Report on the Big Data and Extreme-scale Computing (BDEC) Workshop, Charleston, SC, USA, April 29-May1, 2013

1 Introduction

This report on the Big Data and Extreme-scale Computing (BDEC) workshop offers an initial account of the effort to develop a plan for sustained international cooperation in the design and development of a new generation software infrastructure for extreme scale science. The meeting, the first of a planned series, derived much of its impetus from the earlier work of the International Exascale Software Project (IESP)[1] and European Exascale Software Initiative (EESI)[2]. The goal of the IESP was two-fold: 1) to produce a plan for a common, high quality computational environment for the peta/exascale systems that are expected to arrive over the next decade; and 2) to mobilize and coordinate the work of the international open source software community to create that environment. BDEC retains those goals but changes the point of view.

The IESP, working through a series of eight international meetings held from 2009 to 2012, built on a range of important earlier studies, including [3-6], to produce a widely read and cited "roadmap" document. The IESP Roadmap [7] presented a multidimensional analysis of the major challenges to be overcome in order to create a software infrastructure capable of supporting exaflop performance on next generation systems, and made a cogent case for the urgency of starting that work as soon as possible. Spurred in some degree by the work of the IESP and its Roadmap, the United States, the European Union, and Japan have, in the past three years, moved aggressively to develop their own plans for achieving exascale computing in the next decade.

In the EU, for example, EESI worked with roughly 100 experts in 2010 - 2011 to develop an Exascale vision and roadmap [8], which was presented in October 2011. This has been followed by a second initiative, EESI2 (Sept. 2012 - 2015), which is more dedicated to proposing concrete R&D program recommendations attacking the key issues identified by the IESP and EESI [9]. Some of these issues were flagged as needing concerted collaborations that are global in scope, and one of the tasks of EESI2 is to further develop an international exascale software community in order to achieve that end. As this report makes clear, such global cooperation is essential to the goals of the BDEC as well.

The first BDEC workshop marks the beginning of a distinct new phase of this community movement. The motivation for this second stage is based on the recognition that the "digital data deluge," which was sighted on the horizon well over a decade ago [10], has finally made landfall with impressive force. It is apparent that in the era of "Big Data," when every major field of science and engineering is producing, and needs to (repeatedly) process, truly extraordinary amounts of data, the many unsolved problems surrounding ***wide-area, multistage workflows****—the diverse patterns of when, where, and how all that data is to be produced, transformed, shared, and analyzed*—have to take center stage. Although the IESP and EESI roadmaps shows a clear awareness that extreme scale science inevitably means extreme scale data as well as extreme scale computing, IESP and EESI working groups, for the most part, adopted a traditional HPC (i.e., supercomputer centric) perspective. They were largely (and understandably) preoccupied by the impending software crisis caused by the move to the new paradigm in hardware and systems architecture, a paradigm that demands orders of magnitude more parallelism, places unprecedented constraints on energy consumption, and requires resilience to faults occurring at far higher frequencies than ever before. Data-driven workflow issues received some collateral discussion in the Roadmap, but the focus of attention for the IESP was on the revolutionary innovations in the system software stack that would be needed to address the steep challenges of emerging peta/exascale systems.

The BDEC effort carries forward the general mission of the IESP—*the co-design of software infrastructure for extreme scale science drawing on international cooperation and supporting a broad spectrum of major research domains*; but it reframes the problems involved to fully take account of varied (and evolving) workflow patterns that such different communities of inquiry must create to work with *both data and computing resources that are unprecedented in their scale*. All the imposing design and development issues of creating an exascale-capable software stack remain. But the supercomputers that need this stack must now be viewed as the nodes (albeit the largest and perhaps the most important nodes) in a very large network of computing resources that will be required to generate, collect, manipulate, transform, analyze, and collaboratively explore gigantic mountains of data. Thus, adding big data to extreme computing will require a complete rethinking of the scientific software stack.

As the report below shows, the first BDEC workshop made substantial progress in achieving this necessary change in perspective. Leaders from the HPC software research and infrastructure communities joined major representatives of science domains that are both data and compute intensive to begin to assess the challenges and opportunities for international cooperation on the co-design (and co-development) of software infrastructure for extreme-scale science. As its goal statement makes clear, four leading ideas—*software infrastructure, architecture, co-design, and international collaboration*—framed the context for presentations and discussions that took place at the workshop. Consequently, some brief comments on each of these ideas at the outset may facilitate and improve the reader's understanding of the report on the workshop activities and results that follow.

1.1 Software infrastructure as community focus

The belief that good software infrastructure is absolutely fundamental to a healthy and productive scientific research ecosystem was an important guiding idea for the work of the IESP community; it continues to occupy that central position within the BDEC. The use of the term 'infrastructure' in English is less than a century old, and did not come into use in non-military contexts until the 1970s. Unsurprisingly, therefore, 'software infrastructure' gained currency only in the last decade, most probably because it is an obvious subclass of 'cyberinfrastructure,' which was introduced via an NSF Blue Ribbon Panel around the beginning of the century [11].

Recent developments in the economic analysis of the concept of infrastructure and its social value [12] sheds light on why software infrastructure is so fundamental to the scientific community and so provides useful context for the work of a BDEC meeting. Frischmann nominally defines infrastructure resources as "… intermediate capital resources [that] serve as critical foundations for productive behavior within economic and social systems." He explicates this definition in terms of three essential characteristics, which, when applied to "software infrastructure," clarify its importance for the scientific research ecosystem:

- *Nonrivalrous consumption* —Software infrastructure "… may be consumed *nonrivalrously* for some appreciable range of demand," i.e., its "… consumption by one person does not reduce the consumption opportunities of any other person." (p. 12) Processor resources can also be shared, of course, but they lie closer to the rivalrous end of the sharing spectrum and have to be managed (via software) accordingly. Software, by contrast, is less like apples and more like ideas. Absent intellectual property constraints, anyone with the appropriate hardware can freely use a given piece of software without affecting the ability of others to do the same. This “extra degree of freedom” that software enjoys, as something that exists in replica, allows its availability and use to be managed as a commons; and this, in turn, dramatically increases the value it can deliver to the entire community by way of *positive externalities* and *spillover effects* of various kinds.

- *Derived value* —Since “social demand for [software infrastructure] is driven primarily by downstream productive activity that requires the resource as an input,” it has *derived value.* That is, software infrastructure for computational science is a means, not an end in itself. Instead, it provides a durable foundation of essential functionality that scientists and engineers can, by adding their own resources, build on to achieve narrower and more specific goals. All the typical components of the scientific software stack (e.g., operating systems, numerical libraries, communication libraries, runtime schedulers, etc.) exhibit this kind of derived value because they are all basically facilities (or parts of facilities) that enable science and engineering applications that are further “downstream.”
- *Generic value* —Finally, the value of good software infrastructure is not only derived, it is also generic, i.e., “[it] may be used as an input into a wide range of goods and services, which may include private goods, public goods, and social goods.” Even a high-level, summary review of the history of computing and networking shows that the software infrastructure created to enable basic science and engineering has been used to support a remarkable range of applications in nearly every sphere of social and economic activity. Such a review would show, in general, that the more generic a given piece of functionality is in this sense, the more important it is to provide for it in the form of good (and sustainable) software infrastructure, and correspondingly, the more difficult it is to create an architecture for such software that also maximizes the field of potential impact.

For BDEC, issues surrounding software for storage infrastructure acquire an extra degree of difficulty because, generally speaking, storage infrastructure is still largely balkanized and is usually designed to fall toward the rivalrous end of the sharing spectrum.

