Taming the Beast of Dynamic Resource Management in HPC

Dominik Huber

Technical University of Munich

Papers & Portfolio

1. The Beast of Dynamic Resource Management

Dynamic Resource Management (DRM) improves key HPC system metrics through dynamic (re) assignment of system resources during runtime:

The Beast:

- DRM requires **adaptations across the whole HPC software stack**
- Many past attempts failed due to the **lack of a holistic design**

3. Dynamic Applications

Applications are first-class citizens in the HPC world. **Past attempts for DRM failed to cover the diversity of HPC applications**, preventing its adoption on production HPC systems. The **DPP design is based on an application-system co-design**, to support a wide range of application use cases:

PetSc [1] Parallel numerical software library	DynLAIK [13] Data distribution for dynamicity and fault tolerance	LibPFASST [3] Parallel-in-time integration library
P4est [2] Adaptive Mesh Refinement library	XBraid [4] Multigrid parallel-in-time integration	MPDATA3D [12] 3D semi-Lagrangian multiscale fluid solver
DMR-API [6] Dynamic resources API for computational kernels	AMT-GLB [11] Asynchronous Many-Task Runtime System	DPPInSitu [10] Library for dynamic, asynchronous in-situ techniques

2. DPP Software Stack [9, 7, 8]

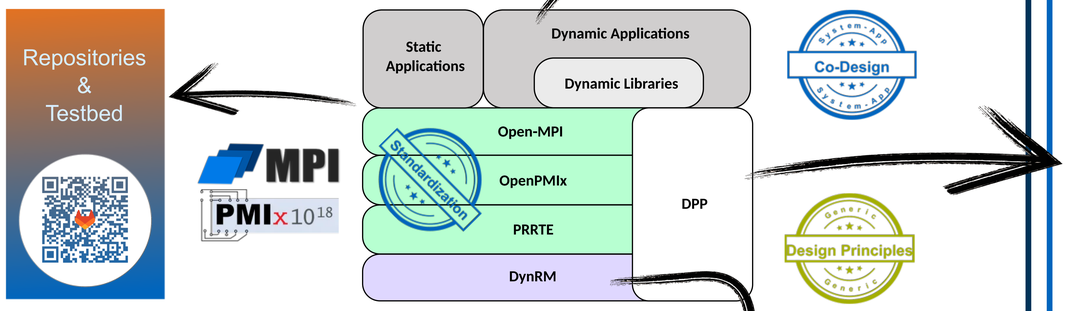

Figure 1 Software stack for DRM based on the DPP design.

4. Dynamic Processes With PSets (DPP) [8]

Full paper

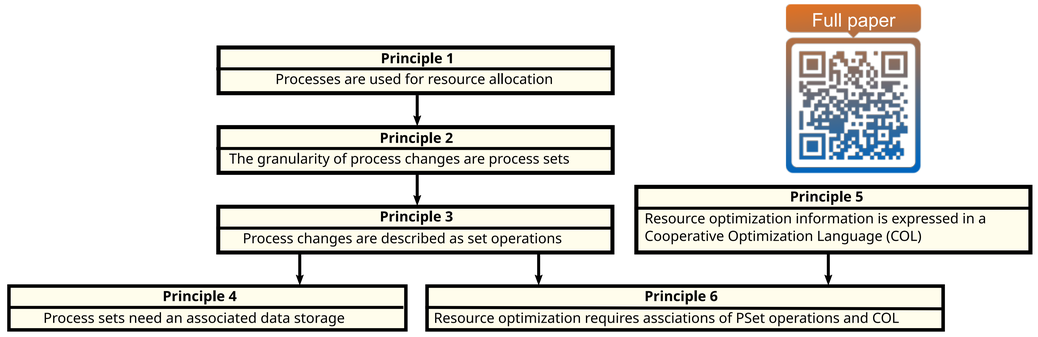

Figure 2 Design Principles for DRM in parallel programming models.

6. Experimental Results [5]

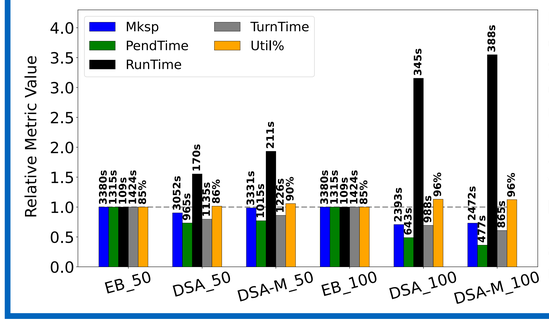

Figure 5 Comparison of system metrics for scheduling experiments on 100 compute nodes using EasyBackfilling (EB), Discrete Steepest Ascend (DSA), and DSA with monitoring (DSA-M) policies and a job mix of $N = 100$ jobs with 50% and 100% dynamic jobs, relative to EB50.

