



HAL
open science

Modeling Allocation of Heterogeneous Storage Resources on HPC Systems

Julien Monniot, François Tessier, Gabriel Antoniu

► **To cite this version:**

Julien Monniot, François Tessier, Gabriel Antoniu. Modeling Allocation of Heterogeneous Storage Resources on HPC Systems. SC 2022 - International Conference for High Performance Computing, Networking, Storage, and Analysis (Posters), Nov 2022, Dallas, United States. , pp.1-1. hal-03878252

HAL Id: hal-03878252

<https://hal.science/hal-03878252v1>

Submitted on 29 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Modeling the Allocation of Heterogeneous Storage Resources on HPC Systems

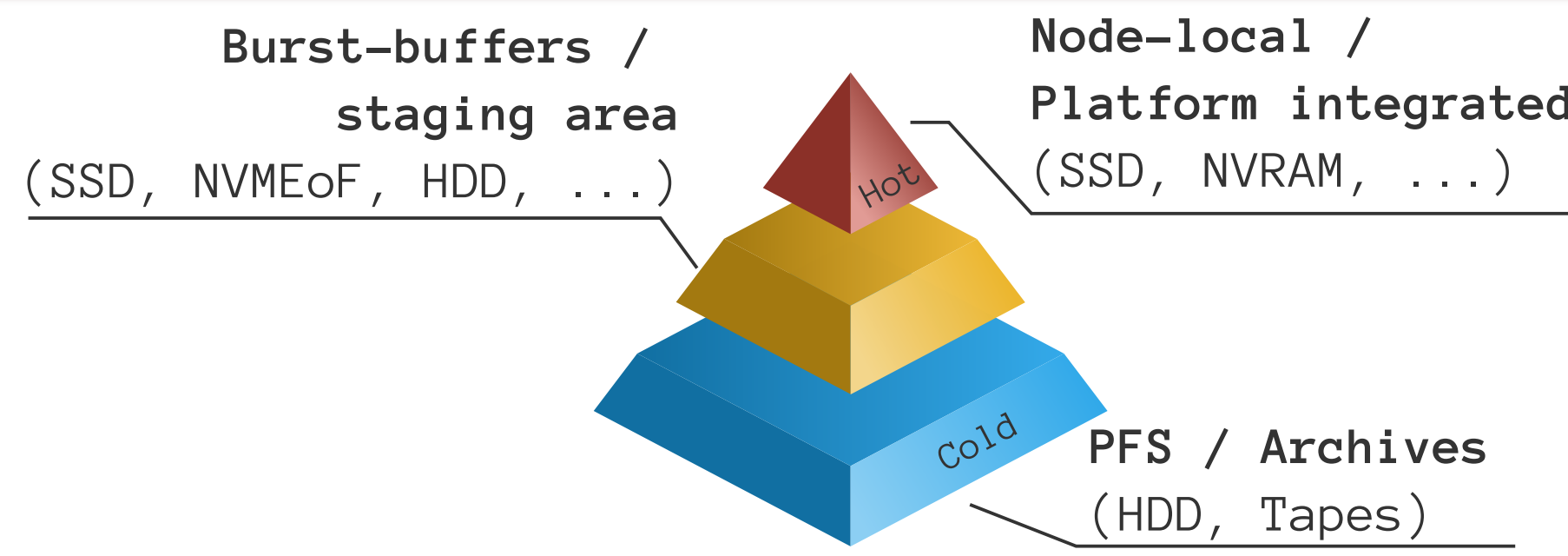
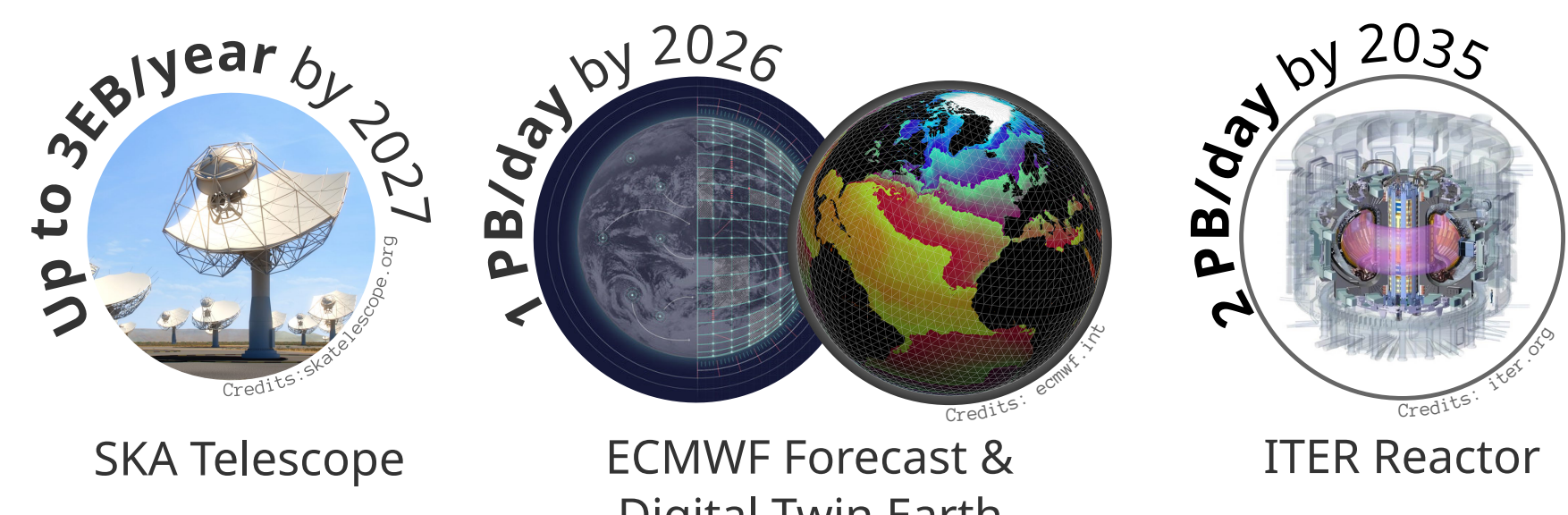
PhD Student:
Julien Monniot - julien.monniot@inria.fr
INRIA Rennes / Université Rennes 1



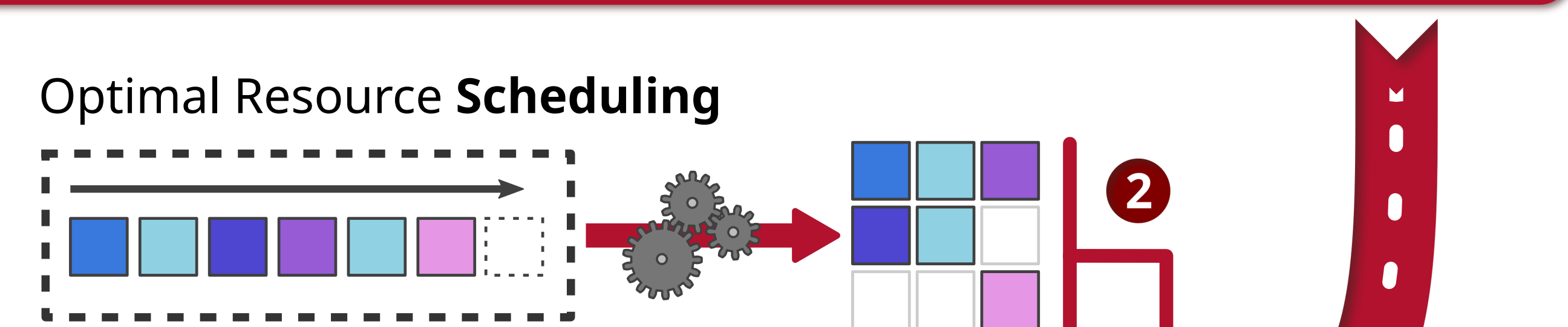
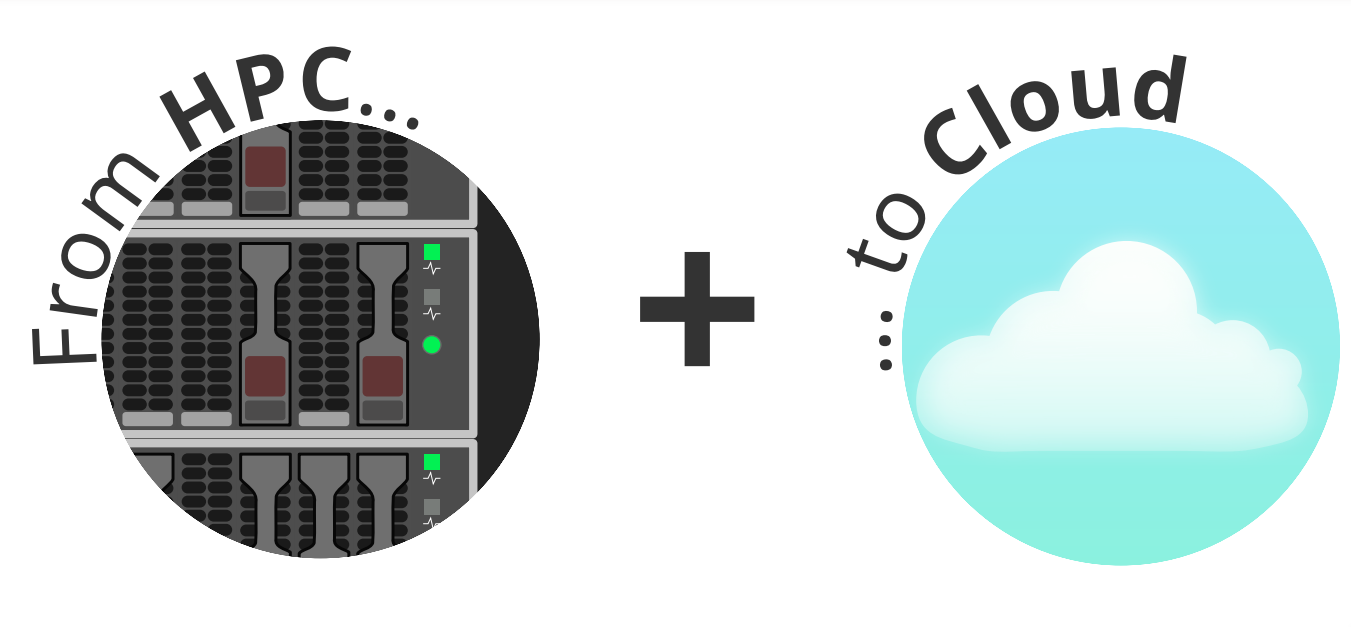
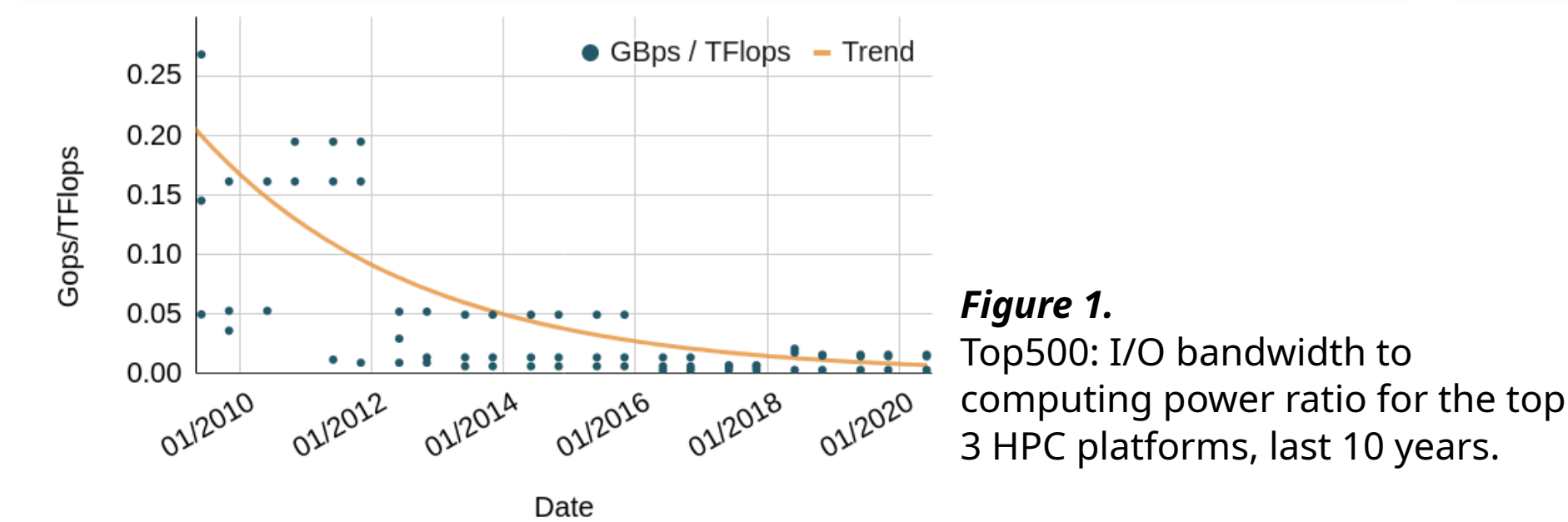
Advisors:
François Tessier - INRIA Rennes
Gabriel Antoniu - INRIA Rennes



