2024 IEEE 24th International Symposium on Cluster, Cloud and Internet Computing Workshops (CCGridW) | 979-8-3503-7751-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/CCGridW63211.2024.00015

Towards an Integrated Performance Framework for Fire Science and Management Workflows

Hena Ahmed*
Halıcıoğlu Data Science Institute
University of California, San Diego
La Jolla, CA, USA
h7ahmed@ucsd.edu

Ravi Shende*
San Diego Supercomputer Center
University of California, San Diego
La Jolla, CA, USA
rshende@ucsd.edu

Ismael Perez
San Diego Supercomputer Center
University of California, San Diego
La Jolla, CA, USA
i3perez@sdsc.edu

Daniel Crawl
San Diego Supercomputer Center
University of California, San Diego
La Jolla, CA, USA
lcrawl@ucsd.edu

Shweta Purawat
San Diego Supercomputer Center
University of California, San Diego
La Jolla, CA, USA
shpurawat@ucsd.edu

İlkay Altintaş
San Diego Supercomputer Center and
Halıcıoğlu Data Science Institute
University of California, San Diego
La Jolla, CA, USA
ialtintas@ucsd.edu

***Abstract*—Reliable performance metrics are necessary prerequisites to building large-scale end-to-end integrated workflows for collaborative scientific research, particularly within context of use-inspired decision making platforms with many concurrent users and when computing real-time and urgent results using large data. This work is a building block for the National Data Platform, which leverages multiple use-cases including the WIFIRE Data and Model Commons for wildfire behavior modeling and the EarthScope Consortium for collaborative geophysical research. This paper presents an artificial intelligence and machine learning (AI/ML) approach to performance assessment and optimization of scientific workflows. An associated early AI/ML framework spanning performance data collection, prediction and optimization is applied to wildfire science applications within the WIFIRE BurnPro3D (BP3D) platform for proactive fire management and mitigation.**

***Index Terms*—Cyberinfrastructure, Workflows, Performance Analysis, Artificial Intelligence, Machine Learning**

I. Introduction

Scientific application workflows have become a key tool in natural disaster mitigation and response. Real-time sensor and satellite data now provide invaluable resources for urgent science analytics to be conducted with remarkable speed and precision. Workflows and larger cyberinfrastructures (CIs) powered by such data can deliver critical knowledge about imminent natural hazards such as wildfires [1], earthquakes [2], and volcanic eruptions [3].

However, the vast influx of raw and pre-processed data from geo-distributed sources presents challenges to the design of scalable cyberinfrastructures for data to knowledge workflows, thus heightening the need for developing a computing continuum of integrated cloud-to-edge resources [4]. A computing continuum especially enables novel implementations of urgent application workflow with particular attention to efficient data processing, and reliable but timely data to knowledge transfer to support urgent decision-making [5].

*Equal contribution

Earlier works presented the WIFIRE cyberinfrastructure of integrated end-to-end workflows for wildfire behavior modeling [6], as well as a use case of the computing continuum to support data-driven workflows for air quality prediction to manage wildfire impacts [7]. The WIFIRE Commons itself is one such use-case for the National Data Platform project, which will leverage the computing continuum to democratize scientific data access and analysis through a national cyberinfrastructure [8]. Fig 1 outlines an early performance pipeline for the National Data Platform (NDP).

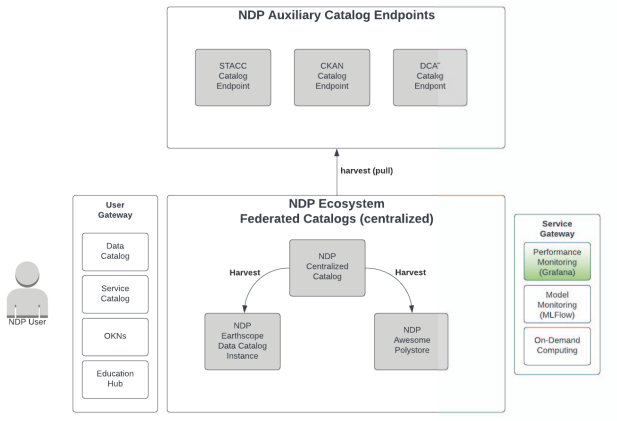

Fig. 1

Our work particularly addresses the issue of performance optimization for the WIFIRE-powered BurnPro3D (BP3D) platform [9], [10]. BP3D is a decision support platform to inform and optimize prescribed burn planning for wildfire management. The platform works in tandem with other WIFIRE frameworks, namely QUIC-Fire fire and atmospheric models and FastFuels 3D fuel structure models, in order to

979-8-3503-7751-4/24/$31.00 ©2024 IEEE
DOI 10.1109/CCGridW63211.2024.00015

identify environmental conditions and ignition patterns that are optimal for prescribed burns.

