“Resource utilization aware job scheduling to mitigate performance variability"

Presenters: Ridham Dholaria and Pedro Lopez

I. INTRODUCTION

- Performance variability lead to users requesting nodes for longer times
- It affects the Operational efficiency of the HPC system
- With the fact that storage is getting cheaper, amount of system-related data being logged has increased considerably
- So, let’s use machine learning models to predict if a job in the scheduler queue will experience variation

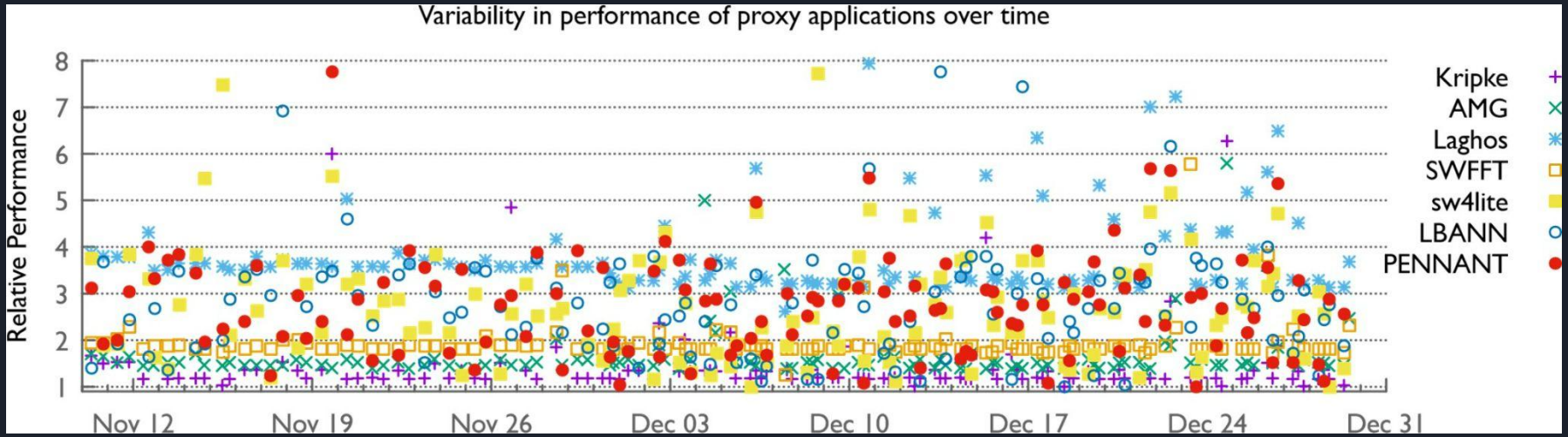
II. RELATED WORK

- Analyzing system monitoring data
- Studies looking at performance trends and root causes
- Existing work has looked at several ways to mitigate performance variability
- Previously people have used machine learning to predict how reliable user-provided run times are

III. DATA COLLECTION AND MODELING

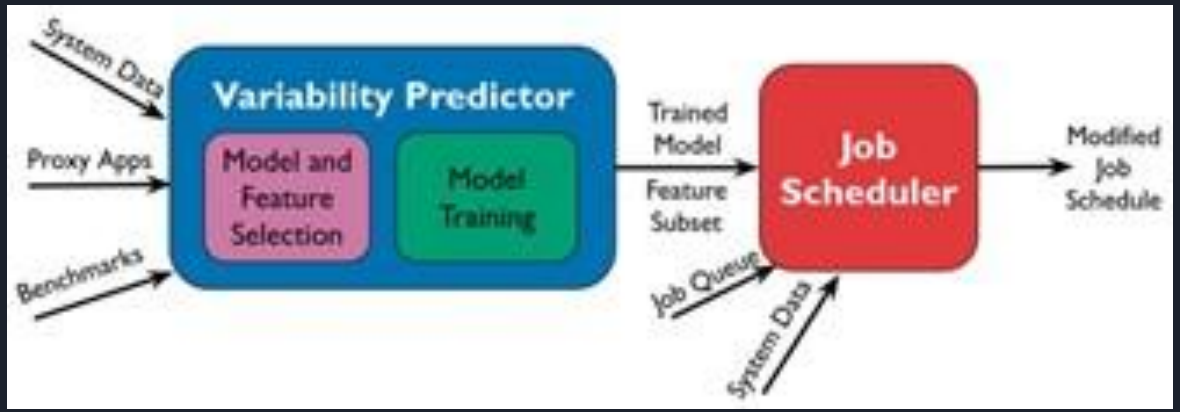
- Intentionally delaying the scheduling of I/O-intensive jobs, but why?
- We will require knowledge of system health and its relationship with application performance
- Let’s predict future occurrences of variability with statistical models.

