Workflow Archetypes White Paper

Version 1.0

Deborah Bard, Taylor Groves, Brandon Cook, Laurie Stephey, Wahid Bhimji,
Steve Farrell, Brian Austin, Kevin Gott, Shane Canon, Kristy Kallback-Rose,
Jay Srinivasan, Hai Ah Nam, Nicholas J. Wright

National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory
Berkeley, CA 94720

Report No. XXXX

May 2023

This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Overview

The NERSC workload is increasingly diverse, with a growing demand for complex high performance workflow capabilities from our user community across all scientific domains. **High performance computing (HPC) workflows** interconnect computational and data manipulation steps, and require significant computational, storage and/or network resources within. The NERSC-10 system will integrate available new technologies and support the emerging needs in AI and experimental/observational science **to accelerate end-to-end DOE SC workflows and enable new modes of scientific discovery through the integration of experiment, data analysis, and simulation**.

This whitepaper describes six archetypal HPC workflows that we expect to support with NERSC-10, drawn from real workflows we anticipate on our systems. **It is important to stress that the NERSC workload is very diverse, with hundreds of such workflows running on the system simultaneously.** Some workflow steps run at close to full machine scale, and some have large data requirements. Collectively, they will require close coupling between NERSC-10 itself, the NERSC data center and beyond. Some will be as complex as the workflows described here, and some will be much simpler. The archetypal HPC workflows are:

1. **High-performance simulation & modeling workflow** (e.g., large-scale multi-physics applications with checkpoint/restart, data post-processing, and visualization) [1-3]
2. **High-performance AI (HPAI) workflow** (e.g., data integration-intensive science patterns such as training, inference, hyperparameter optimization) [4-8]
3. **Cross-facility workflow: Rapid data analysis and real time steering** (e.g. time-sensitive science patterns such as superfacility, edge, and hybrid cloud)
4. **Hybrid HPC-HPAI-HPDA workflow** (e.g. long-term campaign science patterns, AI-in-the-loop, AI-around-the-loop)
5. **Scientific data lifecycle workflow: Interactive, data-analytics and visualization** (e.g. data integration-intensive science patterns such as Jupyter, scientific databases, VSCode)
6. **External event-triggered and API-driven workflow** (e.g. time-sensitive science patterns such as function-as-a-service, microservices)

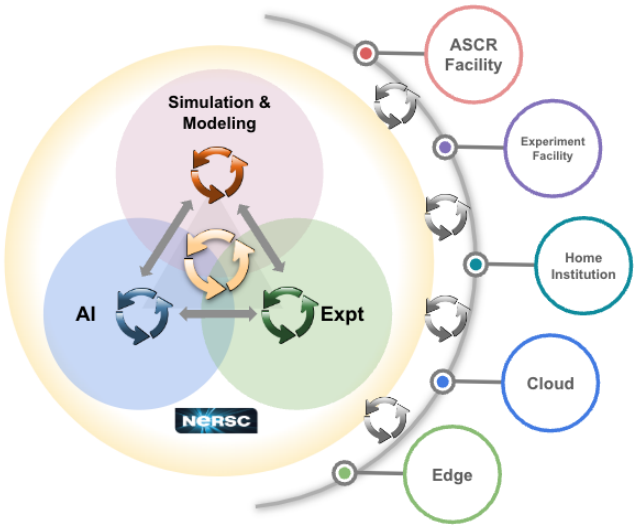
NERSC has long-supported and will continue to advance HPC science campaigns through core workflows that couple computational tasks and data flow (Workflows 1, 2, 3).

Although experiment to high-performance data analysis (HPDA) [9-12] has been a long-time NERSC workflow, in the NERSC-10 timeframe, we expect to see more HPC workflows cross the data center boundary to external resources for new capabilities, workflow resilience, and data access, which requires workflow orchestration and tools for profiling and debugging (Workflow 3, 4, 5, 6).

Emerging HPC workflows will need to seamlessly couple and automate the computational tasks and data flow between or across all three core workflow modes (simulation & modeling, AI, experiment/data analysis) to unlock opportunities for new scientific discovery without creating excessive burden on developers (Workflow 4).

Users across the Office of Science (SC) are seeking greater capabilities to search, analyze, reuse, and combine data from different sources. Data should be leveraged beyond a single experiment and easily re-analyzed, shared, and archived with simple and powerful user-driven data lifecycle management capabilities with Findable. Accessible. Interoperable. & Reusable (FAIR) practices (all). [13]

Integrated science workflows



The conventional approach to accessing and interacting with HPC system resources will need to evolve to enable seamless integration with the larger integrated research infrastructure [14] (Workflow 5, 6).

NERSC is interested in technologies that can help support this vision. The needed resources stated in these examples, based on projected science needs, are given to illustrate expected capabilities rather than precise sizing requirements.

