A galactic approach to neutron scattering science

Gregory R. Watson[1]*, Thomas A. Maier[2], Sergey Yakubov[1] and Peter W. Doak[2]

[1]Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN, United States, [2]Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN, United States

Neutron scattering science is leading to significant advances in our understanding of materials and will be key to solving many of the challenges that society is facing today. Improvements in scientific instruments are actually making it more difficult to analyze and interpret the results of experiments due to the vast increases in the volume and complexity of data being produced and the associated computational requirements for processing that data. New approaches to enable scientists to leverage computational resources are required, and Oak Ridge National Laboratory (ORNL) has been at the forefront of developing these technologies. We recently completed the design and initial implementation of a neutrons data interpretation platform that allows seamless access to the computational resources provided by ORNL. For the first time, we have demonstrated that this platform can be used for advanced data analysis of correlated quantum materials by utilizing the world's most powerful computer system, Frontier. In particular, we have shown the end-to-end execution of the DCA++ code to determine the dynamic magnetic spin susceptibility $\chi(q, \omega)$ for a single-band Hubbard model with Coulomb repulsion $U/t = 8$ in units of the nearest-neighbor hopping amplitude t and an electron density of $n = 0.65$. The following work describes the architecture, design, and implementation of the platform and how we constructed a correlated quantum materials analysis workflow to demonstrate the viability of this system to produce scientific results.

1 Introduction

Emerging global challenges such as climate change will require technological solutions that will only come to fruition with the aid of consequential scientific advances. These advances will not be possible without significant scientific discovery. For example, the ability to transition to other forms of energy beyond fossil fuels hinges on developments such as the storage capacity of batteries and new structural components for wind turbines, and keeping global temperatures within an acceptable range is likely to require direct extraction of carbon emissions from the atmosphere. Many of these technological advances are predicated on a deep understanding of the properties of existing materials and the discovery of new classes of materials. One key technique for ascertaining information about the nature of materials is neutron scattering, which occurs when a beam of neutrons is directed at a material and some of those neutrons interact with the atomic nuclei or the electronic spin and bounce away at an angle. Special detectors can be used to measure the properties of the scattered neutrons and enable scientists to better understand the material's structure.

Oak Ridge National Laboratory (ORNL) has been at the forefront of materials science for over half a century. ORNL operates two world-leading neutron sources: the High Flux Isotope Reactor (HIFR) and the Spallation Neutron Source (SNS), both of which are funded by the US Department of Energy's (DOE's) Office of Science and Office of Basic Energy Sciences. HFIR began operations in 1965 and is currently the most powerful reactor-based source of neutrons in the United States. The SNS is the world's most powerful accelerator-based spallation source of neutrons.

Collecting neutron scattering data is only the beginning of the journey to scientific discovery. To fully understand the material properties, data must first be reduced, analyzed, interpreted (i.e., modeled), and finally published. Instruments at ORNL's facilities record individual neutron events that comprise three elements: a detector pixel identifier, the total neutrons' time-of-flight from source to detector, and the wall-clock time of the proton pulse for the associated neutron. *Data reduction* involves a series of steps to remove instrument-specific artifacts and convert this raw, event-based data into physical quantities of interest such as histograms and images. Reduction has traditionally been a manual process that often requires intervention by an instrument scientist, but fortunately ORNL has been able to automate data reduction for some of the instruments. Although this helps streamline the use of some data for subsequent analysis, there are complex manual processes needed to conduct post-experiment analysis and interpretation by the materials scientists to generate publishable results. This *analysis and interpretation* relies on a byzantine assortment of tools that include a mix of homegrown, open source, and commercial packages (with associated licensing and use restrictions). Automating both the reduction and analysis processes will not only simplify and streamline the experimental process, but has the potential to enable new capabilities, such as the ability of a scientist to use the results to alter the conditions during the experimental run, thereby potentially increasing overall scientific productivity.

To meet the challenges inherent in new materials science discoveries, the number of neutron scattering instruments employed at SNS and HFIR has steadily increased, as have the corresponding complexity and volumes of data being produced. New instruments such as the Versatile Neutron Imaging Instrument are expected to generate orders of magnitude more data than previously available. These improvements have driven a need for more advanced data analysis capabilities, including high-performance computing (HPC) materials modeling, data analytics, ML, and real-time analysis workflows, which require significantly more computational and data-intensive resources than have been needed in the past. The ability to effectively interact and visualize these high volumes of data is becoming an essential aspect of the analysis and interpretation process. With improvements to the way neutrons data is processed, the ability to collaborate directly with facilities such as the Center for Nanophase Materials Sciences and other scientific and computational facilities run by DOE is also becoming an area of significant growth potential.

To address many of the issues arising from the greater burden of computationally intensive data processing and enhanced visualization requirements, ORNL has been investing in a next-generation data analysis platform for neutron scattering

science—the Neutrons Data Interpretation Platform (NDIP). The overarching goal of this initiative was to create an ecosystem to support and enable the growth of science applications for analyzing neutron scattering data. This process began in 2021 with a requirements analysis, the development of a system architecture, and a comprehensive assessment of existing scientific workflow systems that ultimately resulted in the selection of Galaxy as the common framework technology (Watson et al., 2022). In consultation with a number of computer, computational, and instrument scientists, we developed a set of key criteria for the platform, including computing and data resource management capabilities, a workflow execution mechanism with support for human-in-the-loop interaction, a web-based user interface, a scripting application programming interface (API), and integration with security architectures, amongst others. We used this information to develop a high-level architecture for the platform that enabled further refinement and discussion with stakeholders. We then undertook a side-by-side assessment of 37 existing workflow systems against the key criteria. The goal was not to build a new platform from the ground up but to leverage existing technology to deploy the system and produce science as soon as possible.

