A Self Adjusting Slots & Dynamic Job Ordering For Mapreduce Workloads Using Homogeneous And Heterogeneous Hadoop Cluster

Kaveri Hiremath[1], Dr. Reeja S. R[2]
[1] Dept of CS & Engg
[2]Associate Professor, Dept of CS & Engg
[1, 2] Dayananda Sagar University, Bangalore, India

Abstract- *In recent years, Hadoop has become the best platform for information analysis on huge data sets. MapReduce is the simple and standard programming language which is used for BigData applications. Open source apache Hadoop and its MapReduce framework are used. Currently, Hadoop Cluster uses only fixed slot configuration, means fixed number of Mapper slots and Reducer slots throughout the lifetime of the cluster. This fixed slot configuration leads to poor performance. And also MapReduce workloads generally contain a group of jobs in which it consists of multiple Mapper tasks followed by multiple Reducer tasks. The Mapper tasks will only run in the defined Map slots and Reducer tasks will only run in the defined Reduce slots. The general execution contains Mapper tasks are executed before reducer tasks. It is observed that for different MapReduce slot configuration and execution of job orders for MapReduce workloads have extremely distinct performance and system utilization. In this paper, developed a technique called dynamically allocates the slots for MapReduce jobs and using COSHH Scheduler to schedule the MapReduce slots.*

Keywords- MapReduce; Hadoop; Scheduling Algorithms; Slot Allocation; Workloads; Makespan.

I. INTRODUCTION

In cloud environment, cloud scheduler plays a major role in distributing resources for various jobs execution. For processing Big-data applications, simple and standard programming model is used in the Hadoop cluster. Many organizations, researchers are running MapReduce [1] applications on cloud, now a day’s.

Scheduling [4] the jobs in a MapReduce plays a major role in MapReduce applications for improving performance. The default scheduler in Hadoop MapReduce is FIFO scheduler, default scheduler in Facebook is Fair scheduler and Yahoo is Capacity scheduler. All these default schedulers are the examples for MapReduce applications which suits for static clusters. Therefore, need of dynamic scheduler which can allocate MapReduce applications based on the features of the applications.

Make span and Completion time are the two important keywords. Generally, Make span is the time taken from the starting of the job until the completion of the last job and Completion time is the total sum of completed time periods for all the jobs.

In this paper, proposes a COSHH scheduler which is used for classification and optimization of scheduler for Heterogeneous Hadoop system. And also allocates the slots dynamically for MapReduce jobs using Slot Assigner algorithm.

II. PROBLEM STATEMENT

In recent years, Hadoop has become the best platform for information analysis on huge data sets. MapReduce is the simple and standard programming language which is used for BigData applications. Apache Hadoop is the open source platform for MapReduce framework. In Hadoop system the major problem is to maintain the good performance of the system and to reduce the completion time (i.e., makespan) of a set of MapReduce tasks. The basic Hadoop system only allows a fixed slot configuration, i.e., constant number of map slots and reduces slots all over the period of a cluster. By this problem, reduces the performance of the resource utilization and makespan of the MapReduce tasks in the Hadoop system. In this work, proposes and implement a new mechanism to dynamically controls the allocation of map and reduce slots. And also using COSHH scheduler, we can classify and optimize the scheduler for heterogeneous Hadoop system.

III. MOTIVATION

- Currently, the Hadoop framework uses constant number of map slots & reduces slots on each node all over the period of a Hadoop cluster.

- A fixed slot configuration may lead to low resource utilizations and poor performance especially when the system is processing various workloads.

Based on these above challenges of the Hadoop cluster, Discussed proposed algorithm.

A. Dynamic Allocation of MapReduce slots using two phases.

The critical issue is to increase the performance of the system in Hadoop cluster. Two phases, Homogeneous and Heterogeneous phases are used to allocate the map and reduce slots dynamically i.e. controls the allocation of map and reduce slots automatically. By using these methods, minimize the makespan and total completion time.

B. Scheduling tasks and resource utilization and the job heterogeneity.

In this part we used scheduling algorithm, which gives a result of equal distribution of resources among MapReduce workloads in the system and also maximizes the resource utilization and output in multi-user cluster environment. In this algorithm, it considers the heterogeneity and MapReduce allocation of resources in the tasks.

IV. METHODOLOGY

Proposed Method

Dynamic Allocation of MapReduce slots using two phases.

The critical issue is to increase the performance of the system in Hadoop cluster. Two phases, Homogeneous and Heterogeneous phases are used to allocate the map and reduce slots dynamically i.e. controls the allocation of map and reduce slots automatically. By using these methods, minimize the makespan and total completion time.

