Modernizing GPU Benchmarking : Progress in Precision, Reproducibility, and Analysis

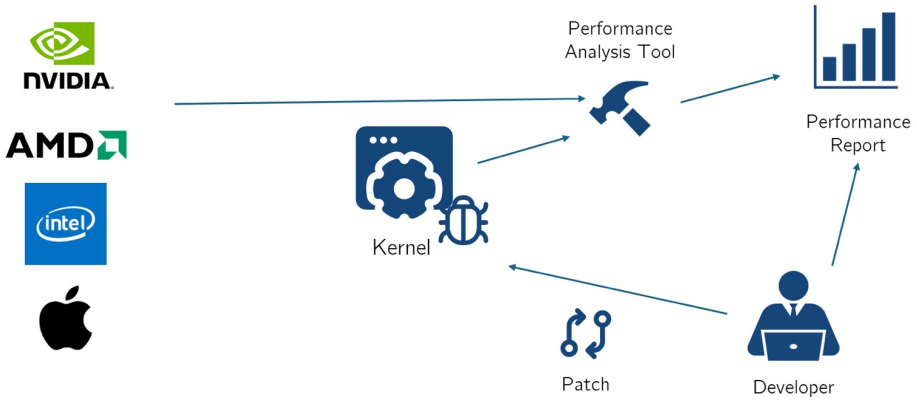
Come Eyraud, Maxime Lamothe

DORSAL

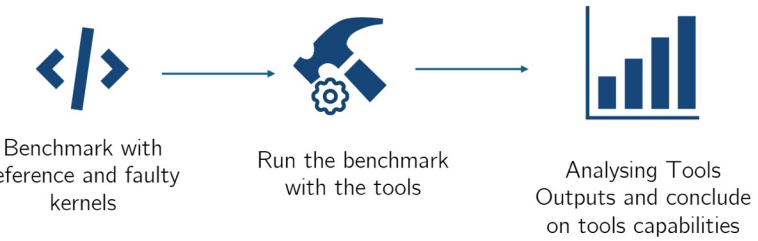
POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Context and Progress since May

Context & Motivation



Proposed Methodology



Modernizing GPU Benchmarking

Non exhaustive list of Literature GPU Benchmarks (Support Cuda or Hip) :

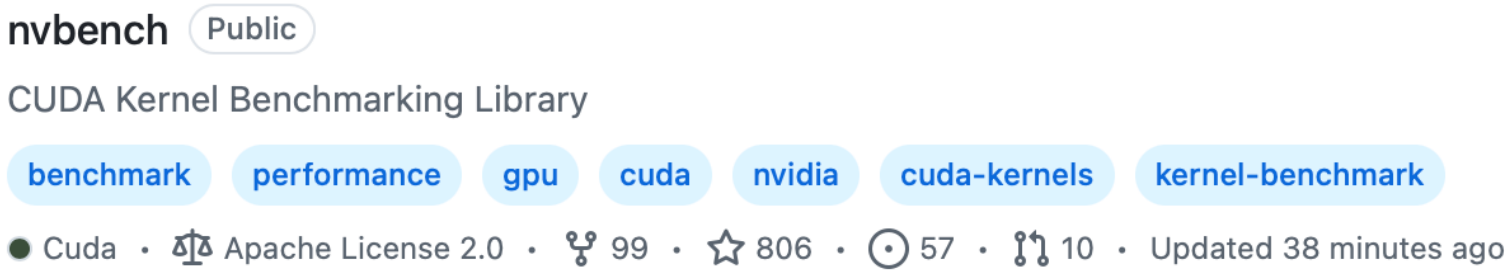
Name	Specificity	Year
Rodinia	Cover Berkley Dwarfs	2009
Lonestar	Irregular Algorithms	2009
Shoc	Scalability for clusters	2010
Parboil	Throughput — optimisation levels	2012
Nupar	Nested kernels — Unified memory	2015
Chai	Collaborative execution CPU - GPU	2017
Altis	More problem diversity	2020
HeteroBench	Python Binding — Multi Kernel — Diverse platforms	2025

Cons :

- Hard-Coded Options
- Low Reusability
- Bad measuring practices
- Each iteration targeting new hardware capabilities
- Diverse implementation style in each benchmark

