

Mapping and Scheduling HPC Applications for optimizing I/O

Jesus Carretero ¹ Emmanuel Jeannot ² Guillaume Pallez²
David E. Singh ¹ Nicolas Vidal ²

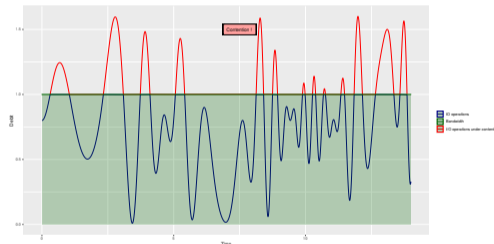
¹University Carlos III of Madrid

²Inria Bordeaux Sud-Ouest

September 11, 2020

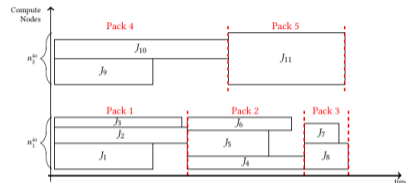
Context

- Concurrent application allocated on different compute nodes
- Racks of compute nodes are associated with I/O resources
- I/O nodes are shared proxies that have to be used to perform I/O
- Bandwidth is shared
- If too many I/O operations are performed, there is contention

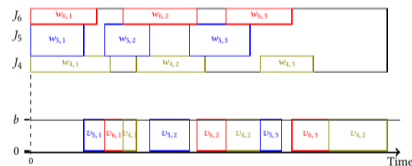


Problems

- Given a set of jobs which perform I/O on a given node, how can we schedule the operation to optimize performance ?
- How do we allocate jobs to racks ?



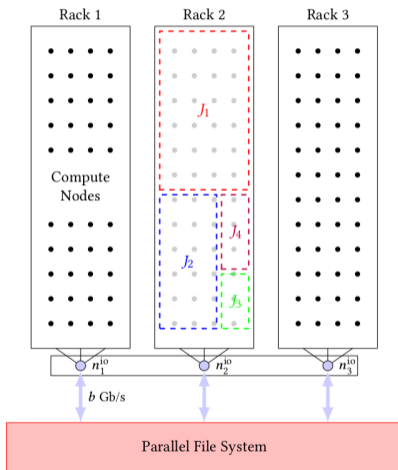
(a) The mapping of 11 applications into packs



(b) The scheduling of I/O operations (bottom) within Pack 2. Computations are allocated on dedicated compute nodes and can start as soon as the I/O is transferred, but I/O operation share the available bandwidth b and can delay applications.

Models proposition

Machine Model



Application Model

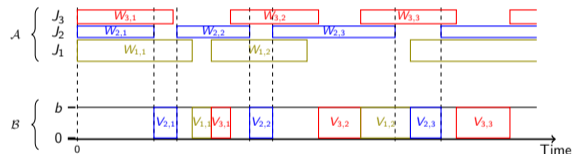


Figure: Schematic overview of three jobs J_1 , J_2 , J_3 scheduled on a bi-resources platform.

Scheduling problem

Problem

Find a schedule that minimizes the total makespan $MS\text{-HPC-IO}$ or the maximum stretch $\rho\text{-HPC-IO}$.

- NP-complete
- Few simple cases: chains of length 1 ; homogeneous tasks
- List-Scheduling (LS) provides a 2-approximation for $MS\text{-HPC-IO}$

→ LS policies are similar in theory but should be different in practice. How do we evaluate them ?

Evaluation

- Clarisse/FlexMPI¹ with synthetic workloads
- Steady state
- Choose between two "families" of heuristics:
 - ① Free the machine as soon as possible
 - ② Grant "fairly" accesses to applications

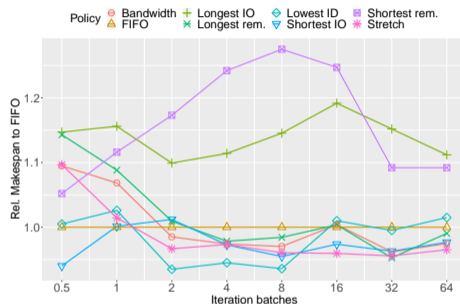
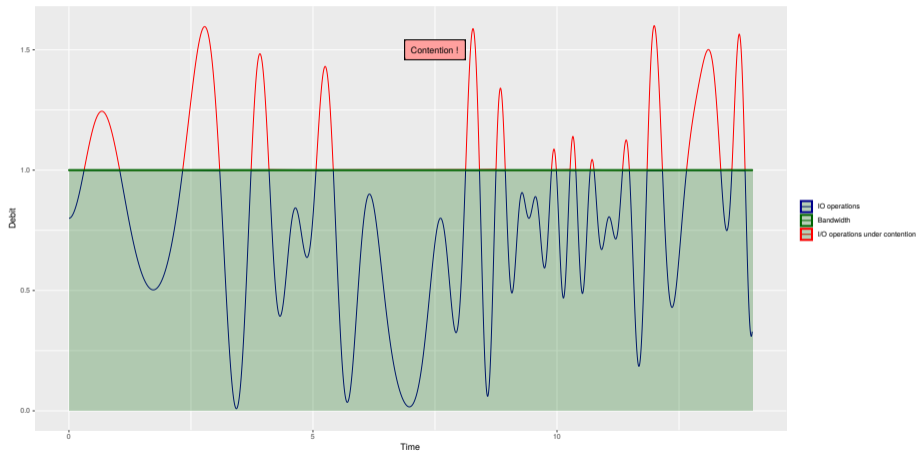


Figure: Relative Makespan to FIFO of the different policies when varying the number of iteration batches.