1.2 Architecture as community perspective

Two main factors combine to incline the BDEC community toward an architectural point of view, as opposed, for example, to a narrower “technical problem solving” perspective. In the first place, the focus of the community on infrastructure means that, as noted above, what is being considered and developed is something with derived value that is supposed to be highly generic, i.e., capable of serving a wide range of purposes. As with planning for a major public building or public space, the architectural approach to creating a large supercomputer, or its software stack, concentrates on the question of what *profile of tradeoffs among fundamental features* will best serve the widest array of goals of the prospective user community. It avoids, as much as possible, any pre-optimization of its basic structure for some special use case that would substantially decrease its value for the majority of other uses. This is especially appropriate in the case of software infrastructure, which can be shared nonrivalrously among an immense user community, and where users need to be free to build and cooperate on top of this base to meet their unique requirements.

The second factor determining an architectural perspective is both more historical and more dramatic: The recent convergence of revolutionary new design constraints on large scale HPC systems, combined with the emergence of the era of Big Data across many scientific domains, has produced a very broad consensus that much of the essential software infrastructure of computational science and engineering is, or will soon be, obsolete. Since we know, from decades of experience, that the fundamental design decisions that are "cooked in" now will, for better or worse, substantially affect the ease (or difficulty) with which future researchers can explore one path as opposed to another, sound architectural thinking has never been more essential than it is now. The IESP took this point of view with regard to planning for a new software stack for peta/exascale systems; the BDEC continues it for the expanded problem, where large, data-intensive workflows that start from or move through these systems also need to be taken into account.

1.3 Co-design as community process

Looking forward to the technical challenges that the scientific community faces as it begins the march toward exascale computing, the IESP contributed substantially to the acceptance of 'co-design' as an important organizing idea for that effort [7]. Originating in the world of embedded systems, 'co-design' historically referred to the "… simultaneous design of both hardware and software to implement in a desired function;" it is taken for granted in this definition that there are multiple stakeholders, on the different sides of the design process, with distinct and conflicting points of view on how the design tradeoffs—needed to jointly satisfy all the given design constraints—should be made. The IESP was somewhat unique in bringing together leading representatives from the scientific software infrastructure research community, computationally intensive science domains, NSF and DOE leadership facilities, major HPC hardware vendors (e.g., Intel, Cray, IBM, HP, NVIDIA, etc.); and various funding agencies (e.g. NSF, DOE, ANR, EPSRC, INRIA, etc), so perhaps it is not surprising that a generalized form of the concept of co-design should bubble up to the surface as a way to structure deliberations.

There are good reasons to think that the BDEC effort will need to go even farther in that direction, seeking to re-conceptualize and implement co-design as a *community process*. In the first place, the deeper understanding that we are acquiring about how to analyze and evaluate software infrastructure, alluded to above, makes it clear that the number of different classes of stakeholders, and the number different dimensions along which design tradeoffs will likely need to be made, are both large. Second, the problem that the BDEC aims to confront—software infrastructure for workflows that are both data and computationally intensive—adds a (wide-area) network dimension to the problem that IESP confronted, which focused primarily on creating a common software stack for extreme-scale systems. The problem of designing software infrastructure for scientific workflows is inherently more technically, socially, and economically complex. This is especially so if, in contrast to the Large Hadron Collider Computing Grid, it needs to be as agnostic as possible with respect to research domains, in order to maximize the potential for joint investment and derived value. Finally, some forms of data (e.g., ecological, climatological, sociological) are inherently historical, which means that the BDEC will, sooner or later, need to address issues of software infrastructure for long-term preservation. If "long-term" means anything more than 50 years, there are currently no good (confidence inspiring) solutions to the problems that seeking long-term data preservation introduces.

1.4 International collaboration as community scope

The reasons for embracing the perspective of a cooperative international community are as straightforward for the BDEC as they were for the IESP, perhaps more so. First, experience shows that the creation of new, high quality software infrastructure for research, capable of meeting the unprecedented requirements of extreme scale science, will demand investment on a scale that no single country, working in complete independence, is likely to be able to make. In his white paper for the BDEC meeting, Dan Reed notes that, “The cost of some scientific instrumentation long ago outstripped the resources of institutions and even some countries, leading to national and international consortia. Today, data volumes now exceed the capabilities of many individual research teams to host and manage, and they are increasingly challenging the capabilities and resources of research institutions and even national research agencies.” [13] BDEC is premised on the idea that this same kind of economic necessity applies to the creation of the scientific software infrastructure to support a research ecosystem through which incredibly large flows of data are moving. Research leaders in the European Union and Japan used the IESP to help coordinate their own exascale software investments with those in the United States, and they show every sign of being prepared to work in the same way with the BDEC on software infrastructure for extreme-scale data.

Second, as already noted, software infrastructure (as well as the scientific data it helps to generate) can be shared in a completely nonrivalrous way. This means that one collective, multi-national investment can serve a very broad spectrum of science and engineering communities, whether they are national or international in scope. But achieving this result will require implementing an open and participatory co-design process in which the requirements and preferred tradeoffs of various computational science communities worldwide are taken into account.

Finally, collaboration in many scientific domains is increasingly transnational, closely coupled, and interdisciplinary; new software infrastructure for extreme scale science must be responsive to this fact. As the report on the BDEC meeting below will show, the instruments and simulations that produce the essential data of interest, the computational resources to be applied to it, and the multi-disciplinary teams who must bring it all together, are distributed around the globe. A second premise of the BDEC is that, because of the international and interdisciplinary character of so many different research domains and transnational scientific workflows, the software research and development community that supports it needs to be coordinated globally as well.

2 Background: Domain Science Representation

As the first in a projected series of meetings that aims to understand what the combination of big data and extreme computing will mean for the software infrastructure of tomorrow’s research ecosystem, this meeting was designed to be formative and exploratory. Like the IESP series, the attendees included leading HPC architects and researchers from major computing centers in the US, Europe, and Japan, software infrastructure experts from both academic and government laboratory communities in those countries, as well as large HPC system vendors. But by the nature of its mission, the BDEC is far more focused on data-driven workflows than its predecessor. Moving data-driven research to the center of attention meant recruiting leaders from research domains who could begin to describe and help analyze some of the diverse, multi-stage workflows, ranging from wide area resource networks to supercomputer machine rooms, that have already emerged over the past few years. This meeting made credible progress toward that goal.

The variety of domain sciences represented at the BDEC meeting in Charleston offered a diverse set of interesting and illuminating cases. As part of the ongoing Square Kilometre Array project (SKA), Tim Cornwell, from the Australian Commonwealth Scientific and Industrial Research Organization (CSIRO), discussed the numerous challenges associated with running a radio telescope facility installed on two continents while sharing computing and storage resources. Also present at the meeting was Shinichi Morishita from the University of Tokyo, who represented the field of Genomics and outlined the massive amounts of data produced by DNA sequencing and the computing resources required to produce a meaningful and timely analysis of a genome. Weather and climate modeling, a long-time user and producer of big data, was represented by Pier Luigi Vidale, University of Reading, who outlined challenges in architecture, workflows, and software required to simulate days to months, years, decades, and millennia of local, regional, continental, and global climates. Additionally, Giovanni Aloisio and Sandro Fiore, University of Salento and the Euro-Mediterranean Centre on Climate Change discussed the problems of interoperability involved in the management of data provenance and other metadata for emerging platforms doing (near) real-time analysis of climate change data. Bioinformatics is another hot topic in big data, and Emory University's Joel Saltz describes the use of multi-dimensional datasets where different instruments observe and collect data from the same subject. The data are then combined for integrated analysis, which allow for a multi-faceted view of the pathology of a brain tumor, for example. Solid earth sciences, which includes collecting and analyzing data on seismic waveforms and seismic noise used for things like earthquake risk analysis and nuclear test monitoring, relies on immense amounts of data, and was represented by Jean-Pierre Vilotte from the French National Center for Scientific Research (CNRS). All of the domain scientists present at the BDEC workshop provided valuable insight into the similarities in challenges to workflow, data logistics, architecture and software design, infrastructure, and more. A series of regular meetings, like the one in Charleston, would undoubtedly help the international computational science community coalesce and collaborate in an effort to better understand these challenges and how they can be solved.

