Modeling Allocation of Heterogeneous Storage Resources on HPC Systems

Julien Monniot, François Tessier, Gabriel Antoniu

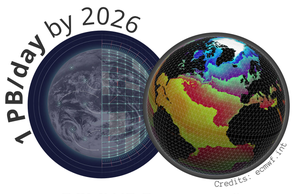
Modeling the Allocation of Heterogeneous Storage Resources on HPC Systems

PhD Student:
Julien Monniot - *julien.monniot@inria.fr*
INRIA Rennes / Université Rennes 1

Advisors:
François Tessier - INRIA Rennes
Gabriel Antoniu - INRIA Rennes

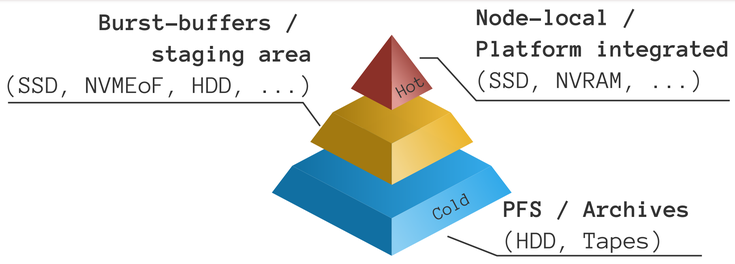
CONTEXT AND MOTIVATIONS

SKA Telescope



ECMWF Forecast & Digital Twin Earth

ITER Reactor



Examples:

Perlmutter *35PB* ***all flash storage*** *at NERSC (2021)*

Aurora *DAOS Storage with Intel Optane* ***persistent memory*** *at ALCF (2022)*

Summit *compute nodes with 1.6TB of* ***NVRAM*** *at OLCF (2018)*

- **Data deluge** from new large-scale scientific workflows
- ⬈ PFlops ⬊ TBps

- **Deeper storage hierarchy**
- New underlying **technologies**
- **Hybrid** platforms / workflows

⬈ **Complexity** & **Underutilization** of resources

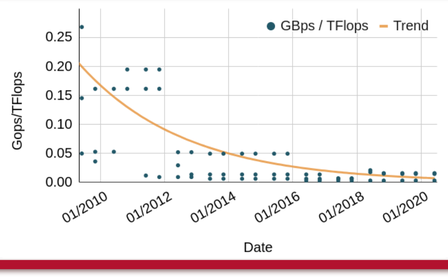
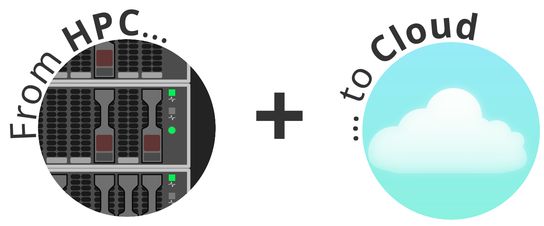


Figure 1. Top500: I/O bandwidth to computing power ratio for the top 3 HPC platforms, last 10 years.



MAIN AXES OF WORK

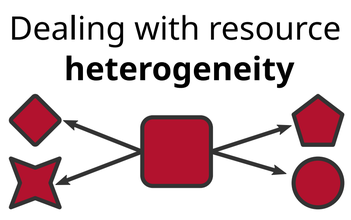
- Design of **scheduling algorithms** for storage allocations
- **Representation** of heterogeneous storage infrastructures
- Analysis of **storage related metrics**

Enable dynamic allocation of heterogeneous storage resources (real or simulated)

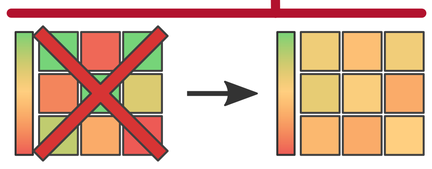
WHAT DO WE AIM FOR?

