ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 3, Issue 1 (Jan. - Mar. 2015)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2015)

Processing of Big-Data to Improve Performance Using Clouds

[I]Roshan Patil, [II]Yogesh Patil, [III]Sashank Shinde, [IV]Ritesh Yadav, [V]Prof. Poi Tamrakar
[I,II,III,IV,V]ISB&M School of Technology., Nande, Pune, India

Abstract
Now day we have see that data is generated from different sources likeclick streams, sensors, social media, and log files,mobile devicesetc.Big-Data is the data that exceeds more than thousands of terabytes. So, maximizing the performance of big-data analytics services over distributed clouds is just becoming a complex task. For maximizing the performance we have considered using parallelization in the distributed clouds. One of the important or serious problem while considering the parallelization of big-data analytics services on distributed cloud .present systems are based on the present status of the network and also on the computational capacities. We use this system to over-come problems like network congestions as well as the node failures. Inproposed system (i) firstly we considers the nodal distribution on the basis of initial delay of input data then, (ii) we do the initial sorting of data based on the delay of arrival and nodes, (iii) nodes will be assigned by using Greedy Bin Packing (GBP) algorithm, (iv) we sort the data and re-assign nodes, if nodes failure occure(v) otherwise compute the computation timeas well as the data transfer delay. Finally, the computation time are analyzedand thedata transfer delay which will be maximize the performance of big-data analytics services.

Keywords
The Distributed Clouds; Big-Data Analytics; Parallelization;Node Failure,Cloudcomputing, Data Mining

I. Introduction

Now days trend to provide Big-Data Analytics services into the clouds. Big-Data means a collection of data sets so large and complex that it becomes difficult to process using database management tools or the traditional data processing applications or we also called unstructural data. Now days we see that data is implemented from different sources like log files, sensors, social media, click streams, andmobile devices etc. Recently this collection of data exceeds more than hundreds of terabytes or petabytes and is being continuously generated. Such type of data is known Big-Data which is represent datasets that cannot be simply analyzed by traditional infrastructures and data management methods [2,3,4]. In this situation to derive data from Big-Data, enterprises have to use effectively big-data analytics into the more sufficient scalable platform. Big-data analytics [4,14,15] is the method of examining large amounts of data of different types unknown correlations,to uncover hidden patterns and other useful information. Such information give us competitive advantages over competitive organizations and result in business benefits, such as increased more effective marketing and revenue. The invention of Cloud computing had been enables the different enterprises to analyze big-data by influencing large amount of computing source or resources that are available on demand with minimum resource usage cost.

One of the helpfull research challenges in this regard is to display how to best use cloud resources distributed by high latencies. In this session, we focus on parallel data mining [1] to analyze large amounts of structured and unstructured data which consumes lots of computing resources, mainly when they are perform with a time constraint. The service providers of clouds may contain enough capacity to store, which are used for such a data services in their data centers. Simplifying this a distributed cloud and loosely coupled which contains applications and legacy resources is the better option, analytics is carried on partially local private resources and the another big data has to be transferred to the external computing nodes that should be maximized performance for the big-data analytics. They also had cost benefits than a single data center as well as more flexible methods[5] [6]. To improve a parallel data mining is not only different clouds which are separated by high latencies but also multiple computing nodes[1], this paper addresses: (1) measure the nodes, i.e. " how many and which" computing nodes in distributed clouds should be used, (2) how to apportion the big data across the parallelized computing environment to ensure the synchronization,i.ecompletion of synchronized data where synchronization means completing of workload portions at the same time or when resources and inter-networks are different and situated in multiple Internet-separated clouds, and (3) partition determination of data, i.e., how to serializeheterogeneous data chunks to computing nodes to avoid underflow or overflow to nodes.Toaddresstheseproblems,we have developedaheuristic cloud- burstingalgorithm,referredtoasMaximallyOverlappedBin- packingdrivenBursting [1]. We hadimproved theadvantageofdatamining parallelization by considering the time overlap: (a) across computing nodes; and (b) between data transfer delay and computation timeineach of thecomputing nodes. While the loads which are unequalmaybeapportionedtotheparallel computing nodes, proposed algorithm canmakesurethat outputs are produce dat the same time without any single slownodeactingasa bottle neck.

Whendataminingisrunonapoolof inter-connected clouds, there is an extended period of data transfer delay,andthedat atransferdelaydependsonthelocation ofeachcomputingnode. Fast transfer of data chunks to slow computing node can cause the dataoverflow,whereasslowtransferofchunksto a fast node can lead tounderflowcausingthenodestobe idle. Ouralgorithmcanreducesuchdata overflowor underflow.

Toevaluatethisapproach,wedeterminedafrequent pattern mininganalytics[7]asaspecifictypeofparallel analytics whose inputs are huge but outputs are far smaller.Afterthis,we deployth eanalyticsonsmallmultipleHadoop[8]clustersinto differentclouds. Theexperimentalresultsshowthatproposed approachoutperforms otherexistingload-balancingmethodsforbig-dataanalytics.

II. Literature Survey

The concept of load distribution had been studied last for number of distributed computingsystems includingparallelarchitecture[9],computinggrids[8], anddata centers[10,11,12].to maximize the performance,Load balancingforparallelapplications hastypicallyinvolvedthedistribution ofload to computing node.Otherwisethecostand delayoverheadshave considered inmany forms. In thispaper we mainly consider the distribution of big-data over computing nodes which

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

areseparatedfarapart from each other. Inthis,theoverheadofthed atatransfercan be significant,asthenetworklatenciesmaybehigh betweennodesduetotheamountofdatabeingtransferred.

The recent research work [13,14] had introduced the impact of the data transfer delay on the performance when they select cloud to redirect service loads, they do not consider further optimization by overlapping the transfer delay of a data with the computation time of previous data. Need of such overlapping has been identified for clusters [15]. Continuation based overlapping of data transfers with instruction execution has been investigated for many-core architecture [9]. However, such type of overlapping is restricted by a pre-defined order of different sizes of data chunks to be transferred to individual nodes. By doing so, we can get the maximum overlap between the data transfer and the data computation.

The other way to optimize the performance of the big-data analytics is scheduling sub-tasks among computing nodes. For example, back filling of tasks at the earlier time than the original scheduled sequence has been considered as a part of batch-scheduling in the data centers and computing clusters [10]. However, such scheduling is processed towards batch processing within data centers and not for big-data apportioning in heterogeneous federated clouds.

Some approaches had introduced task schedulers with load-balancing techniques in heterogeneous computing environments. For example, CometCloud [16] had included a task scheduler to run analytics on hybrid clouds with a load-balancing technique. [17] has introduced some of the heuristics to schedule tasks for heterogeneous computing nodes. None of the above approaches have done deal with the potential tradeoff between the data transfer delay and the computation time in parallel the computation time in parallel execution environments. [18] has introduced a scheduling algorithm to address how many and which tasks in a task queue have to be run in internal cloud and sent to external clouds. They had mainly focused on keeping the order of tasks in the queue while increasing performance by utilizing an external cloud on demand. However, they had not considered how many and which clouds are required and how much data is allocated to each chosen cloud for parallel processing.

For these problems MOBB (Maximally Overlapped Bin-Packing Driven) algorithm has been introduced. In this MOBB approach they had considered (a) how many and which computing nodes in federated cloud should have to be used, (b) completion of synchronized data, i.e. how to apportion the big-data across the parallelized computing environment to ensure the synchronization, whereas synchronization means completing the workload portions at the same time even when resources and inter-networks are heterogeneous and situated in multiple Internet-separated clouds, and (c) partition determination of data, i.e. how to serialize different data chunks to computing nodes to avoid overflow or underflow to nodes. But the MOBB approach had not considered unexpected events like network congestions as well as node failures so as to maximize the performance of parallelization. Also we had provided a load balancing algorithm for the distributed parallel applications.