Modernizing GPU Benchmarking

NVBench : NVIDIA's Benchmarking library

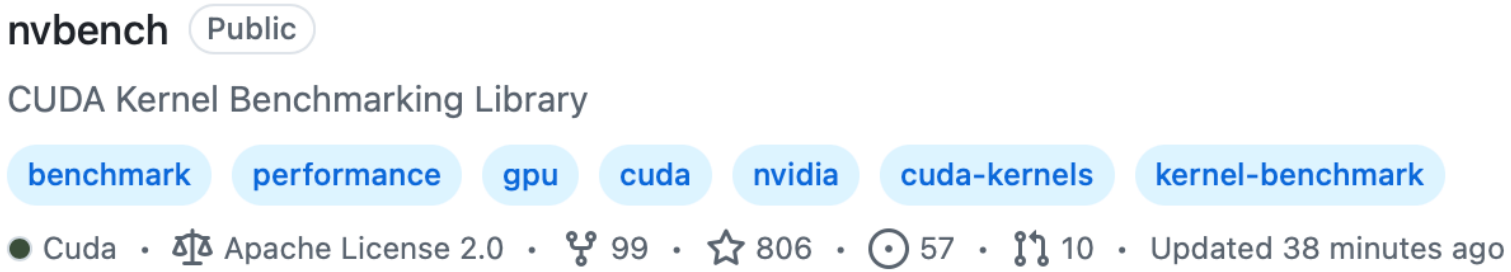


Pros :

+ Define once the kernel, redefine its options via the command line
+ Accurate measurement of Kernels
+ Dynamic stopping criterion
+ Benchmark logic and kernel logic decoupled

Modernizing GPU Benchmarking

NVBench : NVIDIA's Benchmarking library



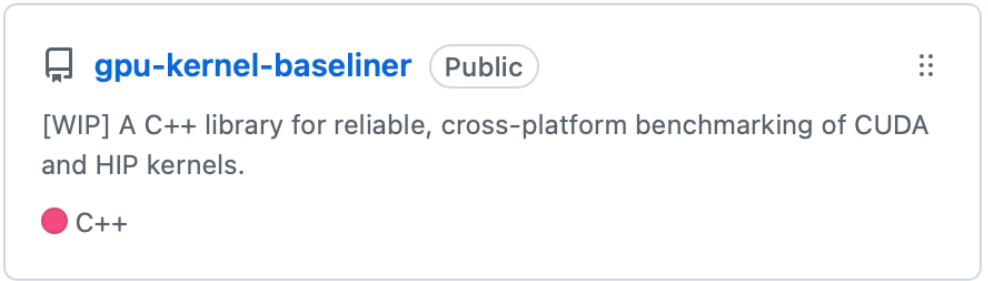
Cons :

- Hard-Coded to the CUDA Runtime
- Arbitrary Stopping Criterion
- Lacks evidence to back its choices and assumptions
- Missing kernel execution comparison
- Not defining strict interface, Code is still Ad Hoc

Designing a Benchmarking Tool

We propose Baseliner, a backend agnostic GPU kernel benchmarking library

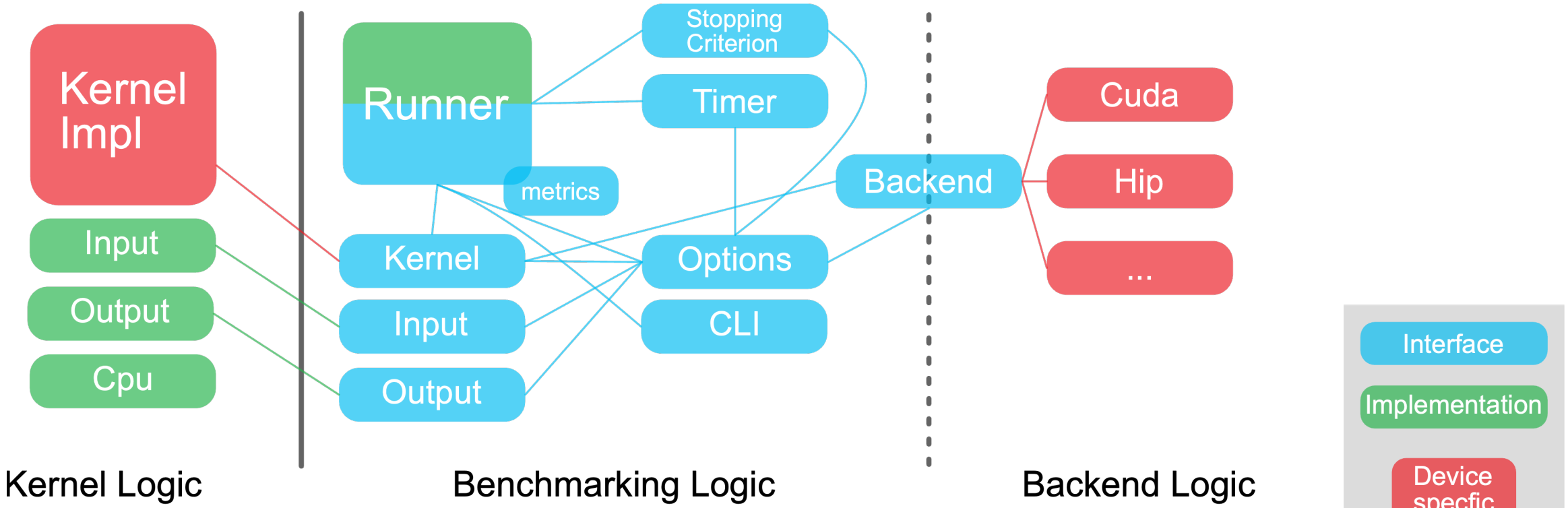
github.com/comeyrd/gpu-kernel-baseliner



Goal :

- Define a standard Kernel definition interface
- Separate benchmarking logic, kernel logic and backend logic
- Provide developers with built-in proven optimal setup for lightweight, accurate and reproducible performance measures.
- Library for both Benchmarking and Regression Tests needs.
- Highly customizable and plugin oriented (future proof)

Designing a Benchmarking Tool



Designing a Benchmarking Tool

Proving the optimal setup and its perks

Quantify the Impact of theses variables

- Different input values
- Work size
- Numbers of warmups
- Flushing cache
- Enqueuing kernels
- GPU Frequency

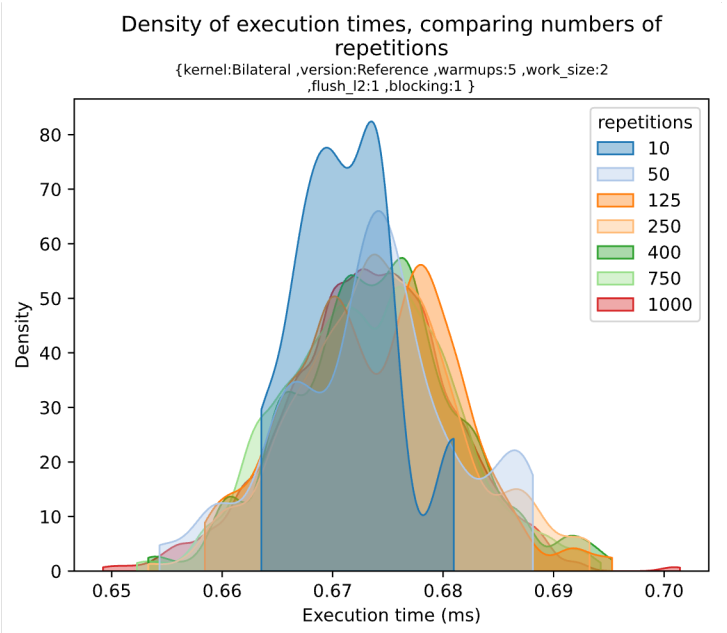
Understanding Kernel Execution Times

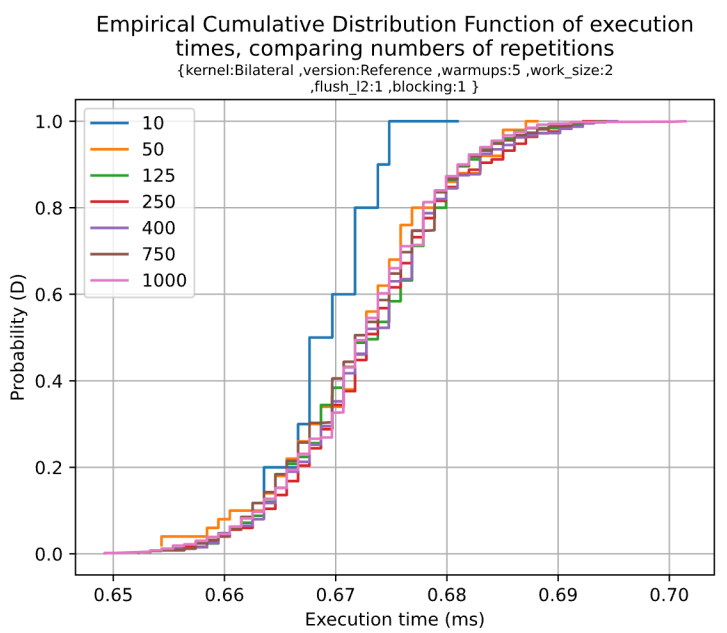
Why Measurement is not trivial :

