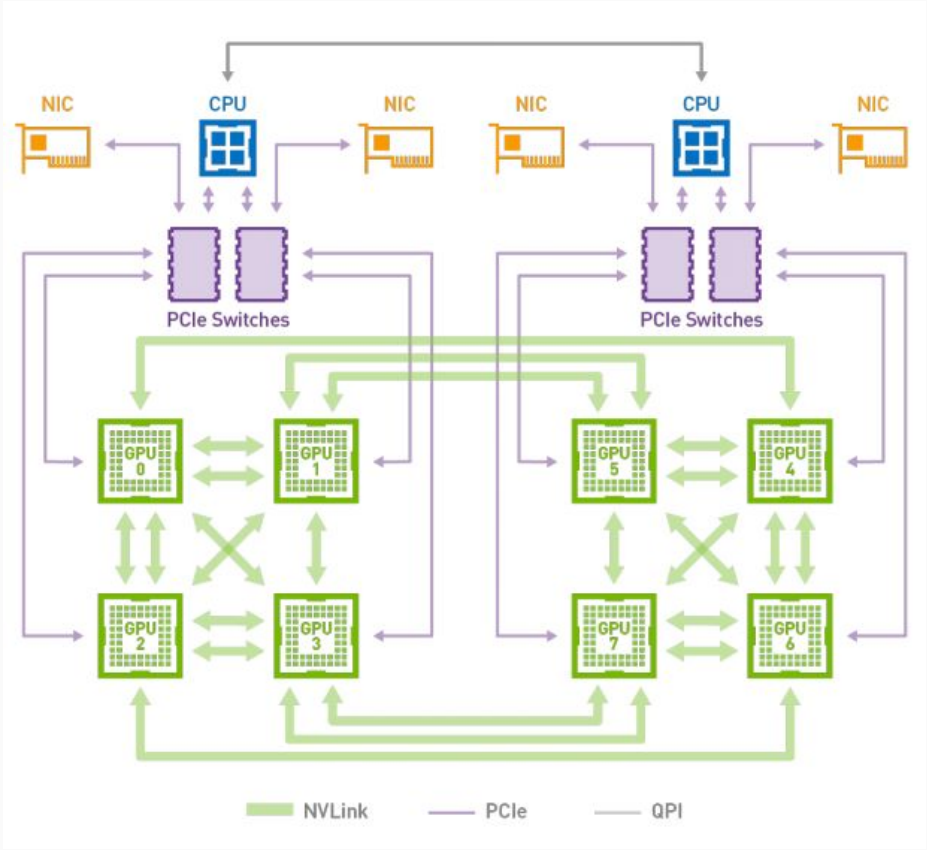
~100 Watt Difference Compared to CPUs

https://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/

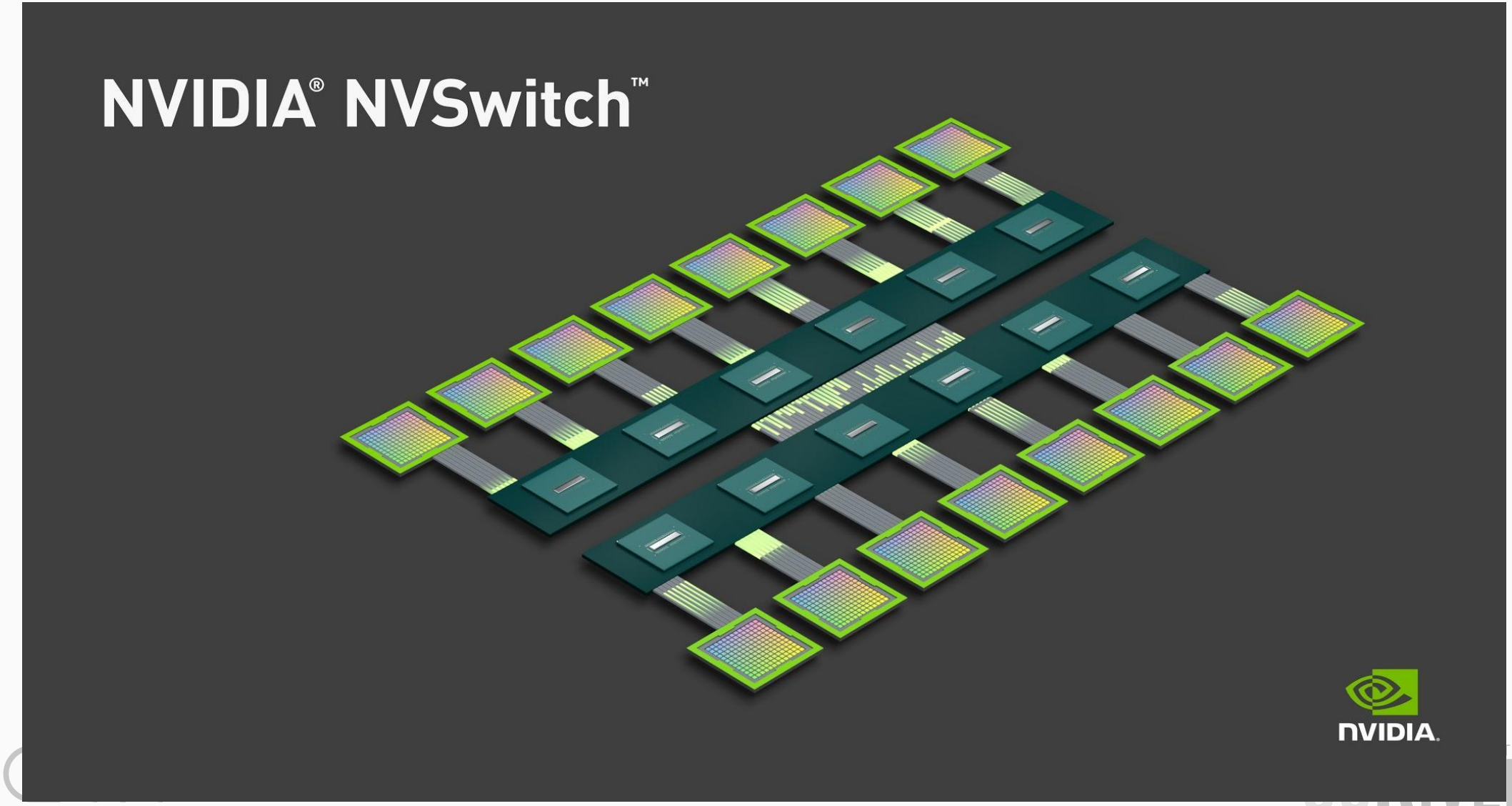
How can we save GPU power in data center environments?

