International Journal of Computer Engineering and Applications,
Volume XII, Special Issue, May 18, www.ijcea.com ISSN 2321-3469

STRAGGLER IDENTIFICATION AND MITIGATION SCHEME FOR DATA INTENSIVE COMPUTING IN CLOUD ENVIRONMENT

Vaibhav Pawar [1], Shyam Deshmukh. [2]

[1]*Department of Information Technology, Pune Institute of Computer Technology*
Savitribai Phule Pune University, Pune

[2]*Department of Information Technology, Pune Institute of Computer Technology*
Savitribai Phule Pune University, Pune

ABSTRACT:

One of the challenges faced by data-intensive computing is the problem of stragglers, which can significantly increase the job completion time. Various reactive and proactive straggler mitigation techniques have been developed to address the problem. The straggler identification scheme becomes important for the improvement of job completion time in the cloud environment. Although the classical standard deviation method is a widely adopted straggler identification scheme, it is not an ideal solution due to certain inherent limitations In this paper we present an improved approaches for straggler Identification and mitigation.

Keywords: Cloud Computing, Data-Intensive Computing, Parallel and Distributed Processing, Straggler Identification.

[1] INTRODUCTION

The Stragglers are slow running nodes in cloud environment. In order to decide whether a node in cloud computing environment is straggler we need node configuration and we need to check how much tasks or jobs the particular node has executed in a given time.So it is necessary for us to schedule the jobs on the node to decide whether the node is straggler or not.For this purpose we need scheduling algorithms for mitigating the effect of stragglers. Stragglers are slow running nodes in cloud environment. In order to decide whether a node in clod computing environment is straggler we need node configuration and we need to check how much tasks or jobs the particular node has executed in a given time.So it is necessary for us to schedule the jobs on the node to decide whether the node is straggler or not. For this purpose we need scheduling algorithms for mitigating the effect of stragglers.

[1.1] ROOT CAUSES OF STRAGGLER:

Internal factors:

- Heterogeneity in the resource capability of a node.
- Competition between the tasks for resource allocation.

External factors:

- Competition for resources Co-hosted applications.
- Skewed Input dataset.
- Slow Processing speed of remote input/output source.
- Defective Hardware.

[1.2] MOTIVATION:

A job can finish its execution earlier as compared to that of individual processing. It would be very good if the job completion time can be further optimally reduced as compared to the existing reactive solutions . So this area of research is interesting. Google first introduced the MapReduce framework in order to parse and the enormous amount of data. Scalability of MapReduce Framework is very high with respect to large clusters. Hadoop is an popular example that shows how the map reduce framework is implemented in an open source manner and has been widely used by industries of both small and large sizes. For speeding up a job's execution time, MapReduce breaks a job into multiple smaller tasks. Execution of these tasks takes place in parallel manner and then submitted to multiple node cluster.Job is said to be completed when all the tasks related to that job has finished its execution. The major advantage of such distributed parallel processing environment is that they have automatic fault tolerance capability, without needing extra costs from the programmer.

[1.3] SCOPE:

The scope of the study is to reduce the execution time of the jobs running on the distributed systems environment. This study also tries for Stragglers avoidance, Improvement in resource utilization , Monitoring of task parameters like CPU utilization, RAM or memory usage , Network resources usage and bandwidth usage, Input/Output Disk Usage, number of threads related to task.

[1.4] PROBLEM STATEMENT:

To implement improved Straggler identification and mitigation scheme in Cloud environment.

