Extreme-scale space-time parallelism

D. Ruprecht*, R. Speck†*, M. Emmett‡, M. Bolten§ R. Krause*,
*Institute of Computational Science, Università della Svizzera italiana, Lugano, Switzerland.
Email: {daniel.ruprecht, rolf.krause}@usi.ch
†Jülich Supercomputing Centre, Jülich, Germany. Email: r.speck@fz-juelich.de
‡Lawrence Berkeley National Laboratory, Berkeley, USA. Email: mwemmett@lbl.gov
§ Department of Mathematics and Science, Bergische Universität Wuppertal, Germany.
Email: bolten@math.uni-wuppertal.de

I. Introduction

For time-dependent partial differential equations, temporal parallelization has been shown to be an attractive way to introduce a new dimension of concurrency in addition to spatial mesh decomposition approaches already widely used. This paper presents scaling results for the time-parallel "parallel full approximation scheme in space and time" (PFASST) [1], [2] algorithm combined with a parallel multigrid method (PMG) [3] in space. We present strong scaling of PMG+PFASST for runs using all 448K cores of the IBM Blue Gene/Q JUQUEEN at Jülich Supercomputing Centre which goes far beyond existing results in [4]. The presented experiments are by far the largest runs of a space-time parallel method in terms of cores used and significantly exceed the 256K cores used in the previous record in [5].

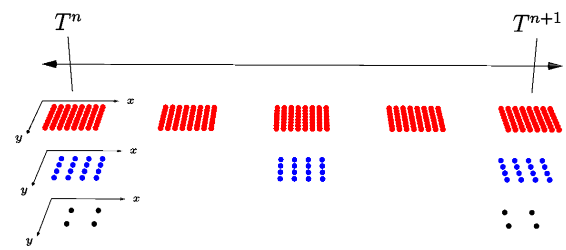
II. Numerical Algorithms

Collocation methods for the integration of initial value problems advance the solution over a time interval $[T^n, T^{n+1}]$ by approximating the integral in the Picard formulation

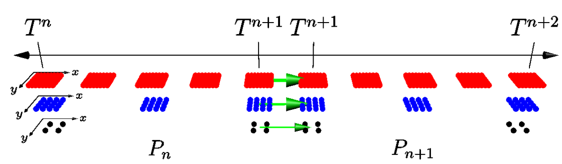
$$y(T^{n+1}) = y(T^n) + \int_{T^n}^{T^{n+1}} f(y(s), s)\ \mathrm{d}s \qquad (1)$$

using numerical quadrature. For Gauss-type collocation nodes, spectral deferred corrections (SDC) [6] are an attractive approach to iteratively solve the nonlinear collocation system arising from the approximation of (1). SDC uses a low-order method, typically an explicit and/or implicit Euler scheme, to perform multiple "sweeps", with each sweep raising the formal order of accuracy of the method, up to the order of the collocation method.

Classical SDC methods only sweep over one set of collocation nodes. In multi-level SDC (MLSDC) [7], sweeps are performed along a hierarchy of space-time meshes, with coarser levels using fewer collocation nodes as well as a coarser spatial discretization. Figure 1a sketches such a space-time mesh hierarchy: On the finest level (red), there are five collocation nodes and a high-resolution spatial discretization. On the intermediate level (blue), the number of collocation nodes is reduced to three and the spatial mesh is also coarser. The coarsest level (grey) features only two collocation nodes and, in the sketch, a spatial mesh with four nodes only. Interpolation and restriction operators are employed to transfer the solution between meshes while a FAS correction allows the solution on the coarser levels to converge up to an accuracy governed by the discretization on the finest level, see [7].



(a) 3-level MLSDC with space-time coarsening (time-serial)



(b) 3-level PFASST with space-time coarsening (time-parallel)

Fig. 1: Graphical sketch of the space-time mesh hierarchy used in MLSDC (upper). PFASST (lower) iterates concurrently on multiple time-slices and sends updated initial values forward in time after each sweep.

The "parallel full approximation scheme in space and time" (PFASST) [1], [2] can be understood as a time-parallel variant of MLSDC. It performs MLSDC iterations concurrently on multiple time intervals while frequently sending updated initial values forward in time. Figure 1b sketches the mesh hierarchy of PFASST: A time-slice $[T^n, T^{n+1}]$ is assigned to a processor P_n, which runs MLSDC on this interval. Initial values on each processor are generated by propagating the initial value using a number of sweeps on the coarse level, similar to the initialization phase in Parareal [8]. On each level, after a sweep is completed, an updated initial value is sent to the processor handling the next time-slice, but blocking communication is required on the coarsest level only, so that PFASST requires minimal synchronicity between the different time-slices, see [9].

To solve the linear problem arising in each step of the implicit-explicit Euler method during the sweeps of SDC or

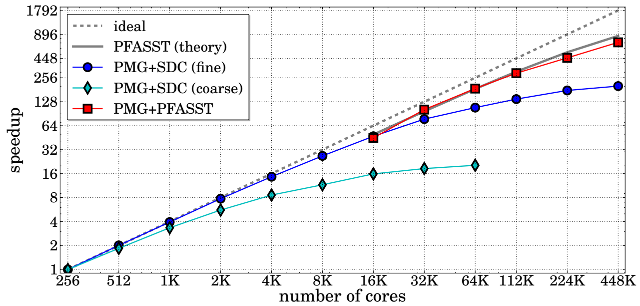


Fig. 2: Total speedup of time-serial PMG+SDC for the coarse problem (light blue), the fine problem (dark blue) and the space-time parallel combination PMG+PFASST (red).