GPU inter-connection is getting complex





GPU inter-connection is getting complex



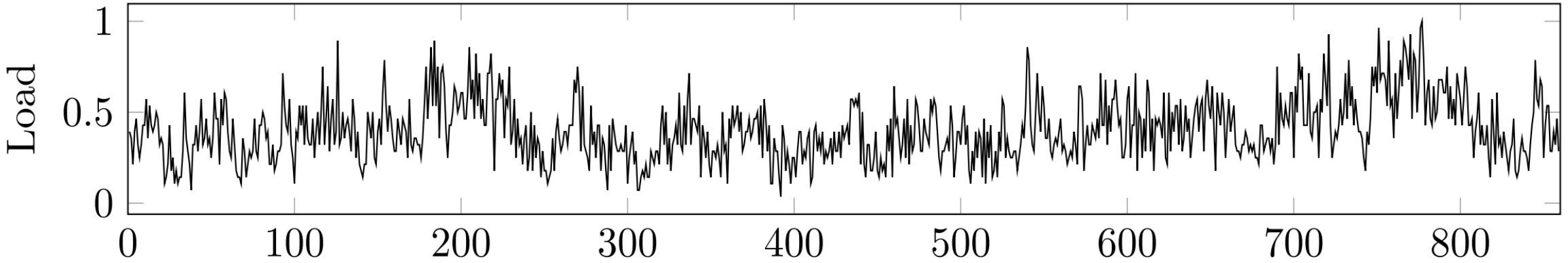
How can we make efficient use of GPU inter-connects?

Power challenges

Varying data center utilization

- Data Center load fluctuates over time
  - Leads to underutilization of hardware resources

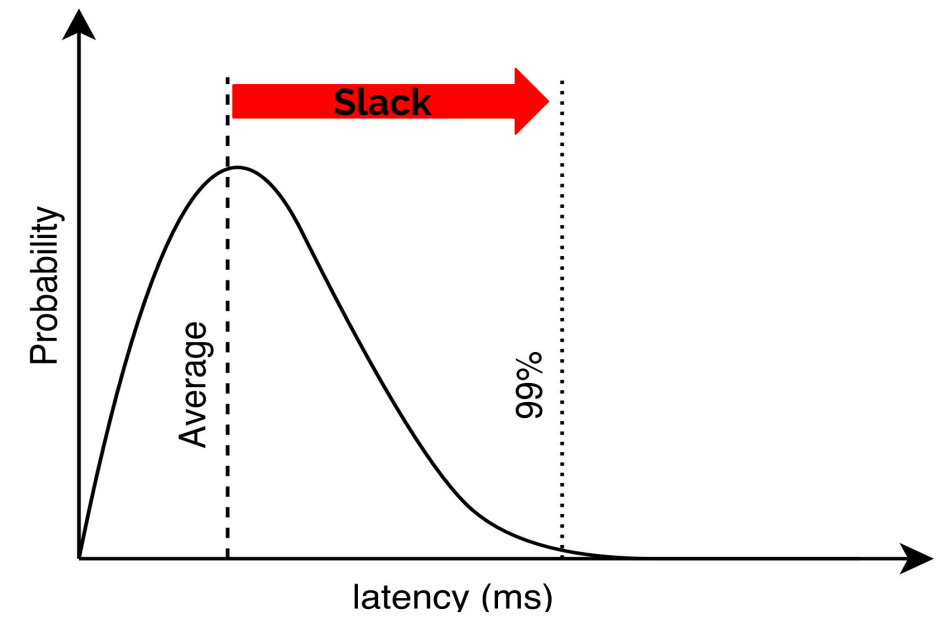
Google's Data Center Trace https://github.com/google/cluster-data/blob/master/ClusterData2011_2.md



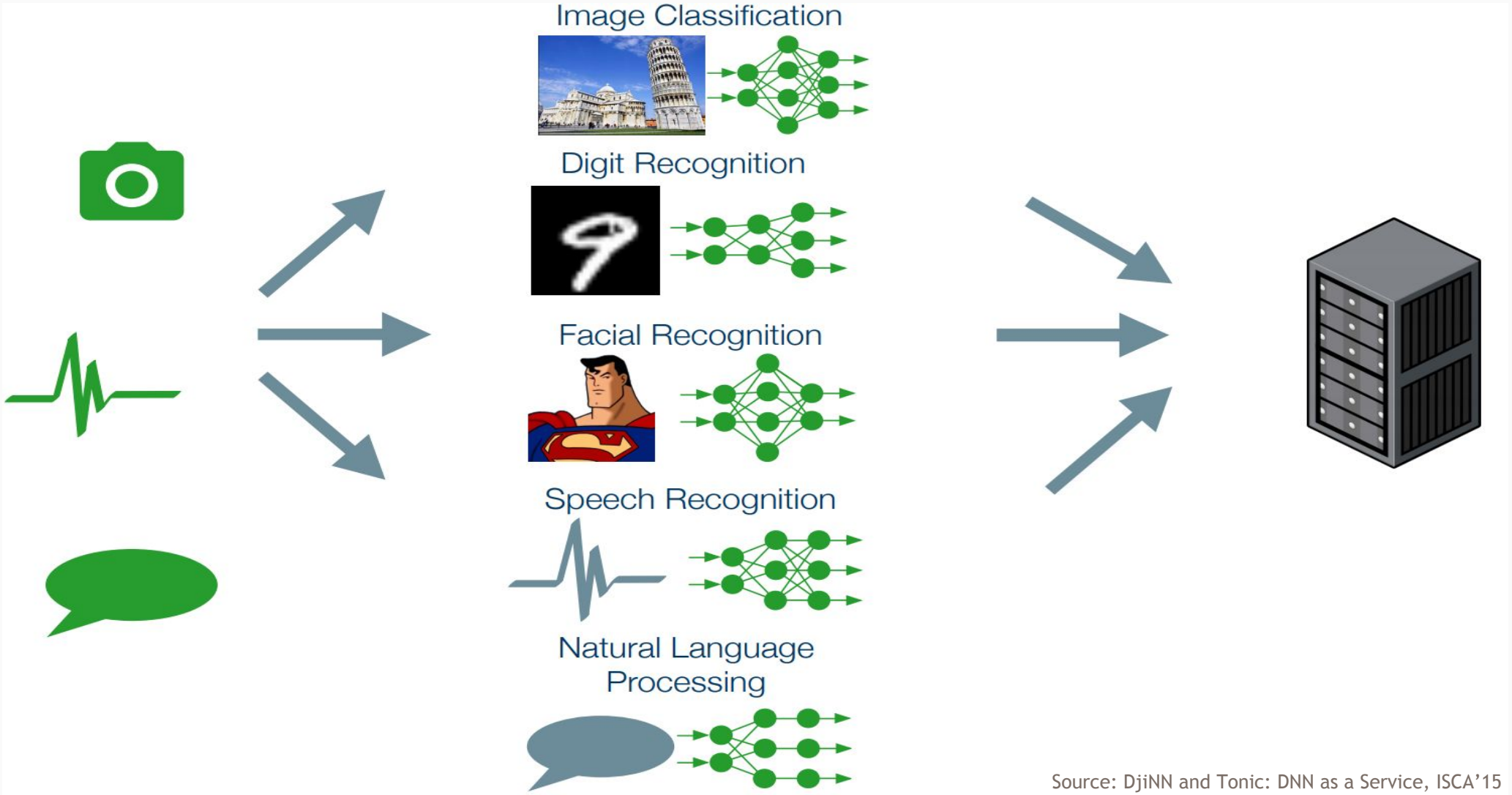
- Common solution:
  Dynamically scale clock Frequency with current load

Tradeoff latency for power

- Slow down request processing
- Requests must meet latency constraints
  - Must be serviced under 99 percentile
  - Costs money/time/energy if over

