Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

spec

Measuring the Energy Efficiency of Transactional Loads on GPGPU

Jóakim von Kistowski[1], Johann Pais[2], Tobias Wahl[1], Klaus-Dieter Lange[3], Hansfried Block[4], John Beckett[5], Samuel Kounev[1]

[1]University of Würzburg, [2]AMD, [3]HPE, [4]SPEC, [5]Dell

April 10, 2018

Context

Server energy efficiency and power measurement

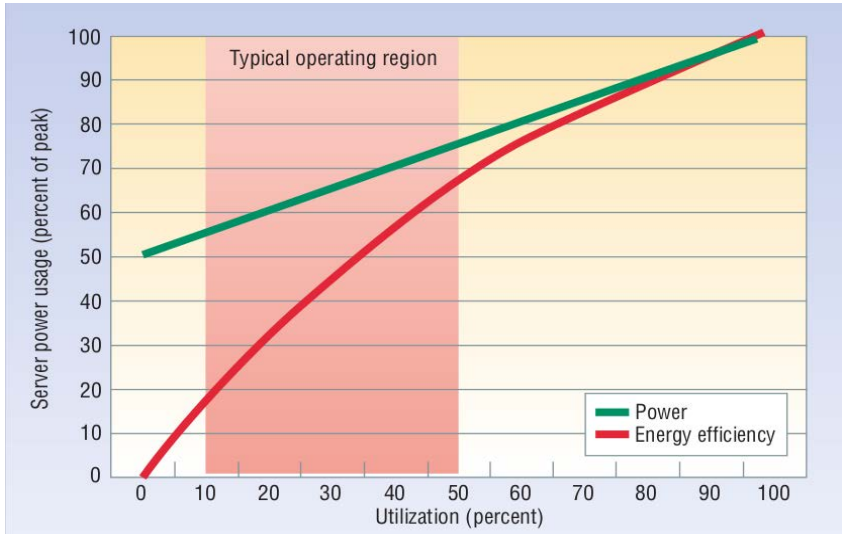
➔ **First step to improvement**

Focus in this talk:
Compute Servers with GPGPU accelerators

Energy Efficiency of Servers

Server efficiency depends on

- Application
- Software stack
- Hardware
- Load level

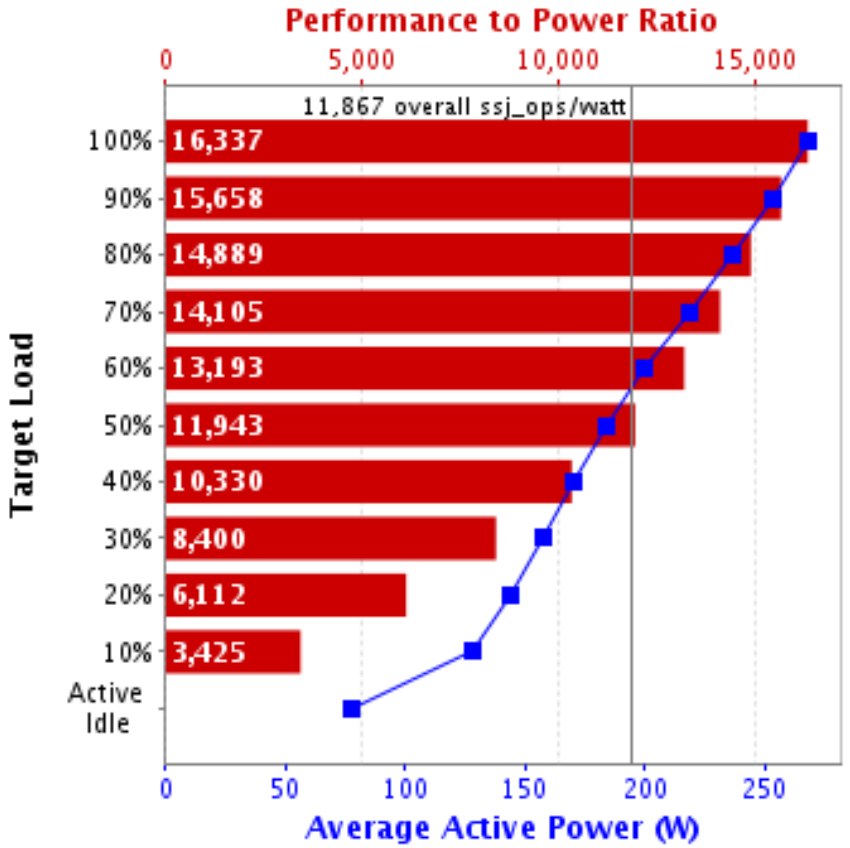


Energy Efficiency and Power Consumption of Servers [1]

Typical server utilization between 10% and 50%

Benchmarking of Energy Efficiency: Driving Development

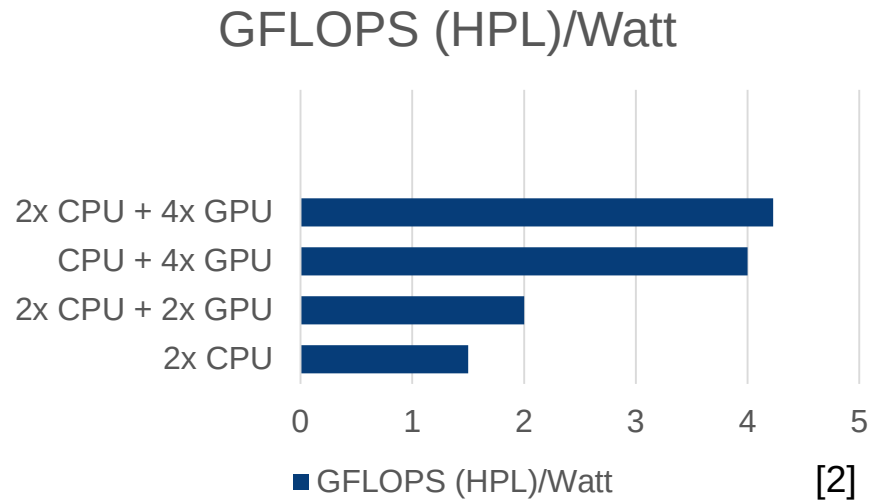
- SPECpower_ssj2008
  - Novel idea: Benchmark efficiency at multiple load levels
- Server efficiency development 2007 – 2019
  - Eff. increase at 100% load: 109.4x
  - Eff. increase at 10% load: 195.3x
- **Goal Today**: Enable measurements and improvements for GPGPUs with different load levels



