Visualization and Analysis for Near-Real-Time Decision Making in Distributed Workflows

David Pugmire
Oak Ridge National Laboratory

James Kress
University of Oregon &
Oak Ridge National Laboratory

Jong Choi, Scott Klasky
Oak Ridge National Laboratory

Tahsin Kurc
Stony Brook University &
Oak Ridge National Laboratory

Randy Michael Churchill
Princeton Plasma Physics Laboratory

Matthew Wolf, Greg Eisenhauer
Georgia Institute of Technology

Hank Childs
University of Oregon &
Lawrence Berkeley National Laboratory

Kesheng Wu, Alexander Sim, Junmin Gu
Lawrence Berkeley National Laboratory

Jonathan Low
A*STAR Computational Resource Centre

***Abstract*—Data driven science is becoming increasingly more common, complex, and is placing tremendous stresses on visualization and analysis frameworks. Data sources producing 10GB per second (and more) are becoming increasingly commonplace in both simulation, sensor and experimental sciences. These data sources, which are often distributed around the world, must be analyzed by teams of scientists that are also distributed. Enabling scientists to view, query and interact with such large volumes of data in near-real-time requires a rich fusion of visualization and analysis techniques, middleware and workflow systems. This paper discusses initial research into visualization and analysis of distributed data workflows that enables scientists to make near-real-time decisions of large volumes of time varying data.**

I. Introduction

Data driven sciences are placing enormous stresses on existing visualization and analysis frameworks. These stresses are occurring across several different axes.

First, increasingly larger volumes of data are being generated, and at increasing frequency as well. For example, in the test case that we are exploring, large scale parallel plasma fusion simulation codes, 10+ GB of data are being generated each second. Similar amounts and frequency of data are generated from sensors inside fusion experiments that are in operation already, and will only increase when large experiments, such as ITER [1], become fully operational.

Second, scientists and data are often distributed in multiple geographic locations. This requires that either the scientists be moved closer to the data, which is fraught with logistical and convenience difficulties, or move data (either all, or selected portions) to where the scientists are located.

The movement of data leads to additional stresses. It is often not possible to move the large volumes of data to remote locations such that scientists can interact, analyze and visualize it in a timely manner that will allow them to make critical decisions. Scientists need to be able to interact with their data in near-real-time in order to monitor and assess the evolution of the science, to determine when problems are arising, or when important phenomena are occurring. Only then can scientists make timely and informed decisions on what actions to take. Finally, the workflows (both simple and complex) required by scientists can require computational resources that may not be available at the remote sites where the scientists are located.

In this paper, we describe initial work from an active research effort to explore data coupling and near-real-time analysis and visualization between two geographically separated sites. In this instance, Singapore and Atlanta, Georgia, USA. Specifically, a plasma fusion simulation running at the A*STAR Computational Resource Centre in Singapore, and the visualization and analysis running at Georgia Tech in Atlanta. We demonstrated this working system at the SuperComputing 2015 Conference in Austin, Texas. At each timestep of the simulation running in Singapore, summary data are being generated and transferred to Georgia Tech where it would be displayed by visualization tools. The scientist is then able to select regions of interest and extract features from the summary data. Once the features are identified, a query is sent for particle data contained within the feature. These queried particles are extracted and transferred to to Georgia Tech for visualization.

In this work we are exploring fusion of a variety of different technologies. We are using a high-level API to provide location independent data access and remote reading and writing. The middleware components implement RDMA over wide-area networks and support data indexing for optimized filtering operations. The data analysis and visualization components use this middleware to facilitate rich interactions with the data. These components use the data subsetting and filtering operations of the middleware to achieve near-real-time interaction with the running simulation. These results show that near-real-time interaction can be achieved, even with the sites are separated by tens of thousands of miles. Further, we are able to show end-to-end data selection and visualization within the tight 10 second time constraint window imposed by the running simulation.

In the remainder of this paper we discuss related work, discuss some of the broader motivations for this work, provide a detailed discussion of the system and the results obtained to date. Finally we discuss areas of continuing and future research.

II. Related Work

The work that we present in this paper is a fusion of technologies and ideas that builds upon work in several key areas including simulation monitoring and steering, in situ visualization techniques, and past XGC1 visualizations.

Past work in the area of simulation monitoring and steering has focused a lot of effort into designing methods for quickly and efficiently visualizing data across a network. Some notable examples include Visapult [2], Visualization Dot Com [3], VisPortal [4], and a Real-Time Monitoring framework for large scientific simulations [5]. VisPortal and Visualization Dot Com build on the foundations of Visapult, and provide a remote distributed visualization framework for efficient visualization of remote simulation data. This framework uses both the local visualization client and the remote data client to perform parallel renderings, decreasing the time to produce the final visualizations. By leveraging Visapult, VisPortal and Visualization Dot Com are able to provide convenient access to simulation data to scientists through an easy to use and access online interface.