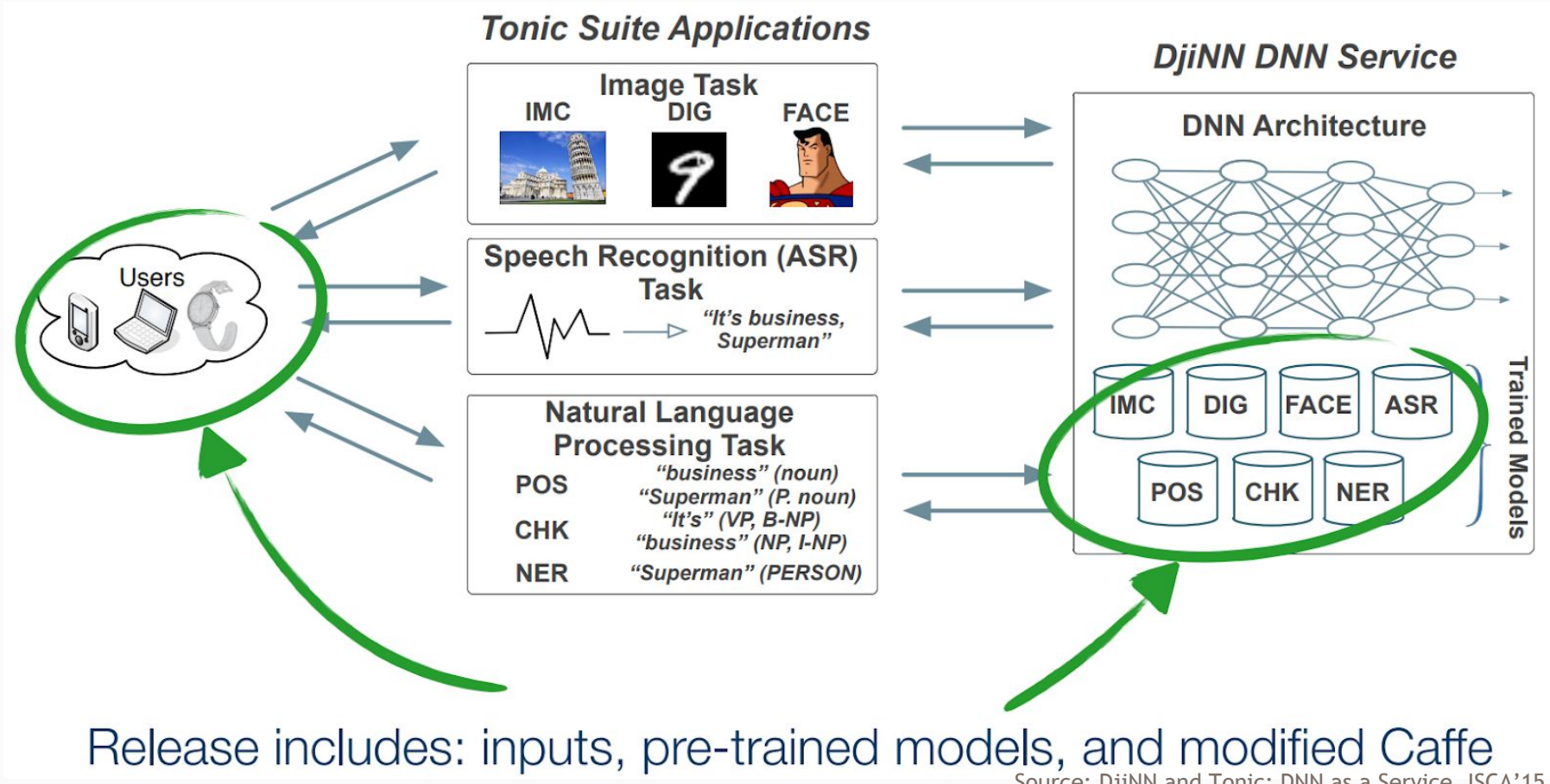


Target DNN as a Service



Source: DjiNN and Tonic: DNN as a Service, ISCA'15

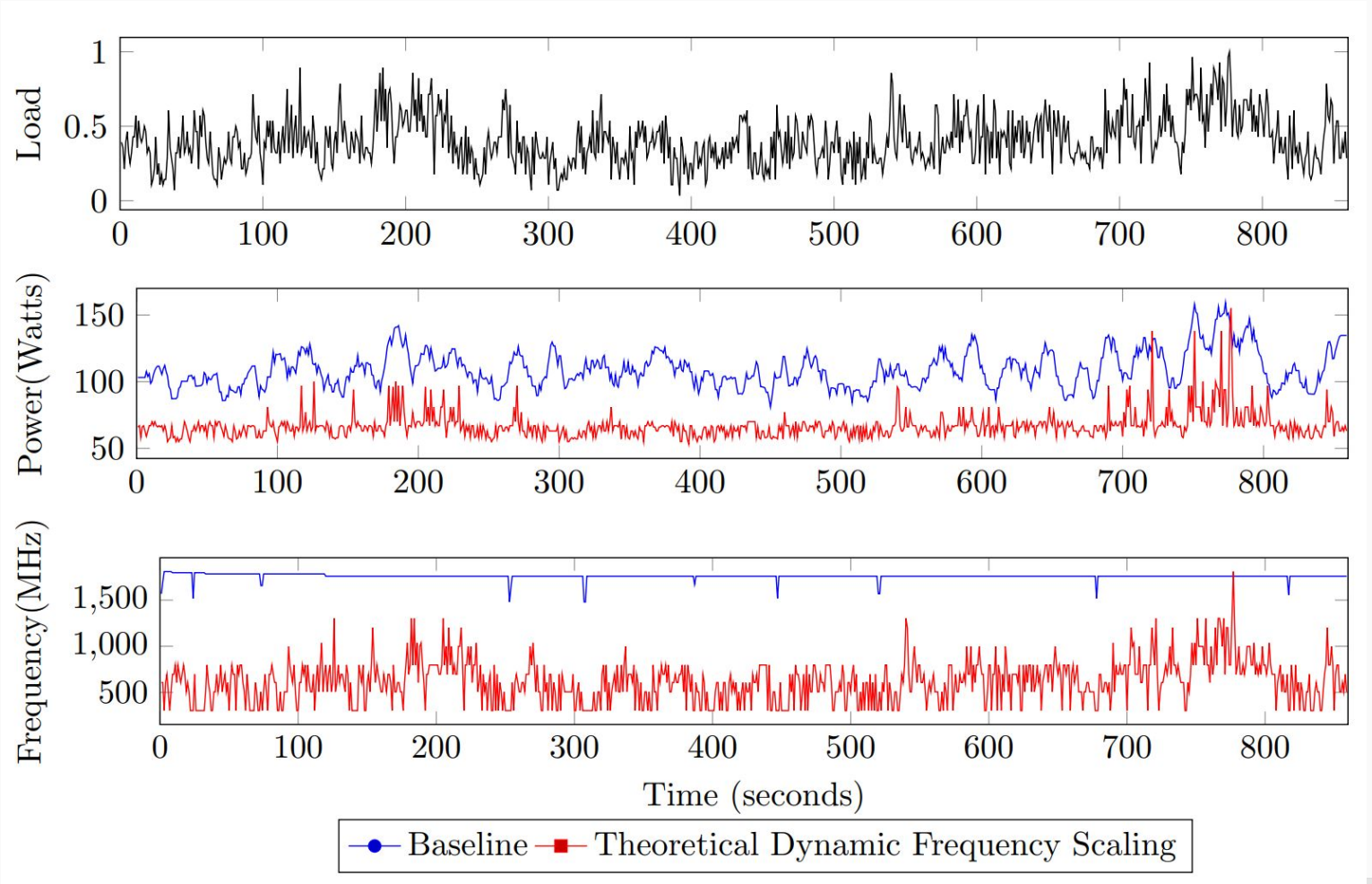
Djinn and Tonic



Source: DjiNN and Tonic: DNN as a Service, ISCA'15

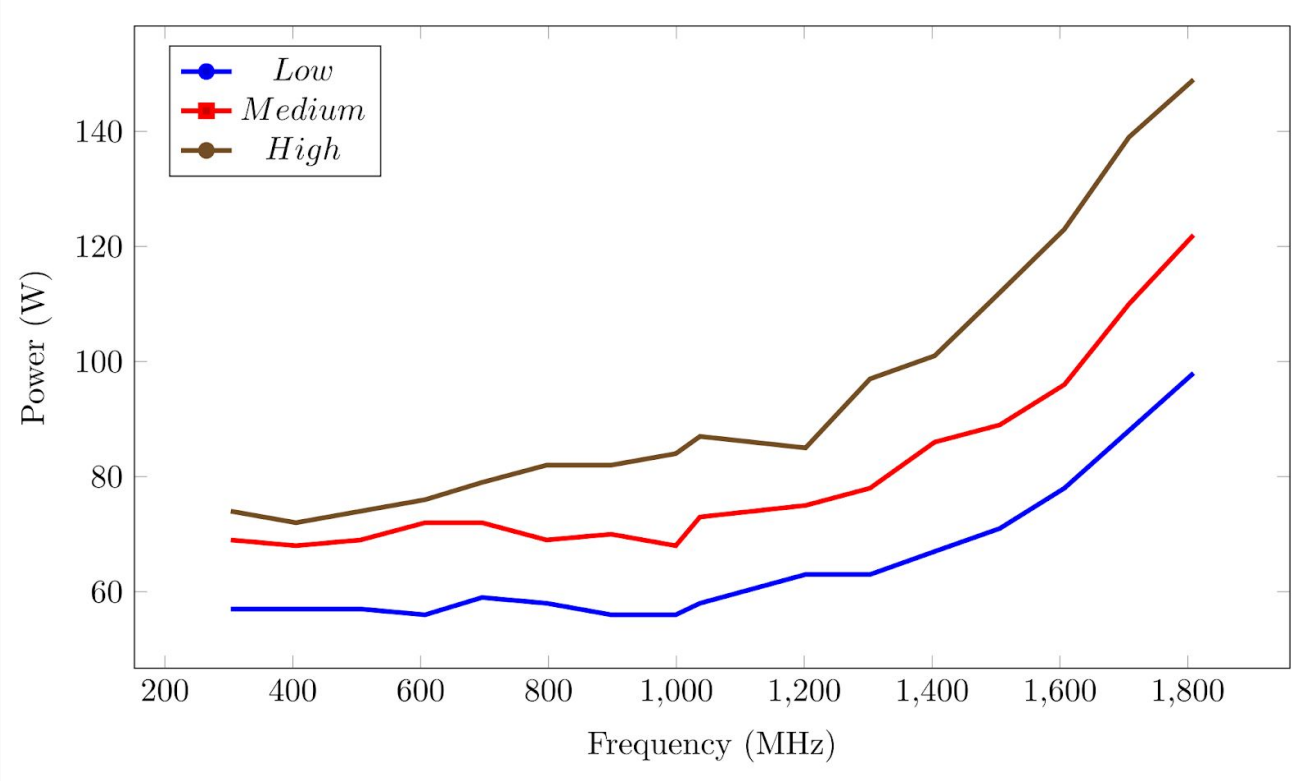
Running on NVIDIA Titan X