ORNL's computational and experimental facilities operate in distinctly independent realms, and interoperating between these facilities requires interaction across different security and administrative domains. As a consequence, considerable work was required to ensure that the platform could schedule computational tasks and coordinate the delivery of data in a timely manner. In parallel to this, a number of science teams were developing new neutron science data analysis workflows that would integrate with the NDIP platform. By early 2022, we had deployed an instance that enabled these workflows to utilize the on-premises virtual computing infrastructure known as the Compute and Data Environment for Science. During 2022, we focused on the ability to seamlessly transfer data between compute resources by utilizing a data broker architecture. We also worked to extend the computational capability to include the HPC systems (particularly ORNL's Summit system) provided by the Oak Ridge Leadership Computing Facility. This culminated in the first demonstration of an end-to-end neutron science workflow across the full extent of computational resources in early 2023. For the remainder of 2023, we consolidated the architecture by developing automated deployment through infrastructure-as-code platforms such as Terraform[1] and Ansible,[2] expanding our computational resources to include stand-alone hardware, incorporating an automated data ingress mechanism for raw and reduced neutrons data, and providing additional data visualization and interactive analysis tools.

To illustrate the importance of investing in a platform such as NDIP, we will show how NDIP was able to produce a scientific result that would have been difficult or impossible with conventional techniques. For this purpose, we will consider one class of materials for which neutron scattering is a particularly

---

1 https://www.terraform.io

2 https://www.ansible.com

powerful probe, known as correlated quantum materials. These materials exhibit many novel properties, including quantum magnetism, high-temperature superconductivity, quantum criticality, and topologically protected states, many of which promise to revolutionize a wide range of technologies. However, predicting how these properties can be tuned and optimized—thereby guiding the synthesis of new useful materials—requires a detailed understanding of the mechanisms that underlie a material's behavior. This is one of the most significant challenges in materials science. Neutron scattering, with its unmatched ability to probe a material's magnetism, has proven to be one of the most important tools in the experimental study of these materials. Together with modeling and simulation, it may be used to obtain detailed knowledge of the microscopic interactions and physical mechanisms that determine a material's behavior.

2 Architecture

From its inception, the goal of NDIP was to provide an ecosystem for developing neutron science data analysis applications. Figure 1A shows a high-level conceptual architecture of the platform. The neutron science applications shown in the diagram are a set of tools and workflows that can be utilized by neutron scientists to reduce, analyze, and interpret neutron scattering data. The intention was to provide both web-based and stand-alone applications that interact with components and services in the platform. The platform itself comprised a number of standard services such as workflow management and data provenance tracking, and additional functionality that could be added through a modular architecture. The platform would in turn communicate with and leverage the federated computing and data services provided by the institutional infrastructure. One of the main challenges encountered while implementing this architecture is that many of these federated services do not currently exist at ORNL. However, we were able to overcome this problem by developing modules within the platform that emulated or replaced these services. This will be discussed in more detail in the following sections.

By building on Galaxy as the core of the platform architecture, we were able to leverage its considerable strengths for a variety of key services. Additionally, the modular design of Galaxy allowed us to adapt and extend the framework so that it was suitable for neutron scattering science applications. Figure 1B shows the high-level architecture after adapting the design to incorporate Galaxy as the core of the platform. The Galaxy framework (Community, 2022) was originally developed for the bioinformatics community in 2005 and since then has been integrated with over 8,000 software packages and has been used to generate scientific results reported in over 10,000 publications. The Galaxy framework provides a graphical interface for interacting with data analysis tools and managing data provenance by using histories. It also provides a workflow engine and the infrastructure to execute workflows on local or remote compute resources. Data can be easily accessed from public repositories, and the data created from analysis in Galaxy can be shared and published through the framework. In addition to the framework itself, Galaxy also provides a public repository of analysis tools, called the *Tool Shed*, and a number of public instances of Galaxy in Europe, North America, and Australia that

are available for free research use. A large Galaxy community also creates and maintains content for the Galaxy Training Network,[3] which provides tutorials and workflows for performing a range of data analysis tasks.

Although Galaxy provides many capabilities that we were looking for in our platform, it was designed for a different community with a different set of requirements. As expected, we needed to add new functionality to support our neutron scattering data analysis environment. The most significant need was a data management capability to ensure that data was available to execute workflows across a range of remote compute resources with separate data storage systems. Although Galaxy has an extensible architecture to support data movement (known as the *object store*), it assumes that the data is available from a centralized storage system. Other functionality we needed included support for authentication using Open ID Connect (OIDC) tokens, the ability to visualize neutron data files, the ability to monitor command execution output, and a number of other smaller requirements. However, we were convinced that the cost to implement a new framework with the same capabilities as Galaxy would far outweigh the cost of implementing these missing features. To date, we have employed ∼5 person-years of effort on core NDIP development activities. It is hard to compare this directly with the effort that has gone into Galaxy, however we know that there have been around 330 contributors making over 76,000 commits to the Galaxy core repository over a period of 16 years (the start of git commit record). Our experience leads us to believe that this is at least an order of magnitude more effort than we have expended to date.

3 Implementation challenges

ORNL's computational and data resources are managed by different groups in different security and administrative domains. This disparate environment makes for a complex situation when trying to provide seamless access to these resources—something that had so far been impossible to achieve. Although Galaxy provides a general workflow platform designed to support a wide range of different environments, its design makes some basic assumptions about the configuration of resources and infrastructure and how these can be accessed. Unfortunately, these do not always align well with the network architecture at ORNL. Figure 2 depicts a schematic of the design we developed to overcome these limitations. The diagram shows a number of administrative domains separated by firewalls (heavy-dashed lines). Domain 1 represents an organizational unit that provides some dedicated compute resources that they want to leverage in the NDIP ecosystem. To do this, a Pulsar server and an RDB client must be available on the compute resource. Domain 2 represents the HPC resources at ORNL. These are fronted by an OpenShift[4]-based cluster that provides job submission capability and access to the shared HPC data storage. One or more Pulsar servers and an RDB client must run on this cluster. Domain 3 represents the main Galaxy instance along with an OpenStack[5] cluster that provides virtualized compute resources. The message