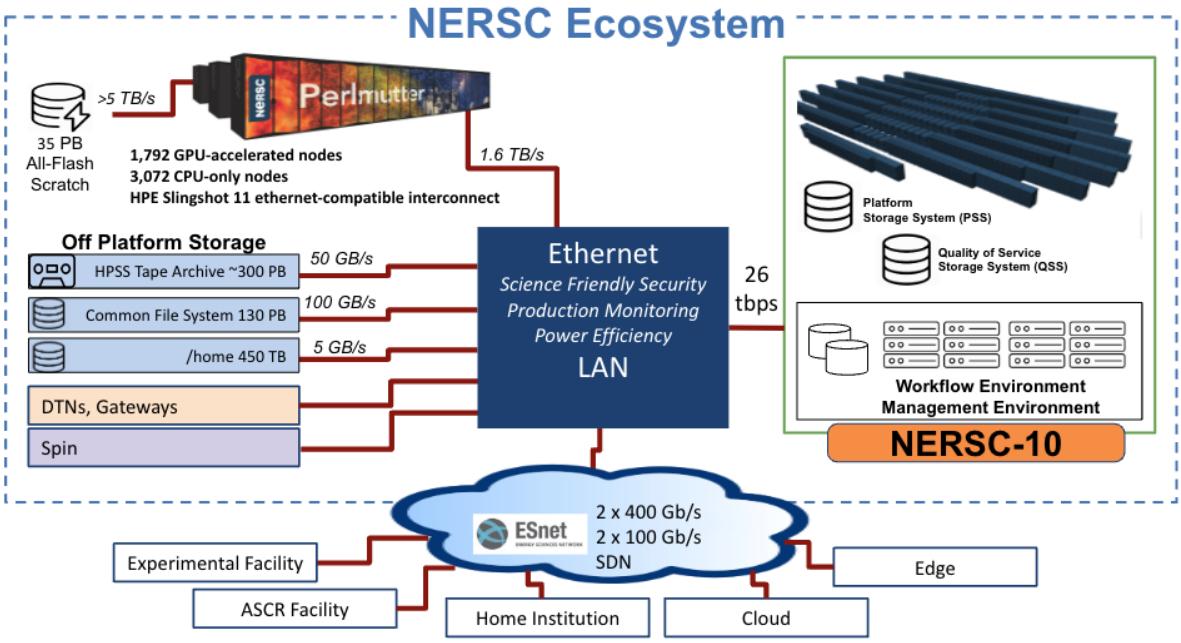
Enabling these complex and novel workflows requires the NERSC-10 system to provide:

- Quality of Service – computation, storage and networking designed to emphasize response-time in addition to system throughput/utilization.
- Seamlessness – tight integration of system components to enable high performance across the workflow steps.
- Portability – Modular workflow execution across heterogeneous HPC, edge and cloud.
- Programmability – APIs to manage data, execute distributed code, and interact with system resources.
- Orchestration – coordinate resource management using different types of scheduling functionality across different resource domains.
- Security, authentication and provenance – control access and generate records of transactions.

The NERSC-10 workflow component benchmarks are designed to demonstrate the capabilities of the platform for the most computationally intensive components of the workflows they represent, but without the data-flow and control-flow complexities of an integrated scientific workflow.

An Evolving HPC Workflow Environment

The NERSC-10 system will be the cornerstone of a dynamic high performance workflow environment at NERSC, to provide new capabilities to the DOE Office of Science community of researchers.



NERSC-10 will have a pool of workflow environment nodes (WENs) to enable an array of interactive user workflow capabilities, including command-line interface (CLI) through ssh and web-based user access modes for:

- login, code compilation, application development, container builds/management, job lifecycle management, small-scale data analysis, and data management.
- long-lived user services and frameworks (e.g., JupyterHub, databases, API services, and message brokers), that are staff-managed or user self-supported.

The WENs will provide a robust and feature-rich environment to leverage the power within the NERSC-10 compute and storage resources and provide flexibility to access capabilities outside of the NERSC Data Center as workflows evolve in the NERSC-10 timeframe. WENs will support a container orchestration platform alongside the HPC resource scheduler, Slurm, from which users can securely and performantly launch job tasks on the compute resources or services running on WENs. Workflows are demanding advanced scheduling capabilities, including real-time scheduling, co-scheduling/multi-tenancy and dynamic resource allocation. The WENs will include hardware and software to enable programmability, composability, and traceability needed by workflows.

Workflow Descriptions

1. High-performance simulation & modeling workflow

Science workflows, whether large-scale simulations or data-intensive simulations (see [APEX Workflows Whitepaper]), are core workflows that form the foundation of more complex workflows. The workflows can have many phases - setup, simulation, analysis, post-processing, visualization, archival, etc. - which have different computational, storage, network and orchestration requirements.