5. Dynamic Scheduling Optimization [5]

We model application scalability (Amdahl's Law) and use Discrete Steepest Ascent (DSA) on the resulting optimization problem:

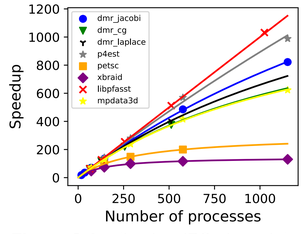

Figure 3 Application (PSet) models

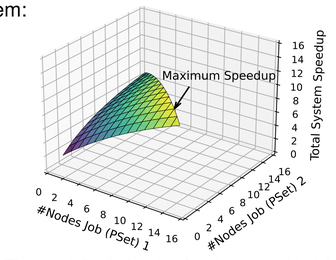

Figure 4 Optimization landscape with 2 jobs

7. Conclusions & Future Work

We developed a new, generic DRM design, which:

- is **applicable to standard system software** such as MPI and PMIx,
- covers a **diverse set of HPC applications** and libraries,
- enables **fine-grained performance-aware dynamic scheduling**, and
- achieves **improvements of key system metrics**, e.g., throughput and utilization.

Future Work:
1) Design and standardization of the COL
2) Improving programmability for application developers

I thank Mert Corumlu, Pierre-Francois Dutot, Tim Ellersiek, Jan Fecht, Keerthi Gaddameedi, Jan Grimm, Sergio Iserte, Yi Ju, Iker Martín-Álvarez, Jonas Posner, Martin Schreiber, Maximilian Streubel, Olivier Richard, and Bora Yilmazer for their support and contributions.

[1] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, **J.** Brown, P. Brune, K. Buschelman, E. M. Constantinescu, L. Dalcin, A. Dener, et al. *PETSc/TAO Users Manual V.3.21*. Tech. rep. Argonne National Laboratory (ANL), Argonne, IL (United States), Mar. 2024. DOI: 10.2172/2337606.

[2] Carsten Burstedde et al. "p4est: Scalable Algorithms for Par. Adaptive Mesh Ref. on Forests of Octrees". In: *Sci. Comput.* 33 (2011), pp. 1103–1133.

[3] Matthew Emmett and Michael Minion. "Toward an efficient parallel in time method for partial differential equations". In: *CAMCoS* 7 (Mar. 2012).

[4] Stefanie Günther, Nicolas R. Gauger, and Jacob B. Schroder. "A non-intrusive parallel-in-time adjoint solver with the XBraid library". In: *Comput. Vis. Sci.* 19.3–4 (2018), pp. 85–95.

[5] **Dominik Huber** and et al et. "Dynamic Resource Management in HPC systems using Dynamic Processes with PSets". In: *2025 IEEE 32nd International Conference on High Performance Computing, Data, and Analytics (HiPC)*. (under review). 2025.

[6] **Dominik Huber**, Sergio Iserte, Martin Schreiber, Antonio J. Peña, and Martin Schulz. "Bridging the Gap Between Genericity and Programmability of Dynamic Resources in HPC". In: *ISC'25 Research Papers*. 2025, pp. 1–11.

[7] **Dominik Huber**, Martin Schreiber, and Martin Schulz. "A Case Study on PMIx-Usage for Dynamic Resource Management". In: *High Performance Computing*. Cham: Springer Nature Switzerland, 2023, pp. 42–55.

[8] **Dominik Huber**, Martin Schreiber, Martin Schulz, Howard Pritchard, and Daniel Holmes. *Design Principles of Dynamic Resource Management for High-Performance Parallel Programming Models*. 2024. arXiv: 2403.17107.

[9] **Dominik Huber**, Maximilian Streubel, Isaías Comprés, Martin Schulz, Martin Schreiber, and Howard Pritchard. "Towards Dynamic Resource Management with MPI Sessions and PMIx". In: *Proceedings of the 29th European MPI Users' Group Meeting*. EuroMPI/USA '22. Chattanooga, TN, USA: Association for Computing Machinery, 2022, pp. 57–67.

[10] Yi Ju, **Dominik Huber**, Adalberto Perez, Philipp Ulbl, Stefano Markidis, Philipp Schlatter, Martin Schulz, Martin Schreiber, and Erwin Laure. "Dynamic Resource Management for In-Situ Techniques Using MPI-Sessions". In: *Recent Advances in the Message Passing Interface*. Cham: SN Switzerland, 2025, pp. 105–120.

[11] Jonas Posner, Tim Ellersiek, Nick Bietendorf, **Dominik Huber**, Martin Schreiber, and Martin Schulz. "Toward Dynamic Resource Management: An Asynchronous Many-Task (AMT) Runtime System leveraging Dynamic Processes with PSets (DPP)". In: *SN Computer Science* 5 (2025). under review.

[12] Krzysztof Rojek and Roman Wyrzykowski. "Performance modeling of 3D MPDATA simulations on GPU cluster". In: *The J. of Supercomputing* 73 (Feb. 2017).

[13] Zafer Yılmazer, **Dominik Huber**, Parab Arjun, Amir Raoofy, and Josef Weidendorfer. "Malleability in LAIK with MPI Dynamic Processes and PSets". In: *Euro-Par 2025: Parallel Processing Workshop Proceedings*. accepted. Berlin, Heidelberg: Springer-Verlag, 2025.