Expediting Higher Fidelity Plasma State Reconstructions for the DIII-D National Fusion Facility Using Leadership Class Computing Resources

Sterling Paul Smith
General Atomics
San Diego, USA
0000-0003-1526-380X

Zichuan Anthony Xing
General Atomics
San Diego, USA
0000-0003-3887-0792

Torrin Bechtel Amara
General Atomics
San Diego, USA
0009-0001-8102-3962

Severin Sebastian Denk
General Atomics
San Diego, USA
0000-0002-9077-4610

Earl William DeShazer
General Atomics
San Diego, USA
0000-0001-6287-7615

Orso Meneghini
General Atomics
San Diego, USA
0000-0001-5100-5483

Tom Neiser
General Atomics
San Diego, USA
0000-0002-8763-3016

Laurie Stephey
Lawrence Berkeley National Laboratory
Berkeley, USA
0000-0003-3868-6178

Oscar Antepara
Lawrence Berkeley National Laboratory
Berkeley, USA
0000-0002-4596-0289

Christopher Mitchell Clark
General Atomics
San Diego, USA
0009-0006-6063-1535

Eli Dart
Lawrence Berkeley National Laboratory
Berkeley, USA
0000-0002-8229-5433

Pengfei Ding
Lawrence Berkeley National Laboratory
Berkeley, USA
0000-0002-4050-1753

Sean Flanagan
General Atomics
San Diego, USA
0009-0005-6490-9883

Raffi Nazikian
General Atomics
San Diego, USA
0000-0001-6235-6692

David Schissel
General Atomics
San Diego, USA
0009-0006-7841-9284

Christine Simpson
Argonne National Laboratory
Lemont, USA
0000-0001-9985-1814

Nicholas Scoville Tyler
Lawrence Berkeley National Laboratory
Berkeley, USA
0000-0002-2633-0230

Thomas D Uram
Argonne National Laboratory
Lemont, USA
turam@anl.gov

Samuel Webb Williams
Lawrence Berkeley National Laboratory
Berkeley, USA
0000-0002-8327-5717

***Abstract*—The OMFIT CAKE workflow for plasma state reconstruction has been automated on the DIII-D National Fusion Facility to run on a combination of computational resources at DIII-D and the National Energy Research Computing (NERSC) Center, utilizing the emerging DOE Integrated Research Infrastructure (IRI). The reconstruction of the plasma state is vital for understanding what occurred in the DIII-D machine during the pulse. This understanding allows informed decisions to be made on how to change the configuration for the next pulse. The initial reconstruction workflow was performed on DIII-D resources for a benchmark case in 62 minutes. The wall-clock time for the benchmark case was reduced to 11 minutes by running on the Perlmutter system at NERSC, which opens the possibility to influence decisions between DIII-D pulses during an experiment. The reconstruction results can be used as inputs for other modeling analyses; the determination of the classification of the microturbulence modes is given as an example model analysis.**

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Acquisition and Assistance and the Fusion Energy Sciences program under Award Numbers DE-FC02-04ER54698, DE-AC02-05CH11231, and DE-AC02-06CH11357.

***Keywords*—fusion, integrated modeling, equilibrium reconstruction, high performance computing**

I. Introduction

The DIII-D National Fusion Facility [1] is a US DOE Office of Science User Facility. It is the principal fusion energy scale experiment in the USA, with over 800 national and international collaborators from 140 institutions worldwide. The DIII-D facility is renowned for its flexibility (control capability and range of actuators) together with its high-fidelity diagnostic instruments for assessing the plasma state and validating theoretical models.

Scientists at DIII-D study plasmas confined by magnetic fields in a toroidal (tokamak) configuration. The plasmas at DIII-D are sufficiently well confined that heating the plasmas with external sources can routinely achieve fusion relevant conditions in the plasma core. The DIII-D facility, in concert with similar scale facilities worldwide, is building the physics and technology basis for future fusion reactors, such as ITER [2].

979-8-3503-5554-3/24/$31.00 ©2024 IEEE
DOI 10.1109/SCW63240.2024.00265

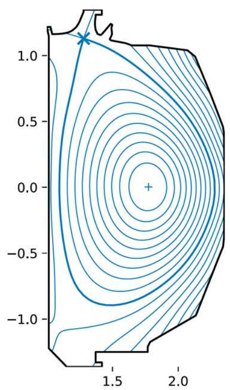

Fig. 1. A poloidal cross section of DIII-D showing the magnetic equilibrium reconstruction for pulse 174082 at 2.3 seconds. The contours represent values of constant magnetic poloidal flux, normalized from 0 at the + to 1 at the bold blue contour. The x-axis is major radius and the y-axis is elevation in meters in real space. The reconsruction provides the mapping from real space to flux space and important information about where the plasma interacts most strongly with the limiting solid surface (the black outline).

Experiments at DIII-D consist of programming actuators (magnets and heating systems) to produce high-performance well-controlled plasma conditions. Plasma pulses of 5 to 10 seconds duration are typically produced in DIII-D, with a typical interval between plasma pulses of 10-20 minutes. Rapid high-quality interpretive analysis of the collected data (typically 50 GB per pulse on DIII-D) enables informed decisions to be made in time for the next pulse.

