Optimised Method of Resource Allocation for Hadoop on Cloud

Shilpitha swarna
Student (M-Tech) Department of Computer science & Engineering
Sai Vidya Institute of Technology, Bangalore,
Bangalore, 560012, India
Shilpitha.swarna@gmail.com

Amogh Pramod Kulkarni
Assistant Professor, Department of Information Science & Engineering \
Sai Vidya Institute of Technology, Bangalore,
Bangalore, 560012, India
Amogh.pk@saividya.ac.in

Abstract**—** Many case studies have proved that the data generated at industries and academia are growing rapidly, which are difficult to store using existing database system. Due to the usage of internet many applications are created and has helped many industries such as finance, health care etc, which are also the source of producing massive data. The smart grid is a technology which delivers energy in an optimal manner, phasor measurement unit (PMU) installed in smart grid is used to check the critical power paths and also generate massive sample data. Using parallel detrending fluctuation analysis algorithm (PDFA) fast detection of events from PMU samples are made. Storing and analyzing the events are made easy using MapReduce model, hadoop is an open source implemented MapReduce framework. Many cloud service providers (CSP) are extending their service for Hadoop which makes easy for user's to run their hadoop application on cloud. The major task is, it is users responsibility to estimate the time and resources required to complete the job within deadlines. In this paper, machine learning techniquies such as local weighted linear regression and the parallel glowworm swarm optimization (GSO) algorithm are used to estimate the resource and job completion time.

Keyword*: Hadoop, parallel fluctuation detrending analysis, machine learning, cloud computing, resource allocation***.**

*****

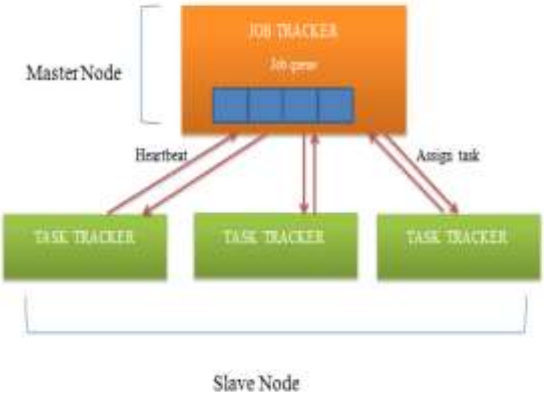
I.INTRODUCTION

In today's era, using internet of things data is collected from various physical devices over the internet. Phasor measurement unit (PMU) device installed in smart grid generate high data, reporting rates present major computational challenges in the requirement to process potentially massive volumes of data. Fast algorithms capable of processing massive volumes of data are now required in the field of power systems. Parallel detrending fluctuation analysis is an approach for fast event detection on massive volumes of PMU data. Cloud computing is a process of utilizing the computer resources which is available over the internet in order to execute the job. Maintaining the hardware and software infrastructure is a tedious job, which is handled effectively by the cloud service provider following rent policy for the instance.

Advantage of cloud computing:

- Scalable: data can scale to 100's of the node.
- Pay only for the resource used and the duration it was used.
- Easy to set up.

Hadoop is an open source project by apache software foundation. Hadoop is widely used in industries and academia community for its remarkable properties like automatically distributing the data [1], scalability, availability and fault-tolerance



Hadoop distributed file system

Hadoop main components are 1. Hadoop distributed file system which contains of one name node that maintains the metadata of the complete cluster. The data in the datanode is replicated at the rate of 3 for fault- tolerance. 2. MapReduce framework, in which it executes the job by assigning it to the map and reduce function. The data-type of MapReduce programming model is key-value records.

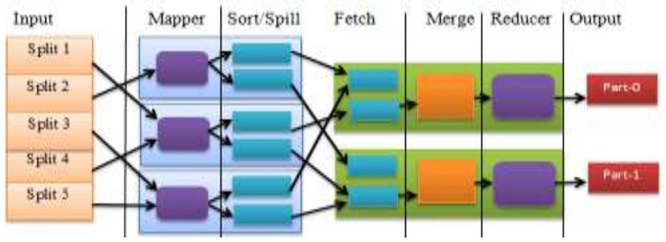
Map function:

$(K_{in}, V_{in}) \rightarrow$ list (K_{inter}, V_{inter}).