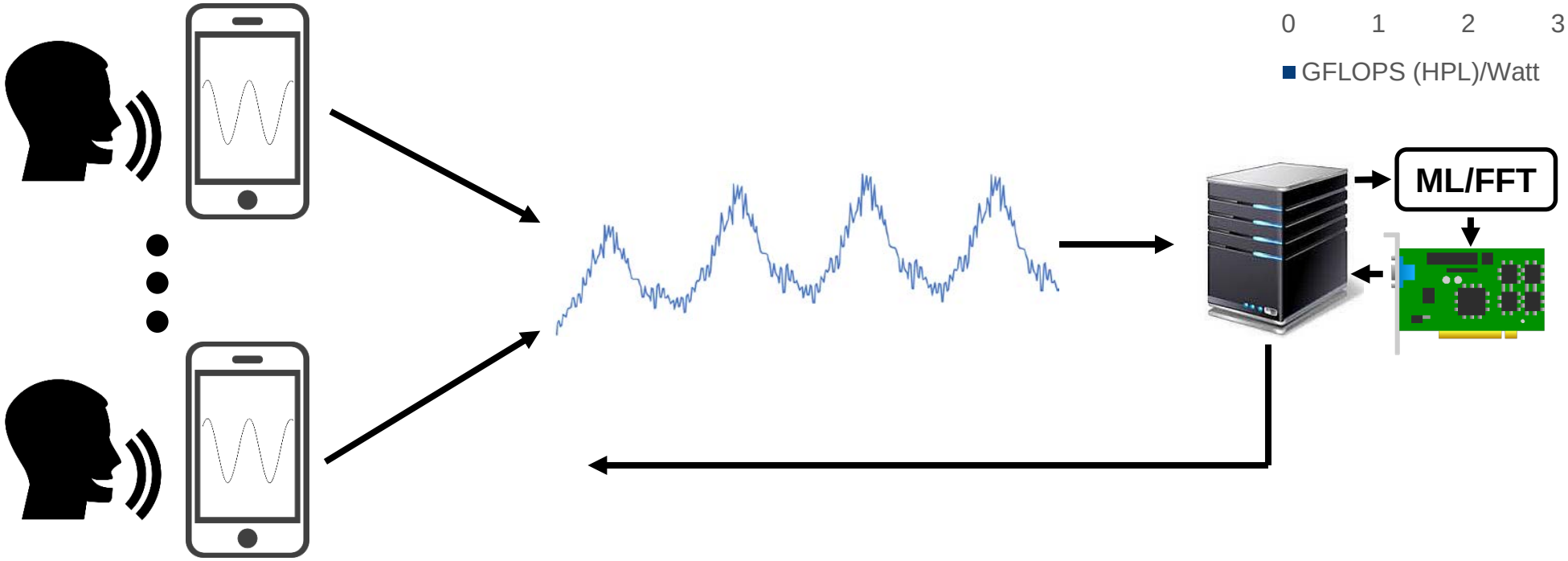
Energy Efficiency of GPGPU

GPGPU efficiency established in HPC applications

- Vision: GPGPUs in **transactional** scenarios
- Example scenario: speech recognition

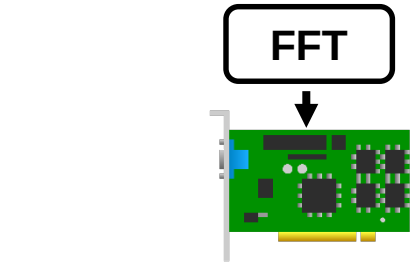


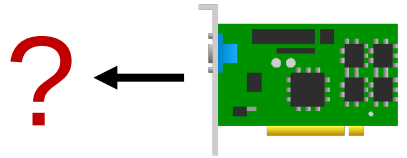
[2]



Transactional GPGPU Loads: Challenges

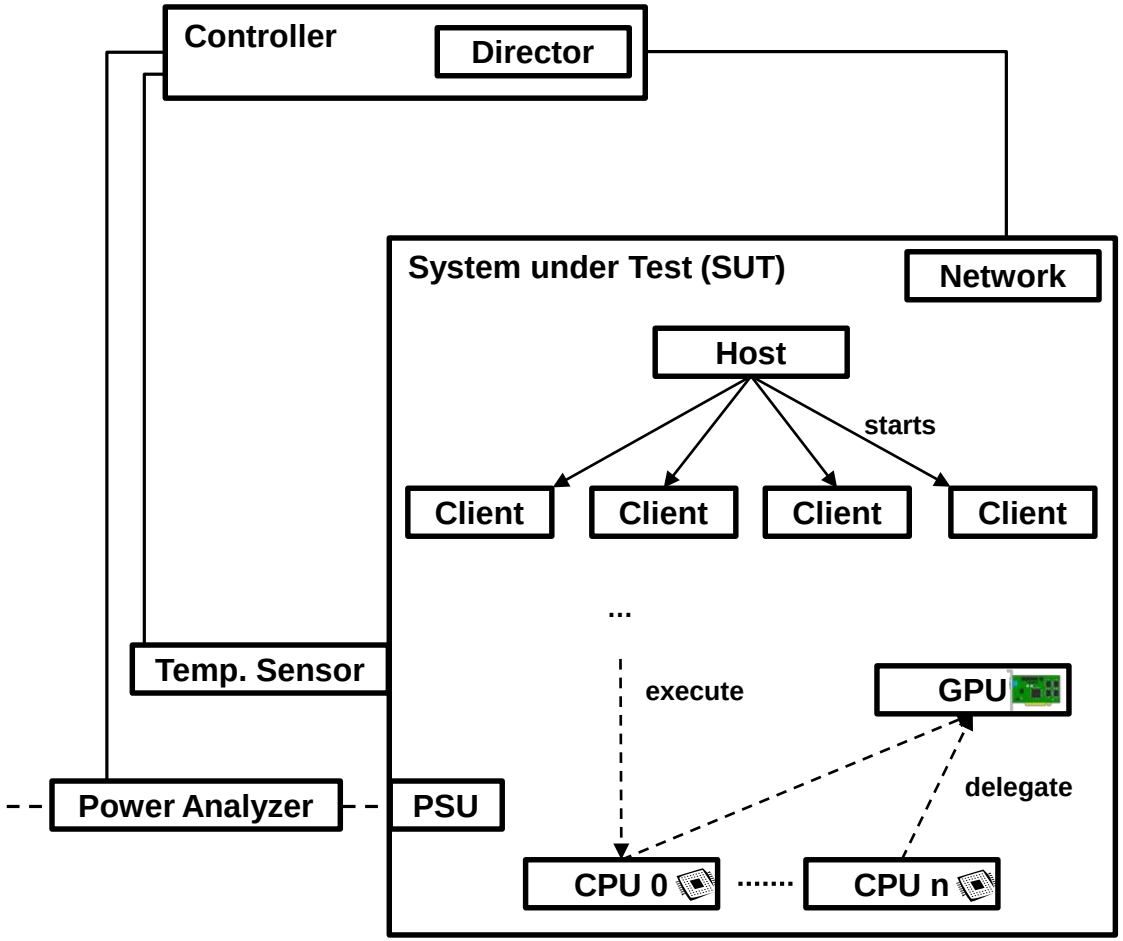
- Find and define transactional scenarios
- Dispatch transactional loads on GPU
  - Other ways of load scaling?
- Verify GPU result
- Enable and compare vendor-specific technologies





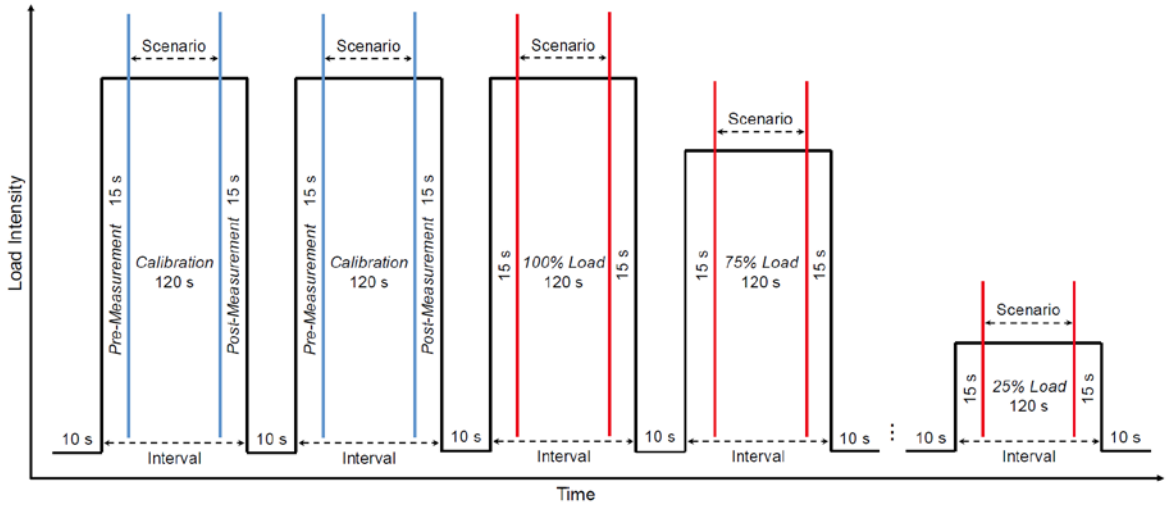
Efficiency Measurement Setup

- System Under Test (SUT) is
  - A single, **physical** AC power server
  - Connected to director machine
- "Host" software
  - Reports performance
  - Launches "Clients" (OS processes)
- "Clients"
  - Execute CPU transactions
  - Delegate work to GPU
- External measurement:
  - AC power meter
  - Temperature sensor



Workload Dispatch and Measurement

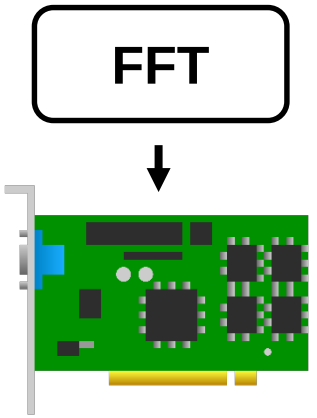
- **Calibration**
  - Run transactional workload at max transaction rate
  - Record transaction rate
- **Load level**: % of max throughput
  - Add waiting times for target load level
- **Measurement**
  - Throughput and power per second
  - Default duration: 120 s ➔ 120 samples



