ECE408 / CS483 / CSE 408 Final Project Kickoff

Carl Pearson

pearson@illinois.edu

Outline

Fast convolution layer forward pass in MxNet

Timeline (1/2)

Wed 3/07/2018: Milestone 1

Make sure `rai` works for you

Wed 3/21/2018: Milestone 2

CPU Implementation

Wed 4/18/2018: Milestone 3

GPU implementation and performance preview

Timeline (2/2)

Wed 5/02/2018: Milestone 4

GPU optimizations

Fri 5/11/2018: Final Submission

Project Release

Project Landing Page

https://github.com/illinois-impact/ece408_project

Includes

- Instructions
- RAI Client
- Final Project Rubric
- Other Guidelines
- Skeleton Code

rai Client

The client may be downloaded from the project readme

You will need your `.rai_profile` file in your home directory.

- `~/.rai_profile` on Linux/macOS
- `%HOME%/.rai_profile` on Windows

This should have been emailed to you.

(demo)

Teams

Team of 3 People

Submit your team name to this form

Modify `.rai_profile` with your team name under `team.name`:

```
profile:
  firstname: Carl
  lastname: Pearson
  username: carlwpearson
  email: carlwpearson@gmail.com
  access_key: auth0|5a0130d32327ea70420b71ef
  secret_key: <snip>
  affiliation: uiuc
  team:
    name: staff
```

Basic `rai`

`rai -p <project folder>` uploads your folder to AWS.

Your code in `rai_build.yml` is executed on AWS in a specific docker container.

The results are streamed back to you in *real time*.

Milestone 1 (Wed 3/07/2018 @ 5pm)

- Make sure rai is working for you
- Choose your teams / team name

Run MxNet baseline CPU code

- Make sure MxNet is working.
- Report execution time.

Run MxNet baseline GPU code

- Make sure MxNet GPU is working
- Report execution time.

Show your stuff in `report.pdf`

- please make this a pdf file

Submitting

```
rai -p <project folder> --submit=[m1,m2,m3,m4,final]

* Enforces a particular rai_build.yml
* Records timing information
* You will need a `report.pdf`.
```

Your code should work correctly here, this is what matters.

We'll look at the latest one before the deadline.

Milestone 2 (Wed 3/21/2018 @ 5pm)

Turn in an updated `report.pdf`

Implement CPU forward pass

- Make sure you can compile/run MxNet CPU code.
- Execution time

Should be pretty straightforward copy from slides / chapter 16

Intro to `nvprof`

- Use nvprof to collect some basic info

Milestone 3 (Wed 4/18/2017 @ 5pm)

Turn in an updated draft of `report.pdf`.

Implement GPU forward pass

- Make sure you can compile/run MxNet GPU code.
- Execution time, profile

Doesn't have to be fast, but it should work. Small changes to milestone 2.

Milestone 4 (Wed 5/2/2017)

Updated `report.pdf`

Implement 3 GPU optimizations

Use `nvprof` / NVVP to describe effect of optimizations

Final Submission (Fri 5/11/2017)

- The **real deal**.

Optimize that GPU convolution.

Turn in a final report.

Rubric

1. Milestone 1 (5%)
2. Milestone 2 (10%)
3. Milestone 3 (10%)
    - Optimization 1
4. Final Optimizations (60%)
    - Optimization 2 (10%)
    - Optimization 3 (10%)
    - Optimization 4 (10%)
    - Optimization 5 (10%)
    - Optimization 6 (10%)
    - Additional Optimizations / detailed insights (10%)
5. Performance Ranking (10%)
6. Report Style (10 %)
    - Clear, concise writing, good layout, and good organization will be rewarded.

Optimizations Rubric

Each optimization will be graded as follows:

```
* Explanation of Performance Impact ( 40% )
* Correctness ( 60% )
```

Ranking Rubric

Each optimization will be graded as follows:

1. The median performance will be determined (how well the class did as a whole)
2. Your performance will be converted to a number of standard deviations above/below that median (how well you did compared to the class).
3. That value will be linearly mapped into the space of 0-10 to determine the ranking grade.

Final report

See project page for up-to-date rubric.

You've been building this final report through all the milestones. Keep the content from the earlier milestones, but be sure to include the following:

- Your team name
- Your team member names
- Your netids
- Your UINs

For each optimization

1. Optimization Approach and Results
    - how you identified the optimization opportunity
    - why you thought the approach would be fruitful
    - the effect of the optimization. was it fruitful, and why or why not. Use nvprof and NVVP to justify your explanation.
    - Any external references used during identification or development of the optimization
    - How your team organized and divided up this work.
2. References (as needed)
3. (Optional) Suggestions for Improving Next Year

Comparing against your peers

```
rai rankings
```

Shows *anonymized* performance results for you and other teams.

`rai` tips and tricks

`rai -p version

Prints the date `rai` was built

Check piazza, if your `rai` is old, download a newer one

`rai -p -d -v

Debug / verbose mode. Prints a bit of additional info while running

- the queue submitted to (should be `rai_amd64`)
- what it thinks your username and team name is

Notes on `batch_size`

Don't modify the `batch_size` in the python script.

You can loop over `B` in the C/CUDA code and split it up there however you want.