Design of Robust Scheduling Methodologies in High Performance Computing

University of Basel

Platform for Advanced Scientific Computing

Doctoral Candidate: Ali Mohammed
Advisor: Prof. Dr. Florina M. Ciorba
Department of Mathematics and Computer Science
University of Basel, Switzerland
{firstname.lastname}@unibas.ch

1. Research Context

- Loops are the main source of parallelism in computationally-intensive scientific applications
- Scientific applications performance on high performance computing (HPC) systems may be degraded due to load imbalance

Challenges

- Load imbalance may be caused by irregular computational load per loop iteration, or irregular and unpredictable computing system characteristics
  - Dynamic loop scheduling techniques are used to address load imbalance in computationally-intensive applications
- Perturbations and failures are expected to manifest increasingly in future HPC systems, with high count of processing elements (nodes, sockets, cores, ...)

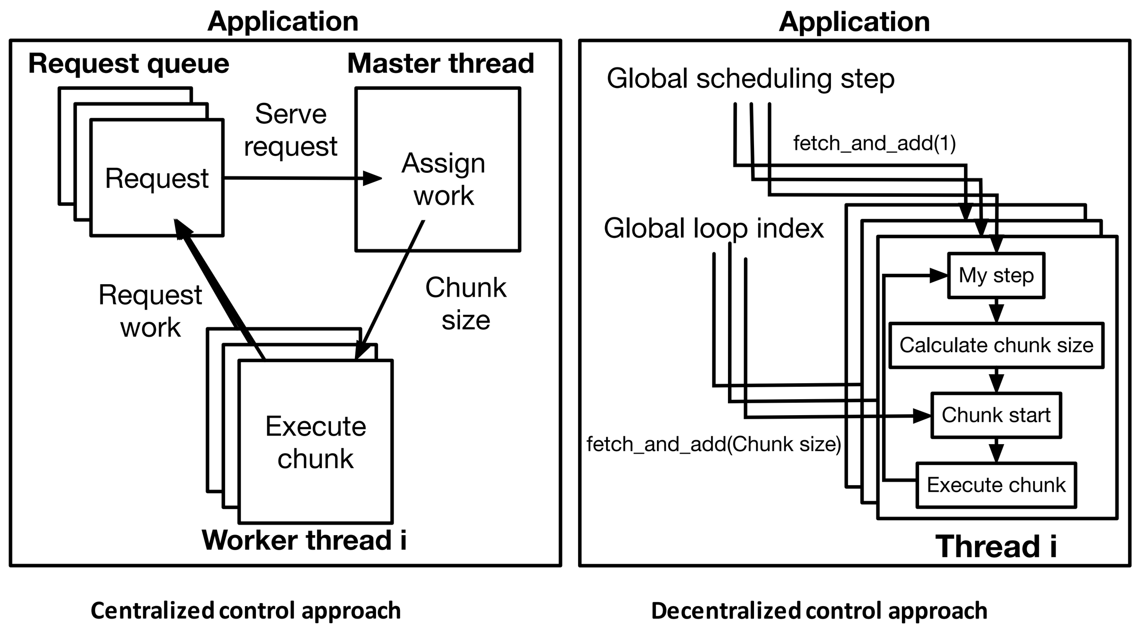
Goal

*Improve computationally-intensive scientific applications performance on HPC systems under unpredictable application and system characteristics via **robust scheduling***

2. Dynamic Loop Scheduling Techniques

Loop scheduling characteristics [1]

Scheduling technique	Category			Chunk calculation		Chunk size		Use of batches	
	Static	Dynamic: Nonadaptive	Dynamic: Adaptive	Deterministic	Probabilistic	Fixed	Variable	Yes	No
Static block cyclic (STATIC)	✓			✓		✓			✓
Self-scheduling (SS)		✓		✓		✓			✓
Fixed size chunking (FSC)		✓			✓	✓			✓
Guided self-scheduling (GSS)		✓		✓			✓		✓
Factoring (FAC)		✓			✓		✓	✓	
Weighted factoring (WF)		✓			✓		✓	✓	
Adaptive weighted factoring (AWF-B)			✓		✓		✓	✓	
Adaptive weighted factoring (AWF-C)			✓		✓		✓		✓
Adaptive weighted factoring (AWF-D)			✓		✓		✓	✓	
Adaptive weighted factoring (AWF-E)			✓		✓		✓		✓
Adaptive factoring (AF)			✓		✓		✓		✓



DLS implementation approaches

3. Research Questions

Q1. How *close* are the *implementations of DLS* techniques in simulation and native codes to their original proposed specifications decades ago?