2.1 What does "Big Data" mean?

It is widely recognized that "Big Data" (BD) is polysemic, and not in a good way. Consequently, BDEC participants gave some attention to the question of what the expression should mean, although it is clear that a useful and well-bounded definition will continue for some time to be a work in progress. Part of the problem is that, like familiar alternatives, such as "data intensive," what counts as BD is relative to other factors, and therefore changes depending on the perspective—processor, memory, bandwidth, storage—from which it is being viewed. Straightforward examples of BD applications in this sense are applications that take all of a supercomputer's memory or more, or that are too complex to process because the relation between computation and data size is non-linear, or that have real-time processing requirements the velocity of which exceeds the I/O bandwidth.

Generally speaking, there are very few large-scale applications of practical importance that are not data intensive when looked at from some relevant point of view. All of the applications discussed at the workshop, whether the data comes from new instruments, from massive simulations, or from distributed sensors, deliver eye-glazing quantities of data at unprecedented rates. From an applications perspective, however, discussions of BD have greatly increased the prominence of "data-driven" applications (such as data analytics, top-down queries and predictive modeling), where the operations are defined and propelled not only by large data volumes and data streams, but also by the complexity or heterogeneity of the data involved. Although researchers have been successful for some time in processing computer-generated, semi-structured data (big simulations) and structured observational data (big instruments), they are now more eager to take on the challenges of high volumes of unstructured and heterogeneous observational data (text, images, medical records, etc.), which often come in massive piles of small units and are asynchronously generated. So in that way, BD is redefining the HPC application landscape.

2.2 Workshop Process

The BDEC workshop was structured around a process that the community developed for meetings that are (of necessity) more exploratory in character. Since the consequences of the convergence of big data and extreme-scale computing are just beginning to emerge, and are therefore largely unknown, this meeting required such an approach. Working with the steering committee, the organizers first developed a list of suggested topics and issues (see Appendix 4) that participants were offered as plausible items for analysis and discussion, both by the contributors of white papers and those invited to give plenary presentations. Some participants addressed all the different areas—workflows, architecture, high-level data issues, software, and interoperability—while others were more selective in choosing topics and issues that fit more closely with their specialty.

The context for the meeting was partially filled out by fourteen short white papers that were submitted in advance of the meeting and distributed to all the attendees. All the presentations and whitepapers, with links to on-line documents, are listed in Appendices 1 and 2, respectively. With that framing of the issues in place, the meeting was segmented and arranged as follows:

- *Plenary presentations*: Both days of the workshop began with leaders in HPC architecture and in various domain sciences offering plenary presentations describing developments in their fields that revealed the impact of large, data-driven workflows. The goal of these presentations was to provide insights and points of discussion for the breakout sessions in the afternoon.
- *Panel presentation by white paper authors*: Each of the afternoon sessions began with a panel on which the authors of the workshop white papers briefly summarized and commented on their white papers. As above, these panels set the stage for the daily breakout sessions where central issues discussed intensively.
- *Breakout sessions for open discussion of the different topic areas*: The major segment in each afternoon was dedicated to breakout sessions in which participants with expertise or interest in a given area would gather to discuss and produce recommendations about that area. Over the two days, all the main issues—workflows, architecture, high-level data issues, software, interoperability—were addressed by separate breakout groups, each of which had both a discussion leader and a scribe.
- *Summary presentations of the results of the breakouts*: Each day concluded with plenary presentations of notes from the breakout sessions.

Standard practice at meetings of this community is to put all presentations online in near real-time, so that participants have access to each presentation that is given shortly after it is given, normally within half an hour. The goal of this practice is to try ensure that the conversation gets increasingly richer and better informed as the meeting proceeds. All the presentations and whitepapers, with links to on-line documents, are listed in Appendix 2.

3 Summary of Workshop Results

As noted above, the organizers developed a list of suggested topics and issues (see Appendix 4) for participants to address in white papers and plenary presentations. Some participants addressed all the different areas—workflows, architecture, high-level data issues, software, and interoperability—while others were more selective in choosing topics and issues that fit more closely with their specialty. Our presentation of results below follows that division of topics and issues.

3.1 Workflow Issues

For the purposes of this report, we define a "workflow," roughly speaking, as a repeatable pattern of actions, or processing steps, that has been organized to accomplish a given purpose under a given set of constraints. In today's society, the processing of digital information has become such a routine part of life that the general idea of creating digital workflows, in this generic sense, increasingly pervades even discussions of personal productivity in popular media. As various white papers and presentations from the BDEC workshop make clear, the concept of workflows (more precisely defined and more thoroughly analyzed) will also dominate much of the thinking about cyberinfrastructure for all kinds of research in the era of data-driven science. The problems inherent in working with data that are streaming out of instruments and simulations at peta- or exabyte rates, or of integrating and analyzing massive, multi-dimensional data sets, are simply too difficult for things to be otherwise. In terms of challenges to workflow, many domain sciences that produce and manage BD share common constraints. Below we sample some issues raised by BD workflows that were particularly prominent at the workshop.

A common way of representing workflows: One thing that many of the workshop discussions made clear was that there is no standard way of describing or representing workflows, and no abstraction to facilitate thinking about workflows in a way that supports well structured comparisons between them. It is clear the community will need time and experience to develop such a common workflow abstraction, and then use that abstraction to describe and classify them. Nonetheless, the meeting offered illustrations of the various ways that digital workflows are currently being characterized/diagrammed and sorted using available (relatively vague) criteria.

For example, workflows that are implemented within a single supercomputer or Internet Data Center (IDC) (e.g., data assimilation in weather prediction and genome analysis) can be characterized as "tightly coupled," since in such controlled environments, stronger guarantees can be given, and hence stronger operational assumptions can be made. By contrast, Grid workflows (e.g., from large instruments like the Square Kilometer Array (SKA) and the Large Hadron Collider (LHC)) tend to be more "loosely coupled," since they traverse the wide area network and multiple administrative domains, and consequently allow for weaker control, and support weaker assumptions.

Workflows are also differentiated by the rate, continuity, and periodicity with which data is injected into them. For example, the LHC, SKA, and many other large instruments that take their data from an analog source, produce massive amounts of data in a constant stream when they are active. These instruments are so sensitive, and Nature is so inexhaustible, that their workflows are dominated by data reduction operations. They often adopt a "waterfall" model for their workflow, in which the data goes through successive stages of refinement and lossy compression, and the stages get increasingly complex (e.g., editing, calibration, imaging, source finding, etc.) as the data moves through the process. Once a substantial segment of the input data is reduced to a "working set," it may have to be moved, but then it will typically (if not invariably) stay resident near the compute resource(s) for days or weeks, while it undergoes analysis. Such leading edge cases suggest that, as we move toward exascale, the widening gap between data movement and computational rates will make in-situ analysis and visualization more of a necessity.

Contrasting with instruments that receive continuous BD inputs from nature, traditional HPC has developed the "simulation-as-laboratory" model. In this approach, large simulations (e.g., climate, earth systems, cosmology, or supernovae) generate immense forward looking workflows (moving from HPC simulation to HPC analysis) in which data is produced by the application in time steps, and is dumped into the pipeline at each successive mark (e.g., every 10 minutes). In some cases, researchers want to perform analysis in real time, while the model is running, to make it possible to steer the stimulation, diagnose problems, or reduce output/transmission costs at the source; in other cases, the data analysis involves integrating elements coming from both the simulation and from a database containing observations from real events. Finally, remote sensing and “omic” (e.g., genomic, proteomic, etc.) applications may represent yet a third type workflow, distinguishable by input patterns, in which the workflow takes in successive but discrete collections of sequence data or satellite images that have been generated asynchronously.

Still a third aspect or characteristic which a workflow taxonomy might take into account focuses on the question of how automated or interactive the workflow is. Workflows that are dominated by reduction operations, such as mentioned above, can be prefabricated and carried out in a highly automated way. But there are various kinds of tasks (e.g., visualization, annotation, analysis, etc.) that involve workflows that usually require far more dynamic engagement with the user as they progress.

Although each of these characteristics may be relevant to a comprehensive taxonomy of BD workflows, natural classifications of artifacts typically begin by examining the purpose (or purposes) that the artifact in question was designed to fulfill. Certainly, generic aims of research, like “analytics” and “exploration,” count as relevant goals that shape and structure the character of workflows. But, absent a better analysis than we have at present, these concepts seem both too general and too vague as they currently stand to provide the basis for developing a proper taxonomy. Further examination of particular workflow “specimens” might, in the future, help to provide a better analysis of the relation between the structure of the workflows and the research aims they are intended to serve.

As noted in the software section below, workshop discussions of workflows revealed at least two clear gaps in software infrastructure: less than adequate coordination between the data movement services and the compute services; and tools that would allow researchers to observe and monitor the workflow in real-time.

Data logistics: One way to conceptualize a common problem brought up by numerous BDEC participants is to think of it as a problem with “data logistics,” i.e., as a problem of managing the time sensitive positioning of data relative to its intended users and the resources they need to utilize. At a minimum, all the international communities that want to analyze data that is generated at one (or a few) locations, but is worked on somewhere else, confront challenges of data logistics in this sense. Both the examples of the Square Kilometer Array (SKA) and the climate modeling communities, (e.g. through the Earth System Grid Federation) illustrate how managing the movement and staging of data, from where it is collected to where it needs to be processed, can take up most of the time to solution. Researchers have long aspired to stream data across a network into running parallel applications, the only generally successful method of data streaming that is widely used, after two decades of trying, still involves simple linear data (e.g., media files) and rote sequential processing at the receiver. Indeed, the well known “tiered” organizational structure of the LHC grid shows that such logistical problems with BD are so significant for these types of applications that they can play a major role in structuring the way in which the entire community is organized.

Arguably, however, the concept of data logistics can be applied much more broadly. At the root of the problem is the fact that data and the computing resources needed to process it have to be co-located in order for work to proceed. But absent the capacity to stream, “co-located” does not mean somewhere on the LAN. From the point of view of “time to solution,” the challenges of just moving massive data objects into and out of the memory of HPC systems can also be characterized as logistical in nature. Hence, one can think of data logistics as defining a continuum, with I/0 issues inside the IDC or supercomputing facility falling at one end, and BD workflows that begin at locations (possibly distributed) that are remote from the HPC system(s) targeted for the workload, falling at the other.

Obstacles to improving data logistics are numerous and fall at several levels. Although online storage and data transfer services like Globus online have had significant success in helping users (and distributed collaborations) manage their data logistics within the research data lifecycle (e.g., moving more than 16 PB as of April 2013), the fact that data is growing at exponential rates, while bandwidth and storage capacity are approaching technical limits, means that current solutions are temporary at best. Moreover, the future of data logistics is likely to be complicated by the fact that issues of energy usage are an increasingly important factor, and the movement of data is now widely understood to be a major item in workflow energy budgets. For workflows at the other end of the continuum—within the data center—data movement between resources, data access contention (i.e., achieving sufficient access parallelism), the numerous complexities of resource scheduling, and the lack of interoperability in internal data formats suggest that a revolution, not just an evolution, will be needed in order to solve fundamental problems of data logistics in the BD era. The widening gap between I/O and computational rates also encouraged the view that in-situ analysis and visualization will have to be part of that revolution.

Intermediate processing: One common theme in the workflow descriptions at the BDEC workshop was the amount of "intermediate" (or pre) processing that data requires before the more substantial analysis and visualization processes occur. Such intermediate transformations are normally described in generic terms: cleaning, subsetting, filtering, mapping, feature extraction, registration, segmentation, etc. The question is whether or not some of these operations are generic enough so that a common set of software tools, appropriately layered and modularized, could be developed to serve the diverse purposes of a number of different communities at the same time. For example, image-driven workflows from fields such as medical imaging, microscopy, and remote (satellite) sensing, utilize all of the operations given above. Although the co-design effort that would probably be necessary to produce it would be challenging to organize, common software infrastructure that (suitably configured) could intermediate processing needs in a wide variety of fields would be a boon to data-driven research.

A common and visible data model: One major obstacle to creating shared software infrastructure for intermediate processing is the absence of interoperable data object models, or, just as importantly, a way of making the object model being used visible. The effort to develop a common model achieved limited success when OODB's were introduced in the 1990s, but that success was largely restricted to tightly coupled systems; they did not succeed for many more loosely coupled situations, which are typical today for many emerging BDEC domains and workflows. And Web based approaches (e.g., REST API, etc.) are likely to be viable for only a relatively small segment of these BD applications. Common object models have been established in some application domains (e.g., in multi-physics applications or climate modeling community), but creating a common software stack that supports interoperability far more generally has proved elusive. Moreover, for any such model to succeed, it will need to be flexible enough to provide data layout distribution options to support the kind of parallelism that applications and the I/0 services that support them will require. Due to the deluge data (e.g. from simulations), In situ (on line) processing is necessary," surrogate" model of the data, based (?) on the estimation of the quantity of information contained in the data.

3.2 Architecture Challenges

As noted above, the architectural perspective is especially appropriate for the BDEC, not only because of its focus on software infrastructure, but also because of the effects of the ongoing revolution in system design and the rapid increase in highly data-intensive science; taken separately and together, these changes are rendering old methods and infrastructures obsolete. Accordingly, a significant subgroup of BDEC participants concentrated on two key questions: 1) How can/should the internal architecture of HPC systems be changed to make them more suitable for data driven applications? And 2) How can/should external storage systems and their interfaces be adapted in order to efficiently orchestrate, as part of the overall workflow, the movement of data into and out of these systems? At this point in time, however, these questions seem to only generate more questions rather than any widely accepted (or even plausible) answers. Below we present a sample of these issues.

Adding energy targets complicates everything: All architectural issues are complicated by the fact that energy efficiency has now become a first class design constraint. Power aware data analytics, for example, is already starting to use approximation (e.g., reduced bit fixed-point representations) in order to achieve vastly improved energy efficiency and performance without introducing significant reduction in accuracy. Such trends suggest that the basic axes that define the design space for HPC systems are very likely to be realigned.

A far more adaptable storage paradigm may be required for HPC: It seems clear that computer architectures will need to incorporate new and more flexible storage paradigms, both in terms of hardware and in terms of software. For instance, large simulations, which have traditionally provided critical design targets for HPC systems, are increasingly facing a range of data-driven challenges, including writing output at high enough rates, moving data to secondary storage fast enough, and visualization-enabled steering in real-time. Such cases raise the question of whether and how the traditional memory/storage hierarchy needs to be restructured: in terms of its shape (pyramid, bell, etc.); in terms of its modes of access (standard file system, streaming, memory-mapped, object database, etc.); and in terms of its depth and composition (e.g., huge RAM or SSD caches or scratch resources inside the system, and more remote cloud resources outside it). Similarly, there is broad consensus that traditional file systems were not built for the kind of high performance I/0 that many data driven applications require. For example, data driven applications that take thousands, if not millions of small, independent files or records as input are a constant source of problems. One thing such applications reveal is the way in which metadata design can have a major impact on peformance and reliability. Applications that require that all data be accessible at nearly the same performance from all nodes are going to present a very difficult use case for extreme scale systems. Unless the storage infrastructure can adapt to meet the profile of the way the application wants to use the data, different storage systems would be needed so that different applications would be able scale in different ways.

Streaming not currently viable as a routine option: Today, supercomputers are not designed to allow for real-time streaming of sensor data from outside, but some future graph applications may require this capability. Researchers in these application communities want to get to a workflow that would support something like large-scale network analysis containing real-time streaming data processing, Hadoop-typed batched jobs, and large-scale graph analysis. Efficient data streaming will also be required for visualizations in this area, since researchers want to carry out these visualizations interactively and in situ as data is produced and as computations proceed. Such problems are closely related to efforts to develop intelligent I/O with in-transit processing (e.g., ADIOS) for functions such as data reduction.

The absence of mini-apps to support system designers: The development of mini-applications is a central aspect of the co-design process. These "mini-apps" are critical because they serve as tools for exploring architectural requirements of different applications, while demanding only a small fraction of the resource commitment to develop and run. Various types of mini-applications (e.g., skeleton apps, Compact apps, and Scalable Synthetic Compact Applications (SSCA), HPC Challenge Benchmarks, etc.), each with a different set of properties, have been defined to support the co-design effort. The more sustainable these miniapps are, the more valuable they become as instruments for charting the evolution of systems over time [14]. Several BDEC speakers noted current scarcity of miniapps suitable to explore HPC architecture alternatives for data-driven research; but they also described opportunities to create such miniapps that were being planned or were already in development for their application area.

For example, there is a tremendous amount of commonality between applications that compose and analyze information that are converging from multiple sensor systems or scientific simulations as another. Examples include numerous fields of biology and medicine, in which omic and imaging data need to be integrated and used in tandem, and applications in earth sciences that combine (coupled simulations) real-time sensor data. It seems plausible to try to define and publish an abstract application class that distills out essential characteristics of applications of this general type, and then promote the creation of one or more mini-applications to serve the purposes of the designers of future systems.

There are already a few prototype tools that offer immediate opportunities on this front. The Numerical Cosmology community, for instance, has developed several tools to evaluate the performance of I/O and of post-processing workflow for their massive simulations. The broader BDEC community might build on and generalize these tools to create a data-driven mini-application to test different approaches to I/O with respect to various parameters.

Other types that might be suitable targets for mini-applications include the following:

- *High velocity* – real time filter/compression, e.g., from lossy compression from large scale instruments (SKA, LSST) or ubiquitous low-quality sensors
- *Large numbers of nearly independent records* – web, financial transactions, twitter feeds. MapReduce and slightly better; cloud platforms, Databases; large scale instrument data (images)
- *Large single records* - May include highly and unpredictably correlated data, simulation results, large-scale graphs, etc.

3.3 Higher Level Data Challenges

Since data, and the observations derived from them, are the foundation of scientific knowledge, there is an obvious sense in which higher level data tools and services lie closer than other BDEC topic areas to issues of scientific validity, on the one hand, and to new methods of inquiry, on the other. Methods for dealing with "data provenance" are an obvious example of the former that was discussed at the meeting. Tools for recording data provenance are designed to enable researchers at some future time to determine whether error or bias or corruption may have been introduced into a particular finding, for example, by the way in which the data was handled and analyzed, or by collateral factors that affected the way it was generated (e.g., in large-scale simulations). Examples of the latter include new methods for combining data from multiple sources (e.g., genomics and medical imaging of the same bodily region, or group of cells) in order to produce richer insights about relationships between changes occurring in multiple dimensions over the same span of time. The BDEC meeting looked at cases and generated ideas and recommendations on both of these fronts.

It is important to note that these discussions are taking place in the context of a growing awareness within the HPC community of the need to address the entire research data lifecycle. Some efforts in this direction are already well underway. Projects like Globus Online, building on the broad acceptance of GridFTP for data transport, have moved steadily to improve their services and offer an integrated solution to researchers. For example, GO's cloud-based "dataset service," which is currently under development, will allow users "… to create, manage, access, search, and share metadata about sets of data elements (files, directories, database rows, …)—with utilities allowing for automated metadata extraction from various data sources." In Ophidia [15], the big data analytics platform, framework operators handle data provenance information jointly with digital object identifiers as part of the analytics processing. It is associated to datacube objects like other system and user metadata information, thus enabling stronger search & discovery capabilities at the filesystem level. The integrated Rule Oriented Data System (iRODS) supports policy-based data grid efforts that are attacking the same complex of problems.

Against that background, below is a sample of some of the issues and ideas discussed at the workshop:

Data provenance is a major issue in the BD era: The set of issues that now fall under the concept of "data provenance" are historically connected to the questions of peer validation and reproducibility, which have long been the hallmark of good science. To enable others to evaluate a given set of results, let alone reproduce them, the original researcher needs to produce a record (expressed as metadata) of how the results were produced so that others can follow. But producing such provenance for BD workflows, given their size and their multi-stage, multi-layered nature, is likely to prove a remarkably difficult challenge. From the point of view of software architecture, the problems involved are made all the more difficult because the goal of enabling future generations of researchers to understand and evaluate results from much earlier work means that the protocols and interfaces that are used for data provenance will have to maintain a high level of backward compatibility over time.

Some domains are already pioneering standards for provenance. For instance, in astronomy, the International Virtual Observatory Association (IVOA) is developing standards for provenance using Unified Content Descriptors (UCD) vocabulary, which will need to be implemented via machine readable semantic tagging in order to enable researchers to work efficiently with these massive data objects; in the areas of climate and earth modeling, research groups are working with the Persistent Identification of Data Sets (PIDS) model developed by the European Data Infrastructure (EUDAT) consortium; and software packages that are widely used in various domains, such as HDF5 and pnetCDF, have support for many aspects of data provenance built in. Still, for the reasons expressed above, the goal of providing an adequate and interoperable set of services and tools for BD metadata in general, and BD provenance in particular, seems likely to remain distant for some time to come.

Policy based data management needed: If infrastructure providers ought to strive to provide an environment in which the tools serve the researchers, rather than the researchers serving the tools, then the rise of BD science and the rapid proliferation of data management tools of various kinds, creates something of an ease-of-use crisis for tool builders. In order to ensure that the data management environment facilitates research, rather than impeding it, data management services need to be based on policies that encapsulate several types of knowledge: protocols that encapsulate knowledge needed to interact with available community resources; interfaces and (sharable) templates that encapsulate knowledge of the processing steps in research analysis workflows, including the storage of workflow results and provenance; and computer actionable rules that encapsulate management policies for research results. As noted above, Globus Online and iRODS-based data services are already progressing in this direction, but the challenge of providing a well adapted and sustainable solution remains an unrealized goal.

Environments that support new types of data-driven research: Although researchers have been successful for some time in processing computer-generated, semi-structured data (big simulations) and structured observational data (big instruments), many fields are now eagerly attacking the problems involving high volumes of unstructured and heterogeneous observational data (text, images, medical records, etc.), which often come in massive piles of smaller units, are asynchronously generated, and demand that close coordination between data assimilation and analysis. The BD challenges that researchers in bioinformatics confront provides one of many illustrations of this point. Data-centric computing environments for such domains need to incorporate new types of methods and algorithms, to supply infrastructures that enable compute-driven methods (modeling and simulation) to converge with experimental ultra-scale data analytics and visualization, and to enable sharing of methods, protocols, experiments and tools through social networks that enable the best methods to emerge as rapidly as possible. In this regard, collaborative environments connected with social networks could drastically change the way scientists interact with each other both inside (for research purposes) and outside (for dissemination purposes) the scientific community.

Shared software infrastructure for intermediate processing: The translation of all scientific data into digital form has opened up a major opportunity space for research methods that integrate or synthesize data of multiple types and/or from multiple sources or sensor modalities. This is particularly true for application areas, now common, that utilize and combine multi-dimensional spatial-temporal datasets. Examples include Radio- and Microscopy Imaging combined with omic data; simulation data (e.g., for oil fields, carbon sequestration, groundwater pollution/remediation) combined with seismic and earth sensor data; and weather prediction based on the real time integration of data from simulations, satellites, ground sensors, and live video feeds. The Google self-driving car provides a more practical, consumer illustration of the use of real-time integrated analysis of correlative data from multiple sensor modalities and sources. The multi-dimensional data space that these applications define tends to be high resolution in each of their correlative dimensions, so that, when even a modest number of data steps is involved, extremely large volumes of data need to be accessed and processed in a coordinated way.

3.4 Software Challenges

Software infrastructure is the primary focus of the BDEC. Consequently, this category cross-cuts with all the other categories. The purpose of calling it out for separate attention is so that a “gap analysis,” where some critical piece of functionality or problematic conflict is identified, can be brought to the attention of the entire community. This activity is especially important in an international context. Although hardware (e.g., large supercomputers and data repositories for open science) tend to be a national investments, software infrastructure can be an international investment because it can be shared in a commons. Yet valuable pieces of software, developed and successfully being used by a given community or country, may be hiding beneath other components in a software stack and unavailable for general use. Another function of this BDEC focus area is to evaluate software interfaces and tools that, because they have achieved high levels of community momentum, need to be promoted for even broader adoption. The "gaps/improvements" list provided below includes all the cases discussed at the workshop that might fall into either.

A number of gaps in the BDEC software infrastructure were highlighted, including the following:

Tools to support real-time monitoring and observation of workflows: Familiar engineering wisdom holds that tools, services, and processes can be improved only so far as they can be monitored and measured. Although certain, very limited, segments of the BD workflow and data management life-cycle can be observed and measured in the necessary way, many tools are still needed to fill in the missing gaps, and a framework that puts monitoring and measurement elements all under a unified management regime is still missing.

Coordination between data movement and compute services: Less than adequate coordination between the data movement services and the compute services affects data logistics at all levels, from the wide area network to the inside of the machine room. This is at least partly, if not substantially, a function of the lack of common interfaces between HPC system software and workflow software, such as libraries/filters that applications could call as standard components to replace explicit I/O.

Automated support for provenance and other metadata: The difficult problems with dealing with workflow provenance, and BD metadata generally, were briefly described above. Some initial requirements for initial work in this area, however, seem relatively clear: provide server-side metadata management capabilities; develop primitives for describing data sets (and provenance) based on the needs of data analytics; provide improvements to search and discovery processes by supporting metadata for cross-linking and indexing correlated digital objects; and build new community-oriented tools to enrich metadata multi-level and collaborative forms of research.

Mechanisms to support fault tolerant workflows in data analysis: When data is lost in mid-flow, there is no way to recover, and processes must be restarted from the beginning. As flows get larger and larger, the lack of such a mechanism will become more and more painful and costly. Support for fault tolerance certainly exists, to some degree, within some stages of a multi-stage workflow (which might be called, "intrastage fault tolerance"), and significant work is being done to provide support between stages ("interstage fault tolerance"); but as the size of data flows continue to escalate, current approaches to fault tolerance will unquestionably be challenged to keep up.

Mini-apps to support infrastructure codesign: As noted above, mini-applications are a critical facilitator of the co-design process, supporting the exploration of alternative architectural choices while demanding a small fraction of the resources. The workshop identified a variety of potential targets for such mini-apps, including applications that synthesize multiple sensors and data sources, support I/O and post-processing workflow, perform high velocity compression or filtering on structured data, and process huge numbers of small, independent files or data objects.

Integration of widely used BD-capable data libraries into standard packages: There exists a simple lack of integration of well-established and robust scientific data formats, such as HDF5, pNetCDF, and GRIB, into many analysis, graphics, and analysis packages that are in common use.

The standard data libraries in familiar user packages, like R, Matlab, and Mahout, do not have the best algorithms or software, either in terms of functionality or in terms of parallelism. Moreover, support for domain specific scripting languages (DSLs) that facilitate the use of these libraries in different areas would be a natural complement to such integration.

Common tools for managing and exploring data: Many large and diverse communities, such as climate modeling, still depend on an assortment of custom-made tools (mostly in Fortran or C) to manage and explore their data. This dependence on a diversity of custom-made solutions severely constrains the ability of the community to compare modeling results across different groups. Since inter-model comparisons are becoming more and more essential, a standard data management and analytics tool kit would go a long way toward overcoming this problem. Numerous applications will also need efficient tools for exploring data *in situ*, either on line or inside the simulation itself.

3.5 Interoperability Challenges

Issues of interoperability are closely related with fundamental questions about the architecture and codesign of hardware and software infrastructure. Consequently, they are of central importance to the BDEC community. Unfortunately, these same factors tend to make them relatively intractable. For interoperability has to mean more than just "everyone adopts the same standard or the same interface." Aside from cases where de facto or de jure monopoly power is exercised, a viable approach to interoperability for infrastructure means designing protocols and interfaces that people *voluntarily adopt* because they can use them to achieve their functional goals while also achieving deployment scalability and sustainability over time. Among the best known examples are the design of Internet Protocol (IP) and the UNIX file system. In the case of IP, in order to be the "narrow waist of the hourglass" in the network protocol stack, IP was designed to be simple, generic, and limited: simple, so that it could be implemented over a variety of network substrates, including those that had yet to be invented; generic, so that it offered functionality that every application using the network required and could build on; and limited in terms of the guarantees it offered, so as to maximize the opportunity for resource sharing and minimize the costs of deployment.

Clearly, all these dimensions represent continua along which different tradeoffs can be made, and the BDEC community has a strong interest in finding opportunities to push toward software infrastructure for data-driven science and engineering that achieves interoperability along similar lines wherever possible. But the complexity of the obstacles to be overcome is apparent in the fact that, in order to maximize the opportunity space for solid data-driven research, interoperable tools and services will need to be achieved at nearly every level of the "data" stack—from storage and data access protocols at the bottom, through workflow tools and services in the middle, to data formats/object models, metadata and data provenance standards at the top.

Some notable efforts already being made on this front were presented at the meeting. The work on community standards for data provenance within the Astronomy community and the climate and earth modeling communities have already been noted above. And the members of the NSF-supported DataNet Federation Consortium (DFC), which aims to create a national data grid for sharing and preserving data collections in order to enable reproducible, data-driven research, have worked to converge on a common idea of where interoperable tools and services are essential. The agreed upon list includes programming languages for clients, policy mechanisms, security, metadata, messaging, and so on. For the most part, however, various extant (and less-than-fully-interoperable) technologies have currently been adopted to provide the necessary protocols and interfaces so that experimental prototypes can be constructed and work can begin. Achieving interoperability in many areas identified as important or essential by the DFC, and by the BDEC community generally, continues to be very much a work in progress.

4 Bibliography

[1] *International Exascale Software Project*, http://www.exascale.org/iesp/Main_Page. 2014.

[2] *European Exascale Software Initiative*, http://www.eesi-project.eu.

[3] R. Stevens, T. Zacharia, and H. Simon, "Modeling and Simulation at the Exascale for Energy and the Environment Town Hall Meetings Report," Department of Energy Office of Advance Scientific Computing Reserach, Washington, DC, pp. 174, 2008. http://www.sc.doe.gov/ascr/ProgramDocuments/Docs/TownHall.pdf.

[4] National Research Council Committee on the Potential Impact of High-End Computing on Illustrative Fields of Science and Engineering, "The Potential Impact of High-End Capability Computing on Four Illustrative Fields of Science and Engineering," Washington, DC, pp. 142, 2008.

[5] P. Kogge, et al., "ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems," DARPA IPTO, pp., 2008.

[6] V. Sarkar, et al., "ExaScale Software Study: Software Challenges in Extreme Scale Systems," DARPA Information Processing Techniques Office, Washington DC., pp. 159, September 14, 2009. http://users.ece.gatech.edu/~mrichard/ExascaleComputingStudyReports/ECSS report 101909.pdf.

[7] J. Dongarra, et al., "The International Exascale Software Project roadmap," *International Journal of High Performance Computing Applications*, vol. 25, no. 1, pp. 3-60, February 1, 2011, 2011.

[8] J. Y. Berthou, et al., "EESI D5.6 Final report on roadmap and recommendations development," European Exascale Software Initiative, pp. 51, 2013.

[9] P. Ricoux, "EESI2 First Intermediate Reports," European Exascale Software Initiative, pp. 30, 2013.

[10] T. Hey and A. Trefethen, "The Data Deluge: An e-Science Perspective," in *Grid Computing*: John Wiley & Sons, Ltd, 2003, pp. 809-824, http://dx.doi.org/10.1002/0470867167.ch36.

[11] D. Atkins, et al., "Revolutionizing Science and Engineering through Cyberinfrastructure: Report of the National Science Foundation Blue-Ribbon Panel on Cyberinfrastructure," Panel Report, pp. 1-84, January, 2003. http://www.communitytechnology.org/nsf_ci_report/.

[12] B. M. Frischmann, *Infrastructure: The Social Value of Shared Resources*: Oxford University Press, USA, 2012.

[13] D. Reed, "Data Economies and Cultural Incentives," in *Big Data and Extreme Scale Computing (BDEC) Workshop*. Charleston, SC, 2013.

[14] M. A. Heroux, et al., "Improving performance via mini-applications," SAND2009-5574; TRN: US201101%%121 United States10.2172/993908TRN: US201101%%121Mon Jan 24 07:54:01 EST 2011SNLEnglish, pp. Medium: ED; Size: 40 p., 2009. http://www.osti.gov/scitech//servlets/purl/993908-082LD1/.

[15] S. Fiore, et al., "Ophidia: Toward Big Data Analytics for eScience," *Procedia Computer Science*, vol. 18, no. 0, pp. 2376-2385, //, 2013.

5 Appendices

Appendix 1: Meeting Agenda (including speaker/summary slides):

- **Session 1 (Chair: Jack Dongarra, University of Tennessee)**
  - 9:00am to 9:10am - Welcome to Big Data and Extreme Computing: Goals and Overview
    - Pete Beckman, Argonne National Laboratory
    - Jean-Yves Berthou, French National Research Agency (ANR)
    - Jack Dongarra, University of Tennessee
    - Yutaka Ishikawa, University of Tokyo
    - Satoshi Matsuoka, Tokyo Institute of Technology
    - Philippe Ricoux, Total SA
    - 1_beckman-bdec.pdf
  - 9:10am to 9:30am – Big Data's Biggest Needs – Deep Analytics for Actionable Insights
    - Alok Choudhary, Northwestern University
    - 2_alok-BDEC-charleston-2013.pdf
  - 9:30am to 9:50am – Big Data and Big Crunch for the Square Kilometre Array
    - Tim Cornwell, Square Kilometre Array
    - 3_cornwell_ska_bdec2013.pdf
  - 9:50am to 10:10am – Turning Large simulations into Numerical Laboratories
    - Alex Szalay, Johns Hopkins University
    - 4_szalay-bdec-2013.pdf

- **Session 2 (Chair: Jean-Yves Berthou, French National Research Agency (ANR))**
  - 10:40am to 11:00am – Can we converge big data, big compute and big Interaction in future hardware and software?
    - Rick Stevens, Argonne National Laboratory
    - 5_BDEC-RICK.pdf
  - 11:00am to 11:20am – Big Data Parallel Processing of Personal Genomes
    - Shinichi Morishita, University of Tokyo
    - 6_BDEC_Shinichi Morishita_U_Tokyo.pdf
  - 11:20am to 11:40am – Beyond Embarrassingly Parallel Big Data
    - William Gropp, University of Illinois
    - 7_gropp_BeyondIndepData.pdf
  - 11:40am to 12:00pm – Weather and Climate Modelling: ready for exascale?
    - Pier Luigi Vidale, University of Reading
    - 8_BDEC_PLV-Apr-2013.pdf

- **Session 3 (Chair: Pete Beckman, Argonne National Laboratory)**
  - 1:00pm to 2:30pm – Panel 1
    - Jean-Michel Alimi, Laboratoire Univers et Théories
    - Reagan Moore, University of North Carolina at Chapel Hill
    - Wolfgang E. Nagel, TU Dresden, ZIH
    - Osamu Tatebe, University of Tsukuba
    - Pier Luigi Vidale, University of Reading
    - Toyotaro Suzumura, Tokyo Institute of Technology

        - BDEC_Panel_1_Complete.pdf

- **Session 4**
    - 3:00pm to 5:00pm – Breakout Groups 1
        - Track 1 – International Collaboration, Frameworks, Funding, and Co-design: Bronis de Supinski, Jean-Yves Berthou
        - Track 2 – Architecture: Rick Stevens, Satoshi Matsuoka
        - Track 3 – Software: Vivek Sarkar, Bill Kramer, Wolfgang Nagel
    - 5:00pm to 5:30pm – Breakout Report 2
        - 15_BDEC_Breakouts_Day1_Complete.pdf

Wednesday, May 1

- **Session 5 (Chair: Philippe Ricoux, Total SA)**
    - 9:00am to 9:20am – Integrative Biomedical Informatics, Big Data and Extreme Scale Computing
        - Joel Saltz, Emory University
        - 16_BDEC_Saltz_May1_2013.pdf
    - 9:20am to 9:40am – On the Role of Indexing for Big Data in Scientific Domains
        - Arie Shoshani, Lawrence Berkeley National Laboratory
        - 17_BDEC_Shoshani_indexing.pdf
    - 9:40am to 10:00am – Big Data and extreme-scale computing challenges in solid Earth Sciences
        - Jean-Pierre Vilotte, Institut de Physique du Globe de Paris
        - 18_BDEC-Vilotte-Final.pdf

- **Session 6 (Chair: Yutaka Ishikawa, University of Tokyo)**
    - 10:30am to 10:50am – Co-existence: Can Big Data and Big Computation Co-exist on the Same Systems?
        - William Kramer, University of Illinois
        - 19_Kramer-BDEC - v1.pdf

    - 10:50am to 11:10am – Synergistic Challenges in Data-Intensive Science and Extreme Computing
        - Vivek Sarkar, Rice University
        - 20_Sarkar-BDEC-presentation-April-2013-v2.pptx_.pdf

    - 11:10am to 11:30am – Computational Challenges in Big Data Assimilation with Extreme-scale Simulations
        - Takemasa Miyoshi, RIKEN Advanced Institute for Computational Science
        - 21_Miyoshi_20130501.pdf

- **Session 7 (Chair: Satoshi Matsuoka, Tokyo Institute of Technology)**
    - 12:30pm to 2:00pm – Panel 2
        - Sandro Fiore, University of Salento & CMCC, Italy

        - Kenji Ono, RIKEN Advanced Institute for Computational Science
        - Geoffrey Fox, Indiana University
        - Andrew Lumsdaine, Indiana University
        - Shinichi Morishita, University of Tokyo
        - Ranga Vatsavai, Oak Ridge National Laboratory
        - 27_BDEC_Panel_2_Complete.pdf