BP3D is a user-facing tool for geographically distributed land managers and fire planners, and is applied in a variety of computing environments with different capabilities. In this paper, we present preliminary steps towards a performance prediction framework that will be used by BP3D users to assess the necessary resource provisions to run BP3D given environmental and fuel data inputs. We present a performance framework for integrating data processing and AI/ML techniques in order to predict resource consumption during BP3D runs, and thereby improve the scalability and reliability of underlying BP3D data workflows. We implement a data processing architecture to collect and prepare performance data for AI/ML analysis, and demonstrate examples of predictive modeling techniques for performance evaluation and provision of our AI-driven data workflows. This is part of a larger work to optimize sub-systems for integrated end-to-end data cyberinfrastructures, which is critical to enabling efficient data processing and modeling across the computing continuum.

The rest of the paper is structured as follows: Section II discusses related works in the existing scientific literature, Section III describes a use case application of our workflow architecture, Section IV describes our methodological approach and design, Section V demonstrates early results of our framework, and Section VI summarizes conclusions and future endeavors of this work.

II. Related Works

Large-scale data cyberinfrastructures are next generation platforms for collaborative research workflow and data sharing. Beckman et al. [11] highlights the need for performance optimization across the computing continuum. Current case studies for scalable CIs include the Virtual Data Collaboratory for interdisciplinary data and science sharing presented by Parashar et al. [8], and the EarthScope [3] framework for open access, real-time geophysical data, modeling, and educational services. Our work also draws from contributions in distributed and multimodal data architectures such as the Quantum Data Hub presented by Purawat et al. [12] and the AWESOME polystore using open-knowledge networks (OKNs) presented by Dasgupta & Gupta [13].

Nguyen et al. [14] presented methods for integrating machine learning techniques in scientific workflow systems to evaluate accuracy and scalability. As described in Parashar et al. [8], the GeoSciFramework (commonly known as EarthScope) is one such case study for scalable architectures of scientific workflows and integrative machine learning environments that operate with continuously streaming geodetic and seismic data. However, an existing key problem area in developing scalable architectures for integrated machine learning and scientific workflows is developing knowledge management techniques to assimilate and prepare data from sources for AI/ML analysis. This concept is also referred to as the "AI-readiness" of data. AI-readiness is especially important for urgent computing applications such as natural hazard modeling and prediction-making. Baru et al. [15] are currently addressing the challenge of finding and matching AI-ready data and models in an integrated platform, while also following guidelines for FAIR [16] for data provenance. Holding scientific work to FAIR data management principles (where FAIR stands for Findability, Accessibility, Interoperability, and Reusability) is a key step to ensuring the responsible deployment of AI models and other data services.

By building upon existing work in integrated AI/ML and scientific workflow architectures as well, this paper will further previous research towards global, integrated cyberinfrastructures that enable equitable data-driven technology sharing.

III. Workflow Use Case

The use case application described in this section was created to predict total resource consumption of BurnPro3D simulations for prescribed burns and wildfire mitigation [9]. An execution of BP3D takes a single set of environmental input data and runs an ensemble of simulations over multiple Kubernetes servers. We created an integrated ML/AI workflow that retrieves the input parameter values given to a BP3D run and resource consumption data that is generated throughout the run and stored on Nautilus servers. The workflow then takes a linear regression approach to predicting total CPU and memory usage of a BP3D run.

For the purposes of this paper, the chosen machine learning method (linear regression) is rather arbitrary, as our intentions at this stage of research are to demonstrate a functional data-driven pipeline for AI/ML performance analysis, as opposed to choosing the most accurate or robust modeling approach for the task. So, we used a basic linear regression implementation with default model parameter values made accessible using the popular Python package Scikit-Learn. In future extensions of this work, we plan to conduct more extensive sensitivity analyses and parameter estimation techniques in order to implement modeling techniques that better fit the given data. The results of this particular use case are discussed in Section V.

