Modeling, simulation and analysis of job scheduling policy changes on supercomputers

João Pedro Macleure Nunes dos Santos, Antônio Tadeu Azevedo Gomes

[1]Programa de Pós-Graduação em Modelagem Computacional
Laboratório Nacional de Computação Científica (LNCC)
Petrópolis – RJ – Brazil

Abstract. *Supercomputers are essential for advancing research in science and engineering. A key challenge in these systems is the heterogeneity of jobs, which complicates scheduling. The static nature of traditional algorithms used in commercial tools implies that scheduling policies must be often revisited. This work proposes a methodology to explore the scheduling configuration space. Our approach combines policy impact analysis with workload-based simulations. We introduce "job shaping" to adapt real workload traces to new policies. A case study on the Santos Dumont supercomputer illustrates our method. We show how policy changes affect different job types and user groups. Our results support the design of simulation-based optimization frameworks.*

1. Introduction

Supercomputers are high-performance computing (HPC) systems that play a pivotal role in advancing science. Since operating supercomputers often represent a significant financial investment, the owner institutions pay close attention to how their computational resources are used [Sterling et al. 2018]. Job schedulers are instrumental in this systems; they implement *scheduling policies* that allow managing and optimizing the allocation of computational resources to user tasks (*jobs*).

The workload of typical supercomputers found in national laboratories is heterogeneous, including tightly-coupled, message passing (MPI) applications, machine-learning applications, and bag-of-tasks (BoT) applications, to name some of the most prevalent today. This heterogeneity adds to the complexity of setting up the traditional algorithms (e.g. backfilling, multifactor prioritization) currently employed by commercial job schedulers. Besides, job scheduling policies are prone to be revisited regularly due to the turnover of users and the projects to which these users are associated.

Several pieces of work have analyzed the workload of supercomputers [Feng et al. 2018, Jakobsche et al. 2023, Wang et al. 2021]. One aspect largely uncovered in the surveyed work is the effect of job scheduling policy changes in both the job input variables (interarrival time between jobs, requested number of CPU cores, expected and actual wall-clock time of jobs) and the system output variables (time of jobs in queue, node utilization). Policy changes may happen to mitigate poor system output variables (e.g. long wait times, low utilization) [Rodrigo et al. 2018], to shape user behavior (e.g. better job wall-clock time estimations) [Hart 2022], or both.

Other pieces of work have developed *scheduling simulators* with different levels of accuracy and performance in the modeling of system behavior during a policy optimization process [Jokanovic et al. 2018, Galleguillos et al. 2020, Klusáček et al. 2020, Simakov et al. 2022]. Fidelity in representing *user behavior* in these simulators should be as important, though. Using real workload traces poses difficulties in this context because variations in policy may lead to valid jobs in the workload traces being incompatible with a policy being tested. Disregarding these jobs is not an option as it changes the actual workload imposed on the HPC system.

1.1. Objectives

In this work, we propose a methodology for exploring the configuration space of job scheduling, based on: (i) analyzing the effects of scheduling policy changes on user and system behavior; and (ii) applying *job shaping* in workload-based, discrete-event simulations to accommodate varying scheduling policies.

Using the Santos Dumont supercomputer[1] as a case study, we first reveal how a change in scheduling policy may have differential impacts on job types and user groups. We then show that our job-shaping strategy captures changes in system behavior regardless of mismatches between the workload traces input to the simulation and the tested policies.

2. Methodology for the empirical analysis

We analyze the empirical cumulative distribution function (ECDF) and statistics of the measures of interarrival time (IAT), job geometry (GEO), and queue wait time (QWT) during two adjacent time periods of a same length, considering three scenarios:

1. **Change of Policy (CP)**: the 1st time period corresponds to a "before-change" (BC) period, and the 2nd time period corresponds to an "after-change" (AC) period, in between of which a change of policy must have occurred;
2. **No Change of Policy in the BC period (NCP1)**: the 1st and 2nd time periods correspond to the 1st and 2nd halves of the BC period, without policy changes;
3. **No Change of Policy in the AC period (NCP2)**: the 1st and 2nd time periods correspond to the 1st and 2nd halves of the AC period, also without policy changes.