One notable example of work in simulation steering is SCIRun [6]. SCIRun presents a programming environment to simulation scientists and easily allows them to modify their simulations interactively as well as create automatically changing parameters based on boundary conditions.

Our work also builds on in situ processing paradigms. In this paradigm, data are processed *while* they are being produced. This is contrasted with a post hoc paradigm where data that was previously written to disk are read back into memory and processed. A wide range of research for in situ methods has occurred, as surveyed in [7]. Our particular focus is on *in transit* methods where data are transferred asynchronously over the network to the memory space of a set of data staging nodes. The ADIOS [8] middleware system, which uses an abstraction of I/O, has the ability to use a variety of transport methods for the movement of data, including in transit techniques. These transport methods include DataSpaces [9], which allows memory coupling between processes running on different sets of nodes, FlexPath [10], which supports a publish/subscribe interface for direct memory access, and ICEE [11] which supports RDMA transfers over wide area networks.

Past work on XGC1 visualization has looked at addressing the data needs of scientists during the course of a simulation run. One successful tool that was developed for XGC1 simulation monitoring was an online dashboard called eSimon [12]. This dashboard was launched with each simulation run, and was responsible for several different common visualization and analysis tasks in XGC1. First, the dashboard was responsible for creating and updating plots of approximately 150 different variables every 30 seconds and plotting 65 different planes for the live simulation. At the conclusion of a run the dashboard would automatically output movies of each of these plots of interest for quick review. In addition this dashboard catalogued simulation output allowing users to search for and retrieve data of interest, without having to locate and search through simulation output files. Finally, this dashboard was available to scientists anywhere in the world through their internet browsers, making the data quickly and readily available.

Following this effort, work shifted towards researching in transit visualization opportunities within XGC1, to take advantage of their use of the ADIOS middleware system. This research focused on the development of scalable visualization plugins that operate within data staging [13]. This work demonstrated the use of the EAVL [14] visualization library used in conjunction with ADIOS and DataSpaces to perform more intensive visualization operations than were possible in the dashboard environmentt.

Our work leverages the ideas from these past projects, and has allowed us to create a visualization and analysis pipeline that is extensible and operates on user driven subselections of live simulation data.

III. Objectives

Our objectives in this research are to explore data coupling and near-real-time analysis and visualization between timevarying producers of large data, and distributed data consumers. This capability for near-real-time access to data will help scientists observe, monitor, analyze the science as it happens, and enable them to make time-critical decisions.

We are working with the XGC1 [15] simulation code, a highly scalable physics code used to study plasmas in fusion tokamak devices. XGC1 is a multiphysics code, including many of the physics effects important for simulating a whole-device, including turbulence, strong-gradient regions, neutral particle effects, full magnetic geometry, and more. This gives the XGC1 simulations the advantage of accurately approximating real-life fusion devices, with the cost being added complexity in identifying and isolating salient features of the simulation.

XGC1 is a particle-in-cell (PIC) code, a common and important method for solving physics problems. As such, XGC1 represents a large class of many different simulation codes. XGC1, like other particle-in-cell codes uses a grid, or mesh, to represent a set of cells, and a large number (billions) of charged particles. At each timestep, each particle's state is updated according to the underlying physics equations, and then the particles are statistically deposited onto the cells within the grid in order to solve a field equation for the electrostatic potential, and also to calculate reduced moments of the particle distribution function (e.g. density, velocity, temperature, etc.). Scientists are interested in both the mesh quantities (potential and moments) and the particles.

In particular, the scientists working with XGC1 are interested in understanding the effects from and drivers of turbulence within the plasma, which substantially degrade plasma performance. A key to this understanding is the analysis and visualization of nonlinear, turbulent eddies in XGC1 simulations, including their 3D structure and the perturbation they cause to particle orbits.

Because of the large volumes of data generated by XGC1, it serves as an excellent test-case for our research. XGC1 simulations routinely generate several TBs of data, and larger runs, such as recent runs for ITER have produced 20 TB of data. Similarly, sensor networks and diagnostics attached to an

