www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 8 August, 2014 Page No. 7917-1920

Hadoop – from where the journey of unstructured data begins…

Mrs. R. K. Dixit, Sourabh Mahajan, Suyog Kandi.

Department of Computer Science and Engineering
Walchand Institute of Technology, Solapur.
rashmirajivk@gmail.com

Department of Computer Science and Engineering
Walchand Institute of Technology, Solapur.
ssmahajan92@gmail.com

Department of Computer Science and Engineering
Walchand Institute of Technology, Solapur.
kandisuyog92@gmail.com

Abstract:***Hadoop is a "flexible and available architecture for large scale computation and data processing on a network of commodity hardware"it is processingtechnology for large scale applications. Nowadays unstructured data is growing at faster rate. Big data generated daily from social media, sensor used to gather climate information, digital picture purchased transactions records and many more. Ultimately there need to process Multi Petabyte Datasets efficiently. Failure in datasets is expected, rather than exceptional. The number of nodes in cluster is not constant. The Hadoop platform was designed to solve such problems as it provides application for operational and analytics. Today Hadoop is fastest growing technology provides advantage for business across industries in world.***

Keywords: HDFS:Hadoop Distributed File System, PB: petabytes**.**

1. Introduction

Scenario of Computing in its purest form has changed multiple time compare to its earlier time. As uses and need of computing changed widely as per technology. First, from near the beginning mainframes were predicted to be the future of computing. Indeed mainframes and large scale machines were built and used, and in some circumstances are used similarly today. The today's trend turned from bigger and more expensive, to smaller and more affordable commodity PCs and servers. Hadoop is a strong tool which is open-source platform manages to develop and analyze large and massive datasets. Hadoop enables you to explore complex data, using custom analyses tailored to your information and questions. In Hadoop system unstructured data is distributed across hundreds or thousands of machines forming clusters, and the execution of Map/Reduce routines to run on the data in that cluster. Hadoop has its own file system which replicates data to multiple nodes to ensure if one node holding data goes down, there are at least 2 other nodes from which to retrieve that piece of information.

2. History

Hadoop was created by Doug Cutting at Yahoo!, who named Hadoop after his child stuffed elephant. Hadoop was inspired by Map Reduce a user defined function developed by google in early 2000s for indexing the Web. It was designed to handle petabytes andExabyte's of data distributed over multiple nodes in parallel further in 2006 yahoo hires Doug Cutting to work on Hadoop with a dedicated team which became top level project of apache in 2008.Hadoop is used by the top level apache foundation project, large active user base, mailing lists, users groups, very active development, and strong development teams. Today Hadoop has its own eco-system and versions which has lot of development.

3. Assumptions and Goals

Assumption and goals of Hadoop are very clear node failure i.e. hardware failure, streaming datasets, large datasets, distributed file system.A distributed file system is designed to hold a large amount of data and provide access to this data to many clients distributed across a network. An HDFS instancemay consist of hundreds or thousands of server machines, eachstoring part of the file system's data.The main and basic feature of HDFS is a distributed file system designed to run on commodity hardware. The fact that there are a hugenumber of components and that each component has a non-trivialprobability of failure means that some component of HDFS is alwaysnon-functional. Therefore, detection of faults and quick, automaticrecovery from them is a core architectural goal of HDFS.HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. Applicationsthat run on HDFS need streaming access to their data sets. They arenot general purpose applications that typically run on general

purposefile systems. HDFS is designed more for batch processing rather thaninteractive use by users. The emphasis is on high throughput of data access rather than low latency of data access. POSIX imposes many hard requirements that are not needed for applications that are targeted forHDFS. POSIX semantics in a few key areas has been traded to increasedata throughput rates.Applications that run on HDFS have large data sets. A typical file inHDFS is gigabytes to terabytes in size. Thus, HDFS is tuned to supportlarge files. It should provide high aggregate data bandwidth andscale to hundreds of nodes in a single cluster. It should support tens ofmillions of files in a single instance. HDFS applications need a write once-read-many access model for files. A file once created, written, andclosed need not be changed. A Map Reduce application or a web crawler application fits perfectly with this model. There is a plan to support appending-writes to files in the future. A computation requested by an application is much more efficient if itis executed near the data it operates on. This is especially true when the size of the data set is huge. This minimizes network congestion andincreases the overall throughput of the system. The assumption is that it is often better to migrate the computation closer to where the data is located rather than moving the data to where the application is running. HDFS provides interfaces for applications to move themselves closer to where the data is located. HDFS has been designed to be easily portable from one platform to another. This facilitates widespread adoption of HDFS as a platform of choice for a large set of applications.HDFS has Master/slave architecture. An HDFS cluster consists of a single Name node, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of Data nodes, usually one per node in the cluster, which manages storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of Data nodes. The Name node executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to Data nodes. The Data nodes are responsible for serving read and write requests from the file system's clients. The Data nodes also perform block creation, deletion, and replication upon instruction from the Name node. Some of the goals of HDFS are Very Large Distributed File System, Assumes Commodity Hardware and Optimized for Batch Processing, Runs on heterogeneous OS

