Priority Based Task Scheduling In Cloud Environment To Achieve Energy Awarness

Atindrasinh P. Vaghela[1], Prof. Bela Shrimali [2]
[1, 2] Dept of Computer Engineering
[1, 2] LDRP - Institute of Technology and Research, Gandhinagar, India

Abstract- *Task scheduling is an exigent activity on Cloud especially for the tasks with different priorities. Not all jobs sent to Cloud server hold the priority. There are jobs which are more important than others such as real time activity, stock market type application and so on. It is the responsibility of the scheduling algorithm run on Cloud datacenter to arrange the incoming jobs based on their priority. Further, such scheduling algorithms may result into more energy consumption. Hence, in this dissertation we aim to (a) provide an efficient scheduling approach for priority based tasks and to (b) reduce energy consumption. We test our approach for different factors such as execution time and SLA violation.*

Keywords- Cloud Computing, task scheduling , Energy Awareness

I. INTRODUCTION

NIST[9] defines clouds definition as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.[9]

The main reason for the existence of different perceptions of cloud computing is that cloud computing, unlike other technical terms, is not a new technology, but rather a new operations model that brings together a set of existing technologies to run business in a different way. Indeed, most of the technologies used by cloud computing, such as virtualization and utility-based pricing, are not new. Instead, cloud computing leverages these existing technologies to meet the technological and economic requirements of today's demand for information.

Classification Task Scheduling

We are classifying scheduling methods in cloud environment generally into three groups: resource scheduling, workflow scheduling and task scheduling,. The entire classification of task schedule is portrayed in Fig. 1 Resource scheduling performs mapping of virtual resources among physical machines and workflow scheduling is done to schedule workflows constituting an entire job in a suitable order. Task scheduling methods may be centralized or distributed. It can be performed in homogeneous or heterogeneous environment on dependent or independent tasks. In centralized scheduling a single scheduler is there to do all mappings whereas in distributed, scheduling is partitioned among different schedulers. In case of distributed scheduling it has high implementation complexity. But here, processor cycles are saved; as the work load is distributed to partner nodes. Though centralized scheduling is easy to implement, it losses scalability and fault tolerance as it always has a bottleneck of single point of failure

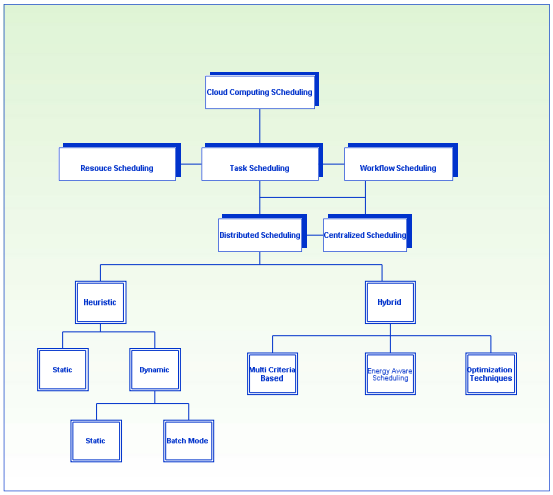

Fig. 1 Classification of Scheduling Methods in Cloud Computing Environment

This Paper presents an algorithm that tries to schedule the task in such an order that apart from considering only the priority of tasks, it tries to calculate expected executable time of different tasks on different services. The priority is calculated by considering different attributes of task such as User Level, Expected Priority, Length, Waiting Time etc.

The rest of the paper is section I introduction

II. RELATED WORK

The authors in[2] an algorithm that tries to schedule the task in such an order that apart from considering only the priority of tasks, it tries to calculate expected executable time of different tasks on different services.

The authors in[3] proposed a bandwidth-awared schedule algorithm in cloud computing by using simulated annealing based greedy. By introducing the simulated annealing algorithm, while at the same time combined with greedy selection policy. Algorithm considered the network bandwidth dynamic changes in computer clusters in the cloud computing environment. The results indicate that our algorithm can be more suitable in cloud computation environment.

The authors in [4] the importance of virtualized data centers and multicore processors for attaining energy efficiency in Cloud computing. It proposes an energy aware Cloud based model termed as Green Cloud Scheduling Model (GCSM). The proposed GCSM implements a scheduler unit called as Green Cloud Scheduler that makes task allocation and scheduling decisions considering the energy aware capability of the heterogeneous Cloud nodes and also considers the nodes that can fulfill the task completion time limits as imposed by the users.

The authors in [5] a new non-trivial and practical subclass of tasks, called urgent tasks. Briefly, a task is urgent if it is executed right after it is ready or it can only wait one unit time after it is ready. Practical examples of embedded realtime systems dealing with urgent tasks are all modern building alarm systems, as these include urgent tasks such as 'checking for intruders', 'sending a warning signal to the security office', 'informing the building's owner about a potential intrusion', and so on. By using propositional logic, we prove a new result in schedulability theory, namely that the scheduling problem for asynchronous and preemptive urgent tasks can be solved in polynomial time.

The authors in [6] a preemptive scheduling algorithm to schedule the tasks in a real-time system. According to the algorithm, the weights of the value density and urgency of a task impact on its priority are adjusted by two parameters p and q. Moreover, the parameter _ is properly chosen to avoid system thrashing. The experiment results of the simulations show that the algorithm is prior to the analogous algorithms.

III.PROPOSED METHOD

A. Problem Statement

To Schedule the task based on priority of tasks, it tries to calculate expected executable time of different tasks on different services and check the energy efficiency through scheduling. The priority is calculated by considering different attributes of task such as User Level, Expected Priority, Waiting time etc.

B. Scheduling Model

the following three criteria's are considered : (1) Higher priority tasks should be scheduled prior to the one's having lower priority. (2) Complete the task as early as possible which eventually will lower the cost of using resources of the service provider and also achieves QoS at user level. (3)Consider Dependent task and scheduling each task as per minimum amount of time and Calculate energy using equation.

1. Denomination of Services

As we are pretty much aware about the fact that in cloud computing, we have resources enclosed into cloud services. We are donating service set as P(q),so P(q)={P1,P2,P3… Pm} where $m \in I^+$, I^+ is positive set of Integers. Service P(z) is donated by a multi-tuple as follows: Parameters(P(z)) = {P_Code, PSP, PName, PAtt}, Where each one is defined as under:

1) P_Code is the ID;
2) PSP is the service provider;
3) PName represents the name as well as method of service.
4) PAtt represents attributes of the service. We define PAtt={Storing Capacity, Processing Elements,and Bandwidth}.

2. Task Denomination

Task is something i.e. more often regarded as that container which contains the needs of the users that are required to be fulfilled. In general, Tasks are organized as
:
Dependent tasks.
Independent tasks.

Assume the task in denoted as T(*k*), then, T(*k*)={T_1, T_2, T_3…..T_k}, Where $k \in I^+$, I^+ being the set of positive Integers. Now suppose W(*i*)∈ W(*k*). Then We define Task W(*i*) as:

T(*i*) = {T_Code, TUserLevel, TPriorityExpectation, TLen, TSubTime} Where:

(1) T_Code is the ID of W(*i*).
(2) TUserLevel denotes the privilege class or level of the user, we set them as TUserLevel={X, Y, Z};
(3) TPriorityExpectation is the Expected Scheduling priority of task T(*i*). In other words, it represents the urgency of the task. We define it as: TPriorityExpectation∈{VI, I, M, L} Where: VI= Very Important. I= Important. M= Medium. L= Low.
(4) TLen is the length of task T(*i*). Here length corresponds to the unrolled loop instructions in taskT(*i*).
(5) TSubTime is the submission time of T(*i*).

3. Priority of Tasks

Priority of tasks having different parameter. Here,we will mainly focus of priority of taks on the following:

User Level, Task urgency, Task Load and Time queuing up.
According to the attributes of task specified above, we will corresponds the User Level with *TUserLevel*. Task Urgency is shown via *TPriorityExpectation*. Task Load will correspond to *TLen*. And the Time Queuing up (TQU) can be easily calculated by current time and *TSubTime*.With the introduction of TQU, priority of the task can be dynamically changed.

In order to calculate priority that includes the attributes specified above, we will use the concept of "Standardization" that will merge these different attributes together to form a single attribute. Standardization is really beneficial when we want to converge wide range data within a certain range. In order to standardize the priority, we will apply the following formula:

$$\mathbf{SDZ(i)} = \frac{\mathbf{StandAtri(i) - LowestVal}}{\mathbf{HighestVal - LowestVal}}$$

Where SDZ(i) is the Standardization resultfor task Ti. SA_i is the special attribute value of task T(i).HighestVal and LowestVal are the highest and lowest value of special attributes respectively from all tasks.

Now we can use this standardization formula to calculate priority as

- **PRIOR(i)** = α*SUL(i) + β*SPE(i) + γ*SLen(i) + δ*STQU(I)

Where PRIOR (i) is the calculated priority of task T(i). α,,, Represents load factor of priority, and $\alpha+\beta+\gamma+\delta=1$. Perhaps, changing load will make an impact on the priority. SUL(i), SPE(i), SLen(i), STQU(i) shows corresponding standardized values of TUserLevel, TPriorityExpectation, TLen, TSubTime respectively.

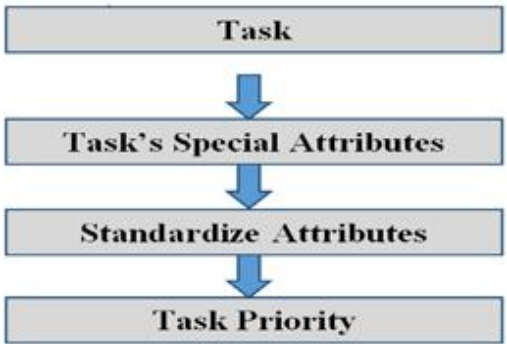

Fig2. Priority of Tasks[2]

4. Selecting Appropriate Service

Cloud service provider main goal is increase profit without affecting SLA . So Cloud provider want to reduce EET(expected execution time). This can be achieved only if execution time of the tasks is minimized. Suppose EET be the expected Execution time for a given task T_i having length TLen on service P(z) (provided P(z) has no load when T(i) was submitted). Then:

$$EET(Ti) = \frac{\mathbf{TLen}}{\mathbf{P(z)MIPS}}$$

Where MIPS is the speed of computer's processor elaborated as Million Instructions per Second.

Now, for cloud provider, to earn maximum profit, ETT (Wi) should be Minimum. Hence If CET is Cost of executing task on a service then,

- **CET(Wi)** = min[EET(Wi)]

In order to find min[EET(Wi)], The following algorithmic approach can be used which tries to find the minimum completion time of each task on different services. Moreover, certain definitions that are required to be known in advance are illustrated as follows:

Definition 1.ETM: stands for "EXPECTED TIME MATRIX" is the matrix that is stored on a central mapping system or scheduling system. The basic function of this matrix is that it includes expected time for execution of all tasks on all the services. The values along the row indicates the expected time

for executing(ETE) a task on different services, Whereas values along the column represents the ETE of different task on same service. Value(i,z) denotes a particular ETE value for Task T(i) on service P(z).

Definition 2. STTS: stands for "SHORTEST TIME TO START" is a matrix that contains the shortest time after which the service can start again after completing the assigned tasks. Value(z) denotes the shortest time for service P(z). This matrix is also placed on the scheduling system.
*Definition 3.*STTC: stands for "SHORTEST TIME TO COMPLETE" is basically a matrix that contains the minimum level of time required by all tasks on all services. This includes ETM of all tasks and STTS of all services. The values along the row indicates the STTC of a task on different services, whereas those along the column indicates the STTC of different task on a single given service.

- **STTC(i,z) =** ETM(i,z) + STTS(z)

5. Dependent task and Energy Consumption calculation

If one task is dependent to another task then this task can be schedule after that task.

Consider Dependent task and scheduling each task as per minimum amount of time and Calculate energy using given equation as below:

Energy consumption Calculation equation $E_{[11]}$

$$E = \sum_{i=1}^{m} ETVi * Ai + (M - ETVi) * Ii$$

Where $Ai=10^{-8}*(MIPS)^2$ J/MI
Ii=0.6*Ai
ETVi= Execution Time of ith VM
Ai=Energy consumption of ith VM in an active state
M=Makespan
Ii=Energy consumption of ith VM in an ideal state

C. Proposed Algorithm

Step1: For given set of Task(T) Evaluate special attribute of task Wi.
Step2: Standardize the value of special attributes as described in section 3.
Step3: Evaluate the priority of task Ti using the formula stated in section 3.
Step4: Sort Task set Task(T) according to priority.
Step5: Find EET(Ti) using the formula stated in section 4.
Step5: Create and initialize ETM and STTC matrix.
Step6: Evaluate STTS for each service.
Step7:Find Min[EET(Ti)] using the formula sated in section 4.
Step8: Consider dependent task
Step9: Provide particular task to VM considering dependency
Step10: Calculate energy consumption using the formula stated in section 5.

IV. EXPERIMENTAL EVOLUTION

In the experimental evalution we have consider various parameter. Let's take one example of Three phases. Put some values of different parameter and calculate it using given formula in 3 section.

First Phase example :

Higher priority tasks should be scheduled prior to the one's having lower priority.

T_C	TUL	TPE	TLen	TST	SUL(j)	SPE(j)	Slen(j)	STQU(j)
T1	25	66	3750	10	1	1	0.39286	0.81818
T2	20	60	2000	12	0.5	0.89286	0.14286	1
T3	15	50	3000	9	0	0.71429	0.28571	0.72727
T4	25	15	1200	7	1	0.08929	0.02857	0.54545
T5	20	55	6500	2	0.5	0.803571	0.785714	0.09091
T6	15	45	8000	3	0	0.625	1	0.18182
T7	25	35	1500	5	1	0.44642 9	0.07142 9	0.36364
T8	20	10	3500	1	0.5	0	0.35714 3	0
T9	15	20	1000	8	0	0.17857 1	0	0.63636
T10	25	30	2000	2	1	0.35714 3	0.14285 7	0.09091

Continue….

α	β	γ	δ	PRIOR(I)		A_PRIOR(I)
0.3	0.2	0.2	0.3	0.824025974	T1	0.824026
0.3	0.2	0.2	0.3	0.657142857	T2	0.657143
0.2	0.25	0.35	0.2	0.424025974	T5	0.555844
0.1	0.4	0.2	0.3	0.305064935	T7	0.512662
0.1	0.4	0.2	0.3	0.555844156	T10	0.427273
0.35	0.25	0.1	0.3	0.310795455	T3	0.424026
0.3	0.2	0.2	0.3	0.512662338	T6	0.310795
0.3	0.2	0.2	0.3	0.221428571	T4	0.305065
0.25	0.2	0.25	0.3	0.226623377	T9	0.226623
0.3	0.2	0.2	0.3	0.427272727	T8	0.221429

Table1: Existing algorithm first phase example

Here T_C, TUL, TPE, TLen, TSL shows corresponding values of tasks T(i) in T_Code, TUserLevel, TPriorityExpectation, TLen, TSubTime. A_PRIOR(I) is decending order priority of task

Second Phase example:

Complete the task as early as possible which eventually will lower the cost of using resources of the service provider and also achieves QoS at user level.

TLen	P(z)MIPS	EET(Ti)
3750	250	15
2000	200	10
3000	100	30
7000	350	20
6500	560	11.60714
8000	180	44.44444
1500	850	1.764706
3500	260	13.46154
1000	150	6.666667
2000	500	4

Table2: Calculate Expected Execution Time

ETM			STTS	SbT	STTC
TASK	E(S)	ET			
T1	S1	15	S1	1	16
T2	S1	10	S2	3	13
T3	S1	30	S3	4	34
T1	S2	19	S1	1	20
T2	S2	5	S2	3	8
T3	S2	6	S3	4	10
T1	S3	25	S1	1	26
T2	S3	10	S2	3	13
T3	S3	5	S3	4	9

Table3: Calculate shortest time to complete

ET is expected time.
S1, S2, S3 are different virtual machine or services.
SbT is submission time of tasks different VM.

	S1	S2	S3		Min[T]
T1	16	20	26		**16**
T2	13	8	13		**8**
T3	34	10	9		**9**

Min[T]		
16	T1	S1
8	T2	S2
9	T3	S3

Third Phase example:

Task T1 having highest priority, so it can be execute first. Here T1 task is dependent on T3 task, so as per criteria of dependency T1 task executed after completion of T3 task. It can be shows in below table:

Dependent	
T1	T3
Execution	
T2	
T3	
T1	

TASK	P(z)MIPS	Ai J/mi	Ii	ETV	Energy J	Total Energy J
T1	250	0.000625	0.000375	16	0.016375	0.027915
T2	200	0.0004	0.00024	8	0.0092	
T3	100	0.0001	0.00006	9	0.00234	

Table 3: Algorithm Third phase example

V. CONCLUSION

Energy can be efficiently managed and handle by efficient task scheduling of dependent and independent tasks. In this paper, we proposed efficient method of scheduling based on priority parameter for dependent task. The experimental results shows that method appropriately suitable for dependent tasks, to manage energy efficinently.

REFERENCES

[1] Cloud Computing Bible, Barrie Sosinsky

[2] Kaur, Pankajdeep, and Parampreet Singh. "Priority based Scheduling Algorithm with Fast Task Completion Rate in Cloud." Advances in Computer Science and Information Technology (ACSIT) 2.10 (2015): 17-20.

[3] Zhang, Jie, Xudong Zhu, and Bishan Ying. "A task scheduling algorithm considering bandwidth competition in cloud computing." International Conference on Internet and Distributed Computing Systems. Springer, Berlin, Heidelberg, 2013.

[4] Kaur, Tarandeep, and Inderveer Chana. "Energy aware scheduling of deadline-constrained tasks in cloud computing." Cluster Computing 19.2 (2016): 679-698.

[5] Andrei, Stefan, et al. "Optimal scheduling of urgent preemptive tasks." Embedded and Real-Time Computing Systems and Applications (RTCSA), 2010 IEEE 16th International Conference on. IEEE, 2010.

[6] Chen, Hui, and Jiali Xia. "A real-time task scheduling algorithm based on dynamic priority." Embedded Software and Systems, 2009. ICESS'09. International Conference on. IEEE, 2009.

[7] Mathew, Teena, K. Chandra Sekaran, and John Jose. "Study and analysis of various task scheduling algorithms in the cloud computing environment." Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on. IEEE, 2014.

[8] Zhang, Qi, Lu Cheng, and Raouf Boutaba. "Cloud computing: state-of-the-art and research challenges." Journal of internet services and applications 1.1 (2010): 7-18.

[9] Rajkumar Buyya et. el., "Cloud Computing: Principles and Paradigms", Wiley India Edition

[10] Mishra, Sambit Kumar, et al. "An adaptive task allocation technique for green cloud computing." The Journal of Supercomputing (2018): 1-16.