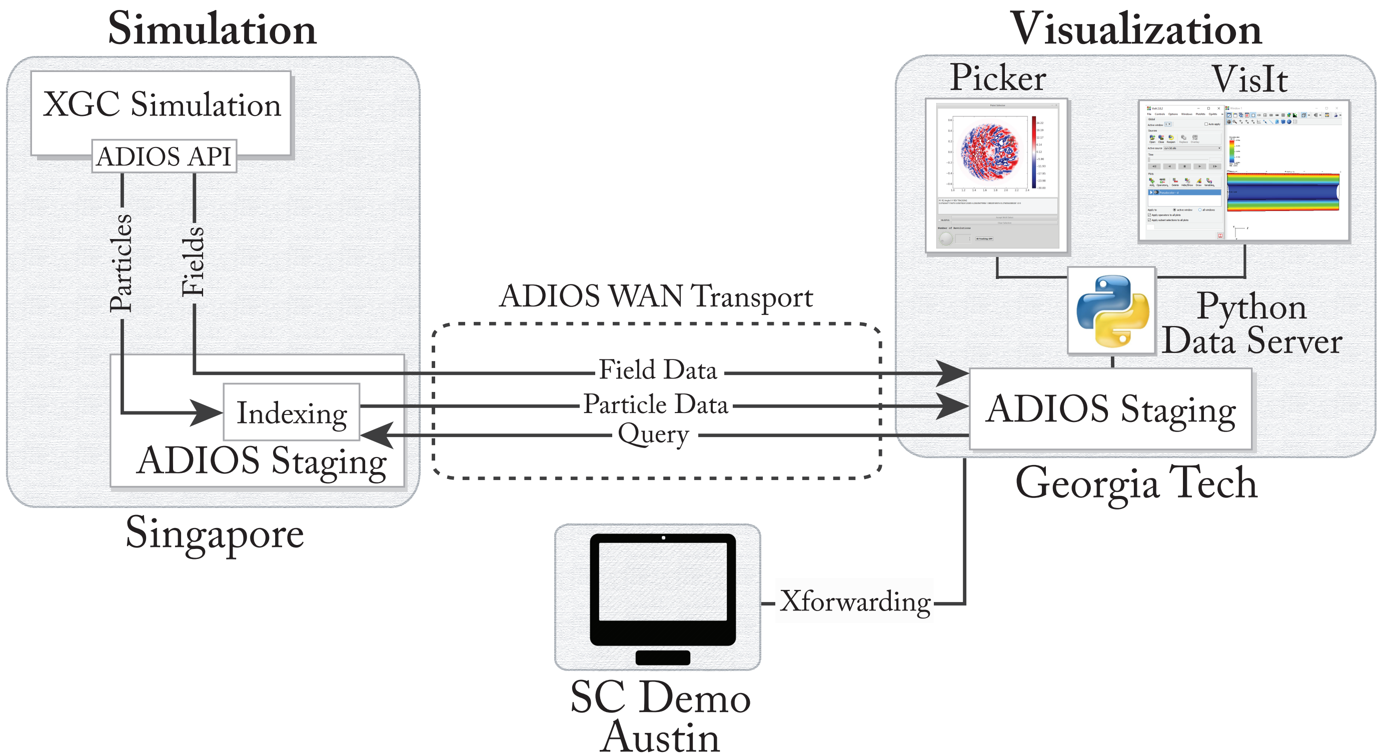

Fig. 1: The data flow pipeline for our workflow showing the distribution of the simulation and data querying in relation to the interactive visualization system.

experimental device like ITER are also expected to generate TBs of data. Simulations and experiments in other domains produce similar amounts of data. Data volume estimates for the Square Kilometer Array Radio Telescope are even larger: around a TB of data every second.

Simulation centers and experimental facilities are scarce, and very expensive resources, and scientists have only fixed windows of time to do their science. Simulations that go awry, or encounter run-time problems translate into real loss of time to do science and the costs associated with running the facility. Experimental facilities face an even bigger problem. For instance, in the case of the ITER reactor, the buildup of instabilities within the plasma could cause physical damage to the reactor vessel. This results in significant costs for repairs, and downtime where other experiments are not able to run. Additionally, all of the data generated by current simulations is difficult to save without incurring significant I/O overhead, reducing the compute cycles available for science computations.

Allowing the scientists to remotely monitor and track their simulations and experiments in near-real-time will allow them to make important decisions. These include:

- Aborting when the simulation or experiment appear to be in an error state, not converging, or not answering the anticipated questions being posed.
- Continue the simulation or experiment as it is progressing as expected.
- Identification of the most important datasets to save or analyze from a simulation before data are lost.
- Steering the simulation or experiment as the results for each timestep are observed and analyzed.

IV. System Implementation

The system we deployed was designed to provide a production-level environment to study near-real-time visualization and analysis for workflows distributed across very large distances. This system allowed us to integrate and study a variety of core components, including interactive visualization, feature detection, interactive querying of very large data, and the management of data movement over a wide area network connection.

