BIG DATA ANALYSIS AND VISUALIZATION: WHAT DO LINACS AND TROPICAL STORMS HAVE IN COMMON?

E.W. Bethel*, S. Byna, J. Chou†, E. Cormier-Michel‡, C.G.R. Geddes,
M. Howison, F. Li, P. Prabhat, J. Qiang, O. Rübel, R.D. Ryne, M. Wehner, K. Wu
LBNL, Berkeley, CA 94720, USA

Abstract

While there is wisdom in the old adage "the two constants in life are death and taxes," there are unavoidable truths facing modern experimental and computational science. First is the growing "impedance mismatch" between our ability to collect and generate data, and our ability to store, manage, and gain understanding from it. The second is the fact that we cannot continue to rely on the same software technologies that have worked well for the past couple of decades for data management, analysis, and visualization. A third is that these complementary activities must be considered in a holistic, rather than Balkanized way. The inseparable interplay between data management, analysis, visualization, and high performance computational infrastructure, are best viewed through the lens of case studies from multiple scientific domains, where teams of computer and scientists combine forces to tackle challenging data understanding problems.

INTRODUCTION

Big data and its attendant challenges—managing it and gaining insight into its secrets—are the subject of a substantial amount of research and development in industry, academia, and the broader scientific community. These challenges are often cited as being among the greatest barriers facing scientific knowledge discovery [1].

Within the high performance visualization and analysis community, several different but complementary approaches contribute solutions to these challenges. One approach focuses on increasing the capacity of the data analysis and visualization processing pipelines [2], and recent results have shown such techniques capable of scaling to extreme concurrency for both production [3] and research codes [4].

An alternative approach is to focus instead on limiting visualization and analysis processing to the subset of data of interest where presumably the interesting subset is much smaller in size than the original data. Projects in this space have focused on blending index and query technologies from scientific data management with visualization and analysis tools to implement what is known as *query-driven visualization* [5]. This class of solution has proven applicable to diverse problems ranging from forensic cybersecurity [6] to accelerator modeling [7].

* Corresponding author ewbethel@lbl.gov
† jchou@cs.nthu.edu.tw
‡ ecormier@txcorp.com

The focus of this paper is on the nexus between those two approaches with an eye towards enabling scientific insight. Three case studies, one from climate modeling and two from accelerator modeling, show how high performance computing technology and advances in visualization and analysis software infrastructure combine to provide new scientific data understanding capabilities brought to bear on problems that can be characterized as "large data analysis and visualization."

CASE STUDY: TROPICAL CYCLONES AND CLIMATE MODELING DATA ANALYSIS

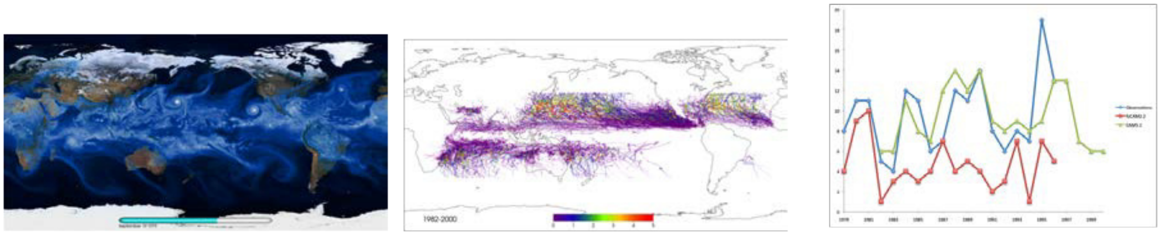
As with many computational science domains, the study of climate and climate change, benefits from increasingly powerful computational platforms. Modern climate codes, which model processes in the atmosphere, ocean, ice caps, and more, produce massive amounts of data. As discussed below, when modeling the atmosphere at $0.25°$ resolution for a period of about two decades of simulation time, the CAM5.1 code [8] produces approximately 100TB of model output. These data sizes will only grow in size with time.

