MAESTRO
DATA ORCHESTRATION

Middleware for Memory and Data-Awareness in Workflows

This project has received funding from the European Union's Horizon 2020 research and innovation program through grant agreement 801101.

Authors
Prof. Dirk Pleiter
Dr. Manuel Arenaz
Dr. Utz-Uwe Haus
Dr. Marc Perache

www.maestro-data.eu

@maestrodata

/company/maestrodata

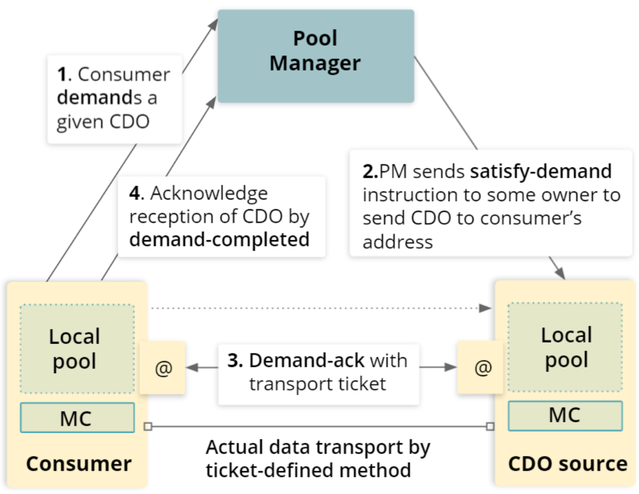
Motivation

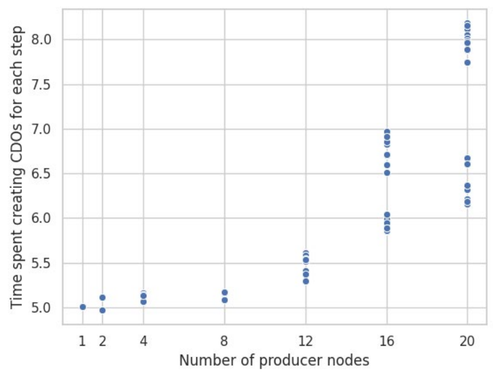
- HPC and HPDA workloads are more and more I/O-intensive
- Performance bottlenecks are usually in the memory and storage systems
- Reducing and minimising data movement is very hard in general
- The HPC software stack was designed in a different era, to solve a different problem

The Maestro consortium is building a middleware library that characterises application data, reasons about how to load and store that data, assesses the cost of moving it and automates data movement across diverse memory systems

Flexible Data Store and Transport

- CDOs could stay at any memory/storage layer
- Initial support for volatile memory, global file systems and object stores
- Support of different data transport mechanisms
  - Shared-memory via MAMBA
  - Network via UDJ
  - Global file system via POSIX I/O, MPI I/O
  - Object store via Mero using the Maestro I/O API (MIO)

