Experiences combining malleability and I/O control mechanisms

David E. Singh

University Carlos III of Madrid (Spain)

The teams

- UC3M
  - Jesus Carretero, Alberto García and David E. Singh
- INRIA @ Bordeaux
  - Emmanuel Jeannot, Guillaume Aupy, Nicolas Vidal
- Available at

  `https://gitlab.arcos.inf.uc3m.es:8380/desingh/FlexMPI`

- David E. Singh and Jesus Carretero. Combining malleability and I/O control mechanisms to enhance the execution of multiple applications. **Journal of Systems and Software.** 148C. Pages: 21-36. 2019.

Overview

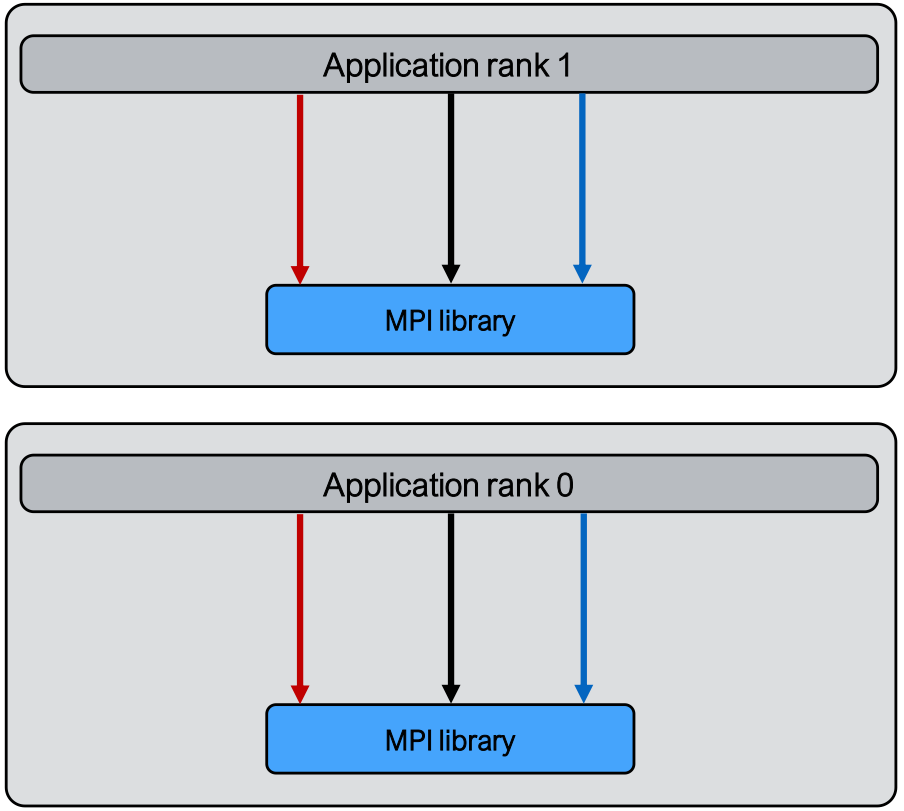
- Create a prototype with a vertically integrated software stack
- Different software components work in a coordinated manner
- Components should be able to adapt dynamically to the platform conditions
- New techniques supporting the efficient execution of applications:
  - I/O scheduling policies
  - Scheduling of malleable applications
  - Application migration