BlueGene/Q JUQUEEN

Time-ranks	Speedup	Efficiency
2	2.16	108%
4	3.97	99%
7	6.20	89%
14	9.65	69%
28	15.12	54%

TABLE I: Speedup and efficiency of the temporal parallelization. For the problem studied here, PFASST at first requires fewer iterations than the time-serial SDC reference, leading to the better-than-ideal speedup on two time-ranks.

PFASST, a parallel multi-grid method is employed, see [3] for general features. A discussion of PMG in the context of MLSDC can be found in [7].

III. Space-time parallel scaling on JUQUEEN

The benchmark problem considered here is the three dimensional heat equation with a forcing term as described in [4]. A method of lines approach is employed, discretizing in space first, using 511^3 nodes and a fourth order compact stencil for the Laplacian on the fine level and 255^3 nodes and a second order stencil on the coarse level. The resulting initial value problem is then solved with PFASST, employing 5 collocation nodes on the fine and 3 collocation nodes on the coarse level.

Figure 2 shows the scaling of PMG+PFASST (red). These runs use PMG with a fixed number of 16K cores in space times 1, 2, 4, 7, 14 and 28 cores in time for PFASST, so that the largest run uses $28 \times 16\text{K} = 448\text{K}$ cores and thus the full JUQUEEN machine. For comparison, scaling of standalone PMG, that is time-serial, single-level SDC with spatial parallelization only is also shown (blue). Both, the time-serial SDC as well as the time-parallel PFASST iterations use a threshold for the residual of 1.0×10^{-10}, leading to a relative error of 1.1×10^{-11}.

The solid grey line indicates the theoretical speedup estimate for PFASST, see e.g. [5] for the formula. The speedup provided by PFASST is very close to the theoretical estimate, demonstrating again that PFASST can efficiently be used even in extreme-scale parallel runs. Although stand-alone PMG scales quite well up to the full 448K cores, the space-time parallel solver combining PFASST and PMG features significantly improved strong scaling: On the full machine, PMG+PFASST provides about a factor of three better speedup than the PMG+SDC run using all cores for spatial parallelization only. Also, because of the aggressive coarsening in space on the higher levels, the efficiency of PFASST even in the very large scale runs is still better than 50%, see Table I.

IV. Summary

The poster presents scaling results of a combination of the time-parallel "parallel full approximation scheme in space and time" (PFASST) with a parallel multi-grid (PMG) method in space. It is shown that the combined space and time parallel approach significantly improves strong scaling compared to standalone PMG. Timings from runs on the full IBM Blue Gene/Q JUQUEEN using up to 448K cores are reported, setting a new record for the number of cores employed in a space-time parallel simulation.

Acknowledgments

This research is supported by Swiss National Science Foundation (SNSF) grant 145271 under the lead agency agreement through the project "ExaSolvers" within the Priority Programme 1648 "Software for Exascale Computing" (SPPEXA) of the Deutsche Forschungsgemeinschaft (DFG). Computing time on JUQUEEN was provided by project HWU12. D. R. and M. E. acknowledge support from SNSF grant 147597.

References

[1] M. L. Minion, "A hybrid parareal spectral deferred corrections method," *Communications in Applied Mathematics and Computational Science*, vol. 5, no. 2, pp. 265–301, 2010. [Online]. Available: http://dx.doi.org/10.2140/camcos.2010.5.265

[2] M. Emmett and M. L. Minion, "Toward an efficient parallel in time method for partial differential equations," *Communications in Applied Mathematics and Computational Science*, vol. 7, pp. 105–132, 2012. [Online]. Available: http://dx.doi.org/10.2140/camcos.2012.7.105

[3] E. Chow, R. D. Falgout, J. J. Hu, R. S. Tuminaro, and U. M. Yang, "A survey of parallelization techniques for multigrid solvers," in *Parallel Processing for Scientific Computing*, ser. SIAM Series of Software, Environements and Tools. SIAM, 2006.

[4] R. Speck, D. Ruprecht, M. Emmett, M. Bolten, and R. Krause, "A space-time parallel solver for the three-dimensional heat equation," in *Proceedings of ParCo: International Conference on Parallel Computing*, ser. Advances in Parallel Computing, 2013, (Accepted). [Online]. Available: http://arxiv.org/abs/1307.7867

[5] R. Speck, D. Ruprecht, R. Krause, M. Emmett, M. Minion, M. Winkel, and P. Gibbon, "A massively space-time parallel N-body solver," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC '12. Los Alamitos, CA, USA: IEEE Computer Society Press, 2012, pp. 92:1–92:11. [Online]. Available: http://dx.doi.org/10.1109/SC.2012.6

[6] A. Dutt, L. Greengard, and V. Rokhlin, "Spectral deferred correction methods for ordinary differential equations," *BIT Numerical Mathematics*, vol. 40, no. 2, pp. 241–266, 2000. [Online]. Available: http://dx.doi.org/10.1023/A:1022338906936

[7] R. Speck, D. Ruprecht, M. Emmett, M. Minion, M. Bolten, and R. Krause, "A multi-level spectral deferred correction method," 2013, arXiv:1307.1312 [math.NA]. [Online]. Available: http://arxiv.org/abs/1307.1312

[8] J.-L. Lions, Y. Maday, and G. Turinici, "A "parareal" in time discretization of PDE's," *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, vol. 332, pp. 661–668, 2001. [Online]. Available: http://dx.doi.org/10.1016/S0764-4442(00)01793-6

[9] M. Emmett and M. L. Minion, "Efficient implementation of a multi-level parallel in time algorithm," in *Proceedings of the 21st International Conference on Domain Decomposition Methods*, ser. Lecture Notes in Computational Science and Engineering, 2012, (In press). [Online]. Available: http://dd21.inria.fr/pdf/emmett_mini_15.pdf