GPU frequency scaling exploits thermal headroom



Diminishing Returns from Frequency

- Power vs Frequency non-linear



Frequency Scaling Challenges

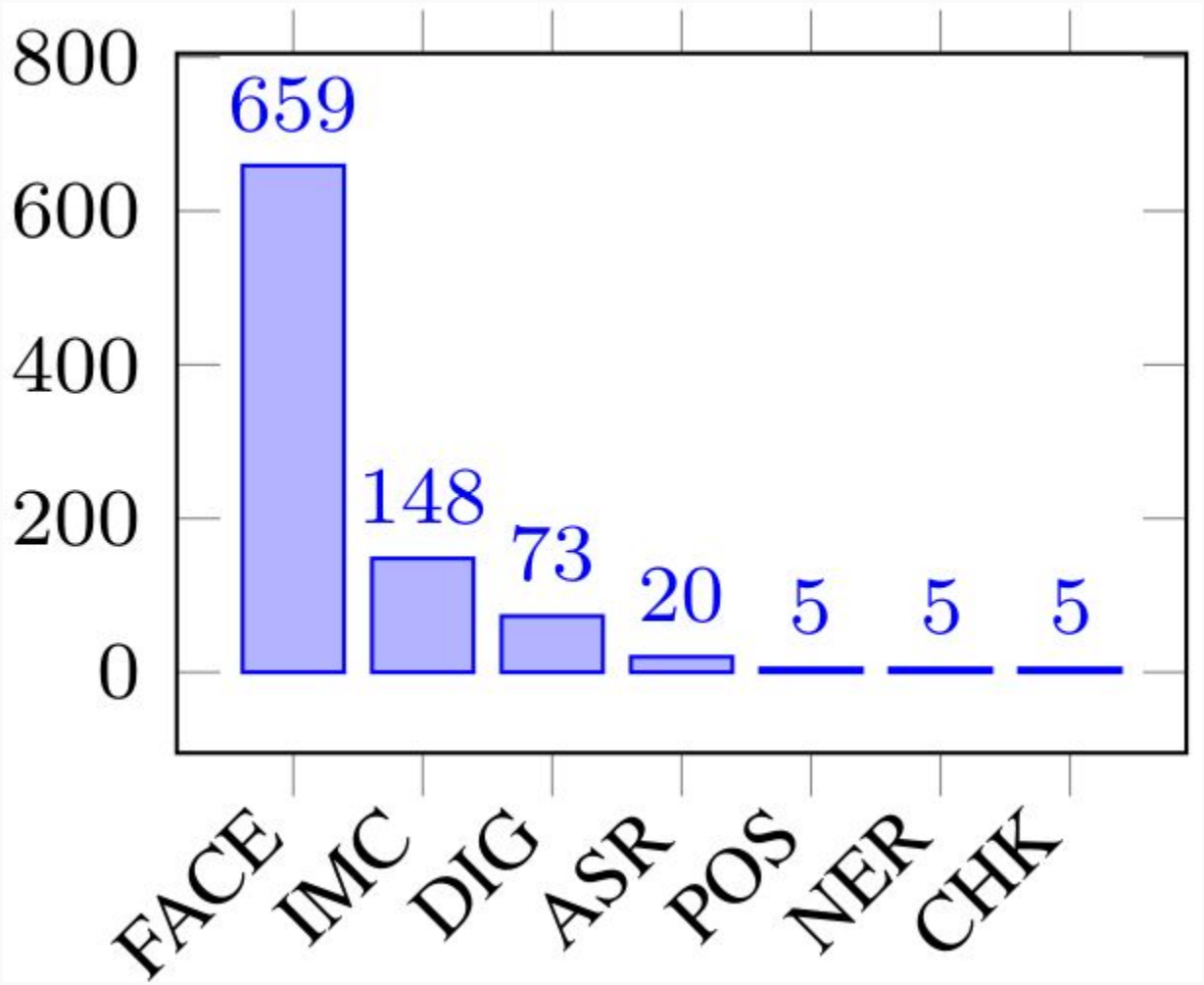
- Frequency scaling achieves limited power savings
- How to trade-off frequency for latency?

