A Simulation-Based Framework to Reduce I/O Contention in HPC

S. Pernice[1], A. Tarraf [2], J.B. Besnard [3], B. Cantalupo[1], D.E. Singh[4], A. Cascajo[4], F. Wolf [2], S. Shende[5], J.C. Perez[4], M. Aldinucci[1]

[1]Department of Computer Science, University of Turin, Italy

[2]Department of Computer Science, Technical University of Darmstadt, Darmstadt, Germany

[3]DataDirect Networks

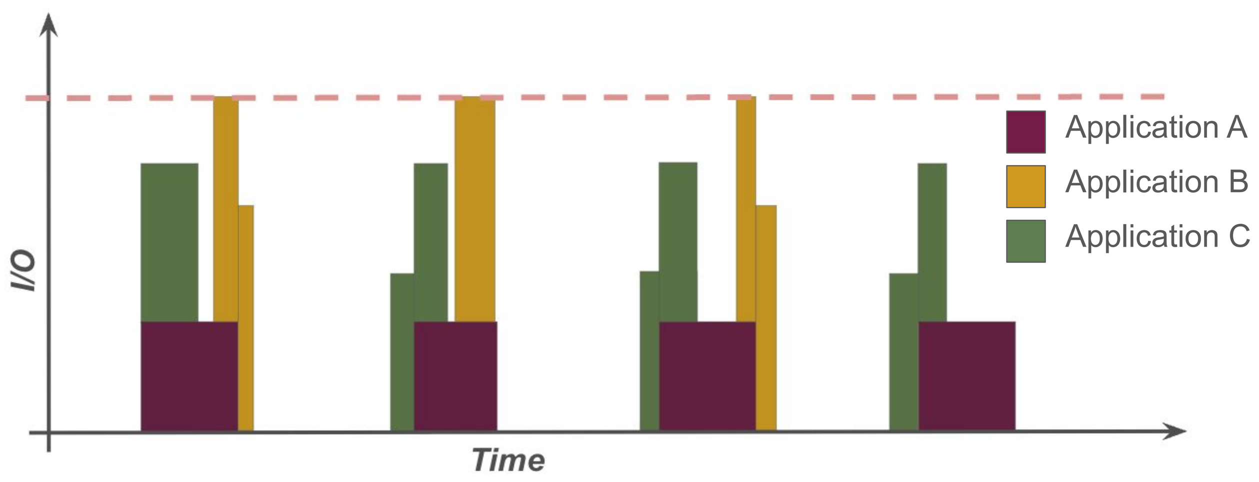
[4]Computer Science and Engineering Department, University Carlos III of Madrid, Madrid, Spain

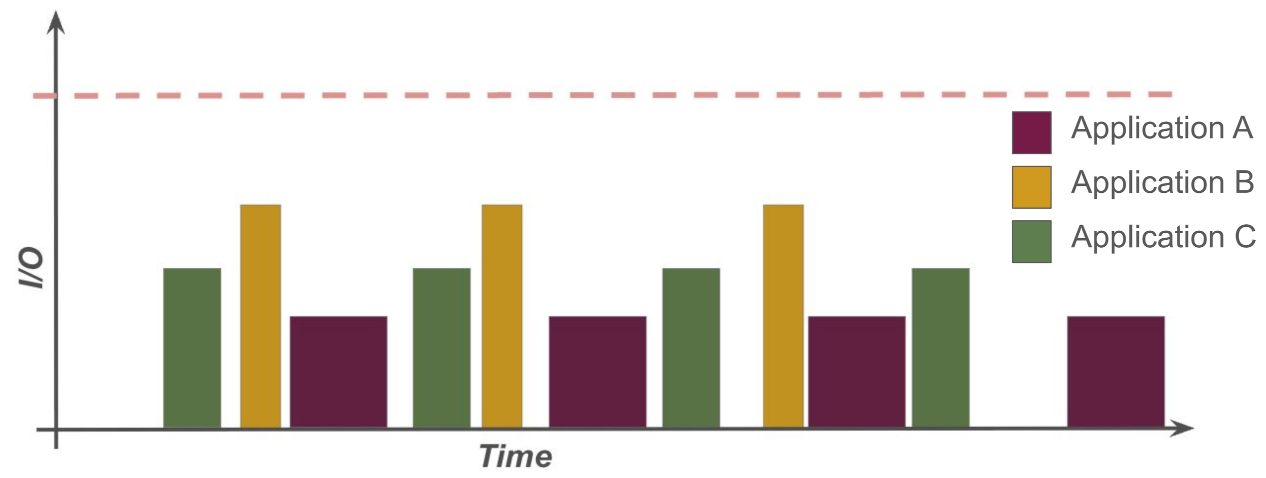
[5]ParaTools, Inc.

