**Chapter 14**

**Big Data Analytics and NoSQL**

**Discussion Focus**

Start by explaining that Big Data is a nebulous term. Its definition and the composition of the techniques and technologies that are covered under this umbrella term are constantly changing and being redefined. There is no standardizing body for Big Data or NoSQL so there is no one in charge to make a definitive statement about exactly what qualifies as Big Data. This is made worse by the fact that most technologies for big data problems and the NoSQL movement are open-source so even the developers working in the arena are often a loose community without hierarchy or structure.

As a generic definition, Big Data is data of such volume, velocity, and/or variety that it is difficult for traditional relational database technologies to store and process it. Students need to understand that the definition of Big Data is relative, not absolute. We cannot look at a collection of data and state categorically that it is Big Data now and for all time. We may categorize a set of data or a data storage and processing requirement as a Big Data problem today. In three years, or even in one year, relational database technologies may have advanced to the point where that same problem is no longer a Big Data problem.

NoSQL has the same problem in terms of its definition. Since Big Data and NoSQL are both defined in terms of a negative statement that says what they are not instead of a positive statement that says what they are, they both suffer from being ill-defined and overly broad.

Discuss the many V’s of Big Data. The basic V’s, volume, velocity, and variety are key to Big Data. Again, because of the lack of an authority to define what Big Data is, other V’s are added by writers and thinkers who like to jump on the alliteration of the 3 V’s. Beyond the 3 V’s, the other V’s that are proposed by various sources are often not really unique to Big Data. For example, all data have Volume. Big Data problems require Volume that is too large for relational database technologies to support. Veracity is the trustworthiness of the data. All data needs to be trustworthy. Big Data problems do not require support for a higher level of trustworthiness than relational database technologies can support. Therefore, the argument can be made that veracity is a characteristic of all data, not just Big Data. Students should understand that critical thinking about Big Data is necessary when assessing claims and technologies in this fast-changing arena.

Discuss that Hadoop has been the beneficiary of great marketing and widespread buy-in from pundits. Hadoop has become synonymous with Big Data in the minds of many people that are passingly familiar with data management. However, Hadoop is a very specialized technology that is aimed at very specific tasks associated with storing and processing very large data sets in non-integrative ways. This makes the Hadoop ecosystem very important because the ecosystem can expand the basic HDFS and MapReduce capabilities to support a wider range of needs and allow greater integration of the data.

Stress to students that the NoSQL landscape is constantly changing. There are about 100 products competing in the NoSQL environment as any point in time, with new entrants emerging almost daily and other products disappearing at about the same rate. The text follows the standard categories of NoSQL databases that appear in the literature, as shown below, but many products do not fit neatly into only one category:

* Key-value
* Document
* Column family
* Graph

Each category attempts to deal with non-relational data in different ways.

Data analysis focuses on attempting to generate knowledge to expand and inform the organization’s decision making processes. These topics were covered to a great extent in Chapter 13 when analyzing data from transactional databases integrated into data warehouses. In this chapter, the use of exploratory and predictive analytics are applied to non-relational databases.

**Answers to Review Questions**

1. **What is Big Data? Give a brief definition.**

Big Data is data of such volume, velocity, and/or variety that it is difficult for traditional relational database technologies to store and process it.

1. **What are the traditional 3 Vs of Big Data? Briefly, define each.**

Volume, velocity, and variety are the traditional 3 Vs of Big Data. Volume refers to the quantity of the data that must be stored. Velocity refers to the speed with which new data is being generated and entering the system. Variety refers to the variations in the structure, or the lack of structure, in the data being captured.

1. **Explain why companies like Google and Amazon were among the first to address the Big Data problem.**

In the 1990s, the use of the Internet exploded and commercial websites helped attract millions of new consumers to online transactions. When the dot-com bubble burst at the end of the 1990s, the millions of new consumers remained but the number of companies providing them services reduced dramatically. As a result, the surviving companies, like Google and Amazon experienced exponential growth in a very short time. This lead to these companies being among the first to experience the volume, velocity, and variety of data that is associated with Big Data.

1. **Explain the difference between scaling up and scaling out.**

Scaling up involves improving storage and processing capabilities through the use of improved hardware, software, and techniques without changing the quantity of servers. Scaling out involves improving storage and processing capabilities through the use of more servers.

1. **What is stream processing, and why is it sometimes necessary?**

Stream processing is the processing of data inputs to make decisions on which data should be stored and which data should be discarded. In some situations, large volumes of data can enter the system as such a rapid pace that it is not feasible to try to actually store all of the data. The data must be processed and filtered as it enters the system to determine which data to keep and which data to discard.

1. **How is stream processing different from feedback loop processing?**

Stream processing focuses on inputs, while feedback loop processing focuses on outputs. Stream processing is performed on the data as it enters the system to decide which data should be stored and which should be discarded. Feedback loop processing uses data after it has been stored to conduct analysis for the purpose of making the data actionable by decision makers.

