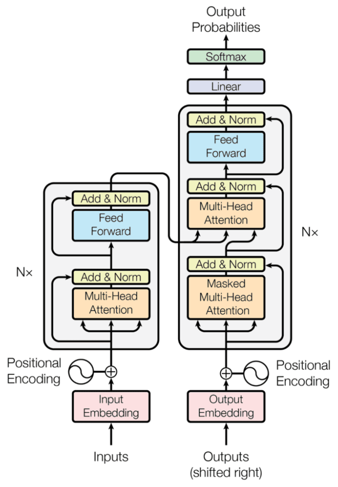
Allegro: GPU Simulation Acceleration for Machine Learning Workloads

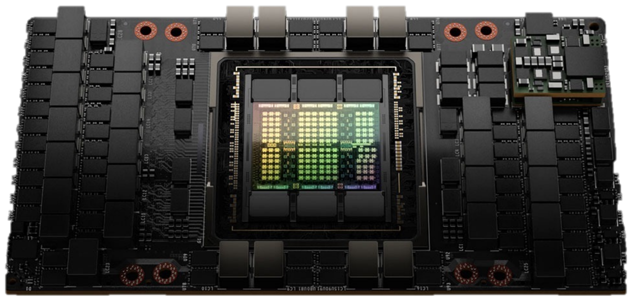
Euijun Chung, Seonjin Na, Hyesoon Kim

HPArch (High Performance Architecture Lab) @ Georgia Tech

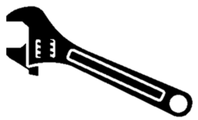
Georgia Institute of Technology

GPUs and Architectural Simulators



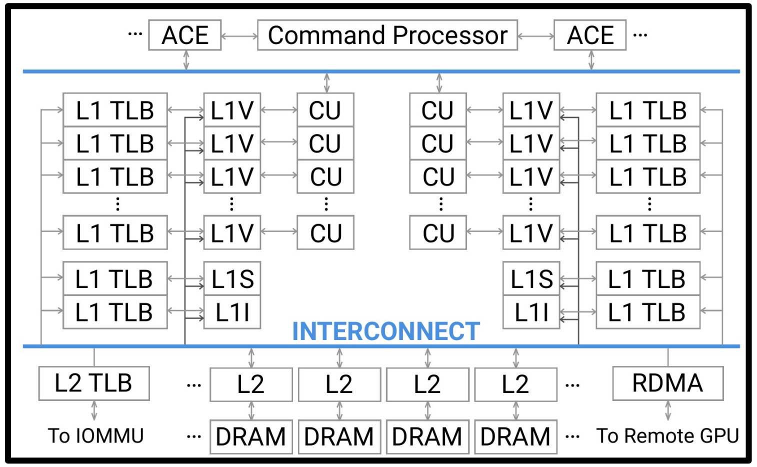


Nvidia's H100 GPU



Architectural design changes

Evaluation & Validation



GPU Simulator

IPC
Cache Hit Rate
Num of Instrs
Memory Access
TLB Hit Rate
…

Source: https://www.nvidia.com/en-us/data-center/h100/
"Attention is all you need", NeurIPS 2017

Source: MGPUSim: Enabling Multi-GPU Performance Modeling and Optimization, ISCA 2019

Motivation: GPU Simulators are **too slow**

TABLE I
THROUGHPUT AND SLOWDOWN OF GPU SIMULATORS.

	Real GPU	Macsim	GPGPU-Sim	MGPUSim
Simulation Rate (KIPS)	4103750	50.5	12.5	27
Relative Throughput	328300	4.04	1	2.16
GPT-2: Generate 100 tokens	0.925 sec	**20.88 hrs**	**3.52 days**	**1.63 days**

