Cloud and HPC Ecosystems for Scientific Experiments

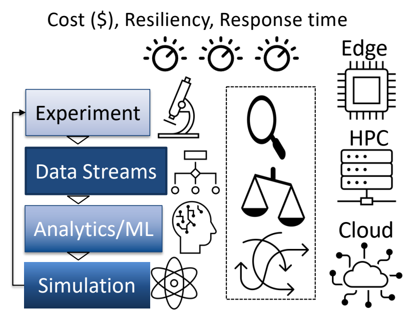
Nathan Tallent (tallent@pnnl.gov), Steven Spurgeon, Pacific Northwest National Lab

February 17, 2023

Focal Area

(Area 1) Knowledge gaps in methodology and infrastructure *for workflow execution*
(Area 2) Challenges implementing AI approaches *for automating feedback to scientists or instruments*

Challenge



Future scientific discovery requires automating data-driven feedback to scientists or instruments to handle the full array of data generated by modern hardware, rapidly make decisions, and extrapolate beyond limits of any one experimental data set. Examples range from automating analysis from simulations, sensors, and AI-driven models to forming real-time loops that can guide instruments or automate experiments, such as high-resolution analysis of material and chemical systems [1,4]. To achieve new levels of automation with machine reasoning, we must harness distributed *scientific workflows* that can exploit continuum computing ecosystems to meet both cost budgets and quality of service, i.e., response-time and resiliency.

A related need is unsupervised learning, required because of the infeasibility of labeling, which usually requires massively distributed training and substantial computational resources. The task of scientific discovery is often ill-suited to transfer learning approaches, which may lack generalizability to accurately describe or assess new experimental features.

We envision coordinated teams of domain scientists and computer scientists that design workflows to meet budget and quality of service requirements. Domain scientists would establish context by defining the domain challenges that represent fundamental limitations imposed by current computing solutions. Computer scientists leverage frameworks for co-design of cost, response-time and resiliency to guide workflow executions that combine multi-system resources, especially near-instrument computing, facility HPC, and cloud. Cloud can complement DOE computing beyond on-demand scale-out because it now drives computing trends by showcasing novel systems (quantum), new platforms (TPUs), system virtualization (containers, serverless), and ML frameworks (PyTorch, TensorFlow).

Rationale

Today's tools naively execute workflows on multiple systems. Customizing data movement and resiliency is critical to meet time constraints but must be done manually—a cumbersome and error-prone process. Cloud's Function-as-a-service (FaaS) model is attractive for cost and availability but has not been designed for meeting workflow time constraints.

Meeting cost, response-time, and resiliency raises fundamental research challenges. *Costs* in cloud vary widely based on service (static instance vs. stateless container), hardware, and availability-resiliency guarantees. The *execution time* of workflow tasks not only depends on partitioning and assignment, but on prioritizing task vs. data movement, data layouts, data movement schedules, and data caching and consistency policies. Task *resiliency* is usually inversely related to time. Even worse, it is usually implicit and fixed, and within workflows result in redundant recovery efforts. Further, workflows can change resiliency semantics, e.g., "final answer" tasks still need checkpointing, but "exploratory tasks" can use best-effort. Finally, workflows that vary with input and time require dynamically adaptive policies.

Vision. There is a critical need for automated co-design techniques that can address the following questions: Given a target budget and quality of service, what is the best selection of resources across facility, cloud, and edge (near-instrument) resources to create a virtual platform? What is the best assignment of policies for task placement, data movement, and task resiliency? Given a set of fixed resources and optional target budget, what is the range of pareto-optimal execution policies and their corresponding tradeoffs?

Narrative

Our approach is to develop transferable co-design techniques and tools within the five thrust areas below. We target workflows for rapid scientific exploration; most are I/O intensive and coordinated with workflow managers. The co-design framework reasons about the cost-time tradeoff space for policies that meet given constraints.

Workflow-guided characterization of performance and resiliency to develop models that drive co-design. To reason about co-design tradeoffs, we develop workflow-specific models of data lifecycles and resiliency relative to key workflow parameters. We use distributed and scalable workflow introspection within I/O middleware to capture data lifecycles between workflow tasks [2, 3, 5].

Coordination and resource partitioning for Cloud and HPC Ecosystems. To meet cost and quality of service constraints in Cloud and HPC Ecosystems, we will efficiently map tasks to execution policies and multi-system resources [5]. The scheduler compares predictions of task performance with execution dynamics and if necessary, adopts recommended alternatives.

Optimizing I/O middleware using customized performance and resiliency policies. To avoid I/O bottlenecks and improve data velocity, we ensure careful task and data placement and explore customized I/O middleware policies.

Optimizing emerging cloud execution models. We propose retaining the attractive properties of FaaS execution but avoiding its overheads. We explore customized workflow performance and resiliency configurations that avoid this overhead.

[1] S. Akers, E. Kautz, A. Trevino-Gavito, M. Olszta, B. E. Matthews, L. Wang, Y. Du, and S. R. Spurgeon. Rapid and flexible segmentation of electron microscopy data using few-shot machine learning. *NPJ Computational Materials*, 7(1):187, 2021.

[2] R. D. Friese, B. O. Mutlu, N. R. Tallent, J. Suetterlein, and J. Strube. Effectively using remote I/O for work composition in distributed workflows. In *Proc. of the 2020 IEEE Intl. Conf. on Big Data*. IEEE Computer Society, December 2020.

[3] O. O. Kilic, N. R. Tallent, Y. Suriyakumar, C. Xie, A. Marquez, and S. Eranian. MemGaze: Rapid and effective load-level memory and data analysis. In *Proc. of the 2022 IEEE Conf. on Cluster Computing*. IEEE, Sep 2022.

[4] M. Olszta, D. Hopkins, K. R. Fiedler, M. Oostrom, S. Akers, and S. R. Spurgeon. An Automated Scanning Transmission Electron Microscope Guided by Sparse Data Analytics. *Microscopy and Microanalysis*, 28(5):1611–1621, 9 2022.

[5] J. Suetterlein, R. D. Friese, N. R. Tallent, and M. Schram. TAZeR: Hiding the cost of remote I/O in distributed scientific workflows. In *Proc. of the 2019 IEEE Intl. Conf. on Big Data*, pages 383–394. IEEE, Dec. 2019.