LUND UNIVERSITY

# A Layered Approach for Dynamic Resource Management in HPC

Dutot, Pierre-François; Fecht, Jan; Gaddameedi, Keerthi; Huber, Dominik; Iserte, Sergio; Minion, Michael; Schulz, Martin; Schreiber, Martin; Schüller, Valentina; Peña, Antonio J.; Richard, Olivier

2024

*Document Version:*
Publisher's PDF, also known as Version of record

Link to publication

*Citation for published version (APA):*
Dutot, P.-F., Fecht, J., Gaddameedi, K., Huber, D., Iserte, S., Minion, M., Schulz, M., Schreiber, M., Schüller, V., Peña, A. J., & Richard, O. (2024). *A Layered Approach for Dynamic Resource Management in HPC*. Poster session presented at 30th International European Conference on Parallel and Distribbuted Computing, Euro-Par 2024, Madrid, Spain.

*Total number of authors:*
11

**General rights**
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

**Take down policy**
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

# A Layered Approach for Dynamic Resource Management in HPC

Hans-Joachim Bungartz[1], Pierre-François Dutot[2], Jan Fecht[1], Keerthi Gaddameedi[1], Dominik Huber[1], Sergio Iserte[4], Michael Minion[5], Tobias Neckel[1], Martin Schulz[1], Martin Schreiber[3], Valentina Schüller[6], Antonio J. Peña[4], and Olivier Richard[2]

## Introduction

Dynamic Resource Management (DRM) allows for dynamic changes of the resources assigned to a job during its execution. DRM has gained considerable interest over the last years as it could provide many benefits to providers of HPC systems and their users, such as improving energy efficiency and throughput.
However, DRM requires changes and interactions throughout the whole HPC software stack. In this research poster, we highlight solutions for dynamic resources on different layers of the HPC software stack, including the OAR resource manager, the Dynamic Processes with PSets (DPP) approach, the Dynamic Management of Resources framework (DMR), and an example application given by an adaptive parallel-in-time integration method within the SWEET software using LibPFASST.
We illustrate how these approaches fit together to provide a holistic approach for dynamic resources throughout the HPC software stack as part of our joint effort.

## Dynamic Resources in the HPC Software Stack

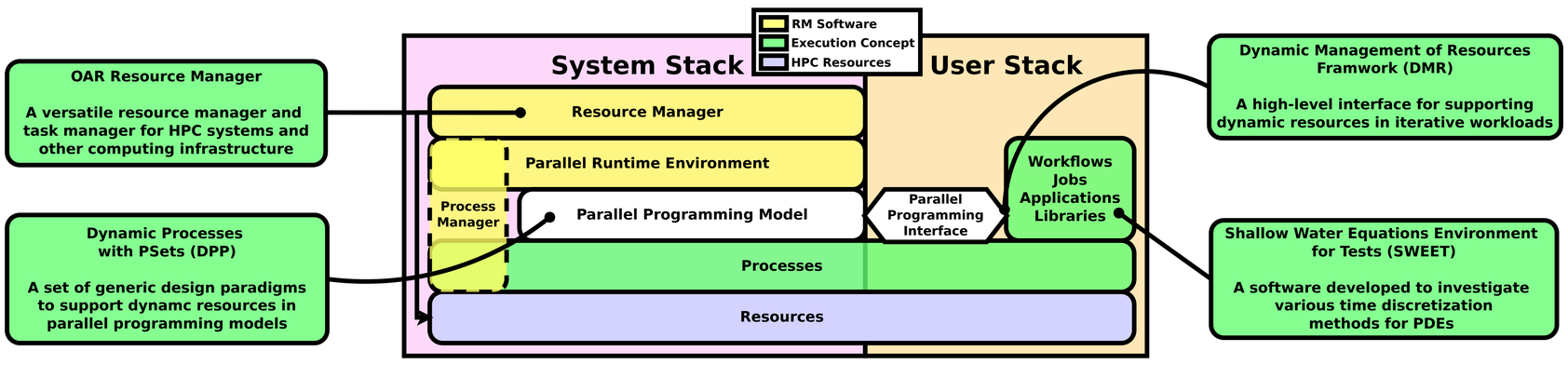



**Figure 1** Overview of targeted layers in the HPC software stack.

## 1. OAR Resource and Job Manager

OAR RJMS [5] introduces the **job envelope to support DRM**.