CONTEXT AND MOTIVATIONS



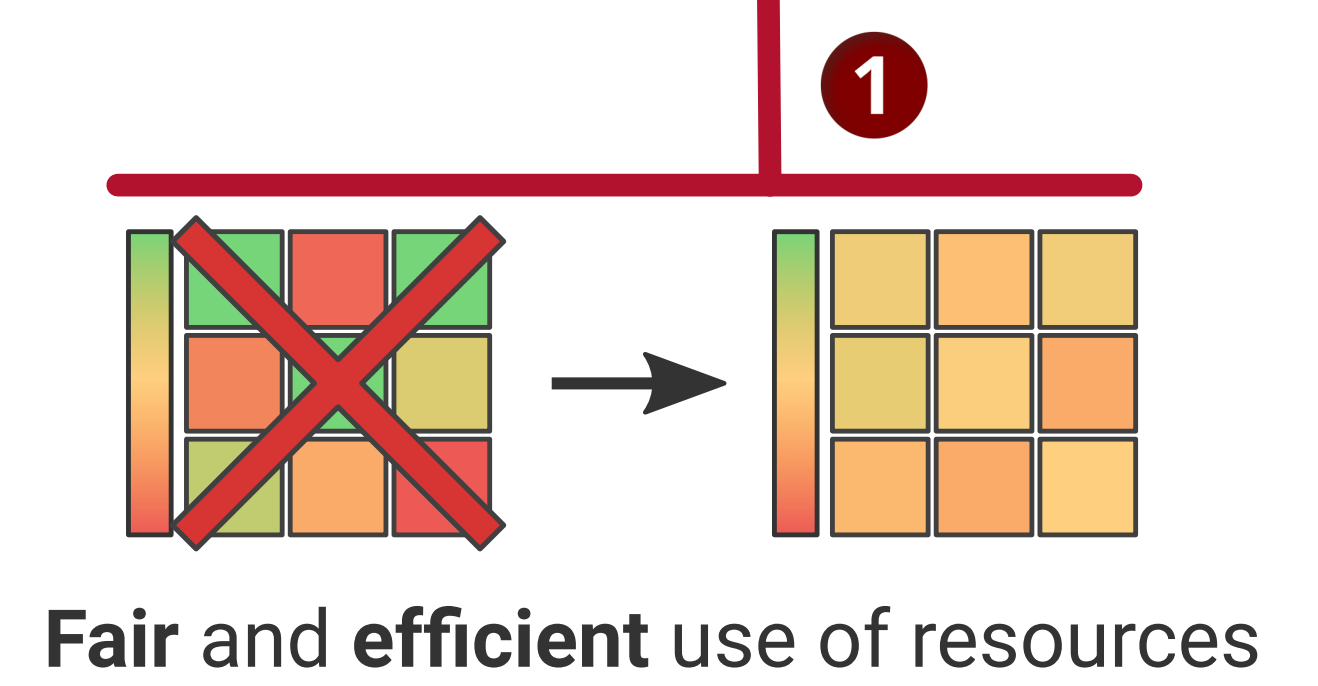
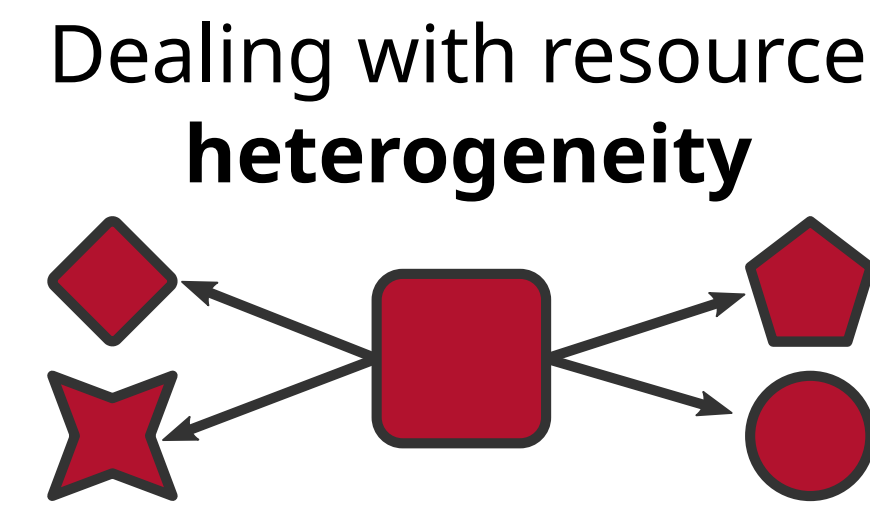
Examples:
Perlmutter 35PB all flash storage at NERSC (2021)
Aurora DAOS Storage with Intel Optane persistent memory at ALCF (2022)
Summit compute nodes with 1.6TB of NVRAM at OLCF (2018)