¹"CLARISSE is a middleware for enhancing I/O flow coordination and control in the HPC I/O stack software." It aims to coordinate the parallel I/O on HPC platforms.

Mapping problem



Packing applications

Model change

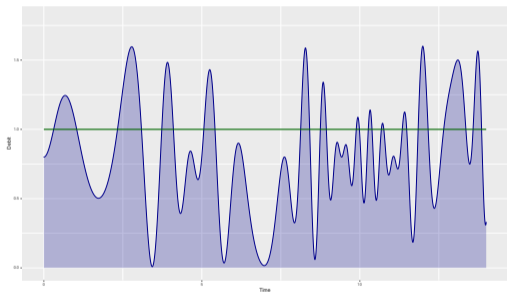


Figure: "Real" behavior

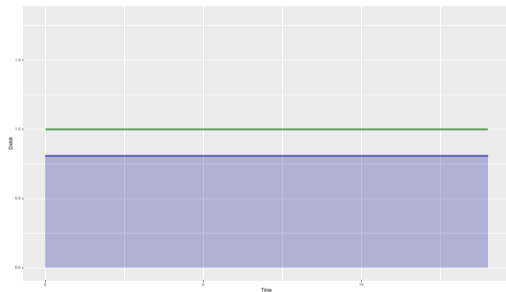


Figure: Continuous model

→ We use this new model to define I/O constraints for pack scheduling

Results

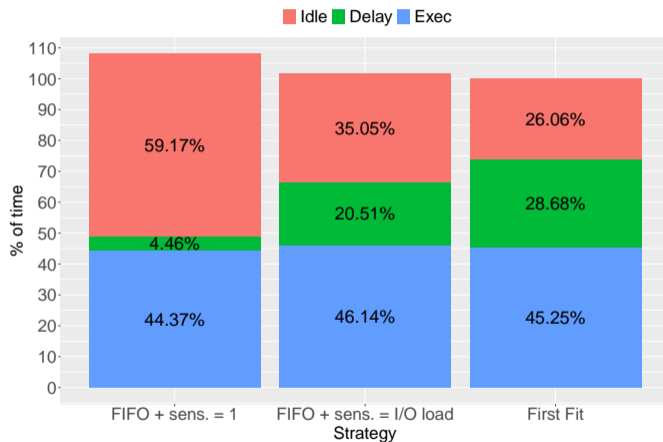
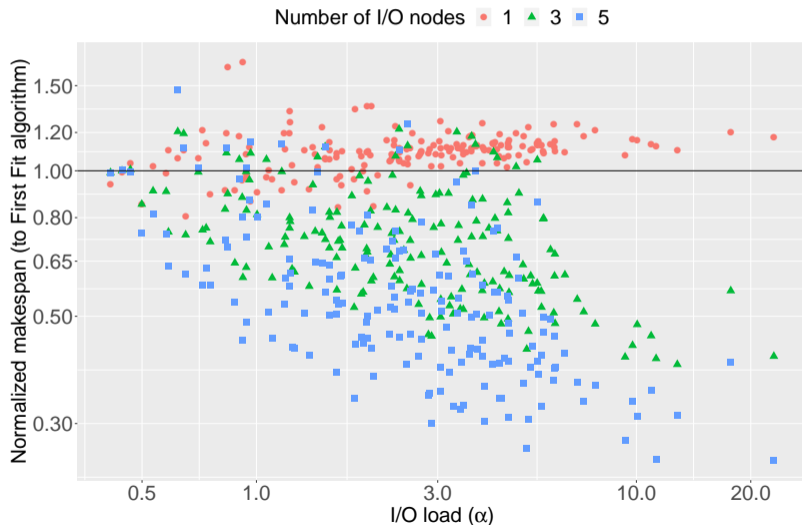


Figure: Average execution, idle and delay time (normalized) for different strategies

Scaling



Conclusion

- Simple fair heuristics which perform well in the long run.
- A strategy to allocate applications together based on an approximation of their resource usage.
- Interesting results:
 - ① Slight degradation of the makespan
 - ② Much better control of the waste
 - ③ Great scaling

Future works

- Remove pack constraint
- Enable backfilling
- Bandwidth sharing
- Discuss I/O threshold