Reduce function:

$(K_{inter},$ list $(V_{inter})) \rightarrow$ list (K_{out}, V_{out}).

Many non-luxury industries cannot afford to use private cloud service hence, they use public cloud which is flexible based on the demand. Amazon EMR is one such cloud service provider [2].



Working of MapReduce

Scheduling the workload for the job is necessary to improve the performance of the job. Task for execution are not allocated at once, the input data is broken into blocks in order to execute parallely on the hardware resource and certain assumption is made that is, each block would roughly take the same duration.
Several Hadoop scheduling methods are present, to share the workload of job among cluster nodes. The default scheduler is First to come first serve (FCFS) in which all available resource are given to the first job [3]. Hadoop on demand (HOD) is a private cluster, maintaining the private MapReduce environment was challenging and it also violated the rules of locality [4]. Fair scheduling is one in which fair allocation of resource takes place; it overcomes the drawback of FCFS in which the short job has to wait long time for resource [5]. The capacity scheduler is similar to the fair scheduler but used at organization, according to which the queue in the cluster are assigned with a capacity [6].
The Hadoop includes 3 main phase namely: map, shuffle and reduce. X.Lin [7] explains the cost based scheduling method which effectively allocates the cloud instances. According to which it only considers map and reduce phase, job execution in multiple waves was not considered. In multiple waves, the first wave of the shuffle gets executed parallelly with the map phase and another wave of the shuffle phase gets sequential executed after the map phase. Hence, consideration of shuffle phase is necessary. Virajith et al [8] presents bazaar model which estimates the resources allocation, [9] presents principle of dynamic allocation of resource, in both [8, 9] overlapping and non-overlapping are not considered.
[10] In this paper K-means algorithm was proposed for large scale data which was used to compute the average centroid weights assigned on both map and reduce function. [11,12] Presents MapReduce model which solves the problem related to co-clustering and fast clustering. [13] Presents Hadoop model in which ant-colony approach (ACO) is used, which divides the input data into many clusters. The ACO is mainly used for discrete problems and it is restricted to the sensing agent. [14] The particle swarm optimization (PSO) algorithm is one in which the swarm movements are dependent on historical positions and are restricted for models containing numerical value. [15] The glow worm optimization (GSO) is used for the continuous dataset and it is used to find multiple optimal solutions for both equal and unequal values.

In this work, both local weighted linear regression model and glow-worm optimization approach is used to estimate the resource and job completion time.

II.RELATED WORK

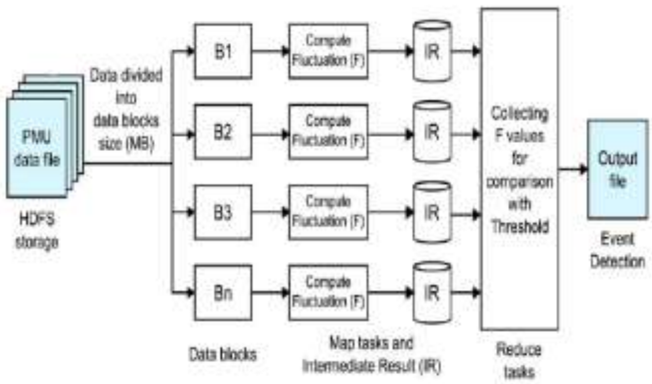
Several research works are carried in order to focus on the performance optimization for MapReduce jobs.

- According to Chenet [16] proposed model considers both sequential and parallel processing, which aims at reducing the cost of the product and time but the reduce slot is not changed with the increase in data set i.e word count and sort, which is arguable in practice.
- Starfish [17] is a self-tuning model that uses the historical job profile information to estimate the duration of the job. The virtual profile provides predicted timings and flow of data view, this adds overhead to estimate the job execution time.
- Morton et, al [18] proposed a model to estimate the performance which considers the execution duration of both map and reduce as same but in practice both map and reduce differ in their own factor and this paper does not consider the shuffle phase.

