Performance Evaluation - Quicksort

Cyril Six

January 2016

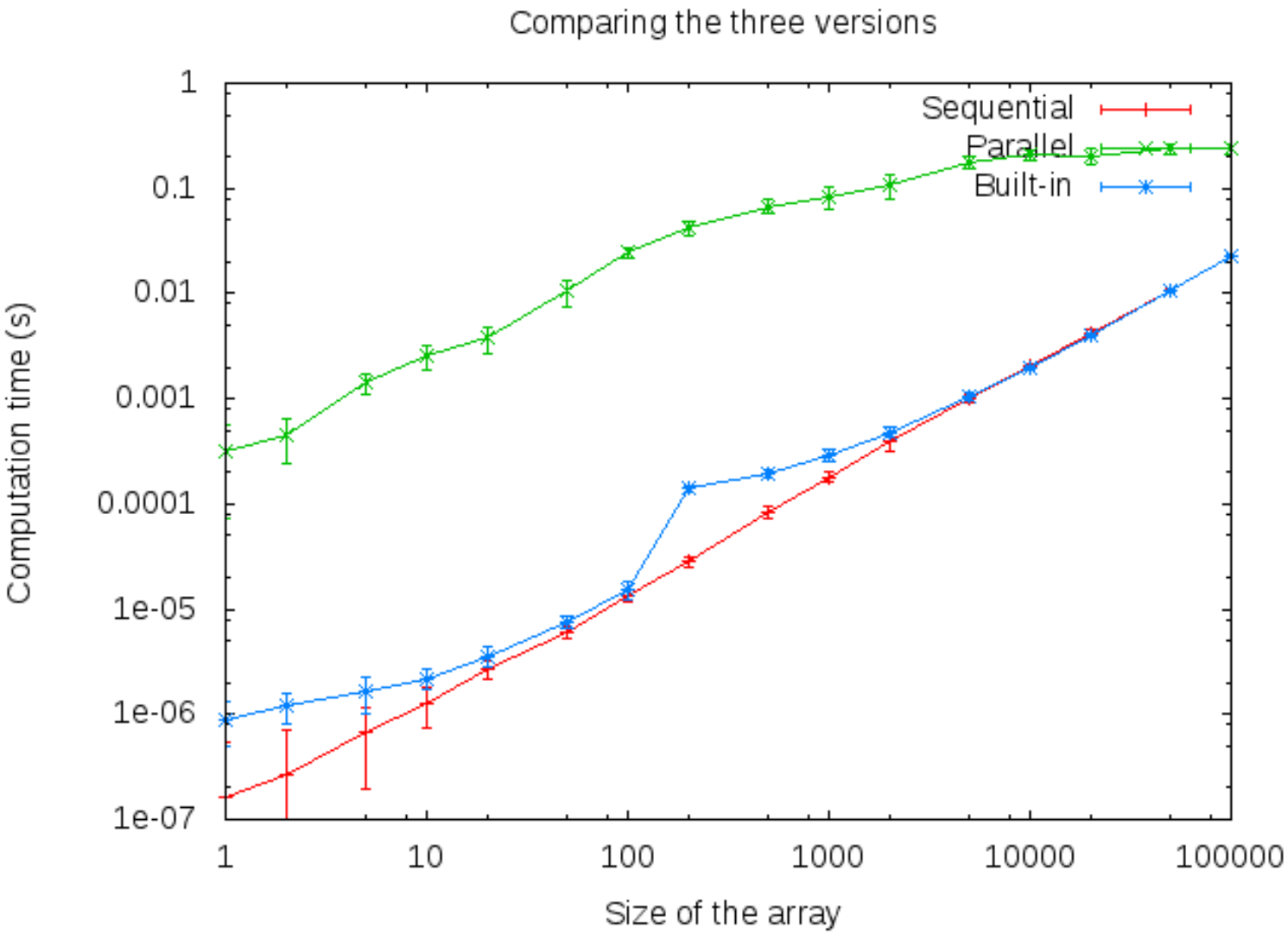
Organization

- Practical changes in the main program
  - Execute only sequential, parallel or built-in version
  - *THREAD_LEVEL* can be defined during the compilation
- Change in the Makefile
  - Possibility to compile with different optimization and thread level