- Average power consumption $[W]$
- Average throughput $[s^{-1}]$

GPGPU Workload

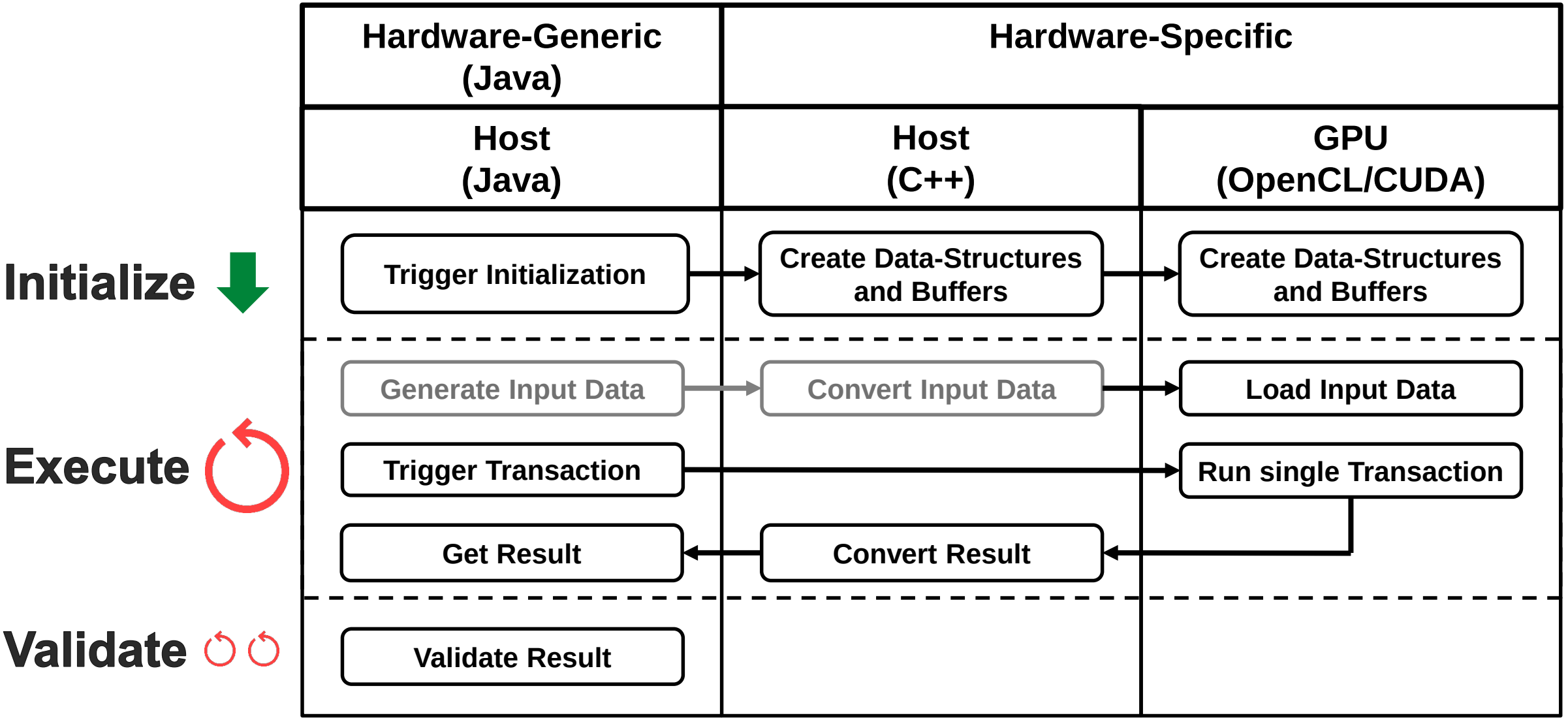
- Fast Fourier Transform (FFT)
  - Implemented as GPGPU-kernels
  - Two kernel implementations:
    - CUDA
    - OpenCL



- Pseudo-randomly generated input signal

- One kernel execution ➔ One transaction

Transactional GPGPU Workload Flow



Evaluation

- **Evaluation goals**:
  - Show reproducibility of results
  - Investigate GPGPU Power / Efficiency scaling
  - Demonstrate applicability for multiple GPU vendors
- Workload (FFT) implementations:
  - **CUDA**: cuFFT-based
  - **OpenCL**: based on [3]

