CS498
Rough Draft
Weaver, Justin
Woods, Aaron

Definition: Big Data

1. Big Data refers to a corpus of data large enough to benefit significantly from parallel computation across a fleet of systems, where the efficient orchestration of the computation is itself a considerable challenge. [Bringing Authoritative Computer to Data]

Precursors to Big data

Microarrays - http://learn.genetics.utah.edu/content/labs/microarray/

What other sources of big data?

microfilm,

Importance:

Big data represents data from various aspects of our digital lives. We already know about Amazon and Google collecting our click-through data to better provide suggestions, better our search experience, and increase revenue. Other areas where data is collected are: scientific research and experiments, peer-to-peer communications (such as texts, chat lines, and digital cellular communication), social networking (facebook, twitter, pinterest, imagur, etc, etc), authorship (digital books, and other multimedia content), administrative data (enterprise, government, healthcare, legal, and financial records), business data (marketing, advertising, e-commerce, stock markets, and business intelligence).

Aggregation:

Big data projects are officially defined as those that generate so many pieces of information that computers are needed to sort through it all. (big data, big challenges; p24)

The Exome Aggregation Consortium has pooled data on more than 90,000 people’s genomes. The size of the database is about 925 Terabytes, more than nine times the size of the Library of Congress’ print collection.

Issues in Big Data

Scientists are finding that results from data collection are differing between labs. For example, if two samples of identical data are sent to separate labs, there is no guarantee that the results will be the same. Consequently researchers are trying to isolate what steps in the research and data collection could have caused the divergence. This is just one example of issues that scientists are facing with respect to big data. Another issue is that there is no standards or schema definitions that are used for storing data in databases. This is probably even more of an issue than the previously mentioned one because it makes merging sets difficult, if not impossible. (big data, big challenges; p 25)

Small variations in the data collection and analysis methods are causing concern, as a certain amount of chaos is injected into the system, which is difficult to isolate and mitigate. In the case of biological microbe analysis, researchers are still able to differentiate samples from those who are sick, and those who aren't. However the collection of data is susceptible to contamination, causing the overall dataset to be only partially reliable. (big data, big challenges; p26).

Although it seems that there should be some standards for data collection, scientists need to attempt to isolate “noisemakers” in their studies as opposed to conforming to a single protocol. (big data, big challenges; p26).

Data size is another issue, but not that the data set is too big, but that the big-data set that is being utilized and maintained may not be large enough for a particular problem. Take for instance trying to isolate the genes responsible for a rare genetic disorder. If the ratio of afflicted is very low, as in 1 in a 200,000, there could be an issue collecting enough data to conduct any substantial research. This can lead to misdiagnosis by associating genetic variation with causation. (big data, big challenges; p26).

Corruption of big data sets have been occurring since the microarrays were popular. As a result, scientists are a little better prepared to tackle this issue in the digital age.

CS498
Rough Draft
Weaver, Justin
Woods, Aaron

Storage:

[Bringing Authoritative Computer to Data]

Maintaining the infrastructure

The amount of data needed to be stored is growing at an ever-increasing rate. With this increased need for data storage comes an increased need for the physical medium upon which the data needs to be stored. On a local machine, the time needed to read and write data is acceptable because it is typically only the data of a single user being processed at any given time. In large-scale data storage facilities however, this is not the case. Large amounts of data will be processed at any given time in order to service a large number of users. The standard file reading approach that is used on a personal computer would be extremely slow if it had to scan through all the data stores until it finally found the bit of information it was looking for.

The model described above will not scale well as data sets become larger and larger. The obvious answer to this problem is a distributed file system, and a method of accessing it in a parallel fashion. In a distributed file system, each node contains only a portion of the total data. Instead of needing to look through each disk to find a piece of data, a central controller can broadcast to each node to look within their own set of data for a match. This approach immediately reduces the amount of time by *n*, where *n* is the number of nodes in the distributed file system. This approach also allows each individual user to be serviced much faster. Over the same period of time, the number of user requests that can be processed will be much larger than the sequential approach.

Another factor in making these operations is to treat each piece of data as an immutable object. By doing this, whenever there are updates that need to be made to a piece of data, the old one can simply be deleted and replaced with the updated version. This prevents the need for searching for specific bits within the data to modify. The number of instructions that are potentially reduced by this implementation are also significant.

A concern not yet discussed is reliability of hardware. The system must be constructed in such a way that any piece of data written to the file system is guaranteed to be there until a user wishes to delete it. Stated differently, data must persist even through hardware failure. To do this, multiple copies of each piece of data are stored, each piece in a different location. When this backing-up is done automatically, the process of restoring data at any particular location is reduced to a copying operation.

In 2013, Huawei an information and communications technology solution provider announced that it’s OceanStor 9000 managed to score top benchmarks for Operations Per Second (OPS). The OceanStor 9000 acheived 5,030,264 OPS in a network file system environment, which is about three times better than the existing technology on the market.