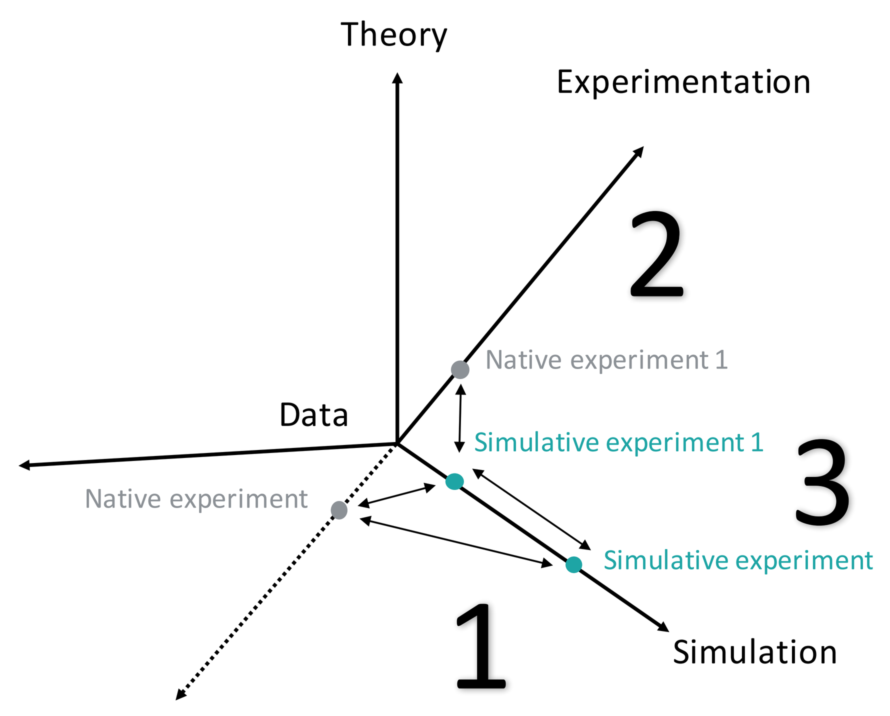
Q2. How *realistic* are the *simulations* of executions of scientific applications using DLS on HPC platforms?

Q3. Given an application, a high-performance computing (HPC) system, and both their characteristics and interplay, *which DLS technique* will achieve improved performance under *unpredictable perturbations*?

Q4. How to tolerate fail-stop *failures* of PEs during execution and maintain a balanced load enhanced application performance?

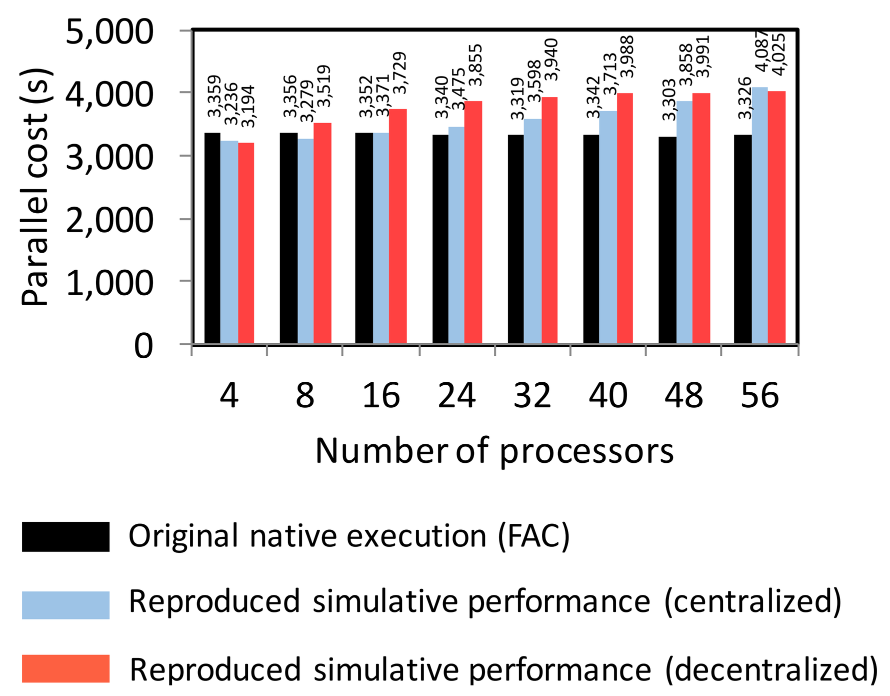
Q5. How to *ensure* applications results *integrity* under silent data corruption (SDC) faults?