*Real GPU: RTX 2080

→ **A few days** to generate **one sentence** with 100 tokens with GPT-2

→ **Reducing the workload size** is a huge problem to solve

→ Allegro's solution: **Kernel-wise Sampling** with execution statistics

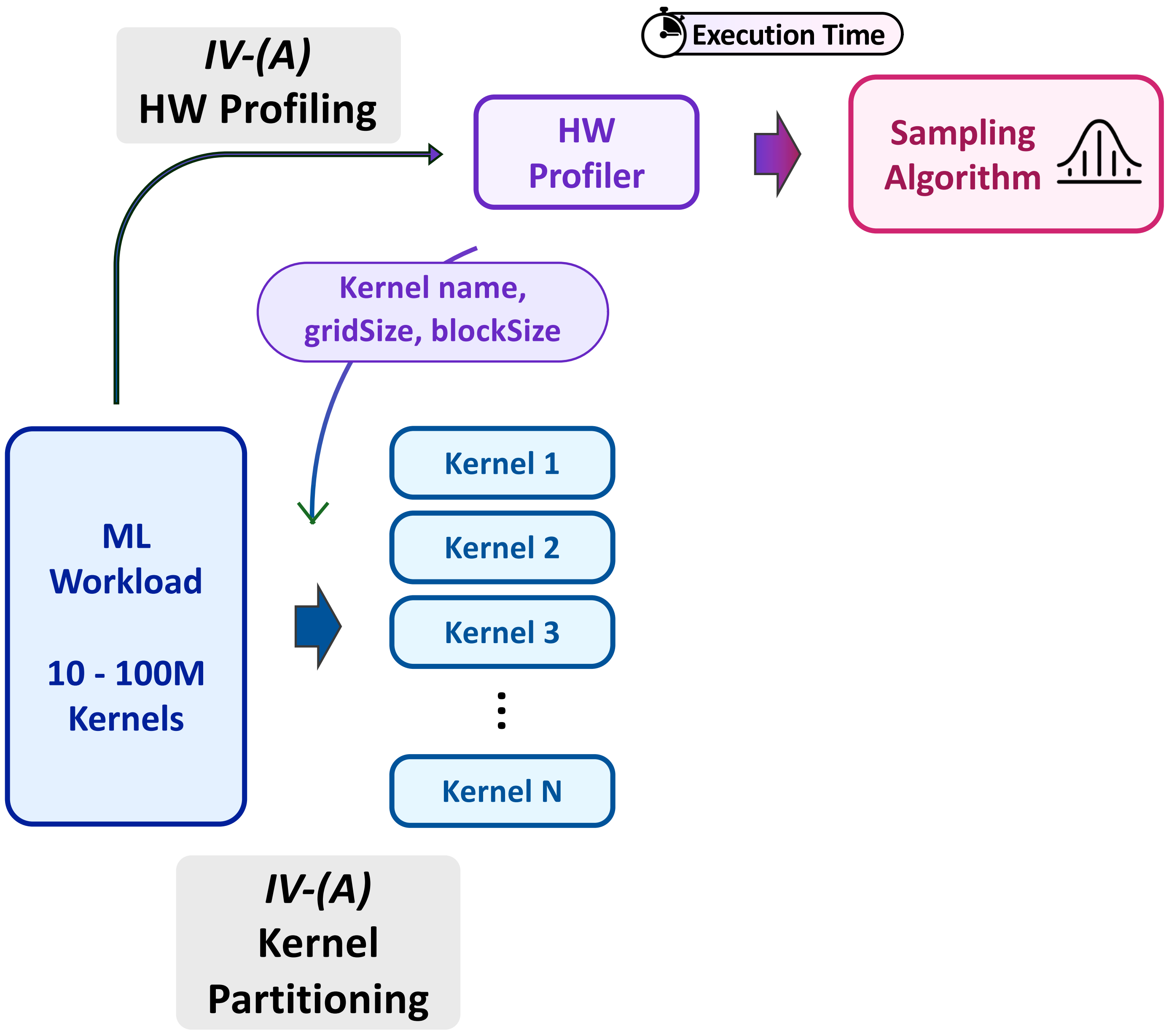
Observations: 1. High Homogeneity

TABLE II
TOP 5 TIME-CONSUMING GPU KERNELS IN RESNET50 [14] WORKLOAD.

Kernel Name	# Calls	Total Time (ns)
cudnn_infer_volta_scudnn_winograd_128x...	19625	1185625785
explicit_convolve_sgemm	3925	964880834
cudnn_infer_volta_scudnn_winograd_128x...	7850	897755249
volta_sgemm_128x64_nn	23550	709594145
winograd::generateWinogradTilesKernel	7850	595149925

✓ Highly **repeated** kernel calls

✓ Good opportunity for **efficient sampling**


Execution Time
IV-(A)
HW Profiling
HW
Profiler
Sampling
Algorithm
Kernel name,
gridSize, blockSize
ML
Workload
10 - 100M
Kernels
Kernel 1
Kernel 2
Kernel 3
Kernel N
IV-(A)
Kernel
Partitioning

Observations: 1. High Homogeneity

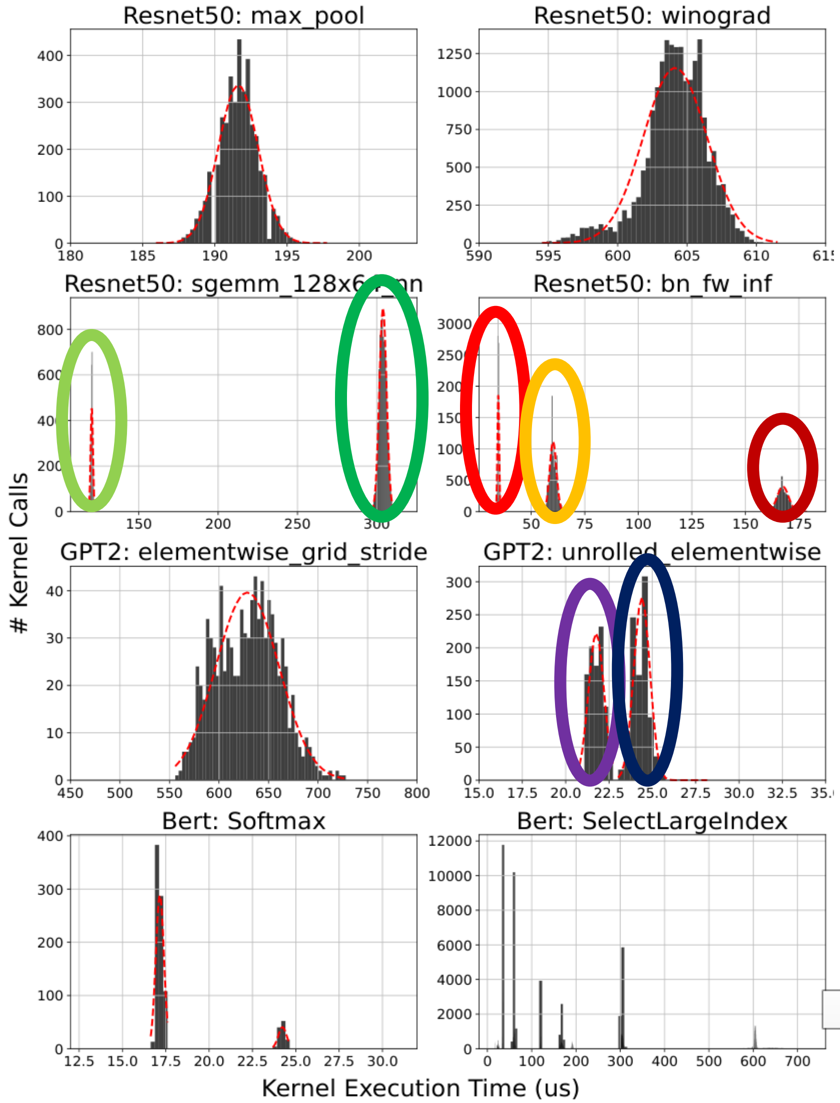
TABLE II
TOP 5 TIME-CONSUMING GPU KERNELS IN RESNET50 [14] WORKLOAD.

