IDS using Profit Based Scheduling in Cloud Computing

[1]Netrika, [2]Sheo Kumar
[1]PG Research Scholar, [2]Assistant Professor
[1]Department of Computer Science and Engineering,
[1]Swami Devi Dyal Institute of Engineering and Technology, Haryana, India

---

***Abstract* - Cloud computing enables a large range of users to access scattered, scalable, virtualized hardware and/or software infrastructure over the Internet. This paper discusses Task scheduling in cloud computing environment. By doing research and analysis of this problem, that aims at task scheduling with minimum total tasks completion time and minimum cost. CloudSim3.0.3 is employed to carry out and simulate the tasks assignment algorithm, and distributed task scheduling. a new clustering algorithm is proposed which is based on K-means. The efficiency of the algorithm with which intrusions are detected is around 90%-95%**

***Index Terms* - Cloud Computing, CloudSim, DARPA**

I. INTRODUCTION

Cloud computing refers to the provision of computational resources on demand via a computer network. User or clients can present a task, such as statement processing, to the service provider, such as Google, without really possessing the software or hardware. The clients computer may contain very little software or data (perhaps a minimal operating system and web browser only), helping as little more than a display terminal connected to the Internet. Since the cloud is the fundamental delivery method, cloud based applications and services may support any type of software application or service in use today.

Cloud computing builds on established trends for driving the cost out of the delivery of services while increasing the speed and agility with which services are deploy. It shortens the time from sketch out application architecture to real deployment. Cloud computing incorporates virtualization, on-demand deployment, Internet release of services, and open source software. From one viewpoint, cloud computing is nothing new because it uses approach, concepts, and most excellent practice that have already been establish. From another viewpoint, everything is new because cloud computing changes how we create, develop, deploy, scale, update, maintain, and receive for applications and the infrastructure on which they run.

II. RELATED WORK

In [6], The proposed scheduling approach in cloud employs an improved cost-based scheduling algorithm for making efficient mapping of tasks to available resources in cloud. This scheduling algorithm measures both resource cost and computation performance, it also improves the computation/communication ratio by grouping the user tasks according to a particular cloud resource's processing capability and sends the grouped jobs to the resource. Usually tasks are scheduled by user requirements .New scheduling strategies need to be proposed to overcome the problems posed by network properties between user and resources. New scheduling strategies may use some of the conventional scheduling concepts to merge them together with some network aware strategies to provide solutions for better and more efficient job scheduling.

In [7] presents a novel heuristic scheduling algorithm, called hyper-heuristic scheduling algorithm (HHSA), to find better scheduling solutions for cloud computing systems. The diversity detection and improvement detection operators are employed by the proposed algorithm to dynamically determine which low-level heuristic is to be used in finding better candidate solutions. To evaluate the performance of the proposed method, this study compares the proposed method with several state-of-the-art scheduling algorithms, by having all of them implemented on CloudSim (a simulator) and Hadoop (a real system). The results show that HHSA can significantly reduce the make span of task scheduling compared with the other scheduling algorithms evaluated in this paper, on both CloudSim and Hadoop.

In [8] presents a scheduling strategy on load balancing of VM resources based on genetic algorithm. According to historical data and current state of the system and through genetic algorithm, this strategy computes ahead the influence it will have on the system after the deployment of the needed VM resources and then chooses the least-affective solution, through which it achieves the best load balancing and reduces or avoids dynamic migration. At the same time, the aut or brings in variation rate to describe the load variation of system virtual machines, and it also introduces average load distance to measure the overall load balancing effect of the algorithm.

III. CLOUDSIM

The CloudSim software framework and architectural components. Earlier versions of CloudSim used SimJava discrete event simulation engine [5] as a provider of core functionalities required for higher-level simulation frameworks, such as queuing and processing of events, creation of system components (services, host, data center, broker, virtual machines), communication between components, and management of the simulation clock. Nevertheless, this layer has been removed in order to allow advanced operations not supported by SimJava.

A Cloud provider, who wants to study the efficacy of different policies in allocating its hosts, would need to implement his strategies at this layer by programmatically extending the core VM provisioning functionality. There is a clear distinction at this layer on how a host is allocated to different competing VMs in the Cloud. A Cloud host can be concurrently shared among a number of VMs that execute applications based on user-defined QoS specifications. Similarly, a cloud application developer can evaluate performance and perform some workload profiling by modeling the application characteristics, and considering network characteristics among different data centers and users using features presented in this level.

