Why High Performance Visual Data Analytics is both Relevant and Difficult

E. Wes Bethel, Prabhat, Suren Byna, Oliver Rübel, K. John Wu, and Michael Wehner

Lawrence Berkeley National Laboratory, Berkeley, CA, USA

ABSTRACT

Data visualization, as well as data analysis and data analytics, are all an integral part of the scientific process. Collectively, these technologies provide the means to gain insight into data of ever-increasing size and complexity. Over the past two decades, a substantial amount of visualization, analysis, and analytics R&D has focused on the challenges posed by increasing data size and complexity, as well as on the increasing complexity of a rapidly changing computational platform landscape. While some of this research focuses on solely on technologies, such as indexing and searching or novel analysis or visualization algorithms, other R&D projects focus on applying technological advances to specific application problems. Some of the most interesting and productive results occur when these two activities—R&D and application—are conducted in a collaborative fashion, where application needs drive R&D, and R&D results are immediately applicable to real-world problems.

Keywords: high performance visualization, scientific computing, large data analysis, climate data analysis, plasma physics data analysis

1. INTRODUCTION

The landmark 1987 report by McCormick et al.[1] coining the phrase "Visualization in Scientific Computing" cites a number of challenges, one of which in particular has grown by leaps and bounds as technology evolves: *the need to deal with too much data.* Whereas, in that 1987 report a supercomputer was one capable of $O(100)$ gigaflops, present day supercomputers are capable of on the order 30 petaflops, with exascale-class platforms just around the corner. Concurrent with our increased ability to generate data, observational platforms and instruments have undergone a similar transformation, leaving us awash with data. And it is not just the world of scientific research that faces these challenges: the same challenges, in different forms, face us in nearly all aspects of our economy and society. While there are many different dimensions to the "big data challenge," the ones we discuss here are those that pertain to gaining scientific insight from large, complex collections of data using high performance computational platforms.

We will focus on two seemingly disparate science application areas—climate modeling and plasma physics—to examine a set of interrelated issues that weave together a theme that large-scale visual data analysis is difficult for several different reasons, and that effective solutions really require a close, intimate collaborative effort between researchers from computer and computational science, as well as domain scientists, an idea put forth in the McCormick et al. 1987 report.

Large-scale visual data analysis is difficult in part because it is really an end-to-end problem. In other words, one needs to consider all aspects of the problem, from the specific scientific question being asked at one end, to how to manage data generation, storage, organization and so forth at the other, as well as processing stages in between, along with the types of resources one might employ to implement and deploy solutions. Requirements issues at each of these stages inform design of components as well as shape the entire ecosystem. Additionally, these problems are difficult because the sheer size of data itself can be a problem. For example, the plasma physics case study (see Sec. 3) uses a simulation code to produce a ≈30TB file *per timestep*, and the researchers running this problem simply were unable to obtain sufficient disk space: the simulation runs for thousands of timesteps, but the storage allocation would permit storing and analyzing a handful of timesteps at any give point in time.

Further author information: (Send correspondence to E.W.B.)
E.W.B.: E-mail: ewbethel@lbl.gov, Telephone: 1 510 486 7353

Visualization and Data Analysis 2013, edited by Pak Chung Wong, David L. Kao, Ming C. Hao, Chaomei Chen,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 8654, 86540B · © 2013 SPIE-IS&T
CCC code: 0277-786X/13/$18 · doi: 10.1117/12.2010980

Another issue is the confluence of complexity between the underlying data and the science questions being asked. A significant challenge facing us as a society is coming to grips with the potential impact of climate change. A climate science case study (see Sec. 2) shows how researchers are developing new technologies that are flexible and adaptable to quickly provide answers, using some of the worlds largest computational platforms, about how potential future climate scenarios might impact us through a change in the frequency and strength of extreme weather events.

2. CLIMATE DATA ANALYTICS

As with many computational science domains, the study of climate and climate change, benefits from increasingly powerful computational platforms. Modern climate codes, which model processes in the atmosphere, ocean, ice caps, and more, produce massive amounts of data. For example, when modeling the atmosphere at 0.25° resolution for a period of about two decades of simulation time, the CAM5.1 code[2] produces approximately 100TB of model output for a single problem configuration. When a code is run multiple times using different initial conditions, or the physics is slightly perturbed, the result is an *ensemble* collection of data. These ensembles are a crucial tool for international and national organizations, such as the Intergovernmental Panel on Climate Change (IPCC), tasked to assess the human role in climate change and to assess potential adaptation and mitigation strategies for reducing our role in climate change. The anticipated aggregate size of the model output to be used in AR5, the next worldwide assessment, will range into the multiples of PBs due to the number of models that comprise the study, each of which will produce an ensemble collection of output that assumes varying initial conditions and future atmospheric and climate scenarios.