Scheduling tasks and resource utilization and the job heterogeneity.

In this part we used scheduling algorithm, which gives a result of equal distribution of resources among MapReduce workloads in the system and also maximizes the resource utilization and output in multi-user cluster environment. In this algorithm, it considers the heterogeneity and MapReduce allocation of resources in the tasks. Here, we are using COSHH Scheduler to optimize the heterogeneous Hadoop Cluster.

COSHH Scheduler

- COSHH Scheduler is the Optimization and Classification based scheduler which is used for heterogeneous system.
- This Job scheduler is used to improve the makespan and total completion time of the MapReduce jobs.
- The Scheduler considers different at both levels i.e. application level and cluster levels of the system.
- This Scheduler is used to evaluate the performance.

Fig 4.1 shows the system Architecture of the proposed method, which describes Name node, Data node, Job tracker, Task tracker.

Name node:

The0Name0node is the ware equipment that contains the GNU/Linux OS and the Name node software. It is software that can be keeps running on commodity hardware. The framework having the Name node goes about as the master0server and it does the following assignments:

- Manages the document system0namespace.
- Regulates client’s0access to files/documents.
- It additionally executes0file0system operations, for example, renaming, closing, and opening records and directories.

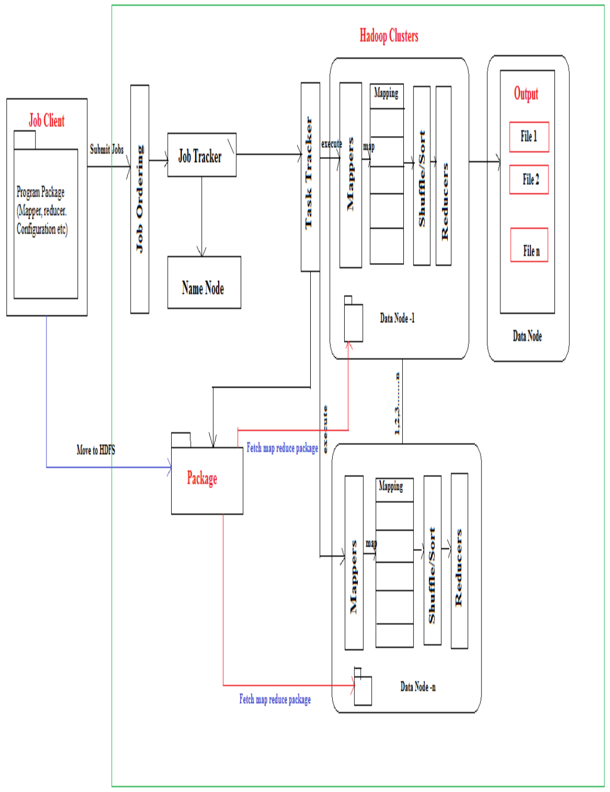


Fig 4.1 – System design for proposed method

Data node:

The Data node is item equipment having the GNU/Linux working framework and Data node programming. For each node (Commodity hardware/System) in a group, there will be an Data node. These nodes deal with the information storage of their system.

- Data nodes perform read-write operations on the record systems, as per client request.
- They also perform operations such as block creation, cancellation, and replication according to the instructions of the Name node.

Job tracker:

Job tracker is otherwise called master node, which is utilized for scheduling jobs and it will keep up the execution of tasks of each node. As applications are running, the Job tracker status updates from the task tracker nodes to keep tabs on their development and, if important, organize the treatment of any failures. The Job tracker needs to keep running on an master node in the Hadoop cluster as it arranges the execution of all MapReduce applications in the bunch, so it's a mission-critical service.

Workload Monitor (WM): Used to estimate the present workloads.

Slot Assigner (SA): It decides the best slot assignment of each node.

Schedule: Assigns tasks to slave nodes.

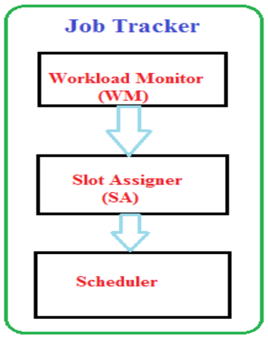


Fig 4.2 Working flow of Job Tracker in method TuMM

Task tracker:

Task tracker keeps running on each slave node which is utilized for execution of MapReduce jobs. As a slave node, the Task tracker gets processing requests from the Job tracker. Its essential duty is to track the execution of MapReduce workloads happening locally on its slave node and to send notices to the Job tracker.