(a) Comparison of Proposed Approach with other load balancing approaches:

We are going to run the mining task using proposed approach and otherdifferent methods that are used in many prior loadbalancingapproaches and then, going to comparthe results. Methods used in this comparison are as follows,

- Computation-based division: In this method,itconsidersthe computation power of each node while load-balancing, rather than considering the data transfer delay and computation delay.
- Fair division: In this method, it equally divides the input dataand distributes that data to nodes. We use this method as a naïve method to show as a baseline.
- Delay-based division: In this method, it considers both the node's computation time and data transfer delay in loadbalancing.
- MOBB approach: In this method, it considers the queuingdelay in each node by distributing userlogs to nodes (i.e., it does not consider the time overlapbetween the transfer delayconsiders how many and which computing nodes can be used for parallel computing.

Here none of the approaches deals with unexpected things or situation like node failure,they deal or consider only computation time and data transfer delay of each. Also doesn't consider the re-sorting or re-assigning of the nodes dynamically as well as the nodes while node failure. Our proposed system consider these concepts like re-sorting of nodes,node failure also we will improve the result then the existing approaches or things.

III. Problem Statement

We have to determine "How Many" and "Which Computing Nodes" in clouds are required where each computing nodes can be a cluster of servers in a data center, and how to deserve given big-data to chosen computing node as well as importantly,we have to look after unexpected events such as node failures and networkcongestions.We are doing this all things to achieve the maximum performance of bigdata analytic services in distributed clouds.

The frequent pattern mining algorithm of the input big-data(e.g. a log file containing user's web transactions) is typically collected in central place over a limited period (e.g. for few years), and To generate an output (e.g. slogs for user groups)this process is done.

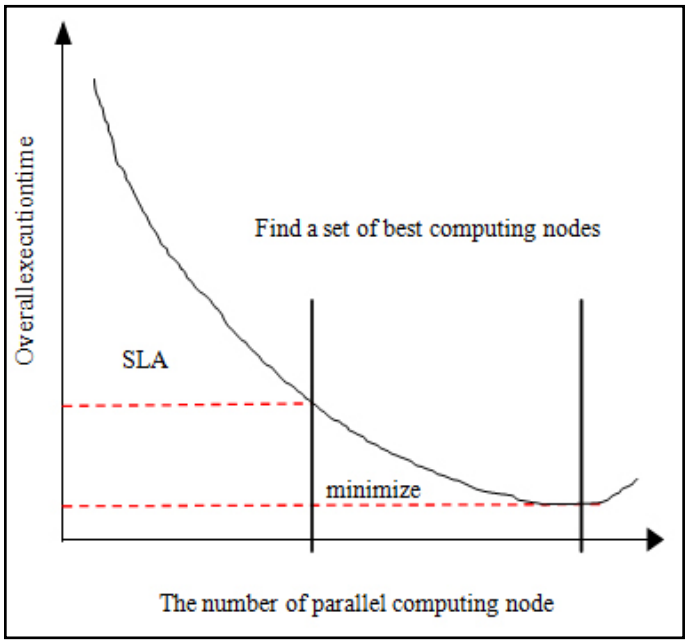

Fig. 1: Relation between the overall execution time and the number of computing nodes

First divided the big-data into a specific number of data chunks (e.g. logs for user groups), and those data chunks are transferred

to computing nodes.We have doing this to execute the mining task in multiple computing nodes for given big-data, Truly,the overall execution time can decrease,as the number of computing nodes increases but the amount of data chunks to be transferred increases. As shown in fig. 1.the overall execution time can start to increase if we use beyond a specifics number computing node. This is because the delay taken to transfer data chunks start to dominate the overall execution time. Meanwhile adding computing node is optionally stopped once a target execution time specified in the Service Level Agreement (SLA) is met.

Our approach or proposed system is designed to address the problem of MOBB approach i.e. MOBB algorithm is not capable to solve the problems such as node failures or Network congestions. Our approach deal with the system to dynamically check whether all the computing nodes are available or not if the node is not available or node failure occurs at that time we re-sort the data and then re-assign the computing nodes, so we can compute for further process. Finally, we have to analyze or observe the data transfer delay and the computation time of the each node. We overcome the problems of finding out which computing nodes and how many computing nodes should be used by the distributed clouds as shown in Fig. 2. Our approach or algorithm aim at maximizing the time overlap across computing nodes when it performs the parallel data mining.

Proposed system maximizes the performance of the parallel mining by considering the dynamic check that the node failure had occurred or not as well as maximizing the time overlap not only across computing nodes, but also between the data transfer delay and the computation time and in each nodes accordingly.

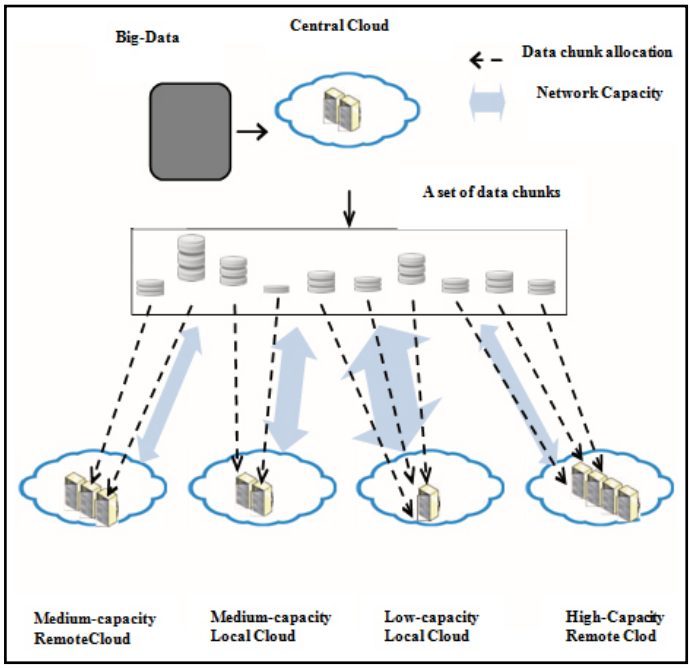

Fig. 2: Data allocation to different capacity clouds.

IV. Proposed System

Firstly, our aim is to find out how many nodes andwhich nodes are to be used. Whatever are the estimates or analyses of the data transfer delay and the data computation time, our algorithm selects the set of parallel computing node, distributes the nodes by considering the data transfer delay, then we perform the initial sorting of data which is based on the computing node and delay of arrival of data. After this we will assign the node, this process is based on the Greedy Bin Packing algorithm, which was proposed in MOBB approach at the same time, we are checking the failure of any nodes. For this, we are assigning the process initially, which will dynamically check for the uncertain events like node failures or

failure, then it will be automatically resort the data and reassign the nodes which will further continue to compute the process to analyze the delay for the computation time as well as data transfer.

As shown in fig. 3, the proposed system is used to through the drawbacks of MOBB mainly reducing the node failure, which was not considered in that algorithm. Firstly, we have to find the total size of data chunks, and then sort the computing nodes in descending order. There are different modules used in this algorithm which is described as follows:

A. Nodal Distribution

For each node first bucket size is find out. The bucket size depends on the data transfer delay. The node with higher delay will have a lower bucket size. The node buckets are sorted in descending order of their size. This nodal distribution will be decide"which and how many" computing nodes to use, how to apportion the data and sort the data in the descending order of their sizes.

To find out the bucket size our algorithm parallelizes the mining task by dividing input data to the multiplescomputing nodes. For this parallelization the size of given data for particular depends on the average delay of mining task on that particular node. Average delay of task for a particular unit of data on a node i is denoted as di also the overall delay for executing a data size s_i is $s_i d_i$. Consider n number of nodes and a set of data with different sizes $s_1, s_2, s_3, s_4, \ldots\ldots s_n$. To ensure parallelization following condition should be satisfied,

$s_1 d_1 = s_2 d_2 = s_3 d_3 = s_4 d_4, \ldots\ldots = s_n d_n$

where $d_1, d_2, d_3, d_4, \ldots\ldots$, dnare the delay per unit of data. Assume r, to be overall execution time of mining task, the data size of each node can be as follows,

$$s1 = \frac{r}{d1}, s2 = \frac{r}{d2}, s3 = \frac{r}{d3}, \ldots\ldots, sn = \frac{r}{dn} \quad (1)$$

Now, consider n as a number of inputs which are distributed each node. Then,

$$r = \frac{s}{\sum_{i=1}^{n} \frac{1}{di}} \quad (2)$$

With this Eq. 2 we got the overall execution time of the mining task for parallelization. This mining task can be achieved if task assigned to a node *i* is limited by upper bound *is*, which can be given as shown,

$$s_i = \left\lfloor \cdot \frac{s}{di \sum_{i=1}^{n} \frac{1}{di}} \right\rfloor \quad (3)$$

Here, if d_i is lower thensiwill be higher to node i. Here, Eq. 3 is used to find the bucket size of each node where higher priority is given to nodes with lower delay.

B. Initial Data Sorting

In this step the input data chunks are place in descending order of their size. This sorting of buckets give the better performance, which is having a delay.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

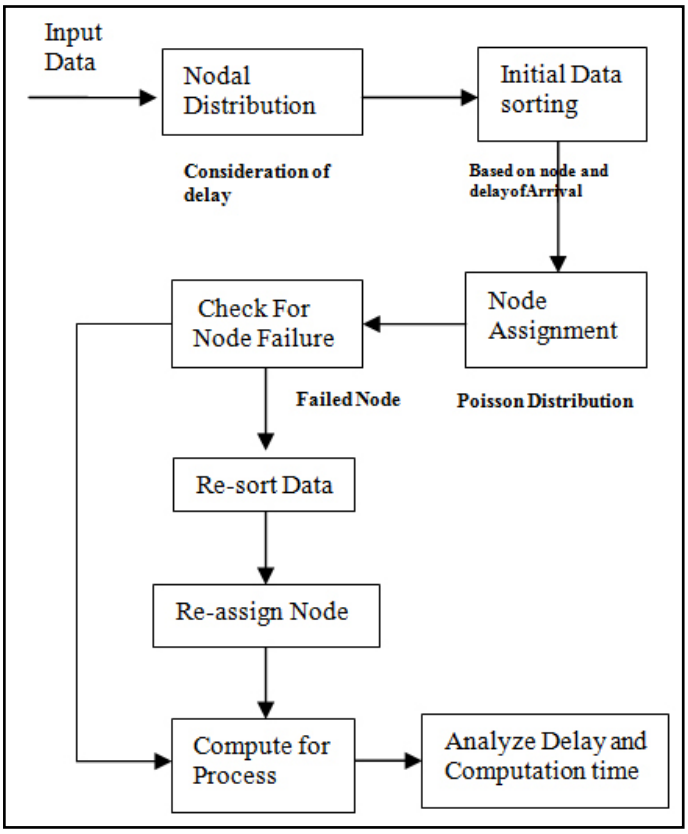

Fig. 3: Model of the proposed system to handle Node failure

C. Node Assignment

The sorted list of data chunks is then assigned to the each node buckets in such a way that assigns more data to node with have less delay. So that larger data chunks are handled by the nodes which have higher preiority. This node assignment is based on the Greedy Bin Packing algorithm which is as follows,

Once we find out the bucket size for each node, then next step is that to assign the data chunks to the each node buckets. We use this Greedy Bin Packing approach to assign the largest data chunks to nodes which havea lower delay (so to reduce the overall data transfer delay).

This involves the two steps as follows:

1) Weighted load distribution

In this, the load is given to the node according to overall data transfer delay. That mean the node having a higher delay this node handle small data. This can be assured by providing an upper bound on the size of data, i.e. bucket size, to be handled by the node.

2) Delay-Based Node Preference

The nodes have the larger bucket size are loaded with larger data chunks, so that each data chunk is given the fairest in overall delay. This will assured by sorting of data chunks in descending order and loading nodes with have larger bucket size first. In this the preference is given to node which has lower delay.

D. Check for Node Failure

First assigning the each data chunks to node, there may be a case that the node to whichthe communication link is failed or the data is to be assigned is down. To handle such type of situation the proposed system will dynamically check whether there is any kind of network congestions or the computing node is down.

