International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-4, November 2019

An Efficient Resource Scheduling Framework for Infrastructure Performance Enhancement in Cloud Environment

M. Abdullah, M. Mohamed Surputheen

Abstract: ***On-demand cloud services must be provided to customers at any time by ways of cloud service providers due to cloud demand. It is obligatory for cloud service providers to lessen large volumes of data, thereby it can reduce costs for maintaining large storage systems.Infrastructure level performance is an important problem which directly affects the overall working of cloud computing environment. The objective of our framework is enhancing the performance of cloud infrastructure. Proposed approach demonstrates high effective in cloud performance enhancement, as it displays enhancement in both the service providers as well as for cloud users.***

Keywords: ***Resource allocation, Cloud Computing, Fuzzy Scheduling, Hadoop, Security and Privacy.***

I. INTRODUCTION

Cyber attacks and isolation are the chief threats in cloud computing technology. Besides, challenges are also included when providing secured and consistent services to customers. Some of the factors on security methods are; OS, memory management, computing networks, virtualization, balancing in load and scheduling in resources etc., Security is a key element in cloud computing. Data can be exposed and leaked within any business areas or an entity and lead to cause of lack in trust [6] & [7].

II. RELATED WORKS

Scheduling in delay is a scheduling policy to improve facts area for Hadoop platforms [8]. The algorithm [8] provided an improved correctness in the line of cost implementation point. Via distributed environment, load balancing could be computed by using Charm++ tool. The effort in [11] suggested building of an unidentified confirmation protocol. In [12], main concept in verification is formation of anonymity position for all the users, is a challenge. In [13], proposed a sample scheme, where cloud computing is shared with trusted platform hold up service [14]. In order to implore the cloud services, requirements of the patrons must be vivid and intelligent to produce Service Level Agreements (SLA), would extend all services, supplies along with effectiveness.

III. CLOUD METRICS

In order to enhance cloud services, metrics must be taken into account since a metric delivers information about features of a cloud property by its parameters namely, unit, rules and expression and the values emerging from the observation of the property (Refer Figure 1). To estimate a particular response time property from one client to another, it can be considered at the customer response time metric employed in a cloud email service search feature. By considering the customer response time metric, it can drive indispensable information which can be employed for verification of observations and analyzing the results.

Metrics for cloud computing services can be described employing the Cloud Service Metric model which states higher level perceptions about metric definitions. Nevertheless cloud metrology has to be understoodin a proper way. Frequent terminologies namely, definition of measurement, metricor groups of measurement artifacts, which consists of have numerous definitions and these myriad terminologies become tedious for the cloud service customer to associate services or depend on third party tools to observe the health of the service. And cloud provider should be to analyze service whether it is operating correctly or to permit its service to arrive into an intricate federation of cloud service. International organizations shouldconstruct a group of metrics which should be reliable, shareable and trustworthy .By defining the metrics; it is not only increases the support of the decision-making process but enhances the different phases in cloud service lifecycle.

CC is regarded as a competent data management [15] which employs k-median clustering for managing data. Even thoughcloud computing entails high profitable technology for business value, it lacks certain security features.Hence it is obligatory to ensure that data and infrastructural resources in cloud are scheduled and data deduplication is carried out in an efficient way. Current frameworks have some limitations and not able to resolve certain drawbacks encountered in cloud [16].

Manuscript published on November 30, 2019.
* Correspondence Author
M.Abdullah*, Corresponding Author, Research Scholar, Department of Computer Science, Jamal Mohamed College (Autonomous), (Affiliated to Bharathidasan University), Tiruhcirappalli, Tamilnadu, India. (Email : majmc08@yahoo.com)
Dr. M. Mohamed Surputheen, Associate Professor, Department of Computer Science, Jamal Mohamed College (Autonomous), (Affiliated to Bharathidasan University), Tiruchirappalli, Tamilnadu, India.