A key step towards scientific understanding of a tokamak plasma is state reconstruction. This reconstruction of magnetically confined plasmas, often called equilibrium reconstruction [3-7], forms the foundation upon which many properties of the plasma can be understood. An example reconstruction is given in Fig. 1 The lowest fidelity analysis uses only magnetic measurements obtained around the outside of the plasma to constrain the reconstruction. Adding kinetic constraints (such as temperature and density profiles measured internal to the plasma) provides the highest fidelity plasma state reconstructions but is computationally time consuming and involves human-in-the-loop analysis. Developing very rapid (minutes vs hours) automated high-fidelity kinetic and magnetically constrained reconstructions in the control room would dramatically improve the quality of information available to inform operational decisions.

The automated orchestration of data retrieval, analysis, and code execution for these kinetically constrained equilibria has been achieved through the CAKE (Consistent Automatic Kinetic Equilibrium) module [8-9] of the OMFIT (One Modeling Framework for Integrated Tasks) integrated modeling framework [10-12]. Seeking to make these higher fidelity reconstructions available during operation of the DIII-D National Fusion Facility [1], we have leveraged the capabilities of the NERSC [13] Superfacility [14] project to expedite the availability of these reconstructions. In addition to the high level of parallelism possible on Perlmutter at NERSC, focused attention on the computational efficiency of each step of the analysis process and OMFIT workflow (described in Section II) enabled dramatic increases in the rate and absolute quantity of high-fidelity reconstructions. For a benchmark case, the CAKE workflow took 62 minutes wall-clock time on the Iris computing cluster at DIII-D; after optimization, the benchmark case took only 11 minutes on the Perlmutter system at NERSC.

The Iris cluster at DIII-D consists of 4 login nodes with 16 CPU cores per node and 30 worker nodes with 16 CPU cores per node, of which each user can only request 16 CPU cores for jobs that take less than 1 day. Thus running CAKE on Iris was limited to running on 16 CPUs. In contrast, the Perlmutter machine at NERSC has 30 login nodes with 64 CPU cores per node and 3,072 CPU-only worker nodes with 64 CPU cores per node (and 1792 GPU accelerated nodes that CAKE does not use). The workflows, timings, and optimizations are further described in Section III.

There are ~50 automated data analysis workflows that execute on the local Iris cluster after each DIII-D pulse. In 2018, using only low-fidelity magnetic reconstructions produced on the Iris cluster, a collaboration was initiated with ALCF [15] to automate the evaluation of "vacuum magnetic islands" in the plasma induced by 3D magnetic fields in DIII-D [16]. This analysis required computationally expensive field line tracing in the plasma geometry, enabled by ALCF resources [15]. The effort provided many insights into what logistics were needed for remote Leadership Class Computing to support DIII-D operations. However, the effort proved too labor intensive at the time, before the advent of the Superfacility and IRI APIs and real-time queues. In the subsequent years, significant progress has been made in developing the Superfacility and more recently IRI capabilities, triggering the work in this paper for automated high-fidelity reconstructions. These developments allowed the present effort to automate the CAKE workflow on Perlmutter at NERSC to support DIII-D operations, starting in April 2023. The connections from DIII-D to NERSC utilize the DOE high speed ESnet network. During the 2023 DIII-D experimental campaign, almost 20,000 time slice reconstructions were produced, with almost 10,000 of those being highly converged, in contrast to ~4000 manually produced reconstructions in the past 35 years of DIII-D operation.

Having the high-fidelity CAKE equilibria provide a foundation for evaluating computationally expensive models to inform on important physics properties of the plasma. For instance, the high-fidelity reconstructions enable the determination of the properties of turbulence and transport in the plasma by running gyro-kinetic models [17-18] on leadership class computing resources. It would be computationally wasteful and uninformative to evaluate models of the turbulence on the lower fidelity equilibria, however the rapid evaluation of higher fidelity CAKE equilibria opens up the analysis pipeline for high-fidelity transport and stability calculations that lend important insights into the properties of the plasma.

In section II, some improvements to the OMFIT framework are described. The description and optimization of running the CAKE workflow within OMFIT are given in Section III. Some turbulence transport results utilizing the CAKE equilibria are given in Section IV. The future outlook for using leadership class computing facilities to support DIII-D operations and research is given in Section V.

II. OMFIT

The OMFIT framework was born out of a desire to efficiently couple experimental data to fusion relevant codes to interpret the plasma state and its dynamics, and to make predictions for future experiments. The core framework is not specific to fusion; it can loosely couple the data and codes of any research discipline, but a vast majority of the applications of the framework have been in the fusion realm.

The OMFIT framework in its initial form is described in detail in [11]. The CAKE module of OMFIT takes advantage of many of those original features. For instance, the CAKE module runs codes external to OMFIT whose inputs are Fortran namelists, which OMFIT can manipulate. CAKE also uses features of OMFIT that were later developed to better interface with HPC centers and handle time dependent analyses. Finally, OMFIT was improved to provide profiling, dependency package management, automated testing, and usage statistics. These features and improvements of OMFIT will be described in the following subsections.

A. Improved namelist parsing

As part of the optimization effort here, it was recognized that namelist loading in OMFIT could represent a substantial part of the execution time. For a benchmark case, the parsing was slightly reworked to produce a 20% faster loading time.

B. HPC resource interfaces

As OMFIT began to run more complicated analyses that took more compute resources, interfaces to the PBS and SLURM schedulers were developed. The interfaces could track jobs to completion while still providing the standard out and standard error streams from the jobs to the user. An interface was also developed to provide the queue/partition that satisfies the CPU, memory, and wall clock time requirements of a particular process.