One challenge facing the climate science community is a rich legacy of visualization and analysis tools that are serial in implementation, yet such tools are not capable of processing modern-sized data sets due to memory constraints. Worse, such tools are often incapable of performing the type of analysis required to gain understanding in ever-larger and ever-richer collections of data. Specific questions that will be part of AR5 include the study of the frequency and intensity of extreme weather events in a changing climate. Extreme weather events include things like severe storms, extreme precipitation and temperature events, droughts, and the like. These extreme weather events will have severe anthropogenic impact, ranging from the loss of coastal and island habitat due to sea level rise to changes in agricultural patterns. The legacy software tools that have served the community well for the past three decades are simply not capable of processing the sheer volume of climate model or observational data required by the study, nor do they contain the fundamental algorithms needed to answer these specific climate science questions.

In 2010, the U. S. Department of Energy's Office of Biological and Environmental Research funded three projects aimed at producing the next generation of software for analyzing and visualizing massive climate data collections, with an eye towards equipping the climate science community with the software infrastructure needed to conduct a study of unprecedented magnitude. The subsections that follow present work by Prabhat et al. 2012[3] and Byna et al. 2011[4] that focus on software infrastructure for extreme-scale analytics of climate model and observational data.

2.1 Scalable Analytics Infrastructure

Output from codes like CAM5—which sample and discretize both space and time in a regular fashion and are often generated by ensemble simulation experiments for varying physical input parameters and initial conditions—exposes data parallelism in many different ways to analysis and visualization tasks. Many types of visualization or analysis codes can be run independently, and in parallel, across individual ensemble members, timesteps, spatial regions, and individual grid points.

There are several pressing needs and challenges within the climate modeling community. First is the growing "impedance mismatch" between the ability to collect or generate data, and the capacity to perform analysis and visualization on massive data sets. Second is the need to be able to quickly and easily create and test a variety of different types of quantitative analysis activities, such as spatiotemporal feature detection, tracking, and analysis. Third is the ability to execute such capabilities on large, parallel platforms, which have resources sufficient to process massive data sets in a reasonable amount of time.

Towards addressing these needs, Prabhat et al., 2012,[3] developed the Parallel Toolkit for Extreme Climate Analysis (TECA). Their objective was to enable rapid implementation of a variety of user-written and customizable spatiotemporal feature detection, tracking, and analysis algorithms in a single framework that accommodates different varieties of data parallelism. Within that framework, there are a number of capabilities that are common to all feature detection tasks, such as loading data files, accommodating different calendering systems, data scatter, etc. They demonstrate application of this framework to the detection analysis of different types of extreme weather events, such as tropical cyclones (see Sec. 2.2), extra-tropical cyclones, and atmospheric rivers (see Sec. 2.3).

To the user-developer who wants to implement a new feature detection and tracking algorithm, they write code that is handed a 1D/2D/3D sub-block of data and perform their computations on that block, storing results in a "local table." Later, the local tables are gathered and processed in a serial post-processing phase, which is typically much smaller in size compared to the original problem and easily accommodated in serial. The developer need not make any MPI calls, nor be concerned with the mechanics of data distribution or the gathering of results. This processing pattern is similar in many respects to MapReduce:[5] data distribution followed by a parallel processing stage, in turn followed by a gather and process stage.

2.2 Application: Tropical Cyclones

In terms of monetary damage and loss of or impact to human life, tropical storms are one of the most significant extreme weather events.[6] Gaining a better understanding of the future trend of frequency and severity of the tropical storms is critical to assessing impacts of climate change. High resolution climate models run under different scenarios of human activity provide a means to predict these trends.[6–8] Such models are tested by performing simulations of the recent past and comparing results against available observations as a verification. After assessing a model against these observations, they are run far into the future to investigate how tropical storm characteristics might change.

One well-known prior work for detecting tropical cyclones in model output is the TSTORMS code, developed at the Princeton Geophysical Fluid Dynamics Library.[8] It is representative of many such climate analysis programs: it processes climate model output one time step at a time, reads a handful of variables from the modeling output and then performs a series of computations to produce a relatively small set of candidate points.