Outline

- Architecture overview
- I/O-related policies
- Experimental results

Architecture overview

- MPI application
  - Two processes



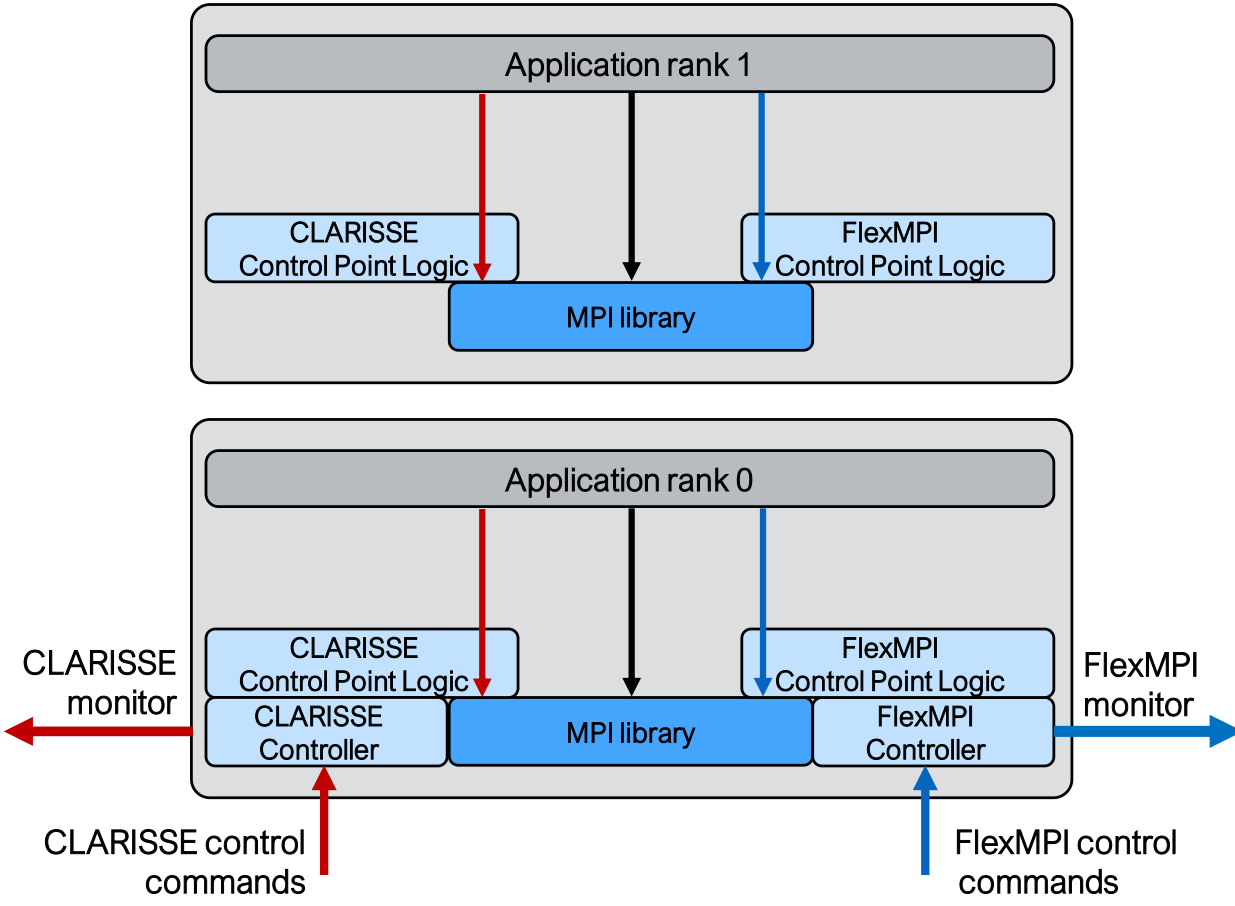
Architecture overview

- CLARISSE provides mechanisms for global data staging coordination.
  - Facilitates the flow of control and data across the I/O stack
  - Supports I/O scheduling
  - I/O monitoring

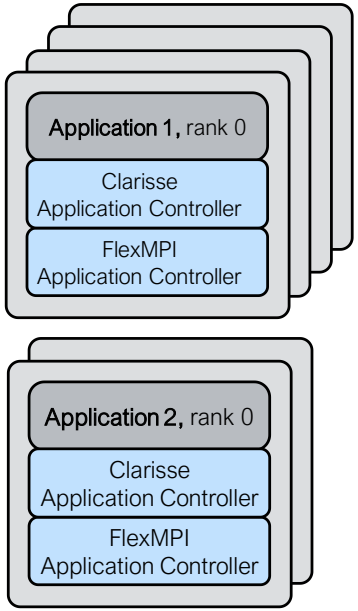
- FLEX-MPI provides malleable capabilities for MPI applications.
  - Dynamic application process creation/destruction
  - Automatic load balancing
  - CPU and communication monitoring (PAPI library)

Architecture overview

- MPI application
- Executed with CLARISSE and FlexMPI

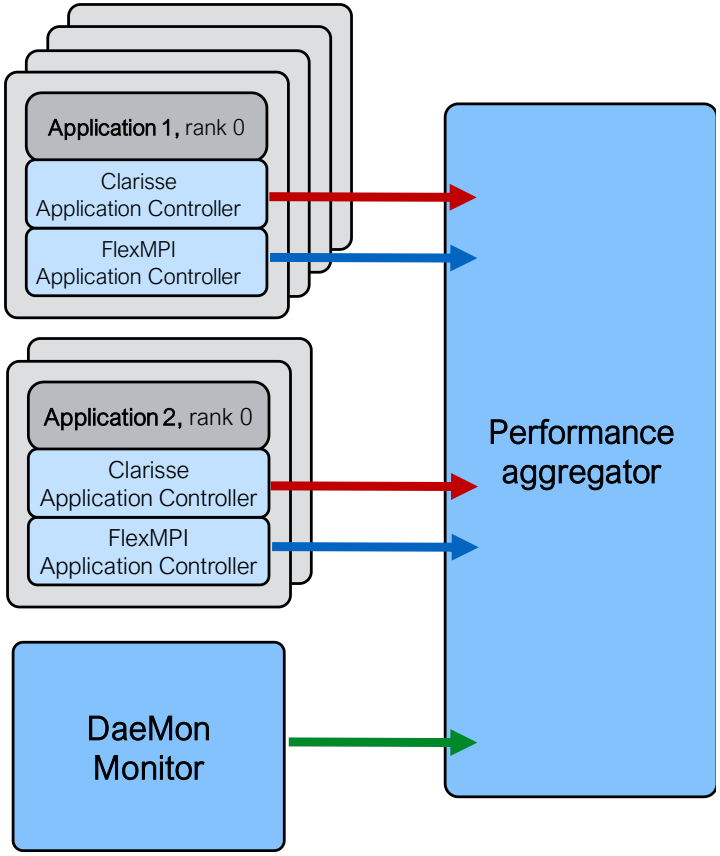


Architecture overview



Architecture overview

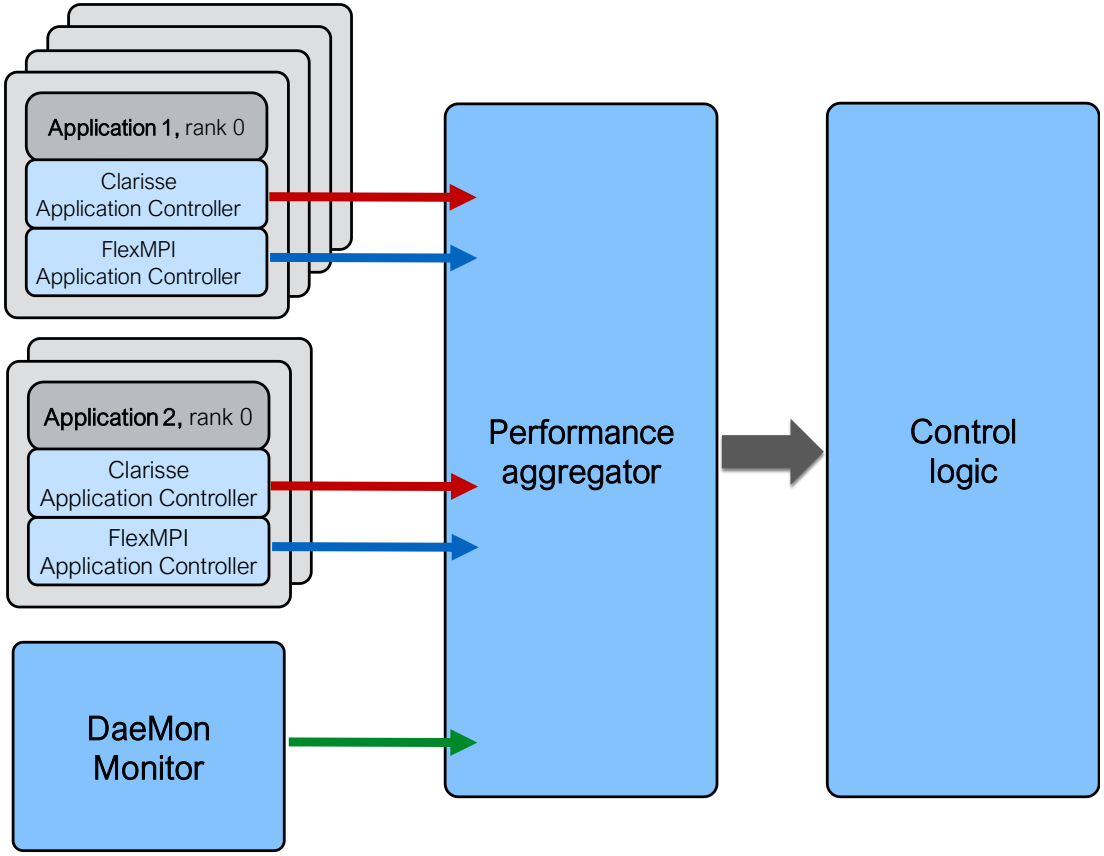
- Gathers application and compute node performance metrics



- Metric recording
- Application modelling
- I/O prediction
- Hot-spot detection
- Detection of performance degradation

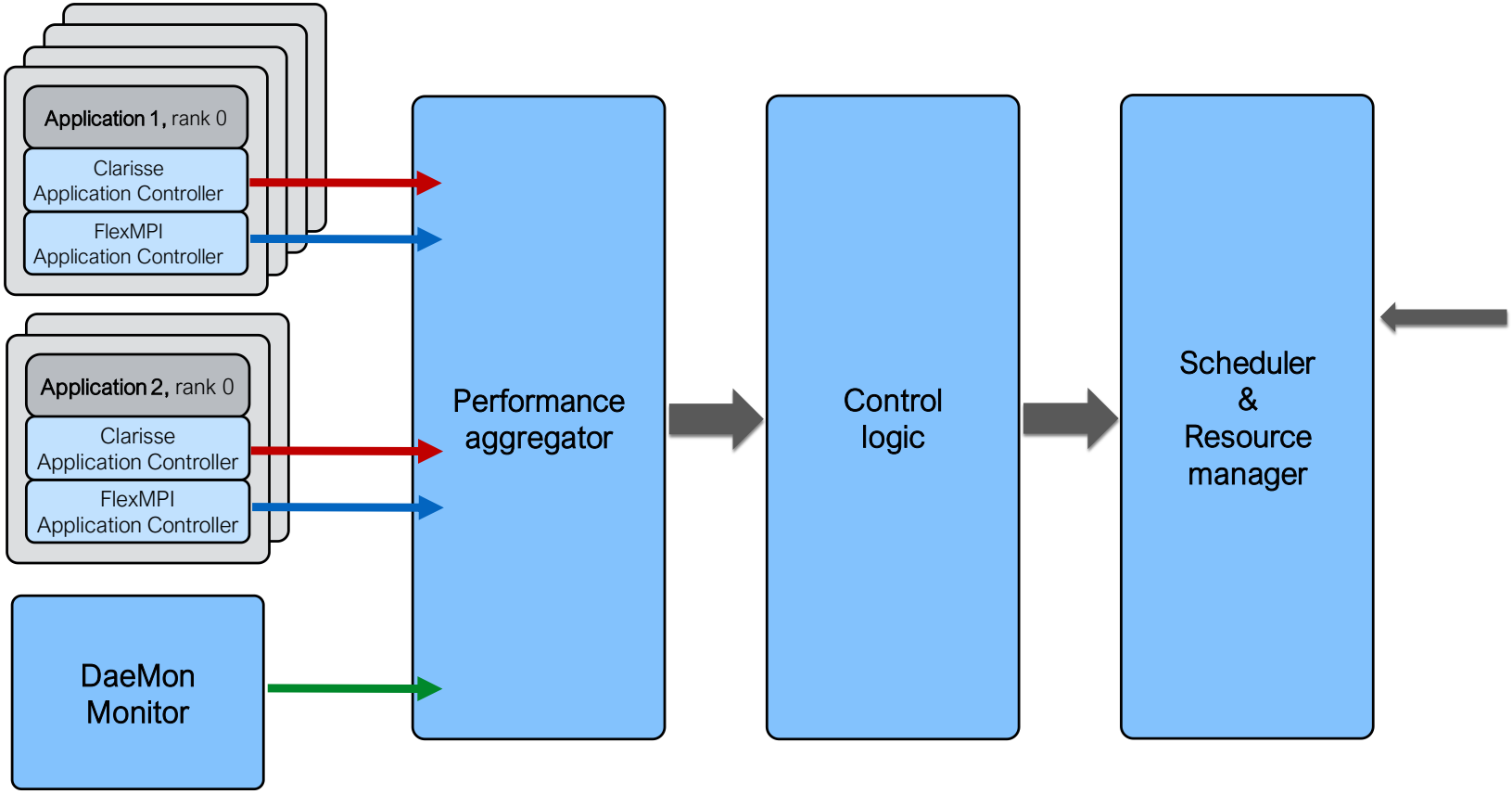
Architecture overview

- Application malleability, migration, I/O scheduling policies, …



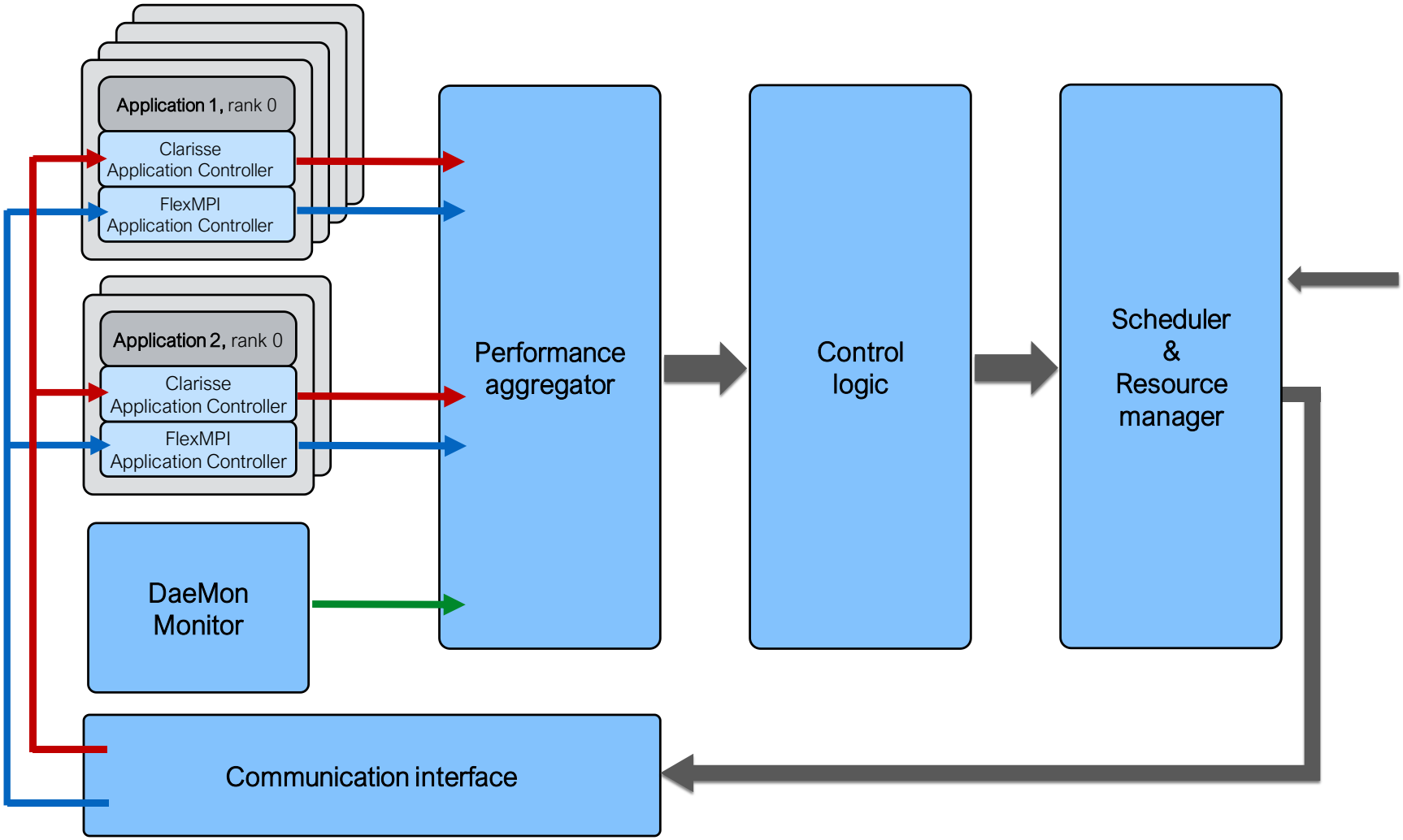
Architecture overview