Contention can significantly degrade performance, making it desirable to minimize it altogether. However, due to its complex nature, predicting contention in advance poses a significant challenge.

We developed a ***modeling workflow*** combining three main components to consistently monitor and model applications running on HPC systems. The modeling enables the exploration of future resource consumption of applications, ultimately allowing the ***design of contention avoidance strategies***.

High Contention:
I/O bursts overlap





Shifted Execution:
I/O bursts distributed over time

Metric Proxy: Collecting Applications and Data

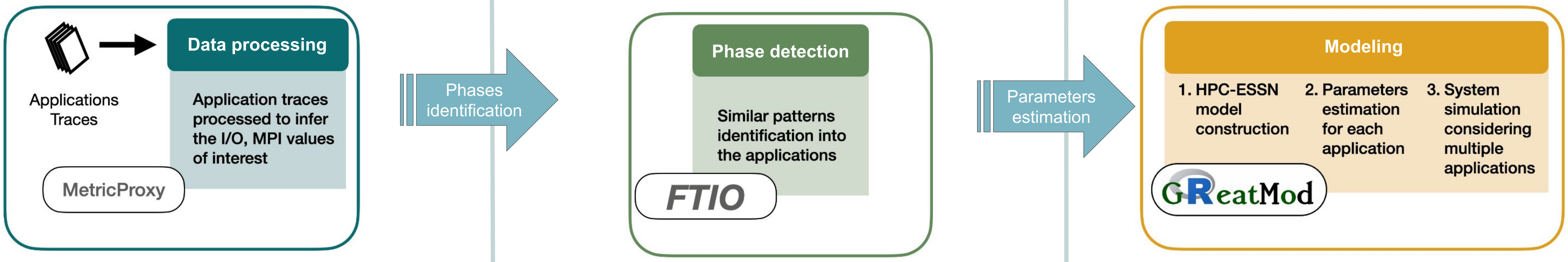
The Metric Proxy [1] is a hierarchical monitoring infrastructure that leverages a *Tree-Based Overlay Network* (TBON) to provide a near-real-time view of the entire system.

FTIO: Capturing and Labeling Periodic I/O Phases

The tool *Frequency Techniques for I/O* (FTIO) [2], uses frequency techniques coupled with outlier detection methods to identify and label the I/O phases of an application through a single metric.