We assess whether the measures from the 1st and 2nd time periods come from different distributions, in each of these three scenarios, employing the two-sample Kolmogorov–Smirnov (KS) statistical test for each of the measures to compare the ECDF of the 1st period ($ECDF_{1st}(\cdot)$) to the ECDF of the 2nd period ($ECDF_{2nd}(\cdot)$). We analyze three different null hypotheses (H_0)—in all of them, p-value < 0.05:

1. **two-sided**: H_0 is that $\forall x, ECDF_{1st}(x) = ECDF_{2nd}(x)$, i.e. $ECDF_{1st}(\cdot)$ and $ECDF_{2nd}(\cdot)$ describe the same distribution;
2. **less**: H_0 is that $\forall x, ECDF_{1st}(x) \geq ECDF_{2nd}(x)$, i.e. $ECDF_{1st}(\cdot)$ produces values that are *at most* equal to the values $ECDF_{2nd}(\cdot)$ produces;
3. **greater**: H_0 is that $\forall x, ECDF_{1st}(x) \leq ECDF_{2nd}(x)$, i.e. $ECDF_{1st}(\cdot)$ produces values that are *at least* equal to the values $ECDF_{2nd}(\cdot)$ produces.

3. Results of the empirical analysis

Our analysis comprises the use of the Santos Dumont supercomputer in the period between 2017-06-15 and 2019-05-15, in between of which a policy change was employed. The change aimed to reduce wait times by creating a new queue for small jobs and adjusting the time limits of other queues to encourage users to better estimate job wait times.

[1] `http://sdumont.lncc.br`

3.1. User behavior

Observing the results of the KS test for the ECDFs of IAT and GEO, there is statistical significance that these measures in the 1st and the 2nd periods do not come from the same distribution—i.e. *two-sided* H_0 is refuted—*in all the three scenarios*. This means that we cannot attribute the change of user behavior *in general* to the change in the job scheduling policy. Indeed, the user behavior is multifactorial, being potentially influenced by variables external to the system (conference deadlines, vacations, etc). Evaluating these external influences is out of the scope of this work.

3.2. System response

In the results of the KS test for the ECDFs of QWT there is statistical significance that this measure in the 1st and 2nd periods do not come from the same distribution in *all the three scenarios*. This means we cannot assume that the QWT after the policy change tended to consistently assume values either shorter or longer than those before the policy change. However, if we take the quartiles of the ECDF into consideration, both Q2 and Q3 decrease in the CP scenario but increase in the NCP1 and NCP2 scenarios, thus suggesting an overall reduction in the QWT.

Taking a closer look at the ECDFs within each of the quartiles for the CP scenario, we observed that jobs with QWT shorter than 5,360 secs suffered from an increased QWT after the change in the job scheduling policy, whereas the other jobs benefited from a decreased QWT after this change. This observation is consistent with the inconclusive results presented for the CP scenario. We then made a cohort analysis to separately evaluate the impact of the policy change on the three measures for jobs with shorter QWT (the 1st cohort) and for jobs with longer QWT (the 2nd cohort).

To assess the correlation between IAT, GEO, and QWT we used the Spearman correlation. The Spearman coefficients calculated over the BC period between IAT and QWT (-0.176) and between GEO and QWT (0.441), although low, suggest expected correlations between the three measures, i.e. a decrease (resp. increase) in the IAT indicates that more (resp. fewer) jobs were submitted in a time frame, which could lead to a potential increase (resp. decrease) in QWT; the same behavior in QWT would be expected if the GEO increased (resp. decreased).

3.3. First cohort: jobs with short QWT

We could not reject the *less* H_0 hypothesis for the QWT in the 1st cohort, confirming that the jobs in this cohort consistently suffered from an increased QWT after the change in policy. Nevertheless, it is not possible to ascertain that the reason for the increase in QWT is due to the change in user behavior (IAT and GEO). The KS tests only indicated that there is a difference in the distributions of IAT and GEO, without indicating whether such measures tend to have lower or higher values when comparing the 1st and 2nd periods.

We also observed that the IAT is longer in the 1st period for Q1 and Q2, being shorter in the remainder quartiles. As for the GEO, we observed larger values in the 1st period for Q1, Q2 and Q3, and smaller values in Q4. The small differences in the Spearman coefficients calculated over both periods (ρ_{1st} and ρ_{2nd}) between IAT and QWT ($\rho_{1st} = -0.237$ and $\rho_{2nd} = -0.341$) and between GEO and QWT ($\rho_{1st} = -0.178$ and $\rho_{2nd} = -0.211$) confirm that we cannot establish a clear connection between user behavior and system response by looking at the three measures alone in this cohort.