C. Multiple time slices

The initial OMFIT modules focused on single time slice analyses. Over time, modules were modified and new ones created that allowed multiple time slice analysis. There were two features added to OMFIT to facilitate these multi time slice modules: an interface to the job arrays feature of the Portal Batch System (PBS) or Simple Linux Utility for Resource Management (SLURM) schedulers and the OMFITcollection class. The job arrays are useful for generalizing a single time slice execution to multiple time slices. The OMFITcollections are useful for collecting objects of the same type, such as exist for each time slice. As the number of time slices under analysis became larger, the successive loading of all subcomponents of an OMFITcollection took more time. To speed up the loading, the ability to load the subcomponents in parallel was implemented.

D. Profiling

The OMFITpythonScript class was instrumented to measure the elapsed time and memory change for running the script. The hierarchy of scripts calling each other was tracked with their times and memories to understand bottlenecks across a workflow.

The OMFIT python scripts can also be invoked in a parallel manner, where the script is executed once per set of parameters. Internally OMFIT uses the python `multiprocessing.Process` method of spawning these parallel processes, where each process gets its own copy of the OMFIT session memory. While this was reasonable for smaller memory footprint OMFIT sessions, this has become a bottleneck in larger memory footprint OMFIT sessions, such as the CAKE workflow invokes. More details on this are provided in Section III.

E. Package management and testing

The OMFIT framework and modules make use of a variety of community packages, both python and otherwise, to form its computing environment. The mamba package manager is used to handle the required dependencies between the versions of these packages and the requirements of OMFIT. One benefit of mamba is that root/su privileges are not required and the interface is the same for any OS. OMFIT is used extensively on macOS and Linux, but it has only been somewhat ported to Windows Subsystem for Linux and has never been successfully ported to native Windows. The users and developers of the OMFIT framework rely on an extensive (but not exhaustive) set of tests to ensure that changes to the codebase or environment do not break past functionality. These tests are especially important for a complicated and interconnected workflow that CAKE requires.

F. Usage statistics

In order to understand the adoption and usage of OMFIT, certain actions are tracked including the opening of an OMFIT session in a GUI (but not command line sessions) and the loading of an OMFIT module in the GUI (but not command line sessions). From this tracking, we are aware of more than 220,000 OMFIT sessions across 1,550+ unique users since tracking began in 2016. There are a handful of core part-time developers; more than 150 users/developers across about 40 institutions have contributed to the OMFIT codebase. The flexibility of OMFIT and the breadth of fusion physics modules makes it a likely starting point for users to take advantage of HPC resources in the future.

III. CAKE

The physics motivation and methodologies for the CAKE workflow are given in [7]. The focus in this section will be on the mechanics of triggering the workflow automatically, the steps in the CAKE workflow, then efforts to profile and optimize the time to solution when executing on the Perlmutter system at NERSC, and finally the storage of results and logs.

A. Triggering

DIII-D uses MDSplus software [19-20] action nodes to trigger the execution of various between-pulse analysis codes. A new action node was created for the Superfacility CAKE workflow, which is triggered by the completion of the charge exchange recombination spectroscopy analysis (CER), called CERQUICK. This analysis is performed on the DIII-D Iris

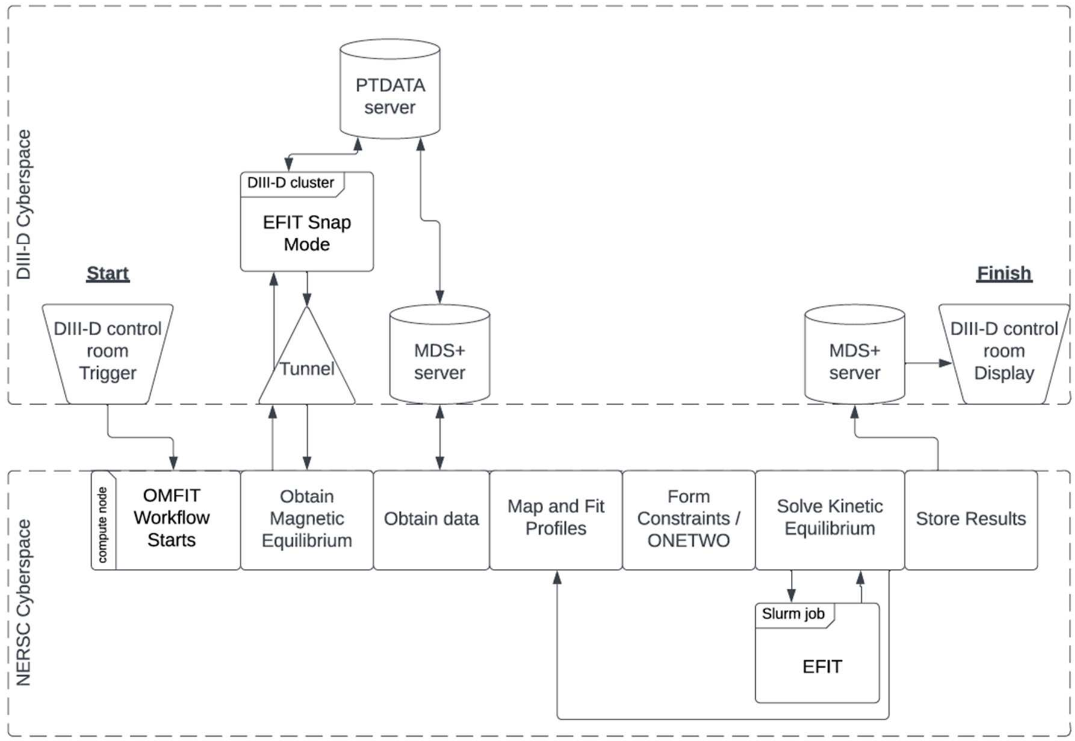

Fig. 2. A high-level outline of the steps in the OMFIT/CAKE workflow and cyberspace in which they are run.

