

Big data

Hadoop implementations



May 3, 2017

Enterprise transaction systems

Kurt Westerman

Big data is one of the leading challenges of the computing and business world today. It is defined by its volume (size,) velocity (rate of aggregation,) and variety (number of unique data types.) Depending on the company the needs of each of these categories differs creating a high demand for customization and optimization of databases to handle them. Many classical implementations that have driven the industry for decades are rapidly becoming obsolete due to the exponential growth of modern data. One of these pillars of batch processing is A.C.I.D transactions. Atomicity means that all instructions of a job must be executed correctly before they are committed or all of them must be rolled back. Consistency ensures every instruction is up to date and adheres to preprocessing and post processing protocols. Isolation implies each instruction is concurrently independent from the rest. Durability ensures once data is committed it will not be lost; this means data replication as well as nonvolatile memory to protect committed data from power loss and hardware failure.

Although ACID is still used, predominantly in situations where currency is involved, it is slow. Some of its requirements are sacrificed to handle data more rapidly, such as true consistency. In many systems eventual consistency is used when availability of some information is better than waiting for the entire job to finish before results are received. Basically Available Soft state Eventual consistency(BASE) is also optimal in massively multithreaded environments where speculative execution predicts multiple outcomes of various branches making use of cycles that would otherwise be stalls. A good example would be leading search engines such as Google, Yahoo, or EBSCOhost.

Application specific optimization is essential to various distributive storage architectures as they scale out. This gives rise to CAP theorem which suggests the three main areas of efficiency are consistency, availability, and partition tolerance. Furthermore, it states no more than 2 of these can be enforced at a time. Since partition tolerance is rarely unnecessary the desired focus of databases fall under two main categories, relational in the case of consistency(CP) or not only SQL(NoSQL) in the case of availability(AP.) Although some primarily consistency and availability focused systems(CA) do exist usually this is only guaranteed on the off-chance partitions are not being made.

One popular NoSQL database is Hadoop; It is a Java based open source framework from Apache that many third parties have been building around since its creation. At its core, it consists of two parts: a MapReduce program and the Hadoop distributive file system(HDFS.) MapReduce is a way of optimizing document searches based on keywords and the number of times they appear. During the map phase counting of the keywords is spread out across the cluster once all the counting, or mapping, is done the reducer combines all the individual keywords and sums up the counts producing the desired information from a vast amount of data. HDFS creates multiple copies of any give data set and spreads them across the cluster both for data durability and parallelism for the multiprocessing leverage of MapReduce. It can also handle files larger than individual nodes by partitioning data sets in blocks and striping them across various nodes. It seeks to bring the processing power to replicated data instead of sending data to processors.

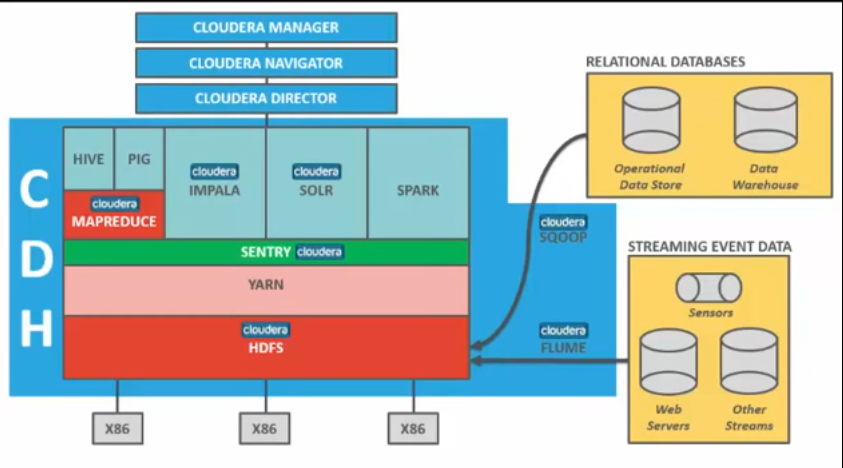
Databases today need to handle a large variety of data so projects like SQOOP and FLUME were augmented for HADOOP. FLUME logs streaming data into HDFS in real time from a multitude of sources like social media, sensors, and machines. This allows analysis of data on the fly as it is created. Also, it is hardware scalable to velocity and volume needs from cluster to cluster [1.] SQOOP is used for transferring bulk data quickly using widely parallel and load balanced busses often from relational databases or warehouses as more and more systems are converting to NoSQL. It utilizes schema on read to ensure the correct data when reading when in HDFS instead of the two-phase commit schema on write approach which can cause a speed bottleneck when transferring large volumes of data in high bandwidth transfers. SQOOP also compresses data for optimal storage to hardware ratios. The goal is to store as much data as fast and efficiently as possible regardless of data types or validity [2.]

