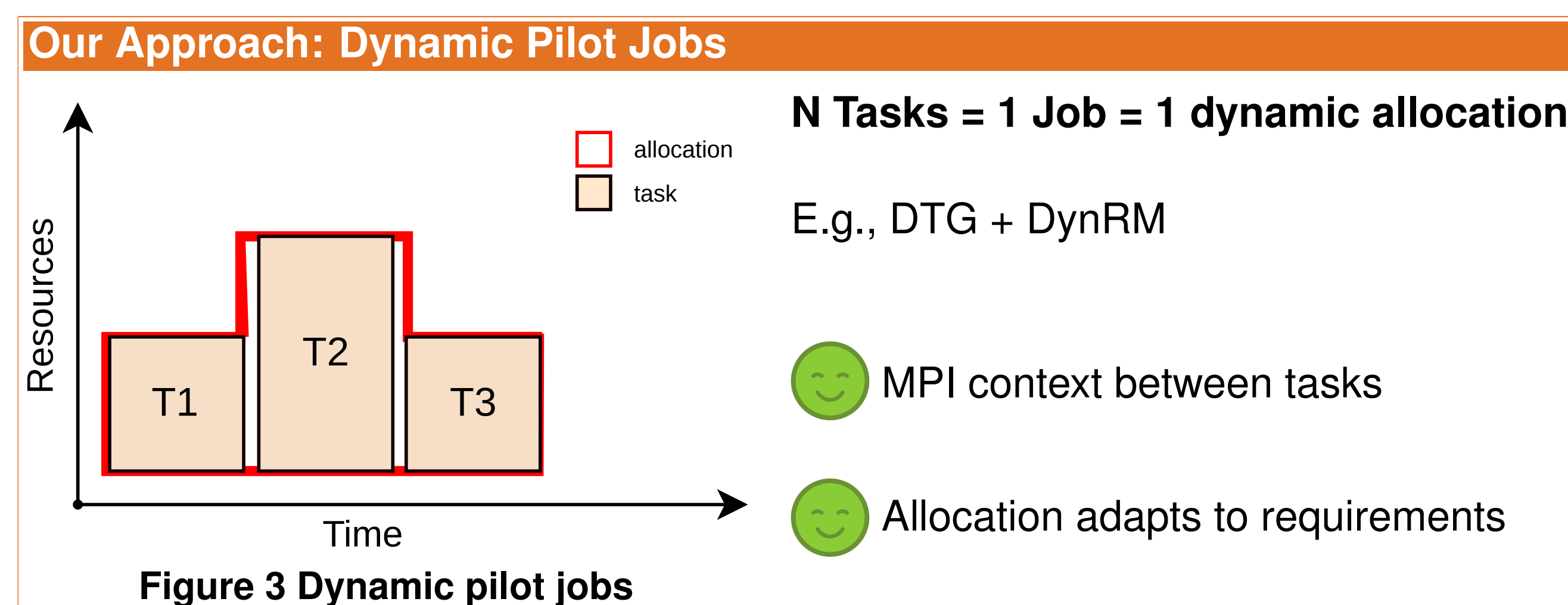
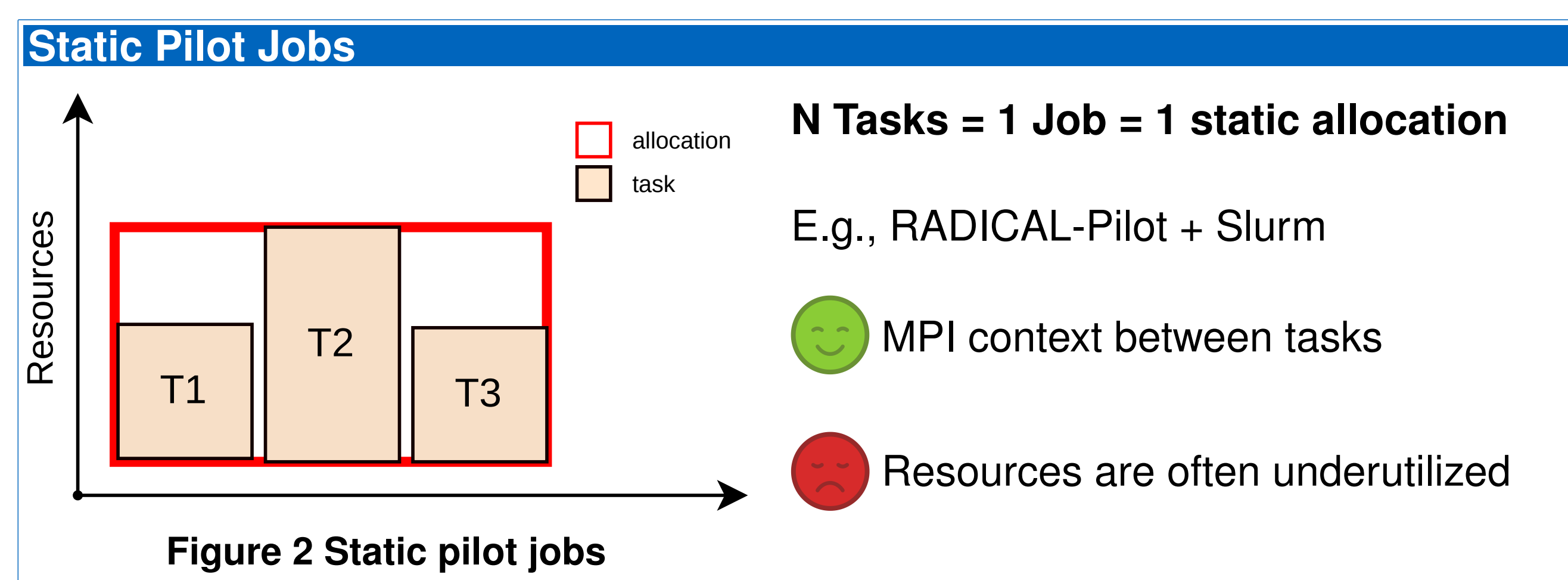
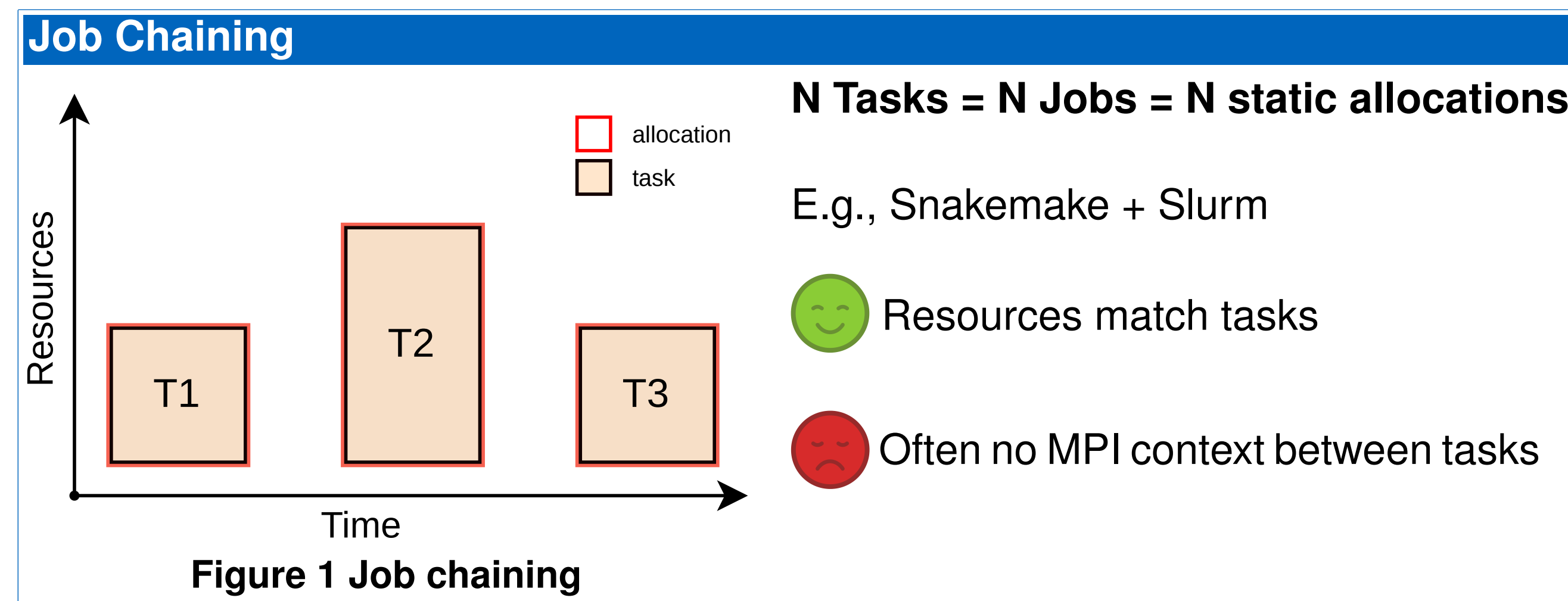


A Dynamic Execution Model for HPC Task Graphs Using MPI Sessions and DPP



Dominik Huber, Ali Berat Cetin, Martin Schreiber, Martin Schulz

1. Problems of HPC Workflows Execution Models



2. Dynamic Task Graphs with MPI Sessions & DPP

Dynamic resource management (DRM):

- Allows **changing resources** allocated to a job at runtime.
- Requires **interactions across the HPC software stack**.

Dynamic Processes with PSets (DPP) [2, 1]:

- Abstraction** to express the **dynamic process structure** of applications.
- Allows **dynamic resource reallocation** by the resource manager.

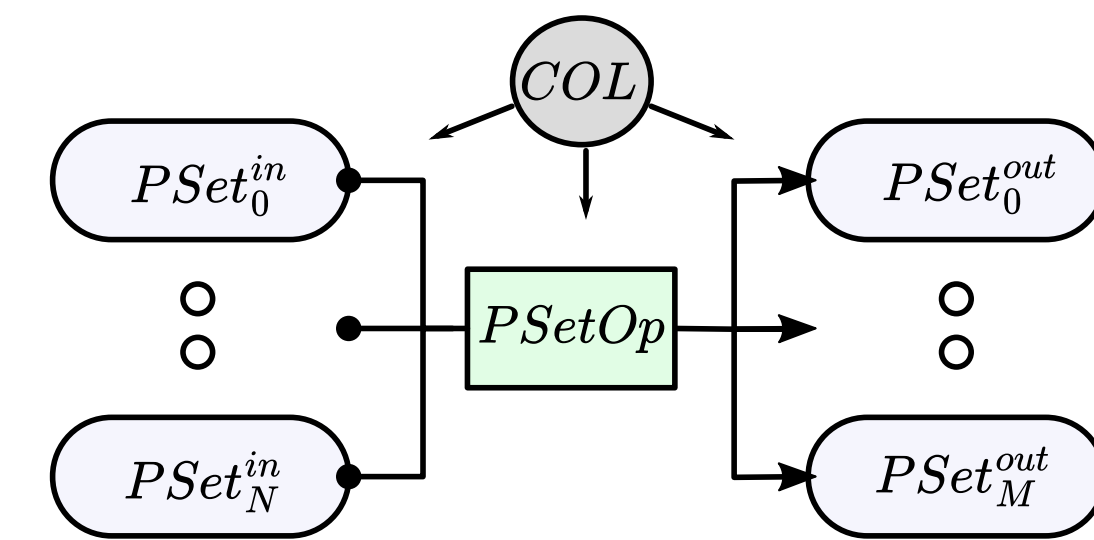


Figure 4 Graph-based abstraction for DRM. Vertices = sets of application processes (PSets), hyperedges = reconfiguration operations (PSetOps).