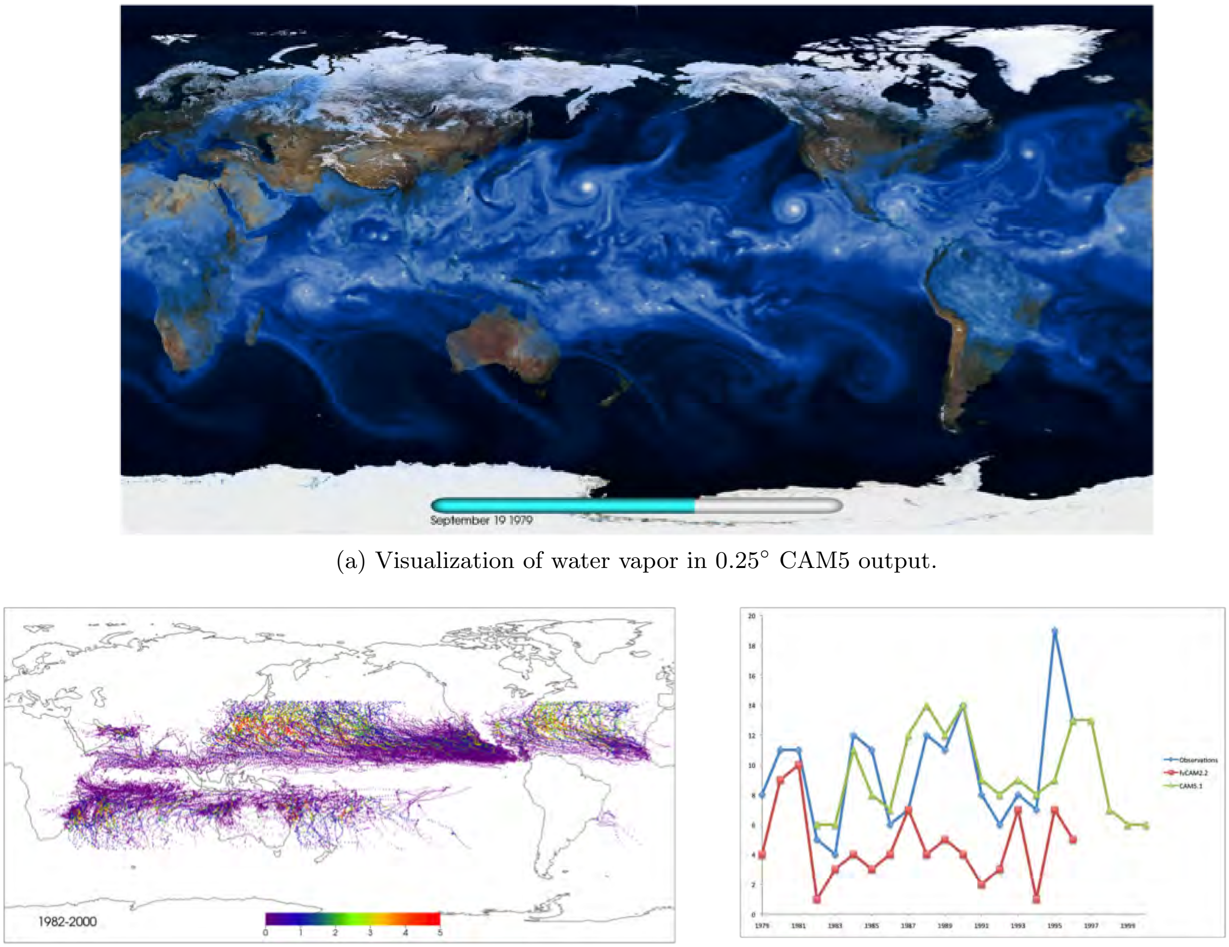
The majority of computation time in the original TSTORMS code, which was a serial Fortran 77 implementation, was consumed by the detection processing stage. The detection stage consists of examining each grid point and determining if it satisfies a multi-variate condition: vorticity must exceed a given threshold; low pressure conditions prevail within a short distance from a vorticity threshold; and there exists a local temperature gradient going from warm (storm center) to cooler. All grid points that satisfy these multi-variate conditions are classified as candidates for the next processing step, which candidates are stitched together into tracks using spatiotemporal constraints related to storm movement over time and known biases in storm direction of movement.

Prabhat et al. implemented the multi-variate detection criteria in TECA, in effect by having TECA invoke the detection kernel from the original TSTORMS code, and applied the framework to 100TB of 19 simulated years of CAM5 model output[*]. At ≈7K-way parallel, execution time was about *2 hours*, compared to an estimated serial run time of about *583 days.* Visual results from a similar, yet earlier tropical cyclones study, is shown in Figure 1. More recent versions of this work, yet unpublished, have scaled to approximately 80,000 cores.

2.3 Application: Atmospheric Rivers

Extreme precipitation events on the western coast of North America are often traced to an unusual weather phenomenon known as "atmospheric rivers" (ARs). Such events occur when long and narrow structures in the lower atmosphere transport tropical moisture distant regions outside of the tropical zone.[9,10] As they can be highly localized, *river* is an apt description of such a narrow stream of moisture moving at high speeds across

[*] Of the original 100TB of model output, about 200GB was used for this particular analysis



(a) Visualization of water vapor in 0.25° CAM5 output.

(b) Cyclone tracks computed from 100TB of CAM5 output, colored by storm categories on the Sapphir-Simpson scale.

(c) Comparing the annual count of observed vs. modeled cyclones.

Figure 1: In this example, a high resolution atmospheric code produces massive amounts of data and the science objective is to study the number of cyclones that form over time. One timestep of model output is visualized (a). The TECA code is run in parallel to identify cyclones and their tracks over time (b). These results are compared to the counts of cyclones observed over the same time period, as well as to a third model's (fvCAM2.2) output (c). This data source for this particular problem was about 100TB of CAM5 output processed on 7,000 cores in about two hours, compared to an estimated serial processing time of about 583 days. Images courtesy of Prabhat, Wehner, et al. (LBNL).

thousands of kilometers. AR events occur in oceans around the globe, including the Atlantic basin affecting the British Isles.