The system, shown in Fig. 1, is spread across two different geographic locations, The A*STAR Computational Resource Centre in Singapore and Georgia Tech in Atlanta, Georgia. Two major components were running in Singapore on the A*STAR Ulam supercomputer; the XGC1 simulation code, and a data staging service. At each timestep of the simulation, output data from the simulation nodes are transferred over the interconnect to the data staging nodes where in-memory data processing operations are performed. Summary data for each timestep are transferred over the WAN to a data staging node at Georgia Tech. These summary data are displayed using interactive analysis and visualization tools. The scientist can then specify regions of interest in the visualization tools. Features in the region of interest from the summary data are

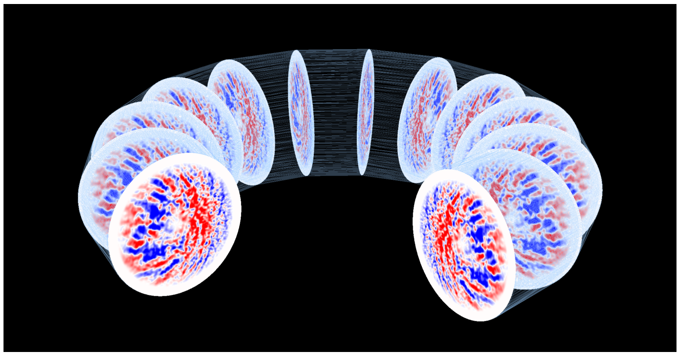

Fig. 2: An example of a mesh in XGC1. A number of planes are equally spaced around the central axis of the tokamak.

extracted and displayed. Additionally, the region of interest is communicated back to Singapore over the WAN where additional data contained in the region of interest are extracted and then transferred for visualization.

Our system consists of a synthesis of a number of different tools and frameworks. The visualization tools located at Georgia Tech include the large-scale parallel production tool VisIt [16], and a simple selection tool we developed. A detailed description of the visualization components are given in Section IV-B. The output from the XGC1 simulation, and data movement between Singapore and Georgia Tech is handled by the ADIOS [8] middleware with WAN support provided by several different transport methods. A brief description of the simulation running in Singapore is given in Section IV-A. Details about the data management and WAN transfer are given in Section IV-C.

A. XGC1 Simulation

XGC1 is a particle in cell (PIC) code. The simulation proceeds by computing the interactions of a very large number of particles, and then depositing the particles onto a finite element mesh. XGC1 uses a mesh illustrated in Fig. 2. A number of 2D planes are positioned uniformly around the torroidal shape of the tokamak. The particles interact within the torrodial space defined by this mesh. The particle deposition step provides important statisical information to the scientists, as well as helps increase the speed and efficiency of the simulation. The number of planes chosen for a simulation run is specified by the scientist in such a way to capture all of the waveform distributions that are expected for the particular run. The number of planes used is typically in the range of 32 and 64.

Because we were interested in the fine scale tracking of 3D turbulent eddies, it was determined that we needed many more planes than typically used. Reconstruction of 3D turbulent eddies from too few planes would lead a high degree of error, as eddies could split or die off in between planes that were too far apart. Because of this, we configured XGC1 to run with a total 512 planes. The variable of interest for computing the eddies is the derivative of the potential, called “dpot”. XGC1 already uses the ADIOS API for data output. In this particular configuration, we configure the ADIOS output to be directed to the data staging nodes, which is described in Section IV-C.

B. Interactive Visualization and Analysis

On the visualization and analysis side located at Georgia Tech, our system consists of three major components: VisIt [16] to allow for interactive visualization (though our pipeline is agnostic to the interactive visualization tool and others such as ParaView [17], EnSight [18], or FieldView [19] could have been employed), an eddy picker to allow the scientist to specify areas of interest, and an ADIOS staging service to manage the movement of data. A small python process is used to coordinate between the ADIOS staging service and the visualization tools, and perform some basic data processing. Python is used to coordinate communication between VisIt and eddy picking tool, and perform some basic data analysis. The eddy picking tool, shown in Fig. 3, was written from scratch using PyQt. The eddy picking tools displays a summary slice from the simulation, and is updated at each timestep from the simulation. It also allows the scientist to interactively select regions of interest that control the feature extraction, and the querying of particle data from XGC1.

Eddies in turbulent fusion plasmas are typically elliptical in shape, so the picking tool supports selecting an ellipse with 3 points: the center point, and two points that lie on the ellipse. From these three points the major and minor radii are computed, as well as the direction of the major axis. Additionally, a dial is provided to control how many revolutions around the torus are used for construction of the 3D eddies.

Selecting a region of interest triggers three things: (1)