One challenge facing this computational 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.

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 their ability to collect or generate data, and their ability 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

ISBN 978-3-95450-116-8



(a) Visualization of 0.25° CAM5 output. (b) Cyclone tracks computed from 100TB of CAM5 output. (c) Comparing the annual count of observed vs. modeled cyclones.

Figure 1: In this climate science 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). Images courtesy of Prabhat, Wehner, et al. (LBNL).

platforms, which have resources sufficient to process massive data sets in a reasonable amount of time.

Towards addressing these needs, Prabhat et al., 2012 [9], 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.

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.

The authors present several case studies where TECA is applied to detecting, tracking, and analyzing different extreme-weather phenomena, including tropical cyclones, extra-tropical cyclones, and atmospheric rivers. Results from a similar, yet earlier tropical cyclones study, is shown below in Figure 1. One of the major accomplishments of their work is the ability to scale the tropical cyclone feature detection, tracking, and analysis code to 7,000 cores to complete analysis/processing of 100TB of CAM5 model output in approximately 2 *hours*, compared to an estimated serial processing time of approximately 583 *days*. More recent versions of this work, yet unpublished, have scaled to approximately 80,000 cores. In that example, the user-written code examines its local data block to determine if that block contains a cyclone, which is defined as a combination of locally high vorticity, low pressure within a defined distance from the high vortex, and a locally warm temperature combined with a temperature dropoff in nearby locations.

CASE STUDY: LINACS, LPAS, AND ACCELERATOR MODELING DATA ANALYSIS

While the previous section focuses on quantitative analysis, specifically feature detection and tracking in climate model output, this section focuses on visual data exploration and analysis of particle-based data produced by accelerator modeling codes. These codes output massive amounts of particle-based data at each timestep; the two examples below consist of hundreds of millions of particles per timestep. Generally speaking, the overall objective is to find and analyze "interesting particles," where the definition of "interesting" varies from study to study. Also, generally speaking, these so-called interesting particles can be characterized as satisfying some set of multivariate conditions in space and time.

As with the climate example, these communities have built up a rich legacy of serial tools over time for visualization and analysis, tools that cannot accommodate modern massive data sizes. The subsections below discuss work aimed at providing the infrastructure for next-generation visualization and analysis tools that are applicable to accelerator modeling output of massive size and on modern supercomputing platforms.

Laser-Plasma Accelerators

One central challenge in the analysis of large LPA simulation data arises from the fact that, while large numbers of particles are required for an accurate simulation, only a small fraction of the particles are accelerated to high energies and subsequently form particle features of interest. During the course of a simulation, multiple high-energy particle beams may form and additional particle bunches may appear in secondary periods of the plasma wave. Studying this acceleration phenomena requires gaining insight into which particles become accelerated, how are they trapped and accelerated by the plasma wave, and