IV. Approach

In this section, we describe our methodology for retrieving resource consumption data and integrating AI/ML-informed decision-making to analyze overall performance of the BurnPro3D architecture for prescribed burn modeling. We describe two primary objectives in developing our AI/ML solution: 1) data preparation for AI-readiness (Section IV-A), and 2) integration of predictive ML/AI modeling methods into the scientific workflow architecture of BP3D (Section IV-B).

A. AI-Ready Data Preparation

We first discuss the steps that were taken towards reaching our AI-readiness objective. Data assimilation and preparation requirements will vary depending on the form of ML/AI analysis being applied. The use case we describe in this paper demonstrates results obtained through linear regression. So, in this section, we discuss the steps taken to achieve data that is AI-ready in the specific context of linear regression modeling.

TABLE I: BurnPro3D Inputs/Outputs

Feature Name	Description
surface_moisture	surface fuel moisture
wind_moisture	fuel moisture of surface winds
wind_direction	direction of surface winds
wind_speed	speed of surface winds
sim_time	estimated minimum runtime (seconds)
timestep	elapsed seconds between simulation steps
run_max_mem_rss_bytes	maximum RSS bytes allowed per run
area	calculated regional surface area
runtime	time for whole run simulation (seconds)

TABLE II: Performance Outputs

Feature Name	Description
pod	unique ID of a Kubernetes pod
node	unique ID of a Kubernetes node
start	datetime-stamp marking the beginning of a run
stop	datetime-stamp marking the end of a run
threads	total # of threads used
memory_requests	Min bytes of memory requested
cpu_usage	total CPU usage seconds
mem_usage	Max bytes of memory used

By accessing the BurnPro3D API, we can retrieve data about individual ensembles of simulation runs generated using the QUIC-FIRE model. This paper deals with two different classes of data relevant to BurnPro3D simulations: first are the model input data features (weather and atmospheric data, ignition and fuel conditions, geospatial data) for BP3D models; second are performance data (runtime, CPU usage, memory usage, storage I/O, network usage) generated during a Burn-Pro3D ensemble run.

Each BP3D ensemble is hosted on one or more Kubernetes nodes, where simulation runs are hosted on different pods in the node. The performance data is then stored as time series data on a Nautilus server, where it can be queried using PromQL (a querying language for Prometheus servers in Nautilus) and visualized on a Prometheus web user interface and/or Grafana dashboard.

Prior to retrieving performance data, we store the following identifiers for each BP3D run: the set of all input parameters passed into the simulation, the unique and corresponding ensemble IDs (i.e., the pod/node pair in Kubernetes), and total simulation runtime. Once these identifiers are stored, we proceed to collect the start/end timestamps and input parameter values for each run per ensemble. This data set will later be used as training data for performance optimizing AI/ML methods. Occasionally, a simulation run will fail, as indicated in the data by `NA` timestamps in the start, end, or total simulation time categories. We chose not to include failed runs in the final training data set. Table I describes input and output features collected after a simulation run of BurnPro3D, and Table II describes performance features retrieved during the run.

We can then retrieve the performance data that was collected and stored in our Nautilus server during each simulation run, and pre-process the JSON-formatted data to achieve tabular data sets suitable for basic AI/ML models. For our experiments, we are focusing on resource consumption over the duration of a BP3D simulation run. So, we query for the minimum memory requested for each pod, the total CPU usage, and total memory usage during the time range of each run. Total CPU usage and memory usage of a BP3D simulation will be the target variables that we want to predict using AI/ML modeling methods. We also collect the partial CPU and memory usage of a run from the start time until certain time periods (either predetermined or pseudo-random) to represent refreshed prediction times. This data will then be used with an AI/ML approach to predict the performance consumption metrics of the entire duration of each run.

B. Predictive Model Design

Having assembled an AI-ready data set, we now want implement a predictive model to evaluate the performance of BP3D simulations. This stage follows a standard experimental procedure of: 1) choosing a predictive modeling method, 2) data pre-processing and feature analysis, and 3) model fitting, training, and testing.