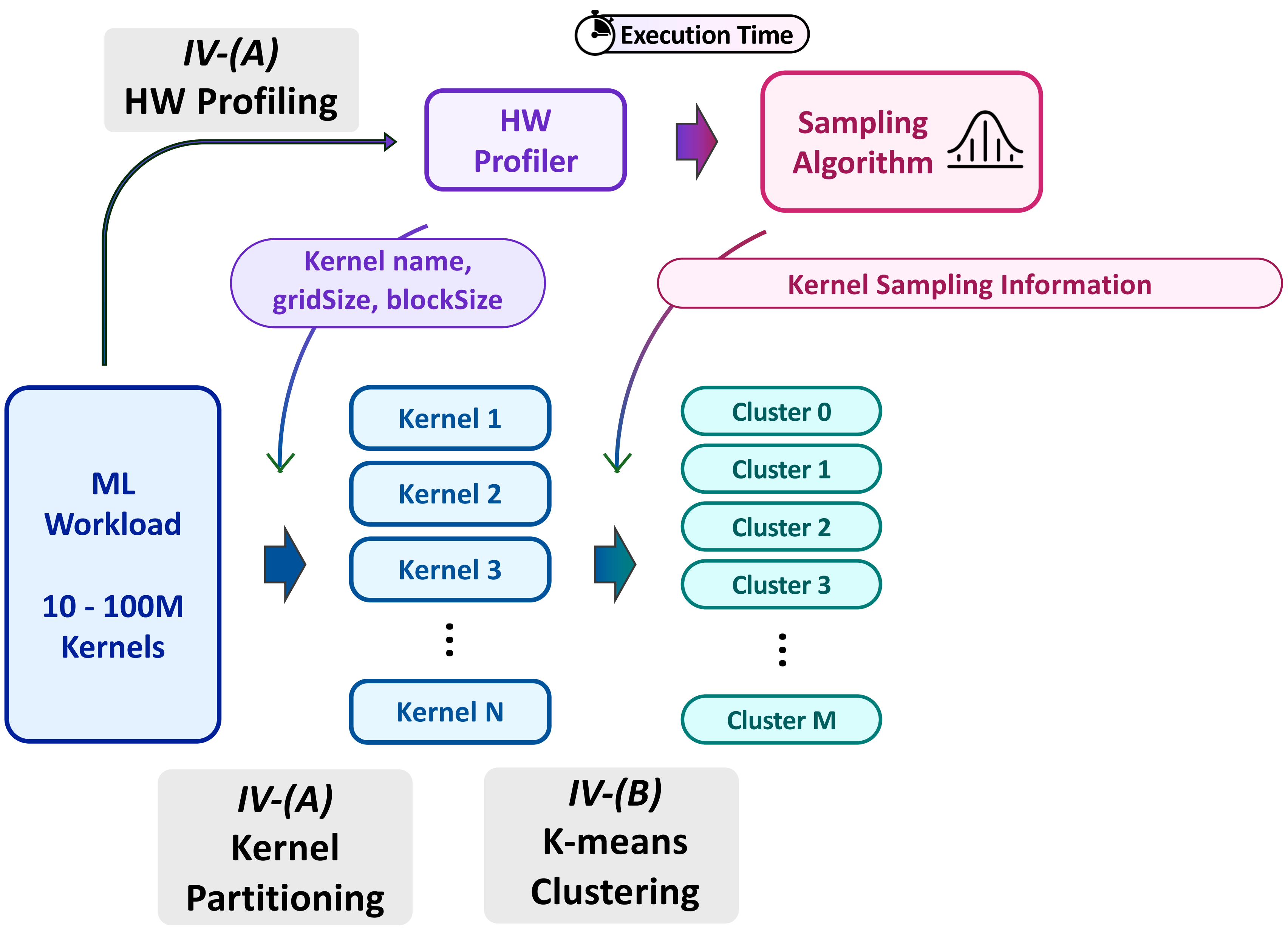
Kernel Name	# Calls	Total Time (ns)
cudnn_infer_volta_scudnn_winograd_128x...	19625	1185625785
explicit_convolve_sgemm	3925	964880834
cudnn_infer_volta_scudnn_winograd_128x...	7850	897755249
volta_sgemm_128x64_nn	23550	709594145
winograd::generateWinogradTilesKernel	7850	595149925

- Kernel names on the top of each subplot.
- Red dotted lines are normal distribution with same mean and variance.

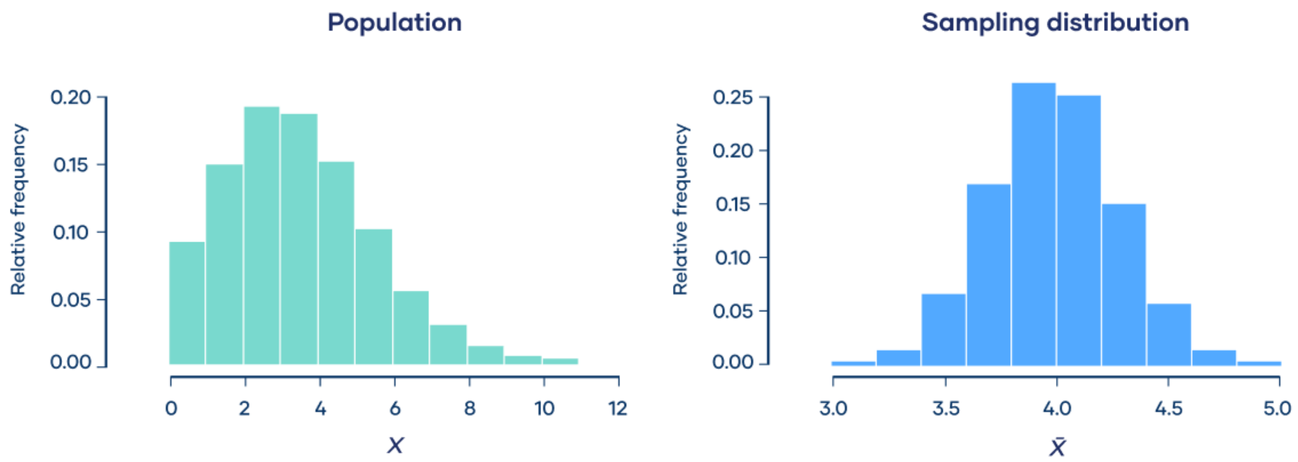
✓ **Narrow** execution time distributions