ISBN 978-3-95450-116-8

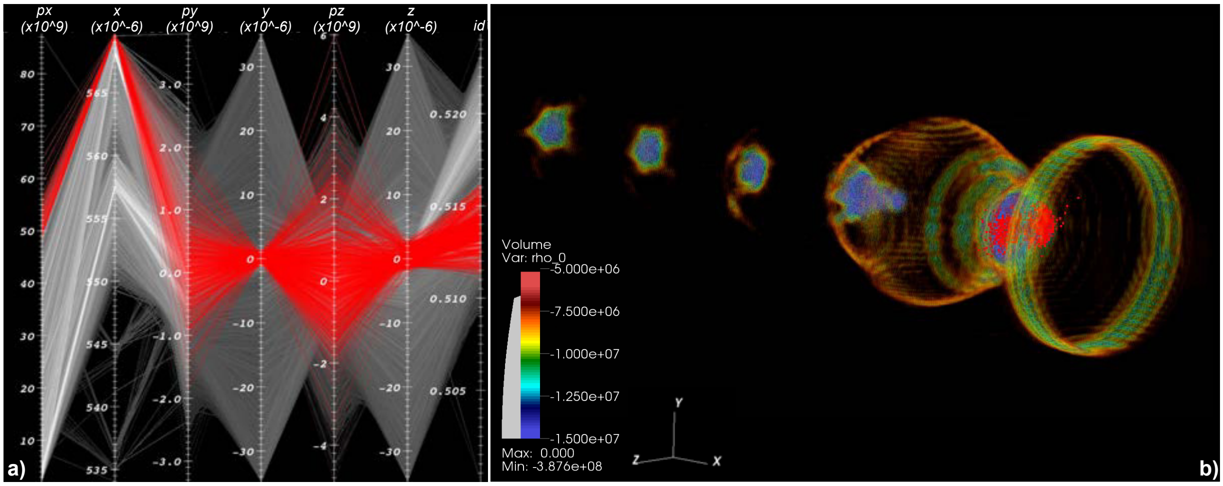


Figure 2: Query-driven visualization and analysis of a large 3D plasma-based particle acceleration data containing $\approx 90 \times 10^6$ particles per timestep. On the left (a), parallel coordinates of timestep $t = 12$ showing: (1) all particles above the base acceleration level of the plasma wave (Query: $px > 2 \times 10^9$) (gray) and (2) a set of particles that form a compact beam in the first wake period following the laser pulse (Query: $(px > 4.856 \times 10^{10})AND(x > 5.649 \times 10^{-4})$)(red). On the right (b), volume rendering of the plasma density illustrating the 3D structure of the plasma wave. The additional selected beam particles are shown in red. Image source: Rübel et al., 2008 [7].

how these beams evolve over time. These questions are the subject of work by Rübel et al, 2008 [7].

To identify those particles that were accelerated, the authors, who included two accelerator physicists, performed an initial threshold selection in px at a late timestep of the simulation. This initial selection restricts the analysis to a small set of particles with energies above the phase velocity of the plasma wave. Figure 2a shows an example of a parallel coordinates plot of such a selection (gray).

Based on the results of the first query, the selection was further refined to select the main particle beams of interest. The authors first increased the px threshold to extract the particles of highest energy. This initial refinement often results in the selection of multiple beam-like features trapped in different periods of the plasma wave.

As illustrated in Figure 2a (red lines), to separate the different particle beams and to extract the main particle beam, the selection is then often further refined through range queries in the longitudinal coordinate x and transverse coordinates y and z. In the selection process, parallel coordinates provide interactive feedback about the structure of the selection, allowing for fast identification of outliers and different substructures of the selection. High performance scientific visualization methods are then used to validate and further analyze the selected particles shown in Figure 2b.

The next step was to trace, or follow, and analyze the high-energy particles through time. Particle tracing is used to detect the point in time when the beam reaches its peak energy and to assess the quality of the beam. Tracing the beam particles further back in time, to the point at which the particles enter the simulation window, supports analysis of the injection, beam formation, and beam evolution processes. Based on the information from different individual timesteps, a user may refine a query, to select, for example, beam substructures that are visible at different discrete timesteps.