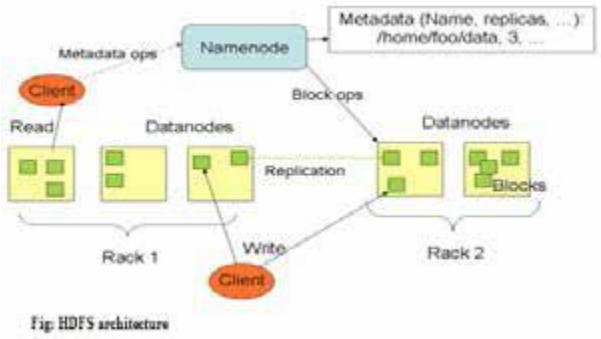

Fig: HDFS architecture

Figure1: HDFS architecture

4. Hadoop's components

Hadoop Distributed File System: HDFS, the storage layer of Hadoop, is a distributed, scalable, Java-based file system adept at storing large volumes of unstructured data. Map Reduce is a software framework that serves as the compute layer of Hadoop. Map Reduce jobs are divided into two (obviously named) parts. The "Map" function divides a query into multiple parts and processes data at the node level. The "Reduce" function aggregates the results of the "Map" function to determine the "answer" to the query. Hive is a Hadoop-based data warehousing-like framework originally developed by Facebook. It allows users to write queries in a SQL-like language called HiveQL, which are then converted to Map Reduce. This allows SQL programmers with no Map Reduce experience to use the warehouse and makes it easier to integrate with business intelligence and visualization tools such as Micro strategy, Tableau, Revolutions Analytics, etc. Pig Latin is a Hadoop-based language developed by Yahoo. It is relatively easy to learn and is adept at very deep, very long data pipelines (a limitation of SQL.)

GRA - GLOBAL RESEARCH ANALYSIS X 101 HBase is a non-relational database that allows for low-latency, quicklookups in Hadoop. It adds transactional capabilities to Hadoop, allowingusers to conduct updates, inserts and deletes. EBay and Facebookuse HBase heavily. Flume is a framework for populating Hadoop withdata. Oozie is a workflow processing system that lets users define a series of jobs written in multiple languages – such as Map Reduce, Pig and Hive -- then intelligently link them to one another. Oozie allows users to specify, for example, that a particular query is only to be initiated after specified previous jobs on which it relies for data are completed. Flume is a framework for populating Hadoop with data. Ambari is a web-based set of tools for deploying, administering and monitoring Apache Hadoop clusters. Its development is being led by engineersfrom Hortonworoks, which include Ambari in its Horton works Data Platform. Avro is a data serialization system that allows for encoding the schema of Hadoop files. It is adept at parsing data and performing removed procedure calls. Mahout is a data mining library. It takes the most popular data mining algorithms for performing clustering, regression testing and statistical modeling and implements them using the Map Reduce model. Sqoop is a connectivity tool for moving data from non-Hadoop data stores – such as relational databases and data warehouses – into Hadoop. HCatalog is a centralized metadata management and sharing service for Apache Hadoop. Big Top is an effort to create a more formal process or framework for packaging and interoperability testing of Hadoop's sub-projects and related components with the goal improving the Hadoop platform as a whole.

