

# THE CHANGING PARADIGM OF DATA-INTENSIVE COMPUTING

Richard T. Kouzes, Gordon A. Anderson, Stephen T. Elbert, Ian Gorton, and Deborah K. Gracio, *Pacific Northwest National Laboratory*

Through the development of new classes of software, algorithms, and hardware, data-intensive applications provide timely and meaningful analytical results in response to exponentially growing data complexity and associated analysis requirements.

**T**he continued exponential growth of computational power, data-generation sources, and communication technologies is giving rise to a new era in information processing: data-intensive computing.

According to a 2004 study on data management for science by the US Department of Energy (DOE), “We are entering an information-dominated age. Ability to tame a tidal wave of information will distinguish the most successful scientific, commercial, and national-security endeavors.”<sup>1</sup> Another study on systems biology for energy and the environment, when discussing computational models, noted that “these enormously complex and het-

erogeneous full-scale simulations will require not only petaflop capabilities but also a computational infrastructure that permits model integration. Simultaneously, it must couple to huge databases created by an ever-increasing number of high-throughput instruments.”<sup>2</sup>

More recently, a DOE-sponsored report on visual analysis and data exploration at extreme scale<sup>3</sup> found that “datasets being produced by experiments and simulations are rapidly outstripping our ability to explore and understand them, and there is, nationwide, comparatively little basic research in scientific data analysis and visualization for knowledge discovery.”

## DEFINING THE DISCIPLINE

All of the existing definitions of data-intensive computing tend to focus on handling the problems of massive datasets.<sup>4</sup> However, because new application requirements are driving data-intensive computing, we believe a broader definition is needed.

Data-intensive computing promises not just an evolutionary change in informatics but also a revolutionary change in the way researchers gather and process information, from the hardware and algorithms to the


presentation of knowledge to the end user. Applications in many disciplines are driving a shift in the emphasis of data-intensive computing from focusing solely on large datasets to the broader realm of issues dealing with the time to reach a solution when data-handling capacity is a significant factor, such as required real-time processing of massive data streams.

Some examples illustrate these new requirements:

- The North American electric power grid operations generate 15 terabytes of raw data per year, and estimates for analytic results from control, market, maintenance, and business operations exceed 45 Tbytes/day. As developers add new high-resolution sensors to the grid, this data volume is increasing rapidly while the time available to make control decisions remains constant.
- A new generation of climate models that explicitly resolves, rather than parameterizes, cumulus clouds will produce a 1,000-fold increase in data, from 8 Tbytes for a single 100-year simulation at current coarse resolution (100 km) to 8 petabytes per run for the same simulation at the planned 3-km resolution.
- The latest genomics and proteomics programs in biology produce massive datasets from experiment and theory that require new approaches to discovery. Production of proteomics data by just our laboratory can exceed 10 Tbytes/day; the national rate of proteomics data production exceeds 100 Tbytes/day from such activities as microbial processes related to the creation of new biofuels.
- Social networking sites such as Facebook ([www.facebook.com](http://www.facebook.com)) capture and store petabytes of heterogeneous information and maintain complex networks that link users. Mining this data to create new, high-value applications for users is an immensely challenging problem. For example, Google sorts through 20 Pbytes per day.<sup>5</sup>
- The intelligence community is challenged with extracting useful knowledge from large amounts of communications data that is many orders of magnitude beyond its current ability to analyze. Analysts need to repeatedly filter through many terabytes of data to extract the information relevant to national security issues.
- High-energy physics remains a leading generator of raw data. For example, the Atlas (<http://atlasexperiment.org>) experiment for the Large Hadron Collider (LHC) at the Center for European Nuclear Research (CERN) will generate raw data at a rate of 2 Pbytes per second beginning in 2008 and store about 10 Pbytes per year of processed data.

A petabyte represents the content of 5,000 Blu-ray disks (each holding 200 Gbytes), 0.6 km of them laid end-to-end. But that is just the tip of the iceberg. IDC estimates that in 2007, the digital universe consisted of 281 exabytes (281,000 petabytes—about halfway to the moon with Blu-ray disks).<sup>6</sup> IDC also estimates that the amount of information will grow by a factor of 10 in just five years (more than twice to the moon and back).