system, however efforts are underway to accelerate the instrument analysis using the DOE-ASCR Integrated Research Infrastructure. The action node spawns a python script that communicates with NERSC via NERSC's Superfacility API. The API allows a new process to be spawned at NERSC via a trusted certificate that must be renewed every month. The python script also records a code run in the Data Analysis Monitor (DAM) system [21] at the start and end of the script. The NERSC processes were first spawned on Cori and then were migrated to Perlmutter. A service account was established at NERSC and at DIII-D to handle these interactions and the remote runs. The establishment of the service accounts represents a shared trust between NERSC and DIII-D and can be used to facilitate future workflows.

B. Running

The CAKE workflow, originally developed and run on DIII-D's Iris cluster, has been migrated to the Cori and

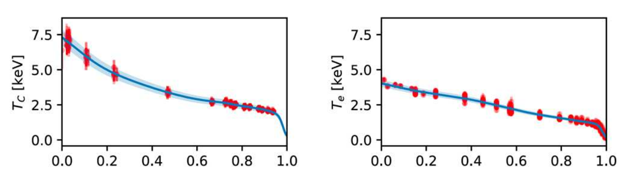

Fig. 3. The experimental carbon and electron temperatures (T_C and T_e), for DIII-D pulse 174082 at 2.3 seconds mapped to the normalized flux surface label. The red points are measurements; the blue cuves are fitted curves.

Perlmutter machines at NERSC. The Superfacility workflow is shown graphically in Fig. 2. Once triggered, the OMFIT/CAKE process determines the valid time slices, filters the data, maps the data to flux space, performs profile fitting (temperatures and densities), calculates constraints (including fast ion calculations with ONETWO [22-23] as well as bootstrap current calculations), and generates kinetic plasma state reconstructions using EFIT [1-5]. In the DIII-D tokamak, neutral beams of deuterium ionize in the plasma to form a cloud of high energy ions, and the partial pressure of these ions must be calculated with ONETWO as it is not measured. During this workflow, the OMFIT/CAKE process fetches data from the MDSplus server at DIII-D at several points during the workflow as needed, mostly concentrated at the beginning of the workflow. These fetched data include diagnostic data (Thomson Scattering and CER) used to generate kinetic profiles, initial magnetics only plasma state reconstruction data used for initial profile mapping, neutral beam data used for fast ion modeling, and others. An example of mapped data with its associated profile fit is given in Fig. 3 Additionally, the initial generation of inputs to the EFIT code are performed on DIII-D systems, but otherwise all calculations and modeling are performed on NERSC computers. An additional iteration of profile fitting, constraint modeling, and EFIT runs are performed using the resulting reconstruction of the previous iteration. The additional iteration is done to reduce the effects of using the non-kinetic plasma state reconstruction to map the diagnostic measurement locations. Finally, there is the option in the workflow to refine the equilibrium reconstruction by varying EFIT's numerical parameters (knot locations) so that more reconstructions reach a low numerical convergence error (1e-8) desirable for MHD modeling purposes. No additional data fetching from DIII-D is performed in the second iteration. After the workflow completion, the results, including the kinetic profiles as well as the reconstructions, are uploaded to DIII-D's MDSplus servers.

In its current state (reported as the fastest execution time in the next section), the OMFIT process is started on a single node at NERSC in its realtime partition. Typically, 50 timeslices are requested for each shot, and a lesser number of reconstructions are produced depending on data availability for that slice. When the workflow executes the EFIT portions without knot optimization, it uses a SLURM job array, with one processor per timeslice. When knot optimization is activated, it uses SLURM job arrays with 10 processors per timeslice. The ONETWO step is run on the same node as OMFIT, but using OMFIT's parallel script capabilities to run all time slices in parallel (subject to memory constraints, see section III.D). For calculations within libraries that OMFIT calls, OMP_NUM_THREADS is set to 64.

This workflow has been automatically performed for DIII-D pulses since pulse #194226, generating 19,358 valid times slices. The automatic runs originally did not include the additional EFIT numerical error reduction scheme due to runtime constraints. The distribution of convergence errors can be seen in Fig. 4. A pleasant surprise was that there were so many time slices that obtained such low convergence error (<1e-8) without needing the additional knot optimization step.

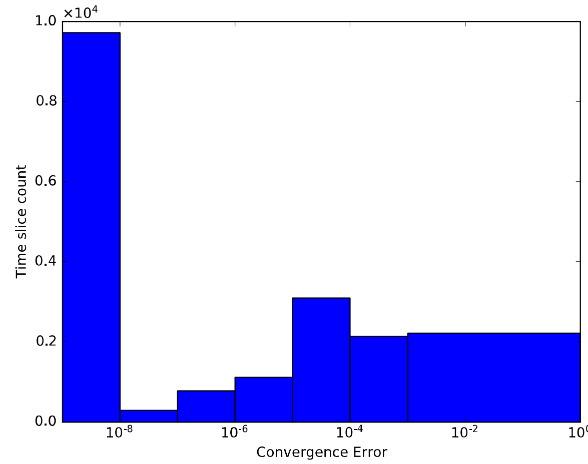

Fig. 4. The histogram of EFIT convergence errors for the successful CAKE run timeslices, which did not include knot variations in EFIT.