We have two target features in our data to indicate resource utilization during a given simulation run: total CPU usage (as a percentage of total CPU capacity) and total memory usage (as a percentage of total memory available). Before training the model, we calculated Pearson correlation coefficients to determine which of the variables (which consist of BP3D simulation input parameters and resource consumption data collected from a BP3D run) in our data set were most strongly indicative of changes in resource utilization. Figure 2 shows the correlation matrix for our assembled data set.

We select features that are strongly correlated (i.e., with a Pearson coefficient $> .5$) with CPU Usage and/or Memory Usage. We then proceed to the model fitting stage, which would resemble the linear regression use case described in Section III.

C. User Experience Design

We consider two potential implementations for determining how users can interact with our predictive modeling workflow. One implementation would automatically refresh model predictions at set time intervals. To do this, we would automate queries to retrieve performance data of the current simulation over set time ranges (e.g. from the start of the run until 5 minutes in), and then rerunning the predictive model, taking into account the most recent query results in order to obtain an updated prediction of resource utilization over the entire simulation run.

A second potential implementation would enable users to refresh performance predictions at any time point after the model has started running, and then continue to update the prediction as desired. There is a brief time period where the run has started, but the user is not able to request a prediction, as it takes time for data to be put onto Kubernetes servers. Our approach is to wait 45 seconds after the run starts, then automatically refresh the prediction. From then onwards, we allow the user to refresh the prediction as desired. This design approach would be similar to the previously described

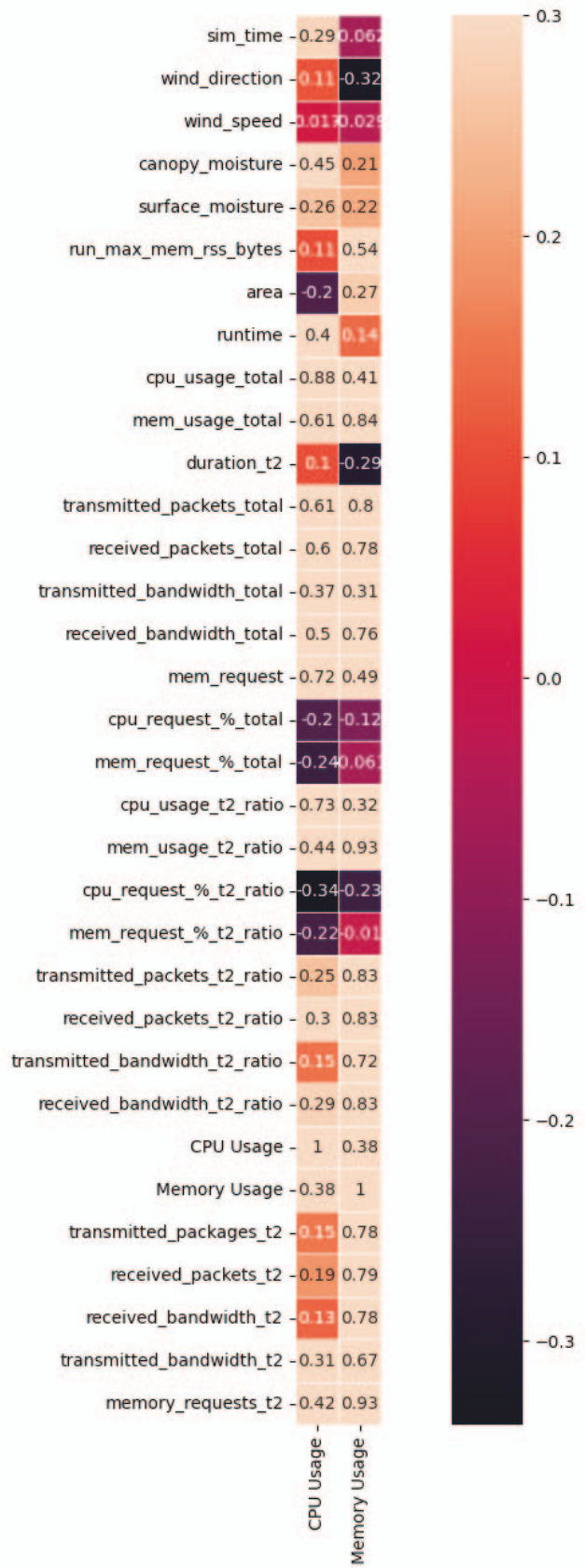

Fig. 2: Pearson correlation coefficients for CPU and Memory Usage.

implementation, except that updated performance data would only be queried when/if the user indicates, rather than at set time intervals. Using this approach, the data collected at each model refresh would consist of performance metrics from the start time of the simulation to the refresh time. The data collected from the start until the refresh time would be used to obtain an updated prediction of resource utilization over the entire simulation run. We simulate these refresh times in our training data by inserting duration columns (like duration_t2 from Figure 2) and querying from the start time until the duration time, generating data such as transmitted_packages_t2. We feed these into the model to predict the final resource consumption metrics.

The decision ultimately centers on balancing user experience and the effectiveness of model training. In the first implementation, the model is trained at consistent time intervals, which could improve its ability to recognize patterns in BP3D resource consumption, and as a result, enhance prediction accuracy. However, this implementation might negatively impact the user experience due to limited user control and potential frustration arising from the passive nature of updates. Conversely, the second approach offers the user greater flexibility, allowing them to update predictions as needed and providing a more interactive experience. While this approach might introduce complexities in model performance, we could potentially mitigate these with issues with strategic data manipulation—such as using a ratio of refresh time to resource metrics instead of analyzing these metrics separately. This ratio is shown in Figure 2 with the _t1_ratio and _t2_ratio metrics. A hybrid approach, integrating the consistent training intervals of the first method and the user-initiated updates of the second could also be considered. This would provide a more balanced approach, though its increased complexity could result in a less intuitive user experience.

V. Early Results

In the initial stages of the workflow, we retrieve resource consumption data from over 900 BP3D runs. The retrieved data will include performance statistics under four distinct categories: CPU usage, memory usage, network usage, and storage I/O. Post-processing, the tabular data will describe performance features for each BP3D run. In addition to the unique pod/node IDs for each simulation run, the final, cleaned performance database will contain the features listed in Tables III - VI.

The next stage of the workflow (AI/ML modeling), outputs predictions of total CPU usage (in seconds) and memory usage (in bytes) during a BurnPro3D run. To demonstrate possible

TABLE III: CPU Quota

Feature Name	Description
CPU Usage	total CPU usage seconds
CPU Requests	number of CPU cores requested
CPU Requests %	(CPU user time) / (total CPU time requested)
CPU Limits	maximum capacity for CPU usage
CPU Limits %	(CPU user time) / (total CPU time limit)

TABLE V: Network Usage

Feature Name	Description
Receive Bandwidth	network bandwidth for receiving bytes
Transmit Bandwidth	network bandwidth for transmitting bytes
Received Packets	# of packets received in a run
Transmitted Packets	# of packets transmitted in a run
Received Packets Dropped	# of received packets dropped in a run
Transmitted Packets Dropped	# of transmitted packets dropped in a run

TABLE VI: Current Storage IO

Feature Name	Description
IO (Reads)	# of I/Os read in a run
IO (Writes)	# of I/Os written in a run
IO (Reads+Writes)	# of I/Os read and written in a run
Throughput (Read)	bytes of throughput read during the run
Throughput (Write)	bytes of throughput written in the run
Throughput (Read+Write)	bytes of throughput read/written in a run

results at this stage, we return to the use case application of linear regression introduced in Section III. For our linear regression model, we selected training features based on the feature analysis results described in Section IV-B. Figure 3 shows the preliminary results of a linear regression model using sample performance data retrieved from BurnPro3D runs.

The linear regression model predicted CPU and memory usage with R-squared error rates of 0.70626 and 0.9221, respectively. However, given the shortage of training and testing data at this stage of our research, there are too few data points to obtain accurate or generalizable predictions through linear regression, and are currently working towards generating and preparing sufficient amounts of BP3D data in order to develop more robust AI/ML models. For the purposes of this paper, our demonstrated AI-readiness and AI/ML modeling pipelines are fundamental building blocks of an early workflow architecture.