- In CPUs, frequency states are supplemented with deep sleep states … which do not exist in GPUs

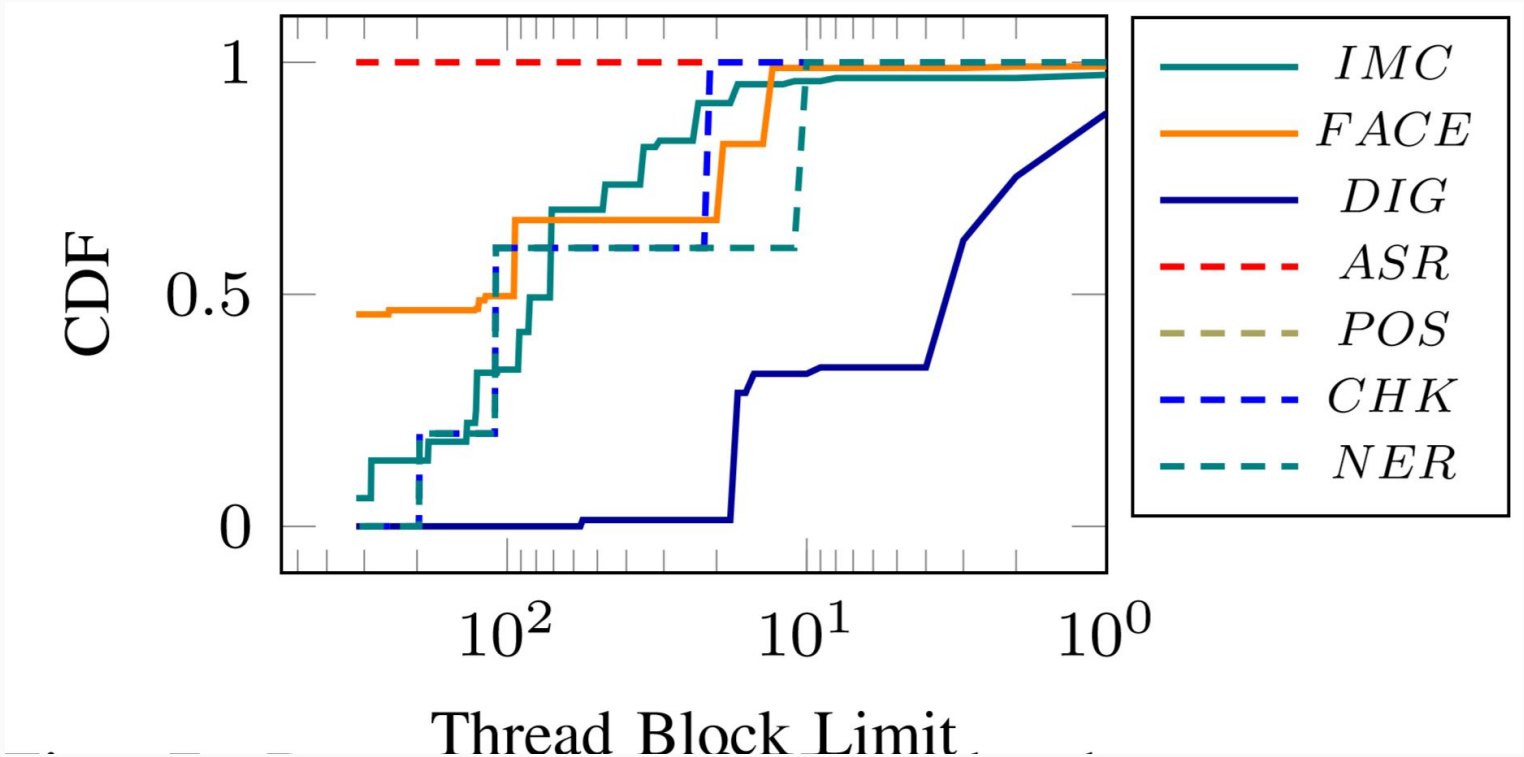
Scale parallelism w/ *Thread Block Scaling*

- Exploit application-level characteristics
- Limiting the amount of thread blocks that a single request uses
- Potentially reduce dynamic power by utilizing less hardware

DNN Inference calls multiple kernels



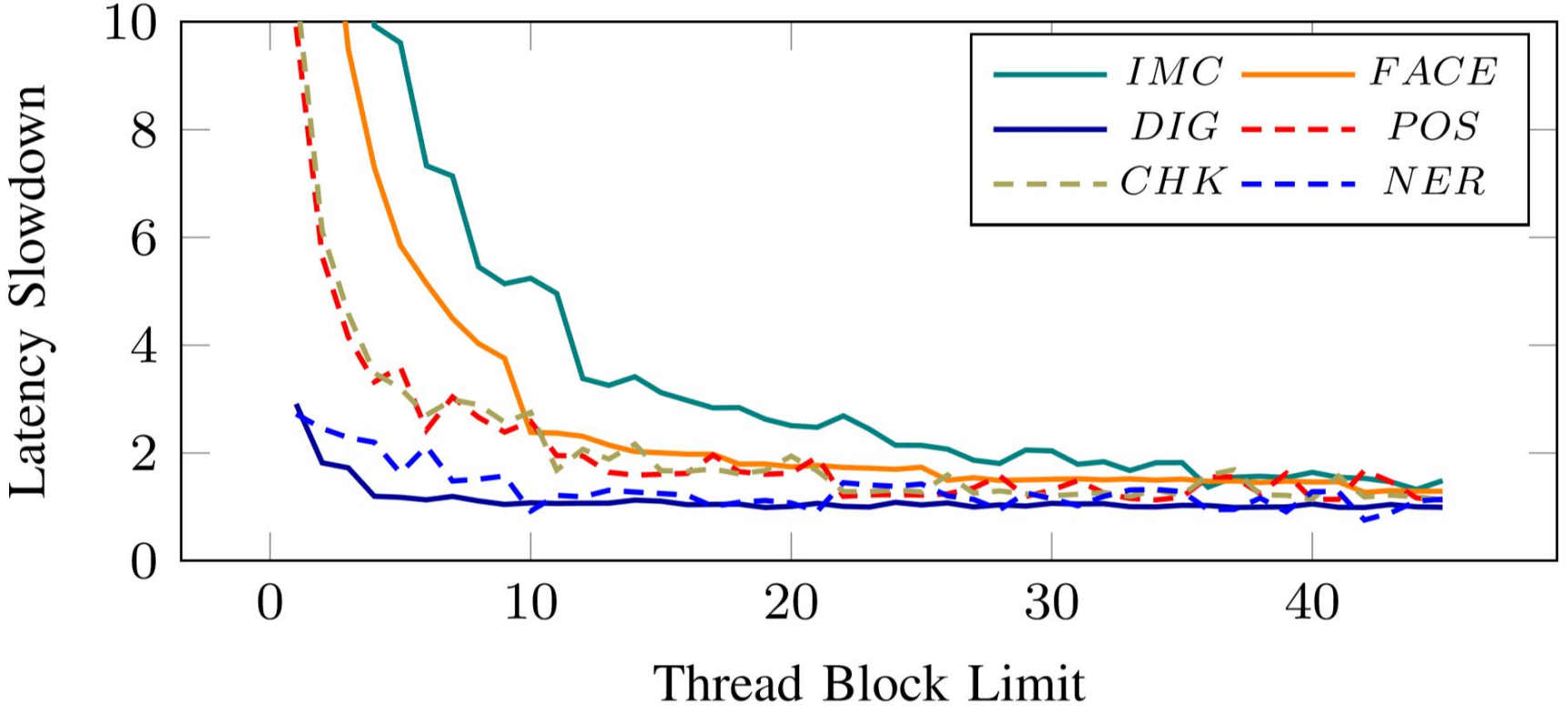
Kernels vary in Thread Block usage



- Most applications do not use all hardware resources, but are provisioned the entire GPU!