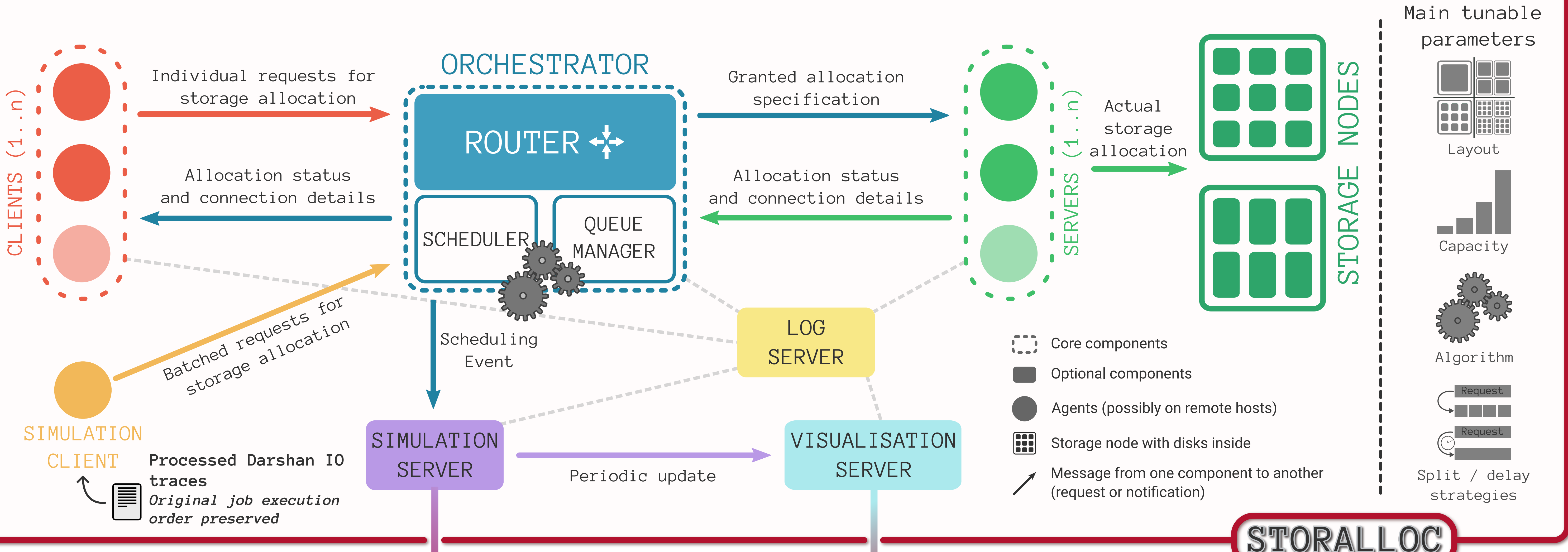
MAIN AXES OF WORK

WHAT DO WE AIM FOR?

- 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)



SOLUTION DESIGN



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)?

