International Journal of Engineering & Technology, 7 (4) (2018) 4227-4232

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet. v7i4.13941
Research paper

Performance evaluation and resource optimization of cloud based parallel Hadoop clusters with an intelligent scheduler.

Manishankar S [1] *, S. Sathayanarayana [2]

[1] Research Scholar, Department of Computer Science, Bharathiar University, Coimbatore, India
[2]Research Supervisor, Department of Computer Science, Bharathiar University, Coimbatore, India
**Corresponding author E-mail: manishankar1988@gmail.com*

Abstract

Data generated from real time information systems are always incremental in nature. Processing of such a huge incremental data in large scale requires a parallel processing system like Hadoop based cluster. Major challenge that arises in all cluster-based system is how efficiently the resources of the system can be used. The research carried out proposes a model architecture for Hadoop cluster with additional components integrated such as super node who manages the clusters computations and a mediation manager who does the performance monitoring and evaluation. Super node in the system is equipped with intelligent or adaptive scheduler that does the scheduling of the job with optimal resources. The scheduler is termed intelligent as it automatically decides which resource to be taken for which computation, with the help of a cross mapping of resource and job with a genetic algorithm which finds the best matching resource. The mediation node deploys ganglia a standard monitoring tool for Hadoop cluster to collect and record the performance parameters of the Hadoop cluster. The system over all does the scheduling of different jobs with optimal usage of resources thus achieving better efficiency compared to the native capacity scheduler in Hadoop. The system is deployed on top of OpenNebula Cloud environment for scalability.

Keywords: *Big Data; Hadoop; Parallel Processing; Intelligent Scheduler; Ganglia Monitor; Super Node; Mediation Manager.*

1. Introduction

Information system serves as a major source of information for many public as well as private sector organization. The information era has put forward a situation were all medium or large organization has adopted an information system of its own or an information system maintained that is maintained by a third-party provider. All these information systems handle a huge set of data including textual data and other multimedia data such as audios, videos, images and documents. In most of the cases these information systems were just a source of data or storage, not for data-based decision support. Concern that was in the mind of many of those information system designers and developers was how to handle the data processing operation without much performance degradation. Many platforms such as Hadoop, Spark, HIVE, HBase all help in handling the huge data also can be termed as Big Data[1]. All data set must be processed by assigning it as a computational processing job. Job scheduling is a mechanism which plays vital role in the system performance. Variety of scheduling algorithm such as First in First out(FiFo), Shortest Job First(SJF) were used by many initial information system managers to schedule the jobs efficiently. [2]. The job scheduling is a critical factor for any cluster-based information system which process huge data. Systems which handle information of size in Peta bytes if uses an algorithm like First come First serve will lead to a performance degradation in large scale [3]. Solution is to adopt some asymptotic strategies of job scheduling which can handles varied jobs and machines. Framing a mathematical model is necessary for optimal computational algorithm for job scheduling. The system is framed by considering a set of jobs which is computed under a single machine and then finding out which job is heavily loaded. Denoting the heavily loaded job in mediation manager and then calculating the best possible resource with which the job is carried forward. The scenario is then extended to cluster-based environment were each clusters highest and lowest job are being noted and then extending the resource as required by the job. A small- scale organization may not adopt such a solution which depends on such a hybrid approach which calculates each job efficiency and then each resource efficiency and extend the resource according to the job's computational requirement. Another common approach of handling cluster efficiency is use of genetic algorithms in job scheduling. Genetic Algorithm (GA) based approaches are carried forward by estimating the architectural model used to cluster and number of jobs that constitute the computational operation. Traditional Map-Reduce operation which is found in Hadoop systems support data processing. The work flow of Map-Reduce is preparing Map copy and then Reduce the data based on Map value. Decision making needed for job scheduling in a Hadoop based system is done with help of improved genetic algorithm. The algorithm is enhanced by considering parameters like map time, job size, cluster size, Execution time, Execution cost. [4]. GA formulation is always represented with percentage of chromosome, Chromosome representation is done as population which define the jobs to be considered and a fitness function is evaluated in the selection process of jobs and resources, which evolves the methodology used in the GA based job scheduling algorithm. The task selection process involves consolidation of the data, optimal values of selected by the genetic representors, match making and non-family scheduling approaches. [5]. Family graphs and analysis of non-family functions suits the computations in a cloud based simulated configuration resulting in an intensive reduce of turn-around time of jobs. Other scheduling approached based on cluster analysis are tested with resource availability. FIFO Scheduler, Fair Scheduler, Ideal

