Conjugate Gradient Method in MPI and CUDA

MATH-454 - Parallel & High Performance Computing

Charles Dufour, January 2023

MPI Implementation

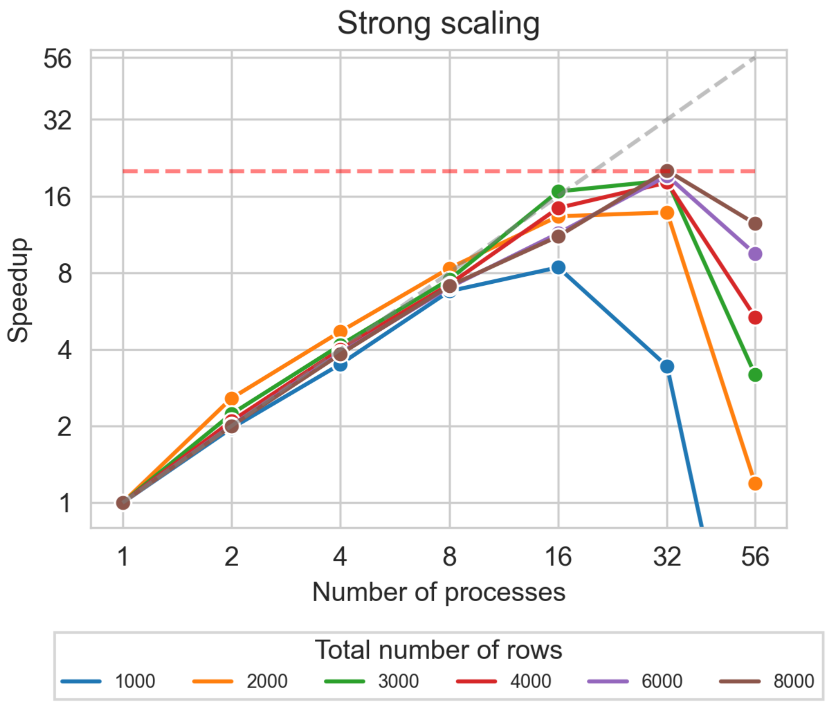
Algorithm 2: MPI Conjugate gradient pseudo-code

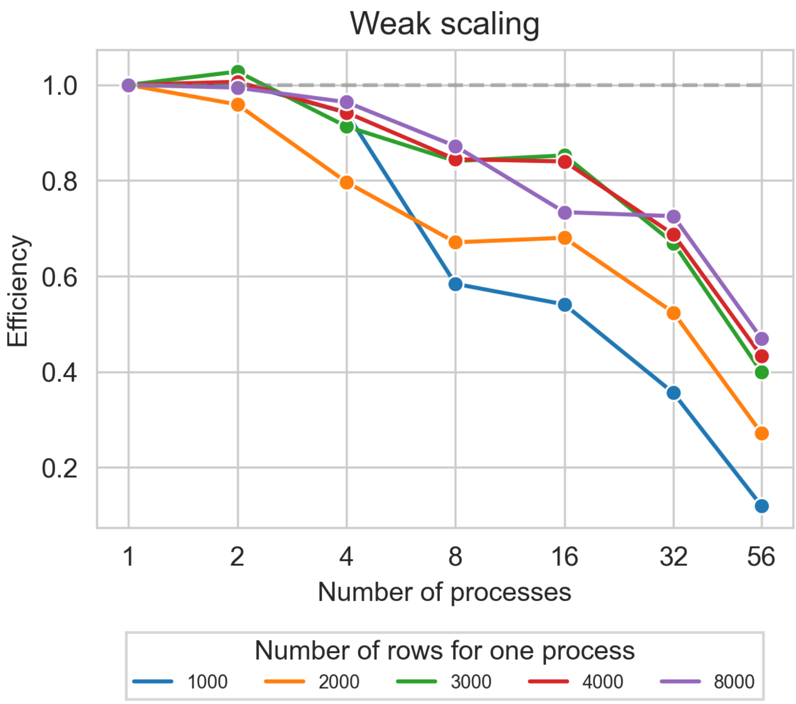
Data: $A_{(i)}, b_{(i)}, x_0, \epsilon$
Result: x such that $Ax \approx b$

$r_{0(i)} = b_{(i)} - A_{(i)}x_0$;
$p_{0(i)} = r_{0(i)}$;
$\gamma_0 = \sum_{i=0}^{p-1} r_{0(i)}^T r_{0(i)}$; // Done with MPI_Allreduce
$k = 0$;
while $||r_k||_2 > \epsilon$ **do**
    Concatenate all the $p_{k(i)}$ into p_k ; // Done with MPI_AllGather
    $z_{k(i)} = A_{(i)} p_k$;
    $w = \sum_{i=0}^{p-1} p_{k(i)}^T z_{k(i)}$; // Done with MPI_Allreduce
    $\alpha = \gamma_k / w$;
    $x_{k(i)} = x_{k(i)} + \alpha p_{k(i)}$;
    $r_{k+1(i)} = r_{k(i)} - \alpha z_{k(i)}$;
    $\gamma_{k+1} = \sum_{i=0}^{p-1} r_{k+1(i)}^T r_{k+1(i)}$; // Done with MPI_Allreduce
    $p_{k+1(i)} = r_{k+1(i)} + \frac{\gamma_{k+1}}{\gamma_k} p_{k(i)}$;
    $k = k + 1$;
end
Concatenate and return x_k ; // Done with MPI_Gather

Amdhal's Law & Gustafson's Law

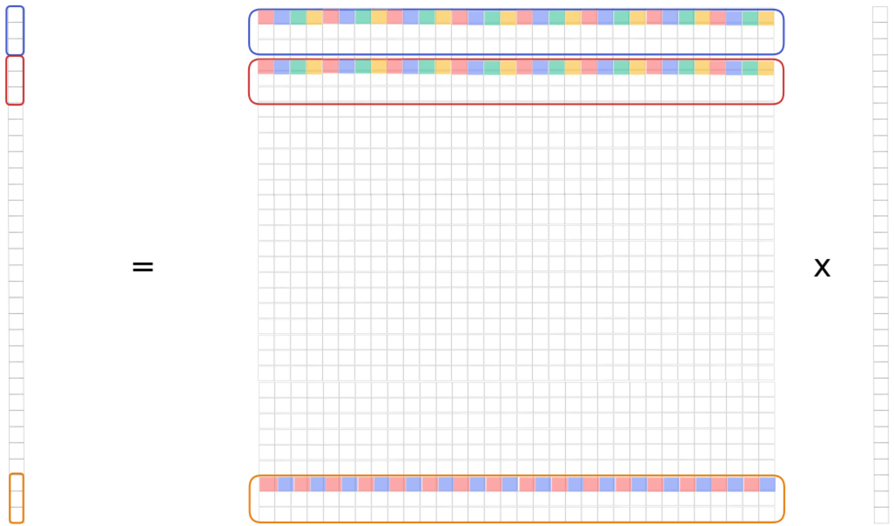
Strong and weak scaling on HELVETIOS





CUDA Implementation

Grid and block size



- 1D grid of blocks
- 2D grid of threads
- Strided access to matrix and vector RHS → coalesced memory access
- Everything on device except stopping condition

Influence of grid and block size

Using matrix of size 10000x10000