One key characteristic recognized in earlier studies of ARs is the moisture flux.[11] However, that quantity turns out to be difficult to observe directly. Ralph et al. 2004[12] established a much simpler set of conditions to identify atmospheric rivers in satellite observations. Their detection is primarily based on the total column Integrated Water Vapor (IWV) content. They identify an AR as atmospheric feature with $IWV > 2cm$, more than 2000 km in length, and less than 1000 km in width. Whereas instruments like satellites can measure and report IWV directly, it must be computed from the water vapor field in model output by performing a vertical integration.

The implementation in TECA consists of the following processing stages. Detection consists of finding all grid points where IWV exceeds a specified threshold, and where the grid points are in certain geographic regions (e.g., they lie outside tropical latitudes). Next, take all candidate grid points and group them together into regions using a connected component labeling algorithm to form a candidate region. For each candidate region,

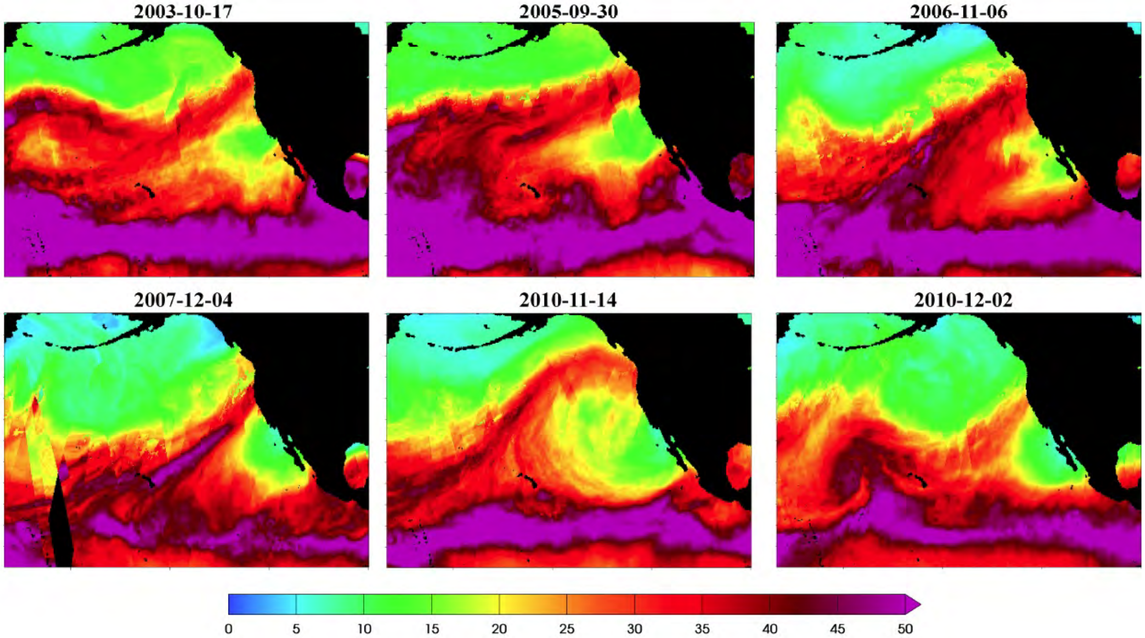


Figure 2: Sample CAM5 AR detections from our parallel code. AR events in February and December of 1991 are plotted. Note that our method is able to detect AR events of different shapes and sizes. A selection of events in the SSM/I satellite record detected by our pattern recognition technique as atmospheric rivers. Note that they share a similar horseshoe-like pattern but vary greatly in their spatial extent. Events that pass over Hawaii, such as occurred on November 6, 2006, are often referred to as "The Pineapple Express." Image source: Byna et al. 2011.[4]

check it against criteria identified by Ralph et al.: does it originate in a tropical latitude and make landfall outside the tropics; determine if the region satisfies geometric size/shape constraints specified by Ralph et al.

Prabhat et al. processed 27 years of CAM5 data (1979-2005) with the TECA implementation of the atmospheric river detection code using ≈10K cores in about *4 seconds*. In contrast, a serial execution of this code would have required about *11 hours*. Figure 2 shows visual results from an earlier, similar study.[4]

3. PLASMA PHYSICS DATA ANALYTICS

Collisionless magnetic reconnection is an important mechanism that releases energy explosively as magnetic field lines break and reconnect in plasmas, such as when the Earth's magnetosphere reacts to solar eruptions. Such a reconnection also plays an important role in a variety of astrophysical applications involving both hydrogen and electron-positron plasmas, and is the dominant mechanism enabling plasma from the solar wind to enter the Earth's magnetosphere. Reconnection is inherently a multi-scale problem. It is initiated in the small scale around individual electrons, but eventually leads to a large-scale reconfiguration of the magnetic field. Recent simulations have revealed that electron kinetic physics is not only important in triggering reconnection,[13–19] but also in its subsequent evolution. These findings suggest the need to model detailed electron motion, which poses severe computational challenges for 3D simulations of reconnection. A full-resolution magnetosphere simulation is an exascale computing problem.

Computational plasma physicists are generally interested in understanding the structure of high dimensional phase space distributions. For example, in order to understand the physical mechanisms responsible for producing magnetic reconnection in a collisionless plasma, it is important to characterize the symmetry properties of the particle distribution, such as agyrotropy.[20] Agyrotropy is a quantitative measure of the deviation of the distribution from cylindrical symmetry about the magnetic field. Another question of significant practical importance is characterization of the energetic particles and their behavior in space and time. For instance, are energetic particles preferentially accelerated along the magnetic field line, and what is their spatiotemporal distribution? During reconnection and near the "hot spot," what is the degree of agyrotropy in the particle distribution along the magnetic field line?

Recent work by Byna et al.[21] has approached this daunting set of scientific questions in a holistic way that considers the end-to-end problem of producing, storing, analyzing, and visualizing plasma physics simulation data of unprecedented scale. Given the science questions above and the desire to run the VPIC simulation at very high resolution, Byna et al. addressed data management challenges in storing, analyzing, and visualizing simulation output. They collaborated with the VPIC team to complete trillion particle simulation runs, producing ≈32TB of particle data per timestep, on 120,000 cores of the Hopper platform at NERSC[†]. They focused on some key research questions from data management (What is a scalable I/O strategy for storing massive particle data output?), analysis (What is a scalable strategy for conducting analysis on these datasets?) and visualization (What is the visualization strategy for examining these datasets?).

3.1 Parallel I/O

In the original implementation of the VPIC code, each MPI domain writes a file containing its particle data[22] into a custom binary file format. This *file-per-process (fpp)* approach is able to achieve a good fraction of system I/O bandwidth, but has a number of limitations. The first is that the number of files at large scale becomes too large. For example, in Byna et al.'s largest scale test, the simulation generates $20,000$ files per timestep. Performing even a simple *ls* command on the directory containing these files has significant latency. Second, the *fpp* model dictates the concurrency of subsequent stages in the analysis pipeline. Often a post-processing step is necessary to refactor *fpp* data into a format that is readable by analysis tools. Third, many data management and visualization tools only support standard data formats, such as HDF5[23] and NetCDF.[24]

Byna et al. used the approach of modifying the VPIC code to write a single global file using a particle data extension of parallel HDF5 called H5Part,[25] which is a "veneer API" for HDF5 that encapsulates much of the complexity of implementing effective parallel I/O in HDF5. The H5Part interface opens the file with MPI-IO collective buffering and Lustre optimizations enabled. Collective buffering partitions the parallel I/O operations into two stages. The first stage uses a subset of MPI tasks to aggregate the data into buffers, and the aggregator tasks then write data to the I/O servers. With this strategy, fewer nodes communicate with the I/O nodes, which reduces contention. The Lustre-aware implementation of Cray MPI-IO sets the number of aggregators equal to the striping factor such that the stripe-sized chunks do not require padding to achieve stripe alignment.[26] Because of the way Lustre is designed, stripe alignment is a key factor in achieving optimal performance.

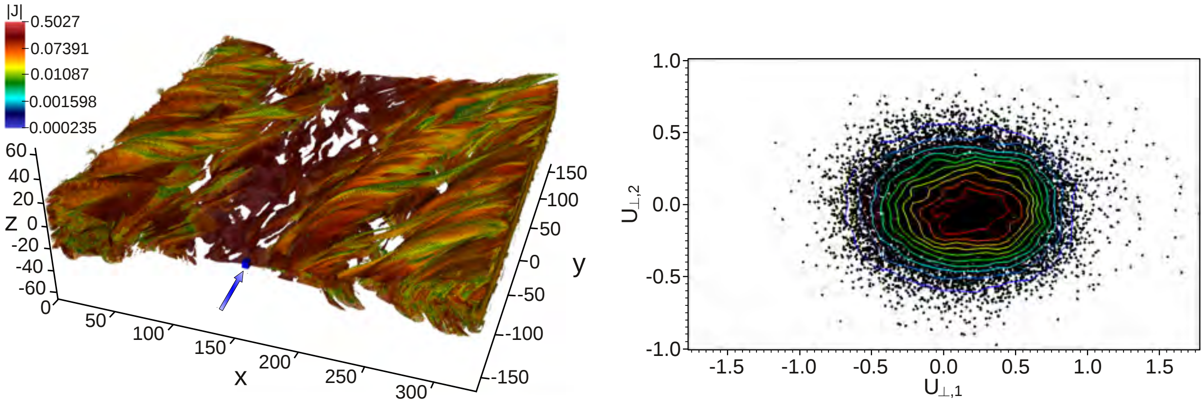
Byna et al. conducted a series of weak-scaling experiments designed to study the behavior of the collective, parallel I/O approach and to compare it with the traditional *fpp* approach. Their results, described in more detail the original study,[21] show that their approach is able to achieve a sustained 77% of peak I/O rate at 120K cores (≈27GB/s measured, ≈35GB/s theoretical peak). The *fpp* approach initially achieved a larger percent of peak performance, but then the performance falls off due to load imbalance in the two-stage non-collective I/O process. The study results show the practicality of using collective I/O in terms of competitive write performance.

3.2 Parallel Index/Query

A central activity in all visual data analysis applications is the process of "discovering" features or phenomena in data collections. Discovery often entails asking questions of one type or another, then searching through collections of data. Byna et al. approach this activity by creating a new, hybrid parallel implementation of FastQuery[27–29] to accelerate the data analysis process of the trillion particle dataset.

FastQuery is a parallel indexing and querying system for array-based scientific data formats. It uses the FastBit bitmap indexing technology[30] to accelerate data selection based on arbitrary range conditions defined on the available data values, e.g., "energy $> 10^5$ and temperature $> 10^6$." The two main functions of FastQuery are indexing and querying. Indexing builds the indexes of the data and stores them into a single file. Querying function evaluates different queries by accessing the data and the indexes conveniently from the same file.

[†] The National Energy Research Scientific Computing Center (NERSC) is the primary scientific computing facility for the U. S. Department of Energy's Office of Science. See `http://www.nersc.gov/`



(a) Isosurface plot of the positron particle density n_p with color indicating the magnitude of the total current density $|J|$. Note the logarithmic color scale. The blue box (indicated by the blue arrow) is located in the X-line region of the simulation and illustrates the query $(157.654 < x < 1652.441)\&\&(-165 < y < -160.025)\&\&(-2.5607 < z < 2.5607)$, which we use in Figure 3b to study agyrotropy.

(b) Particle scatter plot (black) of $U_{\perp,1}$ vs. $U_{\perp,2}$ of all energetic particles (with $energy > 1.3$) contained in the box of an X-line region. Additional isocontours indicate the associated particle density (blue = low density and red = high density). The complete query used to extract the particles is defined as: $(energy > 1.3)\&\&(157.654 < x < 162.441)\&\&(-165 < y < -160.025)\&\&(-2.5607 < z < 2.5607)$. The query results in a total of 22,812 particles. The elliptical shape of the particle distribution is indicative of agyrotropy in the X-line region.

Figure 3: Energetic particles near a reconnection hot spot, shown in the context of the computational domain (left), undergo analysis to determine their level of energy agyrotropy in that region (right). Image source: Byna et al. 2012.[21]

In order to process massive datasets, FastQuery uses parallelism across distributed memory nodes, as well as multiple CPU cores available on each node. The basic strategy of the parallel FastQuery implementation is to partition a large-scale dataset into multiple fixed-size sub-arrays and to assign the sub-arrays among processes for indexing and querying. When constructing the indexes, the processes build bitmaps on sub-arrays consecutively, and store them into the same file. The process of writing bitmaps uses collective I/O calls in high-level I/O libraries, such as HDF5. When evaluating a query, the processes apply the query on each sub-array and return the aggregated results.

Byna et al. implemented a hybrid parallel version of FastQuery using MPI and pthreads.[31,32] The strategy is to let each MPI task create a fixed number of threads. The MPI tasks are only responsible for holding shared resources among threads, such as the MPI token for inter-process communication and the memory buffer for collective I/O, while the threads do the actual processing tasks of creating indexes and evaluating queries. The hybrid parallel FastQuery divides a dataset into multiple fixed size sub-arrays, and builds the indexes of those sub-arrays iteratively. During query processing, the hybrid parallel FastQuery is able to load indexes from the index file and evaluate bitmaps in-memory without involving any HDF5 collective calls.

Byna et al. conducted a series of experiments that focus on better understanding the performance differences between the MPI-only and hybrid parallel versions of FastQuery for index and query operations on the large plasma physics simulation output. For index building using the one trillion particle H5part file, the hybrid parallel version performs about 19% faster than the MPI-only version at 10K-way parallel primarily due to more efficient read I/O (less contention, better load balancing). For query operations, that is searching for those particles that are highly energetic so as to study their behavior, Byna et al.'s implementation completes the search through the 30TB collection of one trillion particles in less than 3 seconds when executed on 1250 cores.

3.3 Parallel Query-driven Visualization

The previous two subsections focus on the technologies and techniques for collecting, storing, and searching simulation data for features or phenomena of interest. An effective visual data analysis application will tie together these capabilities in a way they can be brought to bear on a specific application. Byna et al.'s approach is to implement these capabilities into VisIt, which has been demonstrated to operate at scale.[33] The basic idea here is to implement query-driven visualization capabilities[34] into a production-quality platform that scales well so as to leverage a substantial amount of visualization and analysis infrastructure. Here, a VisIt implementation entails a combination of modifying the H5part file loader in VisIt to use the new FastQuery implementation to execute queries that find energetic particles, then make use of VisIt's capabilities to perform visualization and analysis of the query results.

Using these capabilities, Byna et al. explore answers to several different science questions that center around studying the behavior of energetic particles and their relationship to the magnetic field. One of the primary questions they sought to study, which was previously not possible before their work due to the sheer size and complexity of the underlying data and the inability of off-the-shelf technologies to enable scientific inquiry, is to better understand the spatial distribution of energetic particles in the region of magnetic reconnection. The results, shown in Figure 3, reveal the particle distribution $F(U_{\perp,1}, U_{\perp,2})$ in the vicinity of an X-line. The distribution clearly shows the agyrotropy of the distribution, i.e., the lack of cylindrical symmetry about the local magnetic field. Agyrotropy is an expected signature of the reconnection site in collisionless plasma. While it has been well documented in simple 2D simulations, classification of agyrotropic distributions in 3D simulations have been much more challenging. While some information about agyrotropy can be recovered from coarser-level moment computations, a direct computation based on particle data provides richer information about the structure of particle phase space. With these new capabilities, researchers are now well poised to compute agyrotropy and other finer characterizations of distribution functions.

4. CONCLUSIONS

The two case studies presented here serve to illuminate some key points. First is the notion that the multidisciplinary team has proven to be "the engine" that produces effective solutions to tackle these difficult challenges. Having science questions drive the evolution of solutions, guided by the adaptation to changing technology, has proven to be a successful formula in many different projects. This configuration ensures that new technologies are indeed useful in science, and also encourages cross-field growth of ideas and exchanges. Second is the notion that these complex problems are best solved by considering the entire ecosystem of science questions, data sources, and processing infrastructure to frame requirements that in turn inform design and deployment.

ACKNOWLEDGMENTS

This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research was supported in part by National Science Foundation under NSF grant OCI 0904734. This research used resources of the National Energy Research Scientific Computing Center. Simulations were also supported by an allocation of advanced computing resources provided the NSF at the National Institute for Computational Sciences and by NASA (Pleiades), and the National Center for Computational Sciences at Oak Ridge National Laboratory (Jaguar).

REFERENCES

[1] McCormick, B., DeFanti, T., and Brown, M., "Visualization in Scientific Computing," *Computer Graphics* (Nov. 1987).

[2] Neal, R. B. et al., "Description of the NCAR Community Atmosphere Model (CAM 5.0)," tech. rep., National Center for Atmospheric Research, Boulder, CO, USA (2010). NCAR/TN-486+STR.

[3] Prabhat, Rübel, O., Byna, S., Wu, K., Li, F., Wehner, M., and Bethel, E. W., "TECA: A Parallel Toolkit for Extreme Climate Analysis," in [*Third Worskhop on Data Mining in Earth System Science (DMESS 2012) at the International Conference on Computational Science (ICCS 2012)*], (June 2012). LBNL-5352E.

[4] Byna, S., Prabhat, Wehner, M. F., and Wu, K., "Detecting Atmospheric Rivers in Large Climate Datasets," in [*Proceedings of the 2nd International Workshop on Petascale Data Analytics: Challenges, and Opportunities (PDAC-11/Supercomputing 2011)*], ACM/IEEE, Seattle, WA, USA (Nov. 2011).

[5] Dean, J. and Ghemawat, S., "Mapreduce: simplified data processing on large clusters," *Commun. ACM* **51**, 107–113 (Jan. 2008).

[6] Wehner, M. F., Bala, G., Duffy, P., Mirin, A. A., and Romano, R., "Towards Direct Simulation of Future Tropical Cyclone Statistics in a High-Resolution Global Atmospheric Model," *Advances in Meteorology* **2010**, Article ID 915303 (2010).

[7] Webster, P. J., Holland, G. J., Curry, J. A., and Chang, H., "Changes in Tropical Cyclone Number, Duration, and Intensity in a Warming Environment," *Science* **309**(5742), 1844–1846 (2005).

[8] Knutson, T. R., Sirutis, J. J., Garner, S. T., Held, I. M., and Tuleya, R. E., "Simulation of the recent multidecadal increase of atlantic hurricane activity using an 18-km-grid regional model," *Bulletin of the American Meteorological Society* **88**(10), 1549–1565 (2007).

[9] Ralph, F. M. and Dettinger, M. D., "Storms, floods and the science of atmospheric rivers," *EOS, Transactions of AGU* **92**(32), 265–266 (2011).

[10] Dettinger, M. D., Ralph, F. M., Das, T., Neiman, P. J., and Cayan, D. R., "Atmospheric rivers, floods and the water resources of california," *Water* **3**(2), 445–478 (2011).

[11] Zhu, Y. and Newell, R. E., "A proposed algorithm for moisture fluxes from atmospheric rivers," *Monthly Weather Review - USA* **126**(3), 1998 (725-735).

[12] Ralph, F. M., Neiman, P. J., and Wick, G. A., "Satellite and caljet aircraft observations of atmospheric rivers over the eastern north pacific ocean during the winter of 1997/98," *Monthly Weather Review* **132**(7), 1721–1745 (2004).

[13] Daughton, W., Scudder, J. D., and Karimabadi, H., "Fully kinetic simulations of undriven magnetic reconnection with open boundary conditions," *Physics of Plasmas* **13** (2006).

