Co-scheduling Ensembles of In Situ Workflows

Tu Mai Anh Do, Loïc Pottier, Rafael Ferreira da Silva, Frédéric Suter, Silvina Cáino-Lores, Michela Taufer, Ewa Deelman

Workshop on Workflows in Support of Large-Scale Science (WORKS)
November 14th, 2022

This work is funded by NSF contracts #1741040, #1741057, #1841758 and the U.S. DOE under contract #DE-AC05-00OR22725
In situ analysis

- Post-processing to iterative processing → data is analyzed as soon as generated (in situ)
- Decouple analysis from the simulation to interleave their executions → reduce time-to-solution
- Leverage memory-to-memory for faster data staging

Changing structure

Simulated time

stride

In situ analysis
Scheduling problems

Problem 1 (Co-Sched)
Determine a way to co-schedule simulations and analyses

Problem 2 (Co-Alloc)
Find amount of resources (e.g. number of nodes, number of cores) assigned to each simulation and analysis

- Objective: minimizing makespan
- Constraint: available compute resources (compute nodes, cores, bandwidth)
Challenges

➢ Numerous jobs in ensemble
➢ Complex data dependencies between simulations and in situ analyses
➢ Many-core architectures
   → Brute-force exploration is compute-intensive, and unachievable in time constraint

Approach

We develop a mathematical model to design efficient co-scheduling strategies and resource assignments for workflow ensemble under constraints of the available computing resources
Co-scheduling (Solution for Co-Sched)

Theorem 1 (Ideal co-scheduling)
The makespan is minimized iff each analysis is co-scheduled with its coupled simulation

→ Prioritize co-scheduling analyses with their coupled simulations to improve data locality

What if resources cannot sustain ideal co-scheduling???

Theorem 2
Analyses that are not co-scheduled with their coupled simulation should be co-scheduled together on analysis-only co-scheduling allocations

❖ Reduce a considerable number of co-scheduling mappings that have to explore
1. **Optimal resource assignment**: allocating rational number of resources such that differences among execution time are minimized.

2. **Resource-preserving rounding**: Sum of resources are the same after rounding to avoid underutilized resources.

Take resources from faster applications to accelerate slower applications.

- R3' > R3
- R2' < R2
- R1' > R1

Resource allocation (Solution for Co-Alloc)
Implications of Co-sched

Greedily pick x% largest analyses sorted by computation demand (time to execute on single core) to not co-scheduled with their coupled simulations

Greedily pick x% smallest analyses to not co-scheduled with their coupled simulations

The greater number of analyses not co-scheduled with their coupled simulation, the slower the makespan (align with Theorem 1)

→ Should co-scheduling applications coupling data together to favor data locality

WRENCH-based simulator:
https://github.com/Analytics4MD/A4MD-insitu-ensemble-simulator
Efficiency of Co-Alloc

- Resource assignment at core-level is more important to computation cost
- Co-Alloc results in better makespan in most cases, even though the proposed rounding heuristic does not guarantee optimality

| Co-Alloc | Our Co-Alloc's solution |
|----------|--------------------------|
| Ev-Alloc | Resources are evenly divided |
| n:X      | X is applied at node-level |
| c:Y      | Y is applied at core-level |

Apply Co-Alloc at both node- and core-level
Conclusion

• Determine co-scheduling policies and resource profiles based on an execution model of coupling behavior between in situ jobs in a workflow ensemble.

• Confirm the relevance of data locality and the need of well-management shared resources in co-scheduling concurrent applications.

• We plan to consider co-scheduling interference, e.g. cache interference and leverage cache-partitioning and leverage bandwidth-partitioning technologies to reduce that interference.
Thank You