All these related studies are paired to our study and our method of approach can be incorporated into these MapReduce frameworks.

III.PARALLEL DETRENDED FLUCTUATION ANALYSIS

PDFA is an analysis used for fast detecting the samples from PMU or synchronizer configured window. At Great Britain, PMU device are installed on the smart grid and samples are obtained. The basic frequency considered at Great Britain is 50Hz. If the data obtained from PMU device is processed and analyzed well then it will help to make an efficient use of smart grid by overcoming the drawback of the existing smart grid system like reliability and security. In this paper [19] it deals with FIR (fault impulse response) which is a filter used to measure the PMU data for steady-state value. [20] Presents MPI (message passing interface) based clustering which parallelizes the computing process. [21] In this paper real time PMU data is stored on the cloud because of scalability. Both [20, 21] paper fails to provide fault-tolerance, in cloud fault-tolerance is not provided to the node. In case of node failure, no proper guarantee is provided to assign the running computation task for another node. To overcome above problem MapReduce is used because of its scalability, reliability and fault-tolerance property.



PDFA architecture

$$F(n) = \sqrt{\frac{1}{n}\sum_{k=1}^{n}[e(k)]^2} \quad (1)$$

In PDFA fluctuation (F) is computed for every window configured size, normally its 50 samples. Where n is the configured window size, k is the count of samples and also considers the signal of detrending. In MapReduce model, the fluctuation (F) value is computed in the map phase and the result is compared with the threshold value in the reduce phase. The signal of the high resolution taken from transients is used to analysis the changes that takes place in a short period of time. Any data-intensive application that gets executed on MapReduce model require resource (map and reduce slots) to perform the task. Amazon EMR service is used for Hadoop application. In this work, the MapReduce method is created to calculate the fluctuation of PMU dataset using PDFA approach, resource for this job is estimated using two different techniques called local weighted linear regression (LWLR) and glowworm swarm optimization (GSO).

IV.PROPOSED EMARS MODEL

As the input dataset increase the map task correspondingly increase, if reduce task is kept constant by user's configuration then the volume of intermediate data linearly increase by which duration of executing the reduce phase will also increase. Hence in this proposed system reduce task is kept varying.

A. local weighted linear regression

LWLR is a non-parametric function which considers weight to the instance. The required training data is collected from job profile and the values are considered as rows of the matrix.

Pseudo-code for LWLR

1: Read the values of job size *(S)*, the number of map (N_{map}), number of reduce (N_{reduce}) and execution time *(T)* of the job from the log file and create matrix *M*.
2: Create a Logistic Regression model with the sum of squares estimation between the dependent variable (N_{map}) number of map and independent variable job size and execution time.
3: Create a Logistic Regression model with the sum of squares estimation between the dependent variable (N_{reduce}) number of reduce and independent variable job size and execution time.
4: Find the weight for independent variable, job size and execution time for both MAP and REDUCE models.
5: The number of map is estimated as
$N_{map} = X + \beta 1 * S + \beta 2 * T$ where $X, \beta 1, \beta 2$ are the ω weights learnt from LR map Model.
6: The number of reduce is estimated as
$N_{reduce} = X + \beta 1 * S + \beta 2 * T$ where R2, B1, B2 are the w_j weights learnt from LR Reduce Model.

C. parallel glow worm optimization (GSO)

GSO belongs to swarm intelligence in which natural swarm is considered. A glowworm that emits more light (high luciferin level) means that it is closer to actual position and has a high objective function value. A glowworm is attracted by other glowworms whose luciferin level is higher than its own, within the local decision range. If the glowworm finds some neighbors with higher luciferin level within its local range, the glowworm moves towards them. At the end of the process, most glowworms will be gathered at the multiple peak locations in the search space.