C. Optimizing for Speed

It is desirable to make the CAKE workflow as fast as possible, with the hope of providing a between pulse kinetically constrained plasma state reconstruction in time to inform decision making for the next pulse. In addition, the NERSC realtime queues have an hour limit, so any CAKE runs that did not complete in that time would be stopped by the NERSC scheduler. On the DIII-D Iris cluster the CAKE workflow could take several hours with knot optimization, which motivated the move to NERSC.

As previously mentioned in Section II, OMFIT has the ability to report the timing and memory of the scripts executed in a workflow. The CAKE workflow executes over 400 OMFIT scripts up to 10 layers deep. This reporting provided information to narrow in on those parts that were taking the longest time to try to speed them up, as well as pointing out obvious duplications of scripts being run. Table I summarizes the timing of the substeps of the CAKE workflow for a given reference case exercising different evolutions of the workflow on various machines, where the second EFIT reconstruction does not vary the EFIT numerical parameters to try to lower the convergence error. The timing does not include the time it takes to start up the OMFIT framework and import the CAKE module, which takes about 30 seconds.

Note that in the initial timing on Perlmutter there were unexpected slowdowns due to the unstable filesystem on Perlmutter, as well as having to deal with a new SLURM CG status. As these were overcome, the Perlmutter final timings were better than any Cori or Iris timings.

TABLE I. OMFIT CAKE WORKFLOW SUBCOMPONENT TIMINGS FOR VARIOUS EVOLUTIONS OF THE WORKFLOW IN UNITS OF SECONDS.

CAKE Workflow Segment	Iris	Cori Initial	Cori Best	Perlmutter Initial	Perlmutter Best
1_gather	154	189	52	53	32
2_fit	371	435	89	255	26
3_form_constraint (ONETWO)	675	690	878	789	262
4_make_k_files	130	120	89	76	41
5_run_efit_with_kinetic_constraint	380	224	87	348	52
2nd iteration					
1_gather	231	78	74	50	42
2_fit	362	398	86	223	26
3_form_constraint (ONETWO)	554	564	366	636	62
4_make_k_files	116	82	12	10	7
5_run_efit_with_kinetic_constraint	640	270	154	193	51
6_make_outputs	106	100	56	73	25
----	----	----	----	----	----
Total	3732	3164	1951	2713	630

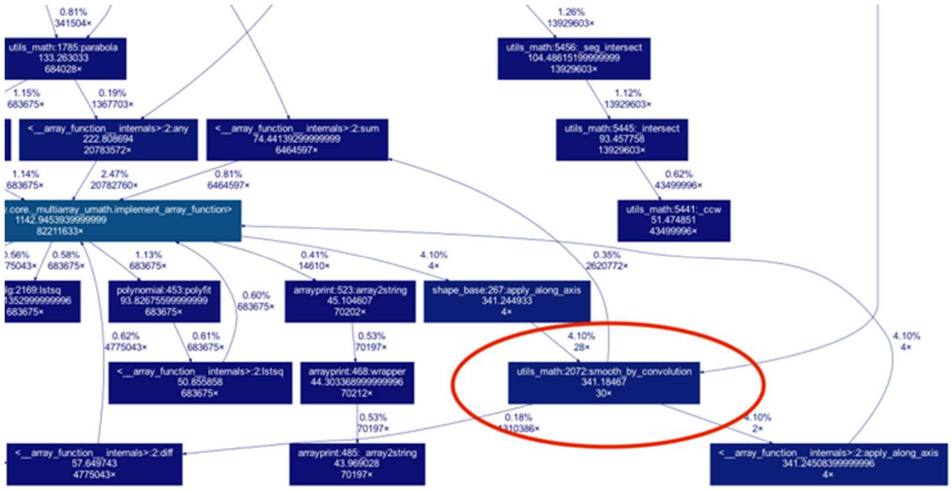

Fig. 5. A portion of a graph generated by analyzing the results of the python cprofile package during the execution of the OMFIT CAKE workflow. Each box shows a function call, the file it is defined in, the amount of time it took during the overall execution, and the number of times it was called. The smooth_by_convolution function call is highlighted as taking a substantial amount of time. Later refactoring led to this function taking much less time in the overall workflow.

In addition to the OMFIT script profiling, the python cprofile package was used to understand specific parts of the OMFIT framework itself that were taking a substantial amount of time. The results were visualized with the gprof2dot tool, and an example subset of the visualization is presented in Fig. 5. This separate profiling was useful for honing in on which framework level functions were being called excessively or taking a larger amount of time.

D. Understanding Memory Consumption

Because of the trivial parallelization across time slices for the CAKE workflow, some of the CAKE components' scripts are run using the parallel capabilities of OMFIT scripts. As previously mentioned in Section II, doing this performs a complete copy of the OMFIT session memory for each parallel process. Given that there are up to 100 timeslices, the memory consumption of the main OMFIT process suddenly became a bottleneck for increased throughput. The total memory consumption of OMFIT as a function of time during the CAKE workflow for a benchmark case is shown in Fig. 6. In this case, the level of parallelism was limited to 64 simultaneous slices. In attempting to execute all slices in parallel, the main OMFIT process could request more memory than available on a node, causing that session to crash. The capability was introduced into OMFIT to detect how much memory was allocated for the overall process, and then only fork as many parallel processes as