[2] LITERATURE SURVEY

Today large number of computer applications are based on the Internet related services having millions and billions of users. This huge volume of data that has been leading the people to take interest in parallel processing and these applications are deployed on cluster environment. Example of this is Google search engine, which makes use of the map-reduce framework to parse large volume of data every day. Another internet based services like e-commerce websites such as Flipkart and social networks like Facebook, Instagram etc. .also processes huge volumes of data. For each action done by user the web application generates large datasets in system logs which then becomes the main resource for the developers and testers for checking the problems in production. Map reduce divides a large job into small tasks that run in parallel on multiple machines, and scales easily to very large clusters of inexpensive commodity computers. Google research study shows that that speculative execution technique can improve job execution time by 44%. Although this approach works well in homogeneous environments where stragglers are detected easily which is not the case when it comes to heterogenous environment. Existing techniques replication based techniques are not able to solve this problem. Speculative execution is a reactive replication-based straggler mitigation approach that generates redundant copies of the slow running tasks, hoping a copy will complete before the original task. This is the most widely used technique today. However, without any additional information, such reactive techniques can not differentiate between nodes that are very slow and nodes that are temporarily overloaded. In the latter case, such techniques lead to unnecessary over-utilization of resources without necessarily improving the job completion times. So we are proposing the proactive techniques for straggler identification and mitigation.

[3] STRAGGLER MITIGATION TECHNIQUES

[3.1] Blacklisting

Blacklisting detects nodes which has bad configuration (e.g., nodes with disk failure) and don't allocate tasks to such node in future on them in future . The cluster configuration of Facebook and Bing blacklists around 10% of their machines. Node failure can be due various reasons like Input/output device failure, regular maintenance causing interference, and hardware failure.

[3.2] Speculative Execution

Speculative execution reactive straggler mitigation technique stops and observes the progress of the tasks of a job and then launches duplicate copy of the tasks that are slower in the execution. However, speculative execution techniques is having a basic disadvantage when executing small tasks as it requires to wait for collecting samples of task performance of all the task and then needs to completes the execution. Such waiting results in limiting the agility of this approach while they deal with stragglers in small tasks.

[3.3] Machine Learning

Machine Learning can also be used for straggler mitigation, The machine Learning approach automatically learns correlations between node-level statistics and task-execution time in the form of decision-trees. The decision tree is easy-to-interpret allows scheduler to estimate task-execution time and, consequently, make a better decision. Machine Learning Technique for Straggler mitigation has small overhead on scheduler and also finds the root causes of stragglers and formulate interpretable rules which could be used in order to schedule tasks with the aim of preventing stragglers in future.

[3.4]Coding Theory

Coding-theory-inspired approaches can be used to mitigate the effect of straggler node in the cloud environment, this is done by embedding redundancy in some of the linear computational steps of the optimization algorithm, thus completing the execution of tasks without waiting for the stragglers tasks to complete. In Coding theory the redundancy is embedded directly in the data and allow the computation to proceed in completely oblivious manner and then the encoding of data is done. Various encoding techniques such as gradient descent and L-BFGS can be used in Coding theory straggler mitigation techniques.

[3.5]Distributed Machine Learning

Distributed Machine Learning is also called as multi-node machine learning algorithms and systems that are designed to reduce the effect of straggler nodes , improve performance, increase accuracy, and scale to larger input data sizes. Increasing the input data size for many algorithms can significantly reduce the learning error and can often be more effective than using more complex methods [8]. Distributed machine learning allows companies, researchers, and individuals to make informed decisions and draw meaningful conclusions from large amounts of data. Many systems exist for performing machine learning tasks in a distributed environment. These systems fall into three primary categories: database, general, and purpose-built systems. Each type of system has distinct advantages and disadvantages, but all are used in practice depending upon individual use cases, performance requirements, input data sizes, and the amount of implementation effort. Distributed machine learning can also be used for straggler mitigation. Distributed machine learning proposes a method called “Batched Coupon’s Collector” (BCC) This technique first partition the entire training dataset into batches and then each worker independently and randomly selects a batch to process and then decides which batch of jobs assigned which node is straggler or not.

