AN ENHANCED MATHEMATICAL MODEL FOR CLOUD BASED DATA ORIENTED JOB ANALYSIS

Manishankar S¹, Santosh Anand²
Department of Computer Science, Amrita School of Arts & Sciences, Mysuru
Amrita Vishwa Vidyapeetham, India
smanishankar@asas.mysor.amrita.edu¹,santoshanand.jha@gmail.com²

Abstract. Data processing or data analytics is the common functionality that is attached to most of the real world applications. The amount of processing required for data oriented tasks or jobs are quiet high. To resolve the processing issue the most common approach deployed is by using a high performance cluster. Setting a cluster over real time infrastructure leads to a very expensive solution. A Cloud based infrastructure remains as an ideal support for setting up a cluster. Managing the cluster over a Cloud is a challenging task as allocation of infrastructure based on the task schedule is a critical parameter. The proposed mathematical model introduces a strategy to allocate the infrastructure and manage the load of the cluster based on Queuing model. The experimental setup is made on top of private cloud and Hadoop based data processing jobs are tested. The proposed data oriented resource optimizer enhances the performance of the cluster by balancing the increased load due to data processing jobs. The result shows the enhanced improvement in performance compared to default resource manager.

Keywords: Cloud Computing, Hadoop Cluster, Queuing model, Resource optimization.

1. INTRODUCTION
Storage of real time generated data from information system is a major challenge in most of the data management system and data warehouses. Storing data is not only the concern it is also important how the data can be processed and handled. Different methods are used to support exploratory, statistical, and predictive which can analyze data with different views that help in interpret and visualizing data according to the various research perspectives [1]. So there arises a requirement for an efficient and structurally strong data processing system like Hadoop to support the data processing.

The research work carried on job oriented data analysis introduces the concept of having multiple jobs run on a cluster, the analysis is termed as job oriented because the scheduler allocates the job based on the type of operation. A queuing model based on response time of cluster is setup to provide a support for the cluster. The elastic cloud helps in reallocating or reutilizing the resource based on the system use. The importance of setting a real time data based analysis in various fields like sensor network can be incorporated where the sensor inputs can be analyzed with cluster [1]. The performance results are compared with the YARN based scheduler. The system is an efficient cloud based solution for jobs of varying sizes and types. The queuing model enhances the performance of the mediation node which is key in making the system faster as it takes in inputs [2].

Real time application like sensor load balancing to optimize energy and enhance the life time of wireless sensor network. The protocol proposed in the research work to design the path which is similar queuing model for making the process execute [3]The major requirement in future storage system is the need for setting up an integrated multi cluster environment which can execute parallel data processing and analytical operations. A kind of system that will optimize the resources involved in storage and data processing system [5] and which has job-oriented view for each of the operation performed on data. Various Thus, the research introduces a efficient and deeper approach involved in Big Data processing and analytics.
2. RELATED WORK