Figure 2: Hadoop Ecosystem

5. Working process of Hadoop architecture

Hadoop is designed to run on a large number of machines that don't share any memory or disks. That means you can buy a whole bunch of commodity servers, slap them in a rack, and run the Hadoop software on each one. When you want to load all of your organization's data into Hadoop, what the software does is bust that data into pieces that it then spreads across your different servers. There's no one place where you go to talk to all of your data; Hadoop keeps track of where the data resides. And because there are multiple copy stores, data stored on a server that goes offline or dies can be automatically replicated from a known good copy. In a centralized database system, you've got one big disk connected to four or eight or 16 big processors. But that is as much horsepower as you can bring to bear. In a Hadoop cluster, every one of those servers has two or four or eight CPUs. You can run your indexing job by sending your code to each of the dozens of servers in your cluster, and each server operates on its own little piece of the data. Results are then delivered back to you in a unified whole. That's Map Reduce you map the operation out to all of those servers and then you reduce the results back into a single result set. Architecturally, the reason you're able to deal with lots of data is because Hadoop spreads it out. And the reason you're able to ask complicated computational questions is because you've got all of these processors, working in parallel, harnessed together.Hadoop implements a computational paradigm named Map/Reduce, where the application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster. In addition, it provides a distributed file system (HDFS) that stores data on the compute nodes, providing very high aggregate bandwidth across the cluster. Both Map/Reduce and the distributed file system are designed so that node failures are automatically handled by the framework. Hadoop Common is a set of utilities that support the other Hadoop subprojects. Hadoop Common includes File System, RPC, and serialization libraries.

6. Map Reduce

Map Reduce is the heart of Hadoop .It is this programming paradigm that allows for massive scalability across hundreds or thousands of servers in a Hadoop cluster. The Map reduce concept is fairly simple to understand for those who are familiar with clustered scale-out data processing solutions. For people new to this topic, it can be somewhat difficult to grasp, because it's not typically something people have been exposed to previously. The term Map reduces actually refers to two separate and distinct tasks that Hadoop programs perform. The first is the map job, which takes a set of data and converts it into another set of data, where individuals elements are broken down into tuples (key/value pairs).The reduce job takes the output from a map as input and combines those data tuples into a smaller set of tuples. As the sequence of the name Map Reduce implies, the reduce job is always performed after the map job.

7. Requirement of Hadoop

Batch data processing, not real-time user facing (e.g. Document Analysis and Indexing, Web Graphs and Crawling).Highly parallel data intensive distributed applications .Very large production deployments (GRID) Process lots of unstructured data .When your processing can easily be made parallel. Running batch jobs is acceptable when you have access to lots of cheap hardware. Hadoop Users: The following companies are the users of Hadoop Adobe,Alibaba, Amazon, AOL, Facebook, Google, and IBM.

8. Conclusion

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. Hadoop is designed to run on cheap commodity hardware, it automatically handles data replication and node failure, it does the hard work – you can focus on processing data, Cost Saving and efficient and reliable data processing. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. HDFS relaxes a few POSIX requirements to enable streaming access to file system data.
HDFS was originally built as infrastructure for the Apache Nutch web search engine project. HDFS is part of the Apache Hadoop Core project.

Reference

[1]Apache Hadoop!(hadoop.apache.org)
[2] Mr. Nileshpatil, Mr. Tanvirpatel, "Apache Hadoop: Resource Big Data Management" International Journal of Innovative Research in Science, Engineering and Technology. ISSN:2319-8753. www.ijirset.com
[3]Hadoop on Wikipedia (http://en.wikipedia.org/wiki/Hadoop)
[4] Praveen kumar, Dr Vijay Singh Rathore, "Efficient Capabilities of Processing of Big Data using Hadoop Map Reduce" International Journal of Advanced Research in Computer and Communication Engineering. ISSN:227-1021
[5] Free Search by Doug Cutting (http://cutting.wordpress.com)
[6]Hadoop and Distributed Computing at Yahoo!(http://developer.yahoo.com/hadoop) |
Yahoo! Inc , Hadoop tutorial from Yahoo! Available : http//:developer.yahoo.com/Hadoop/tutorial/index.html
[7]Cloudera -Apache Hadoop for the Enterprise (http://www.cloudera.com)
[8]www.pentaho.com

Author Profile

Mrs. R. K. Dixitis an Associate Professor in Computer Science and Engineering Department at Walchand Institute of Technology, Solapur. She received his B.E degree from Shivaji University, Kolhapur and M.E degree from Pune University Pune.Her research interests lie in the area of Security, Database, Big Data, Hadoop and Data Mining.

Sourabh SMahajan is a Graduation student pursuing B.E in Computer Science and Engineering from Walchand Institute of Technology, Solapur. His research interests lie in the area of BigData, Hadoop and Data Mining

Suyog D Kandi is a Graduation student pursuing B.E in Computer Science and Engineering from Walchand Institute of Technology, Solapur. His research interests lie in the area of Big Data, Hadoop and Data Mining.