VI. Conclusion & Future Works

We have presented our approach and early results of an integrated AI/ML workflow for performance analysis of the BurnPro3D fire management platform. Our use-case can be applied for the integration of AI-ready data preparation and AI/ML predictive modeling techniques in an end-to-end scientific workflow. where the use-case presented in this paper limits the use of ML/AI to identify relationships between BP3D input parameters and total resource consumption, our ongoing work aims to also optimize resource consumption for the purpose of mitigating uncertainty and improving accuracy of BP3D outputs. This work is part of a broader effort towards integrating AI/ML-driven methods for performance optimization in large cyberinfrastructures, namely the in-progress National Data Platform project. Immediate extensions of this work include the incorporation of a runtime prediction modeling stage into the current workflow, and the introduction of end-to-end uncertainty quantification metrics in order to align our work with FAIR data management standards for scientific research.

Acknowledgments

The authors would like to thank the WIFIRE and WorDS teams for their collaboration and support of this study by NSF 2040676, 2134904 and 2333609, and the Nautilus Kubernetes Cluster of the National Research Platform funded by in part by NSF awards 1730158, 1540112, 1541349, 1826967, 2112167 and 2120019.

TABLE IV: Memory Quota

Feature Name	Description
Memory Usage	number of bytes of memory usage
Memory Requests	minimum number of bytes requested
Memory Requests %	(memory usage) / (memory requests)
Memory Limits	maximum capacity for memory usage
Memory Limits %	(memory usage) / (memory limits)
Memory Usage (RSS)	RSS bytes used
Memory Usage (Cache)	CPU cache memory used

References

[1] M. Gollner, A. Trouve, I. Altintas, J. Block, R. de Callafon, C. Clements, A. Cortes, E. Ellicott, J. B. Filippi, M. Finney *et al.*, "Towards data-driven operational wildfire spread modeling: A report of the nsf-funded wifire workshop," Tech. Rep., 2015.

[2] T. Dittmann, K. Hodgkinson, J. Morton, D. Mencin, and G. S. Mattioli, "Comparing sensitivities of geodetic processing methods for rapid earthquake magnitude estimation," *Seismological Society of America*, vol. 93, no. 3, pp. 1497–1509, 2022.

[3] B. Corsa, M. Barba-Sevilla, K. Tiampo, and C. Meertens, "Integration of dinsar time series and gnss data for continuous volcanic deformation monitoring and eruption early warning applications," *Remote Sensing*, vol. 14, no. 3, p. 784, 2022.

[4] D. Balouek-Thomert, E. G. Renart, A. R. Zamani, A. Simonet, and M. Parashar, "Towards a computing continuum: Enabling edge-to-cloud integration for data-driven workflows," *The International Journal of High Performance Computing Applications*, vol. 33, no. 6, pp. 1159–1174, 2019.

[5] D. Balouek-Thomert, I. Rodero, and M. Parashar, "Harnessing the computing continuum for urgent science," *ACM SIGMETRICS Performance Evaluation Review*, vol. 48, no. 2, pp. 41–46, 2020.

[6] I. Altintas, J. Block, R. De Callafon, D. Crawl, C. Cowart, A. Gupta, M. Nguyen, H.-W. Braun, J. Schulze, M. Gollner *et al.*, "Towards an integrated cyberinfrastructure for scalable data-driven monitoring, dynamic prediction and resilience of wildfires," *Procedia Computer Science*, vol. 51, pp. 1633–1642, 2015.

[7] D. Balouek-Thomert, I. Perez, S. D. Faulstich, H. A. Holmes, I. Altintas, and M. Parashar, "Keynote talk: Leveraging the edge-cloud continuum to manage the impact of wildfires on air quality," in *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 2023, pp. 27–31.

[8] M. Parashar, A. Simonet, I. Rodero, F. Ghahramani, G. Agnew, R. Jantz, and V. Honavar, "The virtual data collaboratory: A regional cyberinfrastructure for collaborative data-driven research," *Computing in Science & Engineering*, vol. 22, no. 3, pp. 79–92, 2019.

[9] "BurnPro3D: A Platform for Prescribed Fire Planning and Optimization," 2023. [Online]. Available: https://burnpro3d.sdsc.edu/index.html

[10] D. Roten, L. Wells, D. Crawl, R. A. Parsons, A. Marcozzi, R. R. Linn, K. Hiers, and I. Altintas, "Truetrees: A scalable workflow for the integration of airborne lidar scanning data into fuel models for prescribed fire simulations," in *2023 IEEE 19th International Conference on e-Science (e-Science)*. IEEE, 2023, pp. 1–10.