Servers used

	CPU-only	NVIDIA GPU	AMD GPU
Model	Fujitsu TX1320 M1	Dell R730	Reference Platform
Sockets	1	2	2
CPU	Intel Xeon E3-1281 v3	Intel Xeon E5-2699 v4	AMD EPYC 7601
Cores	4	2 x 22	2 x 32
Memory	32 GB	128 GB	128 GB
GPU	None	NVIDIA Tesla K40m	AMD Radeon Instinct MI25
GPU Memory		12 GB GDDR5	16 GB HBM2
OS	RHEL 6.7	RHEL 7.5	Ubuntu 18.04

Reproducibility

- **Run-to-run variations**:
  - 20 experiment re-runs
  - Investigate per-load-level power and throughput reproducibility
  - Metrics:
    - Coefficient of variation (CV)
    - Relative min-max difference

GPU Run-to-Run CVs

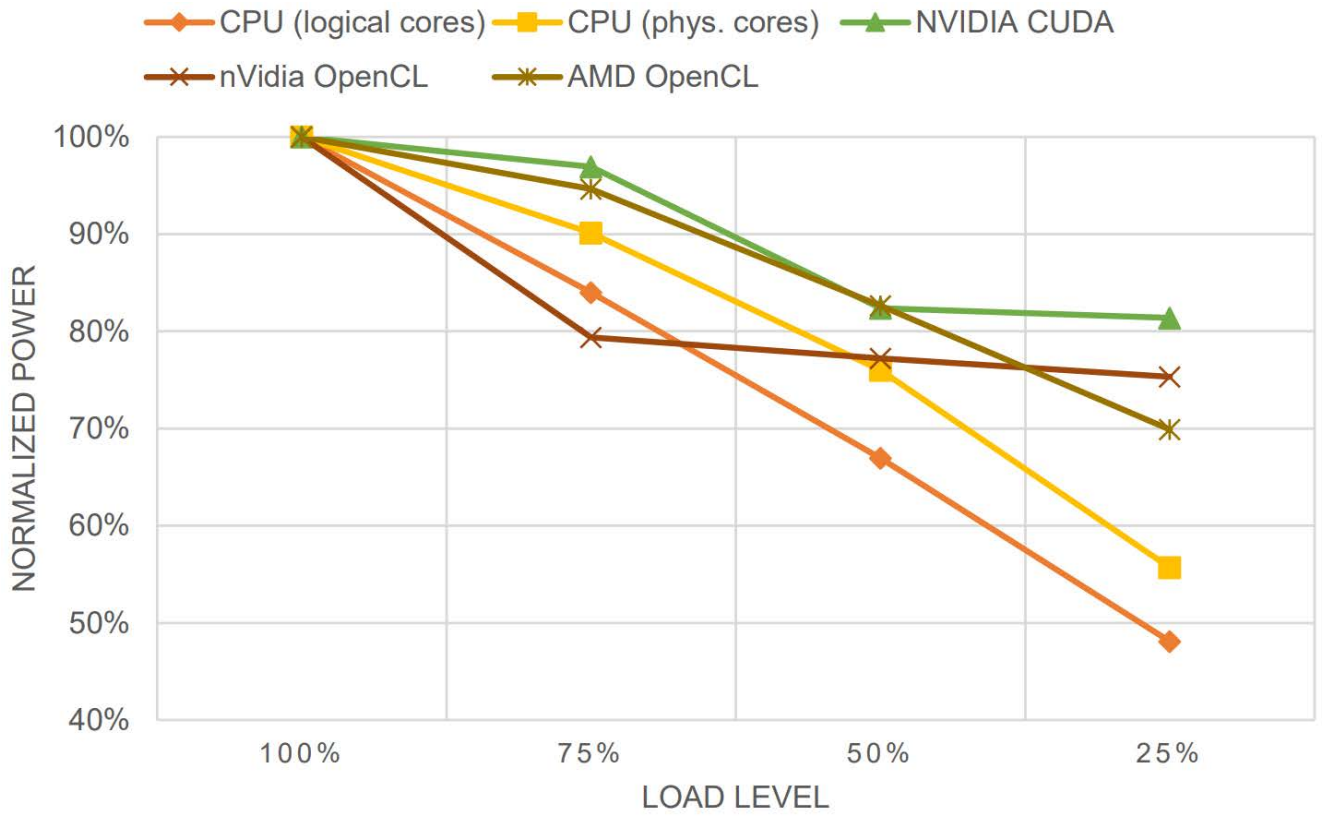
Server GPU	Load Level	Throughp. CV	Pwr. CV
NVIDIA Tesla K40m	25%	**0.2%**	0.9%
	50%	**0.2%**	1.0%
	75%	**0.2%**	1.1%
	100%	**0.2%**	**1.4%**
AMD Radeon Instinct MI25	25%	0.7%	0.7%
	50%	0.7%	0.7%
	75%	**3.0%**	**0.3%**
	100%	1.2%	0.6%