Scheduler, EDF-Preemptive works in hand to hand with native HDFS(Hadoop Distributed File system) scheduler[6]. All these kinds of the scheduling approaches has provided a wide spread influence in the efficient management of data clusters. The sudden growth of data in Information System(IS) of the organizations from GB to PB made all these approaches to popular to use in a Big Data environment. Even though handling of the huge incremental data has been made possible by the Big Data experts, still there exist a critical problem of performance degradation and resource under or over utilization which has made all the existing systems to be possibly slower than expected when computing a huge data. The solution that many designers had in mind was to increase the infrastructures and resources tremendously so that performance degradation can be efficiently handled, Even though the solution looks very much feasible but the cost incurred in such a setup turns out be very huge. Ideal computational systems really doesn't exist, but some effort have to be made to generate a nearly ideal system which handles performance degradation with help of the available resources only. The further parts of the paper talks about related works, proposed architecture for the Hadoop cluster, intelligent scheduler and results of performance improvement and resource optimization.

2. Related work and methodology

As the work is majorly on scheduling algorithms and computational platforms which process the data and balance the resources efficiently. Some of the related works on scheduling and computational platform are described below.

Hadoop is one such system which shoulders the huge Information system processing which can cater millions of people at a time [7] [2]. let us see some of the important scheduling algorithms which has been serving as a back bone for Hadoop based clusters.

Initially genetic algorithms played some vital roles known for their capability in handling gene data. Distributed and flexible scheduling approach was proposed based on job-shop which handle FMC(Flexible Manufacturing Cell)data ,which makes use of heuristic optimization technique, and introduce a new genetic algorithm for handling the distributed and flexible job scheduling in Hadoop[8], the job to FMU(Flexible Manufacturing units was done by considering population ,chromosome selection, cross over operator, mutation operator of genetic algorithm. Although the Genetic Algorithm used in Job-Shop has an upper hand compared to flexible job scheduling scheme, still the parallel execution of the genetic function and breaking function in to parallel blocks to meet the requirement of the incremental data was not considered in the research.

Hadoop schedulers are normally used to address various performance issues, each of the default scheduler available in Hadoop can address each of the performance metric in computation. But devising a single algorithm that handles all the scheduler activities is a ideal and impractical solution for Hadoop. FIFO is the most common or the default scheduler in Hadoop, FIFO basically in larger system like Hadoop can result in Performance degradation. Fair sharing scheduler uses a pool of map-reduce slot for each of the resource and each pool is shared among the users in such way that users can share resources equally over a time. COSHH(Classification and Optimization based Scheduler for Heterogeneous Hadoop systems) has a functionality classifying the input jobs to find the best possible matching resource class , it makes use of the Linear programming problem to find the best possible match. [9].

As cloud took over the storage profile for most of the information systems, private cloud came as solution for the workload balancing for the Hadoop based systems. This led to the development of Cloud based bench mark like Cloud Rank D, which has been implemented with many combination of scheduling algorithm supporting Hadoop, such as FIFO, Capacity, Naive fair sharing, Fair sharing with delay and Hadoop on demand scheduling. Cloud improves the performance of Hadoop system by 20%. [10]. Evaluation parameter considered for assessing the performance are Throughput, Data processed per second. The jobs were allotted to the Cloud based solution and the results indicated that computational time in Nano seconds(Ns) is improved. The analysis and operation performed give us a outlook that different algorithm has different types of influence in task scheduler, so choosing a right scheduler at right time improves Hadoop cluster performance. A lack of a dynamic approach in choosing the data, based on the data disparities were not taken in to consideration for the cloud-based scheduler.

The system heterogeneity is one of the key thing that needs consideration while operating with large scale data, which is parallelly computed in cloud environment. Cluster size, user profile, load of work carried out are some the important factor affecting the Hadoop system performance. Various type of Hadoop cluster systems were considered for research like homogenous small, homogenous large, heterogenous small and heterogenous large. Schedulers were tested to find the efficiency in various cluster systems and a guideline was prepared to choose schedulers in different possible situations and application that can be considered in Hadoop cluster.[11]

Cloud handles the data intensive operational like Map-Reduce which support various type of real time application[12].Improving the computational acceleration for computational intensive operations require a GPU based processing platform. CUDA (Compute Unified device Architecture) the parallel computing platform and programming model helps to utilize the GPU resources efficiently . GPU energy efficiency is a key factor in reduced power consumption for Hadoop based cluster systems. Few of the most complimenting approach for GPU's are JCUDA (Java bindings for Compute Unified device Architecture), JNI(Java Native Interface), Hadoop streaming and pipes. Comparison of CUDA GPU and JCUDA with respect to the Hadoop approach and Map-Reduce task gives an idea of how processing can be made efficient in the Hadoop cluster. This study paves a light on how to integrate the most popular programming models to Hadoop. [13]. All though comparative studies has provided many computational approaches supporting the usage of computational intensive calculation, but for data intensive application how to effectively supply GPU resources is not addressed in existing research which requires a two-level Map-Reduce system.

