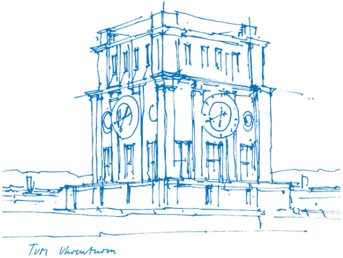
School of Computation, Information and Technology
Technical University of Munich

Dynamic HPC resources for PinT part II: Algorithmic perspective

Hans-Joachim Bungartz[1], **Jan Fecht**[1], **Keerthi Gaddameedi**[1], **Dominik Huber**[1], **Michael Minion**[4], **Martin Schreiber**[2], **Valentina Schüller**[3]

February 6th, 2024



[1] TUM School of CIT, Technical University Munich, Boltzmannstraße 3, 85748 Garching, Germany
[4] Lawrence Berkeley National Laboratory (LBNL), 1 Cyclotron Rd, Berkeley, CA 94720, USA
[2] Université Grenoble Alpes, Laboratoire Jean-Kuntzmann, Inria, 38000 Grenoble, France
[3] Centre for Mathematical Sciences, Lund University, Sölvegatan 18, 223 62 Lund, Sweden

Outline

1. SWEET
   - Governing equations
   - Space discretization
   - Time integration
2. Dynamic PFASST
3. Results

Shallow Water Equations Environment for Tests

PDE solver framework developed for research on time integration methods for weather and climate simulations.

Applications: Shallow water equations, Advection, Burger's . . .

Domains
Bi-periodic plane
Sphere

Space discretization
Finite differences
Spectral methods
Spherical harmonics

Time discretization
Runge-Kutta
Semi-Lagrangian
Parareal
REXI
PFASST
MGRIT . . .

SWEET

SWEET - Shallow water equations on the rotating sphere

Governing equations

$$\frac{\partial U}{\partial t} = \mathcal{L}(U) + \mathcal{N}(U) \tag{1}$$

where $U = [\phi, \zeta, \delta]^T$,

- $\phi \rightarrow$ Geopotential
- $\zeta \rightarrow$ Vorticity
- $\delta \rightarrow$ Divergence

SWEET - Shallow water equations on the rotating sphere

Governing equations

$$\frac{\partial U}{\partial t} = \mathcal{L}(U) + \mathcal{N}(U) \tag{1}$$

- $\mathcal{L}(U)$ contains stiff terms
- Linear wave motion induced by gravitational forces and a diffusion term

$$\mathcal{L}(U) = \begin{bmatrix} -\bar{\phi}\delta + \nu\nabla^2\phi' \\ \nu\nabla^2\zeta \\ -\nabla^2\phi + \nu\nabla^2\delta \end{bmatrix}$$

SWEET - Shallow water equations on the rotating sphere

Governing equations

$$\frac{\partial U}{\partial t} = \mathcal{L}(U) + \mathcal{N}(U) \tag{1}$$

- $\mathcal{N}(U)$ contains relatively less stiff terms
- Non-linear operators and Coriolis forces

$$\mathcal{N}(U) = \begin{bmatrix} -\nabla \cdot (\phi' V) \\ -\nabla \cdot (\zeta + f) V \\ k \cdot \nabla \times (\zeta + f) V - \nabla^2 (\frac{V \cdot V}{2}) \end{bmatrix}$$

Space discretization

- Global spherical harmonics transform is applied to the governing equations [3].
- Two step transform:

1. Discrete Fourier transform over the longitude λ

$$U^r(\mu) = \frac{1}{I}\sum_{l=1}^{I} U(\lambda_l, \mu) e^{ir\lambda_l}$$

2. Discrete Legendre transform in latitude $\mu = sin(\phi)$

$$U_s^r(\mu) = \sum_{j=1}^{J} U^r(\mu_j) P_s^r(\mu_j) w_j$$

Temporal splitting
IMEX

$$\frac{\partial U_s^r}{\partial t} = \mathcal{L}_s^r(U) + \mathcal{N}_s^r(U) \tag{2}$$

- Linear term $\mathcal{L}_s^r$ treated with implicit time integration schemes
- Non-linear term $\mathcal{N}_s^r$ treated with explicit time integration schemes

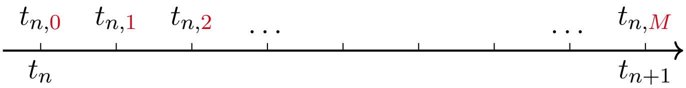
Parallel-in-time integration with PFASST