These challenging examples represent a new era that will require a shift for information technology to incorporate a far more expansive, flexible, and responsive enterprise model and operating philosophy.

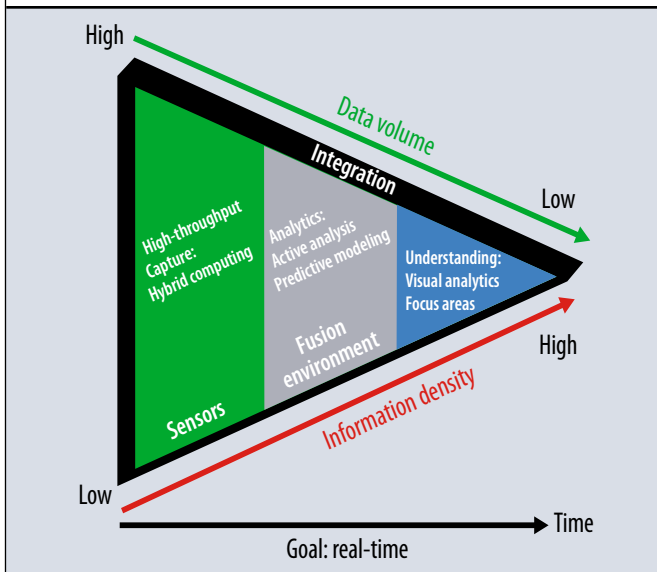


**A new era will require a shift for information technology to incorporate a far more expansive, flexible, and responsive enterprise model and operating philosophy.**

In contrast to compute-intensive tasks where available processing power is the rate-limiting factor, data-intensive computing could be qualitatively defined as “any computational task where data availability is the rate-limiting factor to producing time-critical solutions.” The term “availability” used here includes such factors as latency and bandwidth in hardware systems that impact the capacity to obtain, process, and dispose of data at rates that match the sources’ capability to provide data. However, this definition misses the mark because it would include processes where data has not yet been produced, which is not necessarily a data-intensive problem. We suggest an updated qualitative definition: Data-intensive computing is managing, analyzing, and understanding data at volumes and rates that push the frontiers of current technologies.<sup>7</sup>

Data-intensive computing facilitates human understanding of complex problems. Data-intensive applications provide timely and meaningful analytical results in response to exponentially growing data complexity and associated analysis requirements through the development of new classes of software, algorithms, and hardware. In many application domains, data volumes that must be processed have grown to the petabyte scale while, simultaneously, computational models and algorithms have pushed the performance of existing computer architectures.<sup>8</sup>

Revolutions in scientific experimentation, data sensor diversity, and computing power, and the availability of inexpensive, distributed communications have driven this explosion in the volume and complexity of data. Effective



**Figure 1. Data processing pipeline blueprint. The processing steps in the pipeline reduce large data volumes to create small datasets suitable for visualization or human understanding.**

coupling of computational simulations with experimental and field data through innovations in knowledge management, information analytics, visualization, and decision tools is critical for progress in science, homeland security, and the national energy and economic infrastructure.

To be more specific about what data-intensive computing encompasses, it is valuable to quantify the meaning of this term beyond a qualitative definition. Quantifying the meaning of data-intensive computing is heavily impacted by the complexity of factors that need to be considered. Classifying a problem as data-intensive could clearly depend upon the data rates (gigabytes/s to terabytes/s) and data volumes (terabytes to petabytes) involved, but other factors such as the variability in data rate, bandwidth of data paths, number of data handling units, complexity of the data and analysis, and human limitations in interacting with the data can all be important.

### DATA-INTENSIVE COMPUTING STYLES

Several model solutions for contemporary data-intensive problems have emerged in the past few years.

#### Data-processing pipelines

Researchers use processing pipelines to address many large data problems emerging from the scientific domains. Initially, the researchers capture and store raw data that originates from a scientific instrument or a simulation. The first stage of processing typically applies techniques to reduce the data's size by removing noise or by indexing, summarizing, or marking it up so that downstream analytics can manipulate it more efficiently.

Once the capture and initial processing have taken place, complex algorithms search and process the data. These algorithms create information or knowledge that humans or further computational processes can digest. Often, these analytics require large-scale distributed or specialized high-performance computing platforms to execute.

Finally, the researchers present the analysis results to users so that they can digest and act upon them. This stage might use advanced visualization tools, enabling the user to step back through the processing that has been performed to conduct forensic investigations to validate the outcome. Users might also need facilities to modify parameters on some of the analytics that have been performed and reexecute various steps in the processing pipeline.

As Figure 1 shows, processing pipelines start with large data volumes with low information content. The subsequent processing steps in the pipeline reduce this data to create relatively small datasets rich in information and suitable for visualization or human understanding. In many applications, for example, the Atlas (<http://atlas.web.cern.ch/Atlas/index.html>) high-energy physics experiment, large datasets are moved between sites over high-speed wide-area networks for downstream pipeline processing.

#### Data warehouses

Commercial enterprises are voracious users of data warehousing technologies. Mainstream vendors supply these database technologies to provide archival storage of business transactions for business analysis purposes. As enterprises capture and store more data, data warehouses have grown into the petabyte range. The best-known, Wal-Mart's, has grown over a decade to store more than a petabyte (*Information Week*, 6 Aug. 2007), fueled by daily data from 800 million transactions generated by its 30 million customers.

The data warehousing approach is now finding traction in science. The Sloan Digital Sky Survey (SDSS) SkyServer stores the results of processing raw astronomical data from the SDSS telescope in a data warehouse for subsequent data mining by astronomers (<http://cas.sdss.org/dr6/en>). While the SkyServer data warehouse currently only stores around 10 Tbytes of data, its fundamental design principles are being leveraged in the design of the data warehouse for the Large Synoptic Survey Telescope ([www.lsst.org](http://www.lsst.org)) that will commence data production in 2012. The telescope will produce 6 Pbytes of raw data each year, requiring the data warehouse to grow at an expected rate of 300 Tbytes per year.

#### Data centers

Driven by the Internet's explosive growth, Internet search enterprises such as Google and Yahoo have de-



veloped multipetabyte data centers based on low-cost commodity hardware. Data is stored across a number of widely geographically distributed physical data centers, each of which might contain more than 100,000 nodes. Programming models such as MapReduce<sup>5</sup> and its open source counterpart, Hadoop (<http://hadoop.apache.org>), provide abstractions that simplify writing applications that access this massively distributed data collection.

Essentially, MapReduce distributes data and processing across clusters of commodity computers and processes the data in parallel locally at each node. In this way, massively parallel processing can be achieved using clusters that comprise thousands of nodes. In addition, the supporting runtime environment provides transparent fault tolerance by automatically duplicating data across nodes and detecting and restarting computations that fail on a particular node.

This approach is also attracting interest from the scientific community. The National Science Foundation is partnering with Google and IBM to provide a 1,600-node cluster for academic research ([www.nsf.gov/news/news\\_summ.jsp?cntn\\_id=111186](http://www.nsf.gov/news/news_summ.jsp?cntn_id=111186)). Supported by the Hadoop open source software, this provides an experimental platform for scientists and researchers to use in investigating new data-intensive computing applications.

## CHALLENGES FOR TECHNOLOGIES

While some problems combine both data-intensive and compute-intensive challenges, others clearly fit into one class or the other.

Data-intensive problems occur when the size or complexity of the information source influences the way researchers address solutions or seek answers. Data-intensive computing begins with the analysis and interpretation of massive amounts of data. The data might be needed to build and constrain the space of feasible models that make simulations computationally tractable, but more often the primary focus of the analyses is to derive insights through computationally driven discovery or hypothesis testing.

A variety of tools for the management of data-intensive problems exist or are emerging. However, major gaps in capabilities remain and extant capabilities are not well integrated or always easily adaptable across domains. The problems are exacerbated by the many data-intensive problems that require the computing power available from high-performance computing systems or massive distributed clusters of commodity machines.

### Data management

The first challenge to data-intensive computation is the incoming data, obtained from multiple sources, types, scales, and locations with varying degrees of quality and reliability. Information about the data (metadata) might be either highly relevant and high quality (automated results

from sensors or experiments) or sparse and unreliable. The metadata might be embedded in large amounts of extraneous information, such as news feeds, where evaluation by humans might be desirable or even necessary for proper assertion of semantic or ontological relationships.

Other requirements might add to the complexity of the data ingest process. Must the data be merged into a single stream, as audio and video data are merged in a streaming media presentation, before being analyzed? How should the data be augmented with additional metadata to indicate references and relationships to other data? What data structure contains the most information yet is the most



**The relational database technology that underpins data warehouses is scaling to support petabyte data collections.**


efficient to analyze? How does this data structure indicate where data is missing or is of poor quality? Is there an optimal sampling rate for data at different resolutions? Is the dataflow steady or does it arrive in bursts? Can the bursts exceed processing capacity? If so, how is this handled? How long must the data be preserved? Are there acceptable lossy or nonlossy compression algorithms? The answers to these fundamental questions are needed to define a data-management architecture and establish performance requirements for a data-intensive application.

Various approaches are available for physically storing and accessing massive data volumes. High-performance, clustered file systems such as Lustre ([www.lustre.org](http://www.lustre.org)) and the General Parallel File System ([www-03.ibm.com/systems/clusters/software/gpfs/index.html](http://www-03.ibm.com/systems/clusters/software/gpfs/index.html)) are commonly deployed in high-performance computing centers. These file systems are capable of scaling across thousands of disks to support petabytes of storage within a single file system. They also offer high reliability and availability capabilities by providing redundant paths and automatic recovery from node and disk failures. From an application perspective, they present a traditional Posix-style file system interface for data-intensive applications to exploit.

Underlying the MapReduce programming model is the Google File System, along with its open source counterpart, the Hadoop Distributed File System. While these systems have much in common with traditional distributed file systems, they differ in that they are built based on the assumption that terabyte datasets will be distributed across thousands of disks attached to commodity compute nodes. In such environments, hardware failure occurs regularly.

Hence, data redundancy, fault detection, and computation recovery are core facilities that the file system provides transparently to applications. In addition, files are assumed to be read-mostly, and they can only be created and subsequently appended to. This simplifies data coherency issues and provides high-throughput data access.

In addition, the relational database technology that underpins data warehouses is scaling to support petabyte data collections. Parallel, high-performance relational database engines are capable of automatically parallelizing Structured Query Language queries and efficiently planning query execution on relational data that is distributed in partitioned tables across multiple disks. The declarative nature of SQL queries and the built-in query execution and data management capabilities of relational databases can provide a highly simplified model for data-intensive applications, as long as the database can organize the underlying data into a relational form that can be queried efficiently.



**Process orchestration is a key component in the construction of high-performance, distributed data-intensive applications.**

While each of these data management approaches automatically maintains the necessary file metadata needed to access and manage files, many data-intensive applications, especially in science, have much more demanding metadata management requirements. Understanding how particular analyses are derived at some stage in the future requires capturing provenance information.<sup>9</sup> Provenance captures snapshots of the input and output files to a process, the version of the analysis code executed, and a record of the overall processing steps performed in an analysis pipeline.

Several projects seeking solutions for integrating provenance capture and query with a scientific workflow environment are under way<sup>10</sup> and are showing promising results. Long-term storage of provenance data will add considerably to the storage burden for data-intensive applications. Hence, solutions in which the scientist can guide and optimize when and how provenance is captured will likely emerge as the most attractive and pragmatic options.

A sample data management solution for biology, the PRISM data collection and management system (<http://ncrr.pnl.gov/prism>) currently manages more than 70 Tbytes of proteomics information generated by a variety of mass spectrometers in the Environmental Molecular

Sciences Laboratory at the Pacific Northwest National Laboratory (PNNL). Individual experiments at this national user facility can produce 10 Gbytes of raw data. PRISM's data management system holds the raw spectra and analysis results files and metadata. PRISM's mass tag system tracks all peptides found for each organism and contains the mass tags and results of peak matching for each experimental campaign. A relational database stores metadata about the experiments and the raw and resulting data. Entries in the database point to the actual data files derived from the mass spectrometer and intermediate processing, which are stored as flat files in a hierarchical distributed file system.

### Integration

The data warehouses and data center approaches to data-intensive computing are successful in part because they bring all the data they can analyze to a single logical file system. This supports efficient data access as the analyses can be deployed on processors local to the data, and programs are composed using a single programming model (SQL or Hadoop) and runtime environment.

Unfortunately, many data-intensive applications cannot make this simplifying assumption. The data they need to process is inherently distributed, and legacy analysis codes must be used that are nonhomogeneous in terms of programming models, languages, and execution platforms.

Biology has good examples of this. There are numerous biology-related data sources, including Genbank ([www.ncbi.nlm.nih.gov/Genbank](http://www.ncbi.nlm.nih.gov/Genbank)—more than 110 gigabases stored by February 2008), Entrez ([www.ncbi.nlm.nih.gov/gquery/gquery.fcgi](http://www.ncbi.nlm.nih.gov/gquery/gquery.fcgi)), SwissProt ([www.ebi.ac.uk/swissprot](http://www.ebi.ac.uk/swissprot)), KEGG ([www.genome.jp/kegg](http://www.genome.jp/kegg)), and the Protein Data Bank (PDB; [www.rcsb.org/pdb/home/home.do](http://www.rcsb.org/pdb/home/home.do)—with structural information for more than 46,000 proteins in April 2008).

In these widely distributed, data-intensive applications, copying very large datasets is time-consuming. For example, copying a 1-Pbyte file on a 10-Gbps network with 80 percent utilization takes around 11 days. It therefore becomes crucial that, whenever possible, data analyses execute locally to the data. Achieving this requires mechanisms to invoke computations remotely and return the results efficiently.

In the past few years, Web services have emerged as the predominant means for facilitating distributed computation. Requests and results are transmitted as messages using the SOAP protocol, and these messages can be secured by leveraging existing security investments through Web services security.

Web services are, however, not suitable for transmitting large payloads as they encode data using XML, which can rapidly become verbose as data sizes increase. This has led to solutions in which requests and input

parameters are delivered using Web services, and the results are made available asynchronously over a protocol more amenable for large file transfer, such as the Parallel File Transport Protocol.

As these results typically take some time to produce at the distributed compute resources, their efficient retrieval requires some form of process automation, or orchestration, which is triggered by a notification of availability message. Process orchestration is therefore a key component in the construction of high-performance, distributed data-intensive applications. Approaches based on standards like the Web Services Business Process Execution Language and Kepler (<http://kepler-project.org>) show promise for this purpose, but much research is needed to improve the usability, manageability, and adaptability of these tools for use in large-scale data-intensive applications.<sup>11</sup>

### Analysis

As data sizes increase, so do the compute resources needed to analyze them. In some cases, the analysis scales linearly with data size, and thus is amenable to straightforward parallelization techniques. However, many problems require more complex processing, such as multiple passes over data or graph searching, and they scale superlinearly with data size. As the data sizes increase, these algorithms take exponentially longer to execute and consume vast amounts of resources on high-performance computing platforms.

Undoubtedly, advances in supercomputers will accelerate the performance of these algorithms, although there are challenges here that researchers must address. For example, over the past 20 years, bandwidth has doubled in a period of 1.7-2.9 years, depending on the medium—processor, memory, LAN, disk—while latency has only improved 20 to 30 percent in this doubling period.<sup>12</sup> The speed of light ultimately limits latency; this problem is at the heart of many data-intensive analyses.<sup>13</sup>

To address this limitation, new latency-hiding techniques are emerging in hardware architectures. For example, processors with hardware-based thread support, such as the Cray Threadstorm and Sun's Niagara, can effectively hide data access latencies by efficiently exploiting parallelism through rapid context switching.

But ultimately, in the face of ever-growing data volumes, new algorithms are needed. One approach, *machine learning*, shows promise. It encompasses a wide range of approaches, typically involving Bayesian statistics with or without Markov models. Methods for modeling conditional probability density functions through regression and classification include artificial neural networks, genetic algorithms, genetic programming, K-nearest neighbor, quadratic classifier, and support vector machines. Our laboratory has had some success using support vector ma-

chines on several problems, including homology detection in eukaryotic organisms.<sup>14</sup> The advantages of SVMs include a compact classifier that minimizes the interaction with the data and a training step resistant to overtraining.