→ Clustering & Sampling




IV-(A)
HW Profiling
Execution Time
HW
Profiler
Sampling
Algorithm
Kernel name,
gridSize, blockSize
Kernel Sampling Information
ML
Workload
10 - 100M
Kernels
Kernel 1
Kernel 2
Kernel 3
Kernel N
Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster M
IV-(A)
Kernel
Partitioning
IV-(B)
K-means
Clustering

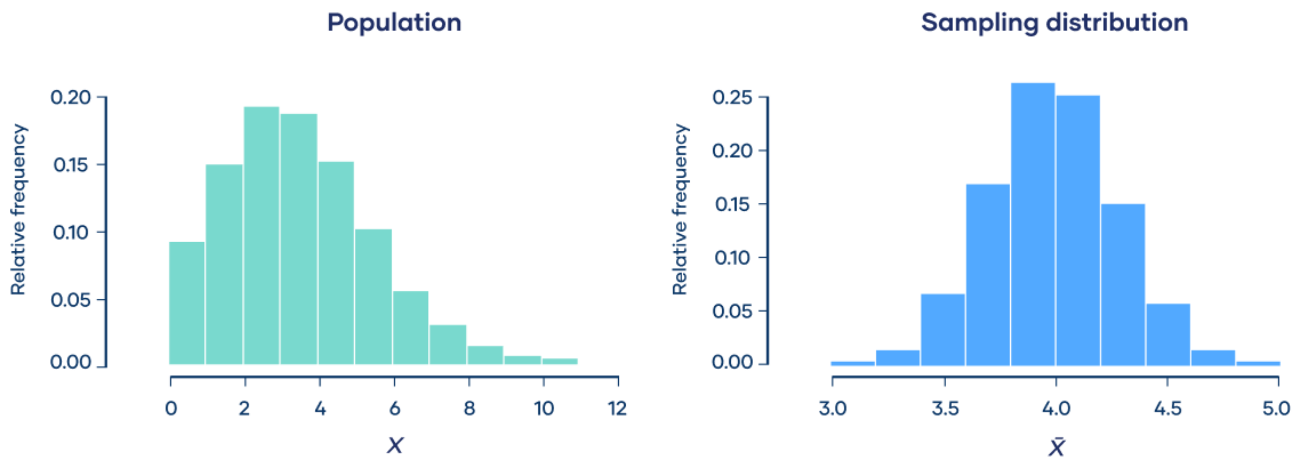
Applying Central Limit Theorem (CLT)



Source: https://www.scribbr.com/statistics/central-limit-theorem/

Central Limit Theorem.
Let $\{X_1, \dots, X_m\}$ be a sequence of m independent and identically distributed (i.i.d.) random variables following $N(\mu, \sigma^2)$.
Then, **the sampled mean converges to $\boldsymbol{N(\mu, \sigma^2/m)}$** as $m \to \infty$.

Applying Central Limit Theorem (CLT)

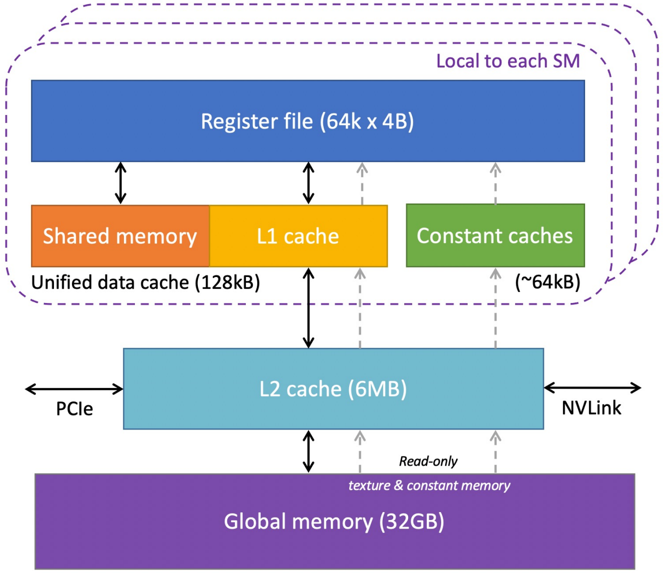


Source: https://www.scribbr.com/statistics/central-limit-theorem/

Central Limit Theorem.
Let $\{X_1, \ldots, X_m\}$ be a sequence of m independent and identically distributed (i.i.d.) random variables following $N(\mu, \sigma^2)$.
Then, **the sampled mean converges to $\boldsymbol{N(\mu, \sigma^2/m)}$** as $m \rightarrow \infty$.

