

High Performance Computing 1b

Parallelization of a 2D Hydro Solver

Remo Hertig
Stephan Radonic

June 30, 2015

Introduction and physics

Our task was to parallelize an existing C code, originally written by Prof. Romain Teyssier in Fortran, which solves the Euler equations in 2D using a Godunov scheme. A hyperbolic PDE in conservation law form is represented as

$$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0 \quad (1)$$

The euler equations are a set of equations which basically state the momentum, mass and energy conservation. Discretization on a grid yields

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta x}{\Delta t} (\mathbf{F}_{i-1/2} - \mathbf{F}_{i+1/2}) \quad (2)$$

where $\mathbf{F}_{i\pm 1/2}$ are the fluxes at the cell boundaries, the Godunov scheme uses various approximations for $\mathbf{F}_{i\pm 1/2}$, depending on the specific variation of the method, e.g upwind scheme, lax-friedrich, ...

Parallelization

Parallelization: Speedup and Scaling

We compare the average time step durations for a single process up to approximately 1500 parallel processes for a fixed problem size (in our case 60994×120). As observable in the strong scaling graph (ref figure) we get a super linear scaling up to 800 processes. The super linearity of the scaling can be explained with cache usage effects.

Optimal cache memory usage only works well for rectangular shaped domains (large x small y for parallelization in x direction). We have compared how a fixed sized problem performs with different x/y ratios (see figure xx). We can clearly see that the performance increases with decreasing y/x size, up to a ratio, where each processes computing domain gets too small and becomes inefficient.

Parallelization: Speedup and Scaling

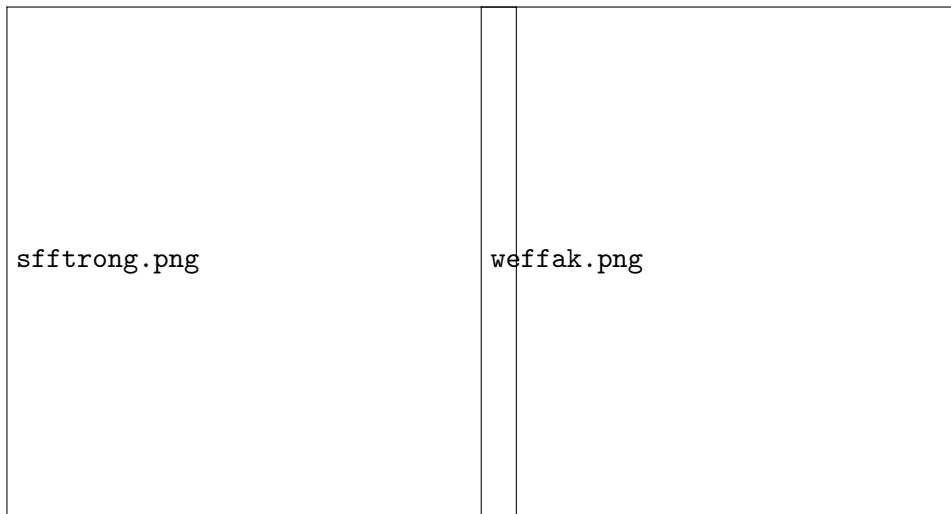


Figure: Strong scaling for a fixed grid

Figure: Performance comparison of a

Parallelization: Speedup and Scaling

...tzututzut

speeddup.png

Output image

To show the power of parallel processing we wanna show an excerpt from our high resolution image (3060×500) at simulation time $t=600$ seconds (corresponds to the 200'000th time step in our simulation). In order to avoid unnecessary wasting of computing resources, we didn't want to use a larger grid size.

