CE956: Statistical Learning
Department of Computer Engineering
Sharif University of Technology
Spring 2019: Room CE204, Sat. & Mon.: 13:30-15:00

Quiz 05 (20 Points) – (May-04-2019)

Solution

Big Data: (each question 5 points)

1. What are the main characteristics of Big Data?
   - Volume (the main characteristic that makes data "big" is the large volume), Variety (multimodal data that are structured and/or unstructured), Veracity (veracity refers to the trustworthiness of the data), Velocity (velocity is the frequency of incoming data that needs to be processed).

2. What is Mapreduce? How does it work? Name a few applications of Mapreduce.
   - MapReduce is a framework for processing parallelizable problems across large datasets using a large number of nodes, collectively referred to as a cluster or a grid. Processing can occur on data stored either in a filesystem (unstructured) or in a database (structured). A MapReduce system is usually composed of three operations (or steps): (1) Map: each worker node applies the map function to the local data, and writes the output to a temporary storage. A master node ensures that only one copy of redundant input data is processed. (2) Shuffle: worker nodes redistribute data based on the output keys, produced by the map function, such that all data belonging to one key is located on the same worker node. (3) Reduce: worker nodes process each group of output data, per key, in parallel. Mapreduce can be used in many distributed processing frameworks and applications such as Hadoop and search in big data.

3. What are the pros and cons of Hadoop?

   Pros:
   - Varied Data Sources (Hadoop accepts a variety of data
   - Cost-effective (Hadoop is an economical solution as it uses a cluster of commodity hardware to store data).
   - Speed and Performance (Hadoop with its distributed processing and distributed storage architecture processes huge amounts of data with high speed).
   - Fault-Tolerant (in Hadoop 3.0 fault tolerance is provided by erasure coding).
   - Highly Available (Hadoop 3.0 HDFS supports multiple standby NameNode making the system even more highly available as it can continue functioning in case if two or more NameNodes crashes).
   - Low Network Traffic (In Hadoop, each job submitted by the user is split into a number of independent sub-tasks and these sub-tasks are assigned to the data nodes thereby moving a

small amount of code to data rather than moving huge data to code which leads to low network traffic).

- High Throughput (Hadoop stores data in a distributed fashion which allows using distributed processing with ease. A given job gets divided into small jobs which work on chunks of data in parallel thereby giving high throughput).
- Open Source (Hadoop is an open source technology).
- Scalable (Hadoop works on the principle of horizontal scalability).
- Ease of use (Hadoop framework takes care of parallel processing, MapReduce programmers does not need to care for achieving distributed processing, it is done at the backend automatically).
- Compatibility (most of the emerging technology of Big Data is compatible with Hadoop like Spark, and Flink. We use Hadoop as data storage platforms for them).
- Multiple Languages Supported (developers can code using many languages on Hadoop like C, C++, Perl, Python, Ruby, and Groovy).

Cnos:

- Issue With Small Files (Hadoop is suitable for a small number of large files but when it comes to the application which deals with a large number of small files, Hadoop fails).
- Vulnerable By Nature (Hadoop is written in Java which is a widely used programming language hence it is easily exploited by cyber criminals).
- Processing Overhead (in Hadoop, the data is read from the disk and written to the disk which makes read/write operations very expensive).
- Supports Only Batch Processing (Hadoop has a batch processing engine which is not efficient in stream processing).
- Iterative Processing (Hadoop cannot do iterative processing by itself).
- Security (Hadoop uses Kerberos authentication which is hard to manage. It is also missing encryption at storage and network levels which are a major point of concern).

4. What is Spark? When and why Spark is preferred over Hadoop?
    - Spark and its resilient distributed dataset (RDD) were developed in response to limitations in the MapReduce cluster computing paradigm, which forces a particular linear dataflow structure on distributed programs: MapReduce programs read input data from disk, map a function across the data, reduce the results of the map, and store reduction results on disk. Spark's RDDs function as a working set for distributed programs that offers a restricted form of distributed shared memory (RAM). Spark facilitates the implementation of both iterative algorithms, and interactive/exploratory data analysis. The latency of such applications may be reduced by several orders of magnitude compared to Hadoop MapReduce implementation. Spark requires a cluster manager and a distributed storage system. For cluster management, Spark supports standalone (native Spark cluster), Hadoop YARN, or Apache Mesos. For distributed storage, Spark can interface with a wide variety, including Alluxio, Hadoop Distributed File System (HDFS), MapR File System (MapR-FS), Cassandra, OpenStack Swift, Amazon S3, Kudu, or a custom solution can be implemented. Spark has Less Latency so it is relatively faster than Hadoop, since it caches most of the input data in memory by the Resilient Distributed Dataset (RDD). Spark is 100 times faster than MapReduce as everything is done here in memory. In addition, Spark supports stream processing, which involves continuous input and output of data.