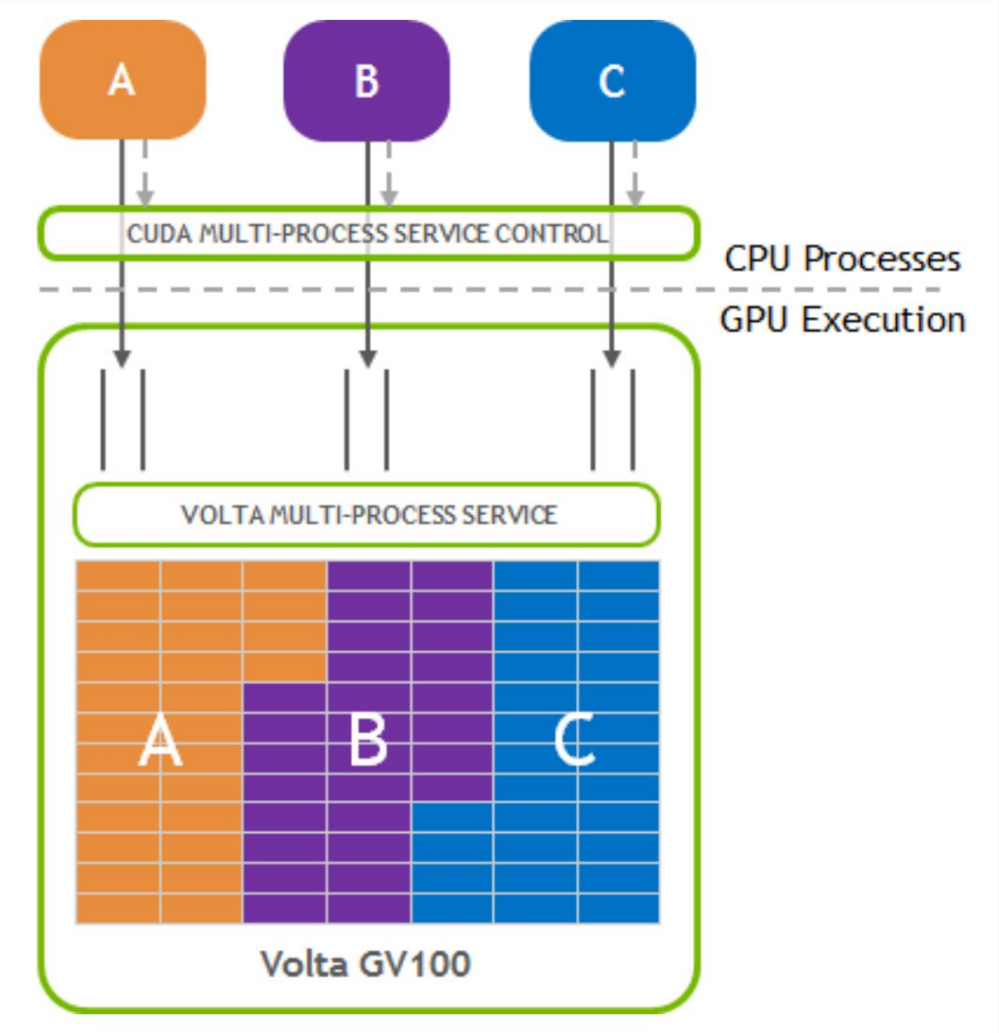
Squeeze kernels into less TBs

- Thread Blocks can be reduced without a major impact to execution time
- Latency becomes an issue at around 75% TB reduction



Enable Colocatating Multiple Requests

- Service multiple requests at the same time on a single GPU
- This allows the frequency to remain low while handling a higher load
- Increasing overall energy efficiency