Workflow scenario (from the APEX Workflows Whitepaper):

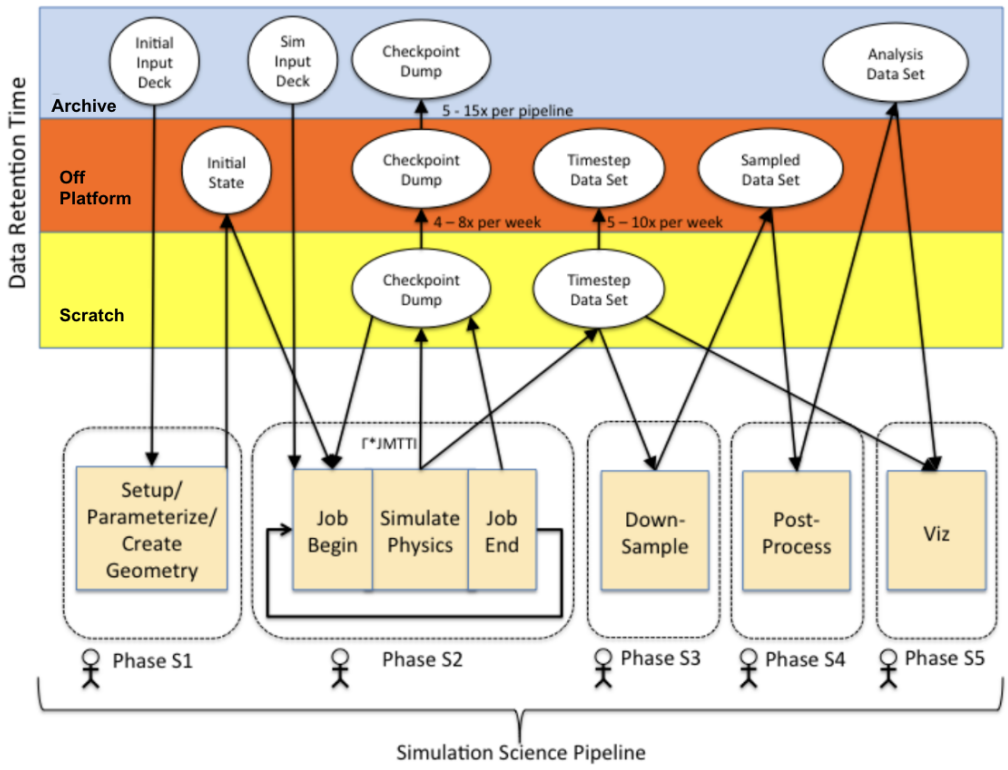


Figure 1: An example of an APEX simulation science workflow.

Simulation and modeling workflows are increasing in complexity as methodologies are evolving to solve higher fidelity or larger scale problems. In modern, dynamically changing scientific simulations, each computational phase may achieve best performance or efficiency on different combinations of processing elements. Some phases may scale up to the full machine and require maximum throughput while some phases may be more limited in scale. Phases themselves may include multiple types of simulations with different resource requirements and I/O patterns. The composition of phases will also be complex as scientists incorporate interactive methods of simulation steering or in-situ visualization to their workflows.

2. High-performance AI workflow

Scientific machine learning (SciML) is a rapidly evolving workload, utilizing a wide variety of models and approaches to parallelism (e.g. deep learning can employ data parallelism, tensor parallelism, layer pipeline parallelism). Effective ML involves a complex workflow that extends beyond simply training a model, including dataset management (storage and curation of data, transfer of datasets, version control), experiment management (reproducibility of training runs, hyperparameter optimization, I/O of datasets and model), and model management (deploying and monitoring of model, scaling, model drift, triggers to restart ML workflow).

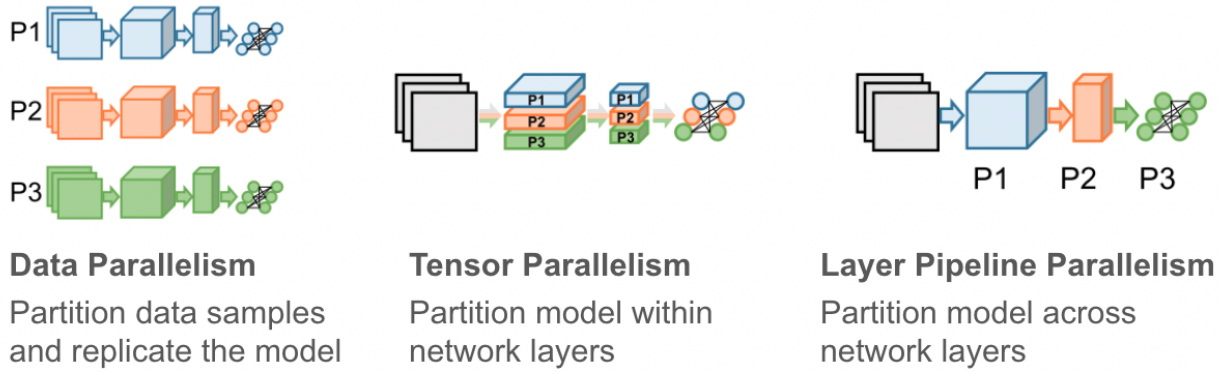


Image from Ben-Nun and Hoefler [15]

Workflow scenario:

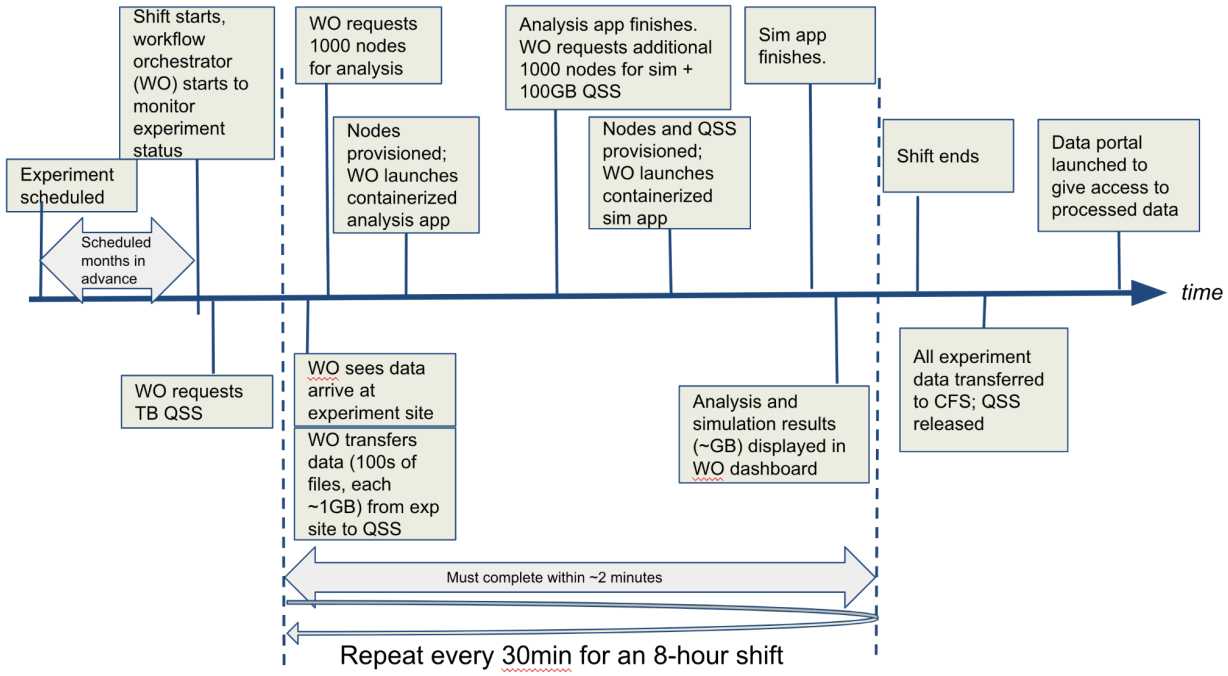
- Training a large AI model (e.g., a deep neural network) on big scientific data
- The datasets may come from HPC simulations, experimental science facilities, or other large scientific databases (e.g., materials databases on MongoDB)
- The deep neural network model may be trained across hundreds to thousands of accelerators utilizing various forms of data and model parallelization, depending on the resource requirements of the model, the data, and the optimization algorithm.
- These workloads are frequently containerized for portability and because vendors will use containers to provide a full cutting edge software stack optimized for their hardware.
- Data will be streamed from a file system designed for fast IOPs (QSS) to provide performance for the small random read patterns of I/O in AI model training.
- The final trained model will need to be saved to storage in off-platform storage for later fine-tuning, or deploying on new data.
- The model may be shared with other science groups, possibly at other facilities.

Accelerating this workflow could include use of specialized AI hardware (at NERSC and/or accessed remotely) for both training and inference.

3. Cross-facility workflow: Rapid data analysis and real time steering

Across the SC research enterprise, an increasing number of projects require the ability to rapidly analyze large amounts of data and integrate data from experiments with both simulations and AI, with the intention of providing research teams with interactive, real-time feedback to steer experiments. Experimental facility users wish to optimize their short data collection window (e.g. microscopy, light sources) or must respond quickly to some unexpected event (e.g. telescopes).

Workflow scenario: An experiment external to NERSC runs and collects data during scheduled shifts (~8 hours). In the 8 hour window, hundreds of analysis iterations occur. Each iteration must complete in two minutes and consists of (1) GB size data transfers from the experiment to QSS via a high-performance data transfer tool such as Globus, (2) analysis of the data, and (3) launch simulation and modeling jobs across O(1000) GPUs, based on the results of the data analysis. This workflow is controlled by a user-provided workflow orchestrator using APIs that must be able to see the data arriving, update log databases, and monitor and control the status of the workflow.



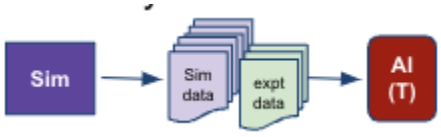
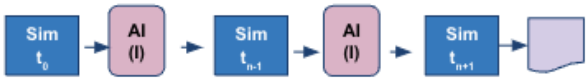
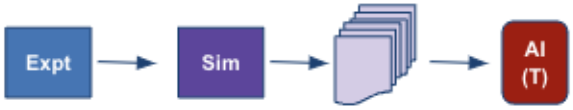
- Scientific instruments write large amounts of data to NERSC for time-sensitive analysis. Data is transferred to QSS, with a backup copy at the experimental facility.
    - O(TB) in total transferred during the 8-hour shift.
    - QSS provides deterministic performance to the application.

