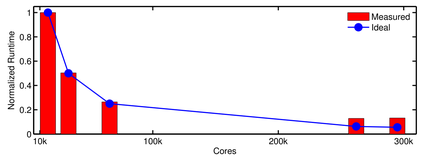
Extreme-scale space-time parallelism

Daniel Ruprecht, Robert Speck, Matthew Emmett, Matthias Bolten, Rolf Krause

Università della Svizzera italiana | Faculty of Informatics | Institute of Computational Science ICS

Motivation



Can we extend strong scaling and make the smallest bar even smaller?

- Exascale HPC systems will have $100{,}000{,}000$ or more cores
  $\rightarrow$ numerical methods must feature an **extreme level of concurrency**
- Time-dependent partial differential equations typically use **mesh decomposition** to parallelize in space
- **Time-parallel methods** add concurrency to ODEs

$$y'(t) = f(y(t), t), \quad y(0) = y_0$$

  and **extend strong scaling limit** of pure space-parallelism.
- **Examples** of time-parallel methods
  - Parabolic multi-grid (Hackbusch 1984)
  - Parareal (Lions et al. 2001)
  - PITA (Farhat et al. 2003)
  - PFASST (Emmett and Minion 2012) [1, 2].
- Very few studies of performance in large- or extreme-scale parallel runs

Multi-level spectral deferred correction

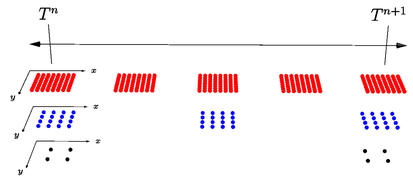


Figure 1: Sketch of the space-time mesh hierarchy used in MLSDC. Coarser levels have fewer collocation nodes, fewer mesh points and possibly lower-order stencils.

- **Picard formulation** of initial value problem

$$y(T^{n+1}) = y(T^n) + \int_{T^n}^{T^{n+1}} f(y(s), s)\, ds$$

- Discretized with **collocation formula**

$$\mathbf{Y} = \mathbf{Y}_0 + \Delta t \mathbf{S}\mathbf{F}(\mathbf{Y})$$

  $\rightarrow$ implicit nonlinear system for $\mathbf{Y}$.
- **Spectral deferred corrections** iteratively "sweep" with a low-order method (e.g. implicit-explicit Euler) $\rightarrow$ overall $\mathcal{O}(\Delta t^k)$ accuracy
- **MLSDC** [5] sweeps on **space-time mesh hierarchy**: space-time coarsening strategies minimize cost of coarse sweeps
- **FAS correction** allows all levels to converge to accuracy of fine-level

PFASST

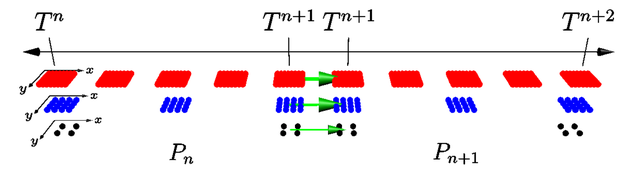


Figure 2: Space-time mesh hierarchy in PFASST. Multiple time-ranks run multiple MLSDC iterations concurrently on different time-intervals. After each sweep, updated initial values are communicated forward in time (green arrows).

- **PFASST** = "Parallel full approximation scheme in space and time"
- **Concurrent MLSDC** iterations over multiple time-steps
- Communication of **updated initial values** after each sweep
- **Minimal synchronicity**: blocking communication only on coarsest level, everywhere else overlap of communication and computation [4]
- **Theoretical estimate** for speedup

$$s(P_T) = \frac{K_S P_T}{P_T \alpha + K_P (1 + \alpha + \beta)}$$

K_S : time-serial SDC iterations
K_P : time-parallel PFASST iterations
P_T : number of time-ranks
α : runtime ratio coarse to fine sweep
β : overhead per PFASST iteration

- Documented scaling for particle-based discretization on up to $262{,}144$ cores on IBM Blue Gene/P [3].