- Resource allocation



Architecture overview

- CLARISSE and FlexMPI control commands



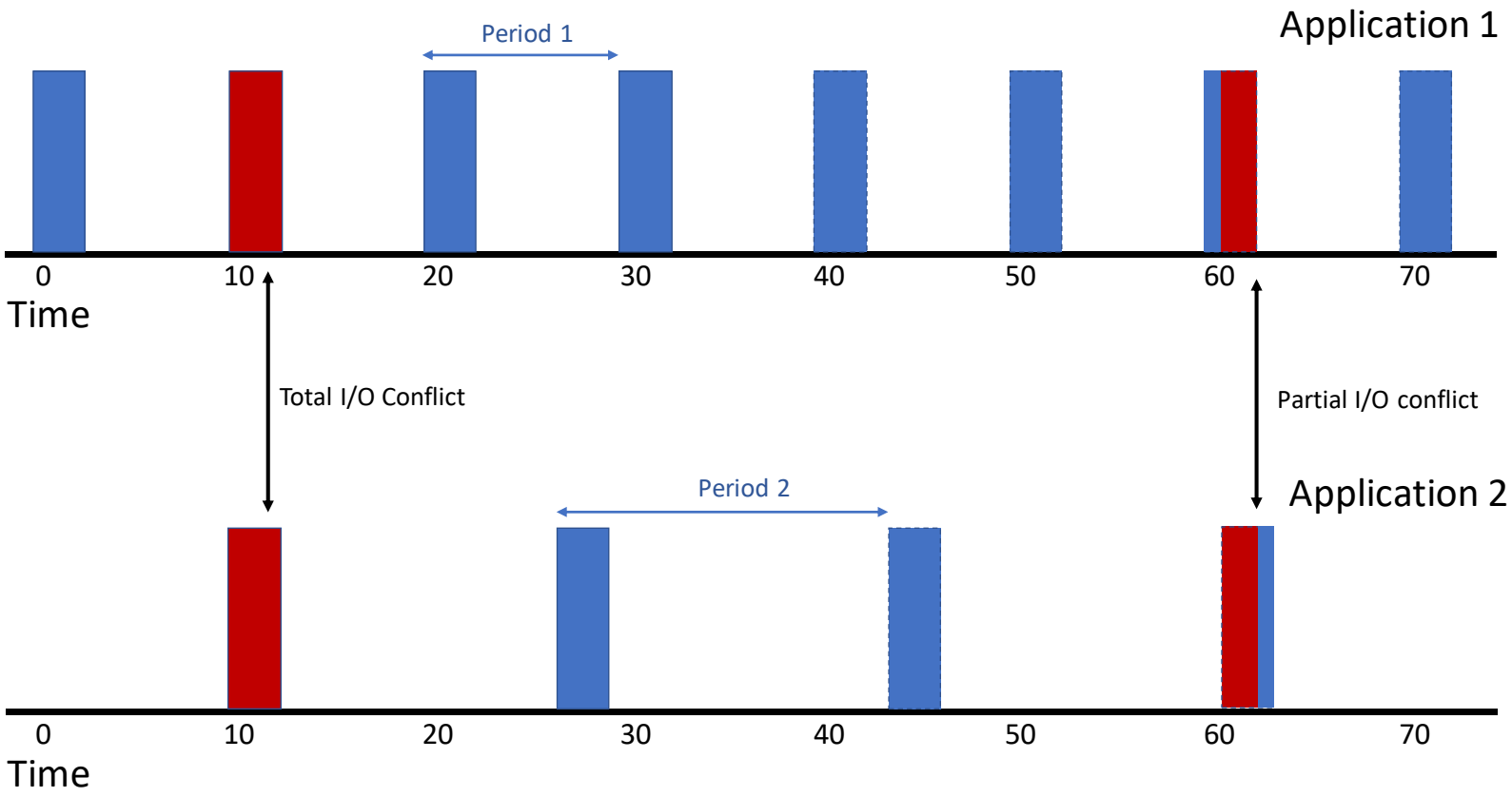
Goals

- Use of application malleability to enhance the I/O performance:
  - Coordinated use of parallel I/O scheduling and malleability for reducing number of I/O interferences