- User is running a user-defined workflow orchestrator on the WEN that uses NERSC-10 APIs and scheduling resources to track data arrival and launches and monitors the compute jobs.
- User-requested network QOS gives a guaranteed response time between compute and storage.
- NERSC-10 external connectivity network bandwidth provides 1 Tb/s.
- Accelerated nodes used for analysis and simulation
    - O(4000) accelerators for analysis and simulation.
    - User uses a containerized application which has been developed/optimized on N10; the containerized application has access to interconnect/optimized MPI/QSS/accelerator functions.
- The analysis application accesses small reference databases that sit inside and outside of NERSC (for example, experiment configuration databases hosted at the experiment site). The databases at NERSC sit on the WENs.
- Analysis results transferred from QSS to off-platform storage at the end of the experiment shift. O(GB)

Resilience is a challenge for experiment sciences. A single system cannot guarantee 24/7 uptime due to planned and unplanned outages. A truly resilient workflow needs to span multiple computing centers. A seamlessly resilient workflow should be able to query system availability and run its containerized application on NERSC-10 or burst to external cloud resources or other HPC facilities. This requires designing workflows to be portable, including the software environment, orchestration across sites, transferable storage/data access, and other technologies to enable workflows to easily switch to a new site when one is unavailable without disrupting the integrity of experiment data. NERSC-10 will need to support commonly used container technologies and facilitate portable workflows. Users also require resilient workflows that are able to adapt to situations where a subset of NERSC resources are available (e.g. WENs are available when the compute partition is down).

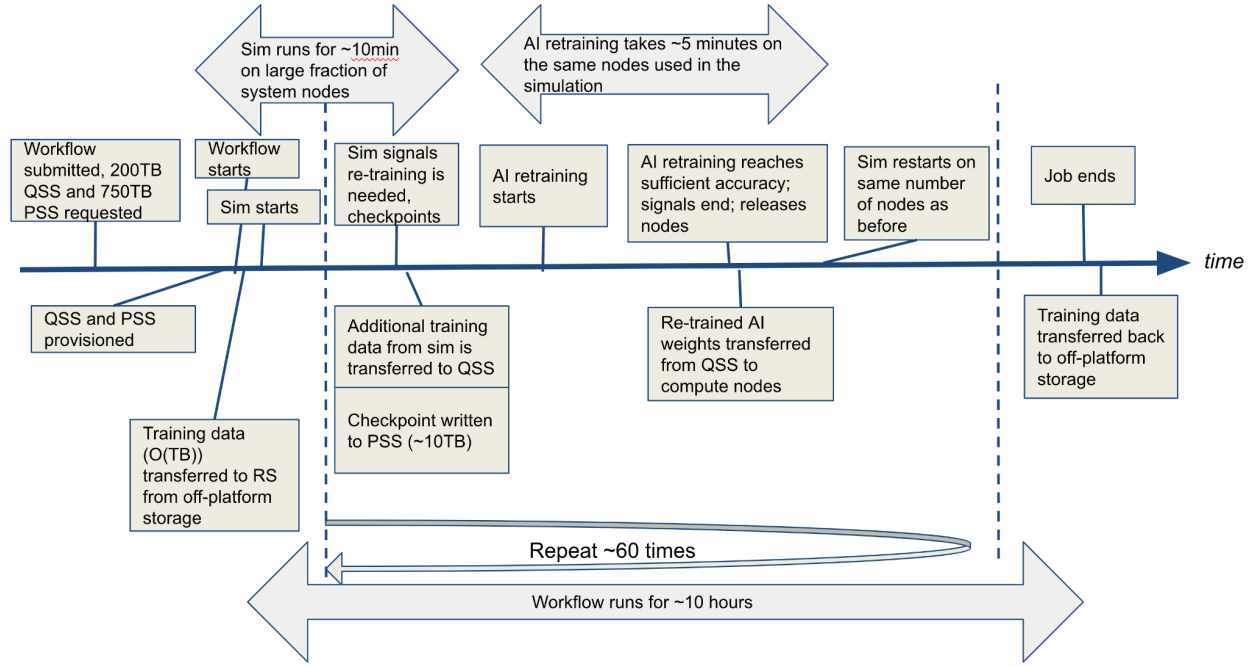
4. Hybrid HPC-HPAI-HPDA workflow

There is an ever growing demand to integrate data across HPC simulations, high-performance AI (HPAI) and high-performance data analytics (HPDA) to create new capabilities and insight. There is also a growing number of ways to create new hybrid workflows, for example:

<table>
<tr>
<td rowspan="3">Simulations generate synthetic data, coupled with experimental data to train the AI model
 
</td>
<td>The AI model is deployed to replace the simulation to provide faster predictive capabilities
 
</td>
</tr>
<tr>
<td>The AI model is deployed to provide initial conditions for future simulations
 
</td>
</tr>
<tr>
<td>Iterations are alternating between the AI model and simulation (coarse/fine grain)
 
</td>
</tr>
<tr>
<td>Experiments spawn a set of simulations across parameters. Simulation data is used to train the AI model.
 
</td>
<td>The AI model is deployed at the experimental facility to guide experiments and inform system controllers
 
</td>
</tr>
</table>

Workflows must be able to seamlessly ingest data from multiple sources and couple simulation and AI tasks with different compute, storage, and network resource requirements.

Workflow scenario: A large-scale running simulation with AI-in-the-loop automatically identifies that an AI inference model used within the simulation needs updating. The simulation checkpoints, and spawns a suite of AI training jobs with different parameters, informed by the simulation, which takes roughly 5 minutes at full-system scale. The simulation then restarts, updating the inference model with parameters from the updated AI model. After 10 hours the workflow ends and all data is transferred to NERSC center off-platform storage for later analysis.



- Both simulation runs with AI-in-the-loop and AI training runs make use of accelerators but sized for the resource needed
    - Up to full system scale for the simulation and full system scale for the AI training.
    - Require O(PB) memory to store AI model and weights
- Data used to train the AI model is transferred to QSS from off-platform storage. The AI training data is supplemented during the workflow with data drawn from the running simulation.
    - O(TB) starting training data
    - O(TB) supplementary simulation data written to QSS every 10 min, total of ~100 TB.
    - Filesystem suited for many random data accesses typical in AI workloads
    - Updated AI weights transferred to off-platform storage at end of workflow for later analysis.
- Checkpoint files and simulation output written to PSS at checkpoint marks
    - O(TB) written every 10 min, total of ~750 TB.
    - Simulation output and checkpoint files transferred to off-platform storage at end of workflow for later analysis.

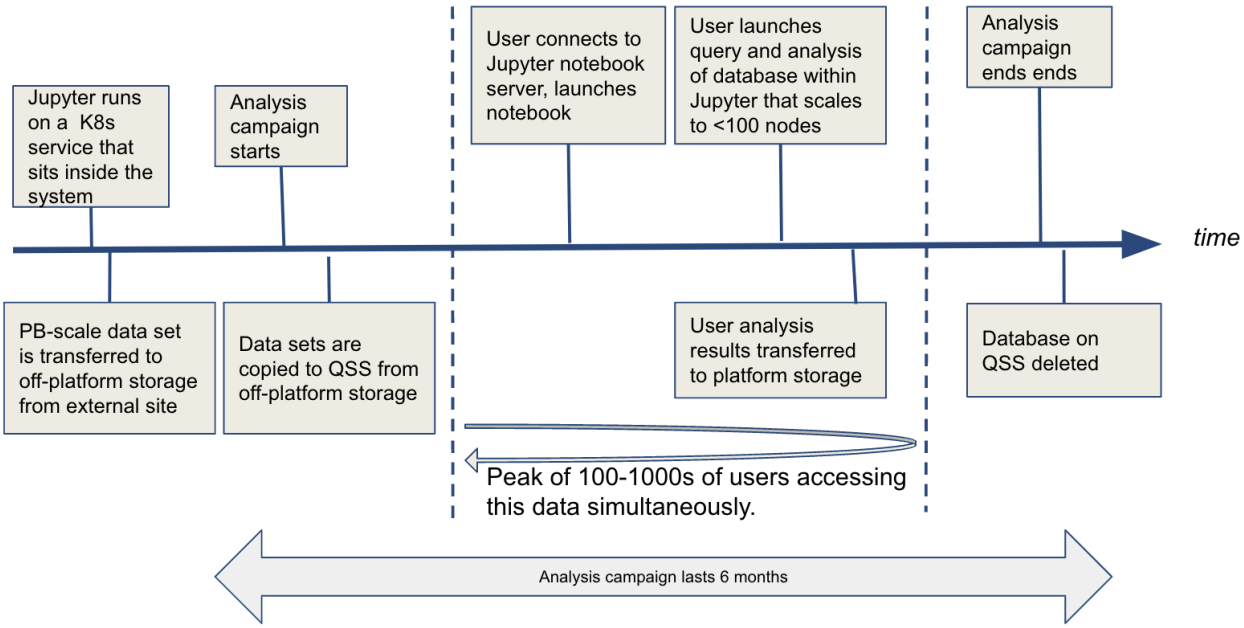
5. Scientific data lifecycle workflow: Interactive data-analytics and visualization

Users across the Office of Science are seeking greater capabilities to search, analyze, reuse, and combine data from different sources. NERSC users store hundreds of petabytes (PBs) of data in NERSC's archival storage and community file system, both data generated by NERSC simulations and data transferred via ESnet from experimental facilities. Users will need ready access to this data from the NERSC-10 system in order to reanalyze it, manipulate it and conduct virtual experiments.

The access patterns for a given set of data is also changeable. Data may be temporary or may progress through several stages in a workflow: import from an external source to be preprocessed and then used repeatedly by user analysis, and finally archived and/or served to the wider community via web or other interfaces. API interfaces to support the movement of data into and between all tiers of NERSC storage are required, and support for transparent, programmable internal data movement is desired. Science teams that move data between multiple locations (e.g. experiment on-site computing, external data store, other HPC centers) will also benefit from a standard interface to storage across sites.