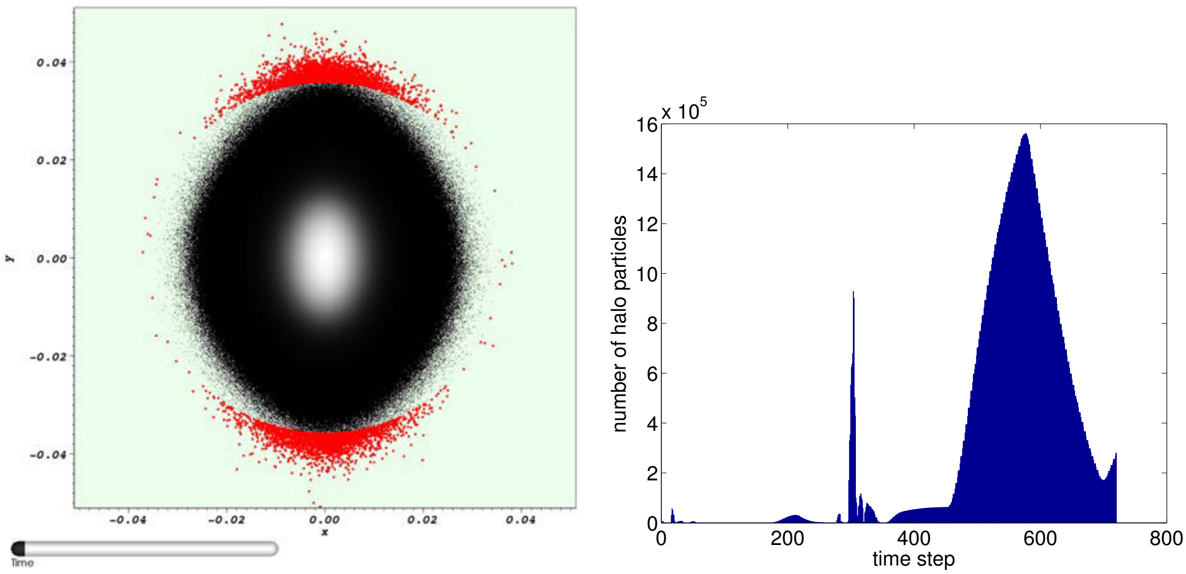
This particular study by Rübel et al., 2008 [7], which included scalability tests out to 300-way parallel in 2008, reduced the time required to track and analyze 300 accelerated particles from hours, using legacy serial processing tools, to less than one second. Whereas the Rübel et al., 2008 study demonstrated tracking tens to millions of particles, due to the mechanics of its internal algorithm, the legacy serial processing tool would not be practical for use on increasingly larger problem sizes.

Linear Accelerators

This case study focuses on visual data exploration and analysis of output from large-scale, high resolution simulations of beam dynamics in electron LINACS for a proposed next-generation Xray free electron laser (FEL) at Lawrence Berkeley National Laboratory (LBNL) [10]. Particle-in-cell-based simulations of this type of accelerator require large numbers of macroparticles ($> 10^8$) to control the numerical macroparticle shot noise and to avoid the overestimation of microbunching instability, resulting in massive particle data sets [11].

Chou et al, 2011 [12] focus on studying the transverse halo and the longitudinal core. In both cases, the criteria a given particle satisfies to be classified as "halo" or "core" is a spatial one. Prior to this work, the typical workflow pattern would be to run a serial analysis tool on each timestep, where the serial code would examine each particle in turn, and evaluate whether or not it satisfied the given criteria. This process is $O(N)$ in computational complexity, given N particles per timestep. But, like the LPA example before,

ISBN 978-3-95450-116-8



(a) Particle density plot (gray) and particles selected (red) by the halo query for timestep 20 of the simulation.

(b) Plot showing the number of halo particles vs. timestep.

Figure 3: Particles of the transverse halo shown in red from a single simulation timestep identified using a spatial condition query (a). The number of halo particles (b) increases over time, indicating a potential problem with this particular accelerator design. Image source: Chou et al., 2011 [12].

this process is dramatically accelerated by using advanced index-query infrastructure: that infrastructure accelerates the processing time by avoiding the examine-every-particle approach and replacing it with one where only those particles that are likely to satisfy the condition are examined.

The authors identify halo particles using the query $r > 4\sigma_r$. The derived quantity $r = \sqrt[2]{x^2 + y^2}$, which is separately computed and indexed, describes the transverse radial particle location. The transverse halo threshold is given by $\sigma_r = \sqrt[2]{\sigma_x^2 + \sigma_y^2}$, where σ_x and σ_y denote the root mean square (RMS) beam sizes in x and y, respectively. Using r for the identification of halo particles is based on the assumption of an idealized circular beam cross section. To ensure that the query adapts more closely to the transverse shape of the beam, one may relax the assumption of a circular beam cross section to an elliptical beam cross section, by scaling the x and y coordinates independently by the corresponding RMS beam size, σ_x and σ_y using, for example, a query of the form $r_s^2(x, y) > 16$, with $r_s^2(x, y) = (\frac{x}{\sigma_x})^2 + (\frac{y}{\sigma_y})^2$. Halo particles identified with this methodology, for one timestep, are shown in Figure 3a. Details of the definition and analysis of the beam core are in the original study [12].