Components of PFASST

- Split intervals $[t_n, t_{n+1}]$ into M sub-intervals $\implies$ $M+1$ points

$$t_{n,M} = t_{n+1}$$

$$t_{n,0} = t_n$$



PFASST

Component 1 - Spectral deferred corrections

$$e(t) = \int_0^t G(t, e(t))\,d\tau + r(t)$$

- 3 quadrature node SDC

PFASST

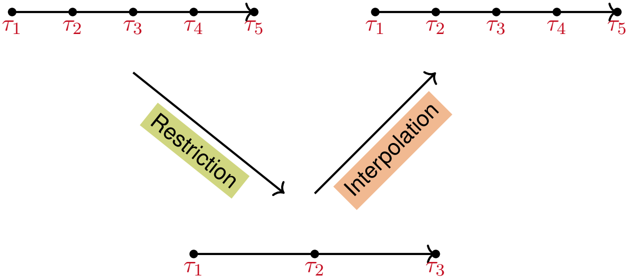
Component 2 - Multi-level with full approximation scheme

$$e(t) = \int_0^t G(t, e(t))\,d\tau + r(t)$$

- 5 node "finer" SDC

PFASST

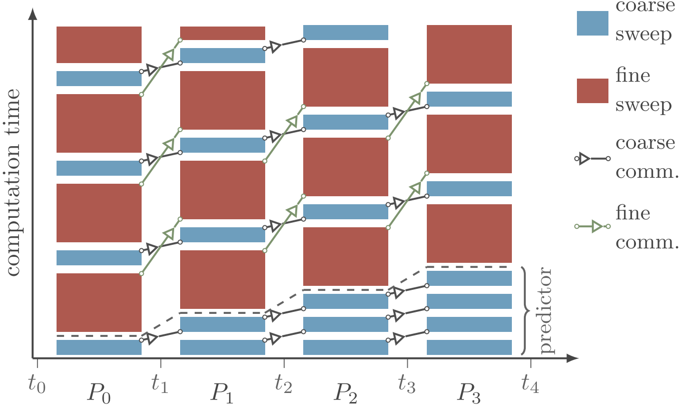
Component 2 - Multi-level with full approximation scheme



- FAS correction term τ = A^l (U^l) - R $\cdot$ A^h (U^h)

PFASST

Component 3 - Parareal



Demonstration of PFASST [4]

Outline

1 SWEET

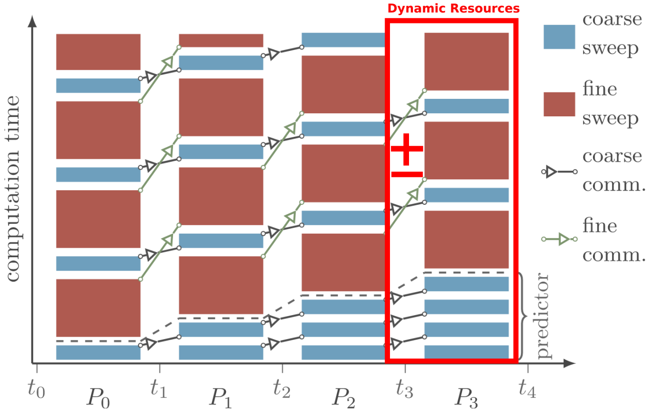
2 Dynamic PFASST

3 Results

Dynamic PFASST
The idea

- Ability to add or remove processes dynamically to the PFASST algorithm
- Newly added processes run PFASST iterations for the next timesteps
- Essentially, adding more parallel timesteps to the existing ones.

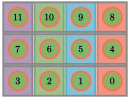
Dynamic PFASST

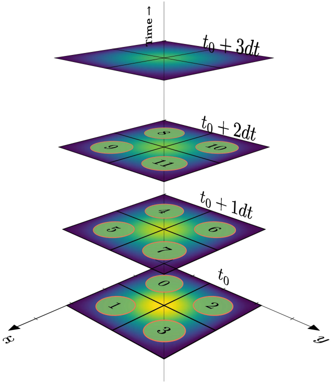


Dynamic PFASST

- LibPFASST - C++ implementation of PFASST by M. Emmett and M. Minion [1]
- LibPFASST has been integrated with SWEET in the past
- Now we work with dynamic version of LibPFASST with SWEET to investigate dynamicity in PFASST.