Extreme-scale space-time parallel benchmarks

- **3D heat equation** with forcing term

$$u_t(\mathbf{x}, t) = \Delta u(\mathbf{x}, t) + f(\mathbf{x}, t), \mathbf{x} \in [0, 1]^3$$

- Space and time dependent **forcing**

$$f(\mathbf{x}, t) = -\sin(\pi x_1)\sin(\pi x_2)\sin(\pi x_3) \times \left(\sin(t) - \nu\pi^2 \cos(t)\right)$$

  with $\nu = 0.1$.
- **Implicit-Explicit Euler** for sweeps: implicit diffusion, explicit forcing
- **Fine level**: 511^3 mesh points, 4th-order compact stencil, 5 collocation nodes
- **Coarse level**: 255^3 mesh points, 2nd-order stencil, 3 collocation nodes
- Residual tolerance 1.0×10^{-10} in SDC and PFASST, **relative error** 1.1×10^{-11}
- Linear systems in sweeps with IMEX Euler $\rightarrow$ **parallel multi-grid** (PMG) in space

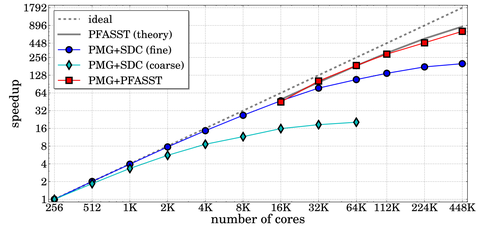


Figure 3: Speedup of space-time parallel PMG+PFASST (red), time-serial PMG+SDC for the fine level problem (blue) and PMG+SDC for the coarse level problem (cyan).

Strong scaling:

- PMG+PFASST (time-parallel) significantly improves strong scaling over PMG+SDC (time-serial)
- On $458{,}752$ cores, the speedup of PMG+PFASST is about a factor of three better than pure space-parallelism
- Additional speedup from PFASST close to theoretical estimate

IBM Blue Gene/Q JUQUEEN

Time-ranks	Speedup	Efficiency
2	2.16	1.08
4	3.97	0.99
7	6.20	0.89
14	9.65	0.69
28	15.12	0.54

Table 1: Speedup and efficiency of the temporal parallelization with PFASST. The reference is time-serial single-level PMG+SDC with 16K cores in space.

Efficiency:

- For benchmark problem, $K_P < K_S$, leading to superlinear speedup at first
- Aggressive coarsening $\rightarrow$ good efficiency of PFASST as long as K_P remains small
- Better than 50% efficiency of PFASST even on full IBM Blue Gene/Q, using 448K cores in total for space-time parallelism

References

[1] M. L. Minion: **A hybrid Parareal spectral deferred corrections method**, CAMCOS 2010.

[2] M. Emmett, M. L. Minion: **Toward an efficient parallel in time method for partial differential equations**, CAMCOS 2012.

[3] R. Speck, D. Ruprecht et al.: **A massively space-time parallel N-body solver**, Supercomputing 2012.

[4] M. Emmett, M. L. Minion.: **Efficient implementation of a multi-level parallel in time algorithm**, DDM 2012 (In press).

[5] R. Speck, D. Ruprecht et al.: **A multi-level spectral deferred correction method**, arXiv:1307.1312 [math.NA], 2013.

[6] R. Speck, D. Ruprecht et al.: **A space-time parallel solver for the three dimensional heat equation**, arXiv:1307.7867 [cs.NA], 2013.

Acknowledgments

- Swiss National Science Foundation grant 145271
- Project "ExaSolvers" within German DFG Priority Programme "Software for Exascale Computing"
- Project HWU12 for computing time on JUQUEEN
- Coauthors and collaborators at

JÜLICH FORSCHUNGSZENTRUM
BERKELEY LAB
BERGISCHE UNIVERSITÄT WUPPERTAL