V. MODULES

There are two following modules:

- Homogeneous
- Heterogeneous

A. HOMOGENEOUS PHASE

We present our solutions that dynamically allocate the slots to map and reduce tasks during the execution of jobs. When one slot becomes available upon the completion of a map or reduce tasks, the Hadoop system will reassign a map or reduce task to the slot. There are totally n tasks and at the end of each task, Hadoop needs to decide the role of the available slot.

In this Phase, we choose two Hadoop benchmarks which are used to analyze the data:

Inverted Index:

Here, we are taken a text documents as input to this benchmark and generate word to document Indexing.

Classification:

In this part, taken some simple Medical data as input and classify the patient's details into predefined clusters.

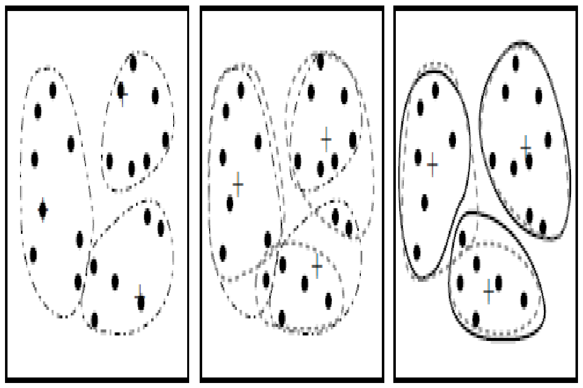
Here, we are using K-means algorithm to classify the dataset in the form of cluster in classification benchmark. By using this algorithm we can easily classify the data into clusters. In this algorithm, input should be k number of clusters and D is the dataset containing n objects. Output: A set of K clusters.

K – Means Algorithm to Perform Clustering

Algorithm

Input:
K: the number of Clusters,
D: a data set containing n objects.
***Output**: A set of K clusters.*
***Method**:*
1. Arbitrarily choose k objects from D as a initial cluster centres;
2. Repeat
3. (Re) assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
4. Update the cluster means, i.e. calculate the mean value of the objects for each cluster;
*5. until no change**;***

K- Means algorithm, first arbitrarily selects k number of objects from D dataset and repeat the step. Next it calculates the mean value of the objects and reassigns the similar objects to the cluster. Then update the mean value of the cluster and repeat the same steps until no changes in the K clusters.



Working of K-Means Algorithm

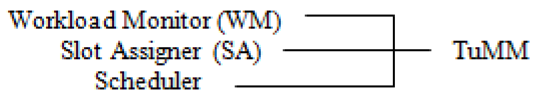
B. HETEROGENEOUS PHASE

When deploying a Hadoop cluster in such a environment, tasks from the same job may have different execution times when running on different nodes.

Working Example:

In this phase, contains client server connection. Here we are storing data in other server and load into the HDFS. The client will load the data set and send the query request to the server. After that server responds the query process and sends the result of that query to the client.

Our main goal of using these phases is to dynamically allocate the MapReduce slots in Hadoop system. To allocate the slots in dynamic manner, we are using method called TuMM. This Method helps to allocate the map and reduce tasks based on the workload information. Here we are using new version of this method i.e. H TuMM which gives similar idea what TuMM method has.



VI RELATED WORK

To generate a collection of tasks, clients characterizes a Mapper work as a key/value pair of middle of the intermediate values and a Reduce work that joins all these intermediate values related with a similar intermediate key. For a wide range of purposes, Google has been effectively utilized the MapReduce [1] programming model. For the accomplishment of this model, they showed to a few reasons. This MapReduce programming model is basic/simple to enhance the resources of a large distributed system for software programmers without having any learning in parallel and distributed systems, since it mask the points of interest of parallelization, fault tolerance, and locality optimization and load adjusting, This MapReduce execution keeps running on a substantial piece of having a machines and is highly extensible. Created several MapReduce projects and MapReduce occupations are executed more than thousands consistently on Google's cluster.

Large-Scale MapReduce [1] Clusters that frequently processes a number of unstructured & semi-structured data in the clouds. To increase the effectiveness or utilization of these MapReduce clusters is the main key Challenge.

[2] Considers a small set to produce workload that consists of MapReduce tasks. By these subsets, they came to know that the order in which these tasks are executed can have a remarkable impact on their overall completion time and the resource utilization. To address this issue, they thought to robotize the design of a MapReduce job schedule that improve the completion time. They used abstraction framework called Balanced Pools to automate this design; this framework efficiently increases the performance of MapReduce jobs in a given workload for creating an optimized job schedule. By simulation, they achieved 15%-38% utilization improvements.

Quincy [3] tells about the scheduling the parallel jobs on MapReduce clusters. In this paper, for scheduling the distributed jobs, they have introduced a extendable new framework called fine-0grain0resource sharing . The scheduling issue is mapped to graph data structure to demands0of0data0locality,0fairness & starvation-freedom to schedule according to a global cost model.