Hadoop++ is one of the frame work of improvement which outperforms both Hadoop and Hadoop DB as well. Hadoop ++ contributes many features to the frame work without bringing in much of the architectural change. These additions are Hadoop plan, Trojan index and Trojan Join. Planning phase has Data load phase, Map phase, Shuffle phase and reduce phase. Index creation, Data co-partition, Query processing and bench marking tasks are also deployed in the existing model. [14].

The important knowledge acquired in this research is that most of the benefits with regards to performance is obtained with help of schema knowledge and anticipation of the data load and query processing time. This kind of approach could be integrated with a Hadoop cluster to improve the operation of data intensive applications. Map-Reduce is though very much performance centric there is still a rigidity in the data flow that is taking place which is actually hindrance to the parallel processing, one of the research idea that has been proposed is bringing an iterative flow to the data .IMap-Reduce(Iterative Map-Reduce) implements an asynchronous flow by providing the same subset of data (chunk) for the map phase as well as reduce phase in a computational job to process iteratively. Twister another approach is included that supplies memory at run time and communication based on message. Map-Iterative fills the result of last reducer into successive iterators. Hadoop online prototyping approach involves a parallel set of map-reduces beyond the normal batch processing of jobs. Thus improving the total rate of pipelining that occurs.[15]. Co-Hadoop a lightweight extension of Hadoop that allows applications to control where data are stored. In contrast to previous approaches, Co-Hadoop retains the flexibility of Hadoop in that it does not require users to convert their data to a certain format thus providing extendibility. Auto parameter tuning, communication optimization, energy efficiency optimization parameters are considered for the platform and a survey is then carried forward I comparison to latest technologies like spark, Hive, SQL-

Like, SCOPE (Separation of code and Parallel Environment) and Dremel. The comparative study results show that scalability is one of the issues that arises when handling a huge data with parallel approaches. So high performance computing, large scale machine learning can be included in the research which assures massive parallel processing [16].

In Hadoop technology name node plays a vital role in deciding the architecture used in storing data. Improving the Name node by aggregating function is a solution for using varies data types. [17]. Aggregator the functionality is responsible for assigning the jobs and data between nodes. As the aggregator plays the vital role the chance for single point failure is avoided. Here Data node is being tracked by the aggregator instead of the Name node in normal architecture. Thus, overall work that is done by the Name node is minimal in this architecture. Performance of the system can be increased by adding multiple aggregator nodes at various levels. A regression based method with assigning data values based on aggregator before Map-reduce can be a solution to the optimize resource in Hadoop.[18]. In smart grid optimization technique a method is used by choosing most similar nodes N, where N refers to the number of most similar nodes present in a cluster as the paradigm for Map-reduce. Similarity is in terms of computational efficiency. So there is an improvement in computational operations in Big Data environment [19]. New generation mobile network is one of the biggest source of the data generated. Optimizing the network bandwidth by applying big data handling technique is essential[20].Grid based solutions with an adaptive scheduler with log information is also deployed which has proven to be a computational strong scheduling scheme against scheduling faults, the applicability of the scheduler in terms of data operations has be included in the literature study[21]. The study of the literature has paved a way to develop a improved capacity scheduler for Hadoop cluster.

Intelligent Scheduler based Parallel Data Processing in Information System.

The detailed review through many of the research that was carried forward in the field of Big data-based Hadoop clusters and various technologies associated gives us a strong hint that handling information system which process huge data always require a well-equipped data processing and maintenance engine. Information system as a combination of multiple platformed application like cloud, mobile, people, big data, social media which can be also termed as Big 5 in information system language. The major objective of any information system manager lies in processing the data collected and disseminate proper information with minimum energy and improved performance. Achieving this objective requires an intelligent scheduler which can handle the incoming jobs. Especially incase of Hadoop where it has the Map-Reduce operations to be reduced by identifying the number of parallel maps or reduces to be performed, for making the system better. In this research, an intelligent scheduler that is a combination of multiple intermediate scheduler and data discretizing function. The requirement for a multi cluster platform arises to efficiently manage an organization which is split logically or spatially divided with in the allocated infrastructure or spread even in different cross functional modules. The system architecture has a hierarchical ordering where each cluster head of the group is considered again as workstation by the super node which performs the integration of the tasks which occur in multi spread platforms. Even a cluster itself is believed to be heterogenous in nature, so that the complexity of achieving the performance with varying configuration systems becomes increasingly hard.