3.4. Second cohort: jobs with long QWT

We could not reject the *greater* H_0 hypothesis for the QWT in the 2nd cohort either, confirming that the jobs in this cohort consistently benefited from a decreased QWT after the change in the job scheduling policy. Besides, and differently from the 1st cohort, we could not reject the *less* H_0 hypothesis for the IAT in the 2nd cohort, which means that the IAT in this cohort was consistently higher after the policy change.

We also observed that the GEO in the 2nd period *seemed* to have lower values than in the 1st period, as seen from Q1, Q2, and Q4. The noticeable increase in the Spearman coefficient between GEO and QWT after the policy change ($\rho_{1st} = 0.297$ and $\rho_{2nd} = 0.741$) confirms this trend, but again we could establish a clear connection between user behavior and system response by looking at the three measures alone.

3.5. Preliminary conclusions

The amount and variety of users of the Santos Dumont supercomputer is too large to allow for the identification of a single pattern relating user behavior and system response. So, we explored the organization of the users in projects to ascertain if groups of users reacted differently to—and were positively or negatively affected by—the change in policy.

3.6. Impact of user groups

We took the 36 projects with the largest amount of jobs. For each one of them we computed an *AC-to-BC ratio* ($A2Br$) for the three measures, defined as the median of the measure in the AC period divided by the median of the measure in the BC period.

In the 1st cohort, a large number of projects had $A2Br > 1$ for the QWT. Besides, the Spearman coefficient between the AC-to-BC ratios for the GEO and the QWT in this cohort (-0.461) sounds counter-intuitive. This may reflect the fact that jobs with shorter QWT tend to have smaller GEO as well, since they are more easily scheduled by the backfilling algorithm. On the other hand, the Spearman coefficient between the AC-to-BC ratios for the QWT and the IAT (-0.33) confirms an expected trend.

In the 2nd cohort, about half of the projects had $A2Br < 1$ for the QWT. Moreover, the Spearman coefficients between the AC-to-BC ratios for the GEO and the QWT (0.449) and also for the QWT and the IAT (-0.354) confirm expected trends.

We acknowledge that using a simple statistic such as the median may not be a proper measure of the users' behavior and the system's response in a project context. Besides, using projects may not be the best criterion for grouping users, because of the variations in their behavior even within projects. Exploring alternatives to the two issues above was left for future work.

4. Workload-based simulator

We used SimPy[2] as our discrete event simulation engine, adding to it domain entities representing computational nodes, queues, job generators (using known probability distributions, histogram sampling, or real workload traces as sources of jobs), and

[2] https://simpy.readthedocs.io/

schedulers (supporting backfilling and multifactor priorization).[3] Besides, we added a "job-shaping" entity, responsible for bringing jobs from real workload traces into compliance with policies being tested as part of an optimization of a scheduling policy configuration. The shaping of a job occurs by means of a change in the ratio of its geometry factors, while keeping the actual value of the geometry unchanged, thus allowing the capture of strong scaling properties of supercomputing applications.

5. Results of the workload-based simulator

Using the same workload traces described in Section 3 as input to our simulator, we identified a set of jobs from the BC period that were not processed with the AC period configuration due to queue configurations that limited the geometry of the jobs to a certain interval, when job shaping was not employed. We then assessed two simple scenarios of shaping these unprocessed jobs considering new policies: (i) doubling the amount of cores and halving the time limit for a short-lived jobs' queue; or (ii) halving the amount of cores and doubling the time limit for a long-lived jobs' queue. This shaping ensured that in each scenario, jobs not processed in the workload injection were processed by the simulator in an alternative queue, maintaining the requested geometry. Moreover, by comparing the ECDFs of the QWT obtained through real data and through the simulator, we observed a fairly similar behavior, specially in the quartiles Q2, Q3, and Q4. We could even observe the simulation mimicking the different effects of the change in policy in jobs with shorter and longer QWT, as described in Subsection 3.2, albeit with a different crossing point.