They evaluate their implementation of this new framework called Quincy [3] on a number of computers using various workloads of data and CPU-intensive jobs. And also evaluate against actual queue based algorithm and develop several rules for each scheduler with & without fairness constraints. This proposed framework gets good propriety when fairness is requested, while effectively improving data locality. The amount of data transferred on cluster while scheduling is reduced up to the factor of 3.9 and throughput increases to 40% by the experiments.

MapReduce [1] is a programming model which is used for data analytics. They consider a multiple data analytics applications can share the same physical resources in the environment of managing MapReduce applications. This sharing of multiple applications with the data center leads to more solid workloads to save the cost and energy. In this shared environment, it is important to maintain0the performance0of MapReduce workloads. To address this issue they introduced a task0scheduler [4] for MapReduce framework. This task0scheduler0allows performance-driven [4] management of MapReduce tasks. This proposed0task0 scheduler adjusts allocation for the MapReduce jobs and dynamically estimates the performance0of0multiple0jobs at a same time in the MapReduce tasks.

Paper [5] tells about the performance and resource utilization of the MapReduce in the multi-job workloads. Existing MapReduce schedulers defines a static number of MapReduce slots to represent the quantity of a Hadoop Cluster. This works for homogeneous workloads, but not for different resources i.e. heterogeneous workloads. Their technique holds job writing information to adjust the slots for MapReduce dynamically and also to increase the performance of the cluster.

Some classical0MapReduce0platforms have some low performance in the resource utilization, since0the traditional0multi-phase0parallel model. To address this problem, they used two technologies [6] i.e. Dynamic0Resource0Allocation & Resource Usage Pipeline. Dynamic0resource0allocation considers the priority of the jobs and requirement of jobs while resource allocation and Resource Usage Pipeline assigns tasks dynamically. By using these techniques they improved their throughput by 21.72%.

The key challenge in MapReduce cluster is to control the MapReduce allocations for different applications to improve the performance. The issue is that for Mapper and Reducer jobs there is no job scheduler in a given ccompletion time, could distribute the relevant number of resources to the MapReduce jobs so that it meets the required Service Level Objective (SLO). To solve this problem, they introduced a framework, ARIA [7]. This framework is based on the observation, firstly they can profile a job which runs routinely and then uses its profile in the MapReduce performance to calculate the chunk of resource enforced to meet a deadline.

These Resource requirements are causing to bring the SLO-scheduler. In this job, ordering is formidable by the EDF (Earliest Deadline First) [7] scheduling which is a supporting policy for real-time processing. The EDF scheduling, as compared to traditional assumptions there is some interesting contrast in MapReduce environments. This creates new opportunities for various scheduling policies.

Allocating a MapReduce cluster between users is inspirable because it empowers statistical lowering cost and allows a user to share a large data set. They find that the traditional scheduling algorithms can perform poorly in

MapReduce due to the characters of the MapReduce settings: available of data location and the dependence between Mapper and Reducer tasks. To overcome these problems, designed a fair [8] scheduler for MapReduce and they developed two simple techniques i.e. Delay Scheduling and Copy-Compute Splitting. By using these techniques, improves the throughput and completion time by 2 to 10.

Dynamic load balancing [9] approach to distributed server farm system. In this paper, they proposed a load balancing approach which effectively overcomes the load imbalance caused by the non-considerable number of large tasks. This proposed approach selects a set of active servers dynamically and allocates the loads in a given time. A new approach deals with very large tasks to reduce the waiting time at server queue. A task deadline can be defined as another parameter to the task priority. This priority calculates dynamically for each task in the server queue.

As we discussed in the above papers, most of the problem is to increase the performance and resource utilization and efficient resource management [10] in large data. Hadoop cluster allocates the fixed size for MapReduce tasks and static slots for MapReduce nodes. To overcome this problem, they design and implemented fine-grained, dynamic & coordinate resource management framework called MROrchestrator [10]. By using this framework they reduce 38% of job completion times and up to 25% increase in the resource utilization.

VII. RESULTS

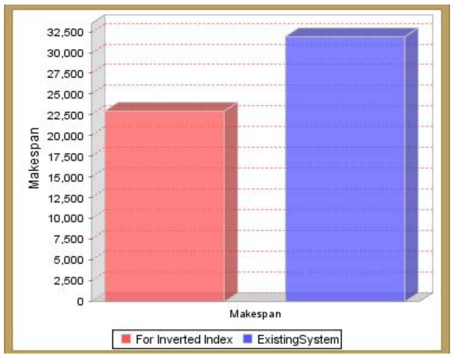


Fig 7.1 Bar graph of analysis in Inverted Index

Fig 7.1 observing the make span of the Map Reduce jobs in the Hadoop system. Make span is the time period from the start of job to the completion of last job for a set of jobs.