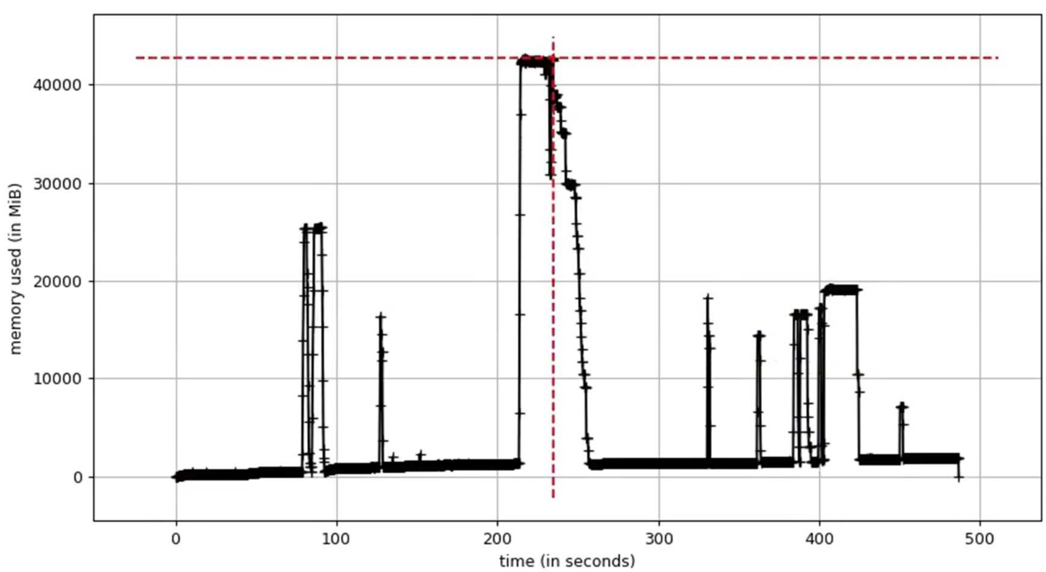

Fig. 6. Total memory consumption as a function of wall clock time during the OMFIT CAKE workflow.

would fit into the allocation. The longer duration spikes at 80, 220, and 420 seconds in Fig. 6 happen while OMFIT is executing computationally intensive steps. The shorter spikes at other times occur when the OMFITcollection elements are loaded in parallel.

E. Storage and Relational database

After the final EFIT iteration, the EFIT results are moved back to the DIII-D Iris cluster, and the results are uploaded to MDSplus using the already established EFIT data loading routine. The fitted profiles are uploaded to MDSplus directly from Perlmutter. The plasma state reconstruction and profiles are also converted to the omas [24] format for use in follow-on workflows. New tables were added to the relational database at DIII-D that track analyses per code run per pulse to be able to tie together the equilibria with the profiles and other later analyses. Output files of various types (g/a/m/k-files, statefiles, hdf5 omas files) are stored on disk at NERSC for future use. Finally, the OMFIT session is saved as a zipped OMFIT project for later inspection. The stdout and stderr logs are also saved to the DIII-D s3 object store and referenced in the DIII-D data analysis monitor (DAM) for later consumption.

One feature of OMFIT is the ability to use the carriage return without newline to produce ascii progress bars for various processes. While these are desirable for interactive sessions, they are less useful for automated workflows. In fact, when the stdout is redirected to a file instead of a terminal, the progress bars are output sequentially instead of overwriting the previous stage; this led to 10,000+ lines of output in the logs that provide little benefit. A recent change to OMFIT allows the suppression of these progress bars to avoid the excessive reporting in the logs.

IV. Transport Follow-On Analysis

A common analysis of experimental data involves comparing measured temperature, density, and rotation profiles to predictions by flux-driven transport solvers that run TGLF[17], linear CGYRO [18], or TGLF-NN [25] as their turbulent transport models. A comparison of the fluxes predicted by TGLF and a quasilinear combination of linear CGYRO runs using inputs from a Superfacility CAKE run is given in Fig. 7. The linear CGYRO simulations are also useful to determine the type of micro-instability that limits the profile at a given radial location in the plasma (see Fig. 8), and the coloring of the points in Figs. 7 and 8 indicates the dominant type of instability in the plasma. From the knowledge of the dominant type of instability, the experimentalist can know what actuators to employ in order to increase the plasma performance (e.g., stabilize ITG and generate an internal transport barrier by increasing the Shafranov shift and local density gradient).

This dominant mode characterization can be parallelized across wavenumber, radius, and time. A typical run contains 21 wavenumbers and 7 radii. Evaluating the stability at each wavenumber takes between 2 minutes, if unstable, and 2 hours, if stable. For a representative case it took 15 minutes per radius per time slice utilizing all the cores of twenty one Perlmutter CPU nodes. Increasing the core count per wavenumber did not yield improvements to timing because of the slowdown of internode communications.

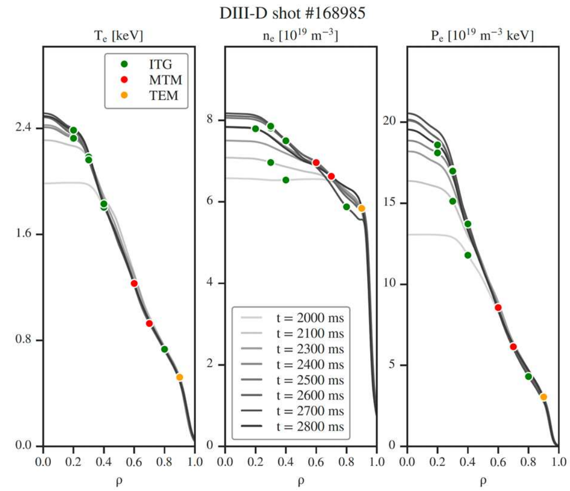

Fig. 7. The experimental electron temperature, electron density, and electron pressure (smoothed curves fit to discrete points of those measurements) for the indicated times of DIII-D pulse 168985. The coloring of points indicate the dominant type of turbulence instability at that radius.

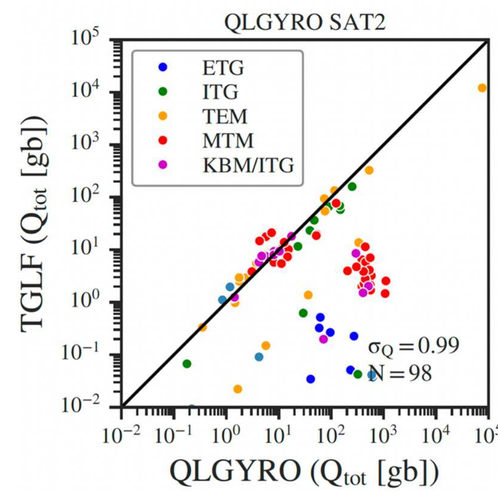

Fig. 8. A comparison of TGLF predicted fluxes to quasilinear CGYRO predicted fluxes for inputs from a Superfacility CAKE run.

V. Future Outlook

The possibility of executing CAKE in a timely manner during DIII-D experiments has excited the pursuit of other automated follow-on analyses, including global MHD stability, pedestal MHD stability, profile predictions, diagnostic aiming, and actuator impacts. To reliably meet the demand for CAKE analyses, the HPC center where CAKE is executed must have 100% availability during DIII-D operations. Since every HPC center has some downtime, we are attempting to expand the HPC centers where CAKE can run, starting with the ALCF Polaris system [15]. The NERSC Superfacility API does not exist at Polaris. However, both NERSC and ALCF have Globus compute capabilities. Also, DIII-D has Globus computing capabilities coming online, so Globus computing is being pursued as the means of launching jobs at either cluster (whichever is operational as determined by Globus compute capabilities).

The CAKE runs in and of themselves are forming a valuable database of high-fidelity equilbrium reconstructions. Besides the between-pulse analyses, the CAKE workflow is starting to be run for historical pulses. These CAKE results are being used to generate a machine learning model of high-fidelity reconstructions. They also form the foundation for broader database validation of models.

Diversifying the HPC centers used poises DIII-D well within the Integrated Research Infrastructure (IRI) effort of the US DOE. As one of three IRI Pathfinder projects, DIII-D is seeking to make improved use of HPC resources. In addition to between pulse analyses, DIII-D is preparing to predict from first principles before each plasma pulse what the pulse will do. The goal of these predictions is to improve operational efficiency, protect the machine against adverse conditions, and validate the predictive models.

The current CAKE workflow does not provide uncertainty quantification on its reconstructions. An effort is underway to develop an Integrated Data Analysis (IDA) framework which makes use of a Bayesian approach to obtain plasma state and profile reconstructions with uncertainties [26]. In addition to the uncertainties provided, this workflow is tightly coupled and would thus avoid the overhead time penalty of passing information via disk that the OMFIT approach incurs.

Beyond DIII-D, there are a number of tokamaks throughout the world (both existing and planned, such as ITER) that could benefit from the off-site analyses and computing resources discussed in this paper. Collaborations will be pursued that will bring this IRI concept to the international stage including the usage of National HPC centers to support ITER operation.

Acknowledgements

This research used resources of NERSC, a Department of Energy Office of Science User Facility using NERSC award FES-ERCAP0029284.

Disclaimer

References

[1] Luxon, J. L. (2005). A Brief Introduction to the DIII-D Tokamak. *Fusion Science and Technology*, *48*(2), 828–833. https://doi.org/10.13182/FST05-A1042

[2] https://iter.org

[3] Lao, L. L.; John, H. S.; Stambaugh, R.; Kellman, A. & Pfeiffer, W.; Reconstruction of current profile parameters and plasma shapes in tokamaks; Nuclear Fusion, 1985, 25, 1611

[4] Lao, L. L., et al. ; Separation of beta_p and li in tokamaks of non-circular cross-section; Nuclear Fusion 25.10 (1985): 1421

[5] Lao, L. L., Ferron, J.R., Groebner, R.J., Howl, W., St. John, Strait E.J., Taylor, T.S. Equilibrium Analysis of Current Profiles in tokamaks. Nuclear Fusion, 30 , 1035 (1990)

[6] Lao, L. L., St. John, H. E., Peng, Q., Ferron, J. R., Strait, E. J., Taylor, T. S., Meyer, W. H., Zhang, C., & You, K. I. MHD equilibrium reconstruction in the DIII-D tokamak. Fusion Science and Technology 48, 968 (2005)

[7] Lao, L. L., Kruger, S., Akcay, C., Balaprakash, P., Bechtel, T. A., Howell, E., Koo, J., Leddy, J., Leinhauser, M., Liu, Y. Q., Madireddy, S., McClenaghan, J., Orozco, D., Pankin, A., Schissel, D., Smith, S., Sun, X., and Williams S., Application of machine learning and artificial intelligence to extend EFIT equilibrium reconstruction. Plasma Physics and Controlled Fusion, 64(7):074001, 2022,