### CROSS-CUTTING REQUIREMENTS

Although data-intensive computing needs vary by application area, in all cases, the data-intensive computing capability must provide an integrated and seamless framework of tools, services, and environments between high-performance computing resources and data-intensive analytics. This will entail tightly bundled mathematics, statistics, and computational sciences capabilities, plus linkage to specific scientific domains and widely applicable core capabilities.



**Inexpensive digital sensors and the ability to move the data to analysis engines are opening new opportunities in nearly every field.**

The hardware architecture must be tailored for data-intensive analytics, and it must be scalable and portable to all venues and platforms with the ability to process complex streaming data and large-volume data repositories. This requires a repository of tools, methods, and expertise that can span the spectrum of solutions from highly integrated, complex systems to isolated analyses on focused problems adaptable for use on a variety of applications.

To understand the various data-intensive computing requirements of a range of scientific fields, PNNL held a series of focus groups with domain scientists in the areas of systems biology, climate modeling, energy systems, homeland security, and computational science. The scientists from each of these domains described a need for creating a comprehensive knowledge discovery environment, especially for nonexperts in a specialty area, to ease access to the flood of diverse data, integrate significantly enhanced modeling capabilities, and guide experimenters to perform optimal experiments. From their input, a set of cross-cutting common requirements related to data-intensive computing was compiled, as listed in Table 1. The table presents requirements and challenges for the architecture components (data acquisition, data management, modeling and simulation, algorithms, information analytics, and computing platforms) for each of these application areas.

The focus groups generally believed that transformational, not incremental, approaches are needed to have a major impact. Comprehensive integrated models are re-

**Table 1. Summary of cross-cutting data-intensive computing characteristics and requirements.**

Architecture components	Cross-cutting characteristics and requirements
Data acquisition	• Unsynchronized exploding amounts of high-volume data produced by a wide range of rapidly evolving experimental, computational, and other information sources.
	• Disparate, incomplete, sometimes perishable data, with no standard formats and of unknown pedigree existing in stove-piped data stores.
	• More data volume might not be useful if it is more of the same; need different data to improve the conclusions drawn.
Data management	• Users overloaded by volumes of data.
	• Since all data cannot be saved, decisions are made regarding what data is important to archive even though humans cannot get their arms around the problem.
	• Raw data is of interest to a few, while the processed data derived from heterogeneous sources is of value to many.
	• There is no common data format or architecture to pull data and information together since data is accumulated, stored, and owned by different groups, creating the need to obtain authority for data movement with the incurred cost and time.
Modeling and simulation	• Simulations need to focus on results that can be tested.
	• Modeling is a multiscale problem in space and time.
	• Predictive models are needed for a variety of mission areas.
	• Analysis can be data-driven or model-driven, and requires a common environment for bringing these together.
Algorithms	• Need improved, trusted, open, efficient, and validated algorithms that correspond to the state of knowledge for classes of problems that are scalable from today's data to massive, heterogeneous datasets.
	• Algorithms are needed to capture important events out of the large data stream, most of which is unimportant.
	• Approximations are currently made because of limitations on spatial scale; more accurate algorithms are needed.
	• Need a universal parser to tag information for analysis.
	• Need algorithms to recognize and predict intent by combining all available data.
Information analytics	• Require an integrating informatics resource manager that takes in sensor data, transforming between heterogeneous datasets and integrating computational tools, and presents results to the human user.
	• Since the scale of data and problems overwhelms visual displays, new approaches are needed to develop the appropriate level of abstraction and to condense and select data to display under the user's control.
	• Collaborative sharing and analysis of datasets and observations are desirable.
Computing platforms	• Current architectures are inadequate for the problem set to access large local and distributed datasets, providing solutions with reasonable throughput to analyze and model at the required spatial scales.
	• Current machines do not have the storage capacity for the amount of data that models use and produce.
	• Need self-healing and intrinsically secure operating systems with high-performance networking that provides built-in encryption.
	• Computational needs range from large central high-performance computing systems to portable lightweight systems such as miniaturized labs on a chip for field deployments.

quired for understanding very complex, multistage physical systems. Support is required for high-speed dataflows in various formats, from disparate, heterogeneous data sources, with classified and unclassified sources, over various communication paths being integrated together with a knowledge discovery tool. Large datasets, including those from fielded sensor systems, can be highly perishable, requiring real-time, centralized integration, identification, and action.

## WHERE DO WE GO FROM HERE?

Having outlined the challenges, the question remains, How will progress be made? Will commercial interests be able to drive solutions that support a healthy, data-driven economy or are there fundamental issues that require pre-competitive investments from private and public sources? A related question, and one of general concern to all forms of high-performance computing, is, Will the solutions to commercial problems be adequate to address broader public interests in areas of national security, public health, environmental stewardship, and the scientific research that supports and advances these areas? What role will each of the interested parties play: the scientific research community, industry, standards bodies, public policy makers, and information consumers?

At some level, the limits of engineering and economy constrain the infrastructure equipment providers. The move to multicore and many-core chips is being driven not because it is a good idea, but because given realistic power limits and the effect on increasing clock speed, there is no alternative way to increase performance at the rate industry expectations dictate. Terabyte disks with only minor performance improvements over a decade ago are another example of how the economics and technology of what is possible create their own challenges. Improved memory volumes without associated bandwidth or latency improvements are a similar phenomenon.

Inexpensive digital sensors and the ability to move the data to analysis engines are opening new opportunities in nearly every field: astronomy, physics, chemistry, biology, climate science, and the fields they support, such as public health and threat detection. The constantly changing balance points of device capability are a fact of life, and while uncomfortable in many respects, they power the engine of innovation.

Software that can mask the technology's constant underlying shifts has a tremendous advantage. Google has had great success using the MapReduce<sup>5</sup> programming model for many different purposes. One reason for this success is the model's simplicity. Its popularity has spawned open source versions such as Hadoop. Is this the kind of spark that HTTP provided, revolutionizing our concept of data access? The infrastructure that makes MapReduce successful is complex and evolving, and is not suitable for all

data-intensive environments. However, the model's simplicity has great appeal in a field that easily gets bogged down with the complexities of semantic translation, metadata management, schemas, federated Web services, and a forest of policy domains.

Frameworks such as MapReduce—which can mask a data-intensive environment's underlying complexity so that humans can do what they do best, be creative, without imposing too many restrictions—are the key enablers of data-intensive computing. Managing the internal complexity is a difficult technological problem that sometimes can be addressed with standards when there are competing, equally valid solutions, but might need fundamental research at other times when there are no obvious solutions.

**D**ata-intensive computing is undergoing a rapid transformation driven by the demands of science, engineering, and commerce. Issues abound, while solutions lag, partly due to the difficulty of defining the full scope of the diverse data-intensive problems. Core issues of data-intensive architectures and approaches need clear definition and concerted efforts if progress is to be made before we collapse under the burden of our data-intensive world. ■

## Acknowledgments

This work was supported by Laboratory Directed Research and Development at Pacific Northwest National Laboratory. Pacific Northwest National Laboratory is operated for the US Department of Energy by Battelle under contract DE-AC05-76RLO 1830.PNNL-SA-54166.

## References

1. US Department of Energy, "Data-Management Challenge," report from the DOE Office of Science Data-Management Workshops, Mar.-May 2004; [www.er.doe.gov/ascr/ProgramDocuments/final-report-v26.pdf](http://www.er.doe.gov/ascr/ProgramDocuments/final-report-v26.pdf).
2. US Department of Energy, DOE Genomics: GTL Roadmap—Systems Biology for Energy and Environment, "GTL Roadmap," DOE/SC-0090, 2005; <http://genomicsgdl.energy.gov/roadmap/index.shtml>.
3. US Department of Energy, "Visualization and Knowledge Discovery," report from the DOE/ASCR Workshop on Visual Analysis and Data Exploration at Extreme Scale, Oct. 2007; [www.sc.doe.gov/ascr/ProgramDocuments/DOE-Visualization-Report-2007.pdf](http://www.sc.doe.gov/ascr/ProgramDocuments/DOE-Visualization-Report-2007.pdf).
4. M. Cannataro, D. Talia, and P.K. Srimani, "Parallel Data-Intensive Computing in Scientific and Commercial Applications," *Parallel Computing*, May 2002, pp. 673-704.
5. J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Comm. ACM*, vol. 51, no. 1, 2008, pp. 107-113.
6. "The Diverse and Exploding Digital Universe," white paper, IDC, Mar. 2008; [www.emc.com/digital\\_universe](http://www.emc.com/digital_universe).



7. I. Gorton, "Software Architecture Challenges for Data-Intensive Computing," *Proc. 7th Working IEEE/IFIP Conf. Software Architecture (WICSA 08)*, IEEE CS Press, 2008, pp. 4-6.
8. C.A. Mattmann et al., "Software Architecture for Large-Scale, Distributed, Data-Intensive Systems," *Proc. 4th Working IEEE/IFIP Conf. Software Architecture (WICSA 04)*, IEEE CS Press, 2004, pp. 255-276.
9. L. Moreau and I. Foster, eds., "Provenance and Annotation of Data," *Proc. Int'l Provenance and Annotation Workshop (IPAW 06)*, revised selected papers, LNCS 4145, Springer, 2006.
10. I. Gorton et al., "The MeDiCi Integration Framework: A Platform for High-Performance Data Streaming Applications," *Proc. 7th Working IEEE/IFIP Conf. Software Architecture (WICSA 08)*, IEEE CS Press, 2008, pp. 95-104.
11. Y. Gil et al., "Examining the Challenges of Scientific Workflows," *Computer*, Dec. 2007, pp. 24-32.
12. D.A. Patterson, "Latency Lags Bandwidth," *Comm. ACM*, Oct. 2004, pp. 71-75.
13. G. Bell, J. Gray, and A. Szalay, "Petascale Computational Systems: Balanced Cyber Infrastructure in a Data-Centric World," *Computer*, Jan. 2006, pp. 110-112.
14. A.R. Shaw et al., "Integrating Subcellular Location for Improving Machine Learning Models of Remote Homology Detection in Eukaryotic Organisms," *Computational Biology and Chemistry*, Apr. 2007, pp. 138-142.

**Richard T. Kouzes** is a laboratory fellow at Pacific National Laboratory. His research interests are collaborative computing, neutrino physics, and homeland security. He received a PhD in physics from Princeton University. He is a Fellow of the IEEE and the American Association for the Advancement of Science, and a member of the American

Physical Society and Sigma Xi. Contact him at richard.kouzes@pnl.gov.

**Gordon A. Anderson** is an associate director for scientific resources at Pacific Northwest National Laboratory. His research interests are proteomics data management and analysis. He received a BS in engineering from Washington State University. He is a member of the American Society for Mass Spectrometry. Contact him at gordon.anderson@pnl.gov.

**Stephen T. Elbert** is a manager at Pacific Northwest National Laboratory. His research interests are scalability, efficiency, and productivity of high-performance systems. He received a PhD in computational chemistry from the University of Washington. He is a member of the IEEE, the American Association for the Advancement of Science, and Sigma Xi. Contact him at stephen.elbert@pnl.gov.

**Ian Gorton** is an associate division director at Pacific Northwest National Laboratory. His research interests include software architectures, component technologies, and middleware. He received a PhD in computer science from Sheffield Hallam University. He is a member of the IEEE and a Fellow of the Australian Computer Society. Contact him at ian.gorton@pnl.gov.

**Deborah K. Gracio** is a computational and statistical analytics division director at Pacific Northwest National Laboratory. Her research interests are integrated computational environments and computational biology. She received an MS in electrical engineering from Washington State University. She is a member of the IEEE and the American Association for the Advancement of Science. Contact her at debbie.gracio@pnl.gov.

## Silver Bullet Security Podcast

In-depth interviews with security gurus. Hosted by Gary McGraw.

[www.computer.org/security/podcasts](http://www.computer.org/security/podcasts)

Sponsored by  SECURITY & PRIVACY  eSential