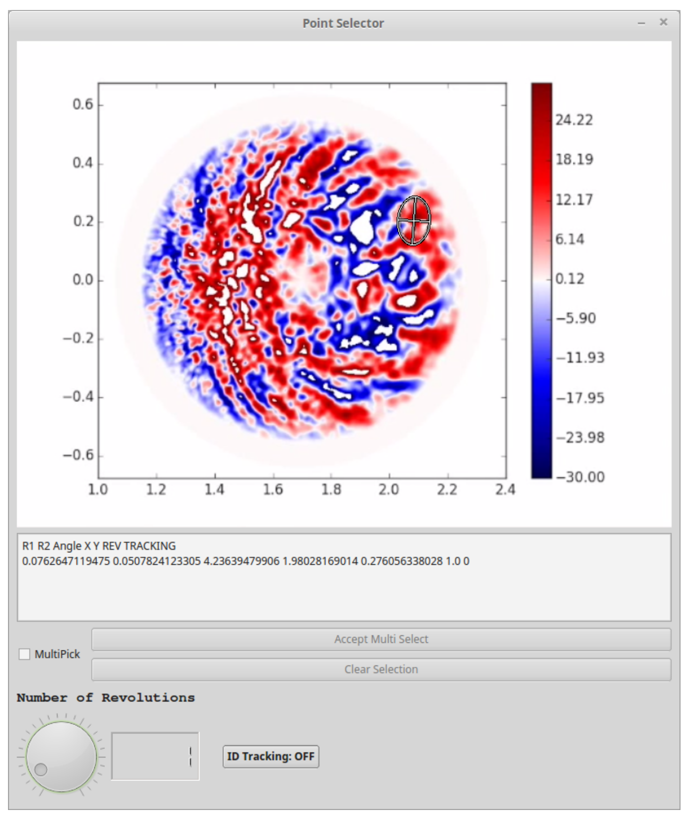

Fig. 3: The eddy selection interface demonstrating the selection of an eddy on a plane of the simulation data.

The magnetic field line at the center point of the ellipse is calculated and displayed. The magnetic field line can be seen in yellow in Fig. 4. (2) The 3D eddy is computed and displayed. Computing the 3D eddy is done by sequentially identifying 2D eddies at each plane in the simulation. A 3D representation of the eddy is obtained from the set of 2D eddies using a ruled surface. Determination of the 2D eddy in each plane is done by computing isocontours within the elliptical area of interest. The value used to construct the isocontours is the value at the pick point in the region of interest. In the case of multiple isocontours within the elliptical region, the isocontour closest to the magnetic field line is chosen, as eddies tend to follow magnetic field lines. The 3D eddy can be seen in orange in Fig. 4. (3) A bounding volume around the eddy in 3D space is calculated. This 3D bounding volume is created by placing a 2D bounding box around each of the isocountour features identified in the previous step, and then connecting the bounding box points in sequence between the planes. Once this 3D bounding volume is calculated, it is sent to Singapore using ADIOS. Once the Singapore side receives the 3D bounding volume information, it extracts the particles that lie within the bounding volume. The extracted particles are sent using ADIOS to Georgia Tech and visualized. The particles, along with the magnetic field line and 3D eddy are shown visualized together in Fig. 5.

The scientists expressed the need to track particles contained in an eddy at a particular time and observe their evolution over time. This capability is activated with a toggle on the picker tool. If "ID Tracking" is turned on, then the IDs of the particles contained in the 3D eddy are cached, and at every timestep these particles are sent from Singapore to Georgia Tech for visualization. With this option, the scientists can follow the evolution of the particle orbits over time, and study their relationship to the field line and the 3D eddy feature. Fig. 6 shows an example of tracking particles by ID.

C. Simulation and Data Processing

The simulation and data processing side, located in Singapore, consists of two primary components: XGC1 running on a set of simulation nodes, and a data staging service running on a separate set of nodes. The ADIOS middleware is integrated in both components and used to glue them together to provide in-situ data management and handle data transfers over the WAN.

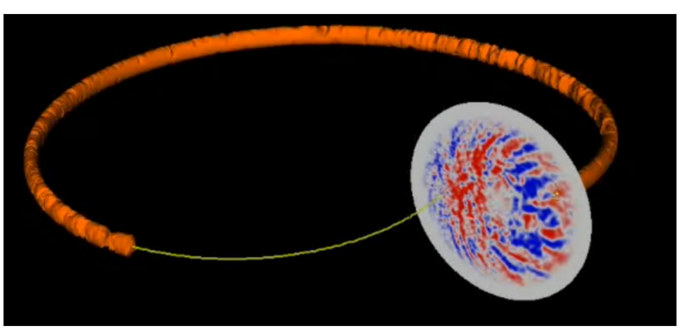

Fig. 4: The VisIt interface window demonstrating the tracking of the magnetic field line and eddy corresponding to the eddy selection in in the picker shown in Fig. 3.

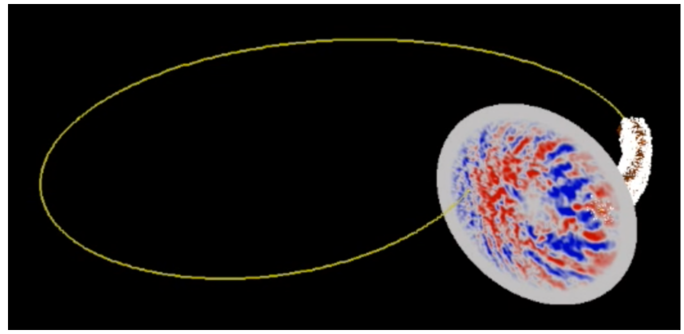

Fig. 5: The VisIt interface window demonstrating particle tracking and disbursement in a turbulent eddy over multiple simulation timesteps.