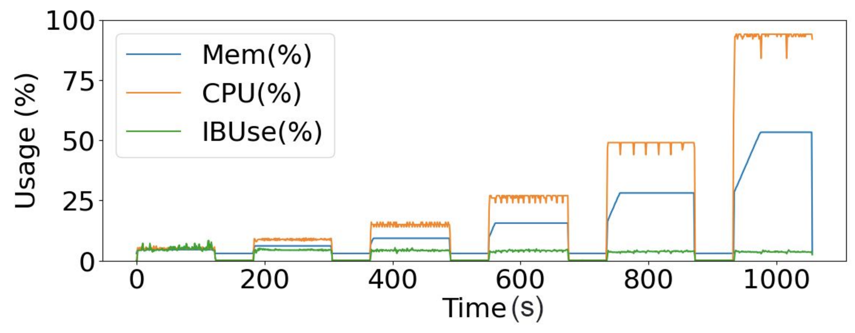
GreatMod: Modeling the Target System

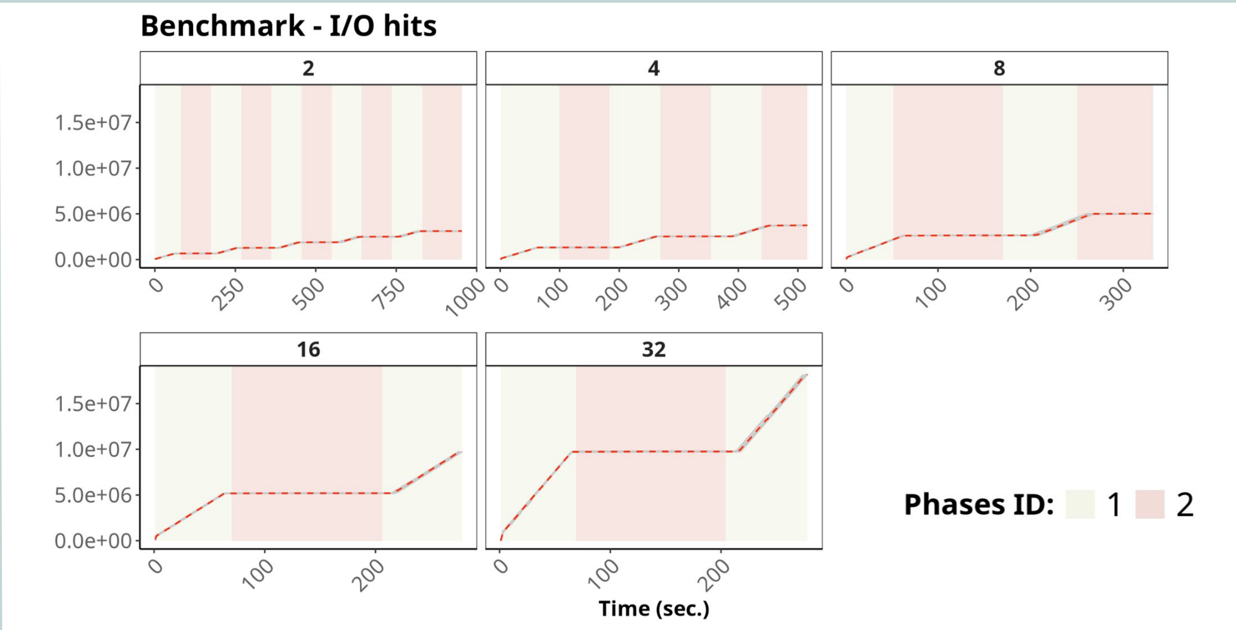
GreatMod uses high-level graph formalisms, including Petri Nets and their extensions *Extended Stochastic Symmetric Nets* (ESSNs) [3], to represent system dynamics.

