

# Analysis of Smart City Related Data on Hadoop Environment

<sup>#1</sup>Prajakta Diwan, <sup>#2</sup>Ajay Fargade, <sup>#3</sup>Sujata Karpe, <sup>#4</sup>Shubham Nayak



<sup>1</sup>diwanpraju18@gmail.com  
<sup>2</sup>ajay25061991@gmail.com  
<sup>3</sup>sujata.karpe1993@gmail.com  
<sup>4</sup>srajnayak143@gmail.com

<sup>#1234</sup>Department of Computer Engineering,

G.H.R.C.E.M, Wagholi, Pune

## ABSTRACT

In recent years, advent of new technologies, devices, smart phone and communication like social media sites, the amount of data produced is growing rapidly every year. To harness the power of big data, you would require an infrastructure that can manage and process huge volumes of structured and unstructured data in real time and can protect data privacy and security. Apache hadoop is an open source software framework written in java for distributed storage and distributed processing of very large datasets on computer cluster. In the proposed work, they consider the constraints in the resource allocation on cloud process in the MapReduce process. For that they proposed the novel technique called DynamicMR framework. It contains the two major operations; they are slot utilization optimization and utilization efficiency optimization. The DynamicMR technique has the three slot allocation techniques they are Dynamic Hadoop Slot Allocation (DHSA), Speculative Execution Performance Balancing (SEPB), and Slot Prescheduling.

**Keywords:** Map Reduce , Hadoop, Clustering, HDFS, Cloud Data Center

## ARTICLE INFO

### Article History

Received: 31<sup>st</sup> October 2016

Received in revised form :

31<sup>st</sup> October 2016

Accepted: 3<sup>rd</sup> November 2016

**Published online :**

**9<sup>th</sup> November 2016**

## I. INTRODUCTION

Hadoop is java based framework that allows to process large data sets in distributed environment. Hadoop has been used by many large scale companies like Amazon, Facebook, and Yahoo[2]. Hadoop consist of two important concepts: Hadoop Distributed File System (HDFS) and Hadoop Map Reduce. Map Reduce workloads may be very heterogeneous in terms of their data size and their re- source requirements, and mixing them within a single instance of a computing framework may lead to conflicting optimization goals. Therefore, isolating Map Reduce workloads and their data while dynamically balancing the resources across them is very attractive for many organizations .Hadoop is an open source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming mode[5]. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Our System relaxes the slot allocation constraint to allow

slots to be reallocated to either map or reduce tasks depending on their needs. Second, thespeculative execution can tackle the straggler problem, which has shown to improve the performance for a single job but at the expense of the cluster efficiency[4]. In view of this, we propose Speculative Execution Performance Balancing to balance the performance tradeoff between a single job and a batch of jobs. Third, delay scheduling has shown to improve the data locality but at the cost of fairness. Alternatively, we propose a technique called Slot Pre Scheduling that can improve the data locality but with no impact on fairness. Finally, by combining these techniques together, we form a step-by-step slot allocation system called Dynamic MR that can improve the performance of Map Reduce workloads substantially.

## II. BASIC CONCEPT

### 2.1 Map Reduce:

Map Reduce is a processing technique and a program model for distributed computing based on java. It contains

two important tasks, namely Map and Reduce. The major advantages of MapReduce is that it is easy to scale data processing over multiple computing nodes.

### 2.2 Hadoop Distributed File System (HDFS):

It is distributed file system designed to run on commodity hardware. This system provides high-throughput access to application data. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. Application that run on HDFS has large data sets.[4] Typically file in HDFS is gigabytes to terabytes in size. It should support tens of millions of files in a single instance. HDFS is designed more for batch process in gather than interactive use by users. Detection of faults and quick, automatic recovery from them is a core goal of HDFS. HDFS has been designed to be easily portable from one platform to another. HDFS has a Master-slave architecture [6]. An HDFS cluster consist of a single Name Node, a master serves that manages the file system namespaces and regulates access to files by clients. In addition, there are number of data nodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on.

### III. BACKGROUND AND COMPARATIVE ANALYSIS

MR1 architecture, the cluster was managed by a service called the Job Tracker. Task Tracker services lived on each node and would launch tasks on behalf of jobs. The Job Tracker would serve information about completed jobs.[9]MRv1 uses the Job Tracker to create and assign tasks to task trackers, which can become a resource bottleneck when the cluster scales out far enough (usually around 4,000 clusters).