[8] https://omfit.io/modules/mod_CAKE.html#mod-cake

[9] Z. A. Xing et al., "CAKE: Consistent Automatic Kinetic Equilbrium reconstruction", Fusion Engineering and Design, 163 (2021); doi: 10.1016/j.fusengdes.2020.112163

[10] https://omfit.io/

[11] O. Meneghini, S.P. Smith, L.L. Lao, O. Izacard, Q. Ren, J.M. Park, J. Candy, Z. Wang, C.J. Luna, V.A. Izzo, B.A. Grierson, P.B. Snyder, C. Holland, J. Penna, G. Lu, P. Raum, A. McCubbin, D.M. Orlov, E.A. Belli, N.M. Ferraro, R. Prater, T.H. Osborne, A.D. Turnbull, G.M. Staebler, and the AToM Team [Integrated Modeling Applications of Tokamak Experiments with OMFIT] Nuclear Fusion, 55, 083008 (2015)

[12] O. Meneghini; L. Lao, [Integrated Modeling of Tokamak Experiments with OMFIT] Plasma Fusion Res. 8, 2403009 (2013)

[13] https://nersc.gov

[14] "Superfacility: The Convergence of Data, Compute, Networking, Analytics, and Software," Book chapter in Handbook on Big Data and Machine Learning in the Physical Sciences, pp. 361-386 (2020)

[15] https://www.alcf.anl.gov/

[16] M. Kostuk, T. D. Uram, T. Evans, D. M. Orlov, M. E. Papka & D. Schissel (2018) Automatic Between-Pulse Analysis of DIII-D Experimental Data Performed Remotely on a Supercomputer at Argonne Leadership Computing Facility, Fusion Science and Technology, 74:1-2, 135-143, DOI: 10.1080/15361055.2017.1390388

[17] https://gacode.io/tglf.html#id9 and references therein

[18] https://gacode.io/cgyro.html#past-and-future and references therein

[19] https://www.mdsplus.org/

[20] https://github.com/MDSplus/mdsplus

[21] S. Flanagan, et al., System to monitor data analyses and results of physics data validation between pulses at DIII-D, Rev. Sci. Instrum., Volume 75, Issue 6, (2004) 2043.

[22] Pfeiffer, W W and Davidson, R H and Miller, R L and Waltz, R E, General Atomics Technical Report, GA-A16178 (1980)

[23] St. John, H and Taylor, T S and Lin-Liu, Y R and Turnbull, A D, Plasma Physics and Controlled Nuclear Fusion Research, vol. 3, p. 603 (1994)

[24] https://gafusion.github.io/omas/

[25] O. Meneghini, S. Smith et al., Self-consistent core-pedestal transport simulations with neural network accelerated models, Nucl. Fusion, 57, 0866034 (2017)

[26] S. S. Denk, T. B. Amara, T. Odstrcil, L. Stagner, S. P. Smith, and R. Nazikian, "Fast inference of plasma profiles and the magnetic equilibrium in preparation of in-between-shot Integrated Data Analysis (IDA)", [draft, to be submitted]

APPENDIX: ARTIFACT DESCRIPTION/ARTIFACT EVALUATION

A. Artifact Description

This paper makes use of the OMFIT framework, the ONETWO code, and the EFIT code, as well as data from the DIII-D National Fusion Facility. The codes were run on the Perlmutter system at NERSC. User agreements are required for access to each of these, as none are completely public.

1) OMFIT

The instructions for gaining access to the OMFIT framework are at https://omfit.io/install.html#get-access-to-the-omfit-source-code . Instructions for installing OMFIT are at https://omfit.io/anaconda.html#anaconda .

2) EFIT

The instructions for gaining access to the EFIT code are at https://efit-ai.gitlab.io/efit/license.html#users-agreement . The EFIT installation instructions are at https://efit-ai.gitlab.io/efit/install.html .

3) ONETWO

Instructions for access to the ONETWO code are not public. If indicated in the request for access to OMFIT, access will be granted to the ONETWO codebase.

4) DIII-D

For becoming a DIII-D User, which grants access to all DIII-D data, follow the instructions at https://d3dfusion.org/become-a-user/

The scripts for running the CAKE workflow and interfacing with the NERSC Superfacility API are in the private repository https://github.com/DIII-D/IRI_D3D to which access can be granted after becoming a DIII-D User.

5) NERSC project

A NERSC account can be acquired at https://iris.nersc.gov/add-user . The “realtime” partition of Perlmutter was used for the timings reported in this paper.

B. Reproducibility of Experiments

The basic CAKE run is obtained with something like

```
omfit                                          \
modules/CAKE/SCRIPTS/between_shot_autorun.py   \
"shot=174082 t_start=1000 t_end=6001 dt_sp=50 \
n_parallel=64         use_realtime=True        \
code='1FWDMnrS'"
```

The memory profiling is obtained with

```
OPENBLAS_NUM_THREADS=64 OPENMP_NUM_THREADS=64 \
mprof run-include-children-python python \
omfit/omfit.py \
modules/CAKE/SCRIPTS/between_shot_autorun.py \
"shot=174082 t_start=1000 t_end=6001 dt_sp=50 \
n_parallel=64 use_realtime=True \
code='1FWDMnrS'"

mprof plot
```

The CPU profiling (cProfile) is done with code from https://github.com/DIII-D/IRI_D3D/issues/32 then plotted with

```
gprof2dot -f cake_profiling.stats -T png \
-o output.png
```