Job Grouping-Based Intelligent Resource Recommendation Framework

Beste Oztop
boztop@bu.edu
Boston University
Boston, Massachusetts, USA

Benjamin Schwaller
bschwal@sandia.gov
Sandia National Laboratories
Albuquerque, New Mexico, USA

Vitus J. Leung
vjleung@sandia.gov
Sandia National Laboratories
Albuquerque, New Mexico, USA

Jim Brandt
brandt@sandia.gov
Sandia National Laboratories
Albuquerque, New Mexico, USA

Brian Kulis
bkulis@bu.edu
Boston University
Boston, Massachusetts, USA

Manuel Egele
megele@bu.edu
Boston University
Boston, Massachusetts, USA

Ayse K. Coskun
acoskun@bu.edu
Boston University
Boston, Massachusetts, USA

Abstract

In the current large-scale computing systems, users from various scientific backgrounds submit batch jobs with a set of requested resources. Manual resource selection in HPC facilities leads to early job terminations and out-of-memory errors due to underestimation of resources, or compute and memory resources sitting idle because of overallocation. In this work, we provide a resource recommendation framework based on job grouping and intelligent prediction methods to provision HPC application resource needs before they are submitted to the system. Our work achieves less than 2% of cases experiencing underpredicted resource requests, and results in fewer overestimations compared to the baseline methods. We also implement a module to deploy the framework on a real HPC system, which comprises the future plans of this work.

CCS Concepts

• **Computer systems organization → High-performance computing**; **Resource allocation**; • **Computing methodologies → Machine learning**; **Supervised learning by regression**.

Keywords

High-Performance Computing, Resource Management, Data-Driven Resource Prediction

ACM Reference Format:
Beste Oztop, Benjamin Schwaller, Vitus J. Leung, Jim Brandt, Brian Kulis, Manuel Egele, and Ayse K. Coskun. 2025. Job Grouping-Based Intelligent Resource Recommendation Framework. In *Proceedings of the SC '25 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis (SC-W'25).* ACM, New York, NY, USA, 2 pages.

1 Summary

High-Performance Computing (HPC) provides researchers from many scientific domains with reliable, fast, and efficient solutions. The performance and efficiency of these large-scale systems are affected by the resource utilization trends. HPC system users make use of the available compute and memory resources by submitting their batch jobs to the corresponding system's workload manager. While doing so, they request resources based on their best knowledge. This manual selection of resources leads to two main problems: under- and over-predictions. In this work, we aim to present machine learning-based and data-driven solutions to both of these drawbacks using historical workload manager logs.

Overprediction of Resources. Users often request extra resources for their batch jobs to prevent them from being killed due to out-of-time or out-of-memory errors. Unnecessarily allocating nodes and requesting more hard wallclock time than needed negatively impacts the scheduling performance. The waiting time and overall response time for the user increase, and idle nodes lower the overall throughput of the system [7].

Underprediction of Resources. Requesting fewer resources than a batch job requires may lead to early job termination, reduced user satisfaction, and lower efficiency in HPC systems. This problem is due to the challenging nature of predicting a job's resource needs, especially when no prior job submissions exist. This is the second problem we aim to solve using a machine learning and job grouping-based prediction framework.

This recommendation framework leverages machine learning techniques to make intelligent resource allocation suggestions to users for their batch job submissions and aims to reduce the abovementioned root causes of resource utilization inefficiencies in HPC systems. The framework consists of two components: (1) an offline stage for training the prediction model and (2) an online stage for making recommendations.

In the offline model training stage, we use the jobs that have been successfully completed. We use 5 different real-world workload manager datasets in our experimental procedure; namely, National Renewable Energy Laboratory (NREL) Eagle [4], Fugaku [2], Marconi100 [1], Sandia, and Boston University Shared Computing Cluster accounting data. We first label-encode the categorical submission features (e.g., user, job, or project name) in these datasets. Next, we group similar jobs according to their submission parameters using K-means clustering from the `scikit-learn` library [6]. We then train a Random Forest Regressor individually for each of the data clusters.

