ML Based Modeling of XSEDE/ACCESS Usage and Performance data for HPC Job Wait Time Estimates and Resource Allocation

Abani Patra
Tufts Institute for AI,
Tufts University Medford, MA 02155
abani.patra@tufts.edu

Bipin Gaikwad
Tufts Institute for AI,
Tufts University, Medford, MA 02155
bipin.gaikwad@tufts.edu

Nikolay Simakov
Center for Computational Research
University at Buffalo, Buffalo, NY 14260
nsimakov@buffalo.edu

Joseph White
Center for Computational Research
University at Buffalo, Buffalo, NY 14260
jpwhite4@buffalo.edu

Thomas Furlani
Center for Computational Research
University at Buffalo, Buffalo, NY 14260
furlani@buffalo.edu

I. Stakeholders and QoS

Our research goal is effective data driven resource allocation, deployment and usage strategies for HPC to meet multiple stakeholder goals. We identify the following primary stakeholders: a) users, b) resource owners, and, c) resource managers. Users want to minimize job throughput with minimal queue wait times while resource owners wish to minimize total cost – capital and operations. Resource managers have to balance both users and resource owners' interests! The recent workload transformation with AI workloads stretching HPC resources makes optimal resource usage necessary. Analytics support for making these complex decisions is therefore critical. The complex mix of user demand (job types, temporal fluctuations) and resource dynamicity and complexity makes this a challenging problem. Quality of Service (QoS) measures can include job queue wait time. In the case of resource managers operations and investment choice policy decisions are critical.

II. Background and Related Efforts

HPC systems and their use is carefully monitored by tools like XDMOD[1], [2], [3], [6], [7] developed in our group's earlier work which has O(100) established users worldwide. The integrated analysis of system usage and performance data allows to build careful models that maximize system use and optimize user quality of service. Related efforts have arisen in Europe [8] and elsewhere especially in the closely related datacenter energy and impact management space which while distinct shares some of the data needs on large scale computing.

III. Method and Approach

We address this challenge by developing predictive models that estimate quantities of interest like user wait times using historical scheduling data and real-time system metrics available from our work in metrics and measurements of the NSF supercomputing systems in ACCESS and earlier XSEDE networks (see for e.g. XDMod [1], [2], [3]). We collect high-resolution operational metrics from ACCESS XDMoD [2]. Models are built using the parameters available at the time of job submission, ensuring that predictive models rely primarily on real-time accessible features. The dataset includes key job submission attributes such as 'Requested Wall Time,' 'Requested Nodes,' 'Cores,' 'GPU Count,' 'Queue,' and 'Field of Science,' with 'Wait Time' as the target variable. We augment the raw data with system state metrics like current queue occupancy as the number of jobs waiting at the submission time and real-time resource utilization metrics, i.e., 'Number of Active Jobs Running', 'Active CPU Cores Utilization', and 'Node Usage Running' to capture system-wide workload conditions. While, data from as far back as 2010 is available for a range of HPC our early efforts have been focused on recent data. Other data available including 'Memory Usage' also has predictive power as our ablative studies indicate. This data is used in both Random Forest and Multi Layer Perceptron models.

A second class of analyses we have started exploring of particular interest to resource managers has been the incremental benefit of additional resources. For example, the tradeoff between additional CPU, GPU and memory resources and their impact on usage metrics like job wait time. Our strategy here has been to use the historical job traces and rerun them under a standard sceheduler like First Come First Served (FCFS) and hypothetical additional resources (20% more CPU or 20% more GPU etc.) and see how measures of QoS improve.

IV. Preliminary Results

Full Training

The first dataset we used was data from one month of 396,879 jobs, split 80% training and 20% testing, ensuring

a representative HPC workload sample. With approximately 94% accuracy on wait time predictions within a 10-minute error margin, our models can contribute to more transparent and efficient scheduling, fostering faster scientific discovery and innovation. Furthermore, in a test, a sample job had predicted wait times ranging from 15 seconds on Anvil-GPU to 99,201 seconds on Bridges2-GPU, and up to 231,453 seconds across different resource choices, under the same job parameters specified [4]. These endpoints reflect differences in resource choice and real-time system conditions, as shown in Table 4 of [4].

Our next target is ongoing work on reliable individual job level prediction of job wait times on multiple resources using only historical data. This will enable users to have more realistic data driven real-time forecast at job submission. We accomplish this by using what we term sliding window training.

Sliding Window Training

We use a sliding window-for generating train and test data. We slide the data window of duration 7 days and 5 minutes of which 7 days is used for training and 5 minutes data is used for testing. The stride for sliding is 5 minutes. The data was collected from one resource anvil in the months 10-2025 and 11-2025. In our first test using 7 days data for training and 5 minutes of data for testing where the training data contains all the jobs that were submitted before the test data window the overall prediction accuracy where the predicted time was within 25% of actual was 50.57% Using only data from jobs that have completed the overall accuracy was 26.52%.

Resource Augmentation

Our early results from this indicate that for most current NSF ACCESS resources adding 20% GPU resources will greatly improve throughput as measured by a normalized user wait time. However, results from data of a single institution was less conclusive where adding a 20% of either CPU of GPU had minimal impact on throughput. Additional details of this are in our forthcoming publication [5].

V. Discussion and Current Work

The current models use key job parameters and performance measures to predict wait times for specific jobs and offer wait time estimates across resources, enabling informed, resource selection by users. This can reduce delays, improve job throughput, and enhance overall cluster efficiency. However, much more needs to be done before we can deliver robust QoS estimates. Some challenges for ongoing work:

- Our historical data comes from HPC that had very different architectures and job characteristics. Nevertheless, we believe there is useful knowledge we can derive from this data though it will better strategies for normalization and harmonization prior to modeling than we currently use.
- Wait time is clearly very dependent on current machine state (in terms of running jobs) and modeling will have to reflect that. We are experimenting with other modeling methods where we can weight more recent data stronger and have constant updates of live data.
- Resource augmentation results are very dependent on scheduling policy. We need to replace the FCFS we have used with more realistic resource specific scheduling simulator.

While promising, further work is needed to refine predictions and expand applicability to complex scheduling scenarios. In future, additional data will be incorporated, with newer data reserved for testing to evaluate model generalization.

References

[1] University at Buffalo Center for Computational Research. 2025. XDMoD: Open XDMoD. https://github.com/ubccr/

[2] Jeffrey T Palmer, Steven M Gallo, Thomas R Furlani, Matthew D Jones, Robert L DeLeon, Joseph P White, Nikolay Simakov, Abani K Patra, Jeanette Sperhac, Thomas Yearke, et al. 2015. Open XDMoD: A tool for the comprehensive management of high-performance computing resources. Computing in Science & Engineering 17, 4 (2015), 52–62.

[3] Aaron Weeden, Joseph P White, Robert L DeLeon, Ryan Rathsam, Nikolay A Simakov, Conner Saeli, and Thomas R Furlani. 2024. The data analytics framework for XDMOD. SN Computer Science 5, 5 (2024), 462.

[4] Bipin Gaikwad, Nikolay A. Simakov, Thomas Furlani, Joseph Patrick White, and Abani Patra. 2025. Predictive Modeling of HPC Job Queue Times: Improving User Decision-Making and Resource Utilization. In Practice and Experience in Advanced Research Computing (PEARC '25), ACM, July 2025.

[5] Bipin Gaikwad, S. Singh, M. Joshi, Nikolay A. Simakov, Thomas Furlani, Joseph Patrick White, and Abani Patra Analyzing HPC Job Wait Times under Resource Scaling Using Historical Workload Data, to appear, PEARC'26, ACM.

[6] J. C. Browne, R. L. DeLeon, C. Lu, M. D. Jones, S. M. Gallo, A. Ghadersohi, A. K. Patra, W. L. Barth, J. Hammond, T. R. Furlani, R. T. McLay, Enabling comprehensive data-driven system management for large computational facilities, in: Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis (SC '13), ACM, 2013, p. 11. doi:10.1145/2503210.2503230.

[7] N. A. Simakov, J. P. White, R. L. DeLeon, A. Ghadersohi, T. R. Furlani, M. D. Jones, S. M. Gallo, A. K. Patra, Application kernels: HPC resources performance monitoring and variance analysis, Concurrency and Computation: Practice and Experience 27 (17) (2015) 5238–5260, cPE14-0402.R1. doi:10.1002/cpe.3564. URL https://dx.doi.org/10.1002/cpe.3564

[8] A. Borghesi, C. Di Santi, , M. Molan, et al. M100 ExaData: a data collection campaign on the CINECA's Marconi100 Tier-0 supercomputer. Sci Data 10, 288 (2023). https://doi.org/10.1038/s41597-023-02174-3 Download citation