As resources are requested or freed, a new RJMS job is created, linked to the initial job envelope, replacing any previous job.

This creates a **sequence of jobs with different associated resources, forming a virtual global dynamic job**.

The different steps in this sequence are interaction points with other involved subsystems as illustrated in the sequence diagram in Fig. 2.

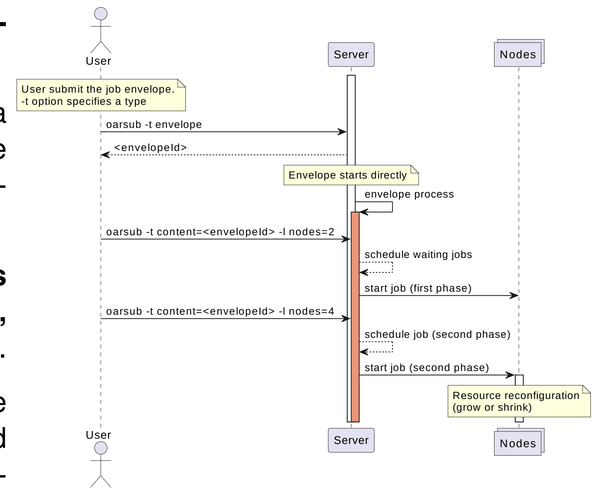



**Figure 2** Sequence diagram of dynamic job execution in OAR with two resource allocation requests.

## 2. Dynamic Processes with PSets (DPP)

Dynamic Processes with PSets (DPP) is a set of design paradigms (Fig. 3) for **generic dynamic resource support** in parallel programming models deduced from prior work [2, 3]. The DPP design paradigms are based on a **system-application co-design** and aim for a flexible and programming model agnostic abstraction. DPP has been realized as a **prototype based on Open-MPI, OpenPMIx, and PRRTE** (Fig. 4).

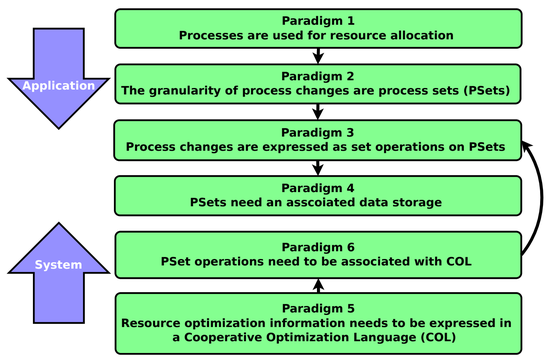



**Figure 3** DPP paradigms

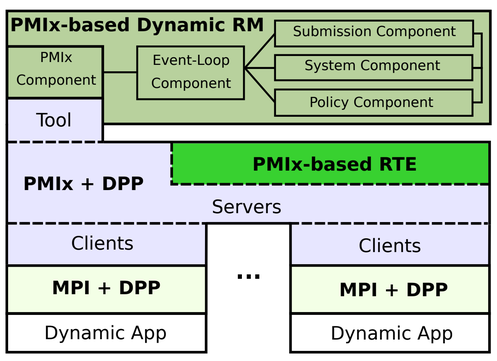



**Figure 4** DPP prototype

## 3. Dynamic Management of Resources (DMR)

The Dynamic Management of Resources (DMR) framework [4] is a high-level API that **facilitates the adoption of dynamicity in HPC codes**. It can abstract different MPI dynamic solutions (such as DPP) into the same syntax (Fig.5).

For instance, in an iterative code, DMR provides a series of operations around the **main loop**, which makes all the **dynamicity logic transparent** to the user (Fig. 6).

