Parallel Conjugate Gradient for Dense Matrices

Federico Betti
MATH-454, Parallel and High Performance Computing

EPFL — January 12, 2023

MPI – Hard scaling

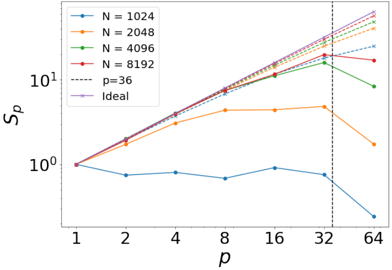
Profiling/Hard scaling analysis

- Matrix vector products takes majority of running time.
- Amdahl's Law: $S(p) \leq \frac{1}{\alpha}$.

N	1024	2048	4096	8192
α	0.0247	0.0090	0.0050	0.0019

Parallelization strategy

- Row-wise domain decomposition.
- Reduction on α_k and r_k and collective gathering for solution vector p_k.
- Computation cost $\mathcal{O}\left(\frac{N^2}{p}\right)$.
- Communication cost $\mathcal{O}(\beta \frac{p-1}{p} N + \alpha \log_2 p)$ [2].



Hard scaling results

- Speedup suffers from communications and synchronization overhead [1].
- Increasing the problem size boosts performance and mitigates drop with network communications ($p > 36$).

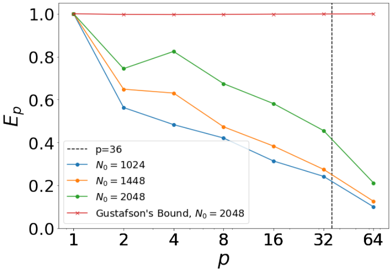
MPI – Weak scaling

Weak scaling analysis

- For weak scaling, we fix the number of iterations for a fair comparison.
- To keep the workload $\frac{N^2}{p}$ constant, from a serial problem size N_0, we then take $N(p) = N_0\sqrt{p}$.
- $\beta(N)$ is always rather small.

Weak scaling results

- Gustafson's Law is too optimistic for relatively small starting points.
- There is evidence for improvement when increasing N_0.



MPI Conclusions

- Synchronization overhead deteriorates parallelization potential. Hybrid OpenMP-MPI may provide benefits.
- Improvement for massively large problems.

CUDA

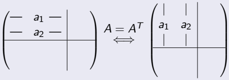
One thread per row approach

- Low GPU occupancy.
- Block-wise problem is too big to use shared memory.

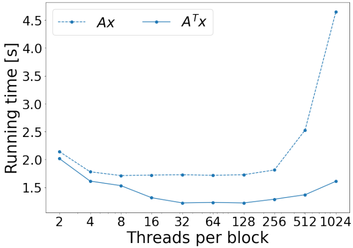
Key observation

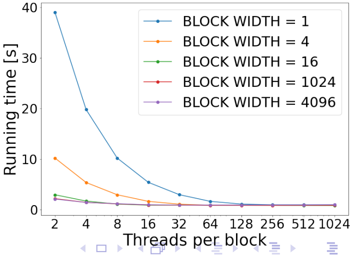
- Computing equivalently $A^T x$ favours memory coalescing.

Multiple threads per row approach



- Small block width gives queuing blocks. Large block width gives previous results.
- Can be improved using shared memory for x.

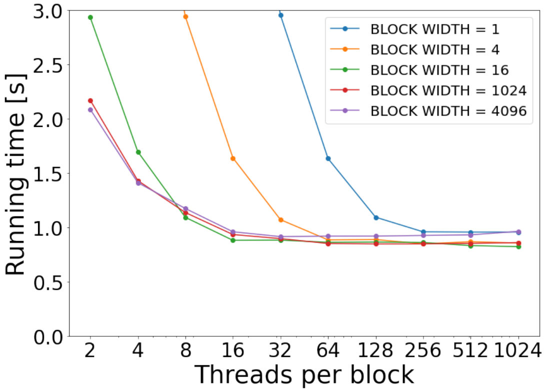




References and appendix

[1] R Oguz Selvitopi and Cevdet Aykanat. *Reducing Synchronization Overheads in CG-type Parallel Iterative Solvers by Embedding Point-to-point Communications into Reduction Operations.*

[2] Rajeev Thakur, Rolf Rabenseifner, and William Gropp. *Optimization of collective communication operations in MPICH.*



N	α
1024	0.0247
1448	0.0148
2048	0.0090
2896	0.0065
4096	0.0050
5792	0.0034
8192	0.0019
11585	0.0012
16384	0.0004