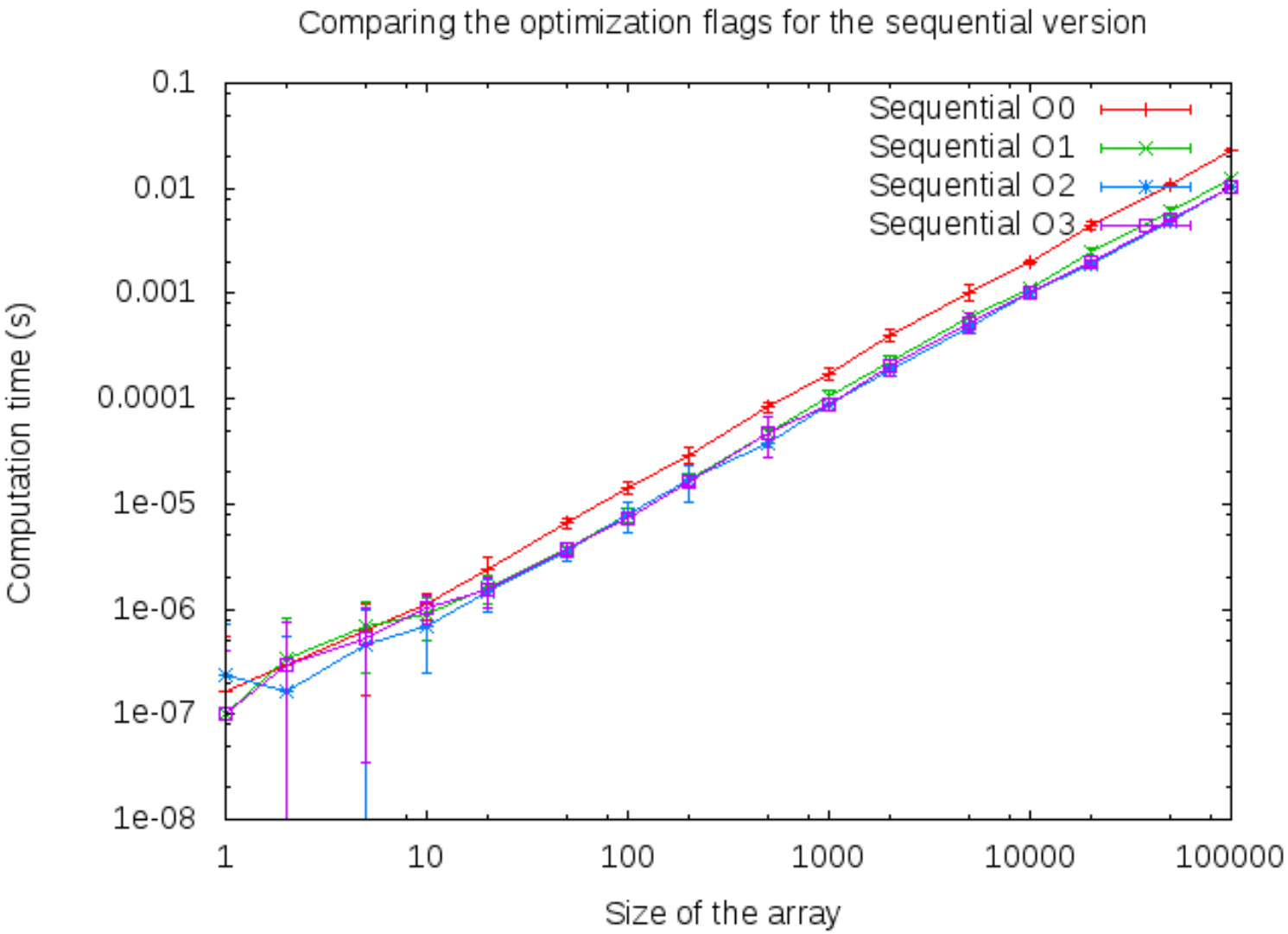
Organization (2)