    I/O interference: *two or more I/O operations that occur partially or totally at the same time competing for the I/O resources*

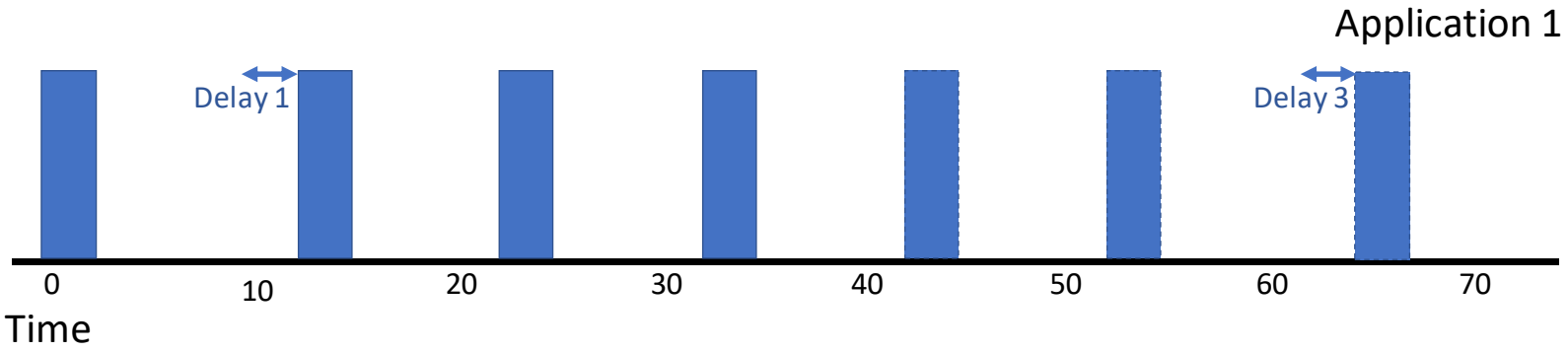
- I/O scheduling policies

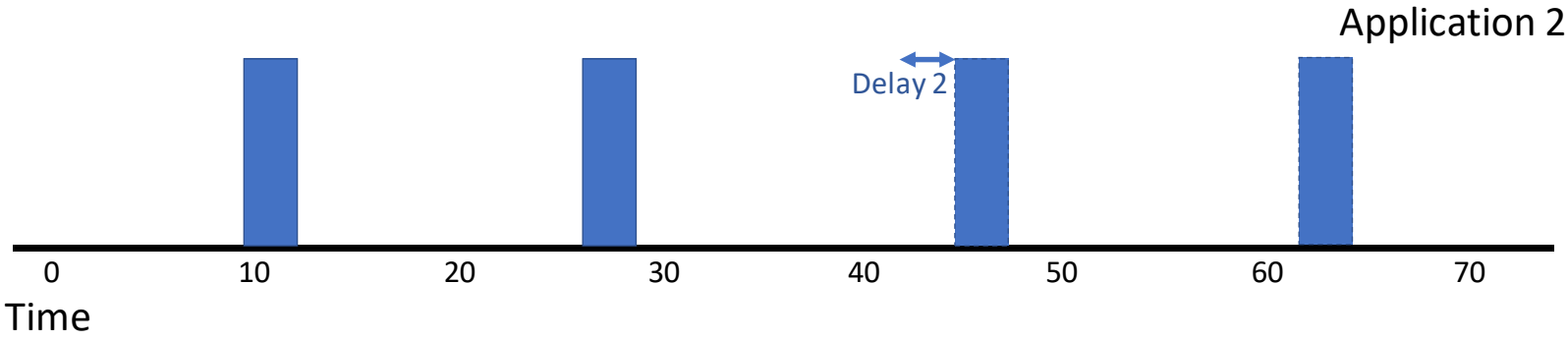
I/O interference



I/O interference

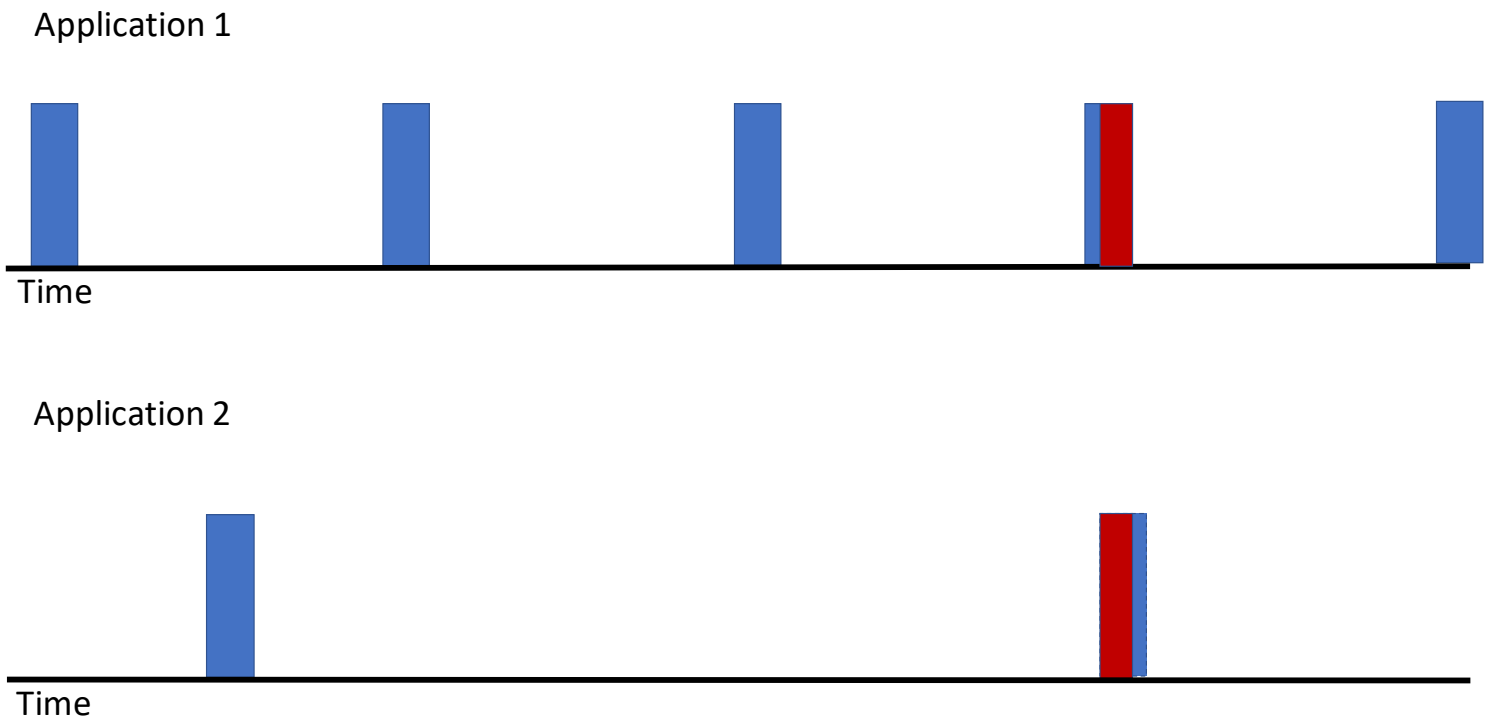
- Solutions:
  - **I/O scheduling**: blocks one I/O operation using a publish-subscribe model