Data Oriented job Analysis in a Distributed Cluster Environment is a field of research where lot of works have been carried on because all real time applications are connected with data storage and retrieval. Survey of recent works in the research field are depicted below [3] Considers the FBC (frequency-based-center) and proposes a new function for performing K-means clustering on a categorical dataset. [4] The work uses the FBC concept to find the new center of the categorical data set. Has developed an algorithmic framework for solving the projected clustering problem, and has tested its performance on synthetic data.[5] SHadoop is implemented. An enhanced and fully attuned version of Hadoop, is intended to shorten the performance time, cost of MapReduce work, particularly for short term work.[6] There are 3 alternates solutions of the K-Means algorithm for incorporating binary data streams. These variations contain the K-means which is extensible online, the , and the K-means incremental approach, which is a anticipated variant that can achieve a higher quality solution in a shorter time. The proposed algorithm can be used to monitor transactions.[7] proposed the work by using the greedy method to select the random point to solve the problem. Using the idea of cluster center initialization. [8] The clustering of the numerical data set is the distance from the Manhattan distance, not the Euclidean distance. In [9] the use of association rule mining (ADMLCAR) anomaly detection Multi-tag classification is used to solve MLC problems. In overall, multi-label classification difficulties are mostly answered by one of two approaches: problem conversion, algorithm adaptation. [10] A compacted over turned directory data structure that might help in creeping for words in dictionary direction such that all the directories built for millions of documents need not be administered has been projected. [11] has implemented PCA-based graphical similarity measurements and generate different Laplacian spectra for spectral clustering. [12] Lecturing the problem of k-medoid clustering by MapReduce agenda for distributed computing on commodity machines to assess its competence. [13], further motivated by the Map-Reduce has revisited the two-stage flow shop problem and have given a dynamic program for minimizing the total flow time especially when all the jobs arrive at the same time. The paper also proposes a scheduling model that captures the challenges in Map-Reduce scheduling. [14], they have proposed simplified and effective strategies which use ratio-slot between maps and reduce tasks which act as an effective knob.

The experimental results show the working correctness and compilation of schemes under both simple, complex and mixed workloads.[15] had studied the Map Reduce job performance in HadoopV1 and YARN with different resource configuration. This made them to propose Map Reduce, which can dynamically manage a number of corresponding tasks running on each node. [16], has proposed in the case study for Smart City Corporation that the data in today’s world is generated rapidly from various sources such as mobile phones, machines, etc and these data generated has to be handled in order to serve a variety of purposes. [17] has studied Map Reduce which is a prevalent time for large-scale data dispensation in cloud computing. Though, the slot-based Map Reduce system (e.g., Hadoop MRv1) may possibly perform poor due to resource allocation in an un-optimized manner.[18] Explains they range real time cluster scheduling approach to explanation for the two-phase calculation style of Map Reduce. Also progress measures for scheduling jobs built on user quantified limit limitations.[19] A new method to scheduling jobs using genetic algorithms (GA) on a service grid. The fitness function to minimize the average execution time of M (≤ N) machines that schedule N jobs onto the grid. Two models were proposed to predict the execution time of a single job or multiple jobs on each machine with a different system load. A single service type model is used to schedule jobs for a single service to a machine, while multiple service type models schedule jobs for multiple services to a machine. The predicted execution time from these models is used as input to the genetic algorithm to schedule N jobs to M machines on the grid. Experiments on a small grid of four machines show that the new job scheduling method significantly reduces the average execution time [20] an algorithm has developed to list heterogeneous tasks on heterogeneous processors in a circulated system. The scheduler runs in an environment changing resources and adapts variable system resources. It uses genetic algorithms to minimize the execution time. Comparing the scheduler with six other schedulers, three batch modes, and three immediate mode schedulers. Experiments show that the algorithm is superior to other algorithms and can achieve near-optimal efficiency.
3. PROPOSED MODEL OF CLOUD-BASED CLUSTER

Real time data tasks and operations related to data processing requires a high performance, high available and reliable computing environment. Cluster based systems are found to support computing environment efficiently. A cloud based computing system is very much desirable as virtualizing the infrastructure helps in a better way to setup a computing cluster with less cost. The proposed architecture is a Hadoop based cluster setup over a private OpenNebula Cloud. Hadoop cluster are responsible for the execution of the Big data processing job, whereas Cloud provides the elastic infrastructure to support the executions.

The proposed model is having multiple cluster which is setup on top of the private Cloud. The Cloud is configured with Virtual machines as depicted in Figure 1 to manage cluster.

![Figure 1. Architecture of the Cloud-based cluster](image)

The cluster has a Name node, Data node and processing nodes. Important task is to monitor the cluster when jobs are executed. A performance monitor is formed with help of a monitoring tool Ganglia. Various types of data processing jobs are tested on top of the cluster. Running parallel jobs of varying size of cluster seems to increase the load of the cluster by a large extend. As data processing jobs have many parallel executing jobs there arises a need for a model to allocate the data oriented jobs in the cluster.

