



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

High-Performance Computing Lab for CSE

2024

Student: Yannick Ramic

Discussed with: FULL NAME

Solution for Project 2

Due date: 25 March 2024, 23:59

HPC Lab for CSE 2024 — Submission Instructions
(Please, notice that following instructions are mandatory:
submissions that don't comply with, won't be considered)

- Assignments must be submitted to Moodle (i.e. in electronic format).
- Provide both executable package and sources (e.g. C/C++ files, Matlab). If you are using libraries, please add them in the file. Sources must be organized in directories called:
Project_number_lastname_firstname
and the file must be called:
project_number_lastname_firstname.zip
project_number_lastname_firstname.pdf
- The TAs will grade your project by reviewing your project write-up, and looking at the implementation you attempted, and benchmarking your code's performance.
- You are allowed to discuss all questions with anyone you like; however: (i) your submission must list anyone you discussed problems with and (ii) you must write up your submission independently.

1. Computing π with OpenMP [20 points]

Strong Scaling Analysis:

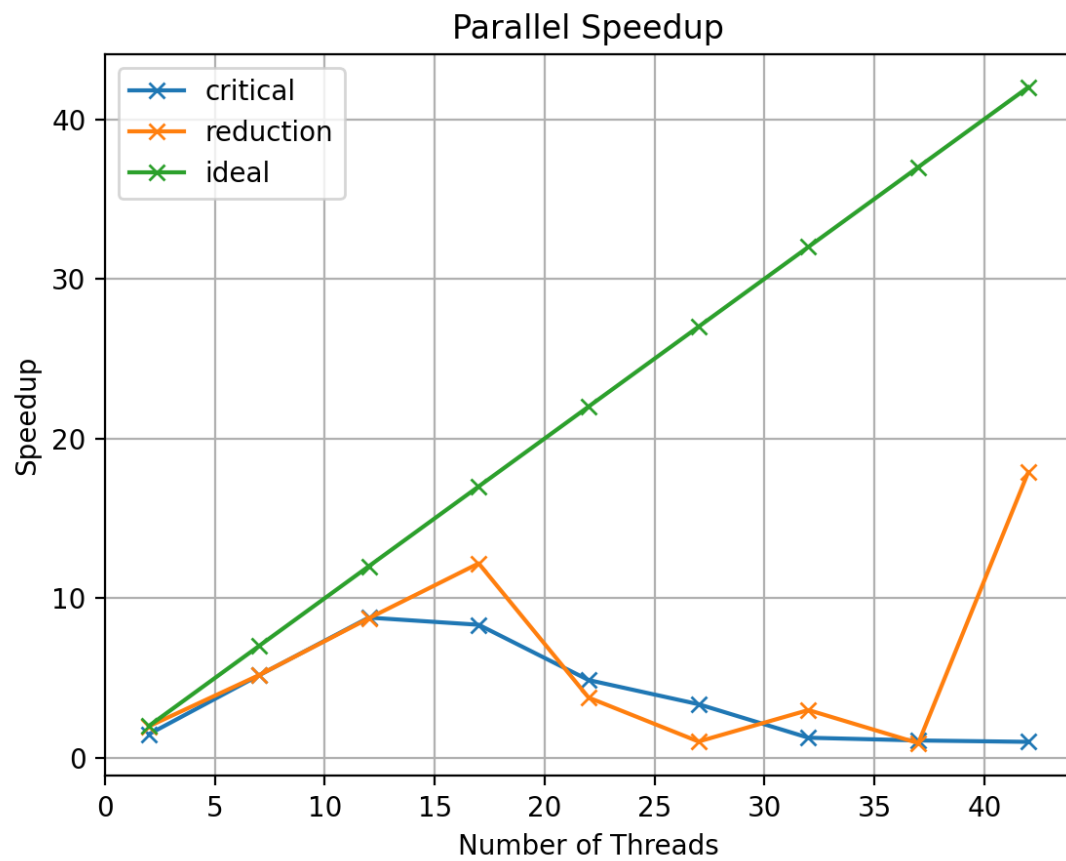


Figure 1: Parallel Speedup

Weak Scaling Analysis:

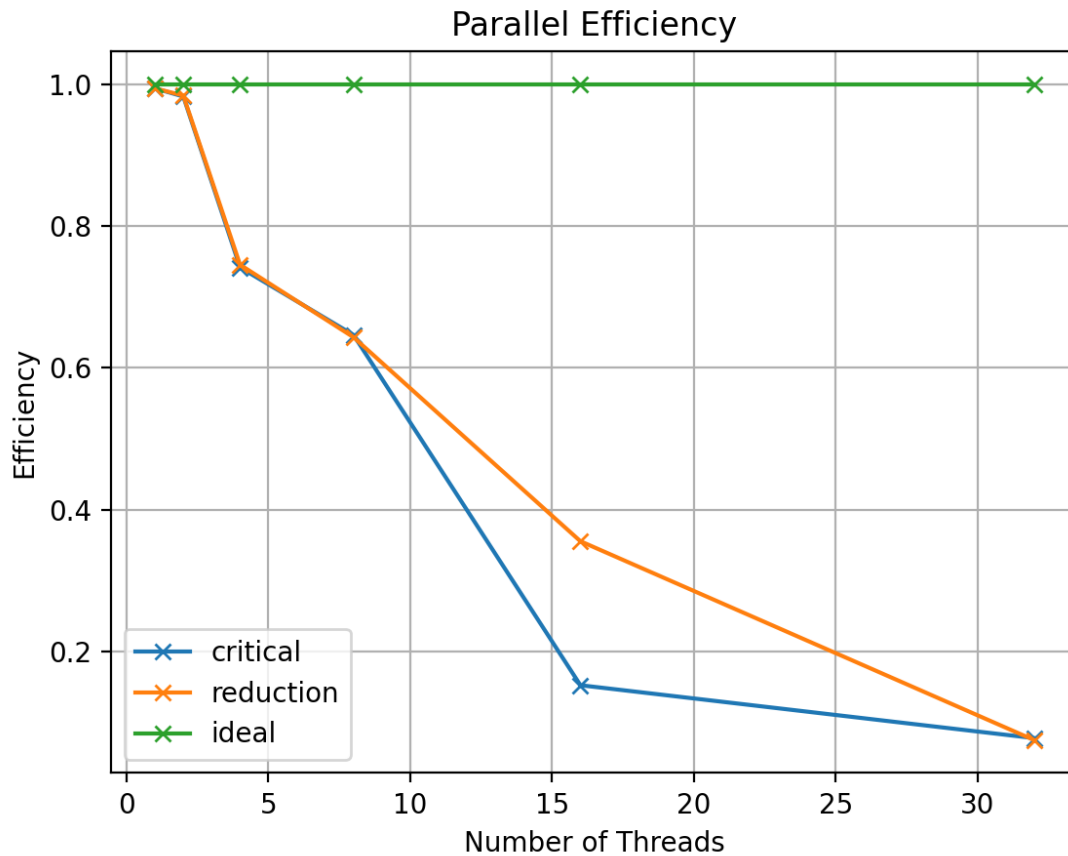


Figure 2: Parallel Efficiency

2. The Mandelbrot set using OpenMP [20 points]
3. Bug hunt [10 points]
4. Parallel histogram calculation using OpenMP [15 points]
5. Parallel loop dependencies with OpenMP [15 points]

```
for (n = 0; n <= N; ++n) {
    opt[n] = Sn;
    Sn *= up;
}
```

Figure 3: Loop Dependencies Problem Description

```
#pragma omp parallel shared(opt) private(n)
{
#pragma omp for firstprivate(lastn) lastprivate(Sn)
  for (n = 0; n <= N; ++n) {
    if (lastn == n - 1) {
      // Use the fast version!
      Sn *= up;
    } else {
      // Use the slow version!
      // Note that S0 = up!
      Sn = up * pow(up, n);
    }
    opt[n] = Sn;
    // Update lastn!
    lastn = n;
  }
} // End OMP
```

Figure 4: Parallelized Code Snippet

6. Quicksort using OpenMP tasks [20 points]