Figure1 depicts the architecture of the parallel data processing system which can handle the storage nodes with help of a super node, which acts a vital support to the inherent scheduler present in the Cluster environment in handling the request that arrive from parallel jobs that are running in cluster. The YARN (Yet Another Resource Negotiator) updates the super node with details of job running and the load on the cluster. The resources depicted in the figure are computational resources, and storage refers to the distributed data stores where the processed data has to be stored or data node of HDFS.

In the proposed model the computation and processing of data is taken care by super node and mediation manager. An

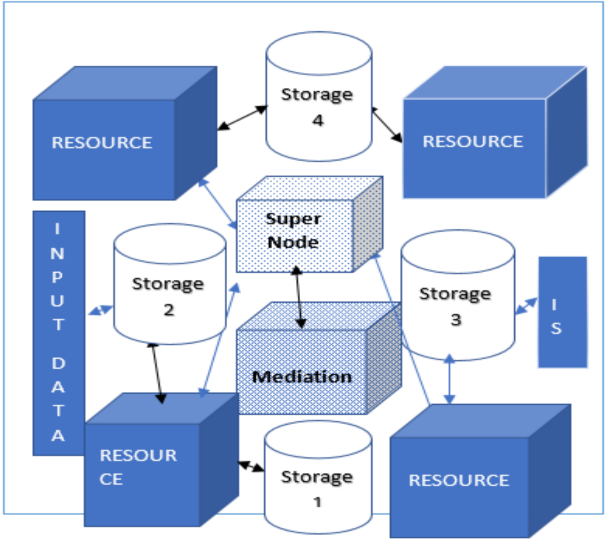


Fig. 1: Architecture of the Proposed System.

Incoming job will be first analyzed based on the parameters obtained from the job request by the user. The super node has an inbuilt intelligent scheduler supported with cross over mutation technique which maps the incoming job to the available resource. The inference obtained from the intelligent scheduler plays a vital role in optimizing the resource demand. The intelligent scheduler initially takes an assumption that there is an average load in each of the resources with help of a queuing technique. The mediation manager calculates the load of the cluster with the help of ganglia monitor and updates the observations with the scheduler in the super node. The scheduler then checks the available resource and then categorize the available resources in to idle resource, average utilized resource and heavily loaded. Values obtained from the mediation is then kept in to a resource table to match the upcoming resource request. A selection process of the genetic algorithm is deployed in choosing the parameter required for job and resource to be scheduled, a fitness value is generated for the jobs as well as resource based on dataset from resource table, cross over mutation technique is used to match the incoming job to the optimal resource available in the list. Thus, for all incoming jobs the best possible computational resource will be allocated, this approach provides an efficient way to optimize the resource as the intelligent scheduler make sure that there is no under or over utilization of the resource-taking place.

The scheduler calculates the load of the resource, categorize the resources based on the load, jobs are also profiled and mapping of job to resource is done.

Input: list of input Jobs – J, list of Resources- R

Job and Resource Parameters -CPU utilization, memory utilization and Load.

Outcome: assigning a job from list J to Resource in list R.

Step1: initially calculate the number of Resource and assign the value to R.

Step 2: Assign a Poisson distribution to calculate the initial probability of the resource being assigned a job.

A set of resources 0,1,2 .with jobs assigned, and average number of jobs assigned is λ which is the rate of load of each resource. The probability of assigning k jobs to a resource in a given time is

$$p(Kjobs) = e^{\lambda} \frac{\lambda^{k}}{k!}$$

This probability is the parameter to estimate the resource availability.

The capacity of the job is measured mathematically as

$Cj = p\ (kjobs) + tr + \frac{d}{\lambda}$

Capacity of each resource Ci and probability of assigning job and tr the time duration for resource is assigned and d is the data rate and λ is the arrival rate.

Step 3: based on the probability the average load of the resource is calculated and assumptions are made.

Step 4: calculate resource and job load parameter from the cluster monitor in mediation with the help of ganglia.

Step 5: recorded parameters are stored in to super node for the scheduler to access.

Step 6: Categorizing the resource based on the parameters in to idle, average loaded and heavily loaded.

Step 7; Categorizing the jobs based on job type and load into low, medium and heavy.

Step 8: Mapping of resource job to resource based on Cross Over mutation.

Cross over is done for the resource and job to find the best optimal resource matching the incoming job.

The resource is marked as optimal and the default YARN scheduler is notified.

Step 9: The Resource manager allocated the selected resource to the incoming job

Step 10: the load of the cluster calculated by mediation for larger time duration

Step 11: visualize the Resource utilization and optimization.