#### 3.1 Limitation:-

- 1) It limits scalability: Job Tracker runs on single machine doing several task like
  - o Resource management
  - o Job and task scheduling and
  - o Monitoring

Although there are so many machines (Data Node) available; they are not getting used. This limits scalability.

- 2) Availability Issue: In Hadoop 1.0, Job Tracker is single Point of availability. This means if Job Tracker fails, all jobs must restart.

3) Problem with Resource Utilization: In Hadoop 1.0, there is concept of predefined number of map slots and reduce slots for each Task Trackers. Resource Utilization issues occur because maps slots might be 'full' while reduce slots is empty (and vice-versa). Here the compute resources (Data Node) could sit idle which are reserved for Reduce slots even when there is immediate need for those resources to be used as Mapper slots.

### IV. PROPOSED SYSTEM

#### 4.1 Slot prescheduling:-

It improves the slot utilization efficiency and performance by improving the data locality for map tasks while keeping the fairness.

Step 1: Compute load factor  $\text{mapSlotsLoadFactor} = \frac{\text{Pending map tasks} + \text{running map tasks}}{\text{cluster map slot capacity}}$

Step 2: Compute current maximum number of usable map slots = number of map slots in a task tracker \* min map Slots Load Factor, 1.

Step 3: Compute current allowable idle map (or reduce) slots for a task tracker = maximum number of usable map slots - current number of used map (or reduce) slots.

#### 4.2 Dynamic Hadoop Slot Allocation:-

It attempt to maximize slot utilization while maintaining the fairness, when there are pending tasks (e.g., map tasks or reduce tasks). We break implicit assumption of MapReduce that the map tasks can only run on map slots & reduce task scan only run on reduce slots. In our proposed system we modify it that map and reduce tasks can be run on either map or reduce slots. There are 3 cases, Consider,

$NM = \text{Total number of Map tasks}$   $NR = \text{Total number of Reduce tasks}$   $SM = \text{Total number of map slots}$   $SR = \text{Total number of reduce slots}$

Case 1:  $NM \leq SM$  and  $NR \leq SR$  The map tasks which are running on map slots and reduce tasks are run on reduce slots, There is no borrowing of map and reduce slots.

Case 2:  $NM > SM$  and  $NR < SR$  We satisfy reduce tasks for reduce slots first and then use those idle reduce slots for running map tasks.

Case 3:  $NM < SM$  and  $NR > SR$  We can schedule those unused map slots for running reduce tasks. Case 4:  $NM > SM$  and  $NR > SR$  The system should be in completely busy state.

System architecture:

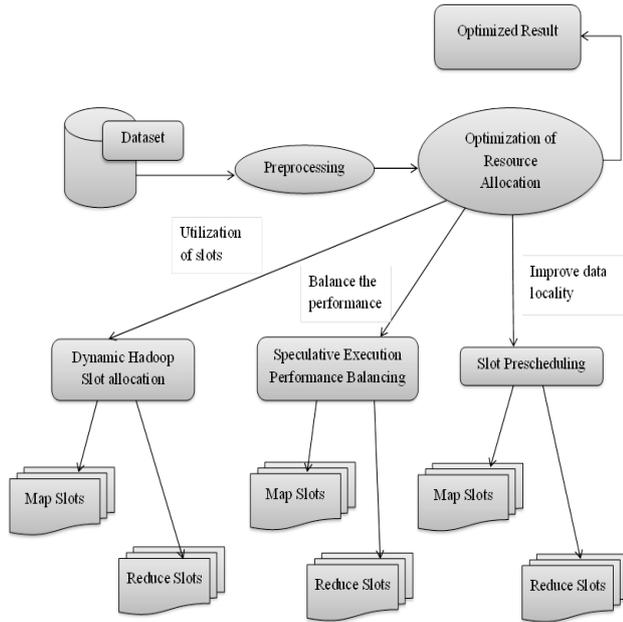


Fig 4.1. System architecture

4.3 Delay time scheduler :

- Time threshold decide on hadoop name node data node configuration .
- After delay time data will be temporary stored on pool and slot factorization will be done.
- Before delay time slot allocation will be occurred by standard hadoop configuration

4.4 Features

- Stores large database at the same time it can analyze the data using Map Reduce Algorithm.
- Hadoop processes data fast which is very useful for Real Time System.
- Improves the performance of Map Reduce workloads with maintaining the fairness.
- 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.
- SEPB identifies the slot inefficiency problem of speculative execution.