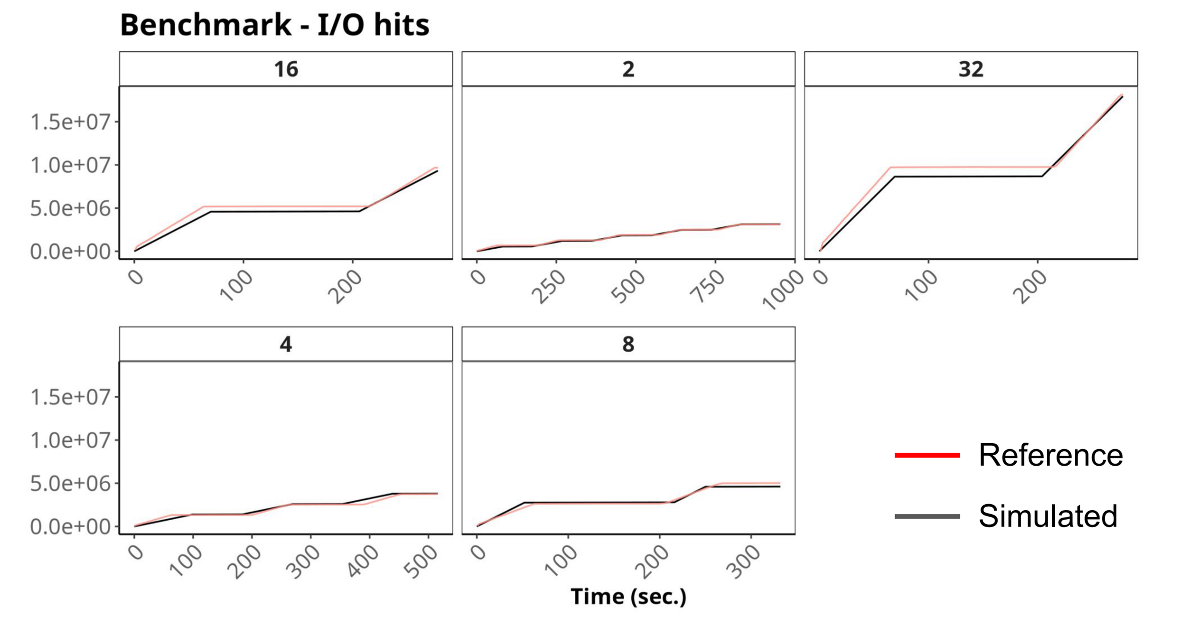


Use-case 1

Malleable benchmarks [4] to allow the generation of diverse malleable applications with distinct usage patterns.