Step 12: continue the process until the job request are over.

The algorithm is termed intelligent as it automatically calculates the load probability of the resource as well as categorize the resource and job based on parameter observed and finds the optimal resource for incoming job and assigns. Presence of intelligent scheduler algorithm improvises the Yarn Capacity scheduler. Normal scheduler with in YARN assigns job resource based on the job size alone this leads to over utilization or underutilization of resource. The scheduler in the proposed system continuously monitor resource utilization and does job scheduling based on the observation. This results in resource optimization thus the proposed scheduler is better than default YARN scheduler. The implementation and result in the next part supports the assertions made about the proposed system.

3. Results and discussion

The proposed research work was carried out in a real time Hadoop cluster environment, setup with help of private IAAS cloud with an Open Nebula Cloud server. The multi node Hadoop cluster was setup in a power full cloud VM environment with data nodes, name nodes and resource manager to support the distributed data processing. The Hadoop clusters was monitored with help of ganglia and mediation manager. The mediation manager recorded and stored values of cluster such as memory, resource, CPU utilization and load.

The jobs are initial tested in the Hadoop cluster and the values of CPU utilization, memory utilization, load on the cluster, job id , resource id and job type are recorded in the mediation in form a table as depicted below in Table 1. These results are the basis for the decision making and categorization of jobs, the super node with help of intelligent scheduler algorithm apply the average load and testing is done. Later the jobs are categorized in to idle, medium and heavily loaded based on the cluster values obtained. This classification done with help of the decision tree algorithm helps in choosing the optimal resource.

The decision tree-based algorithm is used to classify the resources and jobs based on the results depicted in Table 1 The jobs are classified in to high, low, very high, very low and medium based on various parameters like load, memory and sever load. Different types of job like Word count, Number sum, real time shop data and ecommerce data are some jobs that were tested in the Hadoop cluster

Table 1: Job and Resource Performance Obtained from Mediation.

Job Id	R Id	Job Type	Execution Time	Memory	Resource Status	CPU	Load
1	1	H	1.4540000	1971053909	L	4.833333	4.0200
2	2	L	0.2973333	25346068	L	5.666667	2.6000
3	1	L	0.9513333	25023829	L	6.173333	5.7000
4	1	H	1.9890000	27986542	L	5.232333	4.4000
5	3	L	0.3433332	26789979	L	5.723113	3.3000
6	1	L	0.9800000	27000019	L	6.249846	6.7000
7	4	H	2.0100000	30023126	L	5.751234	5.6000
8	2	L	0.4532234	27651239	L	5.956417	4.1000
9	1	L	1.0200000	27609862	L	6.963095	7.8000
10	5	H	1.3900000	438457132	L	6.340000	3.8068

The Hadoop cluster show cased different behavior based on load and thus the job like Word Count classified in to Low and Jobs like Number sum and shop data were termed medium or Very High depending on input size and real time ecommerce processing was termed VL

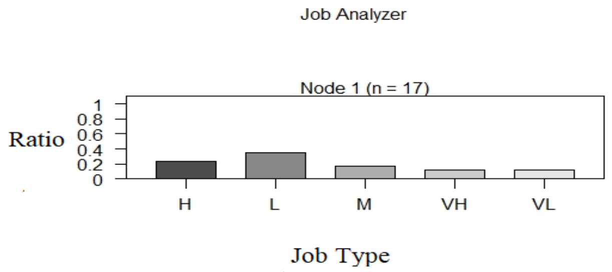


Fig. 2: Categorization of Jobs into Various Level Sets.

The figure 2 depicts job analyzer categorizes the job into various level sets based on the load execution time, cpu utilization and memory. The data was obtained from ganglia do the categorization and with the help of a decision tree algorithm job analyzer which divides the job as depicted in the diagram above. Our training data set consist of 21 various types of jobs based on different inputs.

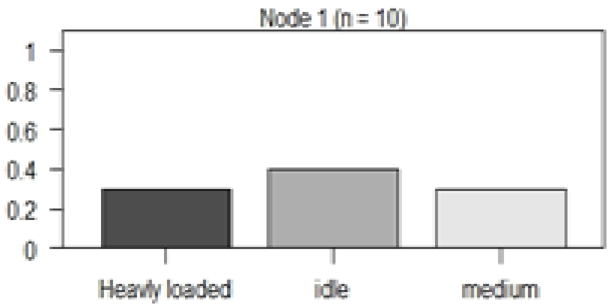
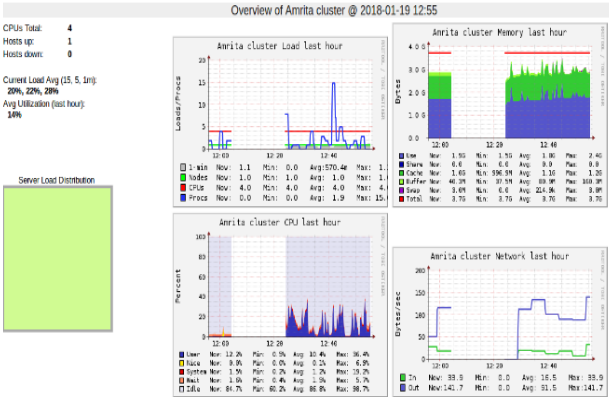


Fig. 3: Categorization of Resources in to Various Levels.