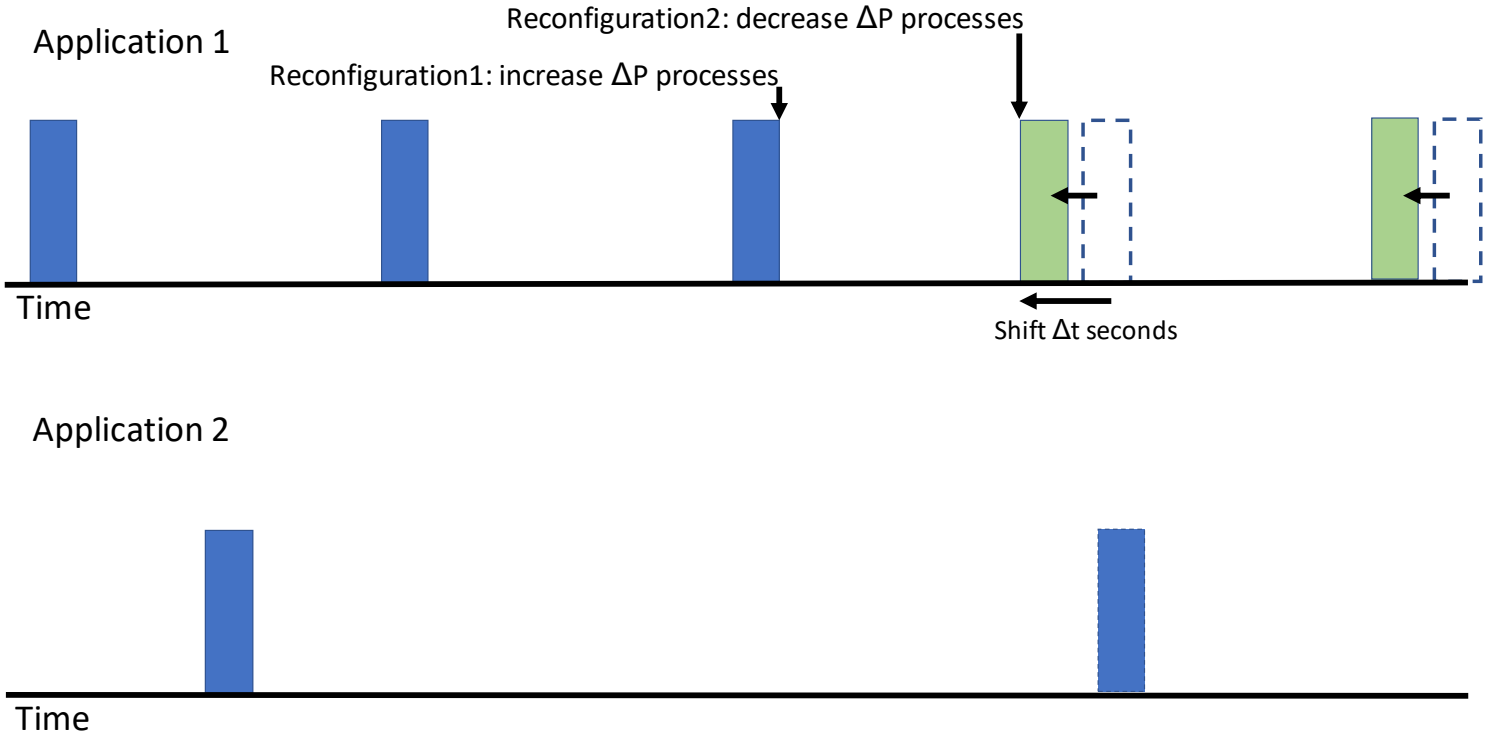
I/O interference

- Prediction of the I/O interference
- Leverage malleability for changing the I/O time stamp