Hadoop Cluster has many type of schedulers, YARN (Yet Another Resource Negotiator) in Hadoop plays a vital role in managing the resources in the cluster. YARN ensures the availability of resources within the Hadoop. To monitor the performance of the system we require a ganglia to be installed. Ganglia being a cluster monitor records the readings from processing environment which can be utilized by the proposed schedule in the research to work efficiently, once the cluster is set up and running it is important to try out various jobs in the cluster and record the performance using ganglia.

The Hadoop cluster is run with a variety of jobs such as Word Count, Number Sum and transaction management. Map Reduce operation on YARN is compared to give a faster result than traditional MapReduce. YARN components are Client who submits a MapReduce Job of any one category above mentioned. Resource Manager controls the usage of resources in the Hadoop cluster. Node Manager is responsible for running the jobs assigned, it creates containers to hold the job and also monitors the usage of the job container. Application Master coordinates MapReduce jobs and schedule task by negotiating with the resource manager. Hadoop Distributed Filesystem shares the jobs between YARN components. The java interface we created runs a code necessary for the job. The job object is identified and a new application is submitted to the resource manager.
The resource manager checks the availability and forwards the job to the node manager. Node manager creates the container for the job and starts the application master. The application manager creates a YARN child required to run the job with the help of a node manager YARN will execute Map as well as reduce task and acquire the job resources from the HDFS. Once the job is submitted an automatic application Id is retrieved from the resource manager. Distributed task allocation and management is done by Application Master if job size is small the Application Master executes both Map and Reduce task in single JVM. Conditions for the job to run on single JVM is

- Less than 10 Mappers in job
- Single reducer task
- Input data size is less than one HDFS block

If the tasks are not satisfying these condition then the scheduler has to check the availability of the resource and see where to place the task. Node manager allocates the containers that has Map and Reduce job present in them. Once the containers are scheduled the total sum of all job containers should not exceed the assigned threshold.

Calculating the physical memory limit of a job is also essential the container memory usage will be termed by a calculation where

**CM-Container Memory**

**HS-Heap size of Virtual machine**

**NL-Native Library**

**MU-Memory used by other spawned process. JP- java VM per Region.**

\[
CM = \sum (HS + NL + MU + JP)
\]

Equation 3.1 depicts that the commuted memory is a total of all virtual machine memory partitions. As it seen even though the default schedulers allocate maximum and minimum values for the memory allocation. For example a Capacity scheduler has a minimum memory of 1024MB to maximum memory of 10240MB, so any request that is between Min and Max will be processed by the scheduler. The memory model in such a way that in allocated memory some part is assigned to Map and other is assigned to reduce based on the number of Mapper and Reducer function and configuration in Yarn site.XML. The Node manager in configuration of Yarn.site.XML has a 10 Mappers or 6 Reducers may be a combination of the above. So the node manager Yarn scheduler capacity of minimum allocation will be given a client request. Based on the availability Application master assigns the Node manager to allocate the container based on number of Map and Reduce present in the Job assigned. For each task Node manager starts a Yarn child, which is executed in the allocated JVM. It also copies resource locally as configurations or Jar files. Then Yarn executes the Map and Reduce tasks. The report of the running job and task is polled by the Application Master. The task continues until the application master is confirmed that job is done. Client can monitor the status of job by polling the application master with help of progress monitor. Once the cluster is set and the performance is observed, there arises a requirement for model for assigning jobs to the YARN so the optimization of the load of cluster is possible
4. Mathematical Model for data-oriented job analysis

To assign P Process to N node in C cluster we require queueing model in which a mediation server MS is introduced which is the entry point of the user request to run job arrives. The jobs are forwarded to the Cluster nodes CNi for processing the jobs. Where i=1,2,3,…N are the nodes of the cluster. The scheduler or load balancer is represented as M/M/1 Queue which works with probability and arrival rates are calculated by Poisson distribution. The arrival rate of jobs are represented as \( \lambda \) to the load balancer as a Poisson process. A service rate is fixed for the cluster nodes, if \( \lambda \) is less than the service rate \( S \), the schedule is fixed and the task is executed.

Each cluster Node access the datacenter DCi with a probability \( \beta \), the rate in which the server will be processing the I/O operation depends on the service rate of the model M/M/1 Queue. For calculating the cluster performance the measurement of response time of the mediation node, cluster node are evaluated.

The total response time \( T \) of the Cloud will be a sum of response time of the queuing servers.

\[
T = T_{MS} + T_{CN} \tag{3.2}
\]

The response time of the MS will be obtained based on response time of the M/M/1 queue, which is calculated as

\[
T_{MS} = \frac{1}{s} \frac{1}{1 - \lambda/s} \tag{3.3}
\]

Where \( S \) is the service and \( \lambda \) is the arrival rate

Processing response time of cluster also plays a vital role in modeling the server. The N nodes of the cluster which is in the queuing model, the response time can be calculated as

\[
T_{CN} = \frac{1}{\alpha} + \frac{P(n,\rho)}{n\alpha - \lambda} \tag{3.4}
\]

Where \( n \) is the number of clusters and \( \alpha \) and \( \lambda \) service rate and arrival rate of jobs. The \( P(n,\rho) \) is the probability of job getting added to the queuing model

\[
P(n,\rho) = \frac{\left(\frac{cp}{c}\right)^n \left(\frac{1}{1-\rho}\right)}{\sum_{k=0}^{n} \frac{cp^n}{k!} + \left(\frac{1}{1-\rho}\right)} \tag{3.5}
\]

Once the response time and memory is calculated the server takes performance values monitored by the ganglia and apply to the queuing model to get the response time and the queuing delay.
5. IMPLEMENTATION AND RESULTS

In order to conduct the proposed research a Hadoop cluster environment is setup with help of Hadoop. The Hadoop cluster is monitored for the performing using ganglia observations collected for various predefined data and jobs are recorded. The Hadoop cluster has component like Name Node, Resource Manager with YARN, File System of distributed nature and Hadoop Map reduce Ecosystem and YARN based parallel processing of dataset. There will be a name node who act as a master and who will have the job tracker and job analyzer which we introduced in the proposed architecture.

The slave node has task trackers which are coordinated by adaptive scheduler and each one has a data node and does the map reduce operation required. YARN with adaptive scheduler enables dynamic resource utilizations and job oriented computations which can run various applications without having any worry about increase the load the resource manager who manages and allocate cluster resources feeds the data to the adaptive scheduler the node manager who enforces node resource allocation and management is connected to resource pool and job table. The application manager manages the life cycle of the application in co-ordination with the proposed job oriented analytic engine.

The results from the performance monitor shows that the cluster is heavily loaded and the utilization initially is degraded. The performance degradation of YARN occurs when the jobs are of varying sizes as depicted in Figure 2.

The Table 1 depicted above show the cluster node response data. Data which gives job id denotes the number assigned to the incoming jobs. Rid depicted as Resource id represents allocated resources. Job type which is categorized based on the data by job analyzer puts the jobs into three categories High, Medium, low. Execution time represents the time taken for the execution of the job in job tracker noted from ganglia. Memory denotes the usage of the memory, R status depicted as resource status denotes whether the job is assigned to a resource or not. CPU depicts CPU utilization.

The overall load of the cluster while a particular job is running is noted as load. The time taken for map and reduce jobs inside the slave node are recorded with help of task tracker and listed as map time and reduce time. Job analyzer which collects the input data and decides on job type based on the mathematical model and decision algorithm.