---

3 https://training.galaxyproject.org/

4 https://docs.openshift.com

5 https://openstack.org

FIGURE 1
High-level architecture of the Neutrons Data Interpretation Platform shown in the green boxes before and after adopting the Galaxy framework as the core of the platform. The yellow boxes represent instruments that generate experimental data. The blue boxes show the federated services and the different requirements after Galaxy adoption. **(A)** shows the platform providing workflow management and data provenance services as core components. Other services are extensible via a plug-in mechanism. This platform relies on a range of external capabilities provided by federated compute and data services that also enable acquisition (and control) of instruments. **(B)** Galaxy provides data and compute management services by utilizing a custom data broker built on top of Rucio along with the Galaxy Pulsar services which communicate using the RabbitMQ message broker. The science applications are now workflows and tools integrated with the Galaxy platform.

server (RabbitMQ in the diagram) is also shown in this domain. Domain 4 represents a data storage resource (perhaps containing experimental data acquired from a scientific instrument). Only an RDB client is required. Communication across domains is limited to messages passed via the message broker (dashed arrows) or outbound connections between the RDB clients and the RDB server when an interdomain data transfer is required.

3.1 Security

One of the goals of NDIP was to provide seamless execution of workflows across a range of computational resources. Although the identity and access management functionality provided by Galaxy supports a number of options, including local username/password administration, OIDC, and external authentication services such as the Lightweight Directory Access Protocol, none offered the functionality we required. This was primarily because authentication/authorization in Galaxy only applies to user identities for controlling access to histories, datasets, visualizations, and other user-related information. Although Galaxy can be configured for remote job execution on clusters or other HPC resources (either directly or by leveraging the Pulsar service), it is limited to two types of remote job submission: either using a service account that has been pre-authorized or by configuring the remote system to allow switching credentials using the sudo[6] command.

Galaxy can use the OIDC *authorization code flow* protocol to authenticate a user. The OIDC specification defines this as a process that allows a user's identity to first be verified by an identity provider; security tokens (e.g., ID token and access token) are then received from the provider which contain claims

about the authentication and authorization of the user. These are JavaScript Object Notation (JSON) Web Tokens (JWTs) that are cryptographically signed. As described in the literature (Jalili et al., 2019), it is possible to then use the identity token to obtain credentials that allow access to resources in a cloud environment. Our goal was to provide similar functionality but utilize the JWT credentials to enable traditional access control mechanisms in an HPC environment using RFC8693[7] token exchange to enable impersonation semantics.

To meet these security requirements, we modified Galaxy to expose the OIDC tokens to Pulsar servers. This simple change allowed the token to be used for additional authorization steps on remote resources. We developed a command-line tool that takes an OIDC access token and uses it to impersonate a user when executing a command (similar to how the sudo command operates). Additionally, we developed a pluggable authentication module (PAM) that allows SSH (Secure Shell Protocol) to issue remote commands that are authenticated by the OIDC access token (Yakubov et al., 2024). Our module is different from other PAM modules that utilize the OAuth 2.0 device flow (Surkont et al., 2020) because it is designed primarily for automated execution of remote commands with transparent credential validation. We describe how these tools are used in more detail in later sections. Finally, we used a combination of these methods to allow containerized tools to assume the same identity as the user who initiated the tool.

3.2 Data management

As discussed previously, separating computing resources into different security and administrative domains causes problems

FIGURE 2
Major components of the Neutrons Data Interpretation Platform. The dashed lines indicate separate security or administrative domains. Traversing a domain generally requires passing through a firewall. Galaxy components are shown in the gray boxes and include Galaxy Server, Galaxy User Interfaces, and Pulsar. Other technologies we leveraged are shown in dotted boxes: RabbitMQ and Rucio. Components we developed locally include the Rucio Object Broker Plugin (OBP) and the Remote Data Broker (RDB) client and server. The Resource Storage Elements (RSEs) are data storage resources that are known and accessible to Rucio. Using a message broker for communication is key to the implementation. All connections are outbound, and this significantly simplifies deployment across multiple domains (as shown) because many firewall policies allow outbound connections or are amenable to opening a single port.

when attempting to provide seamless and secure workflow execution across arbitrary resources. In addition to hindering remotely executed commands, these separate domains also impact users' ability to share/transfer data for analysis. For its part, Galaxy natively supports two primary mechanisms for data management. The first is to rely on a shared file system being available across all the services in the installation. This approach works well in some instances but is not usable when multiple security domains are employed (such as in our case). The second is to use object storage (e.g., S3 object store) to download/upload data to some central location. Although this second approach would work in our environment, it severely limits the data sizes that can be transferred in a timely manner. Our users are expecting files in the gigabyte and terabyte ranges, so this option is inadequate.

Our solution was to employ different protocols depending on the restrictions imposed by the underlying security/administrative domains. After some investigation, we found that Rucio provided the best decentralized multiprotocol solution for data management. Rucio is a scientific data management system developed by CERN and adopted across a number of scientific disciplines. A key feature of Rucio is that it provides centralized management, but the data can be stored on heterogeneous servers. This was particularly important to us because we needed to utilize data servers that were being used for other purposes but could be integrated into the NDIP environment. We implemented a Galaxy object store that can use Rucio for data management and an RDB that simplified transferring data across security domains. These are labeled as the Rucio Object Store Plugin (Rucio OSP), RDB client, and RDB server, respectively, in Figure 2. The RDB is used primarily to

facilitate data management operations from HPC storage when a firewall allows only outbound connections. This enables Galaxy to request data necessary for displaying in Galaxy's web interface or providing input to a tool that executes on a compute resource without direct HPC storage access (e.g., on a cloud-based virtual machine). Similar to Pulsar, the RDB listens to the message queue for commands to send data and then uploads the data to the final destination.