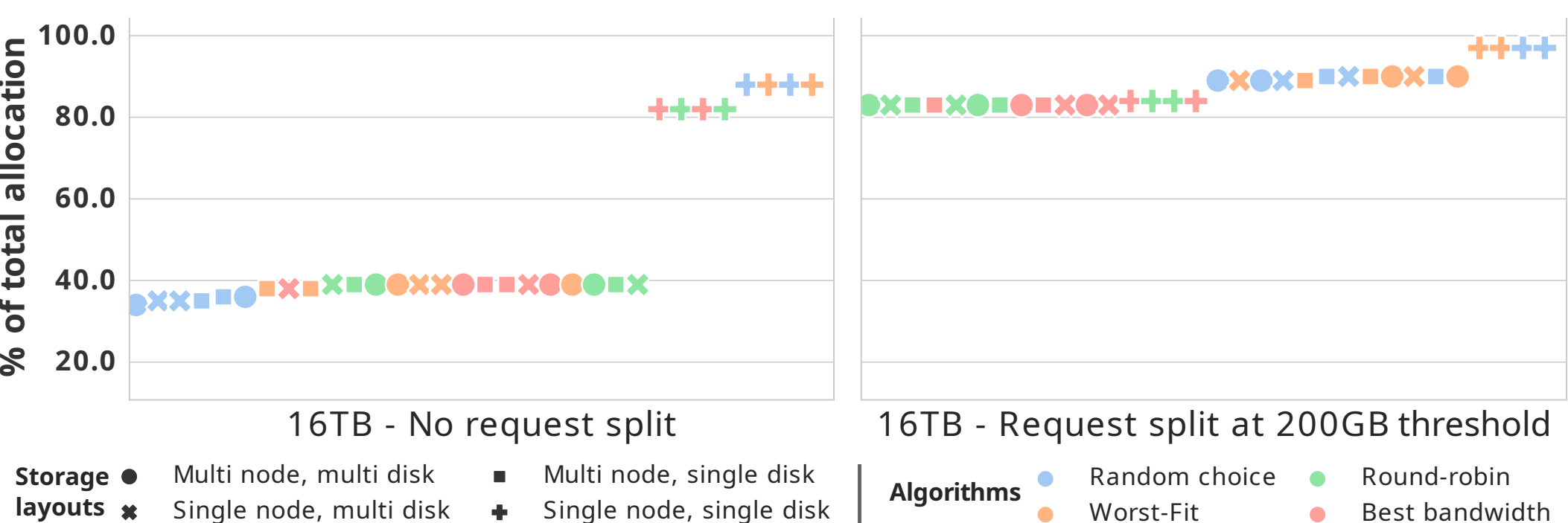


Figure 2. Percentage of the sum of the requested capacities (from all requests of dataset) that could be successfully allocated during simulation. One marker per simulation run, for a 16TB platform. Results show the algorithm and layout in use, and are grouped by split strategy.