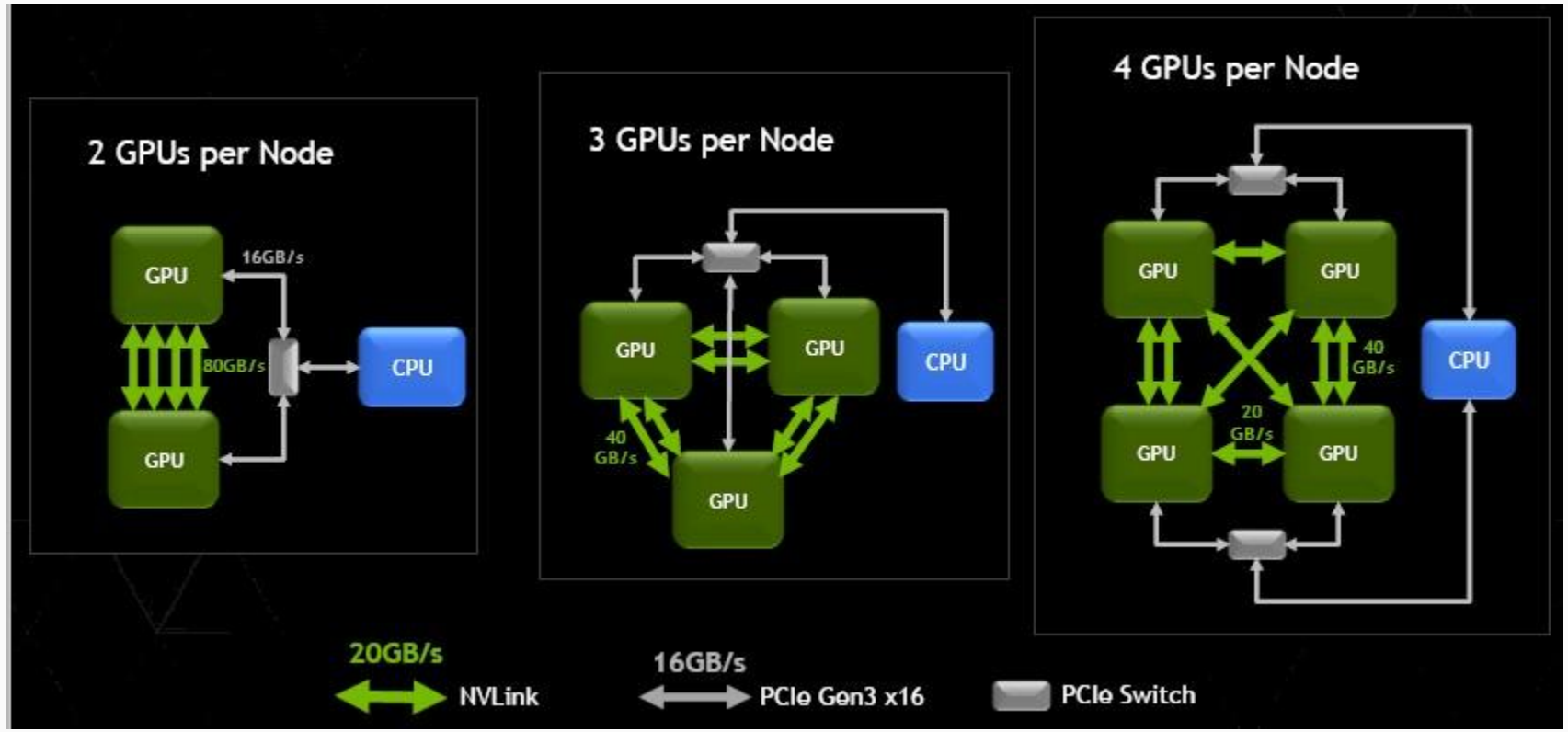
Thread Block Scaling Challenges

- Software vs Hardware implementation?
- How to coordinate thread block scaling with frequency scaling?
- Colocating multiple requests may lead to contention of hardware resources
  - How to allocate resources to kernels?

Communication-related performance challenges

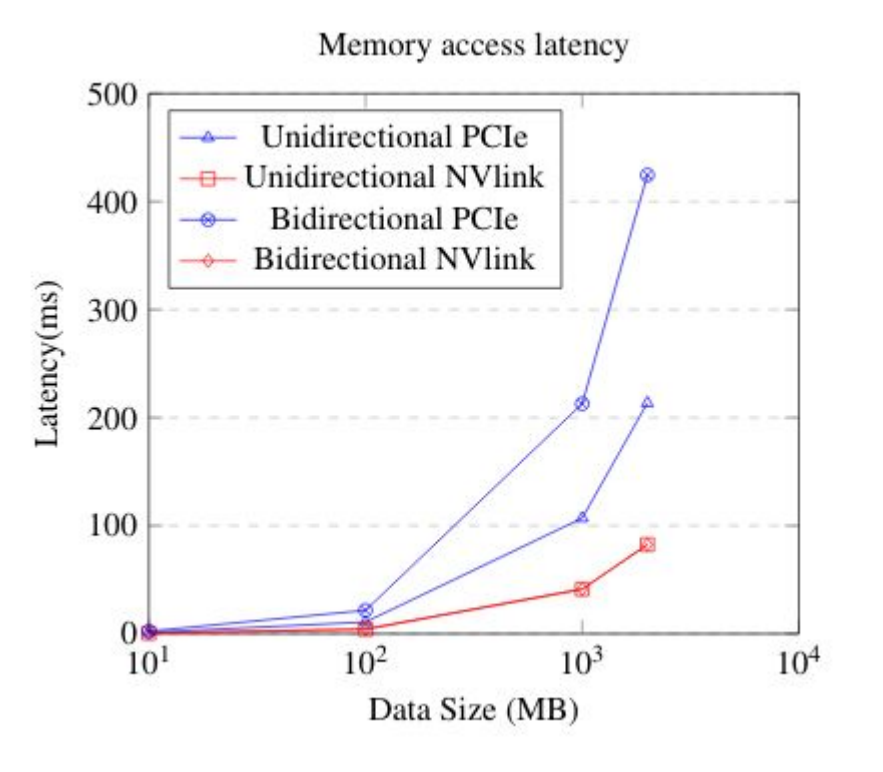
NVLink: Fast communication between multi-GPUs

>



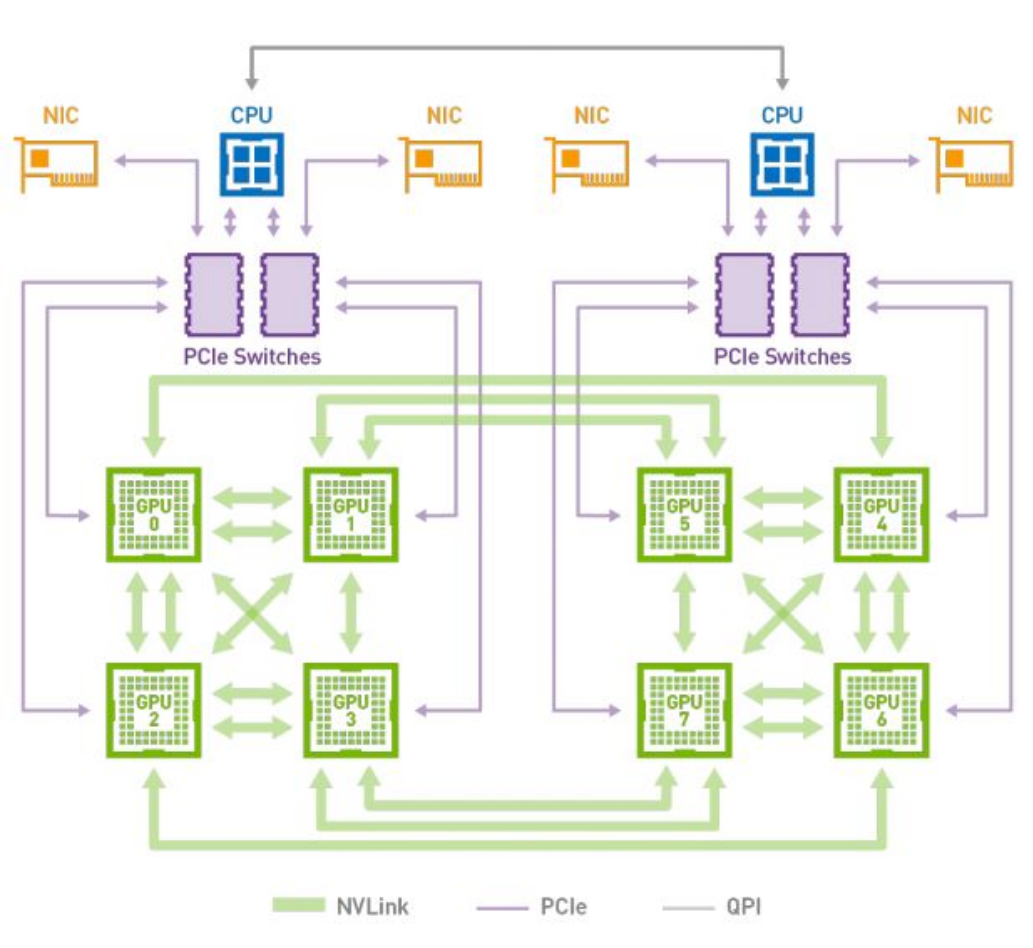
NVLink vs PCIe

- NVLink also provides significantly lower latency
  - Even with bidirectional traffic!



Challenges of complex GPU inter-connects

- Programming Multi-GPU applications is hard
- Not aware of inter-connect topology
- Poor placement of GPU kernel can lead to performance impact



Solutions

- Develop new paradigms and APIs to ease multi-GPU application development

Paradigm	Description
Pool	group of GPUs allocated to the requested container
Topology	topology of all the GPUs in the cluster
Route	A GPU chosen to assist inter-GPU communication
Chunk	Memory available in the intermediate router-GPU
Kernel Migration	Relocate CUDA kernels after GPU reallocation
Policies	algorithms and strategies used to assist allocation, reallocation and routing.