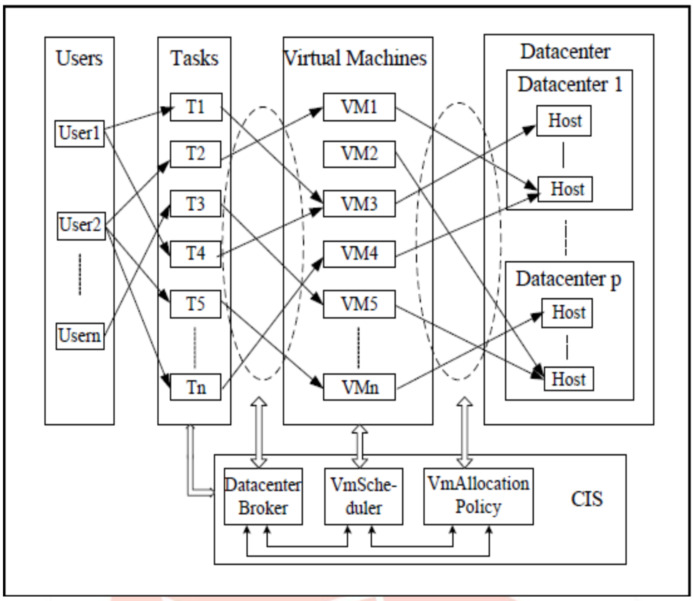


Fig 3.1 Cloud Sim

IV. Proposed Work

Cost of every individual resources use is dissimilar. The priority level can be sorted by the ratio of task's cost to its profit. For simple management, three lists can be built for the sorted task, each list has a marker of priority level such as HIGH, MEDIUM and LOW. Cloud systems can take someone out from the highest priority list to compute. Maps should be scanned every turn to transform the priority level of each task. Parameters are defined as followed:

Ri,n: The ith individual resources by the nth task.
Ci,n: The cost of the ith individual use of resources by the nth task.
Pn: The profit earned from the nth task
Ln: The priority level of the nth task.
The priority level of each task can be calculate as in formula , the total individual resources use is supposed to be n time so the priority level of the kth task is

$$\mathbf{L_n = \sum R_{i,n} \times C_{i,n} / P_n}$$

Equation 4.1

Algorithm

Priority based task Scheduling algorithm in Cloud Computing
This section describe how to design an algorithm of activity based costing method in cloud computing. The specific algorithm is described as followed:

Algorithm of pre-process:
For all available tasks **do**
    Calculate their priority levels L_k
End for
For every L_k **do**
    Sort them and then put them into an appropriate list
End for
While the system is running **do**
    If there is new task coming **do**
        Calculate its priority and then put it into an appropriate list
    End if

End while

Algorithm of process:
Do pre-process **as a thread**
While the system is running **do**
    If every list is not empty **do**
        Process the task which has the highest priority
        Scan every list to modify the priority base on the restrictive conditions
    End if
End while