With all the different data types being stored and replicated as well as search engines and algorithms consuming them a more general purpose batching approach called YARN was developed over HDFS to allow simultaneous real time access from to one or more data sets [3.] Data is valuable and commonly private so a layer of security needs to exist between HDFS and user clients requesting data permissions or authorization one such gatekeeping technology is Sentry from Cloudera [4.]

Many database users are more familiar with the simplicity of SQL based queries for batched workloads so Apache developed HIVE to translate SQL into java MapReduce programs. It is designed for speed, compatibility, and scalability over petabytes of data as is the paradigm for Hadoop [5.] Pig was developed for custom MapReduction and processing of even the most heterogenous data sets. The main feature, Pig Latin, is a language that feels natural for complex programs over large jobs. With Pig transformation can be performed over multiple sets even as it is streaming into a YARN based HDFS [6.]

Modern search engines require more than the counting of key value pairs which lead to Apache Spark for APIs compatible with multiple languages so similar systems or applications don’t have to reinvent the wheel as much. This is great for machine learning and data science in general where algorithms are often iterative environments especially since it allows data to be more efficiently cached in memory for quicker access and evolution. Most of Apache’s software is open source so it makes public collaboration and construction of APIs simple and speeds up the progression of knowledge [7.] The internet of things creates a high demand for public online search capabilities; Apache’s answer to this problem is SOLR. Using HTTP’s get function all consumers can acquire XML, JSON, CSV, or binary results from a Hadoop cluster anywhere around the world in fractions of a second easily with a Google search. This is done through almost real time indexing. Not only is it scalable and fast it is fault tolerant; when a request to a given server doesn’t meet time constraints due to either high traffic or hardware failure it fails over to another server or cluster. Jobs are recursively divided in parallel to available producers to save as much time as possible. SOLR also collects statistics to better balance loads around the web as traffic and data increase exponentially [8.]

Another leading open source developer of Hadoop oriented software is Cloudera, the developer of Impala, it focusses on Massively Parallel Processing(MPP) for business intelligence to reduce latency of immense jobs from hours to mere seconds. It works through 3 daemons across HDFS. The first is Impalad which exists on every node in the cluster to read and write files; the coordinator node is the node the job was launched on it manages job distribution and collection across other Impalad nodes. The next daemon is statestored which only exists on one node it keeps heartbeats of all the nodes in the cluster to determine if a node has failed or otherwise been disconnected in which case all nodes are notified and partial data loss is rerouted. In the case of statestored node failure all nodes assume all other nodes are functional leading to a possible minor performance decrease if assumptions are wrong. The final daemon is catalogd which also exists on a single node optimally the same as statestored since these processes communicate with each other frequently. Its function is ensuring coherence and continuity; when an Impalad writes all other nodes sharing that data need to be invalidated and the modified value should then be distributed where it is requested or replicated. Impala allows large jobs with little to no critical regions to be executed almost or entirely in parallel according to the query and execution plans in the time of several to a few hundred cycles opposed to billions or trillions if it were managed less efficiently or even sequentially [9.]



**[10] Diagram of Cloudera CDH package**

Cloudera’s CDH package for Hadoop contains all the previously mentioned Hadoop related platforms, technologies, and frameworks and can be visualized in the diagram above [10.] This package comes with several system wrappers. The manager is essential for getting the cluster up and running, optimizing various system requirements, can run diagnostics, and allows direct communication with Cloudera support professionals. The Navigator attempts to make an accurate picture of the dynamic and rapidly changing file system, log changes, as well as encrypts this information externally as an added level of security. The director allows the creation of databases in the cloud so you can just rent the hardware as you are using it instead of buying and constructing a warehouse of your own.

In conclusion, the industry has many approaches to big data. Hadoop is just one of many all with their respective strengths and weaknesses. For example, HDFS has a single point of failure in the name node which is a directory of all files in the system. If this node were to fail the entire cluster would go down until it is fixed because it would be impossible to locate data without a schema. Although solutions to all these problems simultaneously is being emerge in NewSQL systems they are still maturing.

**Resources**

[1] HortonWorks  *Apache Flume* available online: hortonworks.com/apache/flume/

[2] HortonWorks *Apache SQOOP* available online: hortonworks.com/apache/sqoop/

[3] HortonWorks *Apache Hadoop YARN* available online: hortonworks.com/apache/yarn/

[4] Cloudera *Apache Sentry* available online: www.cloudera.com/products/open-source/apache-hadoop/apache-sentry.html

[5] HortonWorks *Apache Hive* available online: hortonworks.com/apache/hive/

[6] HortonWorks *Apache Pig* available online: hortonworks.com/apache/pig/

[7] HortonWorks *Apache Spark* available online: hortonworks.com/apache/spark/

[8] HortonWorks *Apache Solr* available online: hortonworks.com/apache/solr/

[9] DeZyre (2017) *Fundamentals of Impala* available online https://www.dezyre.com/hadoop-tutorial/hadoop-impala-tutorial

[10] Cloudera, Inc. (May 27, 2015) *Apache Hadoop & Big Data 101: The Basics* available online: youtube.com