4. How Realistic are Simulations of Performance?



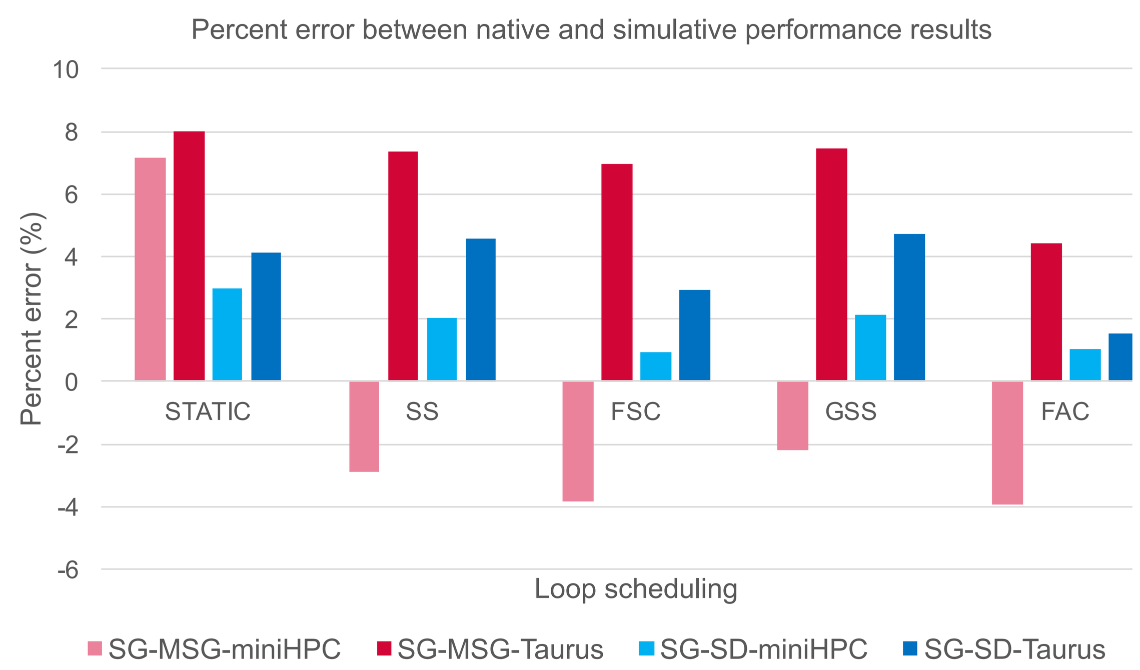
Reproduction and verification approach

Answer **Q1** [2]