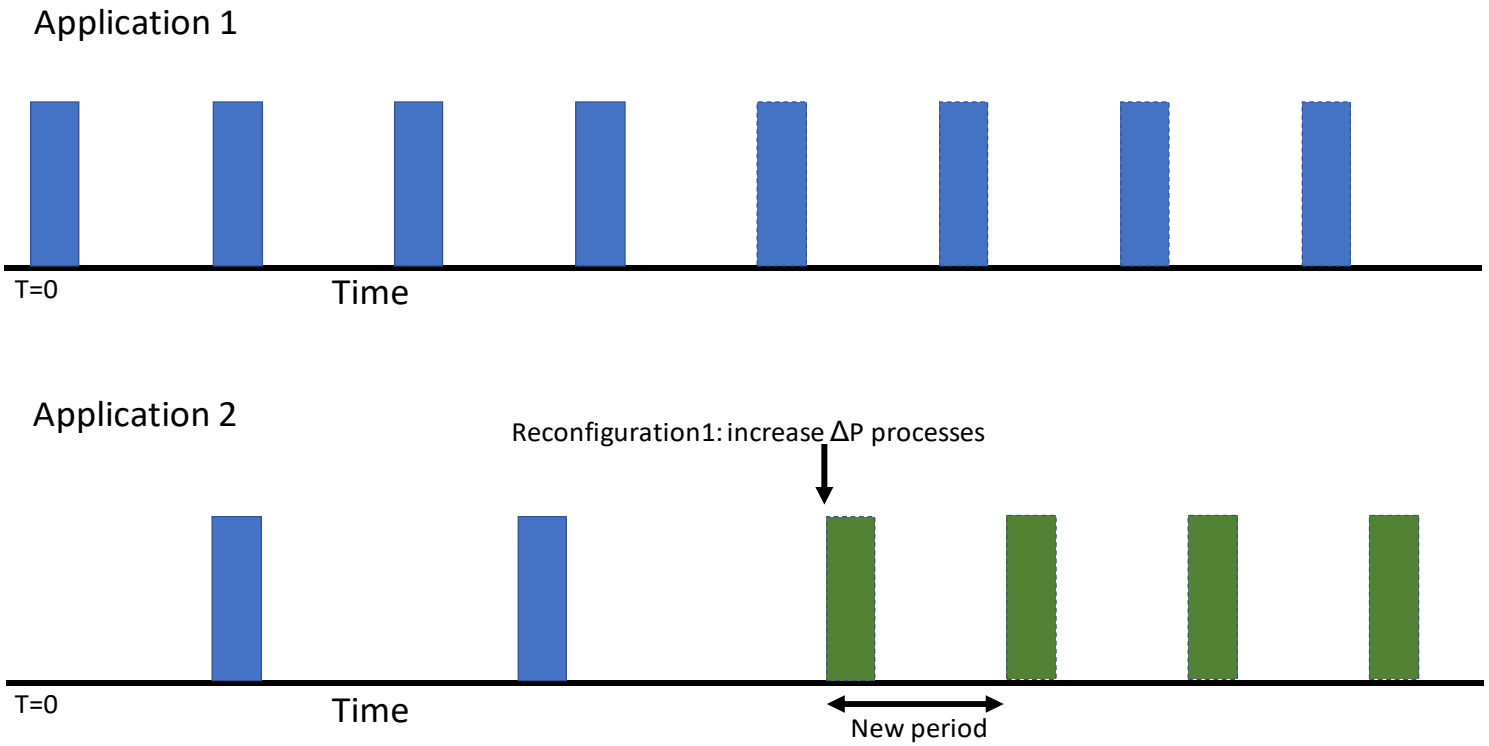
I/O interference

- **Phase shifting**
  - Leverage malleability for changing the I/O access time (phase)
  - Temporary use of computational resources



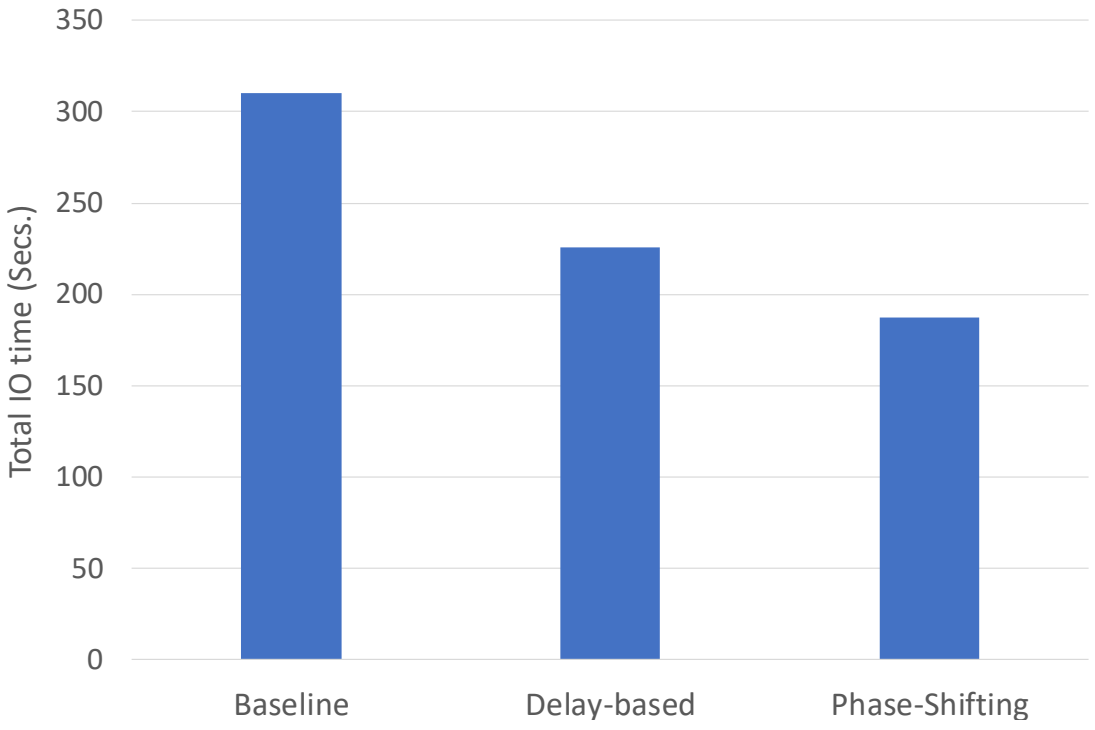
I/O interference

- **Phase coupling**
  - Leverage malleability for changing the I/O period
  - Long-term use of computational resources



I/O interference

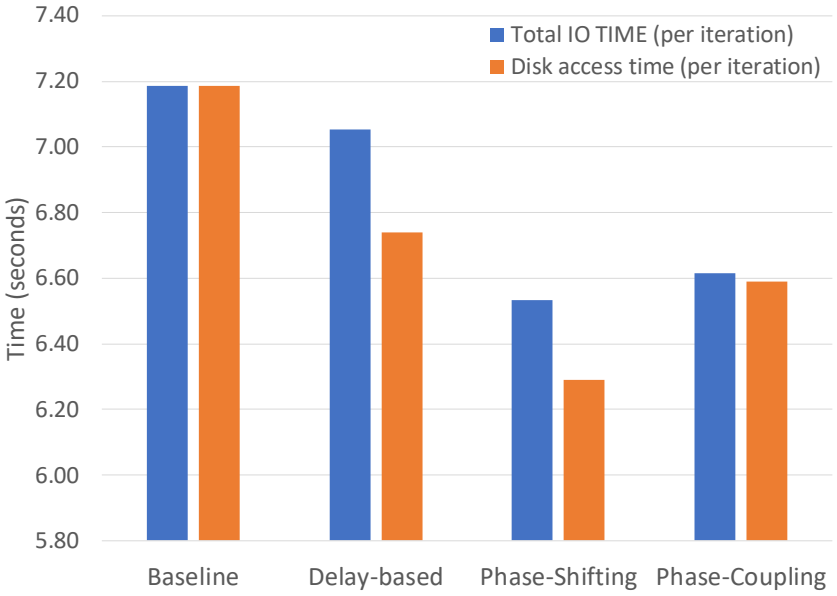
- **Results**
  - Two identical applications executed at the same time.
  - 64 processes