In such a situation one of the process is allocated so that, it will check every time dynamically for the situation. If there is any node failure or network congestions then in these case the data chunks are re-sorted and then data is re-assigned for that particular failure-node

V. Experimental Setup

To implement proposed system, we need following setup:

1) Frequent Pattern Mining: Frequent pattern mining is help to extract frequent patterns from different log file. The log have user's activities in a system in temporal order.A typical example is a web server access log, which contains a information about of web page accesses from the users.

2) Computing nodes: For in this experiment, there are needto small computing nodes to run the frequent pattern mining. There are require three Virtual machines one on Remote site and local site. Each machine will have 100 GB hard drive and 1GB memory with 2.27 GHz Processor.

VI. Conclusion

In this paper, a system is proposed over cloud-bursting which is based on load-balancing algorithm to maximize the performance of big-data analytics services which runs in distributed clouds. Specifically, proposed algorithm will maximize the performance by considering the facts like node failure, task queue, or network congestions as well as improving the performance by computing the difference between data transfer delay and the computation time. Also, this systemcomparesthe performance of our system with the performance of other existing load-balancing algorithms.

References

[1] *T.Mukharjee,G.Jung.N.Gnanasambandam,"Synchronousof Parallel Processing of Big-Data Analytics Services to Optimize Performance in Federated Clouds",2012 IEEE Fifth International Conference on Clouds Computing.*

[2] *A.Jacobs. (2009,Jul.)The Pathologies of Big-Data. [Online]. Present by :http://queue.acm.org/detail.cfm?id=1563874*

[3] *T. White, Hadoop: The Definitive Guilde. O'Reilly, 2009.*

[4] *D.Kusnetzky. (2010,Feb.) What is mean by big data? [Online]. Present by: http://blogs.zdnet.com/virtualization/?p=1708*

[5] *A. Slominski, and Y. Doganata,S. Rozsnyai "Large-scale distributed storage system for business provenance," in Proc. Int'l. Conf. on Cloud Computing, 2011, pp. 516–524.*

[6] *M. Hsu, and H. Zeller,Q. Chen "Experience in continuos analytics as a service (caaas)," in Proc. Int'l.Conf. on Extending Database Technology, 2011, pp. 509–514.*

[7] *R. Agrawal and R. Srikant "Mining sequential patterns: Generalizations and performance improvements," in Proc. Int'l. Conf. on ExtendingDatabase Technology, Feb. 1996, pp. 3–17.*

[8] *Apache.(2011) Hadoop. [Online]. Present by:http://hadoop.apache.org/*

[9] *Z. Lan and Y. Li, "A survey of load balancing in grid computing," Springer, Information Scienceand Computational, vol. 3314, pp. 280–285, 2005.*

[10] *K. Kise, T. Yoshinaga,T. Miyoshi and H. Irie, "Codie: Continuation based overlapping data-transfers with instruction execution," in Int'l. Conf. on Networking and Computing, Nov. 2010.*

[11 *D. Feitelson], Y. Etsion, andD. Tsafrir, "Backfilling using system generated predictions rather than user runtime*

estimates," IEEE TPDS, Vol. 18, pp. 789–803, 2007.

[12] S. Gupta, A. Banerjee, G. Varsamopoulos,T. Mukherjee and S. Rungta,"Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers," Comp. Net., vol. 53, pp. 2888–2904, 2009.

[13] Y. Ho, C. Lu,Y. Huang and L. Fu, "A cloud-based, accessible architecture for large-scale adl analysis services," in Proc. Int'l. Conf. on Cloud Computing, 2011.

[14] SandeepRaut. "Big-Data Analytics", 2011,Present by: http://smartdatacollective.com/rautsam/44182/big-data-analytics by [15]"Big-Data Analytics",Present byhttp://searchbusinessanalytics.tectarget.com/definition/big-data-analytics, Jan 2012by Lisa Martinek.

[16] Keith Collins,Present by: http://www.sas.com/big-data-analytics.html.