- Session 8
    - 2:30pm to 4:30pm – Breakout Groups 2
        - Track 1 – Interoperability: Bronis de Supinski, Jean-Yves Berthou
        - Track 2 - Workflows: Rick Stevens, Satoshi Matsuoka
        - Track 3 - Taxonomy: Vivek Sarkar, Bill Kramer, Wolfgang Nagel
    - 4:30pm to 5:00pm – Breakout Report 2
        - 30_BDEC_Breakouts_Day2_Complete.pdf

Appendix 2: Meeting White Papers

1. Ranga Vatsavai, Budhendra Bhaduri, “Geospatial Analytics for Big Spatiotemporal Data: Algorithms, Applications, and Challenges,” BDEC Workshop, Charleston, SC, White Paper, April 2013.
   raju-bigspatial.pdf
2. Toyotaro Suzumura, “Big Data Processing in Large-Scale Network Analysis and Billion-Scale Social Simulation,” BDEC Workshop, Charleston, SC, White Paper, April 2013.
   BDEC-20130427-suzumura.pdf
3. Ian Foster, “Extreme-Scale Data Lifecycle Management as a Service,” BDEC Workshop, Charleston, SC, White Paper, April 2013.
   Exascale Data Management as a Service_Foster.pdf
4. Jean-Michel Alimi, “The Challenges of the Next Decade in Numerical Cosmology: Big Data and Extreme-Scale Computing,” BDEC Workshop, Charleston, SC, White Paper, April 2013.
   BDEC_JM_ALIMI.pdf
5. Pier Luigi Vidale, Hilary Weller, B. N. Lawrence, “Weather and Climate Modeling: ready for exascale?,” BDEC Workshop, Charleston, SC, White Paper, April 2013.
   BDEC workshop-PL_Vidale.pdf
6. Osamu Tatebe, “File system and runtime system for big data,” BDEC Workshop, Charleston, SC, White Paper, April 2013.
   position-paper-tatebe.pdf
7. Dan Reed, “Data Economies and Cultural Incentives,” BDEC Workshop, Charleston, SC, White Paper, April 2013.
   Data Economies_Reed.pdf
8. Giovanni Aloisio, Sandro Fiore, Ian Foster, Dean Williams, “Scientific big data analytics challenges at large scale,” BDEC Workshop, Charleston, SC, White Paper, April 2013.
   BDEC_Big data analytics for eScience_Ophidia_Aloisio.pdf
9. Wolfgang E. Nagel, Ralph Müller-Pfefferkorn, Michael Kluge, Daniel Hackenberg, “Execution Environments for Big Data: Challenges for Storage Architectures and Software,” BDEC Workshop, Charleston, SC, White Paper, April 2013.
   BDEC-Position-Paper-Nagel.pdf
10. Kenji Ono, “Life cycle management of big data for extreme-scale simulation,” BDEC Workshop, Charleston, SC, White Paper, April 2013.
    Ono-Life-cylce_management.pdf
11. Geoffrey Fox, “Distributed Data and Software Defined Systems,” BDEC Workshop, Charleston, SC, White Paper, April 2013.
    SDNDistributedData_Fox.pdf
12. Andrew Lumsdaine, “New Execution Models are Required for Big Data at Exascale,” BDEC Workshop, Charleston, SC, White Paper, April 2013.
    BDEC_Lumsdain.pdf

13. Shinichi Morishita, “Big Data Parallel Processing of Personal Genomes,” BDEC Workshop, Charleston, SC, White Paper, April 2013.
Morishita-Big Data and Extreme Computing.pdf

14. Reagan Moore, “Integration of Scientific Analyses and Storage,” BDEC Workshop, Charleston, SC, White Paper, April 2013.
Policy-based-data-storage_Moore.pdf

Appendix 3: BDEC participants

- Yutaka Akiyama (Tokyo Institute of Technology)
- Jean-Michel Alimi (Laboratoire Univers et Théories)
- Giovanni Aloisio (University of Salento & CMCC, Italy)
- Pete Beckman (Argonne National Laboratory/U. of Chicago)
- Jean-Yves Berthou (French National Research Agency [ANR])
- Alok Choudhary (Northwestern University)
- Tim Cornwell (Square Kilometre Array)
- Bronis de Supinski (Lawrence Livermore National Laboratory)
- Jack Dongarra (University of Tennessee)
- Teresa Finchum (University of Tennessee)
- Sandro Fiore (University of Salento & CMCC, Italy)
- Geoffrey Fox (Indiana University)
- Kelly Gaither (Texas Advanced Computing Center)
- Al Geist (Oak Ridge National Laboratory)
- William Gropp (University of Illinois)
- Michael Heroux (Sandia National Laboratories)
- Yutaka Ishikawa (University of Tokyo)
- Daniel Katz (National Science Foundation)
- William Kramer (University of Illinois)
- Andrew Lumsdaine (Indiana University)
- Satoshi Matsuoka (Tokyo Institute of Technology)
- William Michener (University of New Mexico)
- Takemasa Miyoshi (RIKEN Advanced Institute for Computational Science)
- Bernd Mohr (Jülich Supercomputing Centre)
- Reagan Moore (University of North Carolina at Chapel Hill)
- Terry Moore (University of Tennessee)
- Shinichi Morishita (University of Tokyo)
- Wolfgang E. Nagel (TU Dresden, ZIH)
- Lori O'Connor (Argonne National Laboratory)
- Kenji Ono (RIKEN Advanced Institute for Computational Science)
- Tracy Rafferty (University of Tennessee)
- Philippe Ricoux (Total SA)
- Joel Saltz (Emory University)
- Vivek Sarkar (Rice University)
- Arie Shoshani (Lawrence Berkeley National Laboratory)
- Rick Stevens (Argonne National Laboratory)
- Toyotaro Suzumura (Tokyo Institute of Technology)
- Martin Swany (Indiana University)
- Alex Szalay (Johns Hopkins University)
- William Tang (Princeton University)
- Osamu Tatebe (University of Tsukuba)
- Ranga Vatsavai (Oak Ridge National Laboratory)
- Pier Luigi Vidale (University of Reading)
- Jean-Pierre Vilotte (Institut de Physique du Globe de Paris)
- Voevodin Vladimir (Research Computing Center - Moscow State University)
- Ramin Yahyapour (GWDG)

Appendix 4 — Suggested questions for speakers to address

1. Architecture:

- What architectural changes are needed for extreme computing storage systems to make them better suited for BD?
- What operational changes are needed to support new storage architectures?
- Looking at future technologies, what future architectures are possible?
- It is quite important for us to run the system for many hours without failure of computations. Another pressing need is to accelerate data transfer between the main memory and secondary disk.

2. Workflows:

- For extreme computing and big data, describe a forwarding-looking workflow, from simulation to analysis.
- What software is missing to support your workflow?
- A plan for achieving interoperability among various systems that one might want to use.

3. Taxonomy:

- There are several forms of data-centric computing linked to extreme computing. One outcome of this workshop is to help describe these modes. Please outline how you use your data and how you answer questions about your science using your data.
- Do you have a data-driven mini-application that demonstrates a new usage model?
- What are cross-cutting concerns for BD (for example: data integrity) See the two problems mentioned above.
- Improvement to address your big data needs? As you look to the future, what are the holes/gaps that have no planned solution?

4. Software:

- What software are you currently using to manage and explore your data?
- What algorithms and software libraries/tools need development and
- Our processing pipeline outputs huge temporary files to secondary disks several times, which should be avoided in some way.

5. Interoperability challenges:

- How to handle Data provenance (location, observed/simulated, type of system concerned) from a data representation and IT architectural point of view? How to annotate existing data sets and develop records for data citation and tracking?
- What Information systems are used for providing semantic capacity to provide effective translation between data and conceptual models used by different communities?
- What IT systems are used for providing information about the actual use of both observational data and simulated data?