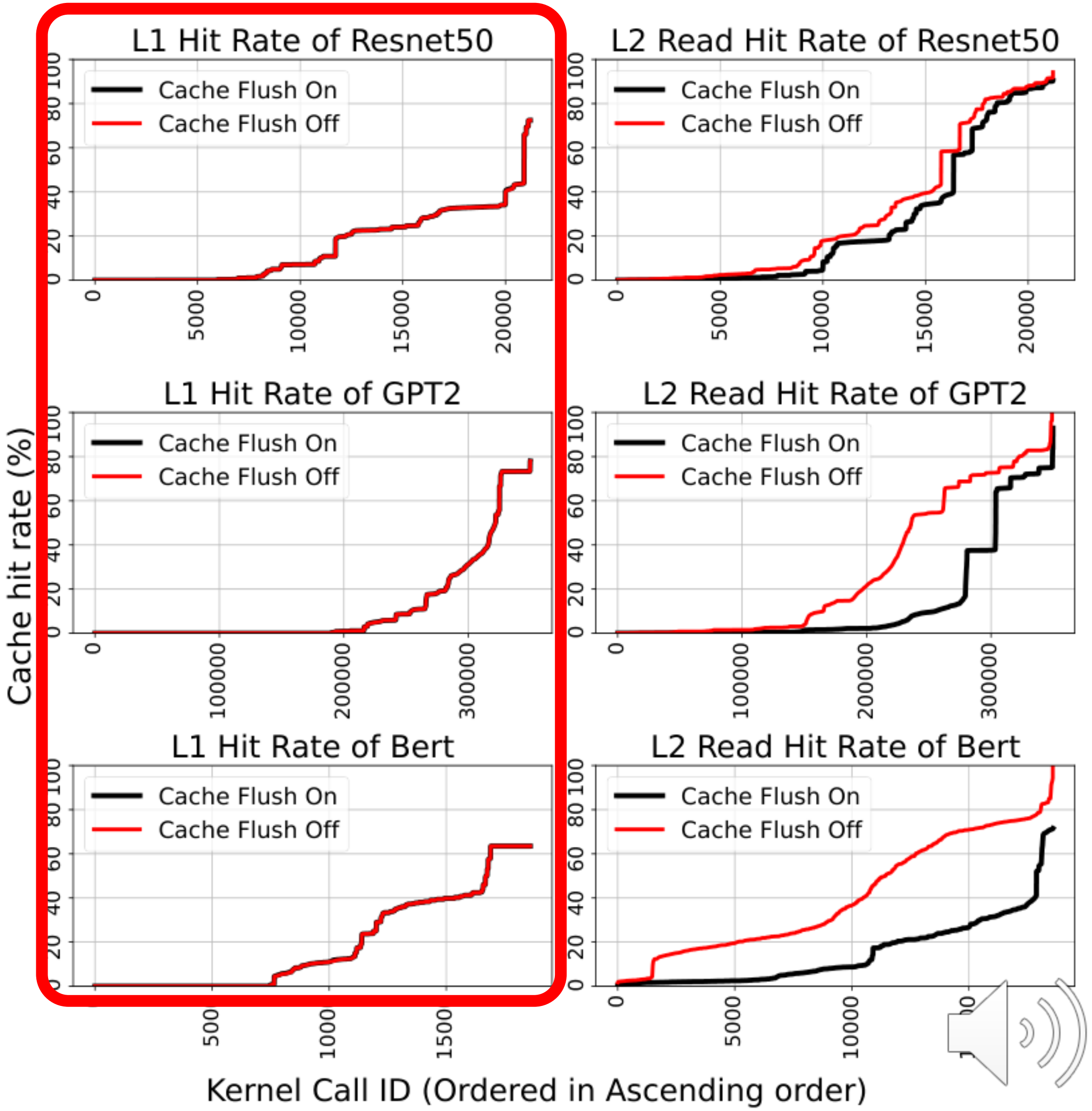
✓ Can we ensure i.i.d.-ness of GPU kernels?

Observations: 2. Cache-Unfriendliness



GPU Cache Hierarchy.

Source: https://cvw.cac.cornell.edu/gpu-architecture/gpu-memory/memory_levels



✓ **Cache Flushing** between kernel calls

→ No difference in the L1 hit rate

→ Small difference in the L2 hit rate

Allegro's sampling algorithm

error $:=$ execution time difference between **full** and **sampled simulation**

m_{min} $:=$ minimum # of samples s.t. ***error < error bound*** $\boldsymbol{\epsilon}$.

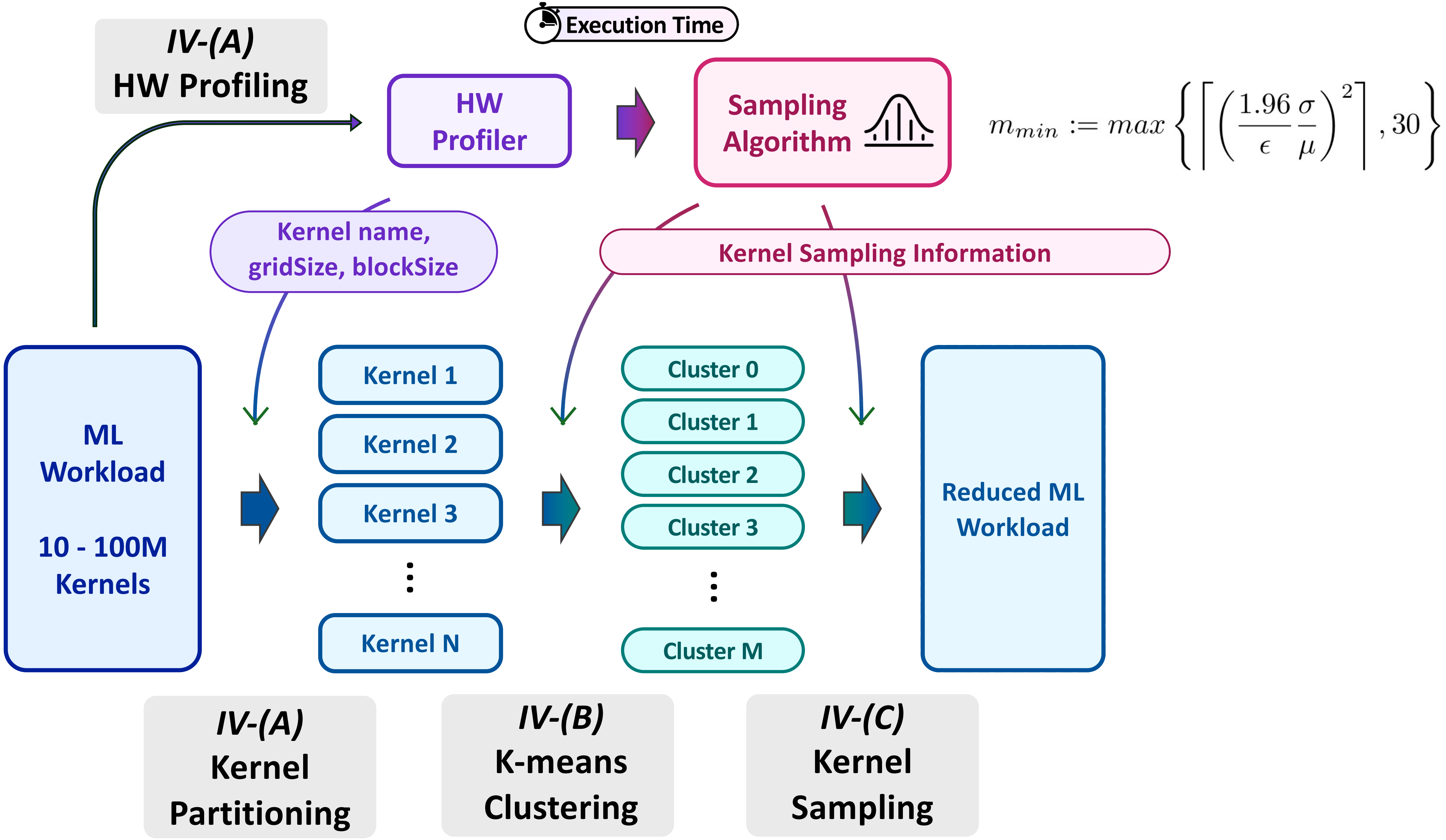
For 95% confidence, m_{min} to ensure ***error < error bound*** $\boldsymbol{\epsilon}$:

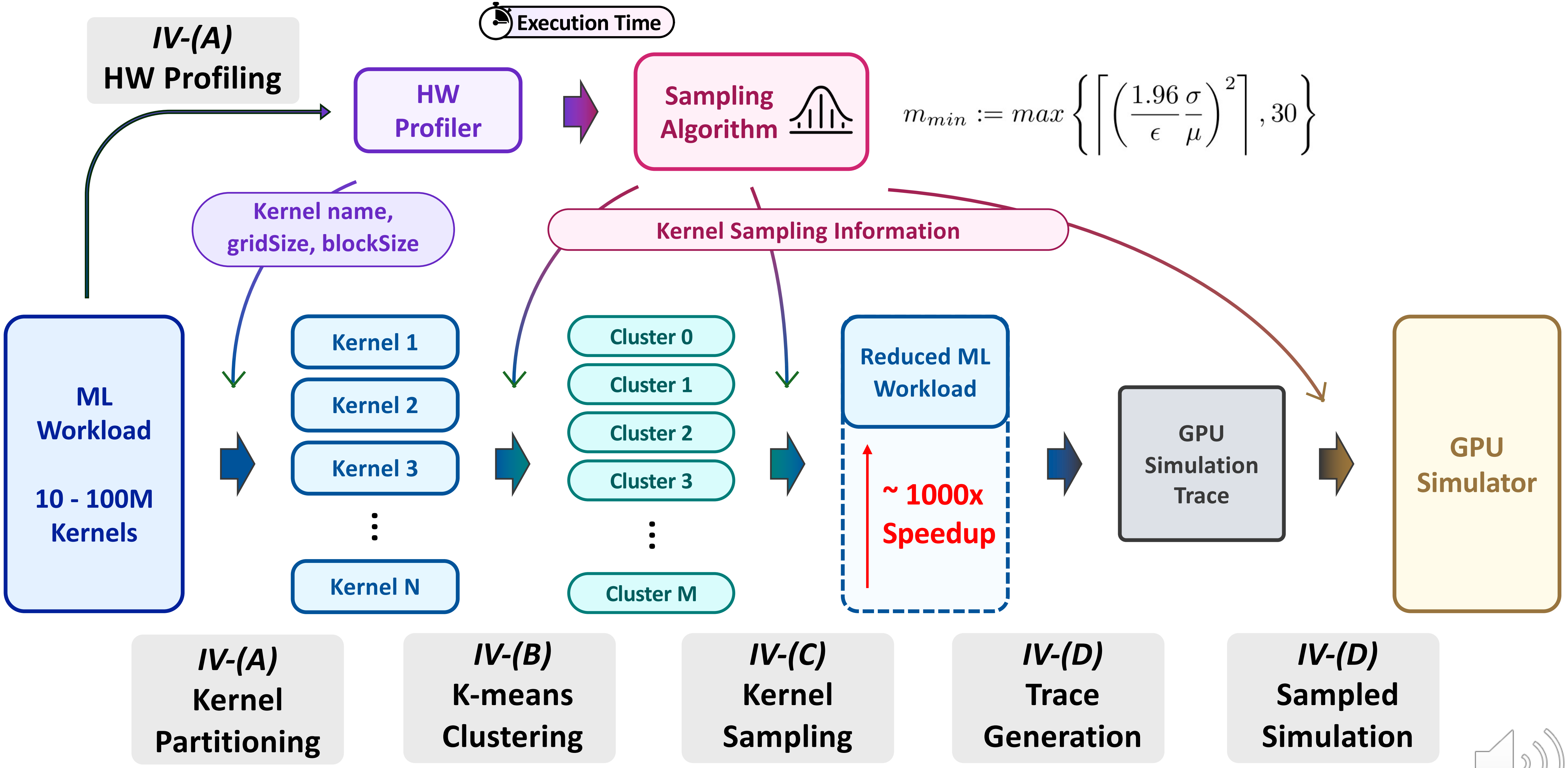
$$m_{min} := max\left\{\left\lceil\left(\frac{1.96}{\boxed{\epsilon}}\boxed{\frac{\sigma}{\mu}}\right)^2\right\rceil\boxed{, 30}\right\}$$

Where μ $=$ mean and σ $=$ stdev of execution times.

→ **Clustering**: Compare m_{min} with a threshold recursively.

→ **Sampling**: Randomly sample m_{min} kernel calls from each cluster.