3.3 HPC resources

Because of firewall restrictions and the absence of shared storage between the cloud and HPC resources, Galaxy cannot directly submit jobs to our HPC systems (including both Summit and Frontier). To address this issue, we leveraged a local OpenShift-based cluster that had direct access to the HPC resources for batch job submission and access to HPC data storage systems. OpenShift is a container orchestration platform for running user-managed persistent application services. Using this capability, we developed a mechanism for navigating the security and administrative problems typically encountered in computing facilities such as the Oak Ridge Leadership Computing Facility.

Our approach was to deploy a Pulsar server that contained the Rucio OSP to manage data and submit jobs for each HPC resource, along with a single RDB client to handle data transfers outside the domain. Both ran as services in Kubernetes pods and listened for messages from Galaxy via a queue on the message broker. When Galaxy had a job ready for execution on an HPC resource, it sent a message containing job information (e.g., tool to execute, tool parameters, input and output data information) to this queue. The Pulsar service then performed the following operations:

- Authorized the job by verifying the OIDC token for the user and extracting the username. If the user was not authorized, then the job was rejected.
- Handled input data by requesting it from the object store (Rucio OSP). Depending on the data location, this could be as simple as creating a symlink to the data on the HPC storage system, or it may have required data to be downloaded from remote storage, in which case the appropriate request was sent to the RDB server.
- Monitored the job execution and sent updates to Galaxy. It also periodically sent updates about `stdout/stderror` to provide live information to the user.
- After the job finished, Pulsar uploaded output data information to the object store (the data itself was not uploaded but remained in HPC storage). The RDB client running in the second pod was then able to process requests to serve this data to external (to HPC) services.

4 Advanced neutron data analysis for quantum materials

One of the science teams involved early in the initiative focused on data analysis for correlated quantum materials. We will use this as an example of the type of complex problem that NDIP is aiming to help solve, and so it is illustrative to describe the nature of the analysis in more detail. In particular, modeling the physics of these systems is itself a grand challenge, so establishing a tight link between experiment and theory for correlated quantum materials is incredibly difficult. This is because the electron-electron interactions in these materials are strong. Mainstream modeling approaches used to analyze neutron scattering experiments often include density functional theory and linear spin wave theory. However, these techniques generally fail to describe the complexity arising from the simultaneous presence of correlated localized electronic states and other, weakly correlated itinerant states and the coupling between them. For example, linear spin wave theory cannot describe the broadening of spin wave excitations due to their interaction with the itinerant degrees of freedom. Instead, sophisticated many-body approaches that accurately handle the interactions between many electrons are needed to reliably model these systems.

The approach taken for developing a reliable neutron data analysis capability for correlated quantum materials was based on dynamical mean-field theory (DMFT) (Georges et al., 1996) and its cluster extension, the dynamic cluster approximation (DCA) (Maier et al., 2005), as implemented in ORNL's DCA++ code (Hähner et al., 2020). These methods are particularly well-suited to treat the coupling between strongly correlated local degrees of freedom with itinerant states that govern the physics in these systems. They also provide a powerful approach (Park et al., 2011) to calculating the dynamic magnetic neutron scattering form factor $S(Q, \omega)$, which is measured in inelastic neutron scattering experiments and provides important information about the magnetic excitations of quantum materials.

DMFT and DCA make the many-body problem tractable by partitioning the full problem into two parts: (1) a single atom (DMFT) or a cluster of atoms (DCA) for which the many-body problem is solved exactly and (2) the remaining sites in the crystal that are coupled with the impurity or cluster through a dynamic mean-field that is determined self-consistently. This approximation retains the effects of local (DMFT) or short-ranged (DCA) dynamical electron-electron correlations but neglects longer-ranged correlations. These methods are particularly well-suited to describe on equal footing the effects of local and short-ranged correlations associated with localized degrees of freedom as well as their coupling with itinerant degrees of freedom. The physics is beyond the capabilities of standard analysis approaches, including density functional theory or linear spin-wave theory, and is particularly important for an accurate description of several interesting phenomena such as the broadening of spin-wave excitations caused by interaction with the Stoner continuum (Do et al., 2022).

The effective impurity (DMFT) or cluster (DCA) problem is solved with a numerically exact quantum Monte Carlo (QMC) algorithm (Jarrell et al., 2001; Gull et al., 2011), which, despite the reduced complexity, is in general still a formidable task that requires high-end computing resources (Balduzzi et al., 2019) owing to the exponential scaling of the computational requirements for solving the underlying many-body problem. Once this computation is converged, an additional analytic continuation

is required to extract the real time or real frequency dynamics for $S(Q, \omega)$ because QMC provides data only with imaginary time or frequency dependence. This is an ill-posed inverse problem that requires regularization, for which we will use a Bayesian inference–based Maximum Entropy (MaxEnt) approach (Jarrell and Gubernatis, 1996) as the workhorse.

Calculations for complex multi-orbital models of interest for most quantum materials, for which the DMFT impurity or the DCA cluster problem is too small to provide sufficient momentum Q-resolution, require the additional computation of a 4-point scattering vertex function, $\Gamma_{\ell_1, \ell_2, \ell_3, \ell_4}(\omega_1, \omega_2, \omega_3)$; a high-dimensional rank-4 tensor that depends on the energies ω, spin; and orbital quantum numbers ℓ_i of two incoming and two outgoing electrons. This quantity is needed to construct the full Q-dependence of $S(Q, \omega)$ through an additional Bethe-Salpeter equation (Maier et al., 2005). Earlier work that used DMFT to calculate $S(Q, \omega)$ for iron-based superconductors neglected the energy dependence of Γ to reduce this complexity (Do et al., 2022). The subsequent analytic continuation requires $O(1,000)$ data bins for this quantity. The complete workflow to predict $S(Q, \omega)$ for a quantum material is illustrated in Figure 3.

Prior to the deployment of NDIP, there was no way to run a fully automated end-to-end correlated quantum materials data analysis workflow using the HPC resources available at ORNL. Users had to build and install the DCA++ code on the destination HPC system themselves, write a job submission script that requested resources necessary for the run, manually transfer the input configuration files, and then submit the script to the job scheduler. On completion of the HPC portion of the run, it was then necessary to copy the resulting data files to a different system to perform the remaining steps due to policy restrictions preventing small jobs from running on HPC resources.