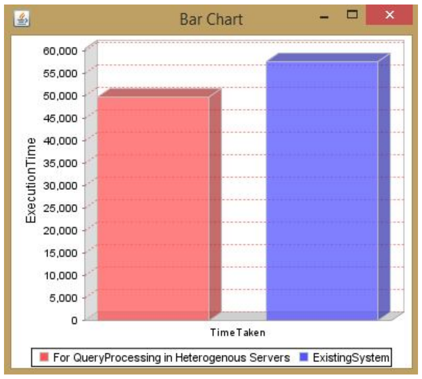


Fig 7.2 Bar chart of Execution time in Heterogeneous phase

Fig 7.2 shows the bar chart of total completion time of the MapReduce tasks which is used to improve the performance of the Hadoop system. The bar graph compares the existing system and Query processing of Heterogeneous system in proposed method.

VIII. CONCLUSION

Now a day's, huge increase in volumes of data and big data. The main key challenge is to maintain this large amount of data, so this became an important point to research. In this paper, we discussed the method TuMM which dynamically controls the MapReduce slots for MapReduce jobs by using Slot Assigner algorithm.

The proposed method consists of two phases i.e. Homogeneous and Heterogeneous phase. Both the phases are used to allocate the slots dynamically which is used to improve the performance of the Hadoop system.

The scheduler is used to maximize the resource utilization and classifies the MapReduce jobs to find a matching of the resulting job classes. The main goal of this project is to improve the performance of the system. In the future work, consider Map/Reduce slot configuration issues for online jobs under FIFO scheduling and also for further, can consider other scheduling algorithms to compare the results to increase the performance of the system. And for Job ordering optimization using bi-criteria algorithm to optimize both make span and total completion time simultaneously.

REFERENCES

[1] J. Dean and S. Ghemawat, “MapReduce: Simplified data processing on large clusters”, Communications of the ACM, vol.51,no.1, pp.107-113,2008.

[2] A. Verma, L. Cherkasova, and R. H. Campbell, “Two sides of a coin: Optimizing the schedule of map reduce jobs to

minimize their makespan and improve cluster performance”, in MASCOTS’12,Aug 2012.

[3] M. Isard, Vijayan prabhakaran, J. currey et al., “Quincy: Fair scheduling for distributed computing clusters”, in SOSP’09,2009, pp.261-276.

[4] J.polo, D. Carrera, Y. Becerra et al., “Performance-driven task co-scheduling for map reduce environments”, in NOMS’10.2010.

[5] J. polo, C. Castillo, D. Carrera et al., “Resource-aware adaptive scheduling for map reduce clusters”, in proceedings of the 12th ACM/IFIP/USENIX international conference on Middleware, 2011.

[6] X. W. Wang, J. Zhang, H. M. Liao, and L. Zha, “Dynamic split model of resource utilization in map reduce”, in DataCloud-SC’11,2011.

[7] A. Verma, Ludmila Cherkasova, and R. H. Campbell, “ARIA: Automatic Resource Inference and Allocation for map reduce environments”, in ICAC’11,2011, pp.235-244.

[8] M. Zaharia, D.Borthakur, J. S. Sarma et al., “Job Scheduling for multi-user map reduce clusters”, University of California, Berkeley, tech.Rep., Apr.2009.

[9] L.Tan and Z. Tari, “Dynamic task assignment in server farms: Better performance by task grouping”, in proc. of the Intl. Symp. On computers and communications (ISCC),2002.

[10] B. Sharma, R. Prabhakar, S.-H. Lim et al., “Mrorchestrator: A fine-grained resource orchestration framework for map reduce clusters”, in CLOUD’12,2012.

[11] Hadoop. Capacity Scheduler. http://hadoop.apache.org/common/docs/r0.19.2/capacityscheduler.html.

[12] Hadoop. Fair Scheduler. http://hadoop.apache.org/common/docs/r0.20.2/fairscheduler.html

[13] TPC-H benchmark.[online].Available: https://www.tpc.org/tpch/

[14] TPC-H benchmark on pig.[online]. Available: https://issues.apache.org/jira/browse/PIG-2397

[15] Yi Yao, Jiayin Wang, Bo Sheng et al., “Self- Adjusting Slot Configurations for Homogeneous and Heterogeneous Hadoop Clusters” DOI 10.1109/TCC.2015.2415802, IEEE Transactions on Cloud Computing