![Figure 2. Performance degradation without Yarn optimization.](image-url)
Table 1. Cluster performance data obtained from monitor.

| Job id | Rid | Job type | Exe time | Memory        |
|--------|-----|----------|----------|---------------|
| 1      | 1   | H        | 1.454    | 1971053909    |
| 2      | 2   | L        | 0.2973333| 25346068      |
| 3      | 1   | L        | 0.9513333| 25023829      |
| 4      | 1   | H        | 1.989    | 27986542      |
| 5      | 3   | L        | 0.3433333| 26789979      |
| 6      | 1   | L        | 0.98     | 27000019      |
| 7      | 4   | H        | 2.01     | 30023126      |
| 8      | 2   | L        | 0.4532234| 27651239      |
| 9      | 1   | L        | 1.02     | 27609862      |
| 10     | 5   | H        | 1.39     | 438457132     |
| 11     | 1   | M        | 1.46     | 773632000     |
| 12     | 3   | L        | 1.04     | 496620612     |
| 13     | 4   | H        | 1.54     | 446523433     |
| 14     | 1   | M        | 1.07     | 7865425548    |
| 15     | 2   | L        | 1.06     | 499932168     |

Once the response time and memory are calculated the model helps in categorizing the job based on arrival rate.

The figure 3 depicts job analyzer categorizes the job into various level sets based on the load execution time, CPU utilization and memory. The data obtained from ganglia used to do the categorization and with the help of a decision tree algorithm job analyzer divides the job as depicted in the diagram above. The jobs are categorized into high, low, medium, very high, very low. training data set consist of 17 various types of jobs based on different inputs.

The YARN with the adaptive scheduler makes critical decisions on allocating jobs to resource of the cluster. Our modified scheduler receives input from the job analyzer and makes mapping of jobs to the table and searches in the resources for the best available resource to execute the job. The use of cross mutation genetic algorithm helps in correlating and mapping jobs to the resource. The algorithm creates the fitness function which can evaluate the job based on parameter like execution time, cpu utilization, memory utilization, load etc and generations of job in accordance to resource created.
The figure 4 above depicts the plot of genetic algorithm mapping resources and jobs. The plot depicts the generations in x-axis and fitness in y-axis. The best mean, median are depicted in plot. The mapping of resources to job helps in resource optimization. The most appropriate resource is identified by the job analyser and the genetic algorithm supporting it. Based on the fitness function and the cross mapping function an exact match of the job to resource is obtained. The results of the comparison of the YARN scheduler and the proposed Queuing model based schedule

Table 1. Execution time results

| Number of Jobs | Number of Resources | Execution Time(YARN) | Execution Time(Adaptive Scheduler) |
|----------------|---------------------|----------------------|-------------------------------------|
| 5              | 3                   | 5.035                | 2.034                               |
| 10             | 5                   | 12.34                | 8.34                                |
| 15             | 7                   | 17.05                | 13.04                               |
| 20             | 8                   | 21.35                | 16.05                               |

The graph represented in Figure 5 represents the comparison of execution time with presence of queuing model and default YARN scheduler in the Hadoop cluster.
6. CONCLUSION

Distributed cluster environment that does computation like data processing and analytics require a proper model that can support the scheduler in handling heterogeneous data processing jobs. So, this work aims to provide a platform which can handle the heterogeneous data operation with accessible infrastructure and resources by including few classifications of that data during preprocessing stage and then applying various analytics with the help of queuing based scheduler for the multi clustered environment. The classification of jobs and resources are based on the mathematical model. The mathematical distribution model is applied to make selection of resource and job based on the input data. A scheduler supervises the correlation between job and resources allocation which enhances the performance of cluster environment and thus providing an efficient job-oriented computation and analysis.