Variability in the performance of proxy applications



IV. RUSH: RESOURCE UTILIZATION AWARE SCHEDULER FOR HPC - (i)
Variability Predictor Module

- A machine learning model is trained offline on historical jobs and system data
- The Variability Predictor Module uses system and control job data to predict Variation



IV. RUSH: RESOURCE UTILIZATION AWARE SCHEDULER FOR HPC- (ii)
Model-based Adaptive Job Scheduler

- The model-based adaptive job scheduler uses the models trained by the variability predictor as input and maps jobs from the queue to system resources.
- proposed scheduler utilizes a machine learning model to delay the scheduling of jobs that will experience variation.

V. IMPLEMENTATION

- Quartz cluster at Lawrence Livermore National Laboratory (LLNL)
- The scheduling algorithm considered predicted variation, system load, and resource utilization to make decisions
- RUSH demonstrated its ability to effectively reduce variation and maximum run time of applications while maintaining other performance metrics in a real-world HPC environment

VI. EXPERIMENTAL SETUP

- To evaluate the effectiveness of the ML models and the RUSH scheduler in mimicking typical workloads on an HPC system
- Tested the scheduling policy under various circumstances: different applications, training datasets, and scaling parameters
- Used proxy applications and models trained on datasets containing runs from multiple applications to assess the generalizability of the scheduler
- Makespan, reliability, resource utilization, and queue time were used to evaluate the performance of the models and the scheduler
- Multiple trials of each experiment were conducted to account for system noise

VII. RESULTS

- RUSH successfully reduced variation and maximum run time of applications compared to the baseline scheduling policy
- Demonstrated improvement in maximum run time, providing a tighter upper limit on applications' running time, which is beneficial for end-users
- The scheduler effectively mitigated performance variability without significantly impacting system throughput
- The results demonstrated the effectiveness of the RUSH system in reducing variation, improving job performance, and maintaining system throughput in a variety of experimental scenarios.

VII. RESULTS

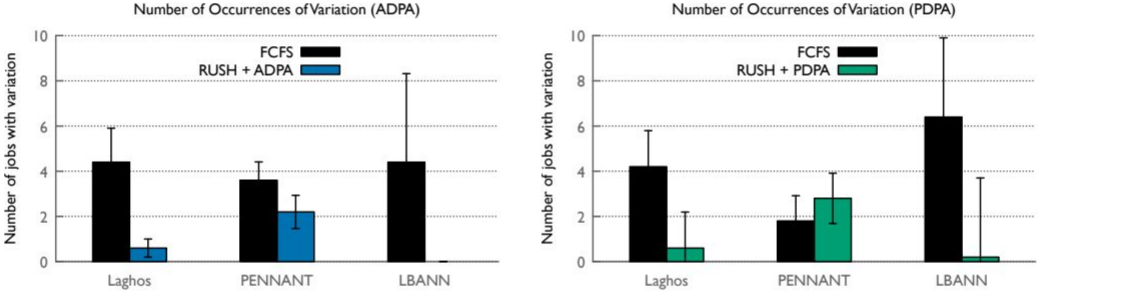


Fig. 4. There is only a slight increase in the number of applications experiencing variation when using the ML model trained on data from all of the applications (left plot, ADPA) and separate applications (right plot, PDPA.)

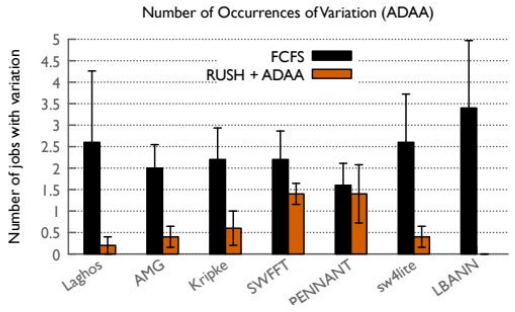


Fig. 5. The number of runs that experience variation significantly reduces under the proposed scheduler (RUSH) for the ADAA experiment when compared to FCFS+EASY.

VIII. CONCLUSION

- The RUSH system effectively mitigates performance variability, reduces maximum run time of applications without significantly impacting makespan or queue time, and improves job scheduling in HPC environments
- The models accurately predict job run time variation, providing valuable insights for the job scheduler to make intelligent scheduling decisions
- Demonstration of the system's generalizability to applications not included in the training data
- Future work could focus on further refining and optimizing the RUSH system, exploring additional features and models, and conducting experiments on larger-scale HPC systems to validate its scalability and generalizability

THANK YOU!

Questions?