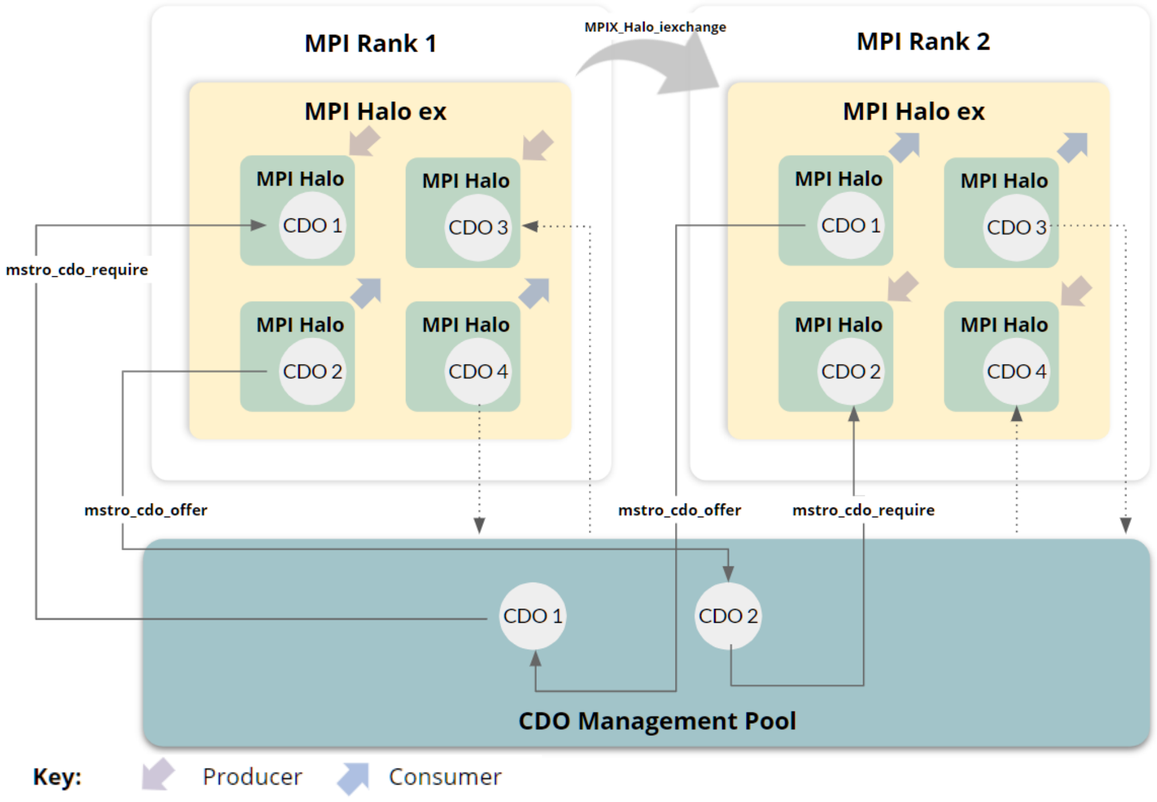




Time required for creating 22236 CDOs per node on a cluster with up to 20 nodes using an I/O test application that mimics a weather forecast application. The benchmark shows that the pool manager allows for more than 50,000 CDOs being created per second.

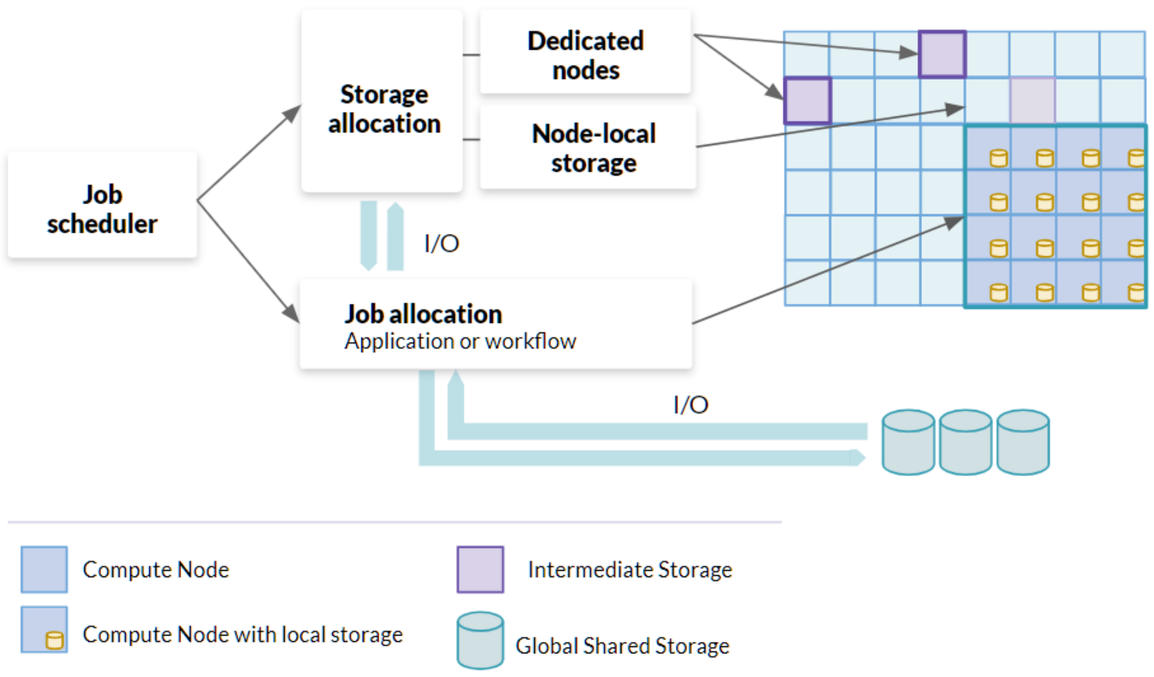
MPI Halo exchange process

To demonstrate data transport capabilities, test its robustness and explore performance, we implemented MPI_Halo, an MPI abstraction for halo exchanges, on top of Maestro as shown in the figure below. To test this version of MPI_Halo we used the mini-app Hydro.