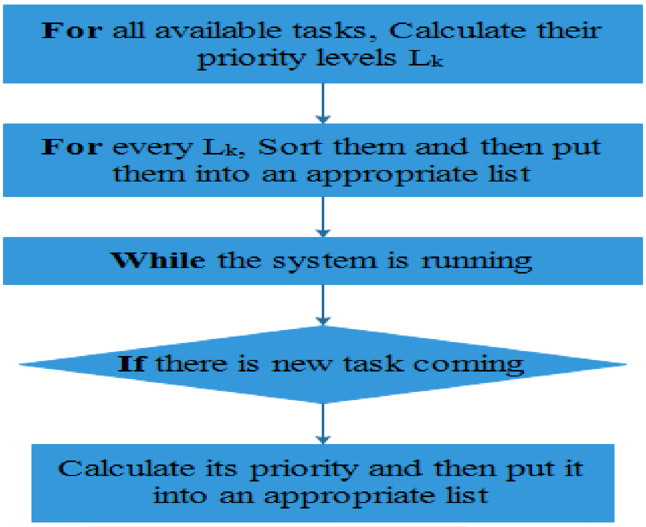


Fig 4.1(a) Flow Chart

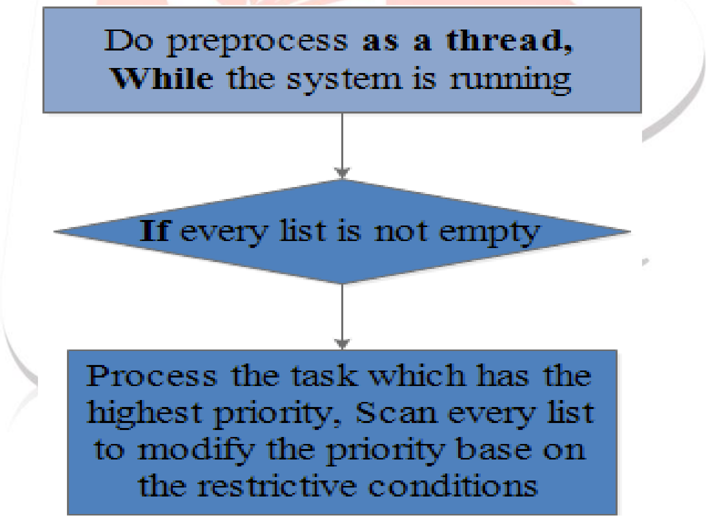
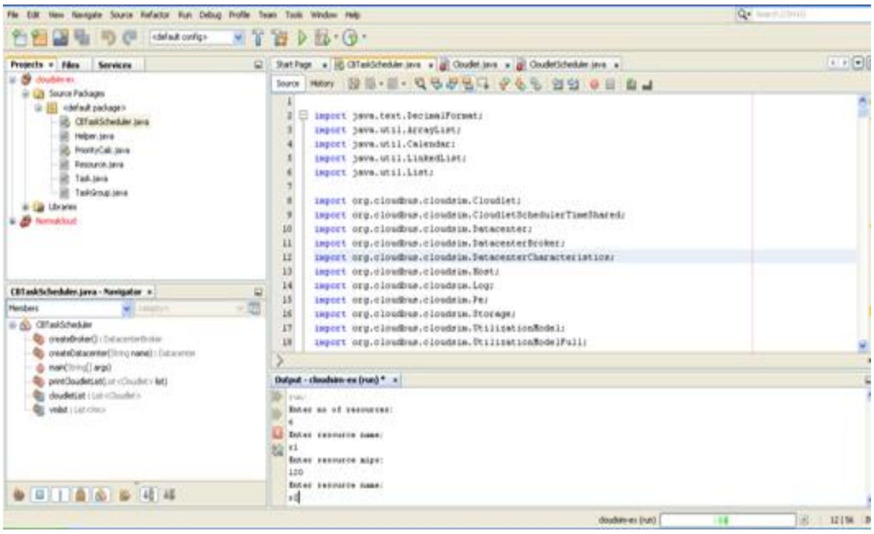


Fig 4.1(b) Flow Chart



We have tabulated the for the results in Table 4.1 and Table 4.2 below.

Table 4.1 Mips of cloud resources:

Resource	MIPS
R1	120
R2	131
R3	153
R4	296
R5	126
R6	210

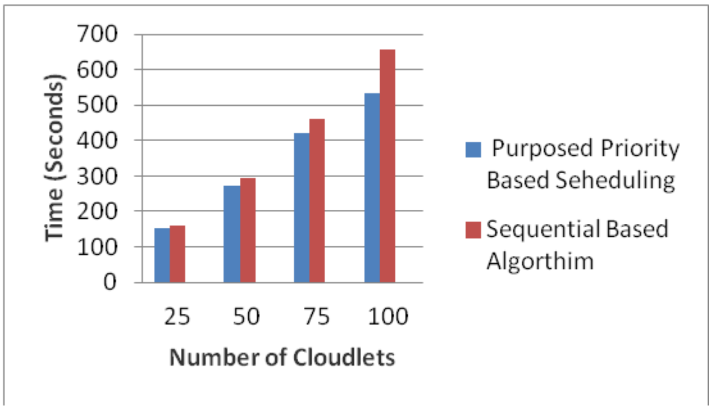
Table 4.2Process Time in Seconds
Process Time in Seconds

No of Cloudlets	**Purposed Priority Based Algorthim**	**Sequential Algorithm**
25	152.23	159.1
50	272.34	294.01
75	422.4	460.03
100	532.12	657.5

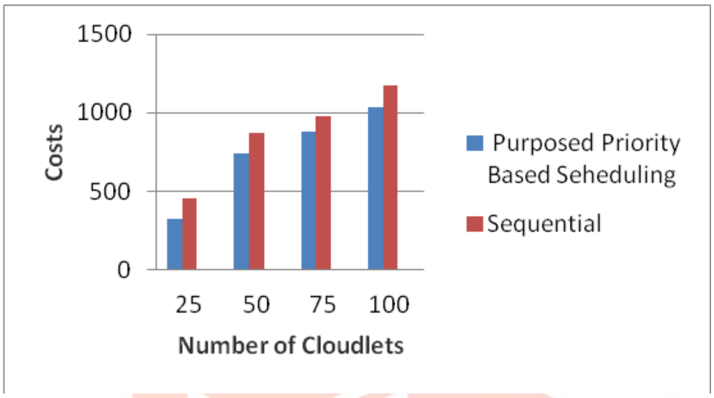
Table 4.3 Process Cost In Rs**.**

No of Cloudlets	**Purposed Priority Based Algorithm**	**Sequential Algorithm**
25	324.21	453.31
50	745.02	875.61
75	881.45	978.61
100	1034.41	1178.31

We have compared the results for processing time and processing cost for various numbers of Cloudlets namely 25, 50, 75 and 100. a) Fig. 4.1 Profit Based Task scheduling algorithm with and without grouping on the basis of time taken for completion of the tasks for the values in Table 4.2 From the below Figure it can be seen that for Profit Based Task Scheduling the time taken to complete tasks after grouping the tasks is very less when compared with time taken to complete the tasks without grouping the tasks.
Graph. 4.2 compares sequential based algorithm, scheduling algorithm with and without grouping on the basis of cost spent for processing the tasks for the values in Table 4.3
From the above Figure it can be seen that for sequential based algorithm Scheduling the processing cost spent to complete tasks after grouping the tasks is very less when compared with the processing cost spent to complete the tasks without grouping the tasks.



Graph 4.1 Profit Based Task Scheduling for Processing Time



Graph 4.2 Profit Based Task Scheduling for Processing Cost

Analysis Data Set

We have performed reduction of dimensionality of the DARPA data set. It is an important step, not only to reduce the complexity of the training process but also to gain an insight as to which network connection features are significant for the process of any network intrusion detection. Having done that, these features are real numbers on different scales collected. TCPDUMP files are converted to arff and csv format file.

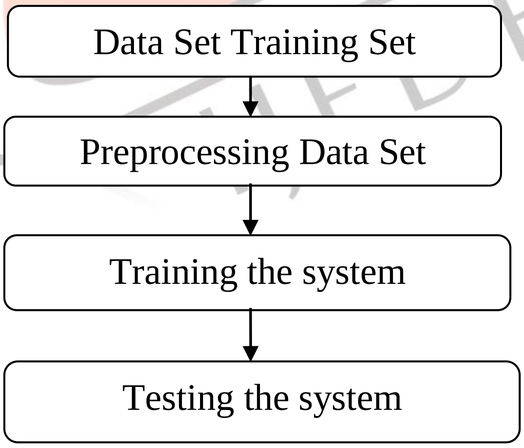


Figure 4.3: Implementation phase

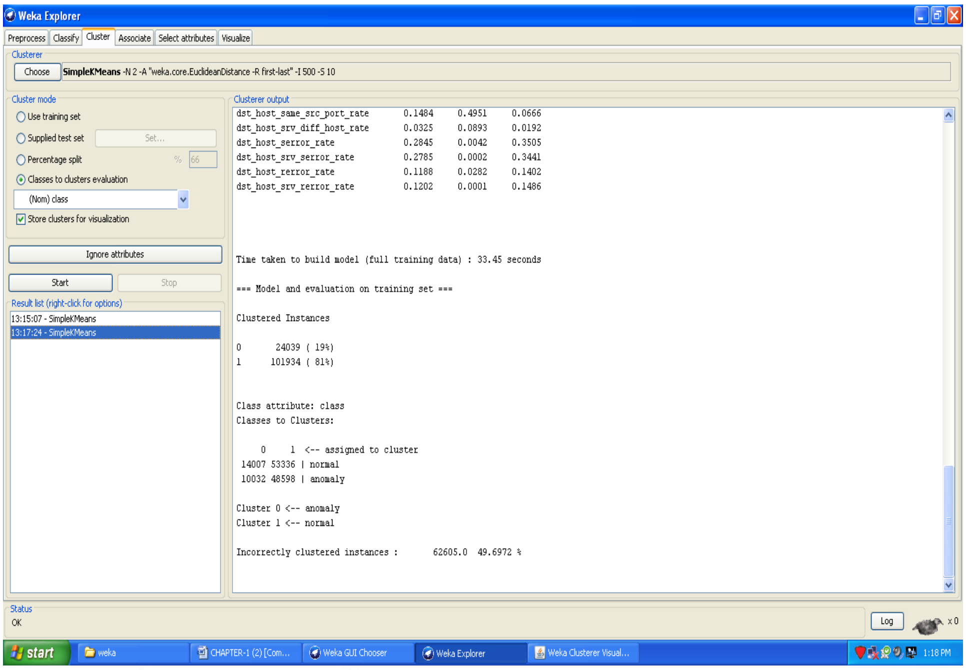


Fig 4.4 WEKA Clustering Class Selection

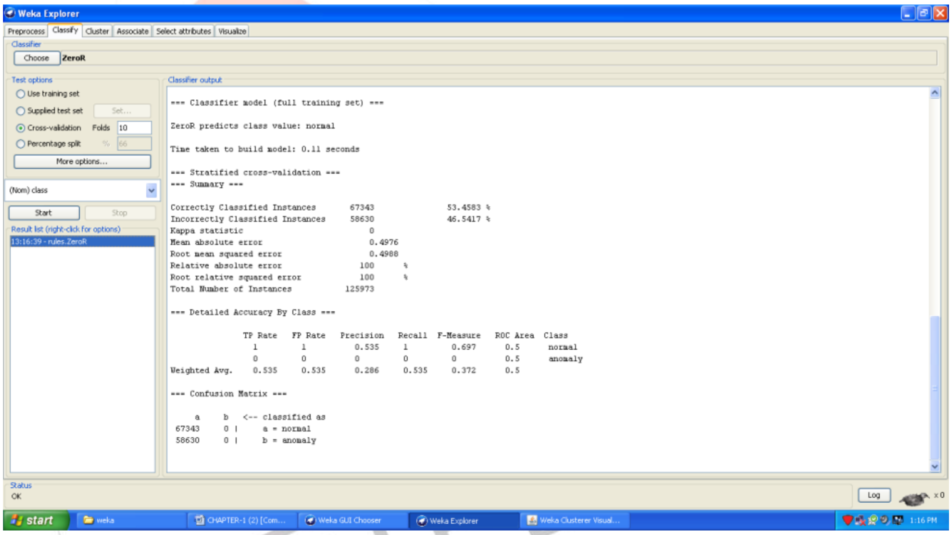


Fig 4.5 WEKA Clustering Class Selection

V. CONCLUSION

The accuracy of this algorithm depends on the training data taken. If the number of instances of a particular type is equal to that of the other then there is an algorithm based on the k-mean clustering for analyzing program behavior in intrusion detection is evaluated by experiments. The preliminary Experiments with the Darpa Data Set audit data have shown that this approach is able to effectively detect intrusive program behavior.

REFERENCES

[1] Catteddu, D. and Hogben, G. (2009). *Cloud Computing: benefits, risks and recommendations for information security.* Technical Report. European Network and Information Security Agency.

[2] Chun B, Culler D. “Market-based proportional resource sharing for clusters”. Technical Report CSD-1092, University of California at Berkeley, Computer Science Division, January 2000.

[3] Creeger, M. (2009). "CTO roundtable: cloud computing," Comm. of the ACM, vol. 52Creswell, J. W. (2007): *Qualitative inquiry and research design : choosing among five traditions*. 2nd ed., Sage Publications, Thousand Oaks, Calif. Cloud Security Alliance. (2009). *Security Guidance for Critical Areas of Focus in Cloud Computing*.

[4] David Chappell “A short introduction to cloud Platforms”, sponsored by Microsoft corporation, Aug 2008.

[5] Antonio Corradi, Luca Foschini, Javier Povedano-Molina, Juan M. Lopez-Soler” DDS-Enabled Cloud Management Support for Fast Task Offloading”2012 IEEE978-1-4673-2713-8/12.

[6] Boonyarith Saovapakhiran, George Michailidis†, Michael Devetsikiotis” Aggregated-DAG Scheduling for Job Flow Maximization in Heterogeneous Cloud Computing”2011 IEEE 978-1-4244-9268-8/11

[7] Chun-Wei Tsai, Wei-Cheng Huang, Meng-Hsiu Chiang, Ming-Chao Chiang, and Chu-Sing Yang” A Hyper-Heuristic Scheduling Algorithm for Cloud” IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 2, NO. 2, APRIL-JUNE 2014.

[8] Jianhua Gu, Jinhua Hu" Jinhua Hu" A New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment"IEEE VOL. 7, NO. 1, JANUARY 2012.
[9] Cesar G. Chaves, Daniel M. Batista, Nelson L. S. da Fonseca" Scheduling Grid Applications on Clouds" 2010 IEEE
[10] Enda Barrett, Enda Howley, Jim Duggan" A Learning Architecture for Scheduling Workflow Applications in the Cloud" 2011 Ninth IEEE European Conference on Web Services978-0-7695-4536-3/11 $26.00 © 2011 IEEE.