MPI Session extensions for DPP are under discussion in the MPI Sessions WG

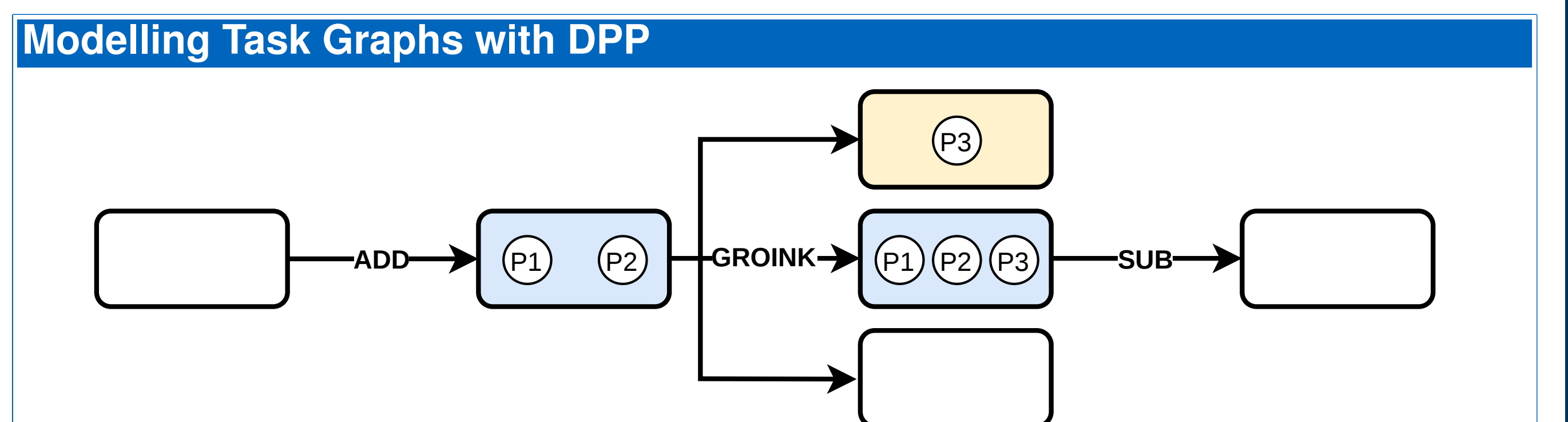


Figure 5 Task execution expressed as PSets and PSetOps. Different PSetOps describe a) Task Launch (ADD), b) Task Resize, i.e., grow (GROINK), and c) Task Termination (SUB).

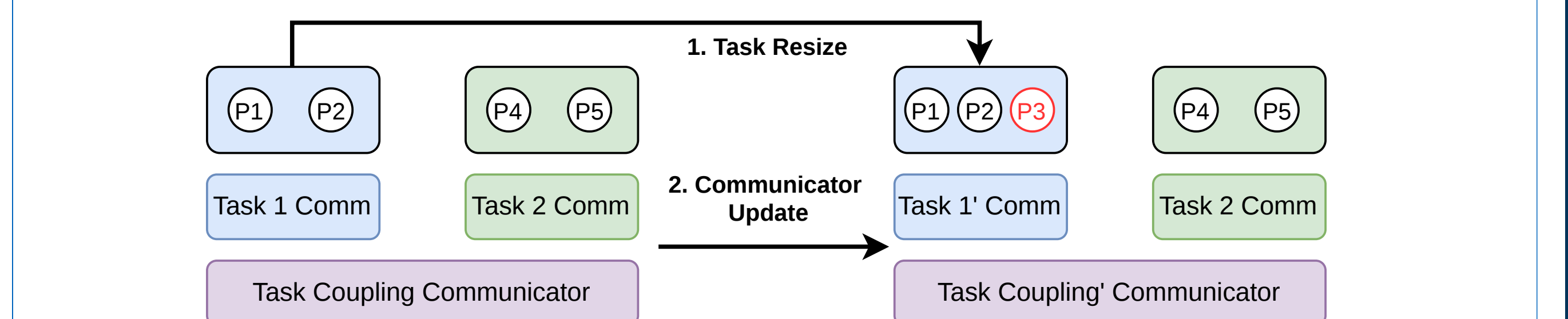


Figure 6 Task Couplings. Intra- & inter-task communicators adapt to resource changes.

3. The Dynamic Task Graphs (DTG) Library

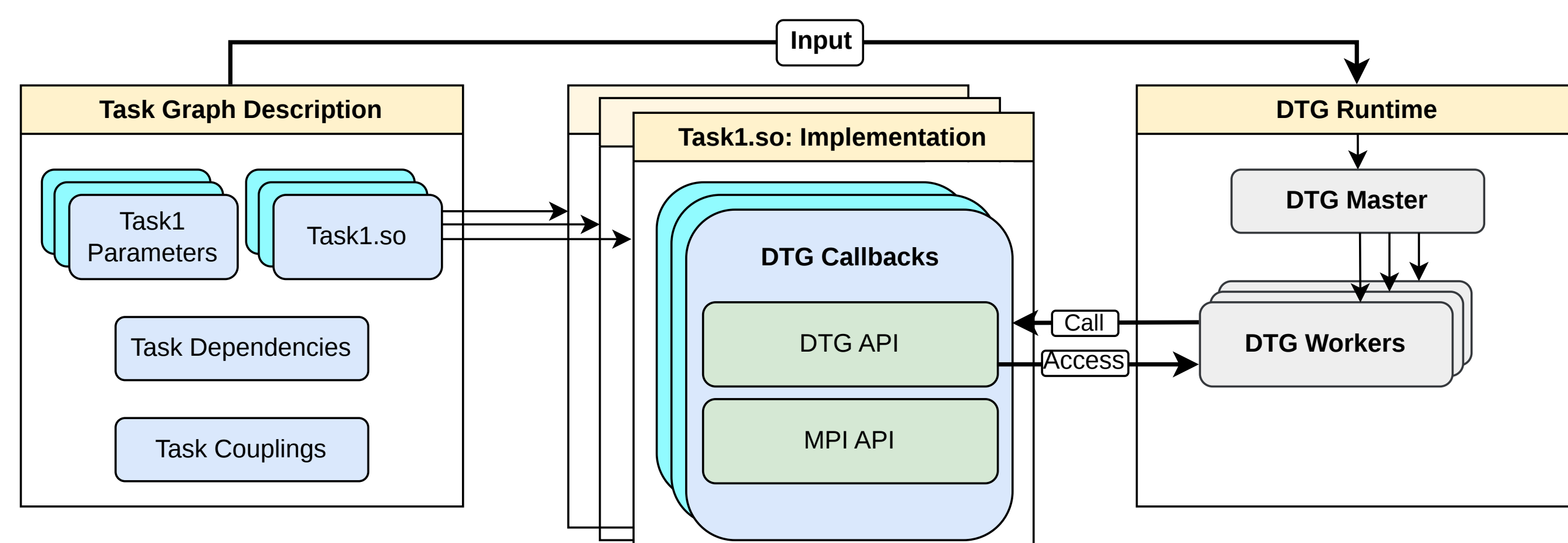


Figure 7 DTG Library user perspective. Users provide a JSON task graph description (left) and task implementations as shared libraries (middle). The DTG runtime (right) spawns workers that invoke user-defined callbacks, which use the DTG and MPI APIs.

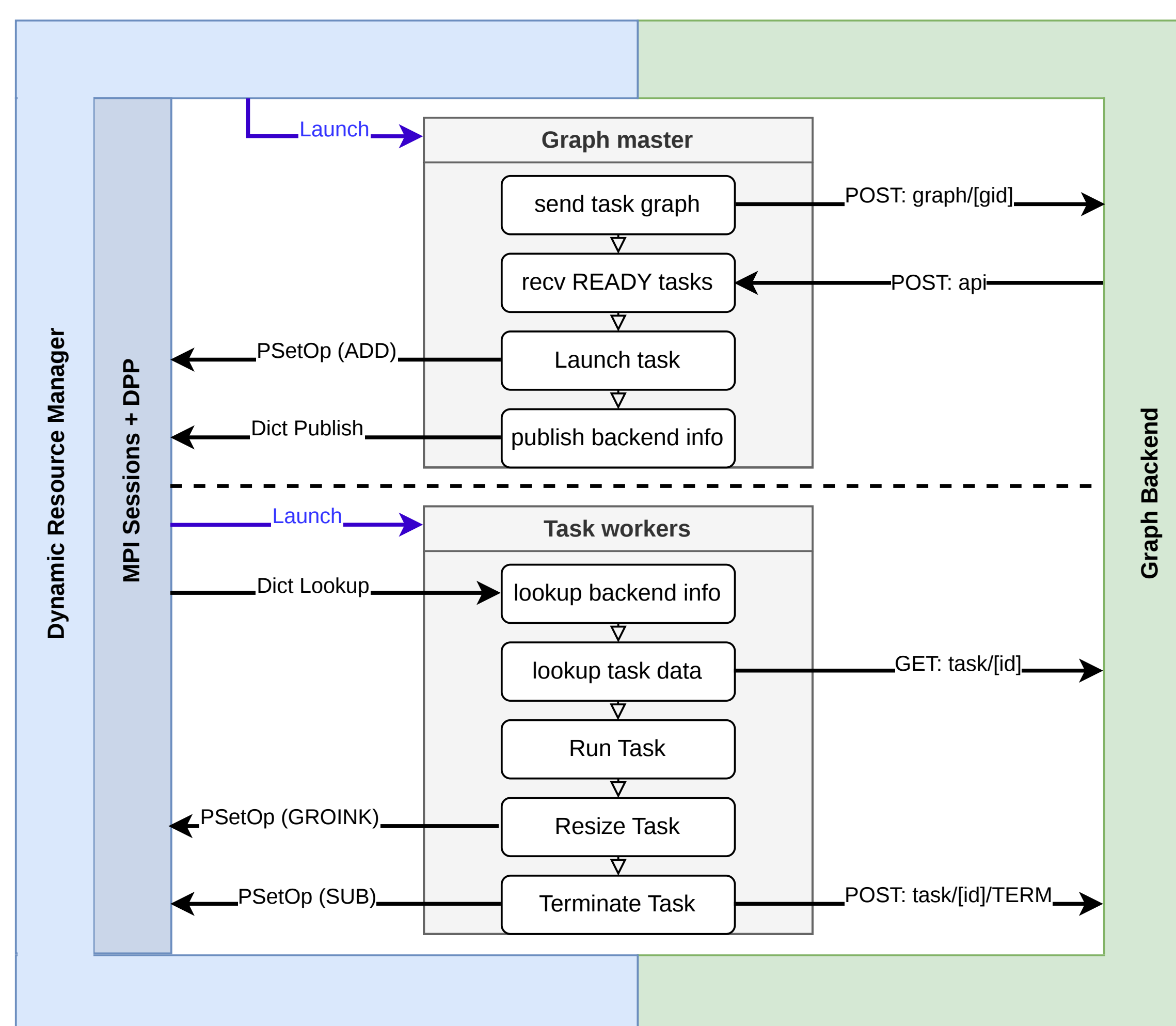


Figure 8 DTG Runtime architecture. It uses MPI Session + DPP to interact with the RM.

4. Benefits of the Dynamic Execution Model

Experimental setup:

- Task workload:** MPDATA3D (3D fluid dynamic solver)
- Task graphs:** Adaptive sampling with pruning and coupled simulations
- Test system:** Grid5000 *gros* cluster (14 CPU nodes)

1. Adaptive Sampling: Improved resource efficiency vs. static pilot jobs!

- Master launches N sampling tasks/epoch, prunes k based on intermediate results.
- Master launches N - k analysis tasks/epoch.

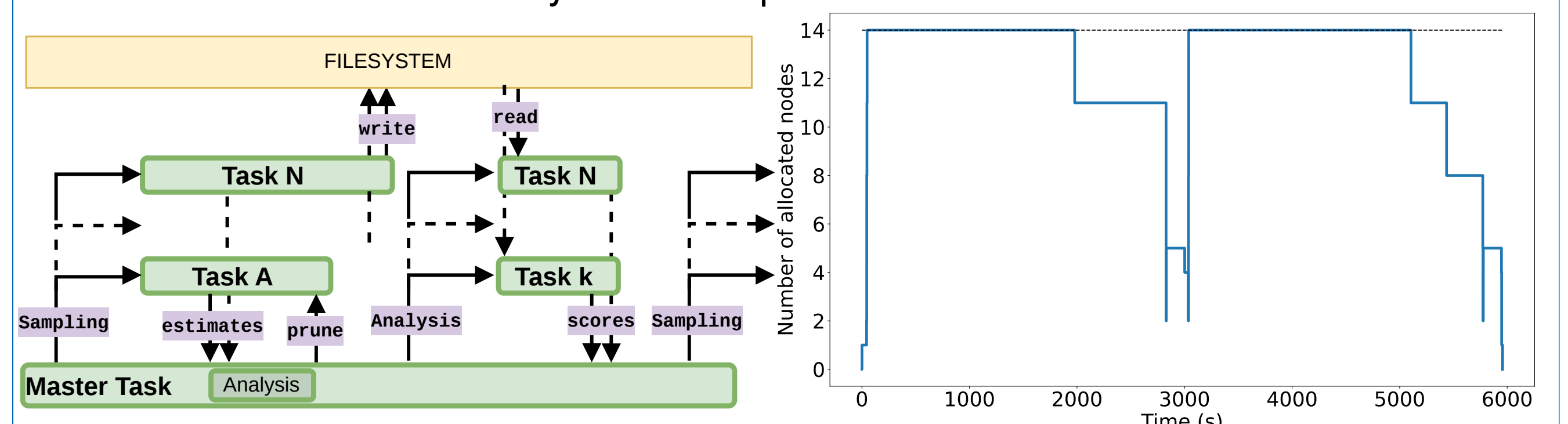


Figure 9 Adaptive Sampling Task graph (left) and resource allocation over time (right).

2. Coupled simulations: Per-simulation resource adaptivity!

- 2 simulation tasks exchange data every k steps; TaskB halves its step size (iter. 18)
- Sync overhead in iter. 21 → Double the number of processes in TaskB to resync

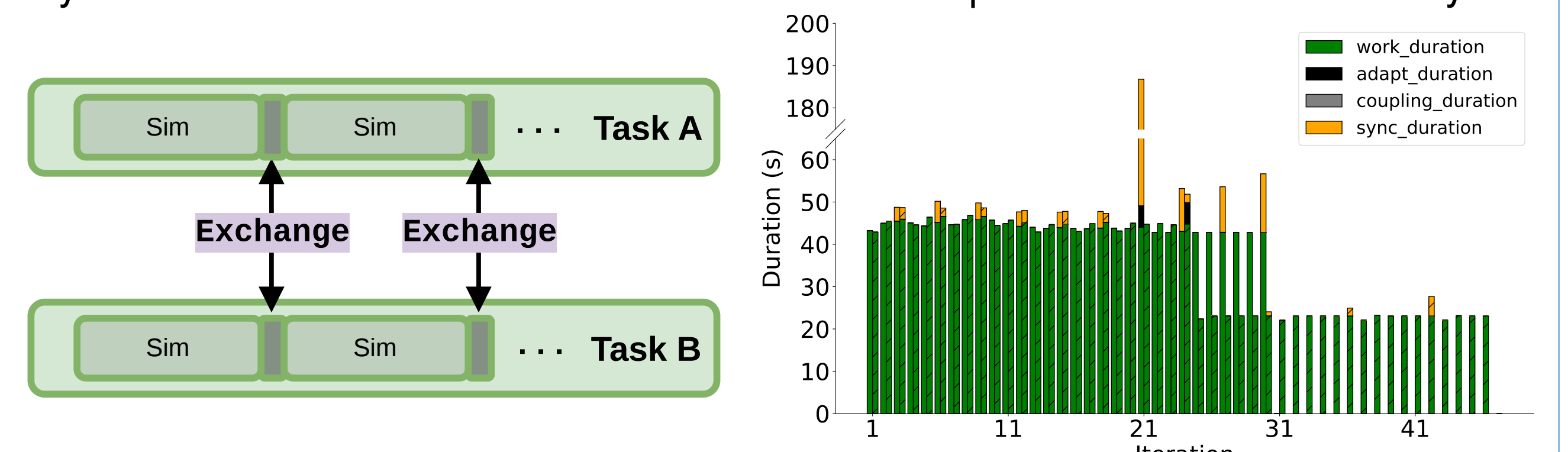


Figure 10 Coupled Simulations Task graph (left) and per-iteration times (right).

Conclusion: MPI Sessions + DPP enable dynamic HPC execution models!

- Overcomes limitations of static MPI World model**
- Real dynamic process management** (no MPI_Comm_spawn!)
- Dynamic resource allocation** to improve resource efficiency
- Future work:** Contribute to standardization in MPI and PMIx