International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 02 Issue: 03 | June-2015 www.irjet.net p-ISSN: 2395-0072

AN INTRODUCTION TO MAP REDUCE APPROACH TO DISTRIBUTE WORK USING NEW SET OF TOOLS

Mr. Narahari Narasimhaiah[1], Dr. R. Praveen Sam[2].

[1]*Research scholar, Bharathiar University, Tamilnadu, India.*
[2]*Professor, Dept. of CSE. Andhra Pradesh, India.*

ABSTRACTION- *Using a specialized query language on highly structured and optimiazed data structures, all processing happens after the information has been loaded into the stores to the traditional relational database[3] world. Across many machines with the computation spread, with intermediate results being passed between stages as files. Instead create a pipeline that reads and writes to arbitrary file formats, this is the Googleapproach, and adopted by many web companies. Typically based around the Map- Reduce[1] approach to distributing work, this approach requires a whole new set* ***of tools, which I'll describe belo****w.*

Key Words: Relational, Distributing, Map Reduce etc…

1. INTRODUCTION

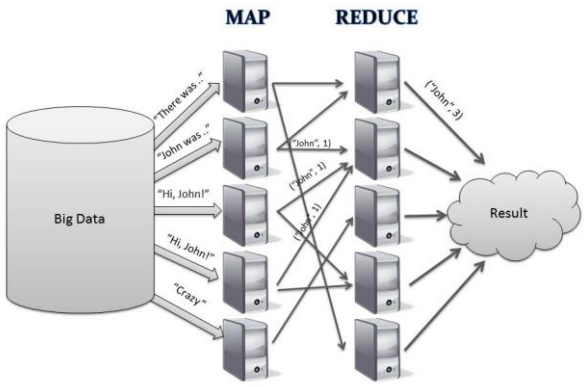

Fig - 1:Mappers and reducers functionality.

Figure 1 is showing, the map and reduce combinatory concept in functional programming languages such as Lisp of the Map Reduce model. In Lisp, a *map* takes as a sequence of values and input a function. It applies the function to each value in the sequence. All the elements of a sequence using a binary operation combines in*reduce*. In 2004 Google introduced the Map Reduce framework to support distributedprocessing on large data sets distributed over clusters of computers. Currently it is an integral[3] part of the Hadoop ecosystem, but it was implemented by many software platforms.

Map Reduce was introduced and specifically designed to run on commodity hardware and to solve large-data computational problems. The input data sets, based on *divide-and-conquer[4]* principles, these are split into independent chunks, which are processed by the mappers in parallel.

Fig -2:Map Reduce Process.

Based on the user-supplied code, to provide the overall coordination of execution is the responsibility of the Map Reduce framework. This includes choosing appropriate machines (nodes) for running mappers; choosing appropriate locations for the reducer's execution; starting and monitoring[2] the mappers execution; sorting[3] and shuffling output of mappers and delivering the output to reducer nodes; and starting and monitoring the reducer's execution.

2. FUNCTIONAL PROGRAMMINGCONCEPTS

MapReduce programs are designedin a parallel fashion to compute large volumes of data. Across a large number of machines, this requires dividing the workload. If the components[1] were allowed to share data arbitrarily, this model would not scale to large clusters (hundreds or

thousands of nodes). To keep the data on the nodes synchronized at all times would prevent the system from performing reliably or efficiently at large scale required for the communication overhead.
MapReduce are immutable for all data elements, meaning that they cannot be updated. It does not get reflected back in the input files, if in a mapping task you change an input (key, value) pair; communication occurs only by generating new output (key, value) pairs which are then forwarded by the Hadoop system into the next phase of execution.

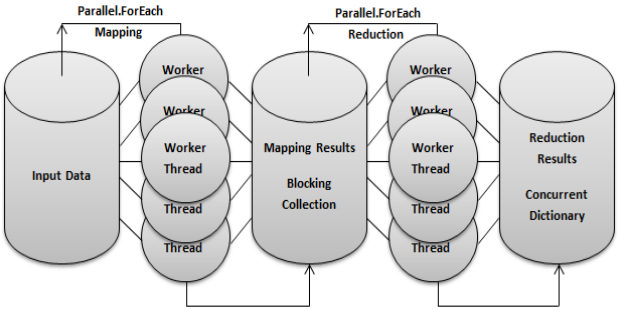

Fig -3: Parallel Map Reduce Process

List Processing: Lists of input data elements into lists of output data elementsare transformed by the Map Reduce[1] programs. Using two different list processing idiomsmap, andreduce; a Map Reduce program will do this twice.
Mapping Lists:Mapping is the first phase of a MapReduce algorithm. An output data element, a list of data elements are provided, one at a time, to a function called the Mapper, which transforms each element individually.
Reducing Lists: Lets you reduce aggregate values together. An iterator of input values from an input list receives a reducer function[2]. Returning a single output value it combines these values together.
To produce "summary" data, turning a large volume of data into a smaller summary of itself, reducing is used. For example, to return the sum of a list of input values, "+" can be used as a reducing function.

3. MAP REDUCING TOOLS

Large volumes of data in parallel by dividing the work into a set of independent tasks is a Map Reduce programming model designed for processing. MapReduce programs are written in a specifically idioms[5] for processing lists of data and particular style influenced by functional programming constructs.
Hadoop:Google's Map Reduce infrastructure, originally developed by Yahoo! as a clone, but subsequently open sourced, Hadoop takes care of running your code across a cluster[2] of machines. Its responsibilities include sending it to each machine,chunking up the input data, checking that the code ran, your code on each chunk, passing any results either on to further processing stages or to the final output location, writing debugging[6] information on each job's progress, and performing the sort that occurs between the map and reduce stages and sending each chunk of that sorted data to the right machine, among other things.
Hive: Hive can program Hadoop jobs using SQL.
Hadoop's focus on the latency may mean that even simple jobs take minutes to complete and large scale processing so it's not a substitute for a real-time transactional database.
Pig: The Apache Pig is a procedural data processing language designed for Hadoop, a series of steps to perform on the data and closer to scripting language, but with a specialized set of functions that helps with common data processing[2] problems.
Cascading:Building explicitly out of Map- Reduce steps feeding into one another, call a Java API, connecting objectsthat represent the operations you want to perform into a graph. The system takes that definition, does some checking and planning, and executes it on your Hadoop cluster. There are a lot of built-in objects for common operations like grouping, sorting, and joining as well as write your own objects to run custom processing code.
Cascalog:Cascalog is a functional data processing interface written for Closure. Influenced by the old Datalog language and built on top of the Cascading framework and processing code at a high level of abstraction while the system takes care of assembling it into a Hadoop job.
Mrjob:Mrjob is a framework to write the code for transparently run it either locally, on Elastic Map Reduce, or on your own Hadoop cluster[5] or processing the data.
Caffeine: From reports and company comments, Map Reduce paradigm appears that Google is using a new version of the Google File System that supports smaller files and distributed masters. The company has moved away from the batch processing approach to building its search index, instead using a dynamic database approach to make updating faster.
S4: Initially Yahoo! hascreated the S4 system to make decisions about choosing and positioning ads, but the company open sourced it after finding it useful for processing arbitrary streams of events. Using the Zoo Keeper framework,S4is to handle unbounded streams of events, and runs it distributed across a cluster of machines to handle the housekeeping details.
MapR:MapR is a Hadoop commercial distribution aimed at enterprises. MapR includes its own file systems that are a replacement for HDFS, along with other tweaks to the framework, like distributed name nodes for improved reliability.
Oozie:Oozie supports a more complex language for describing job flows, allowing you to make runtime decisions

about exactly which steps to perform, all described in XML files. There's also an API that you can use to build your own extensions to the system's functionality.
Greenplum: TheGreenplum system offers an interesting way of combining a flexible query language with distributed performance for NoSQL databases[3]. The Postgres open source database, it adds in a distributed architecture to run on a cluster of multiple machines, while retaining the standard SQL interface.

4. CONCLUSIONS

At the heart of the Hadoop system,this module described as the MapReduce execution platform exists. A high degree of parallelism can be achieved by applications using by Map Reduce. The MapReduce framework provides a high degree of fault tolerance for applications running on it by limiting the communication which can occur between requiring applications and nodes to be written in a "dataflow-centric" manner.

REFERENCES:

[1]. Map Reduce Tutorial - http://mapreduce-tutorial.blogspot.de/2011/04/mapreduce-basics.html
[2]. MapResude Design Patterns - O'Reillt - By Donald Miner & Adam Shook.
[3]. Azure MapReduce - by ThilinaGunarathne and Salsa group, Indiana Unviersity.
[4]. MapReduce Views in Couch DB - O'Reilly - By Bradley Holt.
[5]. BigData for Dummies - by Judith Hurwitz, Alan Nugent, Dr.FernHalper and Marcia Kaufman.
[6]. Hadoop MapReduce Cookbook - by SrinathPerera and ThilinaGunarathne.

BIOGRAPHIES

NARAHARINARASIMHAIAH has 17 years of experience in IT industry. At Cognizant Technology Solutions, he is a Principal Architect-Technology with BFS TAO (Technology and Architecture Office). He holds a MBA, MS in Computer Science degree and MSc in Psychology. Currently pursuing PhD in Cloud Computing.

He is a TOGAF 8, PMI, Sun Certified Java Enterprise Architect, SCJP, MCP.NET, Six Sigma and Certified ScrumMaster as well.

He has experience/knowledge/awareness in Micro Services, Big Data Analytics, Cloud Computing, Mobile Payments, Mobile Development (iOS, Android), Responsive Web/Single Page applications, Solutions Architecture, Service Oriented Architecture, Portfolio Assessments, Technology Rationalization / Consolidation exercises.

Highly astute, goal-oriented and result oriented professional offering a track record as a Technology Partner for a major financial organization. Proven success and experience in managing diverse large teams globally spanning multiple vendors with an on-site and offshore mix. Professional experience in startup as founder/co-founder and CEO organizations. Advising committee member/director in various Non-IT organization(s).

RACHAPUDI PRAVEEN SAM was born in Kurnool City in 1975. He received the B.Tech degree in Computer Science and Engineering with First Class in 1999 from Sri Krishna Devaraya University, Ananthapur, A.P., India; M.Tech degree Computer Science and Engineering with First Class in 2001 from Madras University, Chennai, T.N., India and was awarded Ph.D. degree in Computer Science and Engineering in 2010 from JNTU University, Ananthapur, A.P., India. His Ph.D. specialization is mobile and Ad Hoc Networks(MANETS). He expertise in Computer Networks and Network Security.

He is having 13 years of teaching experience, presently he is working as a professor of Computer Science and Engineering department for G.Pulla Reddy Engineering College (Autonomous), Kurnool City, India. He has a total of 25 publications out of which 13 papers in International and National Journals and 12 papers in National and International Conferences. He is a member of various professional bodies like ISTE, IE, CSI, IAENG, CSTA, and IACSIT.

He received Minor Research Project titled "Developing Disaster Management Applications using Mobile Ad Hoc Network Tested" sanctioned by UGC for a period of 2 years in March 2014.