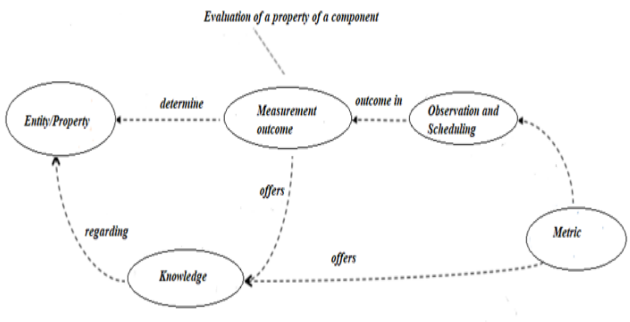

Figure 1: Cloud Metrics

Cloud storage employs facts de-duplication practice to augment the concert of cloud storage. The procedure of data deduplication employed for improvement of efficiency in storage. In a regular deduplication system, duplicated fact recognizes well collected single copy of data storage. Generation of rational pole is employed intended for additional copies as a stand-in for storing excess data. The main gain of employing de-duplication is to diminish cargo space and bandwidth of network [17].The drawback of employing this technique is to take a toll on system tolerance since files denote to the identical chunk of data. Due to the drawbacks of static schemes, de-duplication in cloud storage requires an active plan which has the capability to acclimatize in a variety of entrée patterns and altering the scenery of consumer manners in cloud storage.

IV. DEPENDABILITY ISSUES

In order to lessen storage space in cloud, we can employ data deduplication technique. This can be done by employing hash functions i.e. values to associate chunks of data with redundant data, thereby saving single copy and logical pointers to additional copies are created [17]. To employ de-duplication technique, the following advantages are achieved:

1. Amount of data residing in disk space is reduced
2. Energy consumed by storage systems will be reduced.
3. In cloud disaster recovery, deduplicationplays a vital role by replicating data which speeds up replication time and bandwidth cost savings.
4. Data deduplication reduces physical capacity and network traffic which acts as a backup storage in clouds.

Important performance metrics which is obligatory to follow: overall data transmitted, service downtime and total migration time. De-duplication can overcome extensive downtime which leads to service failure. In virtual machines, de-duplication is employed to lessen storage of dynamic data. Balancing the trade-offs between cargo space space and concert plays a core factor in de-duplication technique.

V. PROPOSED FRAMEWORK

Infrastructure level performance is an important problem which directly affects the overall working of cloud computing environment. The objective of our framework is enhancing the performance of cloud infrastructure. The cloud infrastructure deals with several nodes which eventually lead to queueing of systems, where huge amount of servers displaysinconsistency in terms of processing arrival and service time. Considering the potentially high number of servers in a cloud system, proposed a fuzzy scheduling algorithm, which is efficient and easy-to-implement solution.The proposed framework is integration of a fuzzy scheduling scheme to enhance infrastructure performance of cloud. Our approach demonstrates high effective cloud performance enhancement, as it displays enhancement in both the service providers as well as cloud users. By distributing resources as per business strategies and availability will benefit cloud service providers. In a nutshell, proposed framework consists of following stages: Monitoring the data usage pattern and allocating priority by cloud service provider according to our proposed fuzzy scheduling algorithm, which in turn to enhance infrastructure performance.

5.1 Fuzzy scheduling algorithm

The selection of parameters plays a vital role in providing effective scheduling. Considering real time tasks, three main scheduling parameters; computation time, arrival time, deadline and single output where the single output computes the order of task execution. In fuzzy scheduling system, scheduling parameters are included with real-time tasks. The computation time is determined in terms of response time and utilization. Membership function has defined in a fuzzy group S on the cosmos of discourse P is described as a curve which acts on each tip of constituent of D in the room of input with a value of 0 and 1. The universe of revelation is indicated as,

$$\mu_s: D \longrightarrow [0, 1] \qquad \text{eqn (1)}$$

Where value μs, is a value of membership, states the membership spot of the constituent D in fuzzy set S. Variety of membership functions are trapezoidal MF and Gaussian MF. Here it is employed a triangular MF, is broadly used in real-time execution for successful computations.

$$\mu_D(d) = \begin{cases} 0, & d \leq n \\ \frac{d-n}{w-n}, & n < d \leq w \\ \frac{q-d}{q-w}, & w < d < q \\ 0, & d \geq q \end{cases} \qquad \text{eqn (2)}$$

Here, three parameters {n, w, q} where n<w<q defines the triangle membership. The dcoordinates of the underlying triangular MF in Figure 2 were determined by n, w and q. Here y axis indicates the degrees of membership in the [0, 1] interval and x axis shows the cosmos of discourse.