Figure 3b shows the number of halo particles per timestep identified by the halo query. It shows large variations in the number of halo particles, while in particular, the larger number of halo particles at later timesteps are indicative of a possible problem. In this case, the halo particles and observations of an increase in the maximum particle amplitude were found to be due to a mismatch in the beam as it traveled from one section of the accelerator to the next.

These types of query-based diagnostics provide accelerator designers with evidence that further improvement of the design may be possible, and also provides quantitative information that is useful for optimizing the design to reduce halo formation and beam interception with the beam pipe, which will ultimately improve accelerator performance.

The authors used the FastQuery [12] system for the parallel processing of index and query operations on the NERSC Cray XE6 supercomputing system Hopper. The pre-processing time to build the indexes for all timesteps of the 50TB data set was about 2 hours. By using bitmap indexing, evaluating queries for finding the halo and core required only 12 seconds with 3000 cores, or around 20 seconds with 500 cores. The combination of visual data exploration and efficient data management in the query-driven visualization concept enables, in this way, repeated, complex, large-scale query-based analysis of massive data sets, which would otherwise not be practical, with respect to both time as well as computational cost.

LESSONS LEARNED

These case studies, each of which are field-leading in their own right, all have several themes in common.

Searching, querying. First, all studies involve some element of searching for data with specific features or characteristics. Over the years, this particular topic has been a subject of active research in the scientific data management community. By searching scientific data using the motifs presented here, which consist of high-dimensional, read-only query operations, the bitmap index has proven to be a useful and appropriate indexing structure [13]. The ac-

ISBN 978-3-95450-116-8

celerator modeling case studies here make use of FastBit, which is an Open Source, high performance compressed bitmap indexing implementation [14].

Data I/O. Increasingly, the cost of performing I/O from simulation codes dominates the analysis cycle. In the case of the LINAC case study, the accelerator scientists had never before seen the results of later simulation timesteps due to the inability to write out that much simulation data. This work was made possible through a partnership between computer and computational scientists whereby the IMPACT-T simulation code was fitted to use a high performance parallel I/O library. Looking further ahead, many in the field expect an increasing amount of visualization and analysis to be performed *in situ*, while particle and field data are still resident in-core inside the simulation.

Integrative technologies. In our work, we are finding that it is increasingly the case that disparate technologies—simulation codes, data management, analysis, and visualization infrastructure—must be designed and engineered to work together as a whole. It is increasingly unlikely that these technologies can be considered and used in isolation from one another as design choices for one can have a profound impact on others. This situation is exacerbated by increasingly large and complex data sets and more sophisticated lines of scientific inquiry.

ACKNOWLEDGMENT

This work was supported by the Director, Office of Science, Office and Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231, and used resources of the National Energy Research Scientific Computing Center (NERSC), which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

REFERENCES

[1] Richard Mount. The Office of Science Data-Management Challenge. Report from the DOE Office of Science Data-Management Workshops. Technical Report SLAC-R-782, Stanford Linear Accelerator Center, March-May 2004.

[2] Hank Childs and Mark Miller. Beyond Meat Grinders: An Analysis Framework Addressing the Scale and Complexity of Large Data Sets. In *SpringSim High Performance Computing Symposium (HPC 2006)*, pages 181–186, 2006.

[3] Hank Childs, David Pugmire, Sean Ahern, Brad Whitlock, Mark Howison, Prabhat, Gunther Weber, and E. Wes Bethel. Extreme Scaling of Production Visualization Software on Diverse Architectures. *IEEE Computer Graphics and Applications*, 30(3):22–31, May/June 2010.