Pseudo-code for GSO

1. Read the values of job size(*S*), a number of map (N_{map}), reduce (N_{reduce}) and execution time (*T*) of the job from the log file.
2. Configure number of parallel glow.
3. Each glow estimates the fitness function between the number of map v/s job size and execution time, the number of reduce v/s job size and execution time.
4. At each iteration of glow worm the best fitness function is taken and all worms try to optimize on this fitness function. i.e Lf = (1-p)*Lf + g*Objective.
5. This is repeated until max number of iteration.
6. Two fitness functions one for map and one for reduce are returned.

V.PERFORMANCE EVALUATION

A. Experiment result

The experiment conducted uses the historical job profile information which is obtained by running the job several times with varying job size. Local weighted linear regression model uses job profile information in order to estimate map and reduce slots. Parallel glow-worm uses high luciferin light level to estimate map and reduce slots. Each model would generate map and reduce slots which would be configured to run the input dataset.

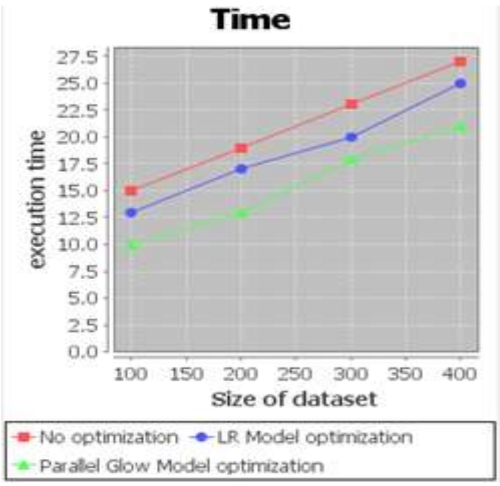


Figure Line chart for PMU dataset

The figure represents the comparison result with no optimization model (default values), LWLR, and parallel glow worm model.

CONCLUSION

The proposed model contains comparison result which shows that machine learning algorithm (which learn from the past experience) provide an efficient result than compared to the default configuration. Hence, optimal results are obtained. Currently in EMARS model it considers task with dependencies, without dependency can be the future work and currently only the major phases are consisdered (map-shuffle-reduce) in later case sub-phases can also be considered.

ACKNOWLEDGEMENT

I thank all the faculty members of Department of Computer Science of SVIT and also thanks to the management of SVIT, Bengaluru for providing research environment at the institute.