IV-(A)
HW Profiling
Execution Time
HW Profiler
Sampling Algorithm
$m_{min} := max\left\{\left\lceil\left(\frac{1.96}{\epsilon}\frac{\sigma}{\mu}\right)^2\right\rceil, 30\right\}$
Kernel name, gridSize, blockSize
Kernel Sampling Information
ML Workload
10 - 100M Kernels
Kernel 1
Kernel 2
Kernel 3
Kernel N
Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster M
Reduced ML Workload
IV-(A)
Kernel Partitioning
IV-(B)
K-means Clustering
IV-(C)
Kernel Sampling


IV-(A)
HW Profiling
Execution Time
HW Profiler
Sampling Algorithm
$m_{min} := max\left\{\left\lceil\left(\frac{1.96}{\epsilon}\frac{\sigma}{\mu}\right)^2\right\rceil, 30\right\}$
Kernel name, gridSize, blockSize
Kernel Sampling Information
ML Workload
10 - 100M Kernels
Kernel 1
Kernel 2
Kernel 3
Kernel N
Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster M
Reduced ML Workload
~ 1000x Speedup
GPU Simulation Trace
GPU Simulator
IV-(A) Kernel Partitioning
IV-(B) K-means Clustering
IV-(C) Kernel Sampling
IV-(D) Trace Generation
IV-(D) Sampled Simulation

Evaluation Setups

GPU: Nvidia RTX 2080 (Volta architecture), CUDA 11.8

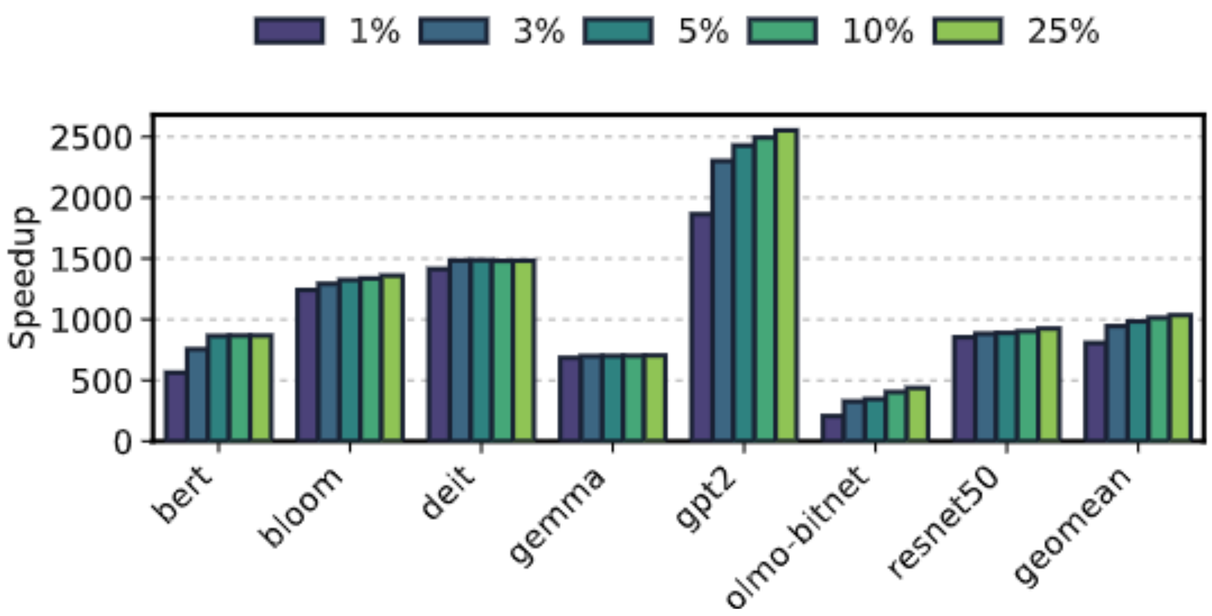
HW Profiler: Nsight-Systems

List of used workloads:

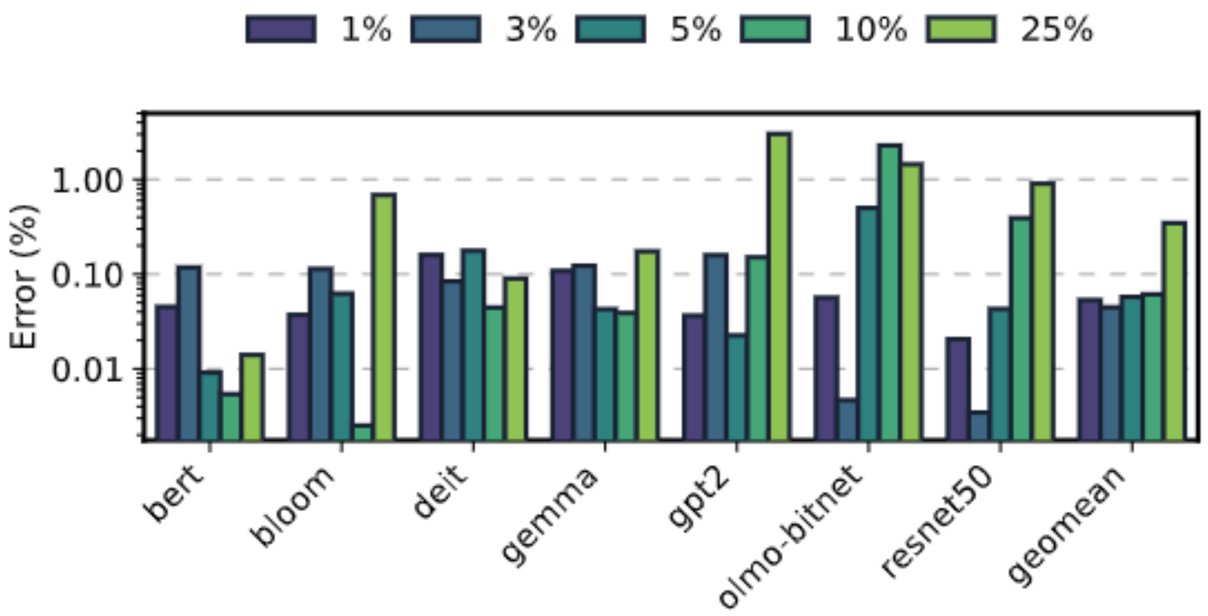
- 7 Transformer/CNN based models
- Python-based implementations from HuggingFace