[11] P. Beckman, J. Dongarra, N. Ferrier, G. Fox, T. Moore, D. Reed, and M. Beck, *Harnessing the Computing Continuum for Programming Our World*. John Wiley Sons, Ltd, 2020, ch. 7, pp. 215–230. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119551713.ch7

[12] S. Purawat, S. Dasgupta, L. Burbidge, J. L. Zuo, S. D. Wilson, A. Gupta, and I. Altintas, "Quantum data hub: A collaborative data and analysis platform for quantum material science," in *International Conference on Computational Science*. Springer, 2021, pp. 656–670.

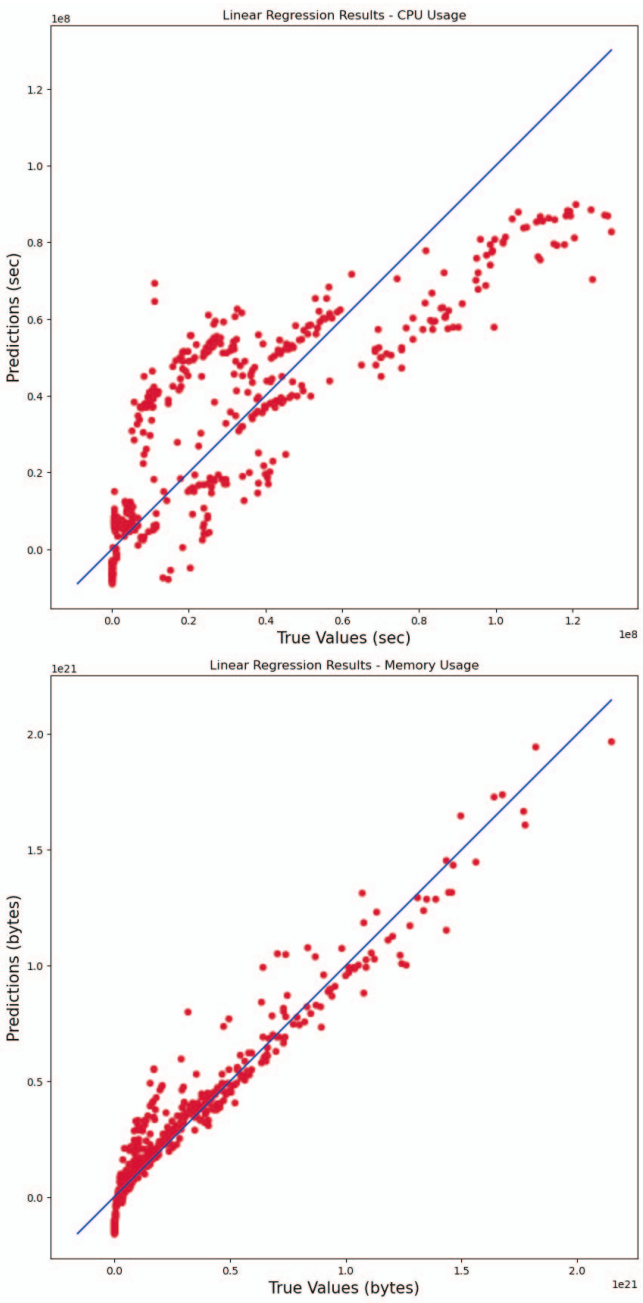

Fig. 3: Preliminary results of linear regression use case (Sec III).

[13] S. Dasgupta, K. Coakley, and A. Gupta, "Analytics-driven data ingestion and derivation in the awesome polystore," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 2555–2564.

[14] M. H. Nguyen, D. Crawl, T. Masoumi, and I. Altintas, "Integrated machine learning in the kepler scientific workflow system," *Procedia Computer Science*, vol. 80, pp. 2443–2448, 2016.

[15] C. Baru, M. Pozmantier, I. Altintas, S. Baek, J. Cohen, L. Condon, G. Fanti, R. Fernandez, E. Jackson, U. Lall *et al.*, "Enabling ai innovation via data and model sharing: An overview of the nsf convergence accelerator track d," *AI magazine*, vol. 43, no. 1, pp. 93–104, 2022.

[16] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne *et al.*, "The fair guiding principles for scientific data management and stewardship," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.