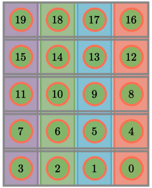
Dynamic PFASST

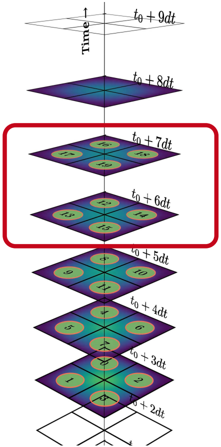




- Example: 2D Heat equation[2]
- 4 parallel processes for space
- 3 parallel timesteps

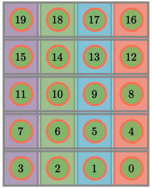
Dynamic PFASST

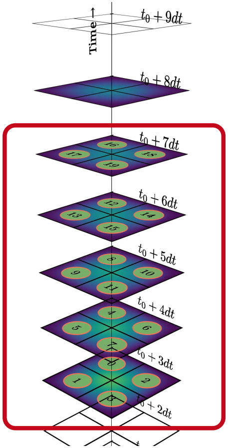




- For the next timestep block
- Add 2 more processes

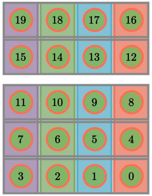
Dynamic PFASST

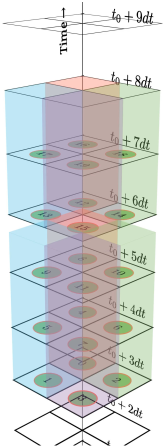




- Parallel timesteps grow by 2
- 5 timesteps in parallel

Dynamic PFASST





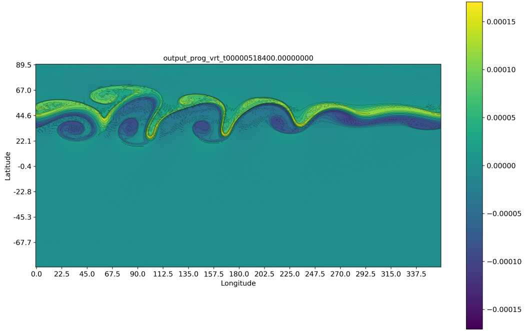
- 4 space parallel procs
- 5 time parallel procs

Outline

1. SWEET
2. Dynamic PFASST
3. Results
   - Galewsky benchmark
   - Residuals
   - Wallclock time

Galewsky benchmark

- Steady test case
- Initial condition - A simple perturbation to mid-latitude jet



At 7 days of simulation time

Parameter	Value
Spatial resolution	256 $\times$ 256
Total simulation time	102400s ($\sim$29h)
Time step size	100
Diffusion coefficient	1e4 m^2s^{-1}

PFASST parameters

Parameter	Value
[Fine,coarse] SDC nodes	[5,3]
Type of nodes	Gauss-Lobatto
[Fine,coarse] sweeps	[2,1]
PFASST iterations	4
Coarsening factor	0.5

Platforms

1. Docker cluster
   - Docker Swarm Toy Box
   - Virtual cluster for testing parallel libraries

2. CoolMUC-2 @ LRZ
   - 28-way Haswell-based nodes
   - 812 nodes
   - 28 cores per node, 2 hyperthreads per core
   - Peak performance of 1400 TFlop/s

Due to node granularity, experiments on the Linux cluster have been limited since the minimum number of parallel timesteps is the numbers of cores per node(here,56).

Residuals

1. Results are preliminary and currently under investigation

Residuals

1. Results are preliminary and currently under investigation
2. Residuals along all the time dimension have been plotted

Residuals

1. Results are preliminary and currently under investigation
2. Residuals along all the time dimension have been plotted
3. Use these residual values to understand how parallelism effects the algorithm

Residuals

1. Results are preliminary and currently under investigation

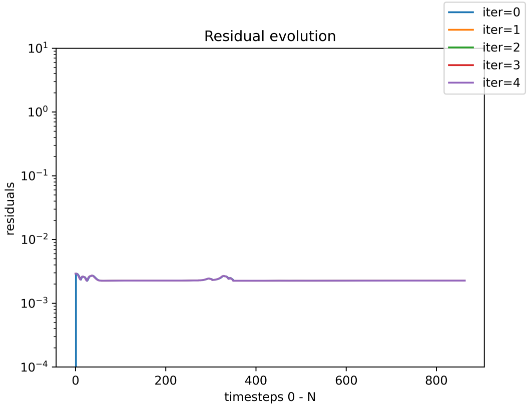
2. Residuals along all the time dimension have been plotted

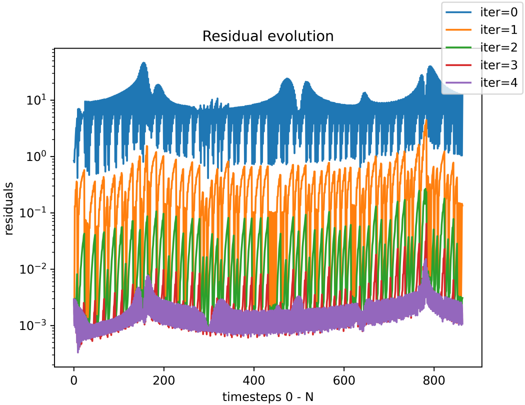
3. Use these residual values to understand how parallelism effects the algorithm

4. Overall goal : To improve resource utilization and efficiency by adopting number of resources along the time dimension "when necessary"

Residuals - Static vs. dynamic resources

Docker cluster



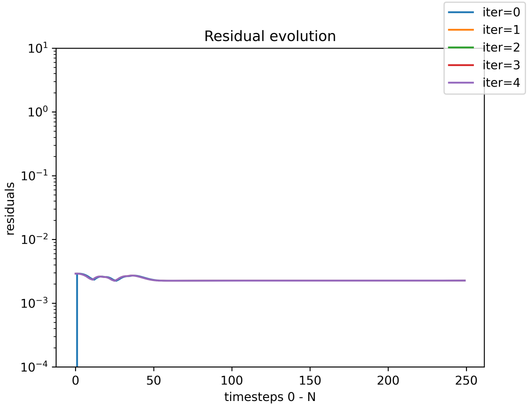


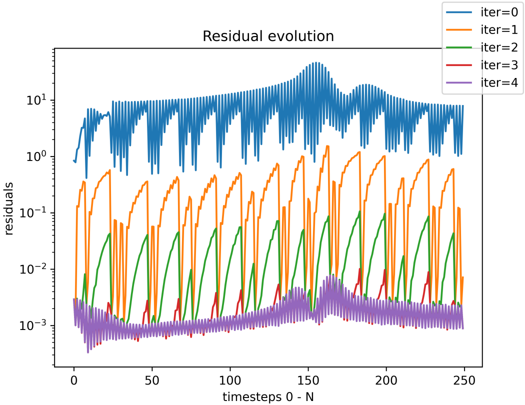
(a) Run with 1 MPI rank.
This is the same on CoolMUC-2

(b) Run with 4 nodes and 4 hosts per node, therefore max. 16 parallel timesteps.

Residuals - Static vs. dynamic resources

Docker cluster



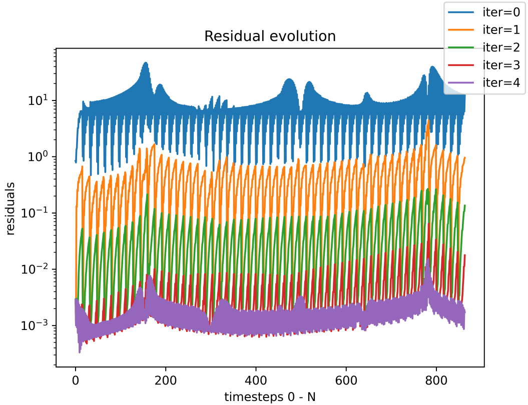


(a) Run with 1 MPI rank.
This is the same on CoolMUC-2

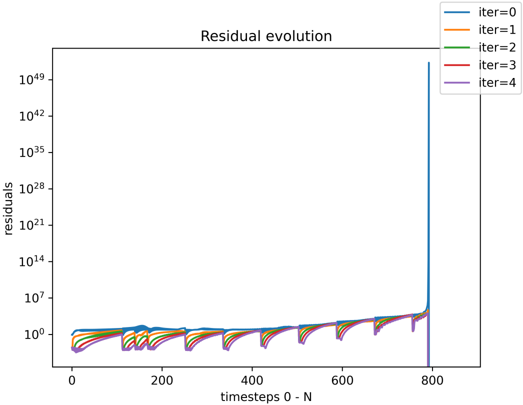
(b) Run with 4 nodes and 4 hosts per node, therefore max. 16 parallel timesteps.

Residuals - Static vs. dynamic resources

CoolMUC-2

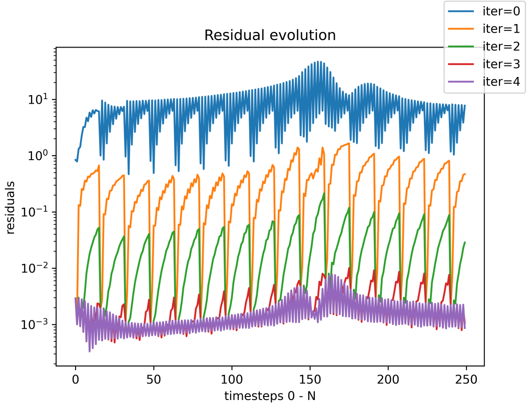


(a) Run with static 16 MPI ranks on a single node.