[4] IMPLEMENTATION DETAILS

The system implementation will be using Apache spark on a multi-node Master Slave cluster Architecture. There will a master node that will assign the jobs to slaves in the form of smaller tasks and the slave nodes will execute these tasks and then the result will be integrated by master node. If a straggler is found, the Scheduler will notify the master node and no job will be assigned to that slave node until a suitable slave node is available and then assign the task to that particular slave node. There will be one master node and four slave node.

[5] EXPERIMENTAL SETUP

HARDWARE:	
MASTER NODE	CPU:-Intel Core i3 Frequency:-2.0GHz RAM: 8GB, Hard Drive: 1 TB
SLAVE NODE	Four Slave nodes with CPU: Intel Core i3 Frequency: 1 GHz RAM: 4 GB, Hard Drive: 500 GB
SOFTWARE:	**VMWARE 10 OR ABOVE** for having multiple Virtual machines running
NAGIOS	For monitoring the behavior of all the nodes.
WORKLOADS	Wordcount, Pi Estimator, Tera-sort programs will be executed on each node to check the performance.
APACHE SPARK	For Cluster Configuration

[6] PROPOSED SYSTEM ARCHITECTURE

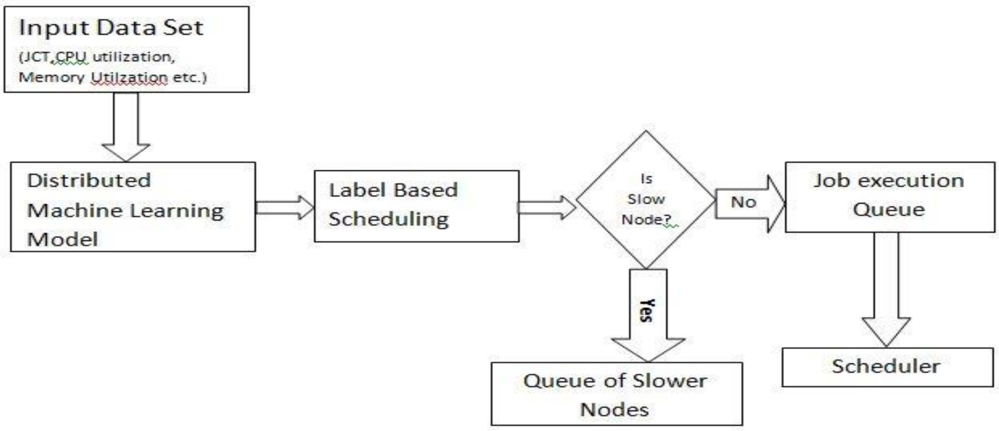

Fig.1 Proposed System for Straggler identification and mitigation

LABELLING THE NODE:-

The node will be termed as straggler or not a straggler based on the criteria follows:-

Straggler Node can be specified, if for the task ti for a job J

$$\mathbf{nd}\ (\boldsymbol{ti}) > (\boldsymbol{\beta} \times \boldsymbol{median}\{\boldsymbol{nd}(\boldsymbol{ti})\})$$

Where ,β is coefficient of threshold (β~1.3)

nd is Normal distribution of the load of input data

Now , this functions returns a Boolean value (0,1)

0 Non-Straggler node

1 Straggler node

[7] EXPECTED RESULTS

It is expected that the distributed machine learning model will identify the straggler node and the scheduler only schedule the job to non-straggler node.If the node is straggler node no jobs will be submitted to that node. We are comparing the results of our improved proactive straggler mitigation scheme with default speculative execution straggler mitigation scheme.It is expected that the Job Completion Time of tasks in Default scheduler will be reduced as that of proposed scheduler, accuracy of identifying the straggler node will also be increased prominently, load balancing will also be improved as we are using distributed machine learning approach.

[8] CONCLUSION

In this paper we presented various straggler mitigation approaches. We have studied the Blacklisting,Coding theory, Speculative Execution, Machine Learning and Distributed Machine Learning Techniques The Machine Learning Techniues proving to have better performance than others. We have compared all the techniques with each other based on the performance. Distributed technique with Machine learning approach is the best amongst the other approaches compared. We have presented a brief analysis of the different mitigation techniques based on various performance parameters.