- Several bash scripts to extract data from several executions
  - For each size, do 30 runs, and compute the average and standard deviation
  - The sizes are chosen in a logarithmic and deterministic scale
- Gnuplot scripts, writing to a timestamped PNG file
  - Using the standard deviation to plot error bars

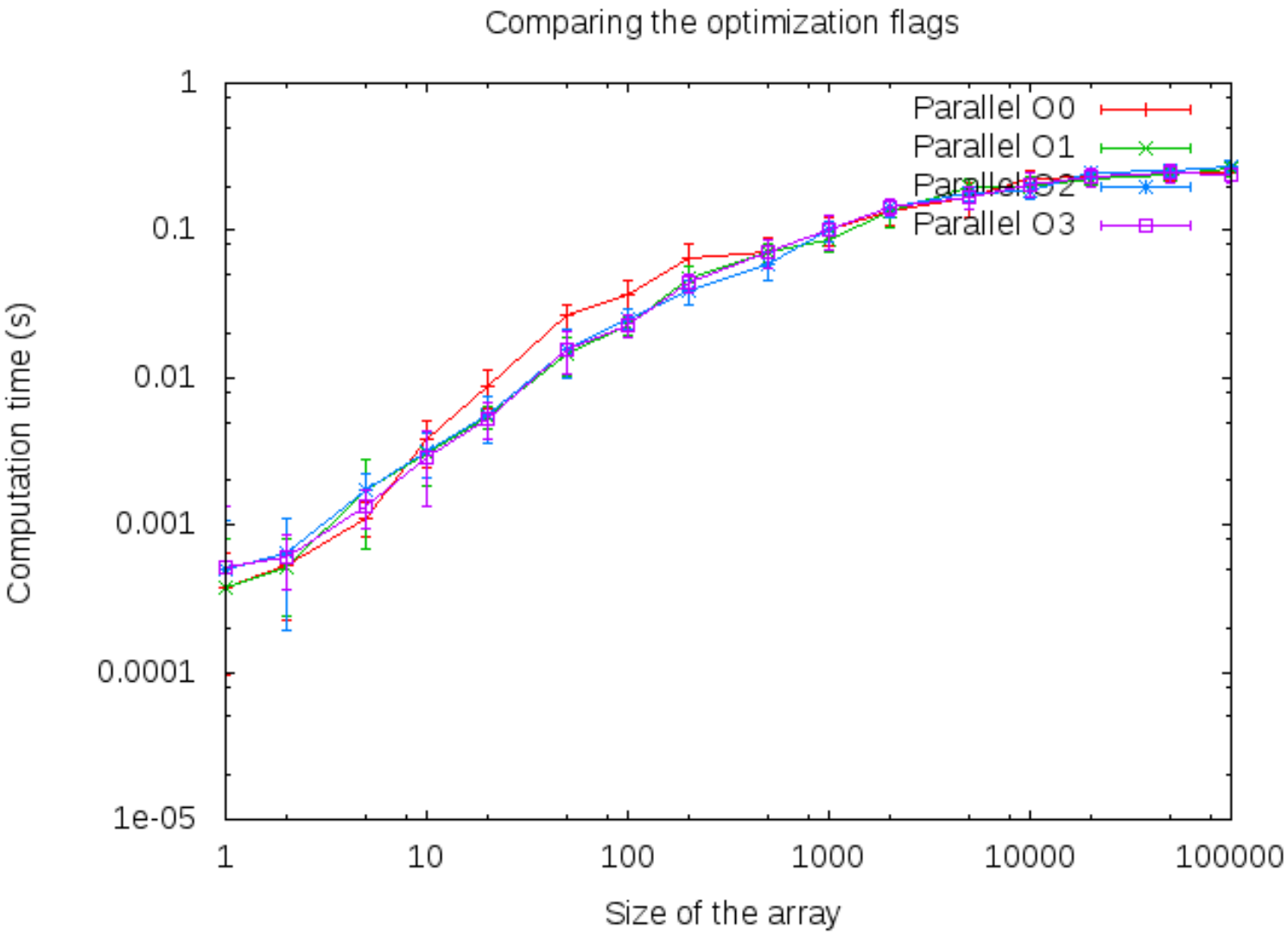
Comparing the different versions



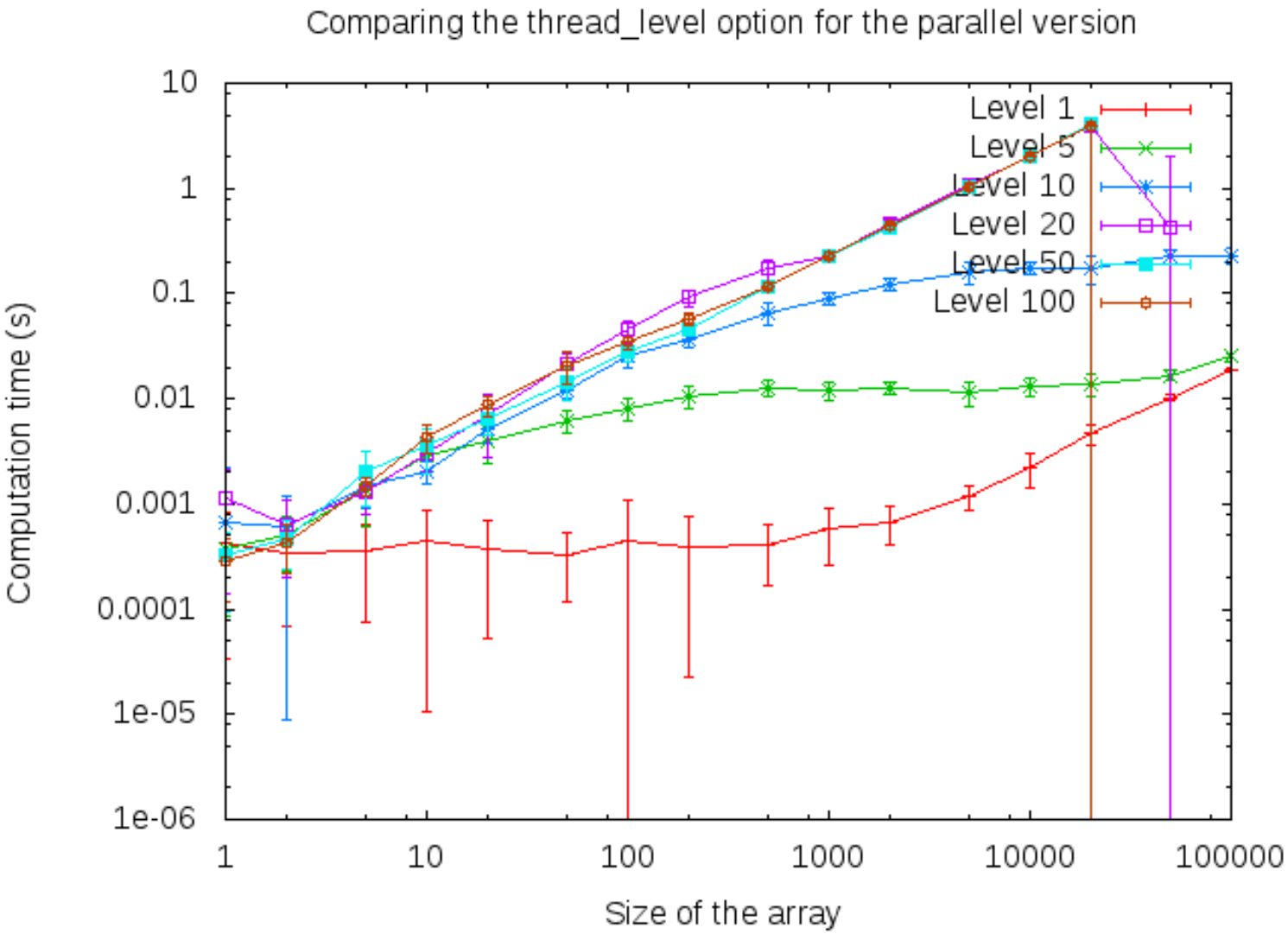
Comparing different optimization levels



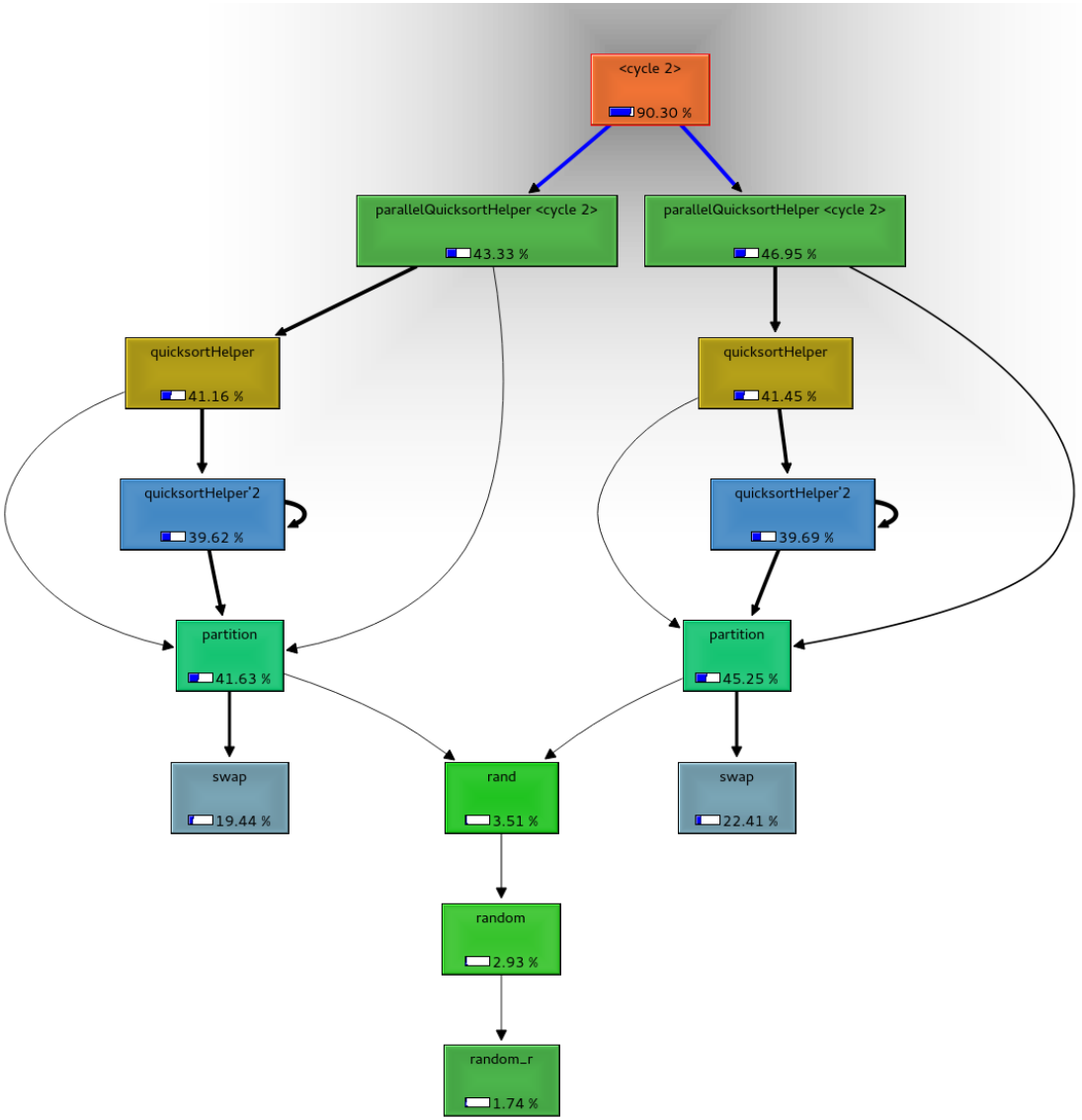
Comparing different optimization levels



Comparing different thread level values



Running callgrind with thread_level=2



My guess

- There seems to be nothing wrong in the sequential code
- What impedes the performance might have to do with scheduling
- Number of threads ? Context switches ?