[14] Daughton, W., Roytershteyn, V., Karimabadi, H., Yin, L., Albright, B. J., Bergen, B., and Bowers, K. J., "Role of electron physics in the development of turbulent magnetic reconnection in collisionless plasmas," *Nature Physics* **7**, 539–542 (2011).

[15] Egedal, J., Daughton, W., and Le, A., "Large-scale electron acceleration by parallel electric fields during magnetic reconnection," *Nature Physics* **8**, 321–324 (2012).

[16] Karimabadi, H., Daughton, W., and Scudder, J., "Multi-scale structure of the electron diffusion region," *Geophys. Res. Lett.* **34** (2007).

[17] Shay, M., Drake, J., and Swisdak, M., "Two-scale structure of the electron dissipation region during collisionless magnetic reconnection," *Phys. Rev. Lett.* **99** (2007).

[18] Klimas, A., Hesse, M., and Zenitani, S., "Particle-in-cell simulation of collisionless reconnection with open outflow boundary conditions," *Physics of Plasmas* , 082102–082102–9 (2008).

[19] V., R., Daughton, W., Karimabadi, H., and Mozer, F. S., "Influence of the lower-hybrid drift instability on magnetic reconnection in asymmetric configurations," *Phys. Rev. Lett.* **108**, 185001 (2012).

[20] Scudder, J. D., Holdaway, R. D., Daughton, W. S., Karimabadi, H., Roytershteyn, V., Russell, C. T., and Lopez, J. Y., "First resolved observations of the demagnetized electron-diffusion region of an astrophysical magnetic-reconnection site," *Phys. Rev. Lett.* **108**, 225005 (Jun 2012).

[21] Byna, S., Chou, J., Rübel, O., Prabhat, Karimabadi, H., Daughton, W. S., Roytershteyn, V., Bethel, E. W., Howison, M., Hsu, K.-J., Lin, K.-W., Shoshani, A., Uselton, A., and Wu, K., "Parallel I/O, Analysis, and Visualization of a Trillion Particle Simulation," in [*Proceedings of SuperComputing 2012*], (Nov 2012). LBNL-5832E.

[22] Karimabadi, H., Loring, B., Majumdar, A., and Tatineni, M., "I/O strategies for massively parallel kinetic simulations." SC 2010 Research Poster Reception (2010).

[23] The HDF Group, "HDF5 user guide." `http://hdf.ncsa.uiuc.edu/HDF5/doc/H5.user.html` (2010).

[24] Unidata, "The NetCDF users' guide." `http://www.unidata.ucar.edu/software/netcdf/docs/netcdf/` (2010).

[25] Howison, M., Adelmann, A., Bethel, E. W., Gsell, A., Oswald, B., and Prabhat, "H5hut: A High-Performance I/O Library for Particle-Based Simulations," in [*Proceedings of 2010 Workshop on Interfaces and Abstractions for Scientific Data Storage (IASDS10)*], (Sept. 2010). LBNL-4021E.

[26] "Getting Started with MPI I/O." `http://docs.cray.com/books/S-2490-40/S-2490-40.pdf`.

[27] Chou, J., Wu, K., Rübel, O., Howison, M., Qiang, J., Prabhat, Austin, B., Bethel, E. W., Ryne, R. D., and Shoshani, A., "Parallel index and query for large scale data analysis," in [*SC11*], (2011).

[28] Chou, J., Wu, K., and Prabhat, "FastQuery: A general indexing and querying system for scientific data," in [*SSDBM*], 573–574 (2011). `http://dx.doi.org/10.1007/978-3-642-22351-8_42`.

[29] Chou, J., Wu, K., and Prabhat, "FastQuery: A parallel indexing system for scientific data," in [*IASDS*], IEEE (2011).

[30] Wu, K., "FastBit: an efficient indexing technology for accelerating data-intensive science," *Journal of Physics: Conference Series* **16**, 556–560 (2005). `http://dx.doi.org/10.1088/1742-6596/16/1/077`.

[31] Henty, D. S., "Performance of hybrid message-passing and shared-memory parallelism for discrete element modeling," in [*SC'00*], IEEE Computer Society, Washington, DC, USA (2000).

[32] Howison, M., Bethel, E. W., and Childs, H., "MPI-hybrid parallelism for volume rendering on large, multi-core systems," in [*Eurographics Symposium on Parallel Graphics and Visualization*], 1–10 (2010).

[33] Childs, H., Pugmire, D., Ahern, S., Whitlock, B., Howison, M., Prabhat, Weber, G. H., and Bethel, E. W., "Extreme scaling of production visualization software on diverse architectures," *IEEE Computer Graphics and Applications* **30**, 22–31 (2010).

[34] Stockinger, K., Shalf, J., Bethel, W., and Wu, K., "Query-driven visualization of large data sets," in [*IEEE Visualization 2005, Minneapolis, MN, October 23-28, 2005*], 22 (2005). `http://doi.ieeecomputersociety.org/10.1109/VIS.2005.84`.