Table 1: Paradigms used in MGML

API	Description
cudaCreatePool	Create pool and mount it
cudaUpdatePool	Add or delete GPUs in pool
cudaDestroyPool	Relinquish GPU pool from user
cudaRefreshTopology	Reallocate GPU pool for performance or energy savings
cudaMemcpyPool	Copy data from one GPU to another within the pool using NVlink routes

Table 2: Core set of APIs in MGML

Solutions

- GPU Kernel scheduling and mapping algorithms
  - Guided by topology information from programmer APIs
- Utilize intermediate GPUs as NVLink routers to allow communication between non-direct connect GPUs
  - Avoids PCIe

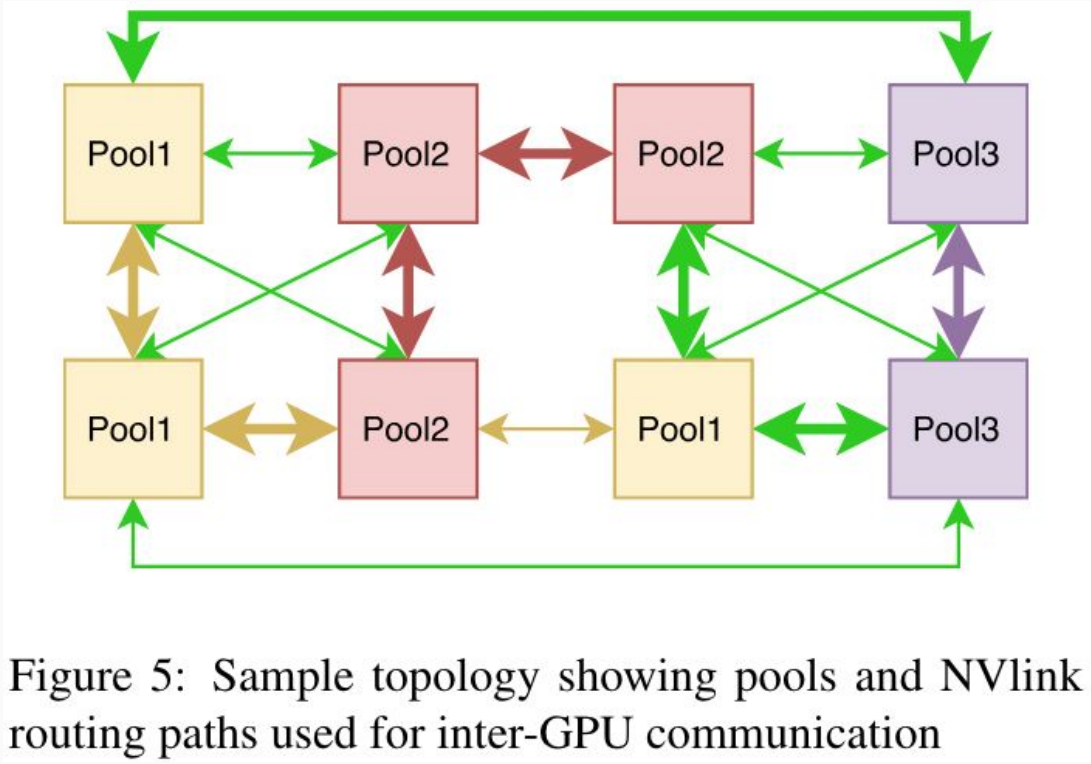


Figure 5: Sample topology showing pools and NVlink routing paths used for inter-GPU communication

NVLink routing preliminary results

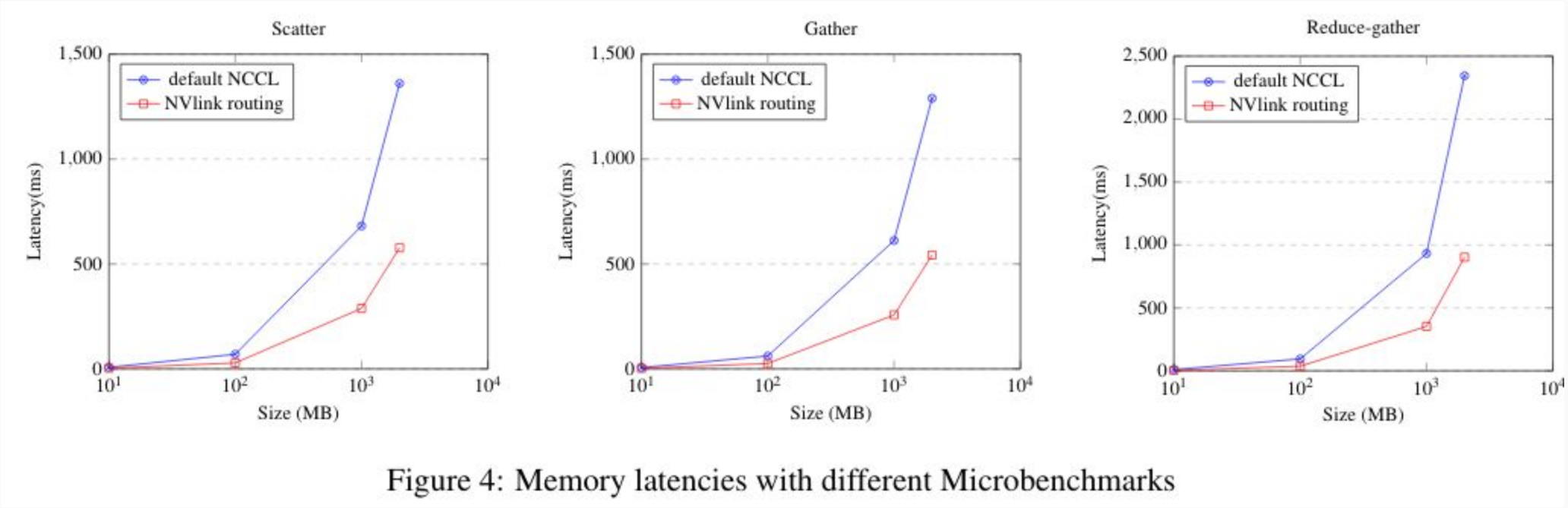


Figure 4: Memory latencies with different Microbenchmarks

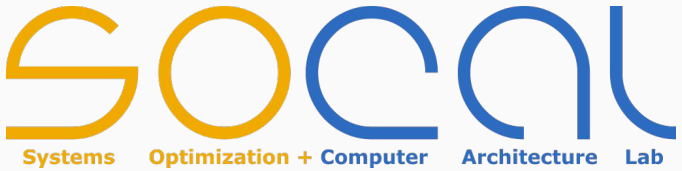
Conclusion

- Modern GPU-based data centers face many power and performance related challenges
- GPUs have limited power savings features (frequency scaling)
  - Parallelism-scaling and co-location offers potential to improve energy efficiency
- Multi-GPU programming and management is made difficult due to GPUs increasingly complex inter-connection
  - Requires new paradigms, programmer-support, mapping/scheduling support, and runtime support.

Thank you! Questions?

Performance and Power Challenges in Data Center GPUs

Daniel Wong
dwong@ece.ucr.edu
University of California, Riverside
Department of Electrical and Computer Engineering



GPU Software View

- Each block contains a grid of threads
- Blocks and threads can be logically grouped in 3 dimensions