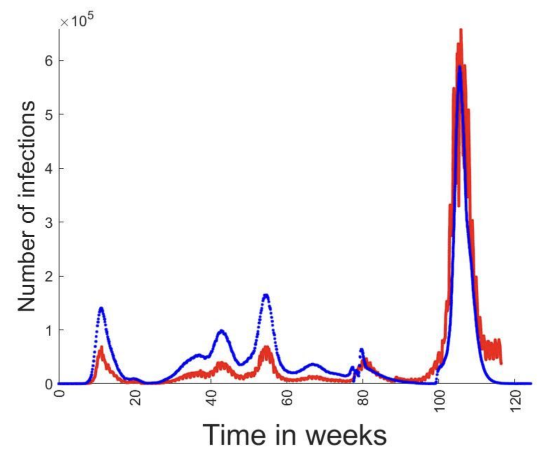


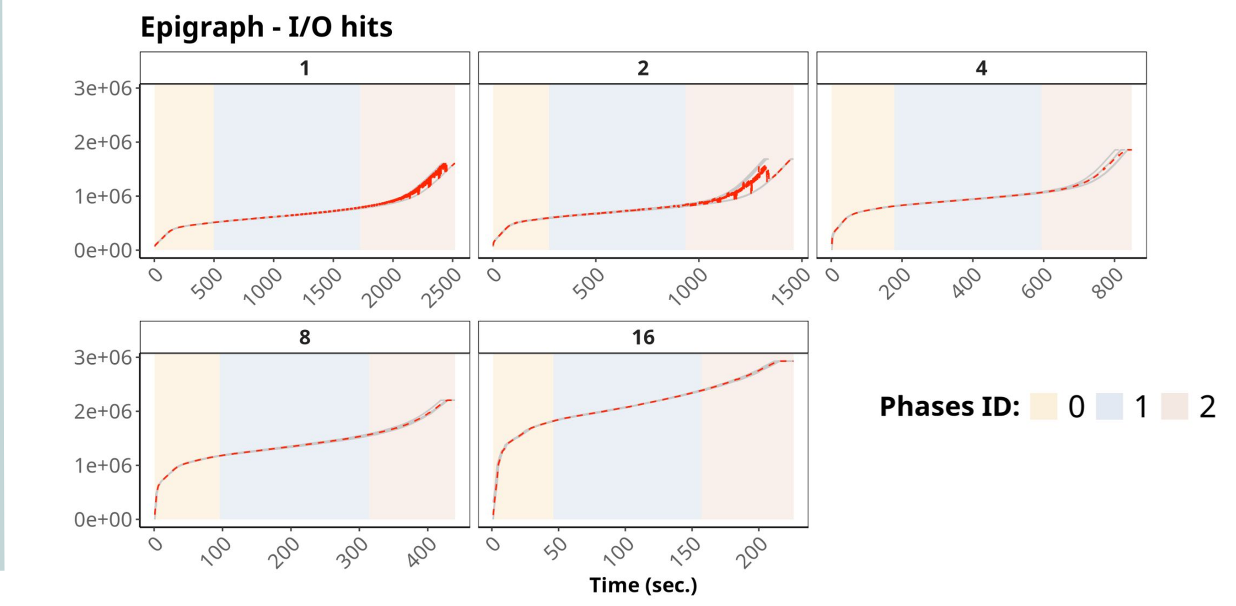


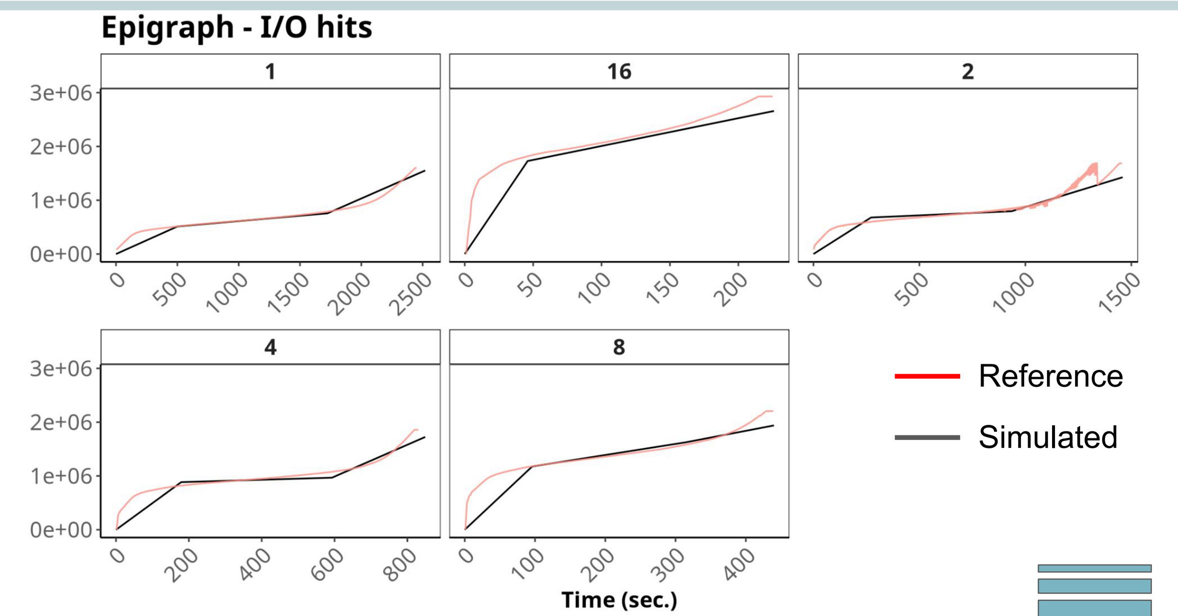


Use-case 2

EpiGraph [5], an agent-based parallel simulator that models the propagation of influenza and COVID-19.







