Big Data — It's not just for Google Any More

The Software and Compelling Economics of Big Data Computing

EXECUTIVE SUMMARY

Big Data holds out the promise of providing businesses with differentiated competitive insights. But those insights, protected by a healthy level of technical complexity, require large-scale data consumption and an innovative approach to the analysis of large public and private data sets. Currently, production Big Data implementations routinely process petabytes and even exabytes of data on a daily basis. Indeed, many of the technologies and implementation strategies for Big Data originated with web companies struggling with unprecedented volumes of data. At the same time, the economics of cloud computing — predicated on declines in the cost of data storage, bandwidth and computing time — point inevitably to the promise of Big Data rapidly becoming a reality for even comparatively small companies.

Table of Contents

INTRODUCTION

Few emerging technologies embody both the potential and the complexity of the digital landscape as clearly as Big Data. Data resides in data sets around the globe in amounts that were all but inconceivable even ten years ago. The explosion of data has resulted from numerous sources, such as the proliferation of consumers using mobile digital data creation devices — from digital cameras to smart phones — and the ubiquitous internet-connected digital monitoring and recording devices used in the public and private sectors and throughout global industry. Moreover, this new data, by virtue of ubiquity, mobility, and traceability, is often rich in metadata that can include its creator's location, identity, purpose, and the data's time and medium of creation.

DATA SCIENCE AND THE BIG DATA OPPORTUNITY

The availability of this new universe of data and the emergence of analytic technologies and methodologies capable of combing through it has had wide impact. One consequence has been that, in science and medicine, the use of empiricism, rather than experimentation, is a growing trend, enabled by large data sets that provide statistical insights. Similarly, successful machine language translation is statistics-based, benefitting from the availability of large data sets.

Another consequence of Big Data has been the growing demand for software skill-sets in areas that both enable, and benefit from, Big Data, such as:

- Machine Learning
- Statistics
- Networking

In a classic case of "necessity is the mother of invention", large web companies sought to capitalize on the valuable information they were accumulating by computing at web scale. As a result, web companies have contributed various technologies to the emerging "Big Data" software stack. We'll look more closely at the role web companies have played in Big Data technology later in this paper.

THE COMPETITIVE EDGE OF COMMODITY HARDWARE: IN THE CLOUD OR IN THE ENTERPRISE.

The technologies emerging from web companies would not alone be enough to sustain real growth in Big Data were it not for accompanying economic trends. Chief among these is the decline in the cost of bandwidth to a level at or about $150 per TB transferred and falling. To put that in perspective, streaming a movie in 1998 at Netflix's current encoding bitrate would have cost $270 per movie. Today their cost is $.05.[1]

Scenario: Retrieve and catalog all of the images on the internet

How does the scenario play out in the world of Big Data? Consider a hypothetical application designed to retrieve and catalog all of the images on the internet. As part of the process, the application will convert each image to a 100X100 thumbnail, compute an MD5 hash on each image, extract EXIF data and store when available, and then push the generated catalog to a central location.

Let's estimate the cost per billion images to create an image catalog of all images on the net. For the purposes of this exercise, we'll assume that the technologies used include:

- Nutch (An apache open-source search engine based on Hadoop.)
- Hadoop
- Amazon EC2
- Amazon Elastic Block Store (A web service for storage but with better performance characteristics than S3.)

We'll use the following cost estimate methodology. We'll randomly select approximately 70,000 domain names, crawl the home page of each, and extract all images from the crawled pages. Next, we'll randomly select 100,000 images from the gathered set. Then, we will retrieve and process selected images as described above. From the costs determined in this exercise, we can extrapolate the cost per billion.

Crunching the numbers[2]

First, let's look at numbers associated with computing and bandwidth capacity. Low-end EC2 instances can successfully retrieve and process about 100,000 images per hour. Once processed, the resulting image data within the catalog averages approximately 2k bytes compressed. Further, we find that it takes about 10,000 hours of computing time to retrieve 1 billion images, with a storage requirement of about 2 TB of data per billion.

We therefore find that the server cost per billion images is $850. This is because low-end EC2 instances sell for $.085 per hour.

We also find that the storage cost is $0. This is because the data will not be stored by Amazon.com, but will be pushed to a site owned by the developer.

Next, bandwidth costs inbound are $0, since inbound bandwidth on Amazon.com is currently free. However, we can assume $750 per billion images in the future at current bandwidth prices. Bandwidth costs outbound are about $300 per billion, noting the cost of $.15 per GB with 1 billion images yielding 2TB of data.

The resulting total cost, then, is about $1,200 per billion images, rising to as much as $2,000 per billion if Amazon.com starts charging for inbound bandwidth.

Some additional considerations are that data acquisition at a rate of 1 billion images per month requires about fifteen servers. Low-end EC2 instances can push about 30mbps of bandwidth, which, multiplied by fifteen servers, puts us at about 450 mbps of bandwidth consumed to retrieve 1 billion images per month.

Cost in context[3]

What if we had performed this same exercise in 2000? To begin, there was no cloud-computing option, so we would have had to actually purchase and provision fifteen rack mounted servers at a cost of about $75,000. Costs associated with a data center infrastructure to support the servers, such as racks, routers and switches, would have been, conservatively, $250,000. The cost of 450 mbps of bandwidth per month would have run about $300,000.

In short, the costs would have been not only $325,000 up front capital investment, but also some $300,000 per month in ongoing bandwidth costs. This puts the cost at about $500,000 just to retrieve 1 billion images processed.