[9] FUTURE SCOPE

The existing techniques are meant for straggler mitigation or for reducing the impact of stragglers. In future we can have techniques for elimination of stragglers altogether without resource blacklisting and optimal use of resources. Although being the best of the existing straggler mitigation approaches Proactive Machine learning Wrangler technique has its own shortcomings too. For datasets having huge difference among the numbers which are obtained from the multiple nodes we say that this technique is not much effective. Thus we could try to generalize the technique for all kinds of datasets eventually improving it's confidence measure too.

REFERENCES

[1] Xue Ouyang1, Peter Garraghan1, Renyu Yang2, Paul Townend1, Jie Xu1 “Reducing Late-Timing Failure at Scale: Straggler Root-Cause Analysis in Cloud Datacenters” In IEEE DSN 2016.

[2] Understanding the Dark Side of Big Data Clusters: An Analysis beyond Failures in 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks by Andrea Ros`, Lydia Y. Chen and Walter Binder.

[3] Failure Analysis of Jobs in Compute Clouds: A Google Cluster Case Study, in 2014 IEEE 25th International Symposium on Software Reliability Engineering by Xin Chen, Charng-Da Lu and Karthik Pattabiraman

[4] A Survey on Hadoop MapReduce Framework and the Data Skew Issues, International Journal of Scientific Research Engineering & Technology (IJSRET), April 2015 , by Nawab Wajid, Prof. Satish S, Prof. Manjunath T N.

[5] Dynamic Resource Allocation for MapReduce with Partitioning Skew, IEEE TRANSACTIONS ON COMPUTERS, September 2014, Zhihong Liu, Qi Zhang, Reaz Ahmed, Raouf Boutaba, Yaping Liu, and Zhenghu Gong , Members, IEEE, Xue Ouyang, Peter Garraghn, David Mckee, Paul Townend, Jie Xu “Straggler Detection in Parallel Computing Systems through Dynamic Threshold Calculation “ In 2016 IEEE 30th International Conference on Advanced Information Networking and Applications.

[6] Neeraja J. Yadwadkar, Ganesh Ananthanarayanan , Randy Katz , “Wrangler: Predictable and Faster Jobs using Fewer Resources “ In SoCC ’14, 3-5 Nov. 2014, Seattle, Washington, USA.

[7] Matei Zaharia, Andy Konwinski, Anthony D. Joseph, Randy Katz, Ion Stoica “Improving MapReduce Performance in Heterogeneous Environments “ In 8th USENIX Symposium on Operating Systems Design and Implementation.

[8] Ganesh Ananthanarayanan, Ali Ghodsi1, Scott Shenker, Ion Stoica “Effective Straggler Mitigation: Attack of the Clones “ In University of California, Berkeley , KTH/Sweden.

[9] Mukhtaj Khan, Yong Jin, Maozhen Li, Yang Xiang and Changjun Jiang ” Hadoop Performanc12. JaideepDhok and VasudevaVarma, Search and Information Extraction Lab,International Institute of Information Technology, Hyderabad, India “Using PatternClassification for Task Assignment in MapReduce”, 10th IEEE/ ACM conference,2010.

[10] Lei Wang, RuiRen, Jianfeng Zhan, and Zhen Jia, Institute of ComputingTechnology, Chinese Academy of Sciences. “Characterization and ArchitecturalImplications of Big Data Workloads”,2016 IEEE.

[11] Zhen Jia1,2, Jianfeng Zhan 1*, Lei Wang1, Rui Han1, Sally A. McKee3, QiangYang1, Chunjie Luo1, and Jingwei Li ,State Key Laboratory Computer Architecture,Institute of Computing Technology, Chinese Academy of Sciences “Characterizingand Subsetting Big Data Workloads” , 2014 IEEE.