Proton

A ***Pro***filer for Tri***ton***

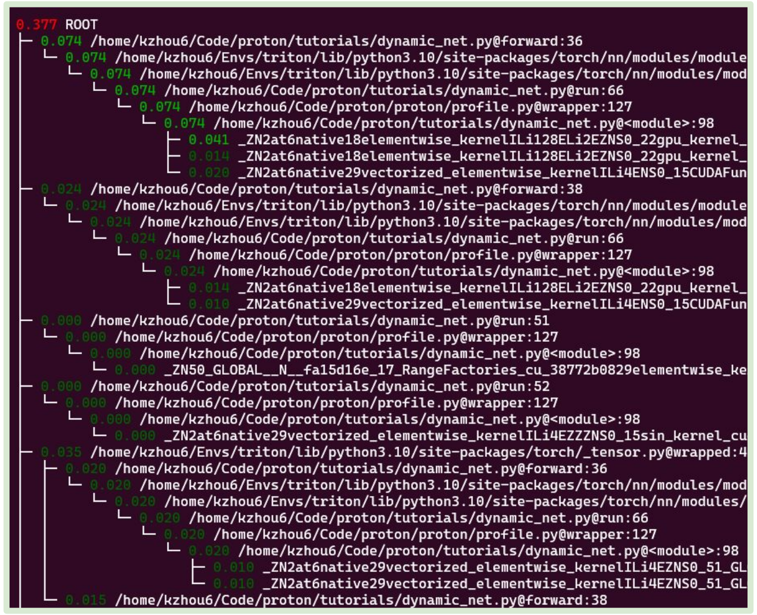
Keren Zhou
kzhou6@gmu.edu

Goals

- Provide a quick, intuitive, and simple way to check kernel performance
  - Open source
  - Multiple vendor GPUs
  - Flexible metrics collection
    - Hardware metrics
    - Software metrics
  - Call path profiling

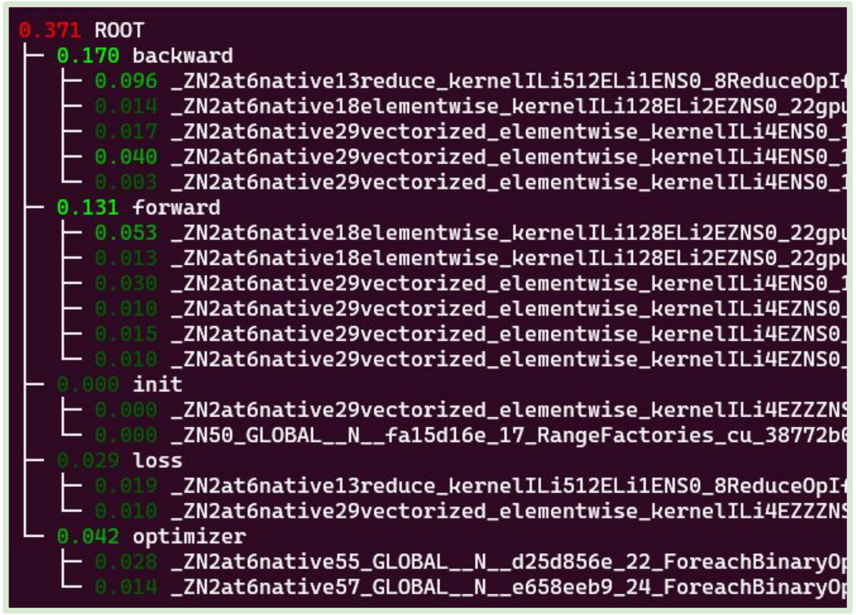
Call Path Profiling

- Profile kernel running time



```
0.377 ROOT
├─ 0.074 /home/kzhou6/Code/proton/tutorials/dynamic_net.py@forward:36
│  └─ 0.074 /home/kzhou6/Envs/triton/lib/python3.10/site-packages/torch/nn/modules/module
│     └─ 0.074 /home/kzhou6/Envs/triton/lib/python3.10/site-packages/torch/nn/modules/mod
│        └─ 0.074 /home/kzhou6/Code/proton/tutorials/dynamic_net.py@run:66
│           └─ 0.074 /home/kzhou6/Code/proton/proton/profile.py@wrapper:127
│              └─ 0.074 /home/kzhou6/Code/proton/tutorials/dynamic_net.py@<module>:98
│                 ├─ 0.041 _ZN2at6native18elementwise_kernelILi128ELi2EZNS0_22gpu_kernel_
│                 ├─ 0.014 _ZN2at6native18elementwise_kernelILi128ELi2EZNS0_22gpu_kernel_
│                 └─ 0.020 _ZN2at6native29vectorized_elementwise_kernelILi4ENS0_15CUDAFun
├─ 0.024 /home/kzhou6/Code/proton/tutorials/dynamic_net.py@forward:38
│  └─ 0.024 /home/kzhou6/Envs/triton/lib/python3.10/site-packages/torch/nn/modules/module
│     └─ 0.024 /home/kzhou6/Envs/triton/lib/python3.10/site-packages/torch/nn/modules/mod
│        └─ 0.024 /home/kzhou6/Code/proton/tutorials/dynamic_net.py@run:66
│           └─ 0.024 /home/kzhou6/Code/proton/proton/profile.py@wrapper:127
│              └─ 0.024 /home/kzhou6/Code/proton/tutorials/dynamic_net.py@<module>:98
│                 ├─ 0.014 _ZN2at6native18elementwise_kernelILi128ELi2EZNS0_22gpu_kernel_
│                 └─ 0.010 _ZN2at6native29vectorized_elementwise_kernelILi4ENS0_15CUDAFun
├─ 0.000 /home/kzhou6/Code/proton/tutorials/dynamic_net.py@run:51
│  └─ 0.000 /home/kzhou6/Code/proton/proton/profile.py@wrapper:127
│     └─ 0.000 /home/kzhou6/Code/proton/tutorials/dynamic_net.py@<module>:98
│        └─ 0.000 _ZN50_GLOBAL__N__fa15d16e_17_RangeFactories_cu_38772b0829elementwise_ke
├─ 0.000 /home/kzhou6/Code/proton/tutorials/dynamic_net.py@run:52
│  └─ 0.000 /home/kzhou6/Code/proton/proton/profile.py@wrapper:127
│     └─ 0.000 /home/kzhou6/Code/proton/tutorials/dynamic_net.py@<module>:98
│        └─ 0.000 _ZN2at6native29vectorized_elementwise_kernelILi4EZZZNS0_15sin_kernel_cu
├─ 0.035 /home/kzhou6/Envs/triton/lib/python3.10/site-packages/torch/_tensor.py@wrapped:4
│  ├─ 0.020 /home/kzhou6/Code/proton/tutorials/dynamic_net.py@forward:36
│  │  └─ 0.020 /home/kzhou6/Envs/triton/lib/python3.10/site-packages/torch/nn/modules/mod
│  │     └─ 0.020 /home/kzhou6/Envs/triton/lib/python3.10/site-packages/torch/nn/modules/
│  │        └─ 0.020 /home/kzhou6/Code/proton/tutorials/dynamic_net.py@run:66
│  │           └─ 0.020 /home/kzhou6/Code/proton/proton/profile.py@wrapper:127
│  │              └─ 0.020 /home/kzhou6/Code/proton/tutorials/dynamic_net.py@<module>:98
│  │                 ├─ 0.010 _ZN2at6native29vectorized_elementwise_kernelILi4EZNS0_51_GL
│  │                 └─ 0.010 _ZN2at6native29vectorized_elementwise_kernelILi4EZNS0_51_GL
│  └─ 0.015 /home/kzhou6/Code/proton/tutorials/dynamic_net.py@forward:38
```

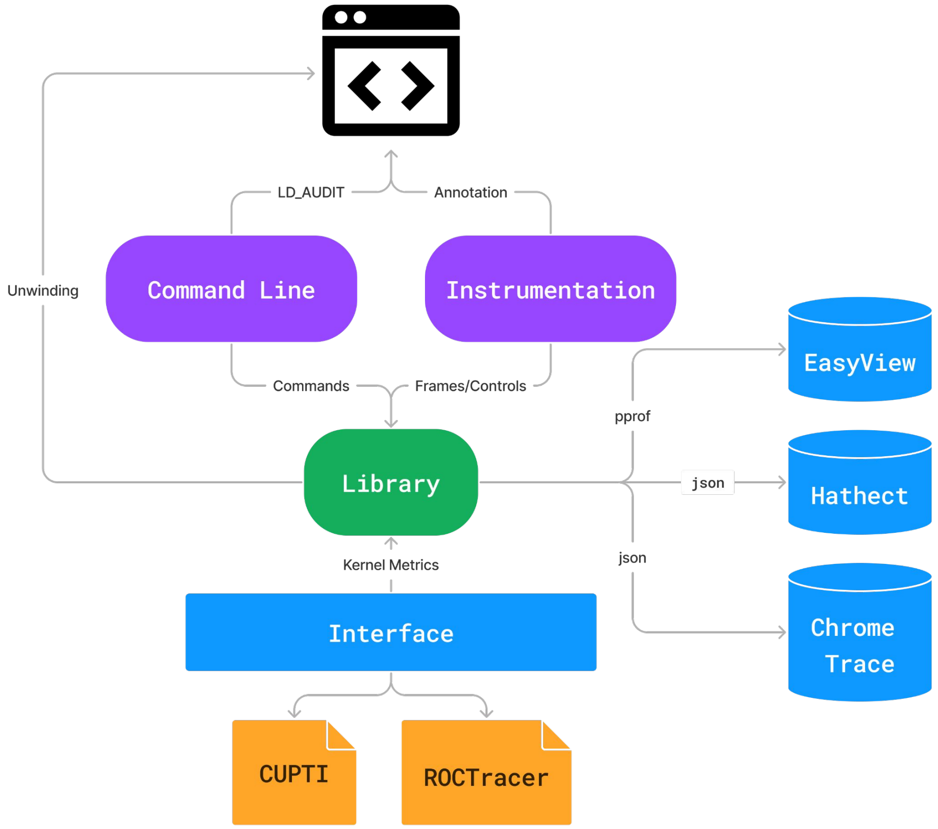
Python Context



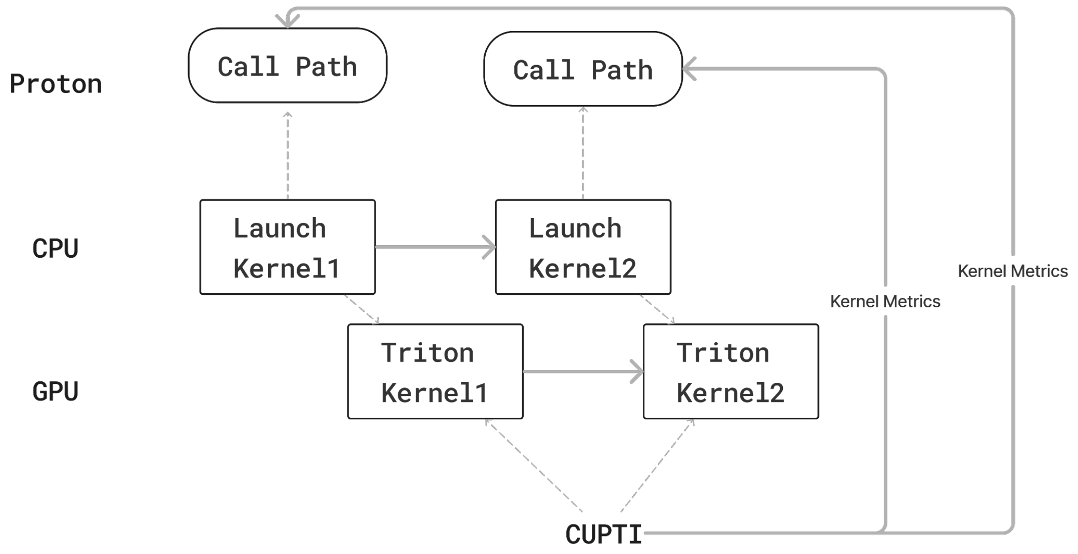
```
0.371 ROOT
├─ 0.170 backward
│  ├─ 0.096 _ZN2at6native13reduce_kernelILi512ELi1ENS0_8ReduceOpI
│  ├─ 0.014 _ZN2at6native18elementwise_kernelILi128ELi2EZNS0_22gpu
│  ├─ 0.017 _ZN2at6native29vectorized_elementwise_kernelILi4ENS0_1
│  ├─ 0.040 _ZN2at6native29vectorized_elementwise_kernelILi4ENS0_1
│  └─ 0.003 _ZN2at6native29vectorized_elementwise_kernelILi4ENS0_1
├─ 0.131 forward
│  ├─ 0.053 _ZN2at6native18elementwise_kernelILi128ELi2EZNS0_22gpu
│  ├─ 0.013 _ZN2at6native18elementwise_kernelILi128ELi2EZNS0_22gpu
│  ├─ 0.030 _ZN2at6native29vectorized_elementwise_kernelILi4ENS0_1
│  ├─ 0.010 _ZN2at6native29vectorized_elementwise_kernelILi4EZNS0_
│  ├─ 0.015 _ZN2at6native29vectorized_elementwise_kernelILi4EZNS0_
│  └─ 0.010 _ZN2at6native29vectorized_elementwise_kernelILi4EZNS0_
├─ 0.000 init
│  ├─ 0.000 _ZN2at6native29vectorized_elementwise_kernelILi4EZZZNS
│  └─ 0.000 _ZN50_GLOBAL__N__fa15d16e_17_RangeFactories_cu_38772b0
├─ 0.029 loss
│  ├─ 0.019 _ZN2at6native13reduce_kernelILi512ELi1ENS0_8ReduceOpI
│  └─ 0.010 _ZN2at6native29vectorized_elementwise_kernelILi4EZZZNS
└─ 0.042 optimizer
   ├─ 0.028 _ZN2at6native55_GLOBAL__N__d25d856e_22_ForeachBinaryOp
   └─ 0.014 _ZN2at6native57_GLOBAL__N__e658eeb9_24_ForeachBinaryOp
```

Custom Context

Design


LD_AUDIT
Annotation
Unwinding
Command Line
Instrumentation
Commands
Frames/Controls
EasyView
pprof
Library
json
Hathect
json
Kernel Metrics
Interface
Chrome Trace
CUPTI
ROCTracer

Inside the Library



Aggregate timing into kernels with the same “group name”

min/max/mean/stddev

User Interface

- Lightweight instrumentation
  - Profile start/stop/finalize (*torchinductor compatible*)
  - Scopes
  - Metrics
  - Renaming
  - Hooks
- Command line (*torchinductor compatible*)

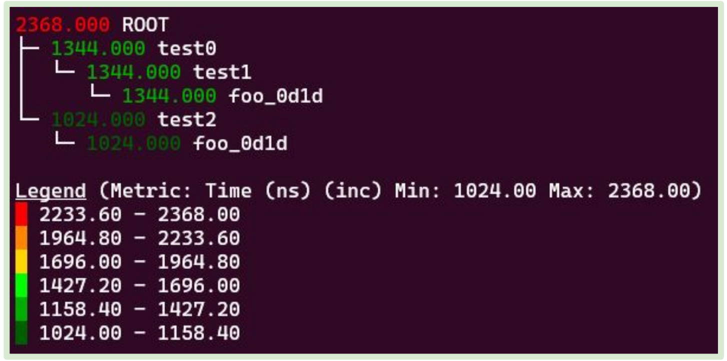
Profile Start/Stop/Finalize

- Profile only interesting regions
  - **proton.start**(name: str, *, backend: str = "cupti", context: str = "shadow", data: str = "tree", hook: Optional[str|callable] = None) -> *session_id*: int
  - **proton.finalize**()
- Skip some regions, but accumulate to the same profile
  - **proton.start(...)**
  - **proton.deactive**(*session_id*)
  - … # region skipped
  - **proton.activate**(*session_id*)
- Profile with multiple concurrent sessions
  - Different views (e.g., *tree*, *trace*, …)