ARCHITECTING FOR BOTH SIDES OF THE DATA EQUATION

Although the cost of working with Big Data declines continuously, working with Big Data still requires us to solve at least two complex software challenges. First, we must find an effective way to harness existing computing power so that it can process Big Data-scale data sets. Second, we must find a way to serve the data that differs fundamentally from current, standard database serving approaches.

Big Data Processing

As with many of the tools and methodologies enabling Big Data, the solutions to the processing software challenge emerged from web-based companies. In 2004, Google released a paper entitled *MapReduce: Simplified Data Processing on Large Clusters*[4] arguing that the MapReduce programming pattern can be used to solve a surprising number of data processing problems, and that such a pattern can be used as a way to dynamically scale the execution of MapReduce programs across an arbitrarily large cluster of machines.

An open source, Java-based implementation of MapReduce known as *Hadoop*, released by Doug Cutting, developer of Lucene and Nutch, among other things, was adopted as an Apache Foundation project and has become a widely used platform and runtime environment for the deployment of Big Data applications.

Hadoop provides a scalable runtime environment that handles most of the complexities associated with doing large scale data processing jobs across large clusters. Hadoop includes both a distributed file system and a distributed runtime for map/reduce execution. Hadoop deploys code, partitions and organizes data, splits jobs into multiple tasks, schedules those tasks taking into account data locality and network topology, and handles failures and their associated restarts. Developers who implement the appropriate map/reduce interfaces in their code receive the benefit of exploiting the compute and I/O capacity of large clusters transparently to their own implementation. In a sense, Hadoop gives them horizontal scalability for free.

Hadoop has become a platform on which developers have layered other kinds of interfaces. Solutions exist, for instance, that accept SQL input and dynamically generate map/reduce jobs that run on Hadoop to extract data stored within Hadoop's distributed file system. Generalized workflow engines have been built on Hadoop. These kinds of systems are built to simplify the ability of end-users to take advantage of the scalable architecture and Big Data capacity of a platform like Hadoop without requiring them to master the MapReduce pattern on which Hadoop is based.

Although Hadoop is an outstanding infrastructure for Big Data Processing, it is not a suitable platform for data serving. Hadoop isn't designed to respond to fine-grained network requests arriving from end-users who expect low latency responses. Serving for Big Data, therefore, requires a different solution set.

Big Data Serving

Just as Big Data calls for new approaches to computing, it also requires new approaches to data serving. Today, single web sites routinely serve user communities larger than the entire population of whole continents. Low-latency responses are a bellwether of web site quality, since users don't like, and will generally not tolerate, long wait times when connecting to web sites. Businesses therefore require highly-available, fast-responding sites in order to compete, but no single server offers sufficient capacity for web-scale computing. As a result, web site operators have embraced a scale-out architecture. Scale-out architectures generally involve the use of highly distributed software architectures in conjunction with the deployment of fleets of commodity servers.

The fundamental challenge in big data serving is to facilitate low latency responses from enormous data repositories. Because traditional databases are often unable to scale to the levels required by the largest web sites, a number of alternative approaches have emerged. One approach often used in conjunction with traditional databases is to use a distributed, in-memory cache alongside an RDBMS. These distributed caches (e.g. memcached[5]) can cache query results and are designed to significantly reduce database overhead by reducing the number of queries. Facebook has contributed significantly to projects such as memcached.[6]

Beyond merely improving existing open-source products, some web companies have open-sourced entirely original software packages. Facebook has released a product called *Cassandra* through the Apache Foundation. Cassandra is described as "a distributed storage system for managing very large amounts of structured data spread out across many commodity servers."[7] Cassandra provides a low-latency persistent database that scales across many machines.

One aspect of achieving low-latency responses in highly distributed systems is to minimize the coordinated transactive overhead of updates and maximize read performance. In that vein, Cassandra represents a persistence model of "eventual consistency". This capitalizes on a key characteristic of many big data applications, which is that they don't have the need for immediate consistency across all server instances.

SUMMARY

As the amount of data being stored around the globe continues to rise and the cost of technologies that enable the extraction of meaningful patterns from that data continues to fall, more and more companies can benefit from the power of Big Data. The current generation of Big Data technologies — as exemplified by Hadoop — represents a relatively early incursion into the Big Data universe. But few would argue that, even today, the benefits of Big Data may constitute a game-changing competitive advantage for any business that is prepared tap into it.

[1] http://blog.streamingmedia.com/the_business_of_online_vi/2010/01/bandwidth-pricing-trends-cost-to-stream-a-movie-today-five-cents-cost-in-1998-270.html. Comcast 25 mbps of home bandwidth would have cost the consumer $25k per month in 1998.

[2] Results of tests run on Amazon's EC2 platform by Advanced Micro Devices, Inc., using Amazon's current pricing (found at http://aws.amazon.com/ec2/#pricing). At the time of the testing, inbound bandwidth at Amazon was free of charge.

[3] Bandwidth costs were extrapolated from various data regarding bandwidth prices a decade ago (e.g. http://www.networkworld.com/news/2000/0626carrier.html). The server costs were a conservative estimate assuming rack mounted server costs of $5k per.

[4] http://static.googleusercontent.com/external_content/untrusted_dlcp/labs.google.com/en/us/papers/mapreduce-osdi04.pdf

[5] http://memcached.org/

[6] http://www.facebook.com/note.php?note_id=39391378919

[7] http://www.cs.cornell.edu/projects/ladis2009/papers/lakshman-ladis2009.pdf