Figure 3 represents the classification of resource based on the results of queuing model, such as calculated probability, arrival rate of job and capacity of resource as depicted in the formula in scheduler. The observed values of the resources are fed in to the super node which performs the intelligent scheduling that assists the capacity scheduler. Mediation node records the CPU utilization, memory utilization and network utilization as depicted in Figure 4



Overview of Amrita cluster @ 2018-01-19 12:55

CPUs Total: 4
Hosts up: 1
Hosts down: 0

Current Load Avg (15, 5, 1m):
20%, 22%, 28%
Avg Utilization (last hour):
14%

Server Load Distribution
Amrita cluster Load last hour
Loads/Procs
Amrita cluster Memory last hour
Bytes
Amrita cluster CPU last hour
Percent
Amrita cluster Network last hour
Bytes/sec
12:00
12:20
12:40

Fig. 4: Ganglia Monitor Recording Values for the Mediation Node.

The YARN the resource manager present in Hadoop cluster is responsible for the job allocation. As YARN has three schedulers for assigning job FIFO, Fare schedule and Capacity scheduler. The resource and job matching, and categorization helps the schedulers allocate the job with optimal number of resources. The intelligent scheduler proposed does the mapping of the resource in exact way thus obtaining a lesser load and less resource utilization to Hadoop cluster. Cross Mutation of genetic algorithm is used in the approach to find the exact matching resource available. The super node which has the intelligent scheduler a modified version of the capacity scheduler assigns the job initially to the resources in idle state, if there is node available in idle state it checks for resources in medium load state, finally only it checks the heavily loaded resources based on the availability and capacity. Thus, the intelligence inside the scheduler plays a vital role in resources management.

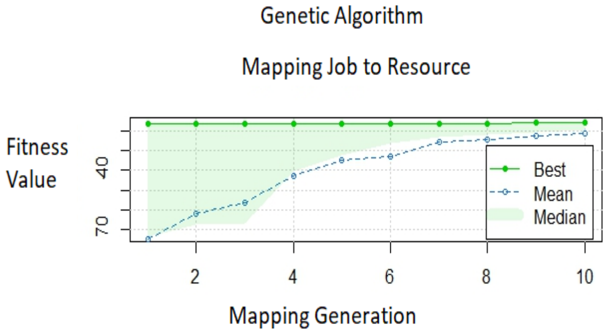
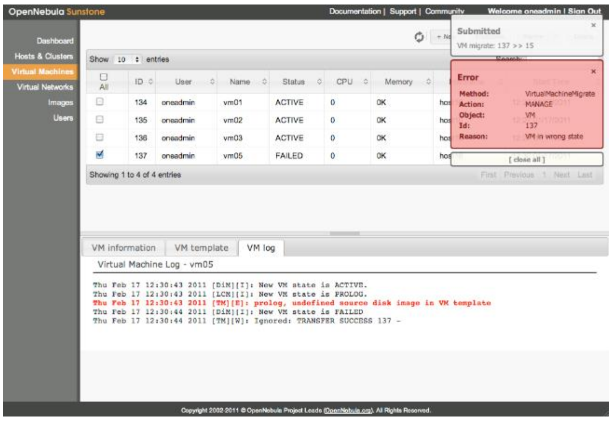


Fig. 5: Genetic Algorithm Plot.

The figure 5 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 is depicted in plot. The cross-mutation results give the exact resource that is matching to the job type.