REFERENCES

[1] Gonzalez, T F 1985 Clustering to minimize the maximum intercluster distance Journal Theoretical Computer Science volume 38,page 293-306
[2] Aggarwal C C, Wolf J L, Yu P S, Procopiuc C and Park J S 1999 Fast algorithms for projected clustering.mln ACM SIGMoD Record Vol. 28, No. 2, pp. 61-72 ACM.
[3] Santosh Anand Pillai and Atulya Radhakrishna A Protocol for the Effective Utilization of Energy in Wireless Sensor Network with Load Balancing Pages: 82-86, DOI: 10.14419/ijet.v7i3.14491
[4] Gu R, Yang X, Yan J, Sun Y, Wang B, Yuan C and Huang Y 2014 SHadoop: Improving MapReduce performance by optimizing job execution mechanism in Hadoop clusters Journal of parallel and distributed computing 74(3) 2166-2179
[5] Ordonez, C 2003 June Clustering binary data streams with K-means. In Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery(pp. 12-19). ACM
[6] Li Q and Liu X 2014 November A K-medoids clustering algorithm with initial centers optimized by a P system. In International Conference on Human Centered Computing (pp. 488-500). Springer Cham
[7] Legany Csaba, Juhas., Sandor Babos and Attila 2006 Cluster Validity Measurement Techniques In proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases pp:388-393 WSEAS
[8]. Santosh Anand and Akarsha R R 2018 A Protocol for the Effective Utilization of Energy in Wireless Sensor Network, Participated and presented paper in ICTEMPS-2018 conducted by Geethanjali College of Engineering and Technolog Pages: 9398, DOI: 10.14419/ijet.v7i3.14495,
[9] Prathibhamol C P, Amala G S and Kapadia M 2016 September Anomaly detection based multi label classification using Association Rule Mining (ADMLCAR). In Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference IEEE pp. 2703-2707
[10] Velusamy K, Venkitaramanan, D Vijayaraju N, Suresh G and Madhu D 2013 Inverted indexing in big data using hadoop multiple node cluster International Journal of Advanced Computer Science and Applications (IJACSA), 4(11)
[11] Kavitha K R, Sandeep S, and Praveen P R 2016 December Improved spectral clustering using PCA based similarity measure on different Laplacian graphs. In Computational Intelligence and Computing Research (ICCCI) IEEE International Conference on pp. 1-6)
[12] Harikumar S and Thaha S S 2016 August MapReduce model for k-medoid clustering. In Data Science and Engineering (ICDSE) International Conference IEEE pp. 1-7
[13] Moseley B, Dasgupta A, Kumar R, and Sarlós T 2011 June On scheduling in map-reduce and flow-shops. In Proceedings of the twenty-third annual ACM symposium on Parallelism in algorithms and architectures pp. 289-298
[14] Yao Y, Wang J, Sheng B, Tan C C and Mi N 2017 Self-adjusting slot configurations for homogeneous and heterogeneous hadoop clusters IEEE Transactions on Cloud Computing 5(2) 344-357
[15] Liang F, and Lau F C 2015 May SMapReduce: Optimising resource allocation by managing working slots at runtime. In Parallel and Distributed Processing Symposium (IPDPS) IEEE International pp. 281-290

[16] Barbieru C, and Pop F 2016 March Soft real-time hadoop scheduler for big data processing in smart cities. In Advanced Information Networking and Applications (AINA) IEEE 30th International Conference pp. 863-870

[17] Tang S, Lee B S and He B 2014 Dynamicmr: A dynamic slot allocation optimization framework for mapreduce clusters IEEE Transactions on Cloud Computing 2(3) 333-347

[18] Kc K and Anyanwu K 2010 November Scheduling hadoop jobs to meet deadlines. In Cloud Computing Technology and Science (CloudCom) IEEE Second International Conference pp. 388-392

[19] Gao Y, Rong H, and Huang J Z 2005 Adaptive grid job scheduling with genetic algorithms. Future Generation Computer Systems 21(1) 151-161

[20] Page A J and Naughton T J 2005 Framework for task scheduling in heterogeneous distributed computing using genetic algorithms. Artificial Intelligence Review 24(3-4) 415-429.