As we deal with complex scientific problems that cross traditional discipline boundaries, it is becoming increasingly important to organize and expand access to the wider community along FAIR (Findable Accessible Interoperable Reusable) principles. In the NERSC-10 timeframe, observational data of the earth, environment and climate is expected to grow dramatically. In addition, data from experiments such as light sources, accelerators, and sequencers will characterize the properties of atmospheric particles, geochemical processes and genomic data, providing additional insights and data sources. The growth in the volume and complexity of data is already challenging scientists' ability to synthesize, model, integrate and understand the data. NERSC needs storage systems that can support the tracking of metadata and dataset versions and enable frequent reprocessing of the data.

Workflow scenario: The NERSC-10 system must be able to support thousands of simultaneous users using Jupyter notebooks for web-based interactive computing, including data integration-intensive patterns. A long-term data intensive campaign can include hundreds of users simultaneously using Jupyter notebooks to run data analytics workflows against a shared data set. Each user notebook analysis can scale from a fraction of a node up to 100s of nodes, accessing PB of data, and include functions that require both CPU and accelerator nodes. The Jupyter servers are deployed through Kubernetes, which allows users to deploy a variety of workflow services through containers. These analytics workflows will access PB-scale data sets stored in files on QSS, PSS, off-platform storage, or in scalable multi-PB databases orchestrated via K8s.



- Processors:
  - Notebooks provisioned on demand by K8s cluster running on CPU and accelerator nodes.
  - Notebooks can share nodes and accelerators.
  - Users may interactively request 100s of nodes for running analytics jobs at scale.
- Storage:
  - Notebooks access data in files stored on PSS and QSS.
  - Multiple PB-scale datasets will be hosted on QSS, and accessed by multiple users simultaneously via a K8s-based framework accessed via Jupyter notebooks.
  - Data stays on PSS for reanalysis and is transferred to off platform storage for sharing and HPSS for archiving on a regular cadence
- Network:
  - Notebooks will establish thousands of connections to persistent services and databases across the NERSC data center.
- Workflow Environment:
  - Hundreds of users running interactively through interfaces like Jupyter or VSCode servers.
  - Small databases hosted on WENs accessed by multiple users
  - Dynamic scaling of resources on a daily or weekly schedule: More interactive use during working hours, more batch processing outside working hours and on weekends.

Currently hundreds of NERSC users use Jupyter to perform interactive development and data analysis tasks on shared nodes or up to a few dedicated compute nodes with limited data access. The demand for more interactive access that can scale flexibly and provide simple and powerful data management capabilities is growing to thousands of users. Co-scheduling or

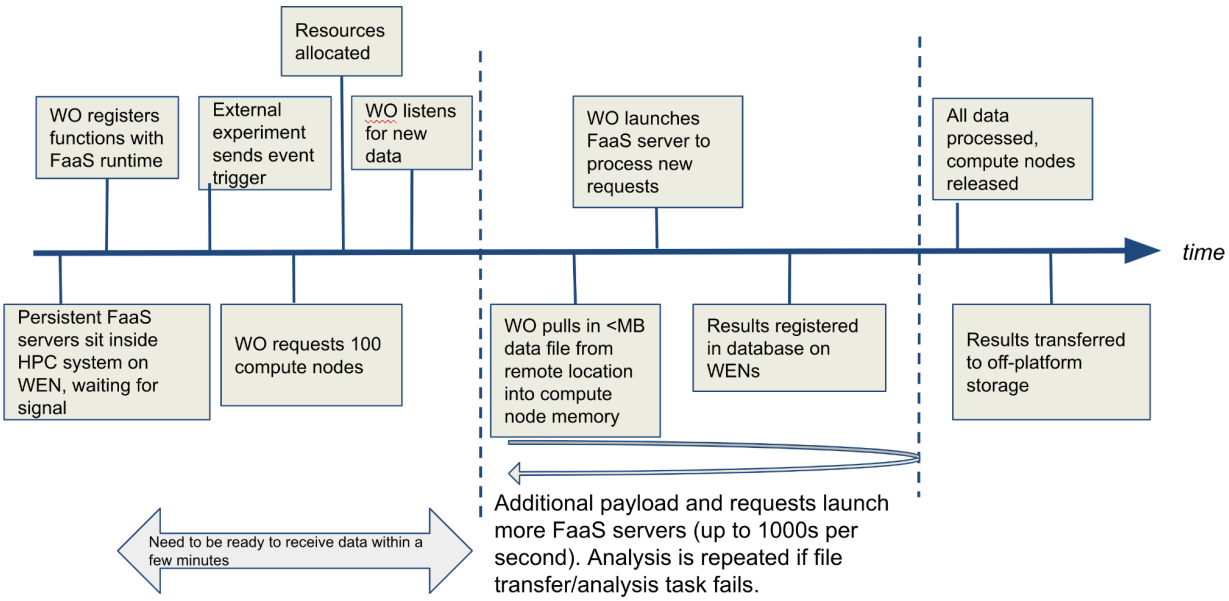
multi-tenancy on-node can maximize system utilization for less intense workflows that can share resources, with isolation of resources at a more granular level. These interactive workflows can also grow to include large multi-node and multi-phase steps with different resource requirements for each phase, requiring technologies that can repartition and reprovision with agility.

6. External event-triggered and API driven workflow

Hybrid HPC and cloud execution model workflows will improve the automation and resilience of workflows. This API driven approach provides predictable execution of scientific applications flexibly across multiple sites to enable workflows to flexibly scale, maximize resource utilization and provide workflow resilience.