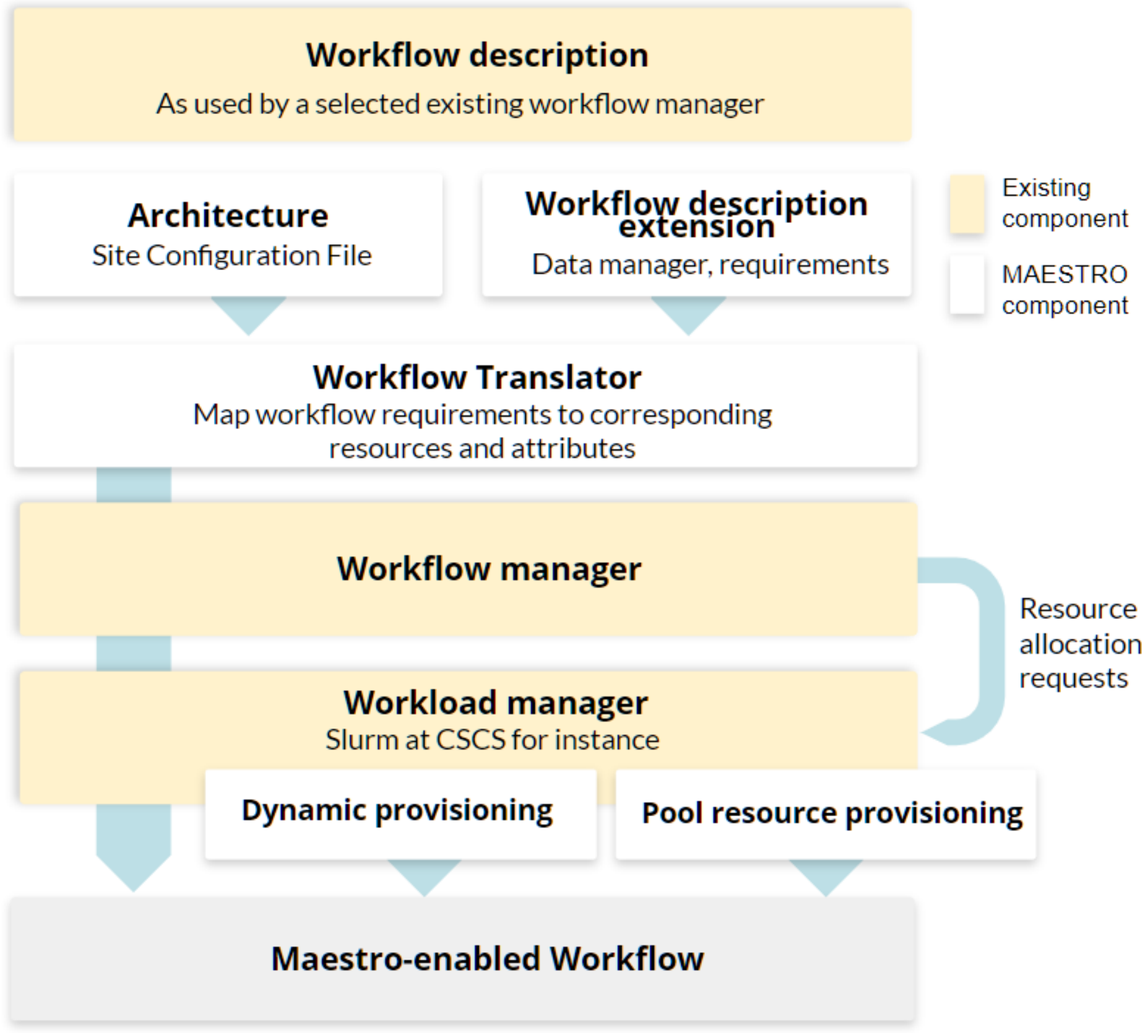
Dynamic resource provisioning mechanism

A solution has been developed that allows to provide access to dedicated storage resources through a dynamically deployed data manager. It has been successfully tested on Piz Daint using the MinIO object store as data manager on top of Cray DataWarp nodes.



Workflow execution framework

Efficient execution of workflows often depend on the ability of efficient data exchange between different components as well as data-aware scheduling. Maestro addresses this challenge by augmenting workflow descriptions that feed into a workflow translator. During this step the augmented workflow description is mapped to available resources to optimise workflow execution.



Run time: 3-year project, started in September 2018

Current status: First prototypes and implementations of key components available: Maestro Core, Maestro I/O, Maestro Dynamic Provisioning and Maestro Workflow Translator

Next steps: Implement Maestro demonstrators based on available components and add smartness features to the middleware

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE · cea · appentra code performance · ETH zürich · CSCS · ECMWF · SEAGATE · CRAY a Hewlett Packard Enterprise company