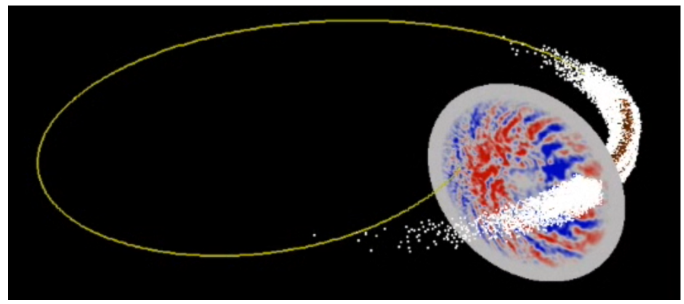

Fig. 6: The VisIt interface window demonstrating particle tracking by ID of particles that were inside a 3D eddy at a particular timestep in the past.

The XGC1 simulation code, which uses ADIOS APIs to handle I/O operations, produces new timesteps once every 10 seconds. The data produced each timestep includes field variables associated with the mesh, and particle data. Compared with the particle data, the field variables are smaller in size and used to monitor simulation progress and analyze the development of turbulent eddies. On the other hand, the particle data is significantly larger (around 62 GB per timestep in our configuration of the simulation). Because of the size, and the small time window available between simulation timesteps, the particle data must be indexed in order for queries from the visualization tool to be retrived in the allowable time frame. As the XGC1 simulation progresses both the field variables and particle data are transferred from the simulation nodes to the staging nodes via the ADIOS middleware APIs.

The data staging service plays four main roles: 1) managing the in-memory data objects (fields and particles) generated from XGC1, 2) indexing particle data for fast retrieval, 3) transferring the field data through the WAN to the visualization side at Georgia Tech, and 4) responding to queries for particle data retrieval from the visualization tools at Georgia Tech. These functions are all integrated within ADIOS, and are executed transparently through calling the ADIOS APIs.

In order to maintain responsiveness to remote users across the world, our system employs two indexing strategies: FastBit [20] indexing and bloom-filter [21] based chunk indexing. FastBit provides a storage efficient way to index data for range queries, while bloom-filter indexing works with chunk-based data. FastBit operates on particle properties, such as spatial coordinates, weights, velocities, etc, and provides results with no false positive, while bloom-filter applies to index particle IDs per chunk but query results can contain false positive answers. In the bloom-filter indexing, the level of false positive rate can be controllable at the expense of index size and performance.

When new simulation data are available, or a query arrives from the visualization tool at Georgia Tech, the data staging service transmits the data through the WAN using transport methods available in ADIOS. We used several different transport methods: ICEE [11], FlexPath [10], and DataSpaces [9]. ICEE and FlexPath were used for long-range RDMA and TCP/IP and DataSpaces was used to synchronize two remote locations over RDMA. These data movement technologies allowed us to perform memory to memory data delivery from one side of the pipeline to the other. The ADIOS middleware makes this delivery transparent to the simulation and visualization at either ends of the pipeline.

V. System Results and Performance

The performance and viability of our system was demonstrated at The A*STAR booth on the SC15 Demo floor. We demonstrated that our system enabled near-real-time interaction with large data sets located around the world. Tests of our system were conducted between Singapore and Georgia Tech with Xwindow forwarding to the showroom floor. We gave our visualization and analysis routines a maximum of 10 seconds to perform an update. This time budget included the time to send 512 planes from Singapore to Georgia Tech, calculate the bounding boxes for the feature of interest on each, send that data back to Singapore, perform the data and particle sub-selection, send that data to Georgia Tech, and perform the visualization update. This maximum time limit kept us below the average time for a new timestep to be produced by the XGC simulation, allowing us to visualize each one as it was produced. Table I presents the size of the data being produced by the simulation, as well as the average data being sent to Georgia Tech after the user made a selection.

TABLE I: Breakdown of the data produced by XGC and processed by our pipeline during the course of the simulation.

Number of Planes	Number of Particles	Number of Time Steps
512	819,200,00	500
Particle Data Size	**3D dpot Data Size**	**Average Data per Selection**
62 GB per step	162 MB per step	0.1% subselection: 62 MB

The amount of data that we ended up having to send from Singapore back to Georgia Tech and process in our visualization routines is one of the main strengths of our system. By identifying the critical subsets of data, as defined by the scientist, we are able provide a near-real-time interactive experience with the running simulation. On average, the amount of particle data moved on each time step was around 62 MB, a mere 0.1% of the total data size.

Additionally, this selection could be done very quickly though our use of FastBit and bloom-filter indexing to perform queries on the simulation side. By having these indices precomputed, subselecting the data in Singapore was reduced to only a few seconds. This allowed our system to remain responsive to user update requests, and enabled new timesteps to be displayed as they were produced. This serves as a demonstration of a significant step forward in accomplishing our goal of a data driven, near-real-time, distributed visualization for a running simulation.