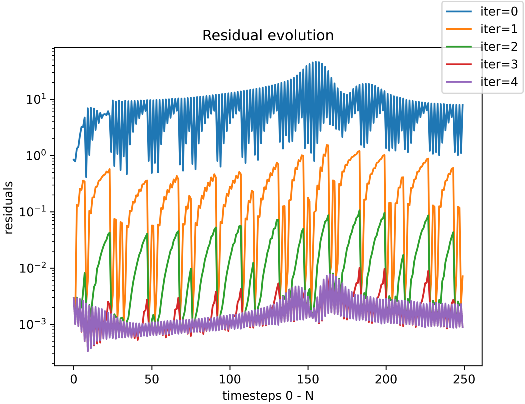


(b) Run with 4 dynamic nodes, with 56 cores each.

Residuals - Static vs. dynamic resources

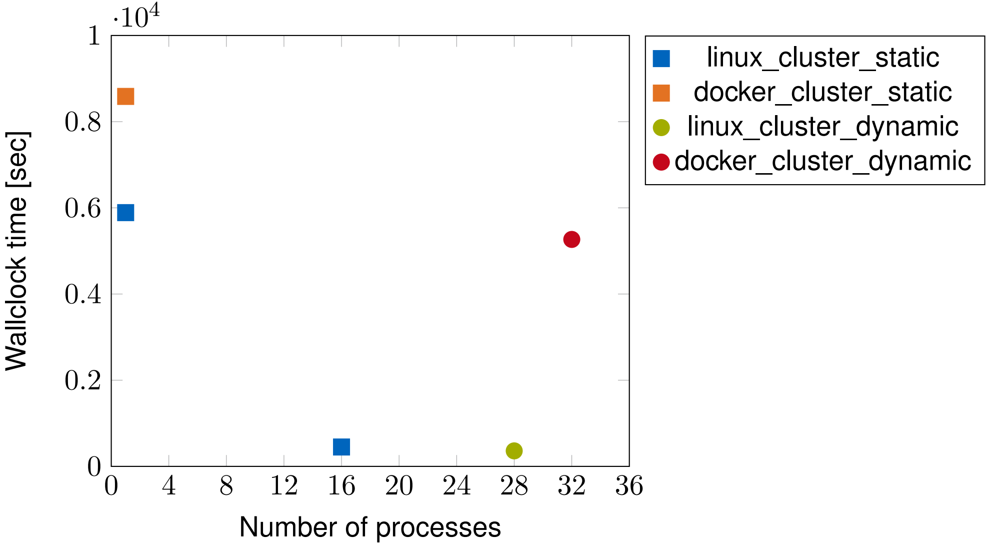


(a) Run with static 16 MPI ranks on a single node on the linux cluster.



(b) Run with 4 nodes and 4 hosts per node, therefore max. 16 parallel timesteps.

Wallclock time



Wallclock time

- Speed up on linux cluster = $\frac{Serial}{16 \;\; static \;\; procs} = \frac{5889}{450} \approx 13$
- Speed up on linux cluster = $\frac{Serial}{28 \;\; dynamic \;\; procs} = \frac{5889}{360} \approx 16$
- Speed up on docker cluster = $\frac{Serial}{32 \;\; dynamic \;\; procs} = \frac{8586}{5267} \approx 1.6$

Work in progress

1. Node granularity -
   - ☐ Adjust this constraint to enable the dynamic addition and removal of cores instead of entire nodes.

2. Resizing -
   - ☐ Incorporate convergence-informed resizing and therefore optimize the adaptivity criterion.

References

[1] Matthew Emmett and Michael L. Minion. Toward an efficient parallel in time method for partial differential equations. 2012. URL https://api.semanticscholar.org/CorpusID:53139971.

[2] Jan Fecht. 2023. URL https://fecht.cc/libpfasst-doc/showcase/.

[3] François P. Hamon, Martin Schreiber, and Michael L. Minion. Multi-level spectral deferred corrections scheme for the shallow water equations on the rotating sphere. *Journal of Computational Physics*, 376:435–454, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.09.042. URL https://www.sciencedirect.com/science/article/pii/S0021999118306442.

[4] Fabian Koehler. 2015. URL https://github.com/f-koehler/pfasst-tikz.