[4] Mark Howison, E. Wes Bethel, and Hank Childs. Hybrid Parallelism for Volume Rendering on Large, Multi- and Many-core Systems. *IEEE Transactions on Visualization and Computer Graphics*, 99(PrePrints), 2011.

[5] Kurt Stockinger, John Shalf, Kesheng Wu, and E. Wes Bethel. Query-Driven Visualization of Large Data Sets. In *Proceedings of IEEE Visualization 2005*, pages 167–174. IEEE Computer Society Press, October 2005. LBNL-57511.

[6] E. Wes Bethel, Scott Campbell, Eli Dart, Kurt Stockinger, and Kesheng Wu. Accelerating Network Traffic Analysis Using Query-Driven Visualization. In *Proceedings of 2006 IEEE Symposium on Visual Analytics Science and Technology*, pages 115–122. IEEE Computer Society Press, October 2006. LBNL-59891.

[7] Oliver Rübel, Prabhat, Kesheng Wu, Hank Childs, Jeremy Meredith, Cameron G. R. Geddes, Estelle Cormier-Michel, Sean Ahern, Gunther H. Weber, Peter Messmer, Hans Hagen, Bernd Hamann, and E. Wes Bethel. High Performance Multivariate Visual Data Exploration for Extemely Large Data. In *Supercomputing 2008 (SC'08)*, Austin, TX, USA, November 2008. LBNL-716E.

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

[9] Prabhat, Oliver Rübel, Surendra Byna, Kesheng Wu, Fuyu Li, Michael Wehner, and E. Wes Bethel. 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)*, Omaha, NE, June 2012. LBNL-5352E.

[10] J. N. Corlett, K. M. Baptiste, J. M. Byrd, P. Denes, R. J. Donahue, L. R. Doolittle, R. W. Falcone, D. Filippetto, J. Kirz, D. Li, H. A. Padmore, C. F. Papadopoulos, G. C. Pappas, G. Penn, M. Placidi, S. Prestemon, J. Qiang, A. Ratti, M. W. Reinsch, F. Sannibale, D. Schlueter, R. W. Schoenlein, J. W. Staples, T. Vecchione, M. Venturini, R. P. Wells, R. B. Wilcox, J. S. Wurtele, A. E. Charman, E. Kur, and A. Zholents. A Next Generation Light Source Facility at LBNL. In *Proceedings of PAC 2011*, New York, USA, April 2011.

[11] J. Qiang, R. D. Ryne, M. Venturini, and A. A. Zholents. High Resolution Simulation of Beam Dynamics in Electron Linacs for X-Ray Free Electron Lasers. *Physical Review*, 12(100702):1–11, 2009.

[12] Jerry Chou, Mark Howison, Brian Austin, Kesheng Wu, Ji Qiang, E. Wes Bethel, Arie Shoshani, Oliver Rübel, Prabhat, and Rob D. Ryne. Parallel Index and Query for Large Scale Data Analysis. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 30:1–30:11, Seattle, WA, USA, November 2011. LBNL-5317E.

[13] Arie Shoshani and Doron Rotem, editors. *Scientific Data Management: Challenges, Technology, and Deployment*. Chapman & Hall/CRC Press, Boca Raton, FL, USA, 2010.

[14] Kesheng Wu, Sean Ahern, E. Wes Bethel, Jacqueline Chen, Hank Childs, Estelle Cormier-Michel, Cameron Geddes, Junmin Gu, Hans Hagen, Bernd Hamann, Wendy Koegler, Jerome Lauret, Jeremy Meredith, Peter Messmer, Ekow Otoo, Victor Perevoztchikov, Arthur Poskanzer, Prabhat, Oliver Rübel, Arie Shoshani, Alexander Sim, Kurt Stockinger, Gunther Weber, and Wei-Ming Zhang. FastBit: Interactively Searching Massive Data. In *SciDAC 2009*, 2009. LBNL-2164E.

ISBN 978-3-95450-116-8