# Tutorial 3a: Roofline and exercise workflow

## Informatik elective: GPU Computing

Pratik Nayak

## Roofline model: Components

- In Lecture 2, we looked at the roofline model. This can be a useful tool to understand the performance of your application.
- There are four main components to plotting a roofline:
    - Measuring the peak FLOP/s of the machine. Please see lecture 2 slides.
    - Measuring peak bandwidth (Next slide)
    - Estimating the arithmetic intensity. Please see lecture 2 slides.
    - Measuring the application performance.

## Roofline model: Measuring peak BW

- Measuring peak bandwidth:
  - For example, Linux-based OS: `sudo dmidecode | grep -A 5 “Configured Memory”` should give you a value in MT/s and you can get the BW (in B/s) by multiplying that value by 8.
  - For Windows, you can go to Task Manager → Performance → Memory and see the speed in MHz there. If you have a DDRAM (double data rate …), then you multiply this by 2 to get the MT/s and then by 8 to the bandwidth in B/s.

## Roofline model: Measuring application performance

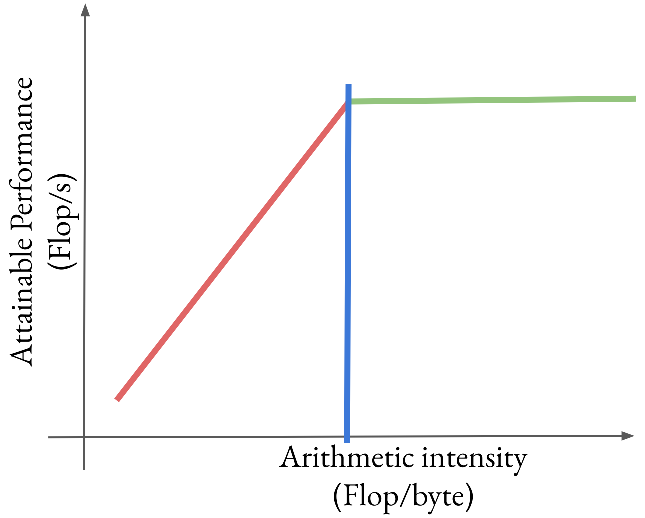
- Measuring application performance:
  - Measure only the runtime of the operations.
  - Ignore data-allocations and deallocations and other auxiliary function calls.
  - Average measurements over a few runs to reduce noise.

## Roofline model: Plotting the roofline

- The roofline consists of: The compute peak (in green) and the bandwidth peak (in red)
- The machine balance (Peak perf / Peak BW) is denoted by the vertical line (in blue)
- The roofline (green+red lines) is plotted using the following formula:

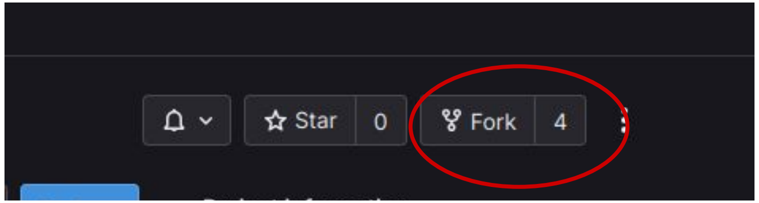
```
    min (peak Flop/s , AI*(peak BW))
```

- For each operation, you then plot the obtained performance (work complexity/time) at `x` = AI of that operation


Attainable Performance (Flop/s)
Arithmetic intensity (Flop/byte)

## Recommended exercises workflow

- To ensure easy and smooth working on the exercises, I recommend you to:
  - Fork the main repo (https://gitlab.lrz.de/2024ws-gpu-computing/exercises)



  - This should create your own copy of the main repo with the path: https://gitlab.lrz.de/<username>/exercises
  - In that repo, go to Manage → Members → Invite Members (on the top right of the page), and invite me (@pratikvn) to the repo with Developer permissions so that I can give feedback and grade your exercises.

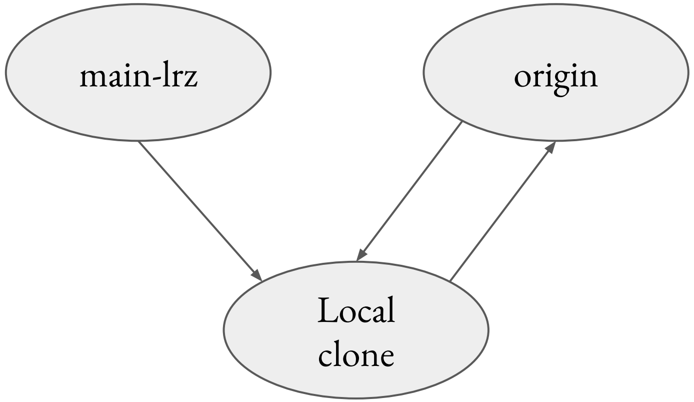
## Recommended exercises workflow

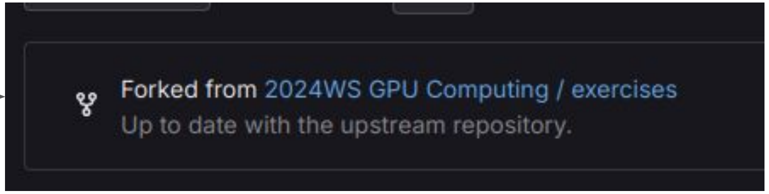
- Always clone (git clone) and work (git push) on your personal fork (https://gitlab.lrz.de/<username>/exercises)
- On your local machine (or the cluster) where you have cloned, you can then add the main repo as an additional remote with (https:// or **preferably with ssh if you have set that up**)
  - `git remote add main-lrz https://gitlab.lrz.de/2024ws-gpu-computing/exercises`
- Now running `git remote -v` on the command line should give you something like:

```
main-lrz     git@gitlab.lrz.de:2024ws-gpu-computing/exercises.git (fetch)
main-lrz     git@gitlab.lrz.de:2024ws-gpu-computing/exercises.git (push)
origin  git@gitlab.lrz.de:pratikvn/exercises.git (fetch)
origin  git@gitlab.lrz.de:pratikvn/exercises.git (push)
```

## Recommended exercises workflow

- `git remote -v` shows that there are two remotes now.
- The idea is to work always work on your local clone and push to your local fork (origin).
- When there are updates in main-lrz, you pull the changes from main-lrz into your local clone, rebase with `git rebase`
- And then you can push the rebased branch into your local fork, so that it is updated. You can see if your local fork is behind main-lrz as gitlab webpage shows it as below




Forked from 2024WS GPU Computing / exercises
Up to date with the upstream repository.

## Submission: Feedback and grading of exercises

- Please ensure that you have given me access to your local fork (see slide 1) so that I can grade and give feedback on your exercises.
- Generally there will be one week of work time for the exercises.
- Commit to your local fork frequently. Shows us that you have not just copied the code from somewhere.
- Once you have your exercise ready to submit, create an Issue on your fork: https://gitlab.lrz.de/<username>/exercises/-/issues , and ping me by username @pratikvn.
  - You can also upload your explanations, performance plots, pdfs there directly if necessary.
- Create an Issue for each exercise. (one Issue for ex1, a new Issue for ex2 and so on)

## Some useful documentation

- Rebasing (`git rebase` command) ensures that your commit history is linear. Here is some documentation to help you better understand how rebase works: https://git-scm.com/book/en/v2/Git-Branching-Rebasing
- I recommend that you get familiar with git by following through with this free hands-on book: https://git-scm.com/book/en/v2