Optimal Resource **Scheduling** ②

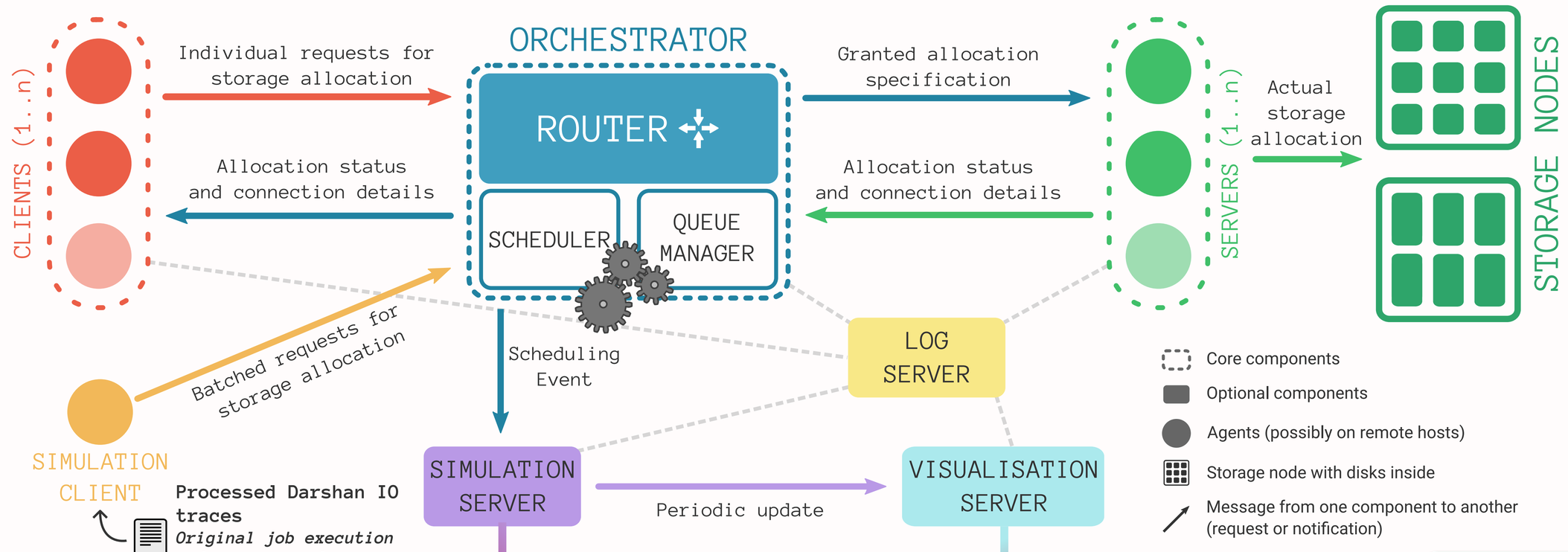
Dealing with resource **heterogeneity** ③



① **Fair** and **efficient** use of resources



SOLUTION DESIGN



Processed Darshan IO traces
Original job execution order preserved

STORALLOC

Storage-aware job scheduler simulator

Extensible with new algorithms

Independent components design

Common messaging interface

Abstraction of heterogeneous storage hardware

SIMULATION RESULTS

Showcase problem: *What is the correct sizing for a burst buffer capacity, when accounting for the effect of a storage scheduling algorithm and strategy (splitting large requests)?*

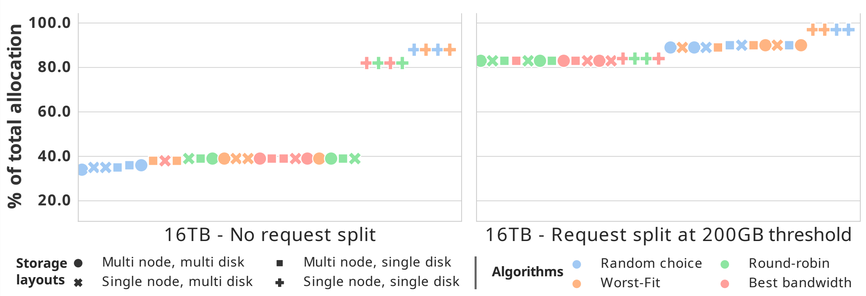


Figure 2. Percentage of the sum of the requested capacities (from all requests of dataset) that could be successfully allocated during simulation. One marker per simulation run, for a 16TB platform. Results show the algorithm and layout in use, and are grouped by split strategy.

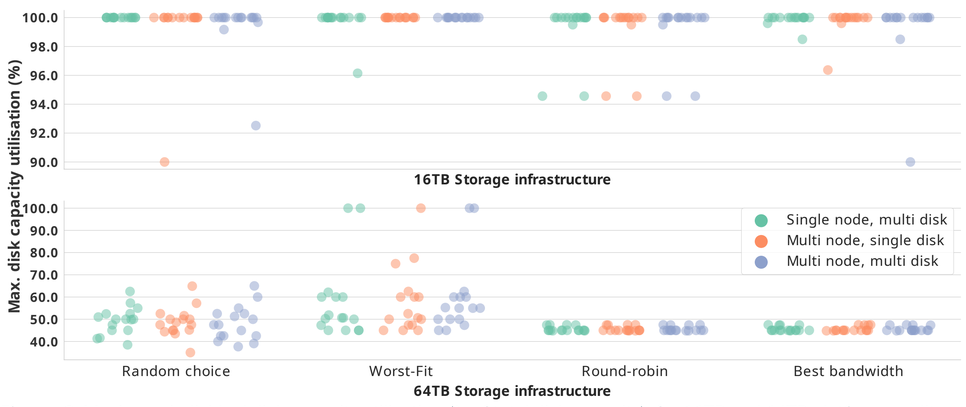
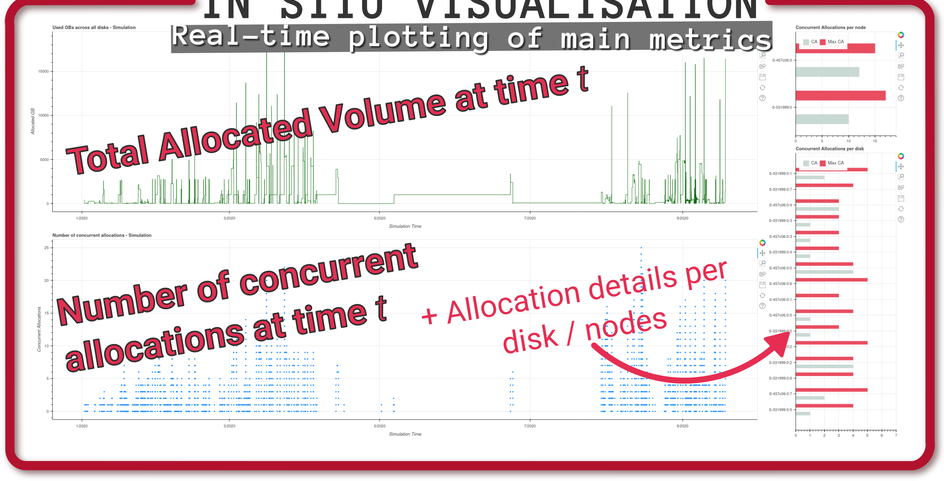


Figure 3. Maximum individual disk capacity utilization (% of total disk capacity), for 16TB and 64TB platforms, split strategy enabled at 200GB. We can confirm 16TB is too small (most disks are full), but the next size (64TB) would be underutilized (very few disks are used at more than 60% of their capacity).

IN SITU VISUALISATION



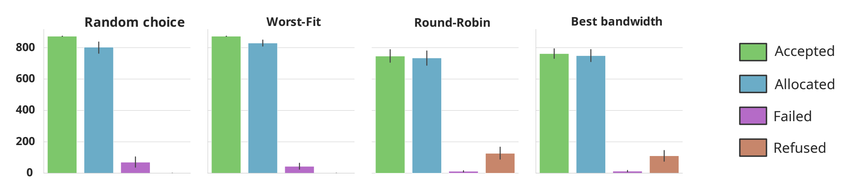


Figure 4. Number of split requests accepted, allocated, failed (during simulation) and refused (before simulation), grouped by algorithm. Random choice and Worst-fit are more permissive and rarely refuse requests, which eventually leads to more allocations. On the other hand, they result in more failed requests (error from the algorithm caught during allocation), which could prove coslty for the user.

Conclusion: *The required storage capacity is not far over 16TB. Splitting the requests and using an algorithm well suited for this strategy is however necessary (random choice or worst-fit is this case).*

IMPLEMENTATION

- Python3
- SimPy (DES simulation)
- ZMQ (Messaging)
- bokeh (Real-time plotting)
- seaborn (Plotting)
- Darshan (IO traces)

GitHub

hephtaicie/storalloc