PYTORCH CONFERENCE

# TorchBench: quantifying PyTorch performance during the development loop

Holding the line of performance and boosting optimizations for the PyTorch ecosystem.

Xu Zhao, Will Constable, David Berard, Taylor Robie, Eric Han, Adnan Aziz

PyTorch Perf Infra Team

Meta Platforms Inc.

F3

## Abstract

Goal: Quantifying PyTorch performance during the development loop.

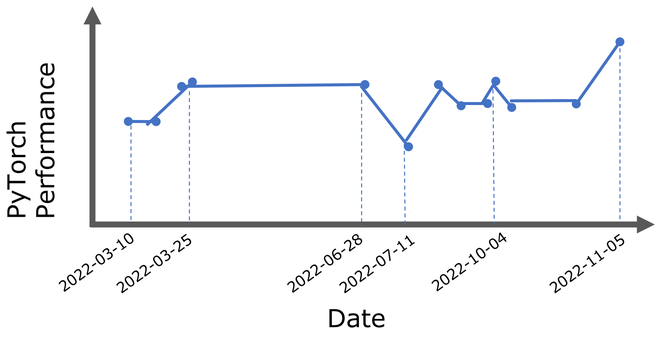



Fig. 1: Quantification of PyTorch performance in 2022.

Specifically, we answer the following questions:

- How do we know a GitHub PR speeds up or slows down PyTorch?
- How do we know an update in vendor libraries (e.g., CUDA) affect PyTorch performance?
- How do we find out the commit that caused PyTorch performance regression?

## Challenges

**Fast response** Each test must finish in seconds instead of minutes or hours.

**Coverage** Need to cover both train and inference across a wide range of PyTorch models and devices.

**Stableness** Need to tune the benchmark environment to achieve stable results.

## Technologies

**Slicing model code** We run one iteration per train/inference test with data prefetched to the device to achieve fast Time-To-Signal.

**Tuning performance on public cloud** We carefully tune the AWS/GCP hardware to stabilize performance results.

**Bisection** We implement a bisector to automatically find out which commit caused the performance signal, including both optimization and regression.

## Performance Signals Detected

| Date | Root Cause | Reason |
|---|---|---|
| 2022-03-25 | builder#1000 | Speedup: cuDNN upgrade |
| 2022-07-11 | pytorch#65839 | Regression: additional runtime check |
| 2022-10-04 | pytorch#84948 | Speedup: enable new cuDNN front-end API |

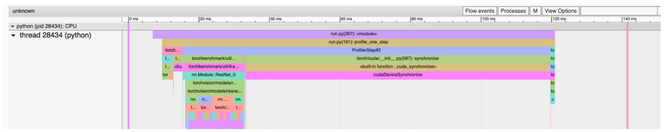
## Boosting PyTorch backend development

### Simple interface to add a new backend

```
1 # Example: torchscript backend
2 @create_backend
3 def torchscript(model: BenchmarkModel, args: List[str]):
4     module, example_input = model.get_module()
5     model.set_module(torch.jit.script(module, example_inputs))
```

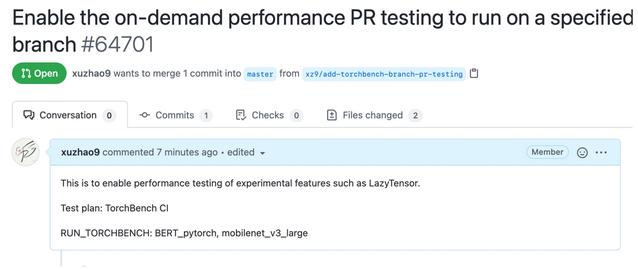
### Profile and run the backend

```
$ python run.py resnet50 --backend torchscript --profile
```



## Benchmarking-as-a-service

**PR-level benchmarking** With a magic word "RUN_TORCHBENCH" in PR, users can choose to benchmark their PR and visualize results at PyTorch HUD.


Enable the on-demand performance PR testing to run on a specified branch #64701

Open xuzhao9 wants to merge 1 commit into master from xz9/add-torchbench-branch-pr-testing

Conversation 0 · Commits 1 · Checks 0 · Files changed 2

xuzhao9 commented 7 minutes ago • edited — Member

This is to enable performance testing of experimental features such as LazyTensor.

Test plan: TorchBench CI

RUN_TORCHBENCH: BERT_pytorch, mobilenet_v3_large

| Metric | 111fe61602 | 7bc72f5a2f | Speedup |
|---|---|---|---|
| nnc-dynamic:autogen-0 | 0.14310094044776633 | 0.14308511896524578 | 0.01% |
| nnc-dynamic:autogen-1 | 0.11164433404337615 | 0.11165716196410358 | -0.01% |
| nnc-dynamic:autogen-10 | 0.017939746397314594 | 0.017773296852828933 | 0.94% |
| nnc-dynamic:autogen-11 | 0.02166838520206511 | 0.021501831093337385 | 0.77% |
| nnc-dynamic:autogen-12 | 0.12938609847333285 | 0.12939167249714956 | -0.00% |
| nnc-dynamic:autogen-13 | 1.8119537853635848 | 1.8118110403884202 | 0.01% |
| nnc-dynamic:autogen-14 | 7.227453680243343 | 7.228049130644649 | -0.01% |

**User customized benchmark** Sometimes users want customized performance metrics or simply microbenchmarks, we support this through user-defined benchmark scripts. GPU memory leak and utilization signals are working-in-progress.

**Nightly and Pre-release testing** TorchBench supports both Nightly and Pre-release testing CIs to catch performance bugs before production.

#### Acknowledgements

This research is from the PyTorch Performance Infra team at Meta Platforms Inc.:

- Xu Zhao (Research Scientist)
- Will Constable (Software Engineer)
- David Berard (Software Engineer)
- Taylor Robie (Software Engineer)
- Eric Han (Software Engineer)
- Adnan Aziz (Research Scientist Manager)

Take a photo to learn more: