TECHNISCHE
UNIVERSITÄT
DRESDEN

Center for Information Services and High Performance Computing (ZIH)

The Heat Example

Linux/x86 Performance Practical, 17.06.2009

Zellescher Weg 12
Willers-Bau A106
Tel. +49 351 - 463 - 31945

Matthias Lieber (matthias.lieber@tu-dresden.de)
Tobias Hilbrich (tobias.hilbrich@zih.tu-dresden.de)

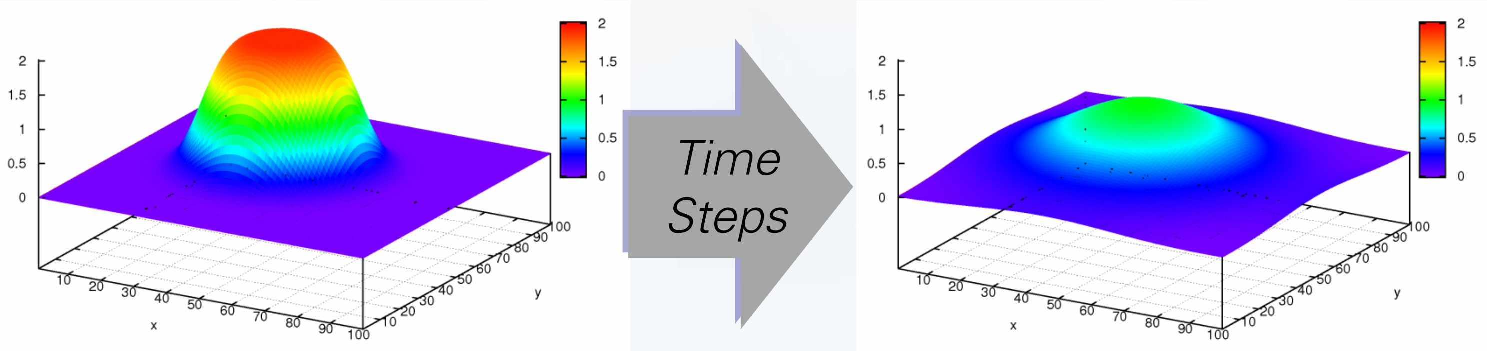
ZIH
Center for Information Services &
High Performance Computing

Heat Equation

- Heat equation solver as debugging example
- Heat equation describes heat distribution in a region over time
- Equation (2D): $\dfrac{\partial u}{\partial t} = k\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}\right)$
- Example implements solution on a 2D grid
- Time step for a grid cell (x,y):

$$\frac{\Delta u[x,y]}{\Delta t} = k\left(\frac{u[x-1,y]+u[x+1,y]-2u[x,y]}{\Delta x^2} + \frac{u[x,y-1]+u[x,y+1]-2u[x,y]}{\Delta y^2}\right)$$

- Visualization as 3D chart:



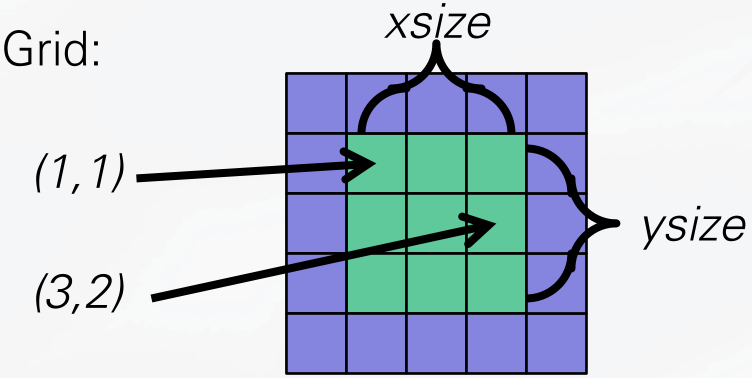
Time Steps

The Example Code

- Available in C and Fortran90
- Key object is the structure for the 2D grid
- Functions:
  - heatAllocate & heatDeallocate – *Creates/Frees the grid*
  - heatInitialize – *Sets the initial heat distribution*
  - heatPrint & heatOutput – *Prints the grid to stdout or a data file*
  - heatTimestep – *Calculates one timestep for the full grid*
  - heatBoundary – *Exchanges boundary data*
  - heatTotalEnergy – *Calculates overall energy amount*
  - Main function – *Contains main loop*

The Example Code - Grid

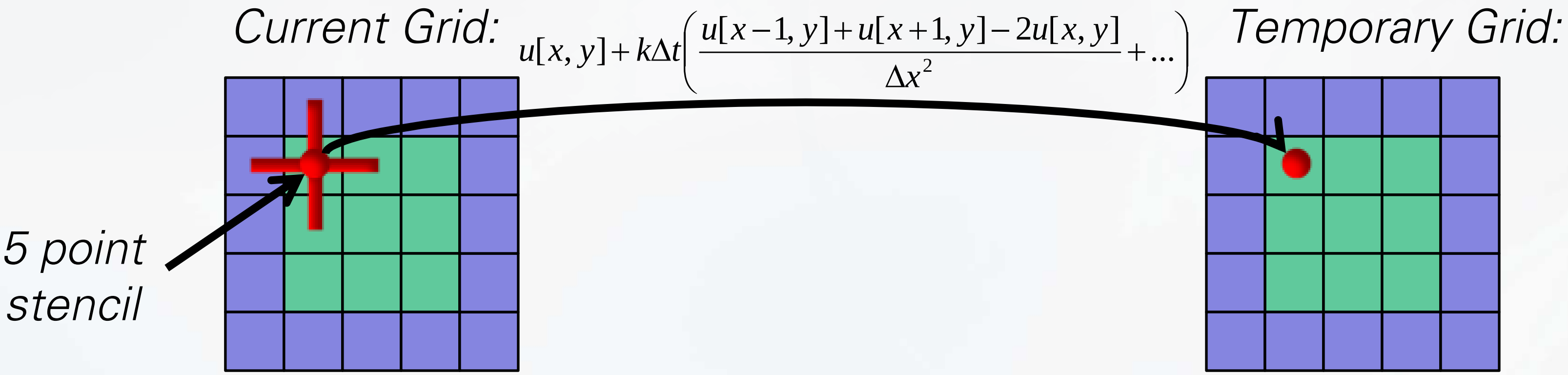
- Grid Structure contains 2 grids and the grid size
- Grid:



🟦 Ghost Cell (Boundary Cell)

🟩 Grid Cells

- Ghost cells used as neighbors, needed for border cells of actual grid
- Time steps are calculated for all actual grid cells, results stored in second grid



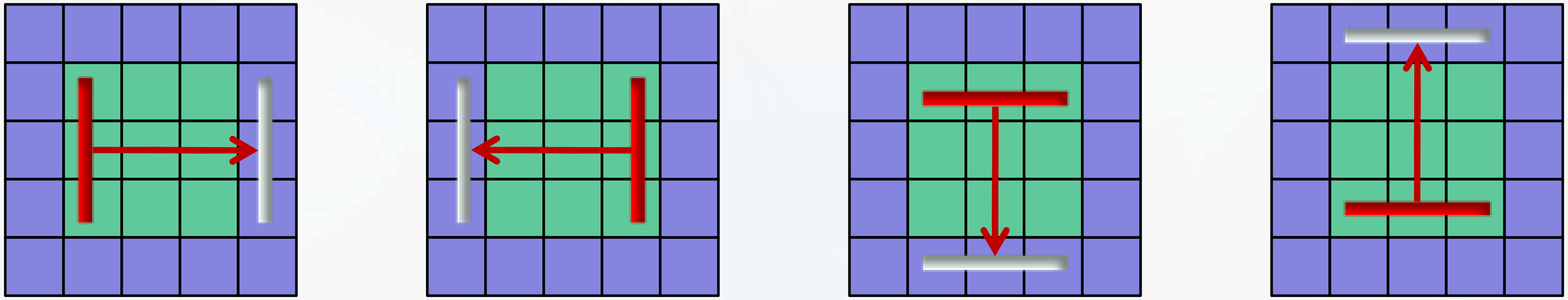
$$u[x,y]+k\Delta t\left(\frac{u[x-1,y]+u[x+1,y]-2u[x,y]}{\Delta x^2}+\dots\right)$$

- Grids switched after calculation for all grid cells

The Example Code - Ghost Cells

- 5 point stencil needs left, right, up, down neighbor for each grid point
- Ghost cells serve as these neighbors
- Their values can either be constant or calculated from the grid
- The code uses periodic ghost cells
- After each time step a copy operation is necessary to update the ghost cells
- Implemented in “heatBoundary”
- Update:

Grid:



(1) Left border

(2) Right border

(3) Top border

(4) Bottom border

The Example Code - Main Program

- Initialization of grid arrays
- Iterates a fixed number of time steps
  - Step computation and (heatTimestep)
  - Boundary exchange (heatBoundary)
- Time measurement around time stepping loop
- Prints total energy at start and end as energy conservation check

Exercise 1: Get some Output

- Choose programming language
  - C: **`cd ~/heat-c`**
  - Fortran90: **`cd ~/heat-f90`**
- Use the Makefile (edit **`CFLAGS`** / **`FFLAGS`** first) or call compiler at the shell
- Add **`-DSMALL`** to the compiler flags to run with a small grid and enable ASCII art output
- Run

Exercise 2: Use the Tools & Optimze

- Remove **`-DSMALL`** from the compiler flags
- Use the tools to find performance problems
  - Gprof
  - Callgrind / KCachegrind
  - PAPI counter
    - C: **`cd ~/heat-c-papi`**
    - Fortran90: **`cd ~/heat-f90-papi`**
- Try to optimize the code - ensure correct result!
  - Compiler flags
  - Code modifications

Exercise 2: Code Modification Hints

- Find most expensive code lines
- Check the loop order in these sections
- Is the copy from thetanew to theta really necessary?
- Are there any expensive math routines called?

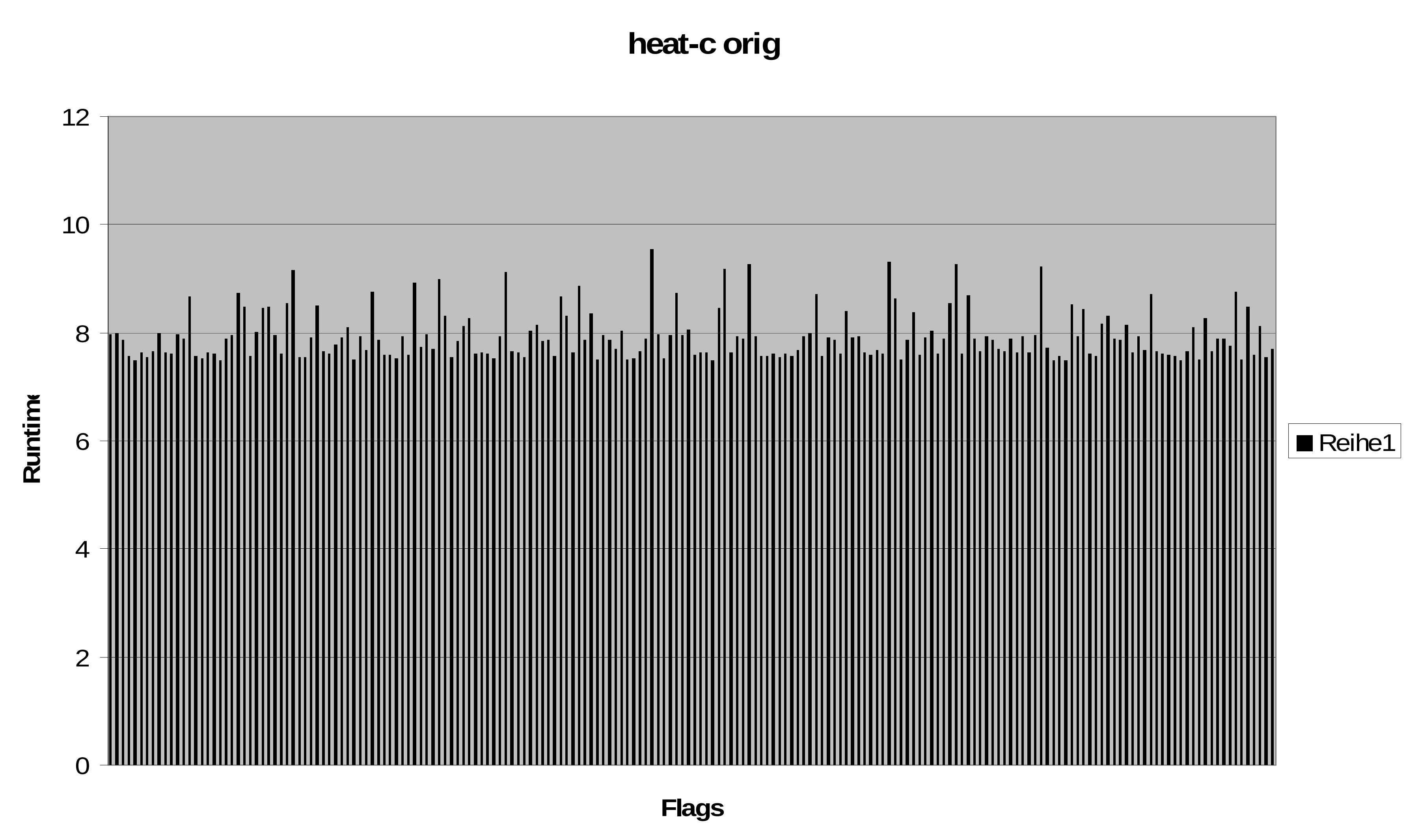
Exercise 2: Code Modifications

- Change loop order in **`heatTimestep`**
  - Gives huge speed-up

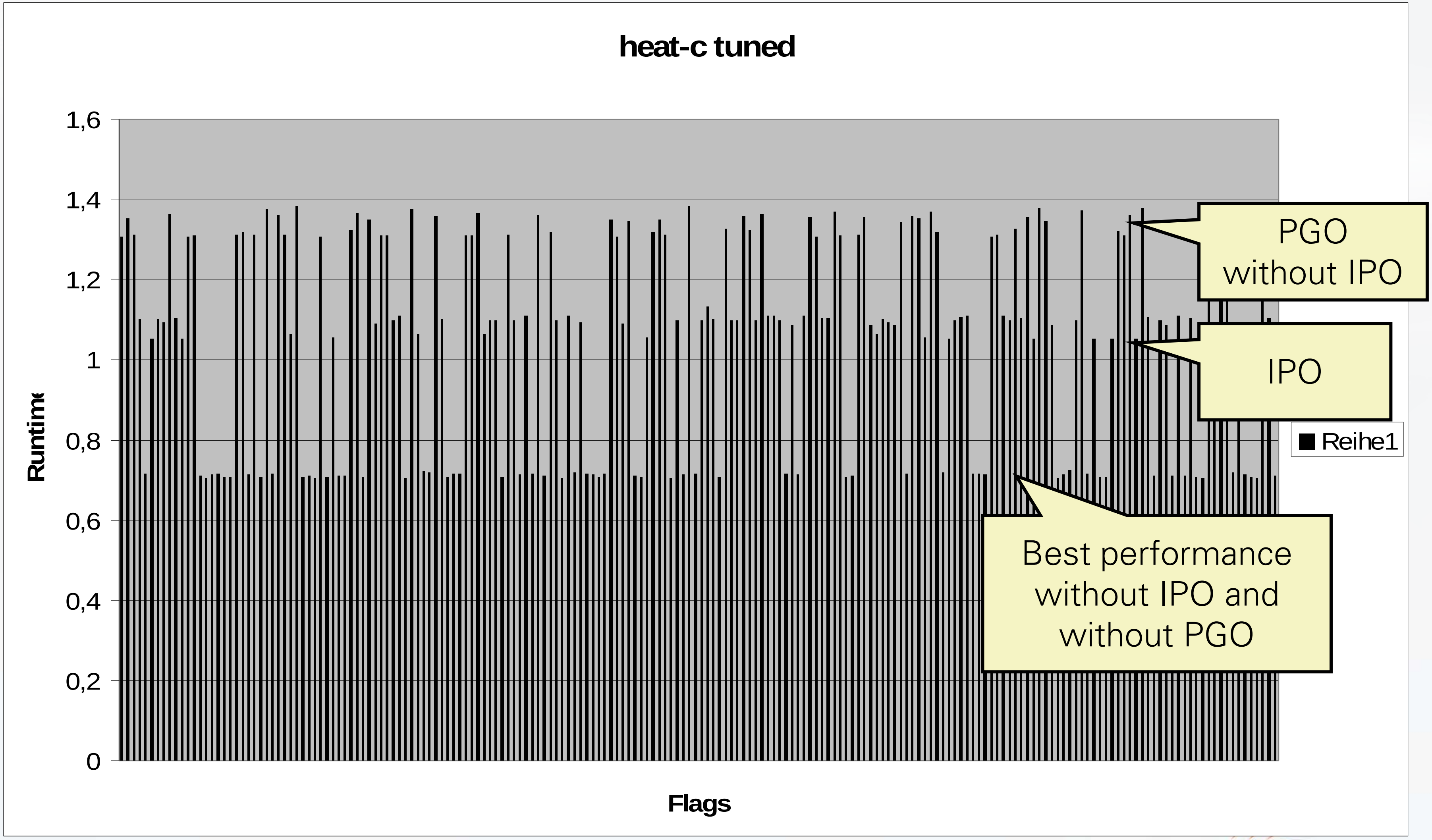
- Remove copy from **`thetanew`** to **`theta`**
  - Swap the array pointers instead
  - Also huge speed-up

- C only: **`fmax`** is very costly (don’t know why...)
  - Replace by ?-operator expression
  - Now C version is faster than Fortran version...

Exercise 2: Compiler Flags



Exercise 2: Compiler Flags

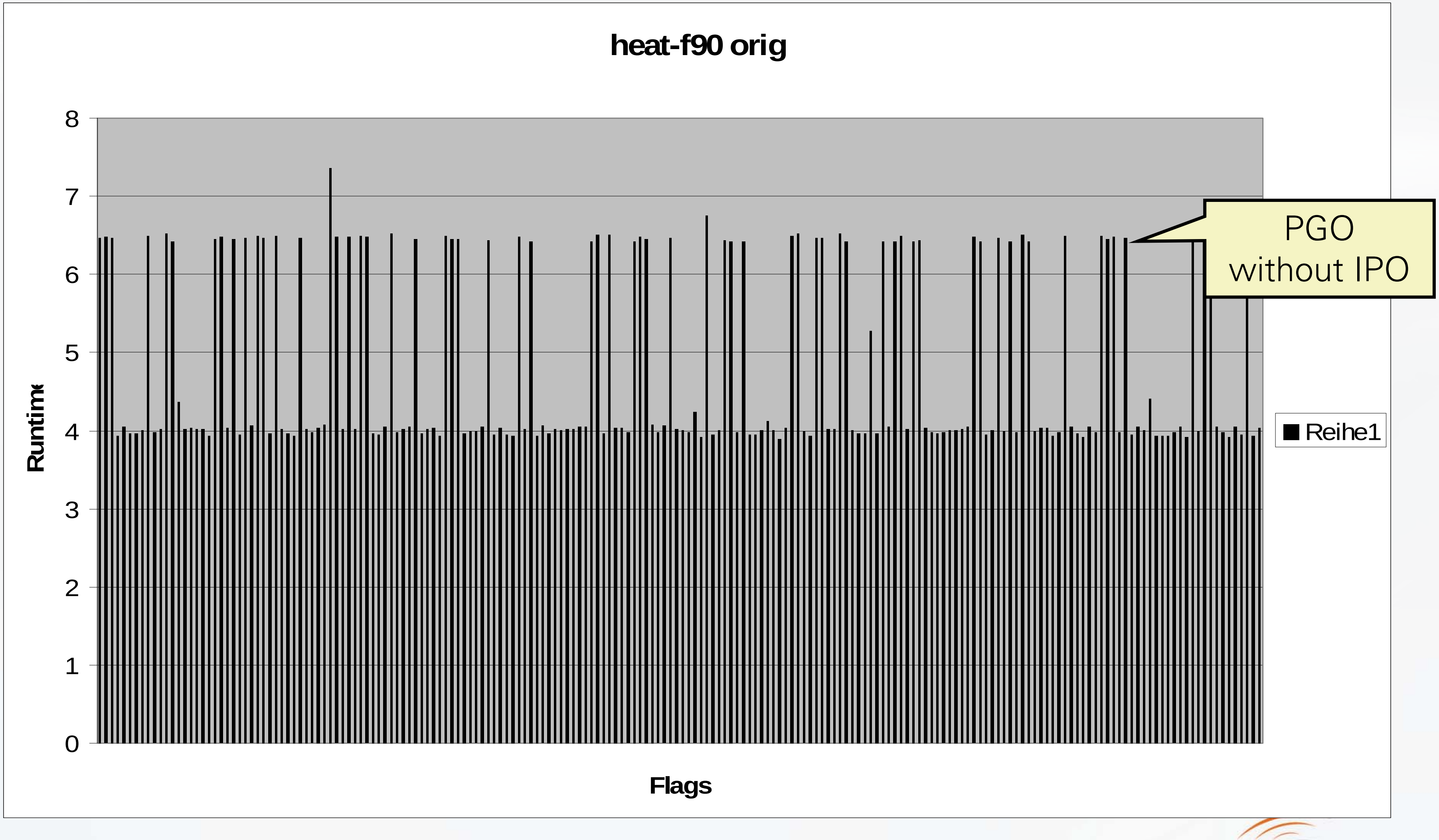


PGO without IPO

IPO

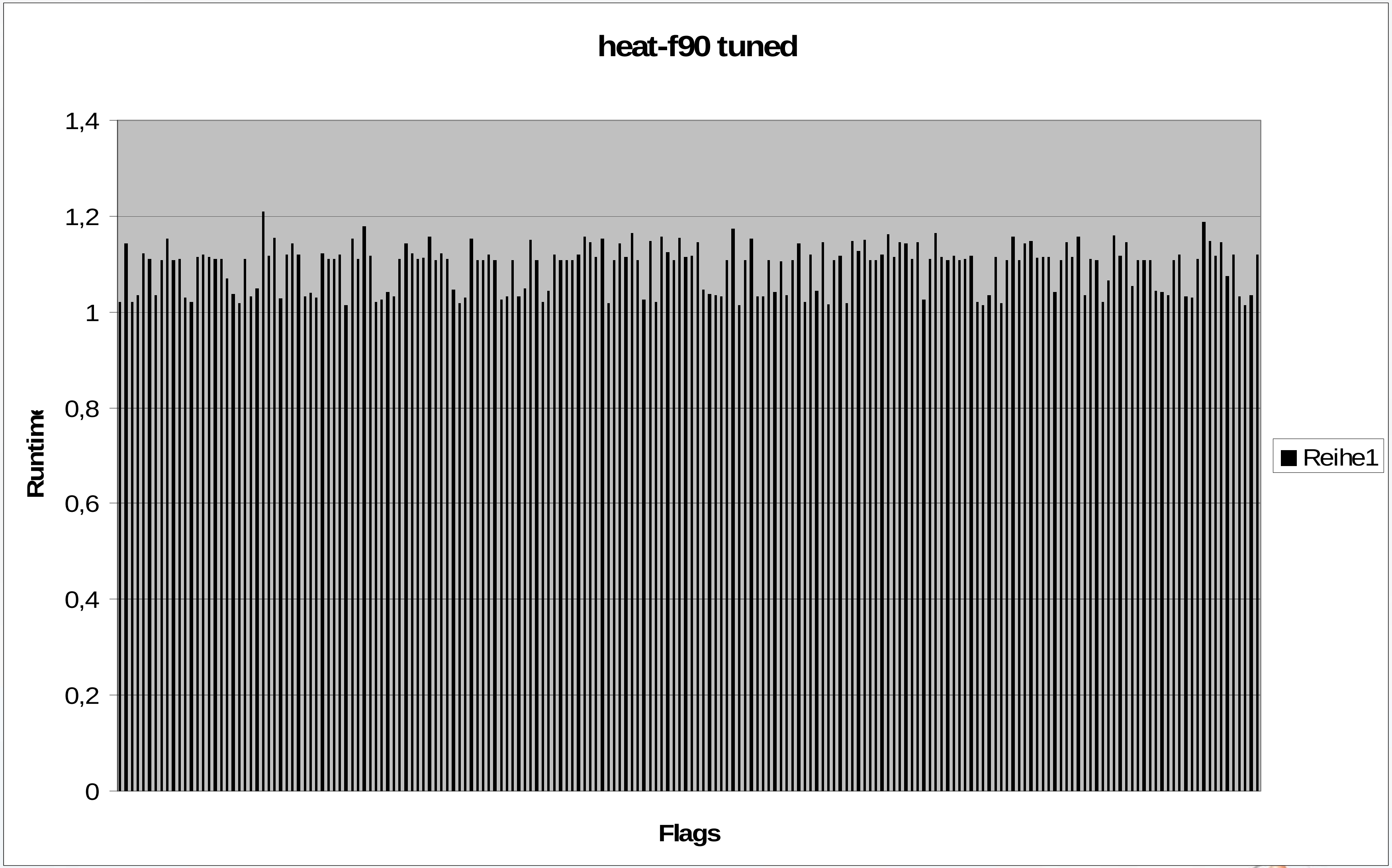
Best performance without IPO and without PGO

Exercise 2: Compiler Flags



PGO without IPO

Exercise 2: Compiler Flags


heat-f90 tuned
Runtime
Flags
■ Reihe1