5 Methodology

Since no previous capability existed to enable the automated workflows of this nature, our primary objective was to demonstrate that the system could execute an end-to-end correlated quantum materials data analysis workflow using a combination of the HPC and cloud computing resources available at ORNL. This would prove our architecture's ability to support advanced neutron

FIGURE 3
Illustration of the DCA++ based workflow to predict the magnetic inelastic neutron scattering intensity $S(Q, \omega)$. The workflow involves two DCA++ calculations: one for convergence and a second one to generate data bins for the 4-point correlation function, which is then used in a Bethe-Salpeter post-processing script that extends the local DCA++ results to the full crystal lattice for full momentum Q resolution. This data is then fed into the final analytic continuation step in which a Maximum Entropy based script generates data that is assembled into $S(Q, \omega)$.

FIGURE 4
DCA++ Workflow in Galaxy. Each step is supplied with one or more inputs and generates an output file. If the data type of an output matches that of an input, then the two steps can be connected together and Galaxy will manage the data transfer. This workflow begins with two data files (`conv.json` and `input.bins.json`) that are used as inputs to subsequent steps and ends with the DCA Plot step.

TABLE 1 DCA++ workflow components showing the source language, input file format, output file format, and the computational resource requirements for the tool.

Tool name	Source	Input format	Output format	Resources
Convergence	C++	json	hdf5	HPC/GPU
Binning	C++	json, hdf5	hdf5	HPC/GPU
Bins for $S(Q,\omega)$	Python	hdf5	txt	Cloud VM
MaxEnt	Python	txt	txt	Cloud VM
Plot	Python	txt	png	Cloud VM

scattering analysis techniques. We felt that this would be a significant advancement in workflow capability at ORNL, as it would transform a manual process requiring advanced developer skills (even if the compiled DCA++ executable was made generally available for other users) into a process that any user could initiate with a few mouse clicks.

To achieve this required some work to transform the DCA++ based workflow to the Galaxy-based workflow shown in Figure 4 so that it could be integrated into the platform. First, we split the workflow into five separate modules: (i) Convergence, (ii) Binning, (iii) Bins for $S(Q,\omega)$, (iv) MaxEnt, and (v) MaxEnt Plot. These modules were then used to create tools that could run on the required (HPC or non-HPC) computational resources and be integrated into the Galaxy platform. Our reasoning for separating the workflow into components is as follows:

- Not all components required (or can even be run on) HPC resources but could be run on smaller and more efficient resources.
- The execution time of each component could be easily measured, which enables us to identify and address workflow bottlenecks.
- The workflow can be restarted at an arbitrary point (e.g., when/if incorrect data is discovered).
- The tools can also be reused to implement other analysis workflows.

Table 1 provides a summary of the tools, including data format and computational requirements. As shown, the first two tools are very compute- and data-intensive and required significant HPC resources to run.

Integrating a Galaxy tool required the creation of an XML file that described, among other things, the input data consumed by the tool, the output data generated by the tool, and how to execute the tool itself. The Galaxy platform manages the transfer of data from tool to tool, thereby ensuring that the data is available in the correct location prior to the tool's execution. Galaxy also initiates the execution of the tool on the remote computational resource. For some tools, the input and/or output files comprise gigabytes of data, so it was important that they were not transferred unless absolutely necessary. We did this by configuring the Rucio object store to ensure that the data from these tools remained local to the computational resources where it was generated. Tools can specify arbitrary mechanisms to use for execution such as running

an executable command, launching a containerized application, or submitting a batch job. Galaxy enables a series of tools to be combined into a Galaxy workflow, and this was done to implement the full DCA++ workflow.

Listing 1 Convergence step input file.

```
"output": {
    "directory": "./",
    "output-format": "HDF5",
    "g4-ouput-format": "HDF5",
    "filename-dca": "conv.hdf5",
    "dump-lattice-self-energy": true,
    "dump-cluster-Greens-functions" : true,
    "dump-every-iteration" : false,
    "dump-Gamma-lattice" : true,
    "dump-chi-0-lattice" : true
},
"physics": {
  "beta": 100.0,
  "density": 0.65,
  "adjust-chemical-potential": true
},
"single-band-Hubbard-model": {
  "t": 1,
  "t-prime": 0.0,
  "U": 8
},
"DCA": {
  "initial-self-energy": "zero",
  "iterations": 10,
  "accuracy": 1e-3,
  "self-energy-mixing-factor": 0.8,
  "interacting-orbitals": [
    0
  ],
  "coarse-graining": {
    "k-mesh-recursion": 3,
    "periods": 0,
    "quadrature-rule": 1,
    "threads": 2,
    "tail-frequencies": 10
  }
},
"domains": {
  "real-space-grids": {
    "cluster": [
      [
        1,
        0
      ],
      [
        0,
        1
      ]
    ],
    "sp-host": [
      [
        20,
        20
      ],
      [
        20,
        -20
      ]
    ],
    "tp-host": [
      [
        8,
```

```
          8
        ],
        [
          8,
          -8
        ]
      ]
    },
    "imaginary-time": {
      "sp-time-intervals": 512,
      "time-intervals-for-time-measurements": 32
    },
    "imaginary-frequency": {
      "sp-fermionic-frequencies": 2048,
      "four-point-fermionic-frequencies": 64,
      "HTS-bosonic-frequencies": 32
    }
  },

  "Monte-Carlo-integration": {
    "warm-up-sweeps": 500,
    "sweeps-per-measurement" : 32,
    "measurements": 10000000,
    "seed": 9854510000,
    "threaded-solver": {
      "accumulators": 4,
      "walkers": 4,
      "shared-walk-and-accumulation-thread": true
    }
  },
  "CT-INT": {
      "initial-configuration-size": 15,
      "max-submatrix-size": 256
  }
```