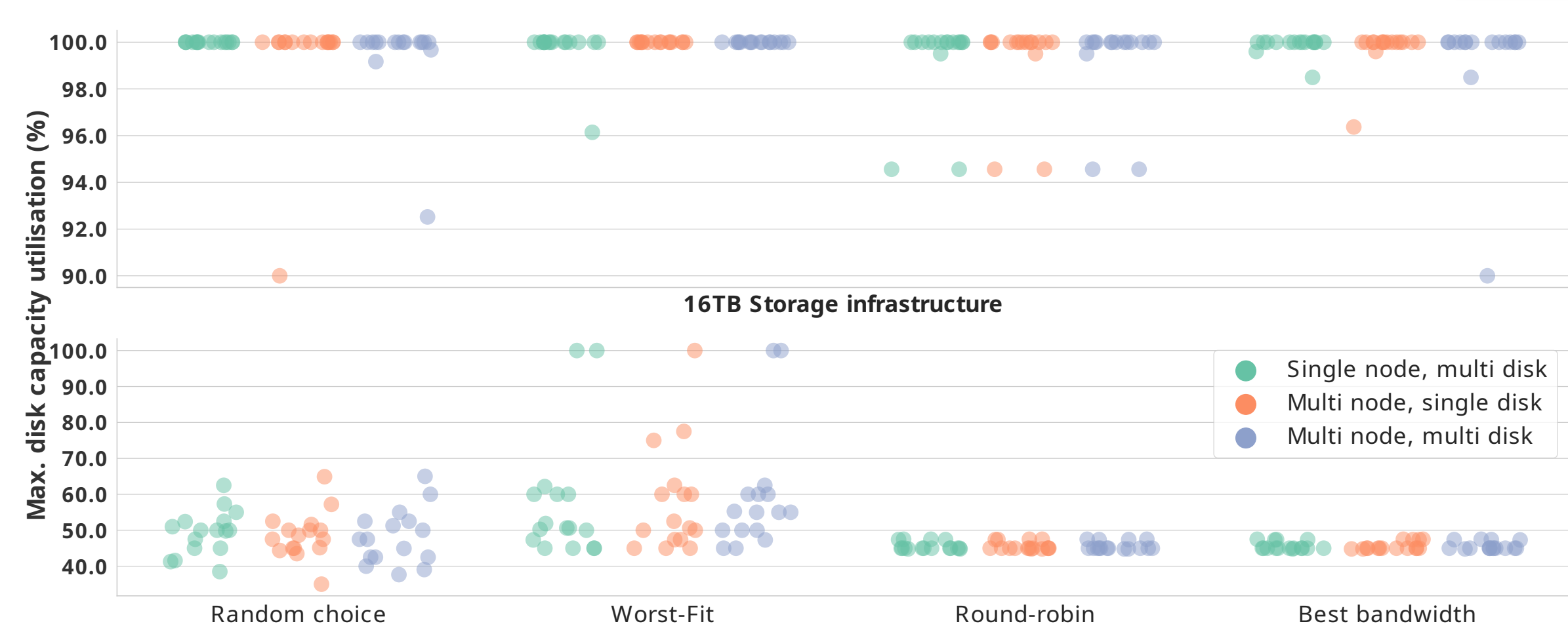
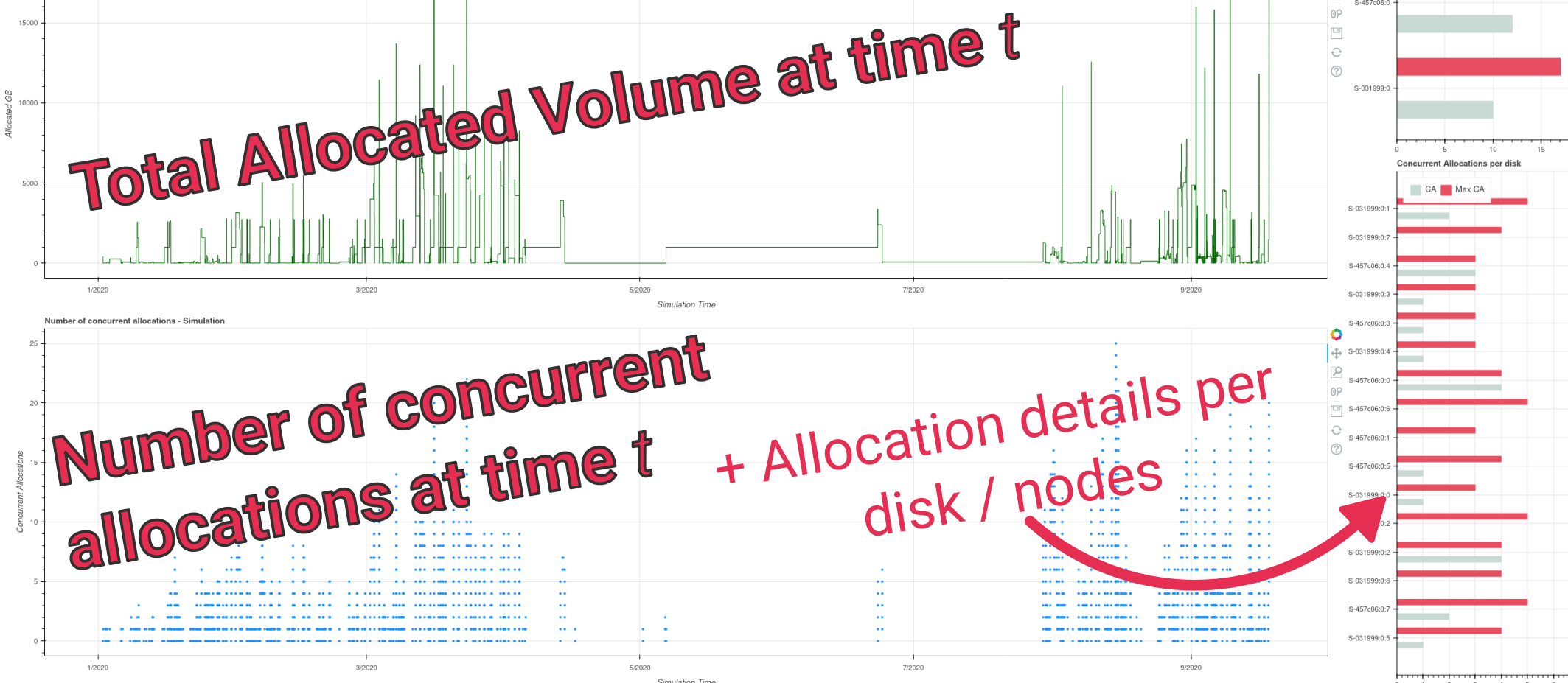


