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 code, which solves the Euler equations in 2D using a godunov scheme. A hyperbolic PDE in conservation law form is representend by a written as

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

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

$$\mathbf{U}_i^{n+1} = \mathbf{U_i^n} + \frac{\Delta x}{\Delta t}(\mathbf{F_{i-1/2}} - \mathbf{F_{i+1/2}}) \tag{2}$$

where $\mathbf{F_{i\pm1/2}}$ are the fluxes at the cell boundaries, the godunov scheme uses various approximations for $\mathbf{F_{i\pm1/2}}$

Parallelization strategy

Serial vs. Parallel Processing

We compare the average time step durations for a single process up to approximatly 1500 parallel processses for a fixed problem size (in our case 60994 x 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.

Oprimal 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 diffrent x/y ratios (see figure xx). We can cleary see that the performance increases with deacrasing y/x size, up to a ratio, where each processes computing domain gets too small and becomes inefficient.

Scaling and Speedup

sfftrong.png

weffak.png

Scaling and Speedup

...tzututzut

speehdup.png

High resolution example

thube2.png