Submitted
VM migrate: 137 >> 15

Error
Method: VirtualMachineMigrate
Action: MANAGE
Object: VM
Id: 137
Reason: VM in wrong state

□ All	ID	User	Name	Status	CPU	Memory
□	134	oneadmin	vm01	ACTIVE	0	0K
□	135	oneadmin	vm02	ACTIVE	0	0K
□	136	oneadmin	vm03	ACTIVE	0	0K
☑	137	oneadmin	vm05	FAILED	0	0K

Showing 1 to 4 of 4 entries

Virtual Machine Log - vm05

```
Thu Feb 17 12:30:43 2011 [DiM][I]: New VM state is ACTIVE.
Thu Feb 17 12:30:43 2011 [LCM][I]: New VM state is PROLOG.
Thu Feb 17 12:30:43 2011 [TM][E]: prolog, undefined source disk image in VM template
Thu Feb 17 12:30:44 2011 [DiM][I]: New VM state is FAILED
Thu Feb 17 12:30:44 2011 [TM][W]: Ignored: TRANSFER SUCCESS 137 -
```

Copyright 2002-2011 © OpenNebula Project Leads (OpenNebula.org). All Rights Reserved.

Fig. 6: Result Of Job Allocated to the Resource.

The effect of cross mutation helps in allocating the exact resource as depicted in the monitor in Fig 6. Thus, the result show that there is considerable improvement in the number of resources that are effectively utilized in the cloud. The results from the cluster monitor, mediation manger before and after the intelligent scheduler is incorporated is taken, and the results are put in the form of a graph depicted in Figure 7. The resource utilization is taken as the aspect for comparison of the existing capacity scheduler in YARN and the intelligent scheduler, the graph is generated based on the count of the resource used in various types of job by the native capacity scheduler and the improvised intelligent scheduler. When number of jobs in X axis are less the number of resource utilized in Y axis remains same for both the schedulers, but as the load increases it is found that native scheduler used more jobs compared to proposed scheduler.

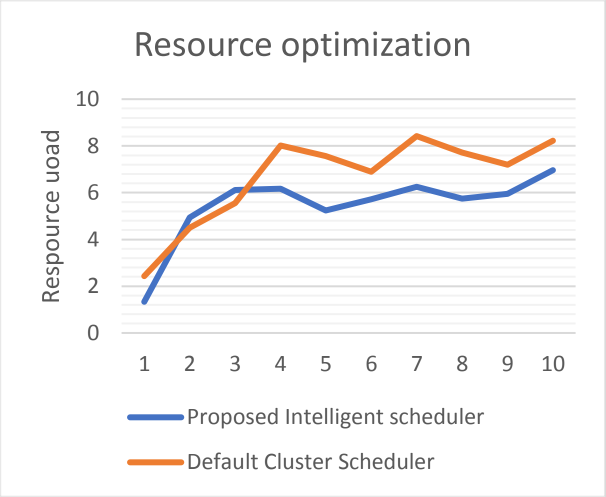


Fig. 7: Visualization of Resource Optimization.

The resource optimization is in terms of utilization of Cluster resource for data processing which is evidently visible in the result. The result thus significantly show case an improvement in the resource optimization in cluster. The proposed system with the presence of intelligent scheduler, super node and mediation manager bring about a performance improvement for big data processing in Hadoop cluster environment. Hadoop cluster with the help of the proposed system can be maintained with lesser resources or all the resources in the cluster being utilized efficiently.

4. 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. So, this work aims to provide a platform which can handle the heterogeneous data with available infrastructure and resources by in incorporating intelligent scheduler, super node and mediation manager. Resource and jobs are classified by the super node with the help of that data collected from mediation manager with help of ganglia. The intelligent scheduler uses a cross mutation genetic algorithm to map the job to the best optimal resource available. The system does efficient management of resources and balances the load of the Hadoop cluster. The proposed model provides an amble opportunity for the cluster to manage jobs of varying load with optimized number of resources. The future enhancement to the system can be done by using virtualized resources to manage the Hadoop cluster. An efficient scalable cloud-based platform with the proposed intelligent scheduler is the future research to be carried out.

References

[1] J. Eckroth, “Teaching Future Big Data Analysts : Curriculum and Experience Report,” 2017.

[2] J. V Gautam, H. B. Prajapati, V. K. Dabhi, and S. Chaudhary, “A survey on job scheduling algorithms in Big data processing,” 2015 IEEE Int. Conference. Electronic. Computer. Communication. Technol., pp. 1–11, 2015.

[3] A. Sfrent and F. Pop, “Asymptotic scheduling for many task computing in Big Data platforms,” Inf. Sci. (Ny). vol. 319, pp. 71–91, 2015. https://doi.org/10.1016/j.ins.2015.03.053.

[4] Q. Lu, S. Li, W. Zhang, and L. Zhang, “A genetic algorithm-based job scheduling model for big data analytics,” Eurasip J. Wireless. Communication. Network. vol. 2016, no. 1, 2016. https://doi.org/10.1186/s13638-016-0651-z.

[5] R. Kune, P. K. Konugurthi, A. Agarwal, R. R. Chillarige, and R. Buyya, “Genetic Algorithm Based Data-Aware Group Scheduling for Big Data Clouds,” in Proceedings - 2014 International Symposium on Big Data Computing, BDC 2014, 2015, pp. 96–104.

[6] D. Cheng, J. Rao, C. Jiang, and X. Zhou, “Resource and Deadline-Aware Job Scheduling in Dynamic Hadoop Clusters,” in Proceedings - 2015 IEEE 29th International Parallel and Distributed Processing Symposium, IPDPS 2015, 2015, pp. 956–965.

[7] D. Jiang, B. Ooi, L. Shi, and S. Wu, “Big Data Processing Using Hadoop: Survey on Scheduling,” Proc. VLDB Endow. vol. 3, no. 10, pp. 272–277, 2010.

[8] L. De Giovanni and F. Pezzella, “An Improved Genetic Algorithm for the Distributed and Flexible Job-shop Scheduling problem,” European. Journal. Operaton. Research. vol. 200, no. 2, pp. 395–408, 2010. https://doi.org/10.1016/j.ejor.2009.01.008.

[9] A. Rasooli and D. G. Down, “A hybrid scheduling approach for scalable heterogeneous hadoop systems,” in Proceedings - 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, SCC 2012, 2012, pp. 1284–1291.

[10] S. Liu, J. Xu, Z. Liu, and X. Liu, “Evaluating task scheduling in hadoop-based cloud systems,” in Proceedings - 2013 IEEE International Conference on Big Data, Big Data 2013, 2013, pp. 47–53. https://doi.org/10.1109/BigData.2013.6691697.

[11] A. Rasooli and D. G. Down, “Guidelines for Selecting Hadoop Schedulers Based on System Heterogeneity,” J. Grid Comput. vol. 12, no. 3, pp. 499–519, 2014. https://doi.org/10.1007/s10723-014-9299-2.

[12] D. Ding, F. Dong, and J. Luo, “Multi-Q: Multiple Queries Optimization Based on MapReduce in Cloud,” Proc. - 2014 2nd Int. Conf. Adv. Cloud Big Data, CBD 2014, pp. 100–107, 2015.

[13] J. Zhu, J. Li, E. Hardesty, H. Jiang, and K. C. Li, “GPU-in-Hadoop: Enabling MapReduce across distributed heterogeneous platforms,” in 2014 IEEE/ACIS 13th International Conference on Computer and Information Science, ICIS 2014 - Proceedings, 2014, pp. 321–326.

[14] J. Dittrich, J.-A. Quiané-Ruiz, A. Jindal, Y. Kargin, V. Setty, and J. Schad, “Hadoop++: Making a yellow elephant run like a cheetah (without it even noticing),” Proc. VLDB Endow., vol. 3, no. 1–2, pp. 515–529, 2010. https://doi.org/10.14778/1920841.1920908.

[15] Y. Zhang et al., “Parallel Processing Systems for Big Data: A Survey,” Proc. IEEE, vol. 104, no. 11, pp. 2114–2136, 2016. https://doi.org/10.1109/JPROC.2016.2591592.

[16] A. Alexandrov et al., “Massively Parallel Data Analysis with PACTs on Nephele,” Proc. 36th International. Conference on. Very Large Data Bases, pp. 1625–1628, 2010. https://doi.org/10.14778/1920841.1921056.

[17] B. Jena, M. K. Gourisaria, S. S. Rautaray, and M. Pandey, “Improvising Name Node Performance by Aggregator Aided HADOOP Framework,” pp. 382–388, 2016. https://doi.org/10.1109/IC-CICCT.2016.7987978.

[18] X. Wu, “A MapReduce Optimization Method on Hadoop Cluster,” Proc. - 2015 Int. Conf. Ind. Informatics - Comput. Technol. Intell. Technol. Ind. Inf. Integr. ICIICII 2015, pp. 18–21, 2016.

[19] A. Vaccaro, L. Troiano, A. Vaccaro, and M. C. Vitelli, “On-line smart grids optimization by case-based reasoning on big data On-line Smart Grids Optimization by Case-Based Reasoning on Big Data,” no. September 2016.

[20] A. Ramaprasath, A. Srinivasan, and C.-H. Lung, “Performance optimization of big data in mobile networks,” 2015 IEEE 28th Can. Conference. Electrical. Computer. Engineering. Vol. 2015–June, no. June, pp. 1364–1368, 2015. https://doi.org/10.1109/CCECE.2015.7129477.

[21] S. Gokuldev and R. Radhakrishnan, “An adaptive job scheduling with efficient fault tolerance strategy in computational grid,”International. Journal of Engineering. Technology. Vol. 6, No.4, pp (1793-1798), 2014.