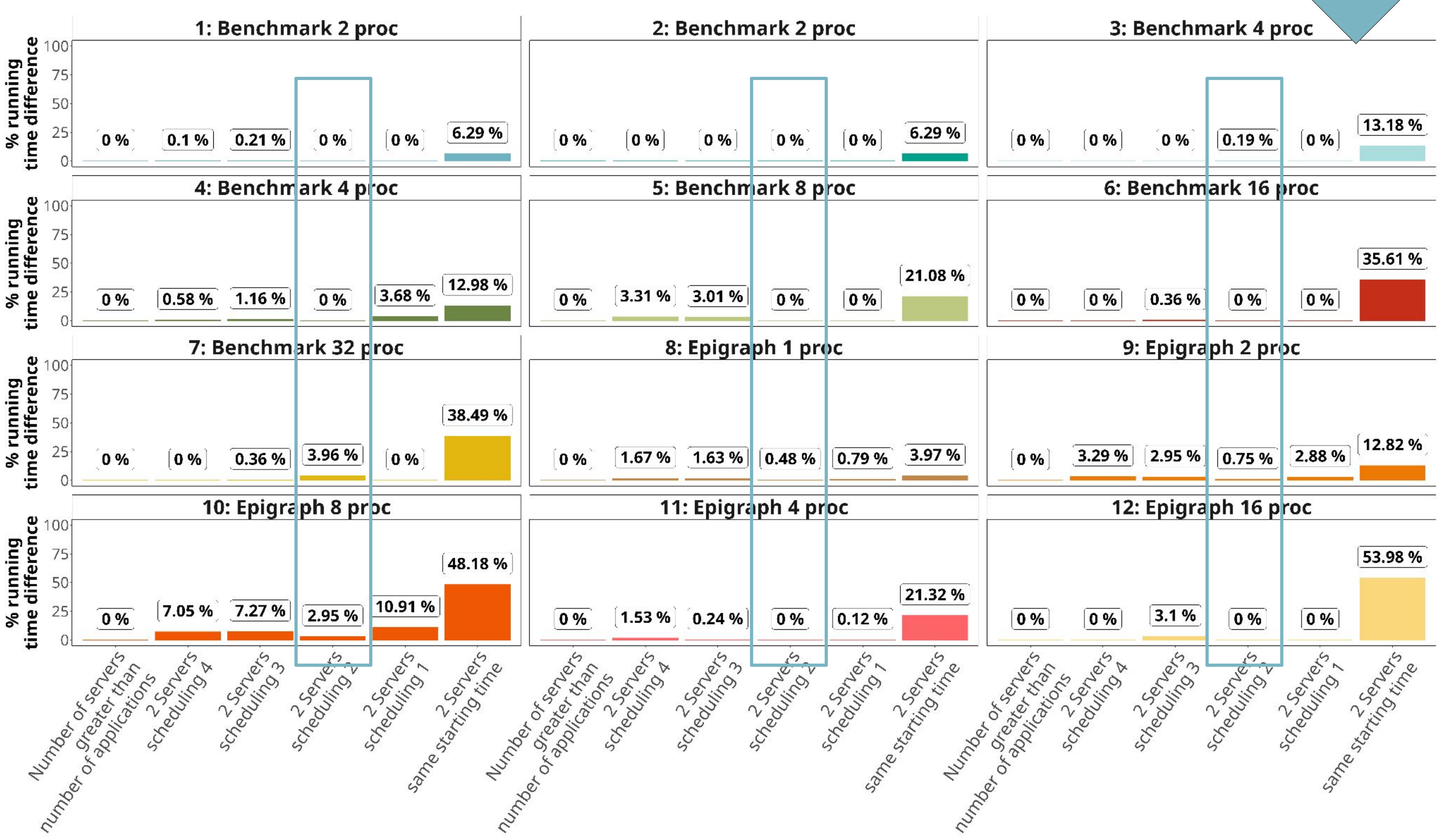
System simulation

We simulate the model to analyze the behavior of multiple concurrent applications. ***We investigated a system comprising two use cases operating with varying numbers of processes.***

The goal is to minimize the number of applications performing I/O operations concurrently.

To evaluate system performance, ***we explored six scheduling approaches***:

1. ***Number of servers greater than number of applications:*** Sufficient servers to accommodate all applications, eliminating queuing in I/O computation. Each application runs independently and concurrently. *(Optimal Resource Allocation Scenario)*
2. ***Servers same starting time***: Not enough servers to meet the demand from multiple applications, resulting in queuing and potential I/O resource contention. Competition for shared resources can lead to delays and performance degradation. *(Resource-Constrained Scenario)*
3. ***Servers scheduling 1-4***: Different scheduling strategies under worst-case conditions. We assess how altering the start times of each application in four separate ways can affect the overall application execution time.



*The **worst-case scenario** shows a significant increase in application running time due to I/O queuing. In contrast, scheduled scenarios achieve performance close to the best-case, though some applications still experience delays from server competition. Among the strategies, **Scheduling 2 proves most effective**, minimizing delays and enhancing overall system performance.*
These results emphasize the critical role of intelligent scheduling in reducing resource contention and improving efficiency.

[1] J.-B. Besnard, A. Tarraf, A. Cascajo, S. Shende, "Introducing the metric proxy for holistic i/o measurements," in High Performance Computing. ISC High Performance 2024 International Workshops, 2025.
[2] A. Tarraf, A. Bandet, F. Boito, G. Pallez, and F. Wolf, "Capturing periodic I/O using frequency techniques," in 2024 IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2024.
[3] P. Castagno, S. Pernice, et. al., "A computational framework for modeling and studying pertussis epidemiology and vaccination," BMC bioinformatics, vol. 21, 2020.
[4] A. Cascajo, D. E. Singh, J. Carretero, "Detecting interference between applications and improving the scheduling using malleable application clones." The International Journal of HPC Applications, 2023.
[5] G. Martin, D. E. Singh, M.-C. Marinescu, J. Carretero, "Towards efficient large scale epidemiological simulations in EpiGraph," Parallel Computing, vol. 42, 2015.

simone.pernice@unito.it

ADMIRE malleable data solutions for HPC