Figure 3. Maximum individual disk capacity utilization (% of total disk capacity), for 16TB and 64TB platforms, split strategy enabled at 200GB. We can confirm 16TB is too small (most disks are full), but the next size (64TB) would be underutilized (very few disks are used at more than 60% of their capacity).

IN SITU VISUALISATION

Real-time plotting of main metrics



Storage-aware job scheduler simulator

- Extensible with new algorithms
- Independent components design
- Common messaging interface
- Abstraction of heterogeneous storage hardware

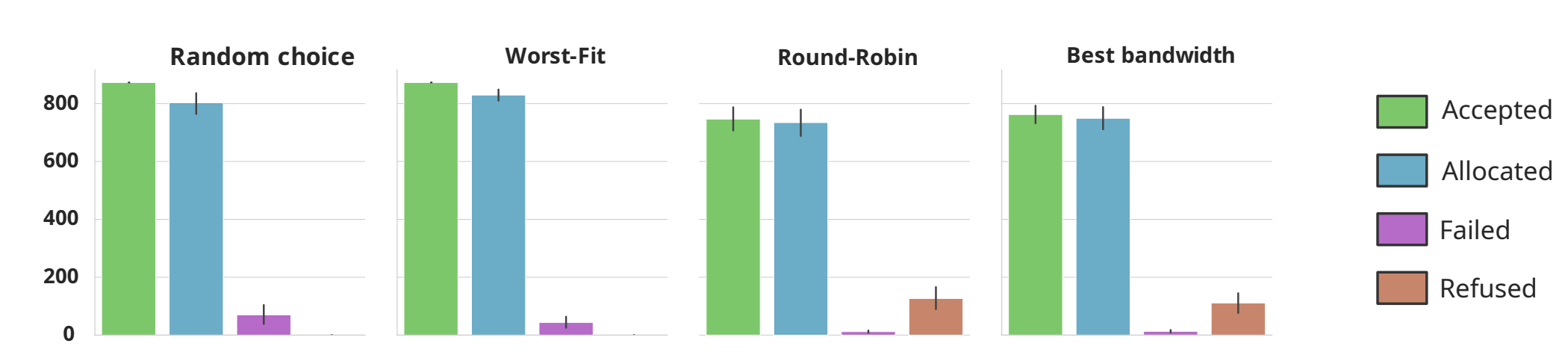
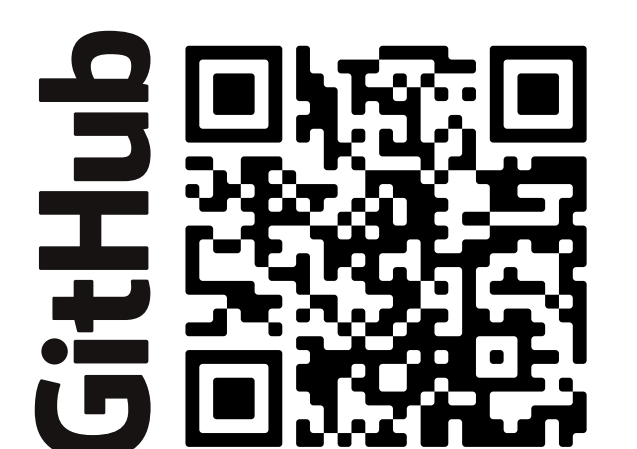


Figure 4. Number of split requests accepted, allocated, failed (during simulation) and refused (before simulation), grouped by algorithm. Random choice and Worst-fit are more permissive and rarely refuse requests, which eventually leads to more allocations. On the other hand, they result in more failed requests (error from the algorithm caught during allocation), which could prove costly 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 SimPy (DES simulation)
- ZMQ ZMQ (Messaging)
- bokeh bokeh (Real-time plotting)
- seaborn seaborn (Plotting)
- Darshan (IO traces)



[heptaicie/storalloc](https://github.com/heptaicie/storalloc)