REFERENCE

[1] "Apache Hadoop," *Apache*. [Online]. Available: http://hadoop.apache.org/. [Accessed: 18-Feb-2015].
[2] B. T. Rao and L. S. S. Reddy, "Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments," *Int. J. Comput. Appl.*, vol. 34, no. 9, p. 5, Jul. 2011.
[3] Scheduling in Hadoop," *IBM- Developer work*. [Online]. Available: http://www.ibm.com/developerworks/library/os-hadoop-scheduling/. [Accessed: 20-Feb-2015].
[4] B. T. Rao and L. S. S. Reddy, "Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments," *Int. J. Comput. Appl.*, vol. 34, no. 9, p. 5, Jul. 2011.
[5] "Fair Scheduling," *Apache Hadoop*. [Online]. Available: http://hadoop.apache.org/docs/r1.2.1/fair_scheduler.html. [Accessed: 20-Feb-2015].
[6] "Capacity Scheduler," *Apache Hadoop*. [Online]. Available:http://hadoop.apache.org/docs/r1.2.1/capacity_scheduler.html [Accessed: 22-Feb-2015].
[7] X. Lin, Z. Meng, C. and M. Wang, "A Practical Performance Model for Hadoop MapReduce," in Cluster Computing Workshops (CLUSTER WORKSHOPS), 2012 IEEE International Conference on, 2012, pp. 231–239.
[8] J. Virajith, B. Hitesh, C. Paolo, K. Thomas, and R. Antony, "Bazaar: Enabling Predictable Performance in Datacenters," Microsoft Research, MSR-TR- 2012-38,[Online].Available:http://research.microsoft.com/apps/pubs/default.aspx?id=162192.
[9] H. Yang, Z. Luan, W. Li, D. Qian, and G. Guan, "Statistics-based Workload Modeling for MapReduce," in Parallel and Distributed Processing Symposium Workshops Ph.D. Forum (IPDPSW), 2012 IEEE 26th International, 2012, pp. 2043–2051.
[10] Z. Weizhong, M. Huifang, and H. Qing, "Parallel kmeans clustering based on mapreduce," in Proceedings of the CloudCom'09. Berlin, Heidelberg: Springer-Verlag, pp. 674–679, 2009.
[11] S. Papadimitriou and J. Sun, "Disco: Distributed coclustering with map-reduce: A case study towards petabytescale end-to-end mining," in Proc. of the IEEE ICDM'08,Washington, DC, USA, pp. 512–521, 2008.
[12] E. Alina, I. Sungjin, and M. Benjamin, "Fast clustering using mapreduce," in Proceedings of KDD'11. NY, USA:ACM, pp. 681–689, 2011.
[13] J. E. Tate, "Event detection and visualization based on phasor measurementunits for improved situational awareness," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Illinois, Urbana-Champaign, IL, USA, 2008.
[14] M. Ismail, "Parallel genetic algorithms (PGAs): master slave paradigm approach using MPI," in *E-Tech 2004*,july 2004, pp. 83 – 87.
[15] K. Krishnanand and D. Ghose, "Glowworm swarm optimisation: a new method for optimising multi-modal functions," *International Journal of Computational Intelligence Studies*, vol. 1, pp. 93–119, 2009. [Online]. Available: http://inderscience.metapress.com/content/H87J31211158L162.
[16] K. Chen, J. Powers, S. Guo, and F. Tian, "CRESP: Towards Optimal Resource Provisioning for MapReduce Computing in Public Clouds," IEEE Transaction Parallel Distrib. Syst., vol. 25, no. 6, pp. 1403 – 1412, 2014.
[17] H. Herodotou, H. Lim, G. Luo, N. Borisov, L. Dong, F. B. Cetin, and S. Babu, "Starfish: A Self-tuning System for Big Data Analytics," in In CIDR, 2011, pp. 261–272.
[18] K. Morton, A. Friesen, M. Balazinska, and D. Grossman, "Estimating the progress of MapReduce pipelines," in Data Engineering (ICDE), 2010 IEEE 26th International Conference on, 2010, pp. 681–684.
[19] L. Wang and C. Singh, "Multi-deme parallel genetic algorithm in reliability analysis of composite power systems," in *Proc. IEEE BucharestPowerTech Conf.*, Bucharest, Romania, Jul. 2009, pp. 1–6.
[20] M. Rallis and M. Vazirgiannis, "Rank Prediction in graphs with Locally Weighted Polynomial Regression and EM of Polynomial Mixture Models," in *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, 2011, pp. 515–519.
[21] J. C.-H. Peng, A. Mead, and N.-K. C. Nair, "Exploring parallel processingfor wide area measurement data applications," in *Proc. IEEE PowerEnergy Soc. Gen. Meeting*, San Diego, CA, USA, Jul. 2011, pp. 1–8.
[22] S. Kadirvel and J. A. B. Fortes, "Grey-Box Approach for Performance Prediction in Map-Reduce Based Platforms," in Computer Communications and Networks (ICCCN), 2012 21st International Conference on, 2012, pp. 1–9.
[23] M. Li, L. Zeng, S. Meng, J. Tan, L. Zhang, A. R. Butt, and N. Fuller, "MRONLINE: MapReduce online performance tuning," in *Proceedings of the 23rd international symposium on High-performance parallel anddistributed computing*. ACM, 2014, pp. 165–176.
[24] J. Shi, J. Zou, J. Lu, Z. Cao, S. Li, and C. Wang, "MRTuner: A toolkit to enable holistic optimization for MapReduce jobs," *Proceedings of theVLDB Endowment*, vol. 7, no. 13, 2014.
[25] R.Thangaselvi, S.Ananthbabu, R.Aruna"An efficient MapReducescheduling algorithm in Hadoop" International Journal of Engineering Research & Science (IJOER) [Vol-1, Issue-9, December- 2015].