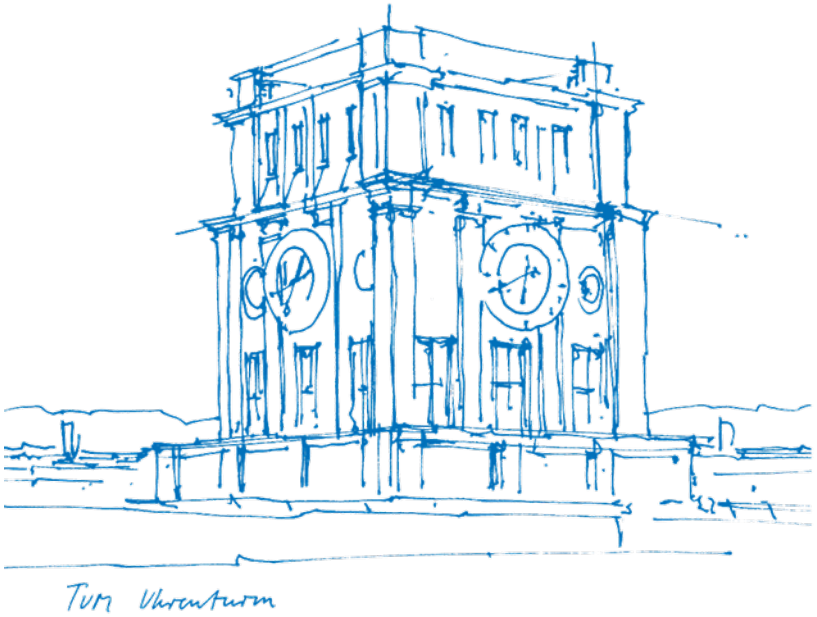
Parallel Programming Tutorial - Sequential Programming

Vincent Bode, M.Sc.
Chair for Computer Architecture and Parallel Systems (CAPS)
Technical University Munich
May 9, 2020



📢 Public Service Announcement

Homework

- You can still switch teams easily during this second exercise
  - If you want to switch afterwards, please contact us so you don't lose points.
- Please make sure your Gitlab repositories are set to private.

Assignment 2: “Mandelbrot”

Assignment: Mandelbrot

- Task: Calculate number of elements in Multibrot set.
  - Like Mandelbrot, but with variable exponent.
  - The exponent is read from `stdin`.
- Using formula $z_{i+1} = z_i^d + c$ where $z, c \in \mathbb{C}$.
- With $z_0 = 0 + 0i$, $c = x + yi$.
- If after `max_iter` iterations, $|z| < 2$, then c is in the set.

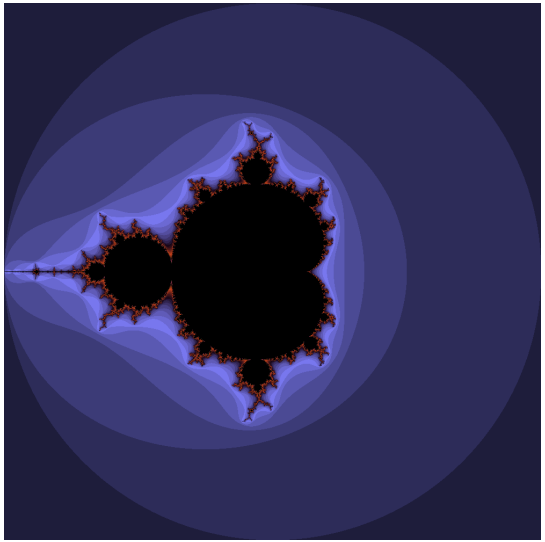


Figure 1: Standard Mandelbrot set ($d = 2.0$).

Assignment: Mandelbrot

- You can use all the resources available to you.
- Parallelize the sequential implementation with pthreads.
- Your speedup should be ≥ 16.
- Evaluation command: `./student_submission -r 1033x1033 -i 1000 -n 1`
- Image is output to PPM image file unless `-n 1` specified.
- Watch out for load imbalance.
- More pthread exercise content next week can help get high speedups.

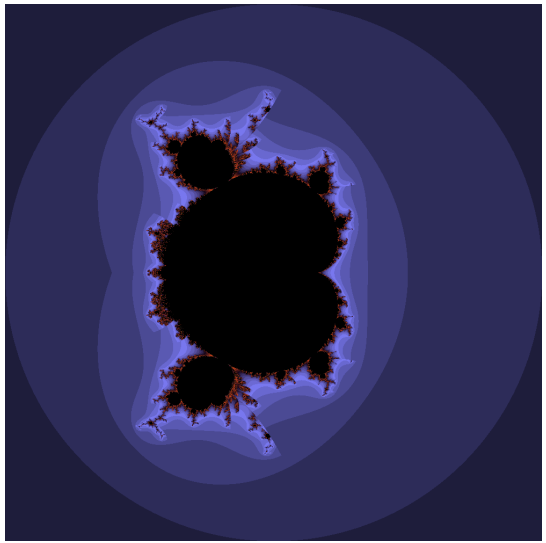


Figure 2: Funky Multibrot set ($d = 2.5$).

Hints on the submission system

- You can see speedup even on failed submissions.
- Free computing time before `getProblemFromInput()` and after `outputSolution(result)`.
- The reference implementation is no longer the sequential implementation.
- Nothing changes on the submission server except the data provided from `getProblemFromInput()`.

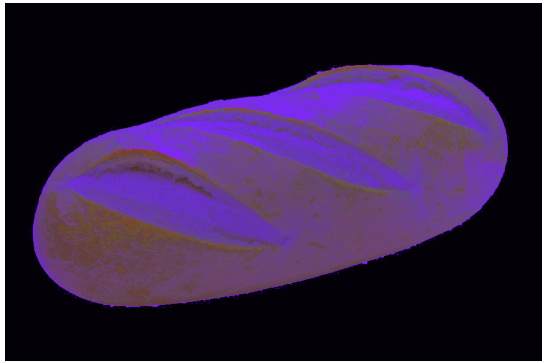


Figure 3: Mysterious Megabrot set ($d = ?$).

Assignment: Megabrot

Good luck on the homework.

Contact us on Moodle, RocketChat, etc.
Q&A Session will be announced.