Retrieval Number: D5286118419/2019©BEIESP
DOI:10.35940/ijrte.D5286.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

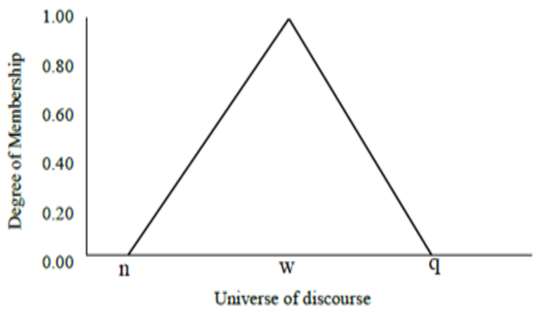

Figure 2: Fuzzy inference model

The Figure 2 presents a fuzzy inference model consists of three main scheduling parameters, namely, computation time, arrival time, deadline and single output.

5.2 Defining the Inputs/Output Membership Function

The input parameters of the fuzzy along with their linguistic groups, namely, medium, long short are employed for arrival time, computation time and limit of the real-time tasks. Each and every task is allocated with the linguistic category for each parameter. The membership function distributions signifying the linguistic notion for fuzzy inputs and runtime priority were distinct. The fuzzy input parameters were defined by employing triangular membership functions are described as:

Table 1: Represents linguistic categories defined for the arrival time of all tasks

Fuzzy Categories	Crisp Input Range
Early	0 – 5
Intermediate	0 – 10
Late	5 – 10

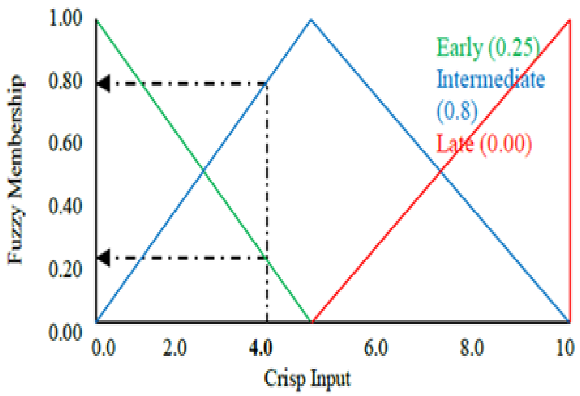

Figure 3: Membership Function for Arrival Time

Table 2: Represents linguistic categories defined for the computation time of all tasks

Fuzzy Categories	Crisp Input Range
Short	1 – 10
Medium	1 – 20
Long	10 – 20

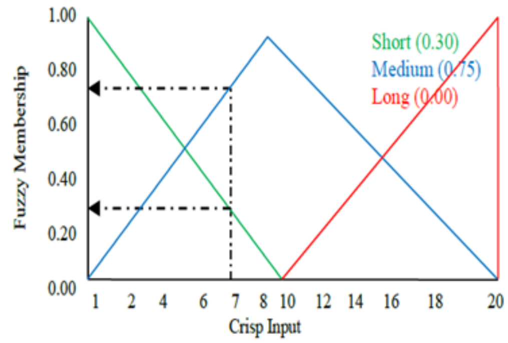

Figure 4: Membership Function for Computation time

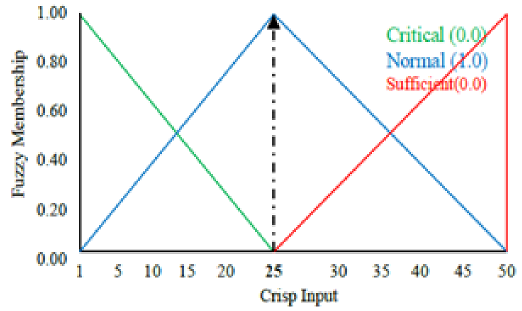

Figure 5: Membership Function for Deadline of all tasks

Table 3: Represents linguistic categories defined for the deadline of all tasks

Fuzzy Categories	Crisp Input Range
Critical	1 – 25
Normal	1 – 50
Sufficient	25 – 50

5.3 Fuzzy Rule Base Arrangement

In fuzzy rule base arrangement, the foundation of the numeral of linguistic groups are defined in each one of the input parameter. Let q, w, n represent the number of linguistic categories defined for computation time, arrival time and deadline, then the fuzzy rule base will have 27 rules i.e.(3 × 3 × 3) rules of fuzzy logic.

Table 4: Fuzzy Rules

S.No.	Arrival Time	Computation Time	Deadline
i.	Early	Short	Critical
ii.	Intermediate	Short	Sufficient
iii.	Late	Short	Critical
iv.	Early	Medium	Critical
v.	Late	Long	Sufficient

These fuzzy rules which have been created are attached with a standard fuzzy union in order to cumulate the membership functions of the three parameters are attached with each task. Hence, we obtain 29 a crisp output value for each task. The crisp output value acts as a runtime priority for task allocation by virtual machines.

When applying these inputs with the membership function, it is obtained the following values:

i) The crisp input for arrival time = 5.0,

The corresponding fuzzy outputs with membership values are;

Early: 0.45 , Intermediate: 0.56, Late: 0.78

ii) The crisp input Computation time =7.0,

The corresponding fuzzy outputs with membership values are;

Short: 0.30, Medium: 0.75 , Long: 0.00

iii) When crisp input Deadline =25.0,

The corresponding fuzzy outputs with membership values are;

Critical: 0.00

Normal: 1.00

Sufficient: 0.00

The results generated only four from these rules:

i. If Arrival time is Early (5.0) AND Computation time is Short (0.45) AND Deadline is Normal (78.00), THEN Runtime priority is High.

ii. If Arrival time is Early (0.25) AND Computation time is Medium (0.75) AND Deadline is Normal (1.00), THEN Runtime priority is Normal.

iii. If Arrival time is Intermediate (0.8) AND Computation time is Short (0.30) AND Deadline is Normal (1.00), THEN Runtime priority is High.

iv. If Arrival time is Intermediate (0.80) AND Computation time is Medium (0.75) AND Deadline is Normal (1.00), THEN Runtime priority is Normal.

VI. EXPERIMENTAL RESULTS

In this implementation, Javabasedtoolkits, namely; CloudSim and HDFS Simulator for simulation. CloudSim for testing the proposed allocation algorithm and simulating of cloudcomputing environments and infrastructure while HDFS simulator employed for replication. The evaluation of the proposed framework takes place via simulating the cloud in terms of a performance and availability. The following components were created:

The Namenode acts as metadata server.

Datanodes acts as file servers in the format of XML, residing in Namenode can store instances of copies.

Various sizes of files employed are: 100 kb, 150 kb, 200 kb, 300 kb, 500 kb, 800 kb, 1 kb, 2 kb, and 3 mb. Ten files are uploaded to employ single data de-duplicators. It was able to reduce the interval between to 85.75% and 94.20% with respect to processing time. When number of files increases, it has created the four de-duplicators which provides the processing time.

Table 5: Processing time saving for upload and update

Number of files	Upload		Update	
	Five	Ten	Five	Ten
10	85.75	90.85	41.78	75.02
100	94.20	97.55	61.79	75.34
1000	91.40	95.58	63.78	82.09
10000	60.10	79.71	75.25	96.17

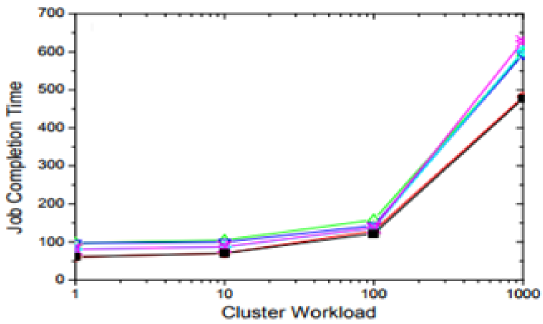

Figure 6: Evaluation based on job completion time, where brown line signifies that the proposed algorithm completes faster

In order to evaluate the proposed algorithm, it has compared with different task scheduling algorithms, namely, MaxCover-BalAssign (MB), Hadoop Default Scheduler (HDS) and Delay Scheduling (DS). It is evident that the proposed scheduling has faster job completion time. It was detected that the response time varies when the system load increases and proved that proposed algorithm bestowed a better performance at a higher level of system load.