6. Conclusion

In this work, we developed a methodology to characterize user behavior and system response after changes in scheduling policies. On applying such methodology to the examination of the workload traces of the Santos Dumont supercomputer over two years, we found out that policy changes affected jobs with short and long wait times differently, illustrating the nuanced effects of scheduling adjustments. Additionally, grouping users according to the projects they belong to proved beneficial in interpreting changes in their behavior, offering a more granular understanding of the responses of both users and systems to policy changes. We believe these results contribute to the broader understanding of supercomputing workload management by shedding light on the distribution shifts in job geometry and interarrival times due to policy changes. Ultimately, these results emphasize the need for adaptive, data-driven scheduling algorithms to meet the evolving demands of scientific and technological research in supercomputers, as previous research in the area pointed out [Carastan-Santos and de Camargo 2017, Tsafrir et al. 2007].

We also presented a job-shaping strategy that accommodates real workload traces to incompatible policies that may arise in optimization procedures based on discrete-event simulations. Our main finding with the case study involving the Santos Dumont supercomputer was that job shaping allows capturing changes in system behavior simply and effectively, regardless of eventual incompatibilities between the real workload traces and the simulated policies. In this way, parameterizing the job-shaping strategy allows for an in-depth evaluation of optimization opportunities, offering insights into how configurations impact overall system performance.

[3] The code is available at `https://gitlab.com/itdf/hpc_sim`.

References

Carastan-Santos, D. and de Camargo, R. Y. (2017). Obtaining dynamic scheduling policies with simulation and machine learning. In *SC17: Int'l Conference for High Performance Computing, Networking, Storage and Analysis*.

Feng, J., Liu, G., Zhang, J., Zhang, Z., Yu, J., and Zhang, Z. (2018). Workload characterization and evolutionary analyses of Tianhe-1A supercomputer. In Shi, Y., Fu, H., Tian, Y., Krzhizhanovskaya, V. V., Lees, M. H., Dongarra, J., and Sloot, P. M. A., editors, *Computational Science – ICCS 2018*, pages 578–585, Cham. Springer Int'l Publishing.

Galleguillos, C., Kiziltan, Z., Netti, A., and Soto, R. (2020). AccaSim: a customizable workload management simulator for job dispatching research in HPC systems. *Cluster Computing*, 23(1):107–122.

Hart, D. (2022). Measuring XSEDE: Usage metrics for the XSEDE federation of resources: A comparison of evolving resource usage patterns across TeraGrid and XSEDE. In *Practice and Experience in Advanced Research Computing*, New York, NY, USA. Association for Computing Machinery.

Jakobsche, T., Lachiche, N., and Ciorba, F. M. (2023). Investigating hpc job resource requests and job efficiency reporting. In *22nd Int'l Symposium on Parallel and Distributed Computing (ISPDC)*.

Jokanovic, A., D'Amico, M., and Corbalan, J. (2018). Evaluating SLURM simulator with real-machine SLURM and vice versa. In *2018 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, pages 72–82. IEEE.

Klusáček, D., Soysal, M., and Suter, F. (2020). Alea – complex job scheduling simulator. In Wyrzykowski, R., Deelman, E., Dongarra, J., and Karczewski, K., editors, *Parallel Processing and Applied Mathematics*, pages 217–229, Cham. Springer International Publishing.

Rodrigo, G. P., Östberg, P.-O., Elmroth, E., Antypas, K., Gerber, R., and Ramakrishnan, L. (2018). Towards understanding HPC users and systems: A NERSC case study. *Journal of Parallel and Distributed Computing*, 111:206–221.

Simakov, N. A., Deleon, R. L., Lin, Y., Hoffmann, P. S., and Mathias, W. R. (2022). Developing accurate Slurm simulator. In *Practice and Experience in Advanced Research Computing 2022: Revolutionary: Computing, Connections, You*. Association for Computing Machinery.

Sterling, T., Anderson, M., and Brodowicz, M. (2018). Chapter 5 - the essential resource management. In Sterling, T., Anderson, M., and Brodowicz, M., editors, *High Performance Computing*, pages 141–190. Morgan Kaufmann, Boston.

Tsafrir, D., Etsion, Y., and Feitelson, D. G. (2007). Backfilling using system-generated predictions rather than user runtime estimates. *IEEE Transactions on Parallel and Distributed Systems*, 18(6):789–803.

Wang, Q., Shen, Y., and Li, J. (2021). User-level workload analysis for supercomputers. In *4th Int'l Conference on Software Engineering and Information Management*, page 68–73, New York, NY, USA. Association for Computing Machinery.