Data-Intensive Scalable Computing

Harnessing the Power of Cloud Computing

Randal E. Bryant

February, 2009

Our world is awash in data. Millions of devices generate digital data, an estimated one *zettabyte* (that's 10^{21} bytes) per year. Much of this data gets transmitted over networks and stored on disk drives. With the dramatic cost reductions in magnetic storage technology, we can readily collect and store massive amounts of data. But, what is all this data good for? Consider the following examples:

The 6000 Wal-Mart stores worldwide record every purchase by every shopper, totaling around 267 million transactions per day. They collect all this information in a 4-petabyte (that's 4 X 10^{15} bytes) *data warehouse* set up for them by Hewlett-Packard. These transactions are a treasure trove of information about the shopping habits of their customers. How much did that that $200 discount on large-screen TVs increase sales, and how much did the shoppers who bought them spend on other things? How many copies of the upcoming John Grisham novel should we stock? Based on long-term weather forecasts, how many snow shovels should we order for our stores in Iowa? Sophisticated machine-learning algorithms can find answers to this question, given the right data and the right computing power.

The proposed Large Synoptic Survey Telescope (LSST) will scan the sky from a mountaintop in Chile, with what can be considered the world's largest digital camera, generating a 3200 megapixel image every 15 seconds, covering the total visible sky every 3 days. That will yield 30 terabytes (10^{12} bytes) of image data every day. Astronomers anticipate being able to learn much about the origins of the universe and the nature of dark matter by analyzing this data.

Every time we have a CAT scan or an MRI taken, millions of bytes are recorded, but currently they are simply turned into a series of cross-sectional images. Imagine a future capability in which the different components of a knee joint could be identified, compared with the data from previous images, and a plan for knee surgery could be generated automatically.

These are just three of dozens of examples where an ability to collect, organize, and analyze massive amounts of data could lead to breakthroughs in business, science, and medicine. Of course, search engine companies have already demonstrated this capability for the information available on the worldwide web, and they've shown they can make good money doing so. But the web is just one of the many possible data sources in our world, and search engines are just one form of data aggregation.

Data storage technology has made dramatic improvements in the cost of storing data. Modern disk drives have capacities measured in terabytes, and they cost less than $100 per terabyte. Consider that a digitized version of all of the text in all of the books in the Library of Congress, essentially the totality of all of humankind's knowledge, would only constitute around 20 terabytes. Of course, data in the form

of images, sound, and video have much higher storage requirements, but still we can think of storage as being an almost-free resource. The challenge is in how to manage and make the most use of all of this data.

At Carnegie Mellon, we've taken on *Data Intensive Scalable Computing* (DISC) as a major focus for our research and educational efforts. We believe that the potential applications for data-intensive computing are nearly limitless, that many challenging and exciting research problems arise when trying to scale up our systems and computations to handle terabyte-scale datasets, and that we need to expose our students to the technologies that will help them cope with the data-intensive world they will live in. Realizing the promise of DISC requires combining the talents of people from across many disciplines, and Carnegie Mellon, with its strengths in engineering, computer science, and many application disciplines, is uniquely qualified to lead the charge.

If we can fit the entire Library Congress on 10 or 20 disk drives at a cost of $2000, what possible technical barriers can there be in realizing low-cost DISC systems and applying them to real-world problems? The core challenge is that communication and computation are much more difficult and expensive than storage. Consider that 1-terabyte disk drive that we can buy for $100. Even if we accept the optimistic claims of the drive manufacturer, we can only read or write 115 megabytes for second between that drive and a computer. That means it would take 2.3 hours just to scan the entire disk. Similarly, it would take around 2.2 hours to transfer that data between two computers connected by a high-speed, local area network, assuming a transfer rate of one gigabit per second. Transferring a terabyte across a slower link, such as over the Internet, can take many hours or even days. So, if we had a terabyte dataset stored on a single disk, then even answering a simple query, such as the average of all of the values, would require several hours, while applying more sophisticated analysis algorithms would be out of the question.

To deal with large scale datasets, we need to spread the data across many disk drives, possibly hundreds, so that we can access large amounts of information in parallel. These disks also need processors and networking, and so we should incorporate them into a *cluster computing system*, comprising around one hundred *nodes*, each having one or two processors, several disk drives, and a high-speed network interface. This cluster can be set up in a machine room, with the nodes and power supplies mounted in racks, and connected by cables. Our proposed $2000 storage system for the Library of Congress has mushroomed into a large-scale computing system costing nearly $1 million.

Programming and operating systems with hundreds of disks and processors is no small task. Consider the issue of reliability. A typical disk drive has a mean time to failure of around 3 years. For the disk drive in my laptop, that means that every 3 years or so I have to deal with the inconvenience of having the drive replaced and restoring its contents from backup storage (hopefully!) But, if we have a system with 150 disk drives, we can expect, on average, than one will fail every week. As we scale up a system, we must anticipate that any of its components—disks, processors, network connections, power supplies, and even software—can fail at any time. Rather than stopping the system every time something goes wrong, we must engineer it to be highly fault tolerant, so that the system can continue operating (perhaps with some degradation in performance) despite multiple component failures. On

the programming side, people have been trying for decades to use parallel computing to improve program performance, yet it remains high on the list of "grand challenge" problems for computer science. The combination of high performance, high reliability, and ease of programming for parallel computing systems remains elusive.

Instead of building and operating our own clusters, one attractive approach is to make use of *cloud computing* systems. The idea here is to let a dedicated organization take on the task of assembling a large system and making it available to others. For data-intensive computing, this can be in the form of a "virtual computing platform," as exemplified by Amazon Web Services (AWS), where customers can buy computer cycles by the CPU-hour and network-accessible storage by the gigabyte-month. (The other cloud-computing model, referred to as "software as a service" provides network-accessible applications, such as email or contact management. This is a valuable service but not suitable for our needs.) From the clients' perspective, cloud computing has the advantage that they do not need to worry about replacing disk drives, dealing with power outages, or updating the operating system software on the nodes with the latest patches. Moreover, they can scale up their computing and storage capacity without building and provisioning new data centers.

On the software side, an open-source framework called "Hadoop," styled after the system developed and used internally at Google to run its web crawling and indexing, makes it fairly easy to develop applications that manage and analyze large amounts of data on cluster systems. The Hadoop File System handles the difficult issues of coordinating the activities of the node processors and disks to implement a large-scale, reliable file system. For example, most files are stored in triplicate on multiple disks, so that there are always backup copies of every file if a disk drive fails. Programmers can write code that operates on data spread across thousands of files via the *Map/Reduce* model pioneered by Google. With this approach, the programmer describes computations to perform on the individual files (map), as well as how to combine the results of these computations (reduce) to produce data in the form of multiple files, with a typical program consisting of a sequence of Map/Reduce steps. The runtime system then handles the low-level details of scheduling the map and reduce tasks on the cluster processors and rerunning any tasks that fail. This frees the programmer from the traditional parallel programming concerns of data placement, scheduling, and failure recovery. Software like Hadoop make it possible to implement scalable and reliable applications running on otherwise unreliable and difficult to manage hardware.

Several companies have made their systems available for use by university researchers and educators. At Carnegie Mellon, we have been fortunate to have access to M45, a cluster system owned and operated by Yahoo!, to foster the growth of Hadoop and other open-source projects (Yahoo! has been the major contributor to Hadoop. More recently, Google and IBM have joined forces to make large-scale machines available to U.S. universities under the auspices of the National Science Foundation. A number of universities are making use of the platform provided by Amazon Web Services, and Microsoft is developing systems and programming tools that are well-suited for data-intensive computing.