Workflows may start and/or end outside of the NERSC Data Center and therefore require programmatic interfaces for effective use. This extends to all features of NERSC-10 including but not limited to scheduling, availability, data movement, service orchestration, network connectivity. This can only be achieved with programmatic interfaces to all features of the NERSC-10 system (e.g. via an API). A machine-to-machine interface is needed to submit work to the NERSC-10 scheduler via an API, with no human in the loop.

Workflow scenario: An event at an external experiment triggers the need for urgent analysis at NERSC. The number of triggers per day will vary based on the experiment needs. With each event data must be transferred to NERSC and entered into a dynamic event processing pipeline that involves multiple steps with different resource requirements and complex control flow topology. An API gateway accepts the external requests and securely routes them to pre-registered functions hosted in a Function-as-a-Service (Faas) runtime. Incoming requests may contain small MB sized payloads and larger data transfers may be orchestrated via handshake protocols. These functions then call further functions, make API calls, interact with other persistent or ephemeral services such as message queues and databases, and otherwise programmatically access other NERSC capabilities in order to process the events. For example a function may prepare and submit a large scale simulation job to the batch queue and another function may process the notification of the job completion and stage further processing. The resources utilized by the workflow and the FaaS service vary dynamically in time with the number and type of events, scaling up to potentially thousands of function invocations per second. These workflows also need the hosting infrastructure to expose status of systems and tasks, storage utilization and network QoS.



- Processors:
  - Each analysis may ramp up to 100s of nodes (CPU and accelerator, may be WEN or compute nodes)
- Storage:
  - Requests spawn queries and analysis of data coming in to the compute node
  - WEN used to host a small database that records results of data analysis.
    - O(GB) output data transferred to off-platform storage at the end of workflow.
- Network:
  - Thousands of small to MB-size individual payloads.
  - FaaS uses a Private Overlay network between nodes and services.
  - Tunnel or other secure mechanism to reach services externally.
  - Time-sensitive analysis requires a user-requested QOS.
  - Network dynamically provisioned to scale up/down with number of transfers.
  - Possible compression and other data manipulation, duplication, etc in-network.
- Workflow Environment:
  - External requests and payload delivered to N10 FaaS runtime.
  - User is running a monitoring service that sits at the experiment site, which needs a view into NERSC system status, job status, storage and network QoS.
  - FaaS services use API to request storage and computational resources.
  - Dynamically tears down and brings up services and network connections.
  - Databases to track progress and store results.
  - Afterwards, processed data on disk is shared publicly via a scalable data portal.

References

[1] Brian Austin et al. NERSC-10 Workload Analysis. (2020)
https://portal.nersc.gov/project/m888/nersc10/workload/N10_Workload_Analysis.latest.pdf
[2] NESAP for simulations. (2023)
https://www.nersc.gov/research-and-development/nesap/nesap-projects/#n4s
[3] APEX workflows whitepaper. (2019)
https://www.lanl.gov/projects/crossroads/__internal/__blocks/xroads_workflows_20190204.pdf
[4] NESAP for learning. (2023)
https://www.nersc.gov/research-and-development/nesap/nesap-projects/#n4l
[5] Pathak et al. FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators. (2022). https://arxiv.org/abs/2202.11214
[6] Chanussot et al. The Open Catalyst 2020 (OC20) Dataset and Community Challenges. (2021) https://arxiv.org/abs/2010.09990
[7] Stein et al. Mining for Strong Gravitational Lenses with Self-supervised Learning. (2021)
https://arxiv.org/abs/2110.00023
[8] Kurth et al. Exascale deep learning for climate analytics. (2018)
https://dl.acm.org/doi/10.5555/3291656.3291724
[9] Superfacility. (2023) https://www.nersc.gov/research-and-development/superfacility/
[10] NESAP for data. (2023)
https://www.nersc.gov/research-and-development/nesap/nesap-projects/#n4d
[11] Bard et al. The Superfacililty Project Report. (2022) https://www.osti.gov/biblio/1875256
[12] Blasche et al. Real-Time XFEL Data Analysis at SLAC and NERSC: a Trial Run of Nascent Exascale Experimental Data Analysis. (2021) https://arxiv.org/abs/2106.11469
[13] Wilkinson, M., Dumontier, M., Aalbersberg, I. et al. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data 3, 160018 (2016).
https://doi.org/10.1038/sdata.2016.18
[14] Ben Brown. DOE's Integrated Research Infrastructure. Biological and Environmental Research Advisory Committee Meeting. April 21, 2023.
https://science.osti.gov/-/media/ber/berac/pdf/202304/Brown_IRI_for_BERAC_202304.pdf
[15] Tal Ben-Nun, Torsten Hoefler. "Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis. (2018) https://doi.org/10.48550/arXiv.1802.09941

Glossary

QSS: Quality of Service Storage System
PSS: Platform Storage System
QoS: Quality of Service
FaaS: Function as a Service
IOPS: Input/Output per second
K8s: Kubernetes
WEN: Workflow Environment Nodes