Listing 2 Binning step input file.

```
  "output": {
      "directory": "./",
      "output-format": "HDF5",
      "g4-ouput-format": "HDF5",
      "filename-dca": "g4_bins.hdf5",
      "dump-lattice-self-energy": true,
      "dump-cluster-Greens-functions" : true,
      "dump-every-iteration" : true,
      "dump-Gamma-lattice" : true,
      "dump-chi-0-lattice" : true
    },
  "physics": {
    "beta": 100.0,
    "density": 0.65,
    "adjust-chemical-potential": false
  },
  "single-band-Hubbard-model": {
    "t": 1,
    "t-prime": 0.0,
    "U": 8
  },
  "DCA": {
    "initial-self-energy": "conv.hdf5",
    "iterations": 100,
    "self-energy-mixing-factor": 0.8,
      "do-not-update-Sigma": true,
    "interacting-orbitals": [
      0
    ],
    "coarse-graining": {
      "k-mesh-recursion": 3,
      "periods": 0,
      "quadrature-rule": 1,
```

```
      "threads": 6,
      "tail-frequencies": 10
    }
  },
  "domains": {
    "real-space-grids": {
      "cluster": [
        [
          1,
          0
        ],
        [
          0,
          1
        ]
      ],
      "sp-host": [
        [
          20,
          20
        ],
        [
          20,
          -20
        ]
      ],
      "tp-host": [
        [
          8,
          8
        ],
        [
          8,
          -8
        ]
      ]
    },
    "imaginary-time": {
      "sp-time-intervals": 512,
      "time-intervals-for-time-measurements": 32
    },
    "imaginary-frequency": {
      "sp-fermionic-frequencies": 2048,
      "four-point-fermionic-frequencies": 128,
      "HTS-bosonic-frequencies": 32
    }
  },

  "Monte-Carlo-integration": {
    "warm-up-sweeps": 500,
    "sweeps-per-measurement" : 32,
    "measurements": 100000,
    "seed": 9854510100,
    "threaded-solver": {
      "accumulators": 4,
      "walkers": 4,
      "shared-walk-and-accumulation-thread": true
    }
  },
  "CT-INT": {
      "initial-configuration-size": 15,
      "max-submatrix-size": 256
  },
    "four-point": {
    "channels": [
      "PARTICLE_HOLE_MAGNETIC"
    ],
    "frequency-transfer": 32,
    "compute-all-transfers": true
```

```
    },
    "analysis": {
        "q-host": [
            [
                10,
                0
            ],
            [
                0,
                10
            ]
        ]
    }
}
```

For the DCA++ workflow, two input files in JSON format were required: `conv.json` (Listing 1) and `input.bins.json` (Listing 2). The first specifies the parameters required for the Convergence tool, and the second specifies parameters for the Binning tool. The contents of these files are available in our GitHub repository. Once these data files were provided as inputs to the workflow, Galaxy could then run an end-to-end simulation by executing the series of tools in sequence. The final tool was used to generate the results as a plot. Notably, the entire workflow was automated and managed by Galaxy, which eliminates the need for user interaction during workflow execution. The workflow could be easily rerun with different input parameters, and all results were stored in a Galaxy history to facilitate sharing with other scientists, reproducibility at a later time, and comparison of different software versions.

6 Results

Our first attempt at running the end-to-end workflow was on the Summit system, as at the time the Frontier system was not due to enter service until January 2024. For this run, we chose 100 nodes (600 GPUs) as a reasonable compromise between system availability and expected runtime. The run eventually completed successfully (in excess of 3 days), and this enabled us to fully verify the operation of the entire workflow.

Due to issues with the correctness of DCA++ code dependencies, we were unable to test large-scale runs on Summit until late 2023, and Summit was taken out of service before we were able to complete a final run. As a consequence, we transferred our attention to Frontier. However, due to issues using the OpenShift cluster to directly submit jobs to Frontier, we had to manually run a Pulsar and an RDB server on one of Frontier's login nodes to allow for Galaxy job submission. Since we had previously verified the full workflow on Summit, we believed that this would not detract from the final result.[8] Additionally, while this was not an ideal situation for the final run, it does demonstrate the considerable flexibility of the architecture in overcoming such an adverse situation.

To demonstrate the flexibility of the system to adapt to different resource requirements and allow for future optimization and scaling studies, we spent some time experimenting with different job sizes for the workflow steps. Since we had no previous experiments to compare with, we decided to focus on generating

---

8   Once the job submission issue is resolved we will revert back to using the OpenShift cluster for future job submissions.

TABLE 2 DCA++ workflow execution results showing the tool runtime, the number of CPUs and GPUs used in each run, and the resulting file sizes.

Tool name	Runtime (s)	CPUs	GPUs	Output data size
Convergence	6,576	1,024	8,192	7.2 MB
Binning	6,626	184	1,472	3.5 GB
Bins for $S(Q, \omega)$	14,220	64	–	1.7 MB
MaxEnt	6,180	8	–	4.9 MB
Total	**33,602**			

The final plot is not included because the runtime and output size were negligible.

simulation results using a large resource allocation that would result in a reasonable wall-time of around 2 hours per step. By varying the requested resources via a parameter in the workflow job step, we found that 1,024 Frontier nodes (8,192 GPUs) for the Convergence step and 184 nodes (1,472 GPUs) for the Binning step produced results in the desired time frame, thus demonstrating the ease at which different resource requirements could be accommodated. One particular feature of this workflow was that the post-processing steps were not parallelized and therefore not able to be run on Frontier due to local allocation size/time limit policies. As a consequence, this provided a good use-case for demonstrating the capability of the system to run part of the workflow on HPC resources and part on a cloud-based virtual machine. During the workflow execution, Galaxy seamlessly transferred the required data from Frontier to cloud storage and executed the remaining workflow steps, hence demonstrating the viability of this approach. Table 2 shows the corresponding job sizes and run times for each tool. The overall runtime for the workflow was 33,602 s which provides a useful benchmark for future improvements in improving the overall efficiency of the workflow.

