ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

Proactive Usage of Slot for Memory Allocation in Hadoop for Government Managed System

Prof. U.M Kalshetti[1], Rohit Bhattacharya[2], Ajit Maid[2], Swapnil Ramteke[2], Nilesh Gaikwad[2]

Professor, Department of Computer Engineering and Information Technology, PVG's College of Engineering and Technology, Pune, Maharashtra, India[1]

Department of Computer Engineering and Information Technology, PVG's College of Engineering and Technology, Affiliated to Savitribai Phule Pune University, Pune, Maharashtra, India[2]

ABSTRACT: Today India is looking forward to take a leap in digital world that is efficient, user friendly and solves problems face by common man. However the government systems at present in a very molecular level still function traditionally and are faced with esoteric problems like corruption, usage of resources and efficient handling of resources .Our paper proposes to solve these problems by bridging the gap of present system with the help of GMSUPH and Proactive Hadoop. Our paper proposes a system to upload documents on an application that can be accessed from any location by the concerned officer which was earlier just documented in files and kept in large warehouses. This can be further used by government for statistical , analysis or verification purposes. We shall do this by making use of our modules over Hadoop .Since Map Reduce is widely used to handle large amount of data, our system makes use of map reduce operation using Hadoop Hadoop is an open source implementation of MapReduce. However performance metrics received from Hadoop suggest it is inconsistent and has a scope for improvement. Our paper proposes modules like PHSA , AEPB and Slot Prescheduling through which we will be able to make use of all map slots for reduce operation and all reduce slots for map operations which was earlier not possible. Our modules will also make maximum utilization of memory allocated for map and reduce slots. Our paper proposes these modules as a layer that sits above the present framework and aims at improving performance of present Hadoop framework. Our paper not only aims at making Hadoop more powerful and efficient but also build an application on it that can be used by our government and help our society in larger sense.

KEYWORDS: Proactive, AEPB, PHSA, Slot Prescheduling, MapReduce, Task Granularity, Yarn.

I. INTRODUCTION

Our paper proposes a system that will digitalize the data stored by Government at a very molecular level by using Hadoop as a back-end distributed system. MapReduce was developed within Google to handle large amount of data efficiently without compromising on performance and resources required. MapReduce eventually became an answer to handle big data which was earlier a complex problem . MapReduce consist of a mapping function which processes given data to intermediate form of tuples. While Reduce function processes values related to a specific key.
Let us consider an example of data defining fruits in a mall. Here, Map function will classify different fruits uniquely e.g All apples in one cluster, all bananas in another and so on. Reduce function will associate a key identifying all apples in a cluster and all bananas in another cluster. This helps in improving the performance of system by compressing over all data.

Apache Software Foundation developed Hadoop as open–source implementation of MapReduce. However after putting many efforts on the development of the software not much work was done on performance metrics of Hadoop. Though Hadoop is an answer to process large amount of data on a distributed framework , it is still unpredictable over giving a reliable performance in different scenarios. Some of the limitations of Hadoop is listed below

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

Sr. No.	Name	Description
1.	Job Execution[4]	In present system map slots can only be used map tasks and reduce slots can only be used reduce task .
2.	Inconsistency[3]	Inconsistent performance under different slot configuration .
3.	Idle[2]	Idle map reduce slot affecting system utilization and performance .
4.	Run time Contention[3]	They can be straggled map/reduce task causing significant delay .

Our paper proposes to implement Proactive Hadoop which will be able to solve the mentioned limitations. Since Hadoop is static in nature, we propose to make usage of slot allocation proactively so that Hadoop will make maximum utilization of its slot without wasting any memory.

II. RELATED WORK

As far as government administration systems are concerned, earlier data was completely stored in documentation form. For every manipulation of data, assigned officer had to physically make changes. However in last few years, Government has adapted the use of computer and digital administration systems, data now is stored in digital form and manipulated with the help of front-end graphical user interface.
Hadoop as stated was static in nature. There is research going on to improve the performance of Hadoop . For example, Yahoo is working with open source communities to bring MapReduce and more applications on YARN[5]. This will empower Spark applications onto existing Hadoop hardware One of the paper, called "PUMA[1]" which represents a broad range of MapReduce applications exhibiting application characteristics with high/low computation and high/low shuffle volumes proposed. There are a total of 13 benchmarks, out of which Tera-Sort[1], Word-Count[1], and Grep[1] are from Hadoop distribution. The rest of the benchmarks were developed in-house and are currently not part of the Hadoop distribution . The three benchmarks from Hadoop distribution are also slightly modified to take number of reduce tasks as input from the user as well as to generate final time completion statistics of jobs. We have used this benchmarks in our system

While a paper proposed a system called "Mantri" that delivers e-active mitigation of outliers in Map-Reduce networks. It is motivated by, what we believe is, the study of a large production Map-Reduce cluster. The root of Mantri's advantage lies in integrating static knowledge of job structure and dynamically available progress reports into a unified framework that identifies outliers early, applies cause-specific mitigation and does so only if the benefit is higher than the cost. In our implementation on a cluster of thousands of servers, we think "Mantri[4]" to be highly effective.
While handling data on large clusters , study shows that restricting the programming model makes it easy to parallelize and distribute computations and to make such computations fault-tolerant. As well as, network bandwidth is a scarce resource. A number of optimizations in our system are therefore targeted at reducing the amount of data sent across the network : the locality[3] optimization allows us to read data from local disks , and writing a single copy of the intermediate data to local disk saves network bandwidth. Thus, redundant execution can be used to reduce the impact of slow machines, and to handle machine failures and data loss.

III. PROPOSED SYSTEM

As discussed earlier , we aim to digitalize traditionally stored data by creating an application "Government managed system using Proactive" .

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

GOVERNMENT MANAGED SYSTEM USING PROACTIVE HADOOP

A)Present Scenario

A Government monitors many states, districts, talukas and villages . These regions consist of different statistical data based on the residents . As this data is in large volumes, today most of that data is documented in traditional ways i.e documented in files. This results into human errors, corruption, mishandling of data and requires infrastructure and money. Today Government manually enters data which is further stored in a server , however there is no way for an officer to access his taluka's data from another computer present in some other location .
Disadvantages of Present System :

- Scope of corruption .
- Lack of transparency .
- Difficulty in handling data .
- Ineffective use of resource

B) GMSUPH

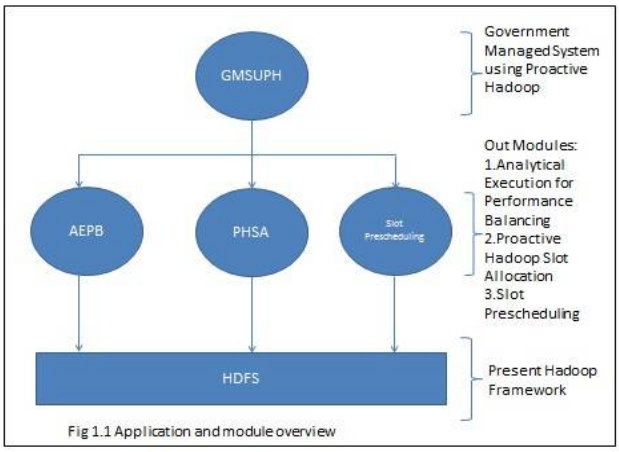

Fig 1.1 Application and module overview

To solve the constraints of present system we will be using multinode configuration in Hadoop . Our paper proposes at bringing this large amount of data on the web using Hadoop. Government managed slot utilization for proactive hadoop(GMSUPH) , is a practical application of PSAMAH which will have feature to submit document like address proof required by the Government . These documents will then be used by Concerning officer to verify the details of residents in future . This will be done by using above mentioned modules i.e ACPB,PHSA and Slot Prescheduling over present Hadoop framework as shown in fig1.2. This will help in improving the performance of our Hadoop and make effective use of system resources. Our paper proposes to digitalize data of government that was so far stored traditionally . Our application will be able to overcome the difficulties faced by the common man as well as the government. This will improve accountability and efficiency of government applications, bring transparency in government transactions, make people aware of advantages of Computer and Internet and bridge the gap between technology and rural To solve the constraints of present system we will be using multinode configuration in Hadoop . Our paper proposes at bringing this large amount of data on the web using Hadoop. Government managed slot utilization for proactive hadoop(GMSUPH) , is a practical application of PSAMAH which will have feature to submit document like address proof required by the Government . These documents will then be used by Concerning officer to verify the details of residents in future . This will be done by using above mentioned modules i.e ACPB,PHSA and Slot Prescheduling over present Hadoop framework as shown in fig1.2. This will help in improving the performance of our Hadoop and make effective use of system resources. Our paper proposes to digitalize data of government that was so far stored traditionally . Our application will be able to overcome the difficulties faced by the common man as well as the government. This will improve accountability and efficiency of government applications, bring transparency in

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

government transactions, make people aware of advantages of Computer and Internet and bridge the gap between technology and rural society.

C) Advantages of GMSUPH :

- Improves the performance of MapReduce workloads write maintaining the fairness.
- Can be used for any kinds of MapReduce jobs (independent or dependent ones).
- Balances the performance trade-off between a single job & a batch of jobs dynamically.
- Slot pre-scheduling improves the efficiency of slot utilization by further maximizing its data locality.
- AEPB identify the slot inefficiency problem of speculative execution.
- Proactive MR consistently outperforms YARN.

Proactive Hadoop modules that are implemented over present Hadoop framework.

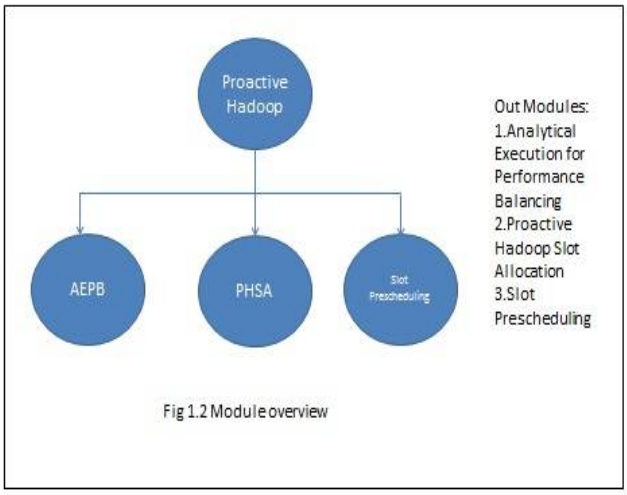

Fig 1.2 Module overview

D) Proactive Hadoop Slot Allocation (PHSA)

Current configuration of MapReduce experiences an under-use of the seperate openings as the quantity of guide and lessen assignments shifts after some time, bringing about events where the quantity of spaces distributed for guide/diminish is littler than the quantity of guide/decrease undertakings. Our dynamic space allotment arrangement depends on the perception[2] that at various timeframe there might be sit without moving guide (or diminish) openings, as the employment continues from guide stage to decrease stage. We can utilize the unused guide spaces for those over-burden diminish errands to enhance the execution of the MapReduce workload, and the other way around. For instance, toward the start of MapReduce workload calculation, there will be just processing map undertakings and no figuring decrease assignments, i.e., all the calculation workload lies in the guide side. All things considered, we can make utilization of unmoving decrease openings for running guide errands. That is, we break the verifiable supposition for current MapReduce framework that the map tasks can only run on map spaces and lessen undertakings can just keep running on decrease openings. Rather, we alter it as takes after both guide and lessen undertakings can be keep running on either outline decrease spaces. However, there are few challenges that should be considered as follows:

- Fairness:

Fairness is a critical metric in Hadoop Fair Scheduler (HFS). We say it is reasonable when the sum total of what pools have been distributed with the same measure of assets. In HFS, assignment openings are first apportioned over the pools, and afterward the spaces are dispensed to the occupations inside of the pool . In addition, a MapReduce work calculation comprises of two sections: map-stage errand calculation and lessen stage undertaking calculation.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

- Resource requirements :

The asset prerequisites between the guide space and lessen opening are for the most part diverse. This is on the grounds that the guide assignment and lessen undertaking regularly show totally distinctive execution designs. Diminish assignment has a tendency to devour a great deal more assets, for example, memory and system data transfer capacity. Essentially permitting diminish assignments to utilize map spaces requires configuring every guide opening to take more assets, which will subsequently lessen the compelling number of spaces on every hub, bringing on asset under-used amid runtime.

E) Analytical Execution Performance Balancing (AEPB)

MapReduce occupation's execution time is extremely touchy to slow running assignments (to be specific straggler) . There are different reasons that cause stragglers, including defective equipment and programming mis-configuration . We arrange the stragglers into two sorts, in particular, Hard Straggler and Soft Straggler, defined as follows :

- Hard Straggler

An errand that goes into a gridlock status because of the interminable sitting tight for specific assets. It can't stop and finish unless we murder it.

- Soft Straggler

An undertaking that can finish its calculation effectively, yet will take any longer time than the normal assignments.

We propose a dynamic assignment allotment instrument called Speculative Execution Performance Balancing (SEPB) for a cluster of occupations with theoretical execution undertakings on top of Hadoop's present errand determination arrangement. Hadoop picks an undertaking from work in light of the accompanying need: first, any fizzled[4] assignment is given the most astounding need. Second, the pending assignments are considered. For guide, assignments with information nearby to the process hub are picked first. At last, Hadoop searches for a straggling errand to execute Speculative.

F) Slot PreScheduling

Keeping the errand calculation at the processing hub with the neighborhood information (i.e., Data territory[6]) is an efficient and critical way to deal with enhance the efficiency of space usage and execution. Delay Scheduler has been proposed to enhance the information area in MapReduce by . It defers the booking for a vocation by a little measure of time, when it identifies there are no neighbourhood map errands from that employment on a hub where its information live. On the other hand, it is to the detriment of decency[3]. There is a trade off between the information territory and decency enhancement with postponement scheduler. It implies that, in HDFS, delay planning is insufficient and there is still streamlining space for information area change. A testing inquiry will be : Are there any arrangements that can encourage enhance the information region while have no effect on dece. Following are the limitations of slot prescheduling of present hadoop .

- Master Failure :

It is anything but difficult to make the expert compose occasional checkpoints of the master data structures described above. If the master errand passes on, another duplicate can be begin from the last check pointed state. In any case, given that there is just a solitary ace, its disappointment is impossible; in this manner our current implementation aborts the MapReduce computation if the expert comes up short. Customers can check for this condition and retry the MapReduce operation if they desire.

- Locality :

Network bandwidth is a relatively 3902carcer source in our figuring environment. We save system transmission capacity by exploiting the way that the information (managed by GFS is put away on the neighbourhood circles of the machines that make up our bunch. GFS[3] partitions each file into 64MB blocks , and stores several copies of each square (regularly 3 duplicates) on various machines. The MapReduce master considers the location information of the data files and endeavors to plan a guide assignment on a machine that contains a copy of the corresponding input data. Coming up short that, it attempts to plan a map task near a replica of that task’s input data (e.g. on a specialist machine

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

that is on the same system switch as the machine containing the information). While running vast MapReduce operations on a significant division of the specialists in a bunch, most info information is perused locally and consumes no network bandwidth.

- Task Granularity :

We subdivide the guide stage into M pieces and the reduce phase into R pieces, as described above. Ideally, M and R should be much larger than the number of laborer[4] machines. Having every specialist perform a wide range of errands improves dynamic load balancing , and additionally accelerates recuperation when a laborer falls flat the numerous guide assignments it has finished can be spread out over the various worker machines.

IV. BENCHMARKS

WE PROPOSE TO IMPLEMENT FOLLOWING MENTIONED BENCHMARKS TO EVALUATE THE PERFORMANCE OF APPLICATION:

Sr no.	Name	Description
1.	Sort	Sort[1] the information in the data documents in a word reference request.
2.	Grep	Finds the matches of a regex[1] in the information documents.
3.	Inverted index	Takes a rundown of reports as information and produces word to record indexing
4.	Adjacency list	It is similar to search engine computation to generate the adjacency and reverse adjacency list of nodes of a graph for use by PageRank[1] -like algorithm.

V. CONCLUSION

Our paper proposes an application that is to build on Hadoop along with our modules like AEPB , PHSA , Slot Prescheduling . Our proposed system can improve the performance of the Hadoop system significantly with the help of used benchmarks (i.e., 46%-115% for single jobs and 49%-112% for multiple jobs). With this we will be able to improve present Hadoop by making complete utilization of memory resources while maintaining a balance between Mapping and Reduce operations. As there is a lot on research going on distributed systems, our paper presents an opportunity to build a system that can be used as an application by our government which was so far not done at such molecular level. Our system also helps in bringing digital India to rural India.

REFERENCES

[1] F. Ahmad, S. Y. Lee, M. Thottethodi, T. N. Vijaykumar. “PUMA: Purdue MapReduce Benchmarks Suite”, ECE Technical Reports, 2012.
[2] J. Dean and S. Ghemawat .”MapReduce: Simplified Data Processing on Large Clusters”, In OSDI’04, pp. 107-113, 2004.
[3] M. Hammoud and M. F. Sakr . “Locality-Aware Reduce Task Scheduling for MapReduce”. In IEEE CLOUDCOM’11. pp. 570-576, 2011.
[4] G. Ananthanarayanan S. Kandula, A. Greenberg, I. Stoica, Y. Lu, B. Saha, and E. Harris,”Reining in the outliers in map-reduce clusters using mantra”, in OSDI’10,pp. 1-16,2010
[5] Z.H. Guo, G. Fox, M. Zhou, Y.” Ruan.Improving Resource Utilization in MapReduce“. In IEEE Cluster’12. pp. 402-410, 2012.