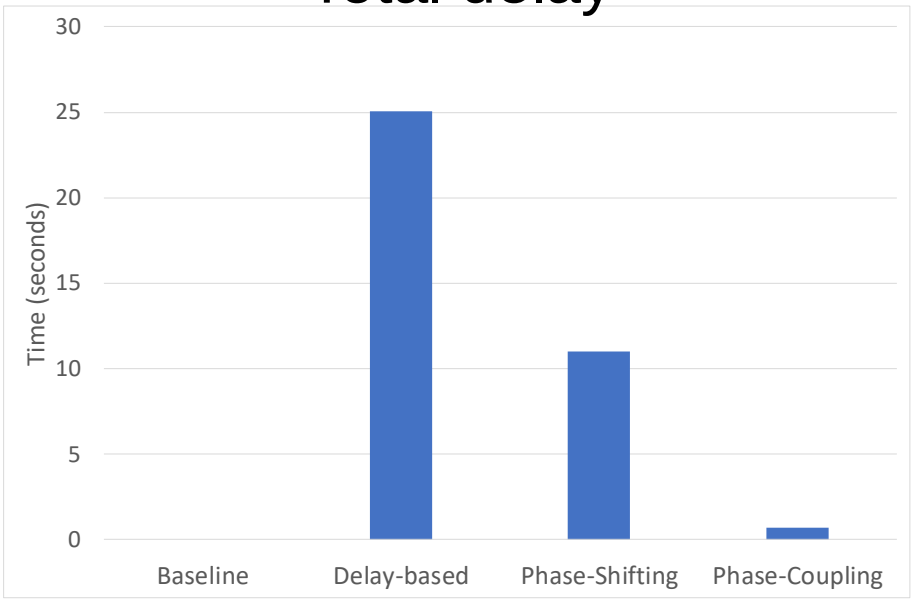
I/O interference

- **Results**
  - Two different applications executed at the same time.
  - 64 and 50 processes

I/O time



Total delay

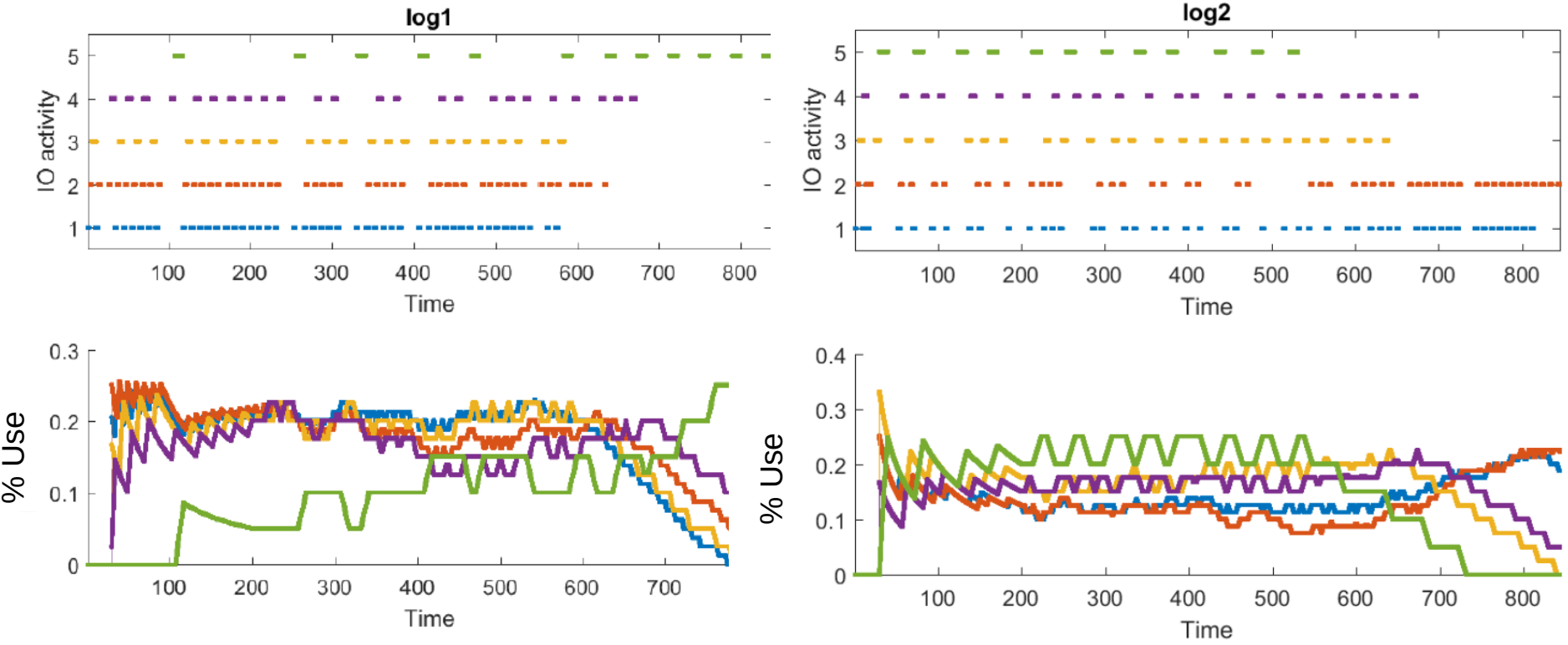


I/O scheduling policies

- Multiple applications with periodic I/O phases
- Application monitor provides information
  - Time stamp, data size, application remaining time
- Different scheduling algorithms
- Make a decision about which conflicting application (or applications) performs the I/O

I/O scheduling

- 5 applications with and increasing I/O intensity
- Shortest I/O first *vs* longest I/O first



Future work

- Extend Clarisse and FlexMPI coordination
- Machine learning techniques used for application modelling
- New performance metrics: energy, QoS