From the neutron science perspective, we were able to provide a significant result, with the final output of the workflow shown in Figure 5. This calculation of the dynamic magnetic spin susceptibility $\chi(q, \omega)$ was performed for a single-band Hubbard model with Coulomb repulsion $U/t = 8$ in units of the nearest-neighbor hopping amplitude t and an electron density of $n = 0.65$, relevant for materials like the copper-oxide high-temperature superconductors with a single electronic band crossing the Fermi energy. From the heatmap showing the imaginary part of $\chi(q, \omega)$ one sees that at this electron density, the magnetic excitation spectrum is broad and incoherent, with most of the spectral weight located at high energies $\omega > t$, except for a small region of lower energy excitations near $q = (\pi, \pi/2)$. The magnetic susceptibility $\chi(q, \omega)$ can potentially be measured in future inelastic neutron scattering experiments.

7 Future work

The NDIP is under active development, and there are many opportunities for improvement. We currently have over 60 neutron scattering tools integrated, including the DCA++ tools and many others. This is only a small fraction of the total tool requirements
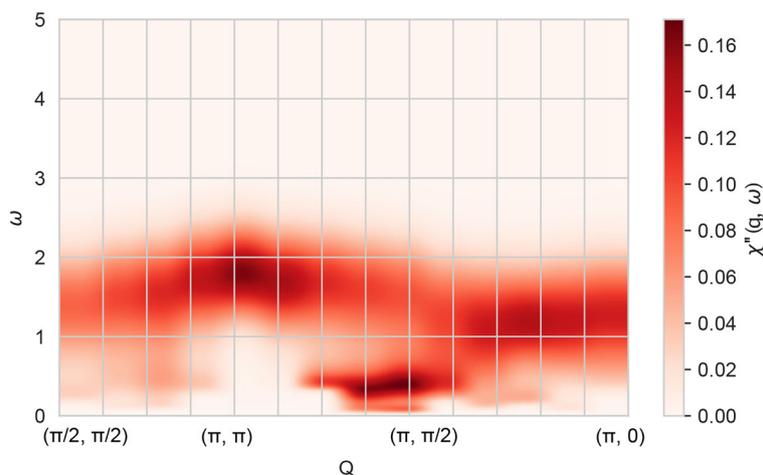
FIGURE 5
Final result from running the DCA++ based workflow on Frontier. The imaginary part of the dynamic magnetic susceptibility $\chi''(q, \omega)$ displayed as a heatmap over frequency ω and along a path in two-dimensional momentum (q) space provides information on the magnetic excitation spectrum of a material. This data was obtained for a single-band Hubbard model with Coulomb repulsion $U/t = 8$ in units of the nearest-neighbor hopping amplitude t and an electron density of $n = 0.65$, relevant for materials like the copper-oxide high-temperature superconductors with a single electronic band crossing the Fermi energy.

for ORNL's SNS and HFIR facilities, which operate over 30 different scientific instruments.

Another important area for future work is to transition to cyberinfrastructure being developed at ORNL and beyond. The ORNL Interconnected Science Ecosystem (INTERSECT) effort (Engelmann et al., 2022) is working to develop infrastructure to more easily connect scientific instruments to computational resources. As this system becomes more stable and functional, we plan to transition NDIP to leverage the services provided by INTERSECT rather than our own. Initially, this would involve the message service but would ultimately extend to job orchestration and data management services. In addition to intra-organizational resources, DOE's Integrated Research Infrastructure (IRI) (Miller et al., 2023) aims to provide connectivity and services between DOE facilities. Many scientists already utilize services beyond their own organizations (e.g., resources at the National Energy Research Scientific Computing Center) and are familiar with navigating them. By leveraging IRI, we expect to bring this capability to data analysis for neutron scattering.

From a security perspective, the use of ODIC tokens provides a uniform and future-proof authentication and authorization mechanism. However, ORNL computational resources still rely on more traditional authentication and authorization. A more scalable and secure approach would be to employ a role-based access control (RBAC) system that uses custom claims embedded in the OIDC tokens. This would allow user roles to be assigned at the institutional level and have this propagate all the way to the computational resources being employed by the user. Our goal is to replace the existing mechanisms with RBAC as this becomes more widely adopted across ORNL's infrastructure.

From the scientific workflow perspective, DCA++ requires leadership-class computing resources to simulate complex quantum materials on a reasonable timescale. This is due to the underlying quantum many-body problem that scales exponentially in the degrees of freedom (atomic orbitals). The DCA++ code currently employs a QMC algorithm to repeatedly solve this many-body problem until the calculation converges. Although this algorithm is exact, its computational burden is often excessive. The use of less numerically expensive surrogates as a stand-in for the expensive QMC solver is therefore an important future goal. Approximate physics-based algorithms used to solve the quantum many-body problem, often perturbative in nature, can speed up simulations significantly, albeit at a reduced accuracy vs. the numerically exact QMC solution. Another complementary strategy that is rapidly gaining traction is the use of ML. If trained by the original QMC algorithm, such ML surrogates could deliver similar accuracy as the QMC solver but at a fraction of the cost. The acceleration of DCA++ simulations resulting from such surrogates would therefore enable rapid materials parameter scans, deployment on smaller-scale resources such as edge devices, and near–real time interpretation and feedback for the experiments.

For this initial experiment we were not concerned with overall runtime or the scalability of the workflow. However, we did observe that one of the workflow steps, "Bins for $S(Q, \omega)$," was a significant outlier in terms of wall-clock time when compared with the other steps. This step would be a worthy candidate for additional optimization and/or parallelization efforts. We intend to perform further experiments to compare the execution time of the workflow steps and overall workflow execution time with the manually executed code to determine scalability and potential optimization opportunities.