1. **Explain why veracity, value, and visualization can also be said to apply to relational databases as well as Big Data.**

Veracity of data is an issue with even the smallest of data stores, which is why data management is so important in relational databases. Value of data also applies to traditional, structured data in a relational database. One of the keys to data modeling is that only the data that is of interest to the users should be included in the data model. Data that is not of value should not be recorded in any data store – Big Data or not. Visualization was discussed and illustrated at length in Chapter 13 as an important tool in working with data warehouses, which are often maintained as structured data stores in relational DBMS products.

1. **What is polyglot persistence, and why is it considered a new approach?**

Polyglot persistence is the idea that an organization’s data storage solutions will consist of a range of data storage technologies. This is a new approach because the relational database has previously dominated the data management landscape to the point that the use of a relational DBMS for data storage was taken for granted in most cases. With Big Data problems, the reliance on only relational databases is no longer valid.

1. **What are the key assumptions made by the Hadoop Distributed File System approach?**

HDFS is designed around the following assumptions:

High volume

Write-once, read-many

Streaming access

Fault tolerance

HDFS assumes that the massive volumes of data will need to be stored and retrieved. HDFS assumes that data will be written once, that is, there will very rarely be a need to update the data once it has been written to disk. However, the data will need to be retrieved many times. HDFS assumes that when a file is retrieved, the entire contents of the file will need to be streamed in a sequential fashion. HDFS does not work well when only small parts of a file are needed. Finally, HDFS assumes that failures in the servers will be frequent. As the number of servers increases, the probability of a failure increases significantly. HDFS assumes that servers will fail so the data must be redundant to avoid loss of data when servers fail.

1. **What is the difference between a name node and a data node in HDFS?**

The name node stores the metadata that tracks where all of the actual data blocks reside in the system. The name node is responsible for coordinating tasks across multiple data nodes to ensure sufficient redundancy of the data. The name node does not store any of the actual user data. The data nodes store the actual user data. A data node does not store metadata about the contents of any data node other than itself.

1. **Explain the basic steps of MapReduce processing.**

* A client node submits a job to the Job Tracker.
* Job Tracker determines where the data to be processed resides.
* Job Tracker contacts the Task Tracker on the nodes as close as possible to the data.
* Each Task Tracker creates mappers and reducers as needed to complete the processing of each block of data and consolidate that data into a result.
* Task Trackers report results back to the Job Tracker when the mappers and reducers are finished.
* The Job Tracker updates the status of the job to indicate when it is complete.

1. **Briefly explain how HDFS and MapReduce are complementary to each other.**

Both HDFS and MapReduce rely on the concept of massive, relatively independent, distributions. HDFS decomposes data into large, independent chunks of data that are then distributed across a number of independent servers. MapReduce decomposes processing into independent tasks that are distributed across a number of independent servers. The distribution of data in HDFS is coordinated by a name node server that collects data from each server about the state of the data that it holds. The distribution of processing in MapReduce is coordinated by a job tracker that collects data from each server about the state of the processing it is performing.

1. **What are the four basic categories of NoSQL databases?**

Key-value database, document databases, column family databases, and graph databases.

1. **How are the value components of a key-value database and a document database different?**

In a key-value database, the value component is nonintelligible for the database. In other words, the DBMS is unaware of the meaning of any of the data in the value component – it is treated as an indecipherable mass of data. All processing of the data in the value component must be accomplished by the application logic. In a document database, the value component is partially interpretable by the DBMS. The DBMS can identify and search for specific tags, or subdivisions, within the value component.

1. **Briefly explain the difference between row-centric and column-centric data storage.**

Row-centric storage treats a row as the smallest data storage unit. All of the column values associated with a particular row of data are stored together in physical storage. This is the optimal storage approach for operations that manipulate and retrieve all columns in a row, but only a small number of rows in a table. Column-centric storage treats a row as a divisible collection of values that are stored separately with the values of a single column across many rows being physically stored together. This is optimal when operations manipulate and retrieve a small number of columns in a row for all rows in the table.

1. **What is the difference between a column and a super column in a column family database?**

Columns in a column family database are relatively independent of each other. A super column is a group of columns that are logically related. This relationship can be based on the nature of the data in the columns, such as a group of columns that comprise an address, or it can be based on application processing requirements.

1. **Explain why graph databases tend to struggle with scaling out?**

Graph databases are designed to address problems with highly related data. The data that appears in a graph database are tightly integrated and queries that traverse a graph focus on the relationships among the data. Scaling out requires moving data to number of different servers. As a general rule, scaling out is recommended when the data on each server is relatively independent of the data on other servers. Due to the dependencies among the data on different servers in a graph database, the inter-server communication overhead is very high with a graph database. This has a significant negative impact on the performance of graph databases in a scaled out environment.

**18. Explain what it means for a database to be aggregate aware.**

Aggregate aware means that the designer of the database has to be aware of the way the data in the database will be used, and then design the database around whichever component would be central to that usage. Instead of decomposing the data structures to eliminate redundancy, an aggregate aware database is collects, or aggregates, all of the data around a central component to minimize the structures required during processing.