Verification of DLS implementation by reproduction

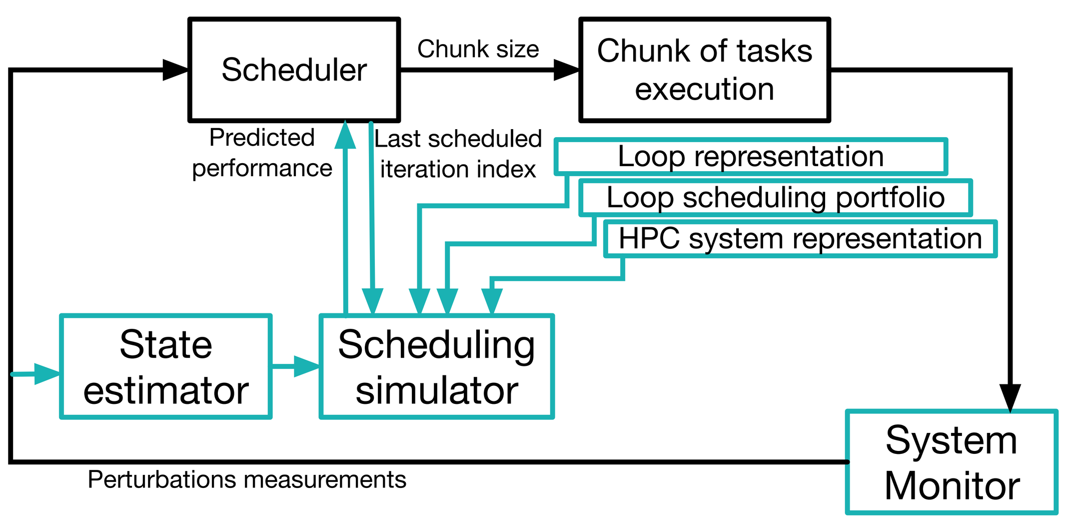
Answer **Q2** [3]



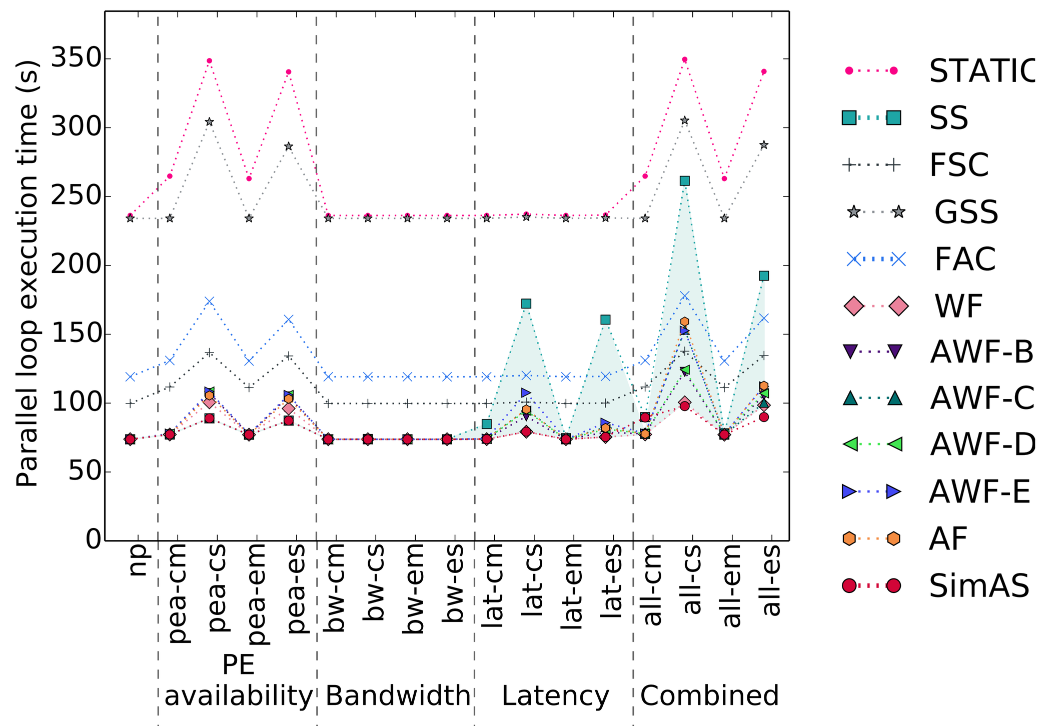
Verification of simulative versus native performance

5. SimAS: Simulator Assisted Scheduling Under Perturbations

Answer **Q3** [4]



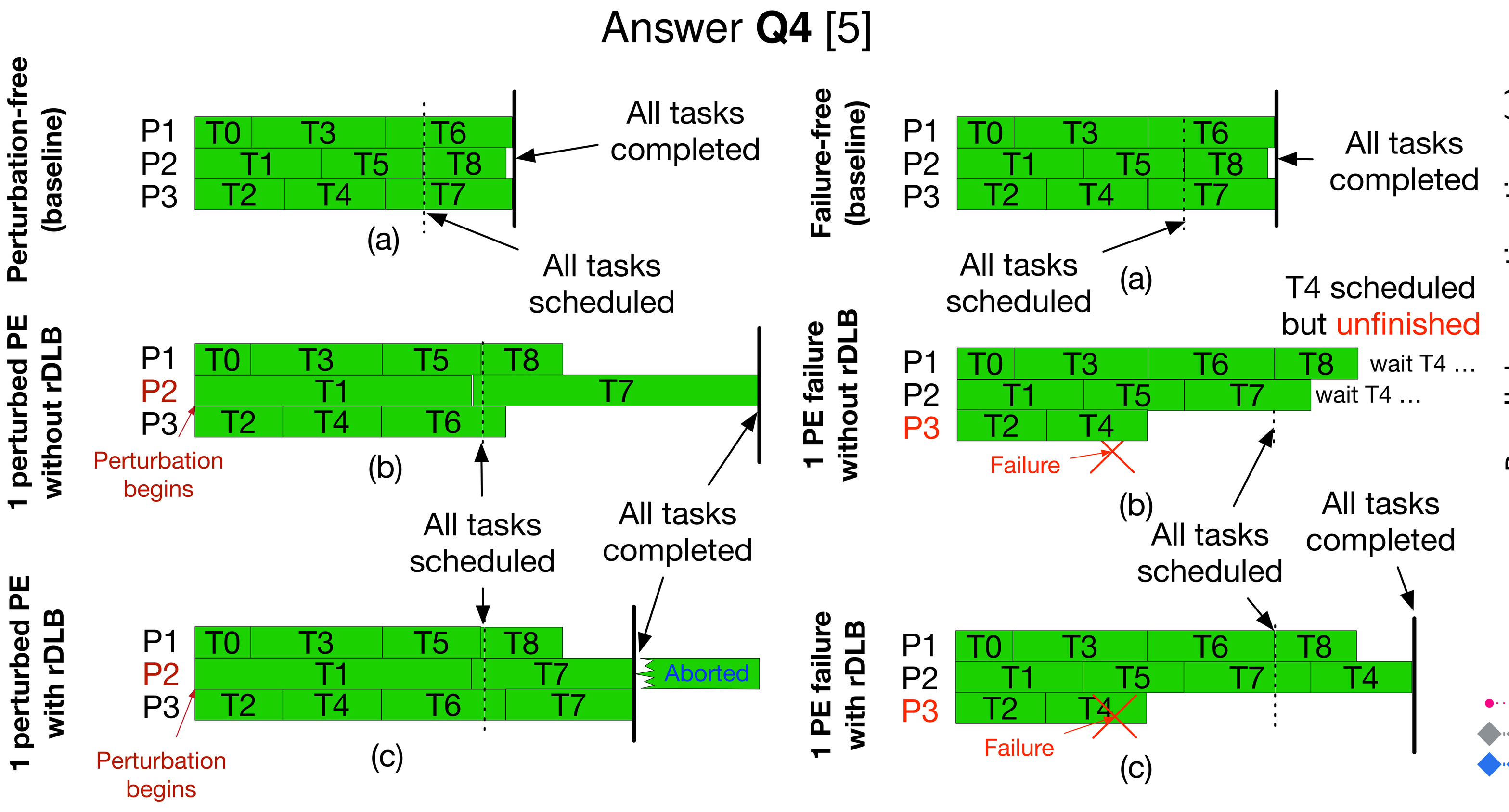
SimAS: Simulator assisted scheduling for DLS technique selection under perturbations