VI. Conclusions and Future Work

Our current work has demonstrated and evaluated several key building blocks (APIs, data representations and indexing, wide area data transfer methods, and efficient visualization engines) to support near real-time analysis and visualization of data across distant sites. This work was motivated by an important category of analysis cases: a hypothesis-driven data analysis case. That is, the analysis workflow, data interactions, and visualization were designed based on the scientists understanding of the formations and paths of turbulent eddies and the orbits of particles within said eddies. This is an important use case as it allows scientists to inspect the behavior of their simulations or experiments and prove/refute hypotheses. In future work, we plan to extend our implementation to support (a) validation analyses and (b) exploratory and discovery analysis use cases. Analysis workflows in such cases may involve a variety of data processing steps for detecting, extracting, and quantifying objects and object features. For example, a validation scenario may access experimental data, extract objects of interest and compare them to objects extracted from running simulations. An exploratory scenario may analyze the data to see if multiple eddies are forming, what the properties of these eddies are, and how they evolve over time. The analysis process can be a pipeline of an object (eddie) detection step, an object segmentation step, a step for computing shape and signal features, and a machine learning step to classify eddies. The validation and exploratory analysis scenarios may involve multiple workflows that are composed and executed by different teams (based on their scientific interests or the level of validation). These scenarios will introduce additional challenges in data management, scheduling of analysis and visualization processes, and efficient wide area data transfers. Nevertheless, we expect the building blocks that we have developed and evaluated in our current work will form a solid foundation on which to build additional functionality to support these future use cases.

As this is an area of active research, we are planning on extending this work in a variety of different directions. First, we plan on using more complex workflows that utilize data from more sources. The work with fusion simulations can be extended to include experimental data, or previously run simulations for comparison. We will rely heavily on the ADIOS middleware to manage the complex, and time critical coordination and movement of data. Additionally, we will continue to work with the various transport methods in ADIOS to optimize performance. These more complex workflows, with different data sources, can employ machine learning techniques to detect features and events automatically. These methods will also serve as mechanisms for steering of simulations and experiments.

We also plan to explore workflows where in situ analysis and visualization are used as end products, or as pre-processing steps for other user defined tasks. As such, we plan to incorporate our previous work with EAVL and ADIOS [13] into these workflows.

For this particular demonstration of nonlinear turbulent eddies, we plan to use more advanced techniques for feature identification and extraction. This includes better identification of 2D features on each plane of the simulation, as well as the 3D construction of the eddies across a set of 2D planes. There is a wealth of research and development that can be utilized for better feature detection. Improved feature detection will allow for better identification of particles within the eddies, and aid in the study of their dynamical behavior in the plasma.

Finally, the infrastructures are largely science agnostic, and so working with additional simulations and experiments will provide opportunities for further expansion.

VII. Acknowledgements

This research was supported in part by the SciDAC Institute of Scalable Data Management, Analysis, and Visualization (SDAV) of the DOE Office of Science through the Office of Advanced Scientific Computing Research (Contract No. 12-015215). This work was supported in part by 1U24CA180924-01A1 from the NCI, R01LM011119-01 and R01LM009239 from the NLM.

References

[1] European Fusion Development Agreement. The iter project, 2006. http://web.archive.org/web/20060824071609/http://www.efda.org/the_iter_project/index.htm.

[2] Wes Bethel. Visapult: A prototype remote and distributed visualization application and framework. *Lawrence Berkeley National Laboratory*, 2000.

[3] Wes Bethel. Visualization dot com. *Computer Graphics and Applications, IEEE*, 20(3):17–20, 2000.

[4] Wes Bethel, Cristina Siegerist, John Shalf, Praveenkumar Shetty, TJ Jankun-Kelly, Oliver Kreylos, and Kwan-Liu Ma. Visportal: Deploying grid-enabled visualization tools through a web-portal interface. *Lawrence Berkeley National Laboratory*, 2003.

[5] Valerio Pascucci, Daniel E Laney, Ray J Frank, Giorgio Scorzelli, Lars Linsen, Bernd Hamann, and Francois Gygi. Real-time monitoring of large scientific simulations. In *Proceedings of the 2003 ACM symposium on Applied computing*, pages 194–198. ACM, 2003.

[6] Steven G Parker and Christopher R Johnson. Scirun: a scientific programming environment for computational steering. In *Proceedings of the 1995 ACM/IEEE conference on Supercomputing*, page 52. ACM, 1995.

[7] Hank Childs, Ma Kwan-Liu, Yu Hongfeng, Whitlock Brad, Meredith Jeremy, Favre Jean, Klasky Scott, Podhorszki Norbert, Schwan Karsten, Wolf Matthew, Parashar Manish, and Zhang Fan. In situ processing. In E. Wes Bethel, Hank Childs, and Charles Hansen, editors, *High Performance Visualization: Enabling Extreme-Scale Scientific Insight*. CRC Press, Boca Raton, FL, 2012.

[8] Qing Liu, Jeremy Logan, Yuan Tian, Hasan Abbasi, Norbert Podhorszki, Jong Youl Choi, Scott Klasky, Roselyne Tchoua, Jay Lofstead, Ron Oldfield, Manish Parashar, Nagiza Samatova, Karsten Schwan, Arie Shoshani, Matthew Wolf, Kesheng Wu, and Weikuan Yu. Hello adios: the challenges and lessons of developing leadership class i/o frameworks. *Concurrency and Computation: Practice and Experience*, 26(7):1453–1473, 2014.

[9] Ciprian Docan, Manish Parashar, and Scott Klasky. Dataspaces: an interaction and coordination framework for coupled simulation workflows. *Cluster Computing*, 15(2):163–181, 2012.

[10] Jai Dayal, Drew Bratcher, Greg Eisenhauer, Karsten Schwan, Matthew Wolf, Xuechen Zhang, Hasan Abbasi, Scott Klasky, and Norbert Podhorszki. Flexpath: Type-based publish/subscribe system for large-scale science analytics. In *14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing(CCGRID '14)*, 2014.

[11] Jong Y Choi, Kesheng Wu, Jacky C Wu, Alex Sim, Qing G Liu, Matthew Wolf, C Chang, and Scott Klasky. Icee: Wide-area in transit data processing framework for near real-time scientific applications. In *4th SC Workshop on Petascale (Big) Data Analytics: Challenges and Opportunities in conjunction with SC13*, 2013.

[12] Roselyne Tchoua, Jong Choi, Scott Klasky, Qing Liu, Jeremy Logan, Kenneth Moreland, Jingqing Mu, Manish Parashar, Norbert Podhorszki, David Pugmire, et al. Adios visualization schema: A first step towards improving interdisciplinary collaboration in high performance computing. In *eScience (eScience), 2013 IEEE 9th International Conference on*, pages 27–34. IEEE, 2013.

[13] David Pugmire, James Kress, Jeremy Meredith, Norbert Podhorszki, Jong Choi, and Scott Klasky. Towards scalable visualization plugins for data staging workflows. In *Big Data Analytics: Challenges and Opportunities (BDAC-14) Workshop at Supercomputing Conference*, November 2014.

[14] Jeremy S Meredith, Sean Ahern, Dave Pugmire, and Robert Sisneros. EAVL: the extreme-scale analysis and visualization library. In *Eurographics Symposium on Parallel Graphics and Visualization*, pages 21–30. The Eurographics Association, 2012.

[15] CS Chang, S Ku, PH Diamond, Z Lin, S Parker, TS Hahm, and N Samatova. Compressed ion temperature gradient turbulence in diverted tokamak edgea). *Physics of Plasmas (1994-present)*, 16(5):056108, 2009.

[16] Hank Childs, Eric Brugger, Brad Whitlock, Jeremy Meredith, Sean Ahern, David Pugmire, Kathleen Biagas, Mark Miller, Cyrus Harrison, Gunther H. Weber, Hari Krishnan, Thomas Fogal, Allen Sanderson, Christoph Garth, E. Wes Bethel, David Camp, Oliver Rübel, Marc Durant, Jean M. Favre, and Paul Navrátil. VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data. In *High Performance Visualization–Enabling Extreme-Scale Scientific Insight*, pages 357–372. Oct 2012.

[17] James Ahrens, Berk Geveci, and Charles Law. Visualization in the paraview framework. In Chuck Hansen and Chris Johnson, editors, *The Visualization Handbook*, pages 162–170, 2005.

[18] Computational Engineering International, Inc. *EnSight website*, June 2015.

[19] Steve M. Legensky. Interactive Investigation of Fluid Mechanics Data Sets. In *VIS '90: Proceedings of the 1st conference on Visualization '90*, pages 435–439. IEEE Computer Society Press, 1990.

[20] K. Wu, S. Ahern, E. W. Bethel, J. Chen, H. Childs, E. Cormier-Michel, C. G. R. Geddes, J. Gu, H. Hagen, B. Hamann, W. Koegler, J. Laurent, J. Meredith, P. Messmer, E. Otoo, V. Perevoztchikov, A. Poskanzer, Prabhat, O. Rübel, A. Shoshani, A. Sim, K. Stockinger, G. Weber, and W.-M. Zhang. FastBit: Interactively Searching Massive Data. In *Journal of Physics Conference Series, Proceedings of SciDAC*, volume 180, page 012053, San Diego, CA, June 2009.

[21] Michael Mitzenmacher. Compressed bloom filters. *IEEE/ACM Transactions on Networking (TON)*, 10(5):604–612, 2002.