Why are these companies providing access to their computer clusters? For one thing, universities supply students who will soon be working on these very systems. If students never see anything more

than a program running on a single machine and operating on datasets of one gigabyte or less, then they won't be prepared to support and make use of large-scale, data-intensive computing. They won't have even thought about the insights that can be gained from analyzing terabytes of data.

There's a benefit to universities beyond education. We're interested in working on computing problems where the amount of data is so large that we can't own and operate the machines that would store and process it. Computer scientists, of course, aren't interested in "cloud computing" so that we can provide something as simple as an email service. Instead, we're working on projects that require very large datasets to be analyzed quickly and accurately. For instance, a group of researchers in the Language Technologies Institute, led by Maxine Eskenazi, has been working to gather documents that can be used to teach English as a second language and in English education for elementary-school pupils. Although there are plenty of documents available on the Web that could be used for reading practice or instruction, making sure that they're suitable for different reading levels isn't easy. They had to sift through 200 million Web pages just to collect 200,000 useful documents. Only large-scale, data-intensive computing makes that possible.

Another research group, led by Stephan Vogel of the Language Technologies Institute, is improving machine translation of one language to another. The modern approach to translating texts from, say, Chinese to English is to scan millions or billions of documents in both languages, and then build statistical models that look for certain combinations of words and make their best guesses at what English words and phrases match those in Chinese. The success of these translation programs is greatly improved when they can be trained with more and more data—we're now talking in terms of trillions of words. Without a large, scalable cluster of computers, dealing with that amount of data would be impossible.

To cite a third example, Alexei Efros of the Computer Science Department and the Robotics Institute, along with his PhD student James Hay, has downloaded six million landscape images from Flickr and set up a program that looked for common features. Using geographic data attached to some of those photos, they then created computer models that were able to identify—with a surprising level of accuracy—where the different images were taken.

Beyond making use of existing clusters and software frameworks, there are many important research problems to be addressed to fully realize the potential of data-intensive computing. How can processor, storage, and networking hardware be designed to improve performance, energy efficiency, and reliability? How can we run a collection of data-intensive computations on the system simultaneously? What programming models and languages can we devise to support forms of computation that do not fit well in the Map/Reduce model? What machine-learning algorithms can scale to datasets with billions of elements? As a research organization, the School of Computer Science views DISC as a source of a large number of exciting opportunities.

There are advantages that universities bring to research in data-intensive computing that companies cannot match. First, the research we do is completely open. We publish our findings and share information freely, which private industry seldom does. Second, history has shown over and over again

that when you give creative people new capabilities, they will come up with new ideas that the originators never imagined were possible. The Worldwide Web, after all, was never one of the original applications imagined for the Internet, but its impact has been transformative.

There are also research problems that can benefit society but have no prospect of forming a profitable business. Companies will not profit from basic scientific research in astronomy, for instance, but that research will enable us to deepen our understanding of the universe.

Within the next five years, more and more research projects at Carnegie Mellon will involve analyzing huge amounts of data on very large-scale machines. Our educational efforts are broadening as well. We've created courses at both the undergraduate and graduate level that give students a chance to run programs on the cluster provided by Google and IBM. We're also involved in several efforts to help the National Science Foundation provide course and teaching materials to get students at other universities involved in large-scale, data-intensive computing.

Data-intensive, scalable computing is revolutionizing our ability to gather and analyze information in all forms. It will lead to new discoveries in science and new forms of entertainment. It will lead to improvements in business practices—and, just like any other disruptive technology, it will transform many different businesses for better and worse. Data-intensive computing will change the way we think about science—the ability to collect and make use of so much data will change our perspective on the kinds of scientific and medical experiments that we consider attempting.

Data-intensive computing will change our education and our research in computer science at Carnegie Mellon profoundly. As our students go off into the world, they must understand this new paradigm in computing. They need to know how to use the technology and fully exploit the capabilities that large-scale computing make possible. At the School of Computer Science, we are committed to presenting our students with these new possibilities and providing them with the right kind of tools and thinking to help this technology advance.