- What is a good CV?
  - Upper bound throughput CV: 5% [4]

All CVs < 5%

Power/Efficiency Scaling with Load Level

- **Throughput scaling**:
  - Load level definition uses throughput
  - Throughout scales linearly with load level
- **Power scaling**:
  - GPUs power scales
  - Little power scaling at low load
  - CPU scales better
- **Efficiency scaling**:
  - Throughput linear
    ➔ Inverse to power



Room for GPU power scaling improvements!

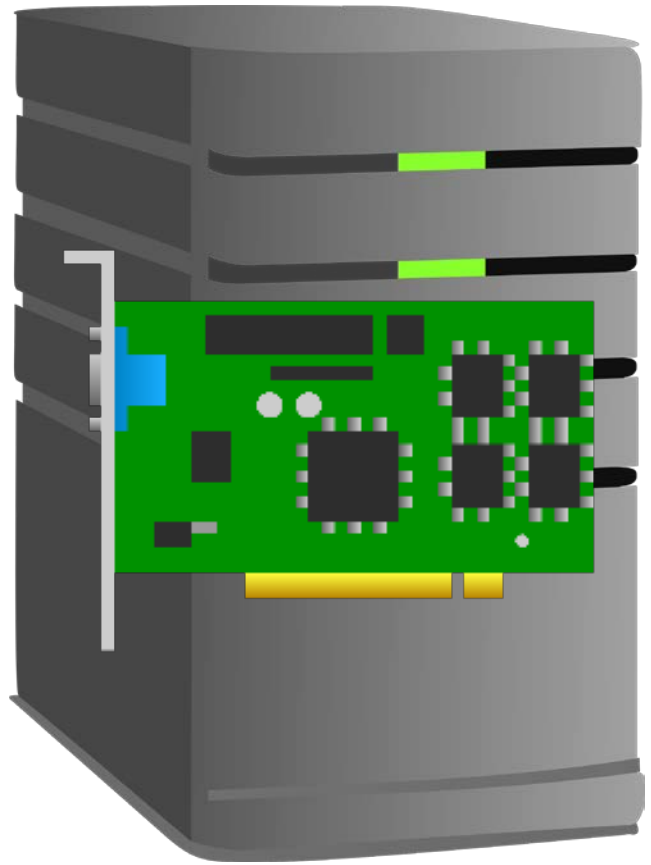
Power/Efficiency Scaling with Dataset Size

- **Challenge recap**:
  - Other ways of load scaling?
- **Dataset scaling**:
  - Vary size of FFT per-transaction dataset (size in packages)
- **Performance scaling**:
  - Performance decreases with size (both OpenCL and CUDA)
- **Power scaling**:
  - Varies with GPU model
  - Decreases or remains constant

Transactional load scaling is preferable

Summary

- **Goal**: Measure energy efficiency of transactional loads on GPGPU servers
- **Challenges**:
  - How to define and run transactional workloads?
  - How to validate transactional results?
  - How to deal with vendor-specific technologies?
- **Evaluation**: Show reproducibility and relevance of methodology
  - Reproducibility: CVs < 5%
  - Relevance: Power scales with transaction rate



Thank You!

SPEC Power Benchmarks and Tools

http://spec.org/benchmarks.html#power

References

[1] L.A. Barroso and U. Holzle. 2007. “The Case for Energy-Proportional Computing”. Computer 40, 12 (Dec 2007), 33–37.

[2] Yunxiang Gao et. al.. 2015. “Performance and Power Analysis of High-Density Multi-GPGPU Architectures: A Preliminary Case Study”, HPCC-ICESS-CSS 2015.

[3] Eric Bainville. 2011. Bealto FFT. http://www.bealto.com/home.html. Last accessed: 10.2018.

[4] K.-D. Lange and Michael G. Tricker. 2011. “The Design and Development of the Server Efficiency Rating Tool (SERT)”. In Proceedings of the 2nd ACM/SPEC International Conference on Performance Engineering (ICPE ’11). ACM, New York, NY, USA, 145–150.