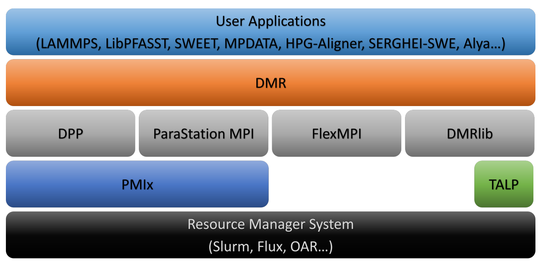


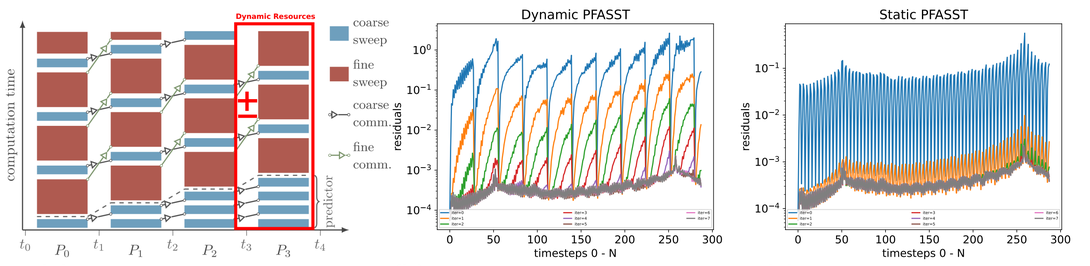

**Figure 5** DMR interaction with applications, runtimes, tools, and resource managers.

```c
void main() {
   DMR_INIT(user_init_data(), recv_expand());
   DMR_Set_parameter(MIN, MAX, SWEET_SPOT);
   DMR_Inhibit_iter(n_iters);
   for(it = 0; it < ITERATIONS; it++) {
       DMR_RECONFIGURATION( send_expand(),
                            recv_expand(),
                            send_shrink(),
                            recv_shrink());
       compute();
   }
   DMR_FINALIZE(user_free_data());
}
```

**Figure 6** Code example for adopting dynamicity with DMR in an iterative application.

## 4. Application example: LibPFASST and SWEET

**SWEET+LibPFASST**: PDE solver framework for mathematical and HPC research on time integration methods for prototypes of weather simulations [6]. It is integrated with the LibPFASST library [1] enabling parallel-in-time integration for the aforementioned prototypes.
**Current state:** Used the DPP approach to enable adaptivity in the LibPFASST library (Fig. 7a) to leverage dynamic resource management in SWEET.
**Future work**: Adaptively controlling resource allocation is crucial to improve convergence (Fig. 7b and 7c).




**(a)** Adaptive PFASST algorithm **(b)** Adaptive PFASST convergence **(c)** Static PFASST convergence

## Summary

As part of our joint effort, we work towards a fully integrated, holistic dynamic resource approach across all layers of the HPC software stack. By this, we aim to improve the efficiency and flexibility of resource usage on current and future HPC systems.

[1]TUM School of Computation, Information and Technology, Technical University Munich, Boltzmannstraße 3, 85748 Garching, Germany
[2]Laboratoire d'Informatique de Grenoble, Univ. Grenoble Alpes, Grenoble INP, CNRS, Inria, 38000 Grenoble, France
[3]Université Grenoble Alpes, Laboratoire Jean-Kuntzmann, Inria, 38000 Grenoble, France
[4]Barcelona Supercomputing Center (BSC)
[5]Lawrence Berkeley National Laboratory (LBNL), 1 Cyclotron Rd, Berkeley, CA 94720, USA
[6]Centre for Mathematical Sciences, Lund University, Sölvegatan 18, 223 62 Lund, Sweden

[1] Matthew Emmett and Michael Minion. "Toward an efficient parallel in time method for partial differential equations". In: *Communications in Applied Mathematics and Computational Science* 7.1 (2012), pp. 105–132. DOI: 10.2140/camcos.2012.7.105. URL: https://doi.org/10.2140/camcos.2012.7.105.

[2] Jan Fecht et al. "An Emulation Layer for Dynamic Resources with MPI Sessions". In: *High Performance Computing. ISC High Performance 2022 International Workshops*. Ed. by Hartwig Anzt et al. Cham: Springer International Publishing, 2022, pp. 147–161. ISBN: 978-3-031-23220-6.

[3] Dominik Huber et al. "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. ISBN: 9781450397995. DOI: 10.1145/3555819.3555856.

[4] Sergio Iserte et al. "DMRlib: Easy-Coding and Efficient Resource Management for Job Malleability". In: *IEEE Transactions on Computers* 70.9 (2021), pp. 1443–1457. URL: https://doi.org/10.1109/TC.2020.3022933.

[5] OAR-Team. *OAR: Versatile Resource and Job Manager (third generation)*. 2024. URL: https://github.com/oar-team/oar3 (visited on 01/22/2024).

[6] Martin Schreiber. *SWEET! Shallow Water Equation Environment for Tests, Awesome!* 2024. URL: https://sweet.gitlabpages.inria.fr/sweet-www/.