Name	# Kernels	Workload Description
Bert	1858800	Performing sequence classification on 10,000 premise/hypothesis pairs using the BERT-Medium-MNLI model.
Bloom	51834362	Generating 1,000 sentences, each with a length of 100 tokens, using the Bloom model.
Deit	792850	Classifying 3,925 ImageNet datasets using the Data-efficient image Transformer (DeiT) model.
Gemma	9079126	Generating 1,000 sentences, each with a length of 100 tokens, from the GEMMA language model.
GPT-2	34981000	Generating 1,000 sentences, each with a length of 100 tokens, from the GPT-2 model.
Olmo-bitnet	2544766	Generating 10 sentences, each with a length of 100 tokens, from the OLMo-Bitnet language model.
ResNet50	2812741	Classifying 13,400 ImageNet datasets using the ResNet50 model.

Evaluation: Speedup and Error



Speedup for $\epsilon = 1\%$ to $\epsilon = 25\%$
Average ~1000x Speedup

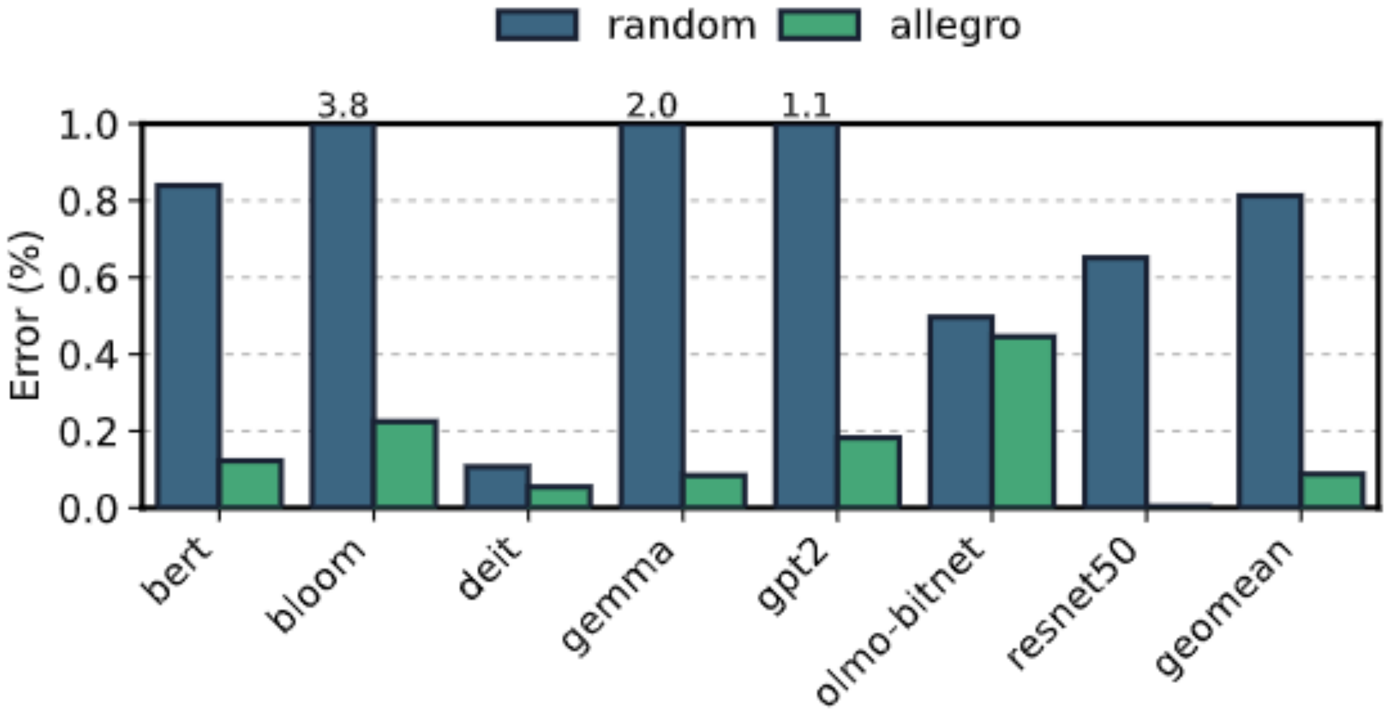


Error for $\epsilon = 1\%$ to $\epsilon = 25\%$
Average 0.057% Error

Evaluation: Comparison

Random Sampling:

Randomly sample kernels until achieving the same speedup as Allegro



Comparison with Random Sampling:
Average ~9.2x lower error

Limitations

✓ **Homogeneity** and **i.i.d.** assumptions:

The error may exceed the error bound ϵ

✓ **Non-ML workloads** with small number of kernel calls:

Ex) Rodinia Suite: typically involves only a few kernel calls

✓ **Cache** warm-ups effects:

Applies to all methodologies aiming for speed-up by sampling

Allegro's contributions

1. Analysis of the **latest ML workloads**' characteristics on GPUs

    Homogeneity and **cache-unfriendly** nature

2. Propose a **statistical approach** to effectively reduce the workload size

    Central Limit Theorem (CLT) for calculating **error bounds**

3. Propose **clustering** and **sampling** method for ML workloads

    ~922x performance boost with **high accuracy (0.057% error)**

    Tested 7 latest ML workloads on Macsim

Summary of **Allegro**

✓ **Homogeneous** and **cache-unfriendly** nature of ML workloads

✓ **Sampling** based on i.i.d. behavior of GPU kernels

✓ **Statistical bounds** on sampling errors

✓ GPU Simulation with **7 latest ML workloads**

- **Euijun Chung**, Seonjin Na, Hyesoon Kim
- HPArch (High Performance Architecture Lab

Georgia Institute of Technology