Scopes

- Only collect the “Master Thread” scope
    - In PyTorch, the thread that train and test models

```
with proton.scope("test0"):
    with proton.scope("test1"):
        foo[1,](x, y)
with proton.scope("test2"):
    foo[1,](x, y)
```



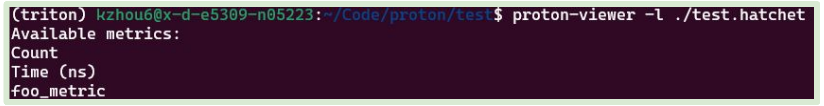
```
2368.000 ROOT
├─ 1344.000 test0
│  └─ 1344.000 test1
│     └─ 1344.000 foo_0d1d
└─ 1024.000 test2
   └─ 1024.000 foo_0d1d

Legend (Metric: Time (ns) (inc) Min: 1024.00 Max: 2368.00)
🟥 2233.60 - 2368.00
🟧 1964.80 - 2233.60
🟨 1696.00 - 1964.80
🟩 1427.20 - 1696.00
🟩 1158.40 - 1427.20
🟩 1024.00 - 1158.40
```

Metrics

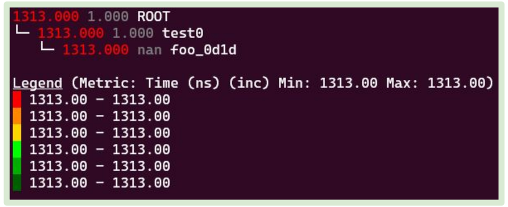
- Asynchronous metrics
  - Come from profilers
- Synchronous metrics
  - Come from users
    - Theoretical flops, bytes
    - Loss
    - Counts
  - `Dict[str, Union[int, float]]`

```
with proton.scope("test0", {"foo_metric": 1.0}):
    foo[1,](x, y)
```

"test0" scope ends with multiple metrics.
Two metrics can be display the same time.



```
(triton) kzhou6@x-d-e5309-n05223:~/Code/proton/test$ proton-viewer -l ./test.hatchet
Available metrics:
Count
Time (ns)
foo_metric
```



```
1313.000 1.000 ROOT
└─ 1313.000 1.000 test0
   └─ 1313.000 nan foo_0d1d

Legend (Metric: Time (ns) (inc) Min: 1313.00 Max: 1313.00)
🟥 1313.00 - 1313.00
🟧 1313.00 - 1313.00
🟨 1313.00 - 1313.00
🟩 1313.00 - 1313.00
🟩 1313.00 - 1313.00
🟩 1313.00 - 1313.00
```

Renaming

- Rename the triton function with a custom name
  - Append launch configurations
  - Append runtime dynamic
  - Append constants
  - e.g., `foo_<num_warps:4>_<fast_math:4>_<branch_0:1>`
- Can be used together with flexible metrics

```
with proton.Rename(foo_rename_fn):
    with proton.Metrics(foo_metric_fn):
        foo[1,](x, y, num_warps=4)
```

Hooks

- Decorators that are less intrusive

```
@proton.metrics(metrics_fn)
@proton.rename(rename_fn)
@triton.jit
def triton_fn():
    …
```

```
def metric_fn(grid_x, grid_y, grid_z,
      num_warps, num_ctas,
      cluster_x, cluster_y, cluster_z,
      shared_memory, stream, function, metadata,
      *args) -> Dict[str, Union[int, float]]
```

```
def rename_fn(grid_x, grid_y, grid_z,
      num_warps, num_ctas,
      cluster_x, cluster_y, cluster_z,
      shared_memory, stream, function, metadata,
      *args) -> str
```

Plan

- Integrate into triton
  - third_party/proton
- AMD GPUs
  - ROCTracer
- Command line interface
- Fine-grained metrics
  - Instruction samples
  - Binary instrumentation-based metrics
  - …
- VSCode Integration