Processing - MapReduce

CS498
Rough Draft
Weaver, Justin
Woods, Aaron

There exists a widespread need to process large amounts of data in a quick, efficient, and accurate manner. We have reached a point in computation where dealing with petabytes (about 1000 Terabytes) of data is a reality. Many of these mechanisms in place are parallel in nature, and are intended to run on clusters of hundreds, if not thousands of processors; but can also be scaled back to a single processor.

Some examples of data collection are the Large Synoptic Survey Telescope, which generates about 30 Terabytes of data per day and Facebook, which generates about 15 Terabytes of data each day. Aside from the immense amounts of raw data that are collected, researchers are attempting to find new ways to combat the latency between multicore processors and mechanical hard-disks is continuing to grow. (The Family of MapReduce and Large-Scale Data Processing Systems).

Researchers require a systematic and generic solution which is scalable to future data growth. Enter Hadoop MapReduce (make sure you are at least in an aircraft hangar, that is one big elephant), the most popular open source MapReduce implementation. Originally, MapReduce was proposed by Google. This allows scientists and researchers to grow their computational power by adding inexpensive nodes to a network, as opposed to updating a single computer to the latest and greatest available components.

MapReduce accepts as input a set of key-value pairs. It then passes these inputs to a reduction method, which has a unique algorithm for transforming the data into whatever form of useful data is desired. Users of MapReduce only have to define the Map and Reduce methods, the magic elephant will take care of the rest. Hadoop operates over a distributed file system called Hadoop Distributed File System (HDFS). This system is best portrayed by the Single Program Multiple Data (SPMD) parallel programming paradigm.

There are other solutions available, such as a parallel database, but these solutions are typically expensive, difficult to maintain and tune for performance, and lack fault tolerance. Hadoop is a redundant system where if one of the nodes goes down or faults during execution time, another node will be able to step in and complete where the faulty node left off.

This infrastructure also aligns with the cloud computing paradigm, which further reduces the cost associated with big data processing services. Java programmers with Hadoop experience are in high demand in industry because of the simplicity of the framework. Aside from the administrative upkeep of the actual program, such as writing the driver and relevant objects, there is only two methods that Hadoop requires be defined. Map, which takes the raw input and is gleaned for important data, and Reduce which merges the potentially massive data sets into something more manageable.

One of the main limitations of MapReduce is the lack of support for joining multiple jobs in a single pass. It is possible to do multiple MapReduce steps on a single dataset, and on additional datasets on the fly. Another limitation is that the data must be reloaded between MapReduce jobs. Because the framework is based on a distributed file system, a multi-step MapReduce still must have the results of the previous MapReduce propagated throughout the HDFS. This is an additional resource expense that includes wasted I/O, network bandwidth, and cpu resources. (The Family of MapReduce and Large-Scale Data Processing Systems)

Parallel Programming Paradigms and Frameworks in Big Data Era

According to IBM, 90% of the world's data has been amassed only since 2010. Let that sink in. Thousands upon thousands of years has our world been in existence. Civilizations have come and gone, empires have risen and fallen. There have likely existed great societies of which there is no record today. Within the last five years though, we have accumulated, aggregated, indexed, and stored 90% of the entire world's recordable data. That speaks volumes of how far we have come as a species, and in what direction we are heading in the centuries to come.

Big Data's Place in diverse fields of interest:

genetics, microbiology, neurology - (big data, big challenges)

physics (particle accelerator), astronomy (telescopes) - The Family of MapReduce and Large-Scale Data Processing Systems

Academic Search Premier:

1. Hesman Saey, Tina. "Big Data, Big Challenges. (Cover Story)." 22-27.
   Science News 187.3 (2015).
   Academic Search Premier. Web. Feb. 2015.

2. Dobre, Ciprian, and Fatos Xhafa. "Parallel Programming Paradigms And Frameworks In Big Data Era." 710-738.
International Journal Of Parallel Programming 42.5 (2014).
Academic Search Premier. Web. Feb. 2015

3. SAKR, SHERIF, ANNA LIU, and AYMAN G. FAYOUMI. "The Family Of Mapreduce And Large-Scale Data Processing Systems."
ACM Computing Surveys 46.1 (2013): 11-11:44.
Academic Search Premier. Web. Feb. 2015.

ACM Digital Library:

1. CAVAGE, MARK, and DAVID PACHECO. "Bringing Arbitrary Compute To Authoritative Data." 40-48.
Communications Of The ACM 57.8 (2014).
Business Source Premier. Web. Feb. 2015.

2. Dittrich, J., Quiane-Ruiz, J., Jindal, A., Kargin, Y., Setty, V., and Schad, J., “Hadoop++: Making a Yellow Elephant Run Like a Cheetah (Without it Even Noticing)”
Proceedings of the VLDB Endowment, Volume 3 Issue 1-2, September 2010, Pages 515 - 529

ProQuest Computing:

1. Huawei; Huawei. “OceanStor 9000 Big Data Storage System Tops SPEC Benchmark Test for the Third Consecutive Year.”
Information Technology Newsweekly, 112. (2013).
ProQuest Computing. Web. Feb. 2015.

Keywords:
Big Data, Parallelism, Hadoop, MapReduce, Management, Storage, Retrieval, Indexing, Aggregation, database,distributed file system, compression.