In the training stage, we use two strategies to avoid the underprediction problem that arises due to regression models. First, we calculate a buffer value based on the amount of underprediction observed in the validation dataset. The buffer value is set to 2σ, where σ is the standard deviation of underpredictions. Our goal is to minimize underpredictions as much as possible in the testing stage by adding this buffer value. Secondly, we alter the training dataset distribution through resampling the data. By doing so, we aim to have a more uniform distribution in the training set by increasing the number of occurrences of less-frequent data points. We represent these methods as 2σ and resampling in our results.

To test our framework's performance, we use the following metrics, where UR stands for the underprediction ratio and OF refers to the overestimation factor:

$$\text{UR}) = 100 \times \frac{\text{Number of Jobs with Underprediction}}{\text{Total Number of Jobs}}, \tag{1}$$

$$\text{OF} = \frac{\text{Requested Resource of Predicted Utilization}}{\text{Actual Resource Utilization}}. \tag{2}$$

We determine the number of clusters for each experimental dataset using the elbow method and perform time-based train-test splits to identify the most suitable train-to-test ratio for our framework. Applying the resampling and 2σ strategies in the training stage, we conduct inference experiments on the execution time, maximum memory usage, and processor requirements using historical data. We have two baseline methods: (1) User resource requests if they exist in the corresponding dataset, and (2) applying a single XGBoost model training as Menear et al. [4] propose.

Our results show a reduction to as low as 1.82% in terms of UR, along with fewer overpredictions compared to the user-requested values. Furthermore, we reduce the prediction error in our target variables compared to the baseline method, where no job grouping exists. Observing the previous trends in our experimental datasets, we conclude that this framework can reduce the resource utilization inefficiencies caused by manual resource selection and potentially reduce wait times for HPC system users.

We also designed and implemented an Lmod [3] environment module on BU SCC, enabling users to receive resource recommendations from our framework based on their previous batch job submissions. This module introduces a low overhead for training the machine learning models and aims to improve overall cluster resource utilization. We include the instructions on how to use the module in this poster as well [5].

As future work, we will focus on deploying this recommendation framework in collaboration with researchers from various scientific domains on a real-world HPC system. We also plan to evaluate the framework's performance on workload manager simulators using our experimental datasets. Our overarching goal is to have more resilient and efficient HPC systems by improving the accuracy of resource predictions.

Acknowledgments

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525

References

[1] Francesco Antici, Andrea Bartolini, Jens Domke, Zeki Kiziltan, and Kento Yamamoto. 2024. *F-DATA: A Fugaku Workload Dataset for Job-centric Predictive Modelling in HPC Systems.* doi:10.5281/zenodo.11467483 Data set.

[2] Andrea Borghesi, Carlo Di Santi, Miha Molan, et al. 2023. M100 ExaData: a data collection campaign on the CINECA's Marconi100 Tier-0 supercomputer. *Scientific Data* 10 (2023), 288. doi:10.1038/s41597-023-02174-3

[3] Robert McLay, Karl W Schulz, William L Barth, and Tommy Minyard. 2011. Best practices for the deployment and management of production HPC clusters. In *State of the Practice Reports.* 1–11.

[4] Kevin Menear, Ambarish Nag, Jordan Perr-Sauer, Monte Lunacek, Kristi Potter, and Dmitry Duplyakin. 2023. Mastering HPC Runtime Prediction: From Observing Patterns to a Methodological Approach. In *Practice and Experience in Advanced Research Computing 2023: Computing for the Common Good* (Portland, OR, USA) *(PEARC '23)*. Association for Computing Machinery, New York, NY, USA, 75–85. doi:10.1145/3569951.3593598

[5] PEACLab (Performance and Energy Aware Computing Laboratory, Boston University). 2025. qsub-rec-module: BU SCC Collaboration for Intelligent Resource Prediction Framework. https://github.com/peaclab/qsub-rec-module. Updated June 13, 2025.

[6] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine Learning in Python. *the Journal of Machine Learning Research* 12 (2011), 2825–2830.

[7] Mohammed Tanash, Huichen Yang, Daniel Andresen, and William Hsu. 2021. Ensemble prediction of job resources to improve system performance for slurm-based hpc systems. In *Practice and experience in advanced research computing.* Association for Computing Machinery, 1–8.