VII. CONCLUSION AND FUTURE WORK

The cloud infrastructure deals with several nodes which eventually lead to queueing of systems, where huge amount of servers displays inconsistency in terms of processing arrival time and service time. Considering the potentially high number of servers in a cloud system, proposed a fuzzy scheduling algorithm is efficient and easy-to-implement to get a solution. The proposed framework is provided solution to augment the infrastructure performance of cloud. To employ more efficiency in storage would be the future scope of work.

Retrieval Number: D5286118419/2019©BEIESP
DOI:10.35940/ijrte.D5286.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

REFERENCES

1. I. Raicu, I. Foster, Y. Zhao. “Many-Task Computing for Grids and Super Computers”, Invited Paper, IEEE MTAGS 2008.
2. Udendhran R, “New FW to Detect and Prevent DoS Attacks in CC Envt.” AJCSSIT, 4 : 12 (2014) 87 – 9, ISSN 2249-5126.
3. J. Dean and S. Ghemawat. MapReduce: Simplified DP on large clusters. In Proc. of the 6th USENIX Symp.on OS Design and Implementation (OSDI), pages 137–150, 2004.
4. M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly.Dryad: DDP programs froms equential building blocks. In Proc. of the 2nd European Conf. on CS (EuroSys), 2007.
5. Apache Hadoop. http://hadoop.apache.org.
6. Udendhran R, A Hybrid Approach to Enhance Data Security in Cloud Storage, ICC '17 Proc. of the Second Intl. Conf. on IoT and CC at Cambridge University, United Kingdom — March 22 - 23, 2017, ISBN: 978-1-4503-4774-7 doi>10.1145/3018
7. Udendhran R, K Muthuramlingam, A dynamic data-aware scheduling for MR in cloud, 2017 4th Intl. Conf..on Advanced Computing and Commn. Systems (ICACCS), DOI: 10.1109/ICACCS.2017.8014617
8. K. Liu, H. Jin, J. Chen, X. Liu, D. Yuan, Y. Yang, A Compromised-Time-Cost Scheduling Algorithm in SwinDeWCfor Instance-Intensive Cost-Constrained Workflows on a CC Platform, Int. J. High Perform. Computing Applns.
9. K. Kaur, A. Chhabra, G. Singh, Heuristics based GA for scheduling static tasks in homogeneous parallel sys. Intl. Jr. of CS
10. J. Kolodziej, F. Xhafa, Enhancing the genetic-based scheduling in computational grids by a structured hierarchical.
11. P. Sempolinski, D. Thain, A Comparison and Critique of Eucalyptus, OpenNebula and Nimbus,In Proc. of the 2010 IEEE Second Intl. Conf. on CC Technology and Science,CLOUDCOM ’10, IEEE Computer Society, Washington, DC, USA, ISBN 978-0-7695-4302-4, 417–426, doi:10.1109/CloudCom.2010.42, URLhttp://dx.doi.org/10.1109/CloudCom.2010.42, 2010, FGCS, 27 (8) (2011) 1035–1046, ISSN 0167-739X, doi:10.1016/j.future.2011.04.011, URL http://dx.doi.org/10.1016/j.future.2011.04.011.
12. B. Palanisamy, A. Singh, L. Liu, and B. Jain. Purlieus:locality-aware resource allocation for MapReduce in acloud. In Proc. of 2011 Intl. Conf. on HPC, Networking,Storage and Analysis (SC), 2011.
13. J. Dean and S. Ghemawat, “MR: Simplified DP on large clusters,” ACM Commn.., vol. 51, Jan. 2008, pp. 107-113.
14. M. Zaharia, D. Borthakur, J. Sen Sarma,K. Elmeleegy, S. Shenker, and I. Stoica. Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. In Proc. of 5th European Conf. on Computer systems (EuroSys), 2010.
15. D. Kreuter, “Where server virtualization was born”, Virtual Strategy Magazine, July 2004
16. Hive Performance Benchmarks. https://issues.apache.org/jira/browse/HIVE-396.
17. S. L. Faraz Ahmad and T. V. MithunaThottethodi., “MR with commn. overlap (marco).http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1412&context=ecetr, 2007.