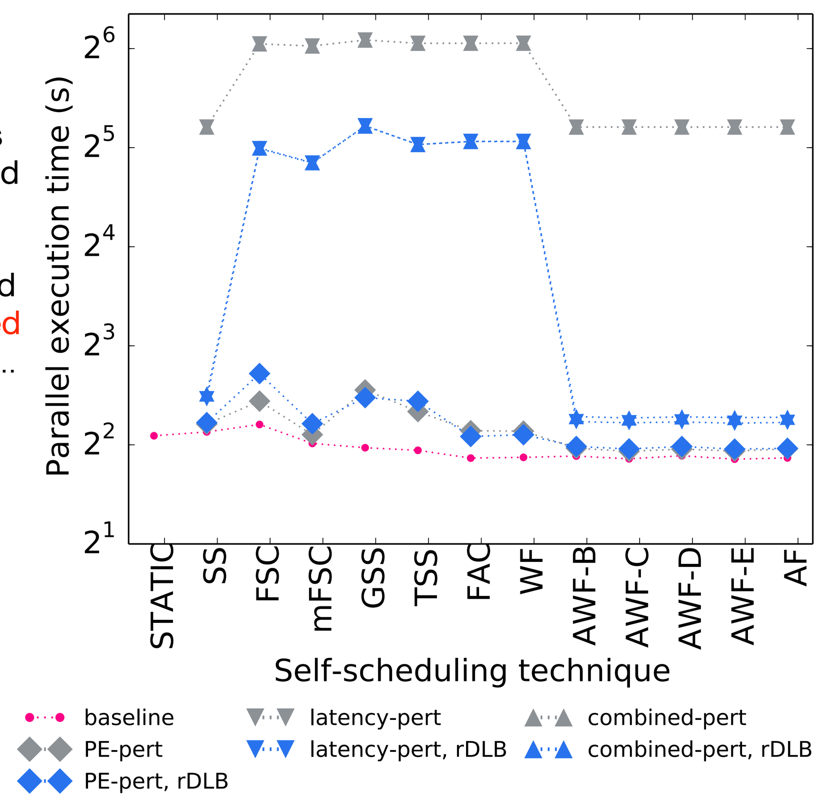
Performance under perturbations

6. rDLB: Robust Dynamic Load Balancing

Answer **Q4** [5]



rDLB: Robust Dynamic Load Balancing of Scientific Applications



Performance under perturbations

7. Next Steps

- Address SDC by replicating computations and solve scheduling challenges of replication (answer **Q5**)
- Analyze load imbalance in scientific applications at multiple levels of software parallelism
- Examining the performance of DLS at large scale via simulations
- Exploit multilevel scheduling to achieve fault tolerant application execution

Acknowledgment

This work is supported by the Swiss Platform for Advanced Scientific Computing (PASC) project SPH-EXA: Optimizing Smooth Particle Hydrodynamics for Exascale Computing

References

[1] I. Banicescu and R. L. Cariño, "Addressing the stochastic nature of scientific computations via dynamic loop scheduling," *Electronic Transactions on Numerical Analysis*, vol. 21, 2005.

[2] A. Mohammed, A. Eleliemy, and F. M. Ciorba, "Performance Reproduction and Prediction of Selected Dynamic Loop Scheduling Experiments," in *Proceedings of the 2018 International Conference on High Performance Computing and Simulation*, 2018.

[3] A. Mohammed, A. Eleliemy, F. M. Ciorba, F. Kasielke, and I. Banicescu, "Experimental Verification and Analysis of Dynamic Loop Scheduling in Scientific Applications," in *17th International Symposium on Parallel and Distributed Computing*, 2018.

[4] A. Mohammed and F. M. Ciorba, "SiL: An Approach for Adjusting Applications to Heterogeneous Systems Under Perturbations," in *Proceedings of the 24th International European Conference on Parallel and Distributed Computing Workshops*, 2018.

[5] A. Mohammed, A. Cavelan, and F. M. Ciorba, "rDLB: A Novel Approach for Robust Dynamic Load Balancing of Scientific Applications with Independent Tasks," in *Proceedings of the 2019 International Conference on High Performance Computing and Simulation*, 2019.