- Non-Deterministic Noise
- Hardware Dynamics
- System interference

Stopping Criteria :

- Fixed Execution
- Bootstrapping (Confidence Interval Mean – Std Deviation)
- Confidence Interval Median
- Nvidia's 'Entropy' Criterion





Understanding Kernel Execution Times

Comparing Executions :

Real Distances (ms)

Total Area (average magnitude of the time difference)

Signed Area (average net time gained or lost)

Probabilistic Metrics (D)

K-S Test (Maximum difference between the distributions)

CVM Test (How interleaved the distributions are)

Probability of Superiority (What's the probability that a run from A is faster than a run from B)

Understanding Kernel Execution Times

Comparing Executions :

Real Distances (ms)

Total Area (average magnitude of the time difference)

Signed Area (average net time gained or lost)

Probabilistic Metrics (D)

K-S Test (Maximum difference between the distributions)

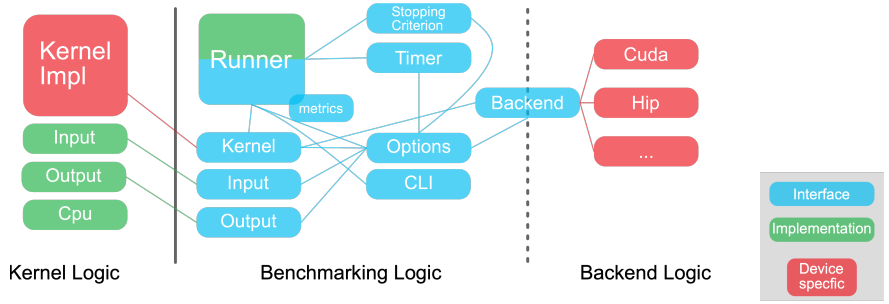
CVM Test (How interleaved the distributions are)

Probability of Superiority (What's the probability that a run from A is faster than a run from B)

- Understanding Performance Profile of a Kernel, Give Card context
  - Using "micro-benchmarks" to get the performance of the card under this setup

Conclusion

Designing a Benchmarking Tool 7



Designing a Benchmarking Tool 8

Proving the optimal setup and its perks

Quantify the Impact of theses variables

- Different input values
- Work size
- Numbers of warmups
- Flushing cache
- Enqueuing kernels
- GPU Frequency

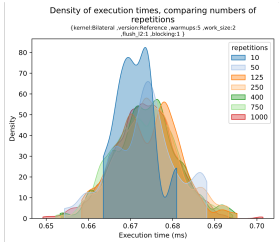
Understanding Kernel Execution Times 9

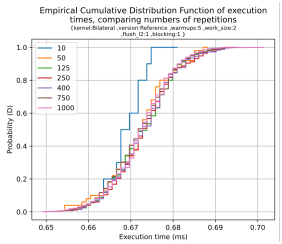
Why Measurement is not trivial :

- Non-Deterministic Noise
- Hardware Dynamics
- System interference

Stopping Criteria :

- Fixed Execution
- Bootstrapping (Confidence Interval Mean — Std Deviation)
- Confidence Interval Median
- Nvidia's 'Entropy' Criterion





Understanding Kernel Execution Times 11

Comparing Executions :

Real Distances (ms)

- Total Area (average magnitude of the time difference)
- Signed Area (average net time gained or lost)

Probabilistic Metrics (D)

- K-S Test (Maximum difference between the distributions)
- CVM Test (How interleaved the distributions are)
- Probability of Superiority (What's the probability that a run from A is faster than a run from B)

- Understanding Performance Profile of a Kernel, Give Card context
  - Using "micro-benchmarks" to get the performance of the card under this setup