8 Conclusion

We have described the architecture and implementation of a platform for the analysis of neutron scattering data and demonstrated that it can be used for real-world problems such as correlated quantum materials data analysis. We have shown that existing technologies combined with new functionally that we developed result in a novel architecture that addresses the needs of a complex computational and data environment across

multiple security and administrative domains. We demonstrated that this system can be used to run real neutron science problems at a massive scale on leadership-class computing resources. We did this by running a numerically exact QMC algorithm using 8,192 GPUs on Frontier followed by data binning for the 4-point correlation function using 1,472 GPUs on the same machine. The remaining post-processing steps were performed by utilizing virtual machines from on-premises cloud resources. The entire workflow was completed in under 10 h of wall-clock time.

For the first time, we (at ORNL) have been able to run a complex workflow utilizing a combination of cloud and leadership-class computing resources to achieve a real scientific result. We have demonstrated that this is not only possible for a specific scientific workflow, but that it can be employed as a general approach to meeting the computational needs of advanced neutron scattering data analysis problems.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

GW: Conceptualization, Funding acquisition, Investigation, Methodology, Project administration, Software, Supervision, Writing – original draft. TM: Conceptualization, Investigation, Methodology, Resources, Software, Supervision, Validation, Writing – original draft. SY: Investigation, Methodology, Software, Supervision, Validation, Writing – original draft. PD: Investigation, Methodology, Software, Validation, Writing – review & editing.

Funding

Acknowledgments

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Balduzzi, G., Chatterjee, A., Li, Y. W., Doak, P. W., Haehner, U., D'Azevedo, E. F., et al. (2019). "Accelerating DCA++ (dynamical cluster approximation) scientific application on the summit supercomputer," in *2019 28th International Conference on Parallel Architectures and Compilation Techniques (PACT)* (Seattle, WA), 433–444.

Community, T. G. (2022). The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update. *Nucleic Acids Res.* 50, W345–W351. doi: 10.1093/nar/gkac610

Do, S.-H., Kaneko, K., Kajimoto, R., Kamazawa, K., Stone, M. B., Lin, J. Y. Y., et al. (2022). Damped Dirac magnon in the metallic kagome antiferromagnet FeSn. *Phys. Rev. B* 105, L180403–L180403. doi: 10.1103/PhysRevB.105.L180403

Engelmann, C., Kuchar, O., Boehm, S., Brim, M. J., Naughton, T., Somnath, S., et al. (2022). "The intersect open federated architecture for the laboratory of the future," in *Accelerating Science and Engineering Discoveries Through Integrated Research Infrastructure for Experiment, Big Data, Modeling and Simulation*, eds. K. Doug, G. Al, S. Pophale, H. Liu, and S. Parete-Koon (Cham: Springer Nature Switzerland), 173–190.

Georges, A., Kotliar, G., Krauth, W., and Rozenberg, M. J. (1996). Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions. *Rev. Mod. Phys.* 68, 13–125. doi: 10.1103/RevModPhys.68.13

Gull, E., Millis, A. J., Lichtenstein, A. I., Troyer, M., and Werner, P. (2011). Continuous-time Monte Carlo methods for quantum impurity models. *Rev. Mod. Phys.* 83, 349–404. doi: 10.1103/RevModPhys.83.349

Hähner, U. R., Alvarez, G., Maier, T. A., Solcà, R., Staar, P., Summers, M. S., et al. (2020). DCA++: a software framework to solve correlated electron problems with modern quantum cluster methods. *Comput. Phys. Commun.* 246, 106709–106709. doi: 10.1016/j.cpc.2019.01.006

Jalili, V., Afgan, E., Taylor, J., and Goecks, J. (2019). Cloud bursting galaxy: federated identity and access management. *Bioinformatics* 36:btz472. doi: 10.1093/bioinformatics/btz472

Jarrell, M., and Gubernatis, J. (1996). Bayesian inference and the analytic continuation of imaginary-time quantum Monte Carlo Data. *Phys. Rep.* 269, 133–195. doi: 10.1016/0370-1573(95)00074-7

Jarrell, M., Maier, T., Huscroft, C., and Moukouri, S. (2001). Quantum Monte Carlo algorithm for nonlocal corrections to the dynamical mean-field approximation. *Phys. Rev. B* 64, 195130–195130. doi: 10.1103/PhysRevB.64.195130

Maier, T., Jarrell, M., Pruschke, T., and Hettler, M. (2005). Quantum cluster theories. *Rev. Mod. Phys.* 77, 1027–1080. doi: 10.1103/RevModPhys.77.1027

Miller, W. L., Bard, D., Boehnlein, A., Fagnan, K., Guok, C., Lançon, E., et al. (2023). *Integrated Research Infrastructure Architecture Blueprint Activity (Final Report 2023)*. doi: 10.2172/1984466

Park, H., Haule, K., and Kotliar, G. (2011). Magnetic excitation spectra in BaFe2As2: a two-particle approach within a combination of the density functional theory and the dynamical mean-field theory method. *Phys. Rev. Lett.* 107, 137007–137007. doi: 10.1103/PhysRevLett.107.137007

Surkont, J., Prochazka, M., Furnell, W., Bockelman, B. P., Velisek, O., and USDOE (2020). *Pam Module for Oauth 2.0 Device Flow*. doi: 10.11578/dc.20220727.6. Available online at: https://www.osti.gov//servlets/purl/1878149

Watson, G. R., Cage, G., Fortney, J., Granroth, G. E., Hughes, H., Maier, T., et al. (2022). "Calvera: a platform for the interpretation and analysis of neutron scattering data," in *Accelerating Science and Engineering Discoveries Through Integrated Research Infrastructure for Experiment, Big Data, Modeling and Simulation*, eds. K. Doug, G. Al, S. Pophale, H. Liu, and S. Parete-Koon (Cham: Springer Nature Switzerland), 137–154.

Yakubov, S., Cage, G., and Watson, G. (2024). *Oidc Utils*. [*Computer Software*]. doi: 10.11578/dc.20240215.1. Available online at: https://github.com/PaNGalaxy/oidc-utils