4.5 Application

- Providing Dynamic MR over Hadoop Framework as a service to IT companies.
- Providing our software as a service to Government System.
- Providing our system to any end-user or company needing Hadoop on multi-node cluster.

- Providing our software as a solution to any company having big data handling issues.

V. WORK FLOW

The admin login to system then upload to file.

First, we can classify the slots into two types, namely, busy slots (i.e., with running tasks) and idle slots (i.e., no running tasks). Given the total number of map and reduce slots configured by users, one optimization approach (i.e., macro- level optimization) is to improve the slot utilization by maximizing the number of busy slots and reducing the number of idle slots. Second, it is worth noting that not every busy slot can be efficiently utilized. Thus, our optimization approach (i.e., micro-level optimization) is to improve the utilization efficiency of busy slots after the macro level optimization. Particularly, we identify two main affecting factors(1). Speculative tasks . (2). Data locality . Based on these, we propose Dynamic MR, a dynamic utilization optimization framework for Map Reduce, to improve the performance of a shared Hadoop cluster under a fair scheduling between users.

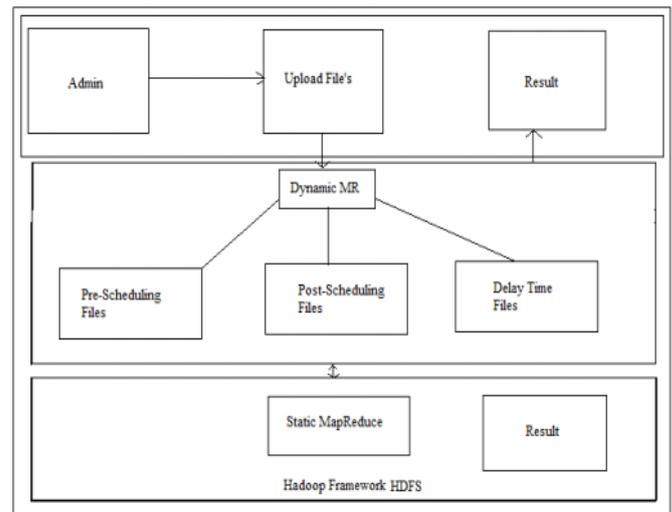


Fig5.1 Workflow of System

VI. CONCLUSION

The aim of the proposed system is to improve the performance of Map Reduce workloads. It considered three techniques: Dynamic Hadoop Slot Allocation, Speculative Execution Performance Balancing, and Slot Pre-Scheduling. Dynamic Hadoop Slot Allocation uses allocation of map to maximize the slot utilization and it reduces the task dynamically. It does not require any prior information or any assumption and it can be run on any kind of Map reduce jobs. Speculative Execution Performance Balancing identifies the slot inefficiency

problem. It manages the balance between single and batch of jobs dynamically. Slot Pre-Scheduling are used to enhance the efficiency of slot utilization by maximizing data locality. We can enhance the utilization by adding above concept in traditional system. A good trade-off between data utility and data consistency.

## REFERENCE

- [1]MapReduceTutorial[http://hadoop.apache.org/docs/r1.2.1/mapred\\_tutorial.html](http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html)
- [2]F:Ahmad; S:Y:Lee; M:T hottethodi; T:N:v ijaykumar :P UMA : Purdue Map Reduce Benchmarks Sui.
- [3]G:Ananthanarayanan; S:Kandula; A:Greenberg; I:Stoica; Y:Lu; B:Saha; andE:Harris; Reininginth
- [4]Apache Hadoop Next Gen Map Reduce(YARN ):[http://hadoop.apache.org/docs=current=hadoop-yarn=hadoop-yarn-site=YARN.html](http://hadoop.apache.org/docs/current/hadoop-yarn-hadoop-yarn-site/YARN.html):
- [5]J:Chao; R:Buyya: MapReduce Programming Model for :NET Based Cloud Com putting :In EuroP ar09; pp:417 428;2009:
- [6]Q:Chen; C:Liu; Z:Xiao; Improving Map Reduce Performance Using Smart Speculative Execution
- [7]J:Deanand S:Ghemawat:MapReduce : Simplified Data Processing on Large Clusters; In OSDI 04; pp 113; 2004:
- [8]Z:H:Guo; G:Fox; M:Zhou; Y:Ruan:Improving Resource Utilizationin MapReduce: InIEE ECluster1410; 2012:
- [9]Z:H:Guo; G:F ox;And M:Zhou: Investigation Of data locality and fairness in MapRe duce: In MapReduc 32; 2012: