Journal of Theoretical and Applied Information Technology

30th April 2022. Vol.100. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

FUTURE HOST LOAD STATE DETECTION WITH HIDDEN MARKOV PREDICTION MODEL

M. AMARENDHAR REDDY[1,2] , K. RAVINDRANATH[3]

[1]Research Scholar, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India – 522502

[2]Asst. Professor, Department of Information Technology, B. V. Raju Institute of Technology, Narsapur, Medak, Telangana, India

[3]Associate Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India – 522502

[1] amar.madireddy@gmail.com, [2] amarenderreddy.m@bvrit.ac.in, [3] ravindra_ist@kluniversity.in

ABSTRACT

In virtual machine(VM) live migration process, the VM selection and host detection which are overloaded or underload is place crucial role. host load is dynamic nature, so detection of overload or under load is very challenging take, the live migration process is done with current host loads state, we proposed method, it use future host state, the future loads detection by using Hidden markov model, by this it avoids intermediate live migration of VM. Our suggested techniques are tested using CloudSim simulations on a variety of PlanetLab actual and random workloads. The experimental results reveal that our suggested algorithms outperform the other competitive algorithms in terms of service-level agreement violations, number of VM migrations, and other metrics.

Keywords: *VM Live Migrations. Host Selection Placement And Host Detection*

1. INTRODUCTION

In recent years, cloud computing has proved the main computing model in the IT sector, which utilises dissipated resources and has achieved tremendous popularity and rapid development in recent years. Encourages versatility in resource sharing and time access. To handle applications and services efficiently, it is essential to use the applicable models and resources Profile that is used to evaluate optimum models The best capital quantity for each workload.Migration of virtual machines is one of the most common methods Migration from VM live and resources is the most important reloading or rearranging resource strategy Migration of virtual machines is one of the most common methods. Migration from VM live and resources is the most important reloading or rearranging tools using the technique ,The data centre is available to maintain the resources provided. Live Living VM migration is known as a migrating technique Full operating system and related software One host to another, where no delay of its operation is noticed by the customer. It

plays a key role in helping Maintenance online, load balance and energy conservation Migration of virtual machines is one of the most common methods Migration from VM live and resources is the most important reloading or rearranging technique and live VM migration is the most

Resource management and distribution during VM migration has become more difficult in modern data centres, as hosting providers are fast growing and have strong complexities, Elasticity of resources and assured supply and trustworthiness. The success of apps in big virtualized data centres is therefore strongly influenced by the infrastructure of the data centre and the seamless network connectivity between VMs. By minimising relocation of VMs between hosts, the network's

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

connectivity costs can be reduced. Customers and service providers also need to develop a cloud storage system that not only minimises operating costs but also the overall network load.

The resource use of a data centre will shift over time as VMs and/or hosts have been created, or as the original hosts fail, or as existing VMs have been deleted. In order to balance the load or server restructuring according the SLA with end users and other problems, it is needed to reorganise the VMs and hosts. In cloud data centre management, host overload/unload identification, VM collection, and VM positioning are the three most significant research problems discussed in live migration.

A host can be in an overloaded or underground state in the first step of host identification. If a host is under-used, all VMs from this host will be migrated and the host is going to sleep/shutdown or the host is going to be found to be a suitable choice for the migration of VMs in future from overloaded hosts. On the other hand, certain VMs have to be chosen to switch from this host to other hosts when a given host is overwhelmed. The difficulties in host overload/underload identification include reducing energy consumption, minimising SLA infringements and preventing deterioration in results.

After a decision is taken to move VMs from a certain host, the VM collection stage selects one or moreVMs from the entire host range of VMs. You must transfer the chosen VMs to other hosts. VM discovery methods are different based on the criteria that are considered to pick the migrated VMs. A critical resource management choice is the difficulty for selecting one or more VMs for migration.The relocation method uses both the source and the destination hosts' network and CPU power as well as makes the VM inaccessible for some time. The output of other VMs on source and destination hosts is also influenced by the increased resource use .

Finally, the selected low-loaded hosts receive migrated VMs through a specific VM placement algorithm.A modern, optimal approach should be taken into account for several factors Algorithm of location of VMs, such as host resources (i.e. CPU, Memory, disc and network storage). Bandwidth), overall data centre energy usage and inter-VM flow. The objective of VM placing is Provide QoS for programmes operating on VMs as far as possible.Once a decision has been made to transfer a VM from a host source to a As a consequence of the selection process, destination host is known; The phase of migration will then take place locally or Breakthrough [30].

One of the host detection algorithms for determining whether the host is underloaded is used in the subload host detection protocol. There is no VM collection mechanism if the host is underloaded. Both VMs must be transferred to the underloaded host. The host is moved to an inactive place.

Effective implementation is the key contribution in this paper Host identification and VM positioning algorithms are studied. Existing algorithms take into account the power balance Present host state consumption and SLA breach. The current host use status and potential host use situation are taken into account by our proposed algorithms. Markov: Markov Basic algorithms in prediction are integrated in a comprehensive way.

The main contribution of this research is the introduction of efficient algorithms through the investigation of host detection and virtual machine deployment. At the current host state, existing algorithms consider the trade-off between power usage and SLA violation. Our suggested algorithms take into account both the current and future condition of host utilisation. In a thorough fashion, Markov-based prediction algorithms are embedded.

This paper organized as in second section of this paper introduces similar works. Section III outlines the forecasting model we propose. Our suggested host load detection algorithm, VM placement technique, and system architecture are presented.. The experimental results are discussed in section IV. The final thoughts and future directions are presented in Section VI.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. RELATED WORK

During VM migration, substantial research has taken place over the past two decades in the management and distribution of data-center resources. Many host algorithms for overload/unload detection have pro-positioned to optimise the use of computers and reduce energy demand alongside the data centre. Authors suggested the averaging algorithm based on the threshold (THR). It calculates the mean of the last CPU usage values and compares them with the threshold already specified. If the average of last n CPU use measurements is smaller than the indicated threshold, the algorithm detects under-loaded state. The algorithm for complex world is inappropriate.

Four policies in two groups were suggested in [3]–[5] authors. The first group is algorithms focused on the adaptive use threshold that include two policies: MAD and InterQuartile (IQR). These policies automatically change use rates on the basis of a statistical study of historical data collected over the VM's existence. The aim is to change the value of the higher threshold depending on the strength of the divergence from the use of the CPU. MAD is described as a statistical measurement dispersion of distributions without a better output Medium to variance. Variance. It's also a stronger estimator Compared to survey variation or standard deviation size.MAD's biggest drawback is the size of the Range is unacceptable for a limited number of outliers. Another mathematical dispersion measurement is IQR. It is referred to as the half or fifty, meaning the gap In descriptive statistics between the third and first quartiles. The host overload in this group is badly predicted.

The second group is algorithms focused on regression, which include both local regression (LR) and robust local regression (RLR). These depend on the forecast for potential use of the CPU. They predict host overload best but are more dynamic. LR is a curve-friendly solution that displays the data pattern. A host is crowded if the overall time for migration is nearer to the trend than a security margin.

In the literature[6]–[11],[36], several other algorithms were offered for the identification of host load. The majority of current algorithms depends exclusively on the statistical or dynamic threshold of the data centre's historic data. In this article, we use historical information to develop a probabilistic model that can more accurately forecast potential hosts

The aim of researchers' [12] focus is to average traffic latency reductions; a traffic-conscious VM placement algorithm has been suggested to achieve this goal.Two versions, known as partitioned and global traffic models, were proposed. In the divided model, the only communication permitted is that between the VMs in the same partition, while communication on the VMs in the same partition with a constant flow rate is not limited in the global traffic model.

The VM positioning problem was proposed in [12] as a multi-objective optimisation problem with a view to minimising net waste, consumption of energy and thermal dissipation costs. In order to find alternatives for assigning VMs, the authors suggest an improved genetic algorithm with a fluffy, many purpose assessment.

Researchers in [14] suggested an algorithm for VM placing Centered on the meta-heuristic Ant Colony Optimization (ACO) where the placement is dynamically calculated Modeling workload consolidation with the existing load Issue as a multifaceted binary packaging case Issue (MDBP). The objective is to pack the This algorithm Less servers for VMs. The algorithm needs information both workload and associated specifications for resources For the placement calculation.

Authors of[15] formulated a VM positioning issue to mitigate SLA infringements, resource overall waste, and power usage as a multi-objective Ant Colony Optimization (ACO) algorithm. Each ant builds a solution to pick VM to its target server in the ACO algorithm. The solution built is estimated by its appropriate mechanism combining SLA breaches, resource usage and energy consumption.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The authors in [16] suggested a joint energy knowledge and application-conscious VM positioning approach based on the principle of multi-objective optimisation by examining a balance between the energy usage of the server and the energy consumption of the contact network. The Algorithm strives to satisfy the server-side restrictions, limit the transfer of network data and reduce energy usage in data centres.

[17] The GM algorithm has been implemented by researchers to deal with the issue of VM positioning, which involves reducing energy use and contact between hosts.A VM placement model with two functions is presented to the authors. The first feature is a linear function of its working load that indicates the energy a server consumes and the energy it uses while the server is idle. The second is the quantity of data between VMs, which shows the network's energy consumption.

A hybrid genetic algorithm was suggested in [18] authors (HGA). The HGA algorithm technique is used to effectively distribute VMs compared to genetics in [16]. To transform the suggested solution into a viable solution, a repair process is in place. This can be achieved by local optimisation and the resolution of current breaches to increase the overall efficiency of a response.

Researchers in[19] suggested VM placement family genetic algorithms to address the limitations VM placement Genetic approach[17],[18]. Genetic approaches. These are the limits High cycle time and premature convergence. [21] In [22] VM Scheduler positioning algorithm to researchers suggested Reduce time of application and server allocation of VM Optimize the use of resources. The algorithm shows Binary search tree (BST) resource list rather than They're in a queue representing them.

A host nominee that meets the needs of VM better.In [3] and [20] authors suggested to switch the VMs from overloaded host into underground host, or from underloaded host, for server consolidation. Power Aware best fit decrease algorithms (PABFD) for VM positioning. The algorithm selects the destination host for the migrated VM and allows the energy consumption to be reduced.The algorithm is based on the trading

The host use and minimal correlation (UMC) VM placement algorithms have been suggested by researchers to relocate VMs from overused hosts and host sub-used. The parameters considered are host use and the association between a VM resource and the host correlation VMs current. If its CPU use has the lower association with all VM CPU use on that host, the algorithm selects the host to receive the migrated VM.

Finally, both the VM location and host detection algorithms have a major drawback use current work to estimate feature load. The feature load estimation on probability of its load which is that they rely on the current host resource use. In order to create probabilities that could more accurately forecast future host load, we propose to use historical data. In order to reduce unnecessary VM migrations, for better SLA breaches, numbers of VM migrations and energy use in the entire system, the Markov based VM placing and host load detection approaches aim to include VM based on the actual and potential use of resources of host and VMs.

3. PROPOSED HIDDEN MARKOV MODEL

The markove model[22] find probability of random variable based on its previous random variable probability ,it is used for predict future value based on its previous value. This model used to predict future work load of host based on its previous loads,by this find effect live Vm migration decision , host loads dynamically change tofuture loads detection is major challenge ,to detect host is overload/under load based on upper threshold value. So host state change time to time its hidden behaviour of the host .So predict accurate state of host based on its load we use its internal state ,it is motitivate has to use hidden markov model

The observed variable V ,in markov model , discrete observed variable the sequence $V_1,V_2,V_3....V_n$. the observed variable mapping to CPU utilization a host,its mapping to m hidden state of host$\{H_1,H_2,...H_m\}$, its map to n states$\{S_1,S_2...S_n\}$.in our Proposed mode the hidden

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

states are under load(U),Overload(O) and normal (N),based on lower threshold and upper threshold value .based up on host load changing direction Underload(U), Underload to normal(UN). normal to Underload (NU), normal(N). normal to overload(NO). overload to normal(ON) and overload.

Based up historical data host state detection by using hidden markov model, the historical data of host maintain its log files, for our prediction take 10 sample of log files , the size of sample determine in algorithms in[3][4][5] ,the hidden state are determining using algorithm 1 in [37],periodically host sate updated.

The first-order chain model of hidden markov used to describe dynamic process. The conditional probability of (V) observed variable.Vn is its observed at time n .$P(V_1,V_2 \ldots V_n)$,using our markove model using historical 10 observation

$$P(V_n / V_{n-1},V_{n-2} \ldots V_1) \approx P(V_n / V_{n-1}) \quad (1)$$

$$P(V_n /,V_{n-1},V_{n-2} \ldots V_1) = \prod_{i=1}^{n} P(V_i / V_i - 1) \quad (2)$$

output observation oi depends only on the state that produced the observation V_i

$$P(O_i/V_n \ldots V_k \ldots V_1,O_1 \ldots O_k \ldots O_n)=P(O_i/V_i) \quad (3)$$

Our proposed model hidden states {U,N,O},observation states {U,NU,UN,N.NO,ON,O}

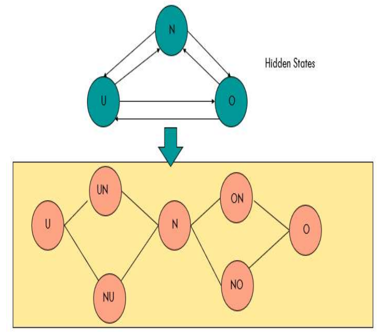


Fig 1. Hidden Markov Hidden States And Observer States

3.1 Architecture

In Fig 2 .cloud architecture used which proposed in[]. The host have multiple virtual machines allocated these are managed by VM manages ,the managing in terms of VM move one host to other host when that host is overloaded. each host have one host manages to maintain normal state of host,it detected host underload/overload also .the Perfomence of host measure in CPU utilization in terms of MIPS(Millions Instruction Per Seconds).

Data center is frequently interact with host ,in data center have VM placement procedure and our proposed host future sate predictive the host states with hidden markov predictive model

3.2 Proposed Host State

In ,Host sate are overload ,normal and underload .which is deter mine by upper load and lowerload threshold values, the proposed host state have U(underload),NU(normal to undeload),UN (under load to normal),N(normal).NO(normal to over load),ON (over load to normal)and O(over load).In algorithm 1,assign state based upon threshold values ,decrement and increment factors .the importance of these two factor is how the load of host is changing its state.

Journal of Theoretical and Applied Information Technology
30th April 2022. Vol.100. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm 1: Host load State Detection

Input: CPU(H_k)utilization of host *k*,*upper_threshold* (*UT*) and *lower_threshold(LT)*
Output : *S* current state of *host { U,NU,UN,N.NO,ON,O }*
If CPU(H_k) ≤ *LT* then
  if CPU(H_k)* Increment *_factor* ≥ *LT* // Decrement *factor*
    S=UN
  else
    S=U
If LT ≤ CPU(H_k) ≤ UT then
  If CPU(H_k)* Increment *_factor* ≥ *UT*
    S=NO
  Else **if** CPU(H_k)* Decrement *factor* ≤ *LT*
    S=NU
  Else
    S=N
If CPU(H_k) ≥ UT then
  if CPU(H_k)* Decrement *factor* ≤ *UT* // increment *factor*
    S=UN
  Else
    S=O
Return S

3.3 Host Detection Algorithm:

Host future load state detection based up hidden markov perdition if future state is { NO,O} its predict as overloaded if current_state in{U,NU} and Future_load in{U ,NU} than host is underloaded its shown in algorithm 1 called Hidden Markov Chain Host Underload/overload Detection(HMCHD).

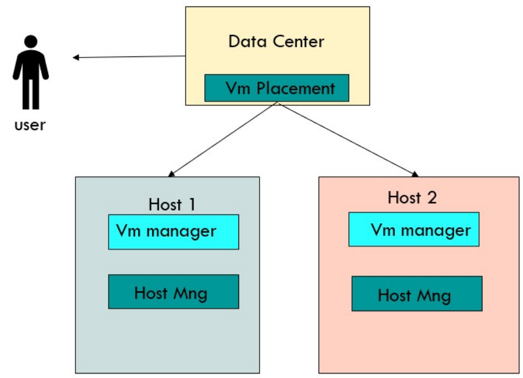


Fig 2. System Architecture

Algorithm 1 : Hidden Markov Chain Host Underload/overload Detection(HMCHD)

Input: *lower threshold(LT)=0.1,upper threshold(UT)=0.9 ,B{FOMCHSD or MadMCHD)*
Output*:underload(T/F),overload(T/F)*
M_D_underload=false
M_D_overload=false
While active_host=true do
If log.Length >=10
  Util=requires MIPS/Host MIPS
Switch(B)
Case FOMCHSD:Break
Case MADMCHD:
  *UT=1-s*MAD*
Current_sate=Host_state(util,LT,UT)
future_state=hidden_markov_state(current_state)
If future_state in{ NO,O}
  M_D_overload=True
Else if current_state in{U,NU} and Future_load in{U ,NU}
  M_D_underload=True
Return M_D_underload ,M_D_overload

VM placement

To VM migration the host need select host which are have under load,the sle

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm 3: markov hidden VM Placement (MHP)

```
Input:  Host_list(H) ,Vm_Selected(SVM)
Output: a Host received selected VM
Min_Power=none
Allocation_host=none
Foreach  h in  H  do
  If (h has_enough_resources for selected   SVM and host_active=T and host_after_allocation()!={O,NO}
Temphost_list1[]=add.h
Foreach h in   Temphost_list1[] do
future_state=h. hidden_markov_state(current_state)
If  future_state in{U,N,UN}
    Temphost_list2[]=add.h
If  future_state in{ON}
    Temphost_list2[]=add.h
Foreach h in  Temphost_list2[]
      Power=estimate_power(h,SVM)
      If (power< minPower) then
           minPower=Power
         Allocation_host=h
Return Allocation_host
```

Experiment setup

In dynamic environment investigation of perfomence of proposed algorithms are very difficult and time consuming, like cloud environment.so we use simulation (cloudSim) for overcome t these problems.it is suitable for modeling cloud computing environment than other simulations are simGrid,GangSim Grid-Sim[26]..[28] .the length of data taken 180 historical dat which I optimal in experiment we are using same experiment setup used in [2] with different workloads ,a data ceneter with V virtual hosts and J physical host,with different values of V and J of workload shown in table

workload	Host	VMs	Mean %	SD
Real (PlanctLab)	800	1053	12.32	17.08
random	55	55	--	--

Workloads characteristics

First our proposed host detection algorithm (HMCHD) compared with madMCHD. because it give better Perfomence among all existing benchmark algorithm namely,mad,lrr,madMCHD, lrr,thr and iqr,with 0.8 threshold [3]-[5],to study impact of VM selection algorithm namely mc(Maximum Correlation),mmt(MInimun Migration Time),mu(Maximum Utilization) and rc(Random choic)

3.4 Parameters

To evaluate the our proposed with existing methods in term of the following metrics

1) Due to migration, degradation of Perfomence (PD)
2) SLA violation time per active host (SLAPAH)
3) SLA violation (SLA)
4) Average SLA Violation (AvgSLA)
5) SLA overall Violation (OralSLA)
6) Number of Virtual Machine Migrations (NM)
7) Energy Consumptions (Eng)
8) Host shutdowns in number

Due to migration, degradation of Perfomence (PD) happen due to vm shift to one host to other host,so the consumer application will impact its performance in terms of CPU utilization its normally 10% of its CPU utilization[26]

$$\text{PD}=1/N\sum_{v=1}^{n}\frac{estimation\ cpu\ utillization\ (v)}{total\ cpu\ rqusted\ by\ v}$$

Where v is host, N is number of hosts

SLA violation time per active host (SLAPAH),application run on a host ,its performance dependent on that host utilization ,if the host utilization is 100% the the application run on this host its Perfomence wad degraded

$$\text{SLAPAH} = 1/H\sum_{h=1}^{H}\frac{Ts}{Taj}$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

T_{sh} = Total time Servered with 100% utilization,Taj=total number times its active

,H total number of hosts.

SLA violation calculated[5] as below

SLA violation= SLAPAH*PD

Average SLA Violation (AvgSLA) is mean difference of all Vm allocation MIPS and Required MIPS

(AvgSLA) $= \frac{\sum_{v=1}^{v} reqMIPS - \sum_{v=1}^{v} allocMIPS}{V}$

Where V indicates number of VMS

SLA overall Violation (OralSLA)= $\frac{\sum_{v=1}^{v} reqMIPS - \sum_{v=1}^{v} allocMIPS}{\sum_{v=1}^{v}(regMIPS)}$

Number of Virtual Machine Migrations (NM),if VM migration increase the Perfomence also decreased ,i

NM(VP,t1,t2)=$\sum_{j=1}^{j} \int_{t1}^{t2} Mig(VP)$

VP is current placement ,migration of VP Mig(VP) between t1 and t2 .

Energy consumption(VP,t1,t2)= $\sum_{j=1}^{j} \int_{t1}^{t2} w(VP,t)$

Host shutdown(HS)=$\frac{1}{n}\sum_{i=n}^{n} h_i$

Where h_i is Active host

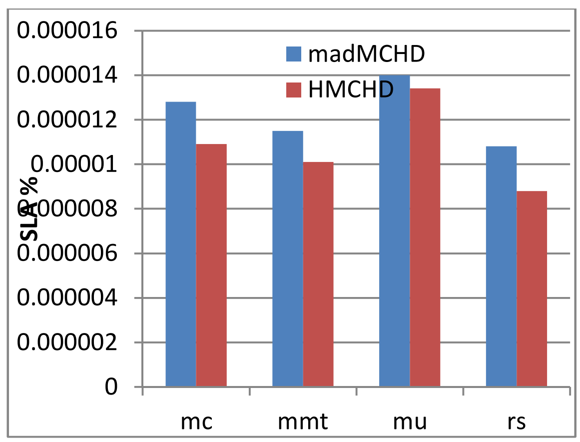
4 EXPERIMENT AND RESULTS:

In fig (a) and fig (a) show the improve the SLA violation with respect both workloads ,due decrease number of intermediate Vm migration, our algorithm HMCHD increase its Perfomence over madMCHD 14.84%,12.17%,14.29% and 18.52% on reals data set and 15.22%,13.80%,15.71%, and 18.33% in random data set for VM selection policies mc,mmt,mu and rs respectively.

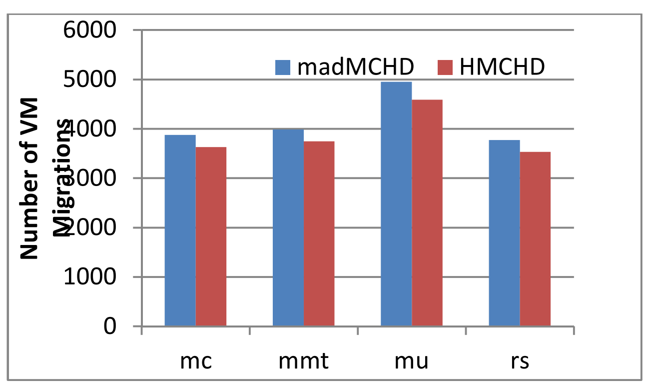
In fig (b) and fig (b) show the improve in terms of number of VM migrations with respect both workloads ,due decrease number of intermediate Vm migration, our algorithm HMCHD increase its Perfomence over madMCHD 6.32%,6.02%,7.33% and 6.46%on reals data set and 6.20%,6.03%,6.65% and 6.37% in random data set for VM selection policies mc,mmt,mu and rs respectively.

In fig (c) and fig (c) show the improve in terms of no of host shutdowns violation with respect both workloads ,due decrease number of active host , our algorithm HMCHD increase its Perfomence over madMCHD 7.18%,8.04%,9.71% and 7.43% on reals data set and 4.65%,5.71%,4.74% and 4.32% in random data set for VM selection policies mc,mmt,mu and rs respectively.

In fig (d) and fig (d) show the improve in terms energy consumption with respect both.

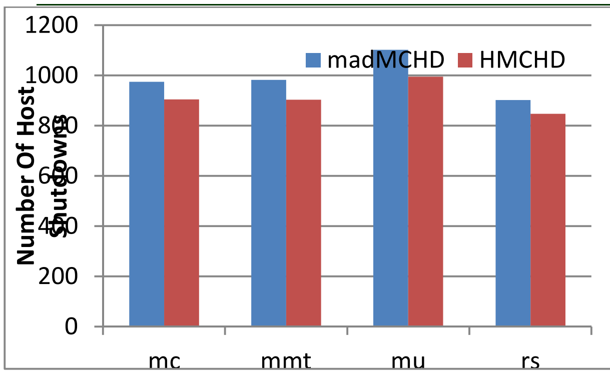


(a)

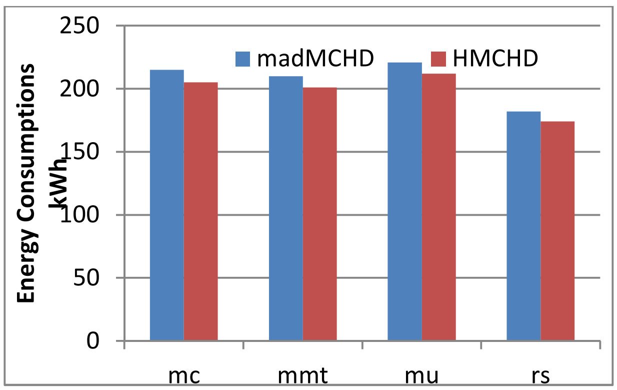


(b)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

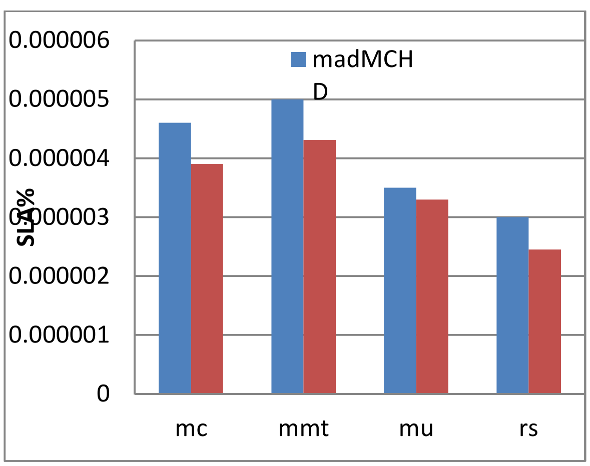


(c)

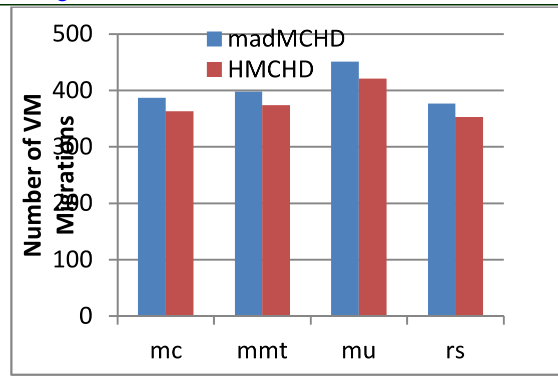


(d)

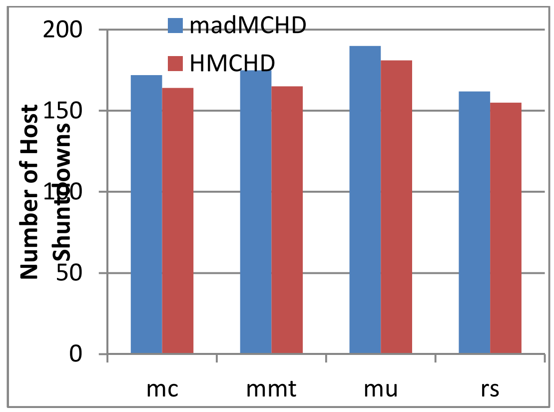
Fig 4 comparison Host detection algorithm with our Vm Selection policies on real data



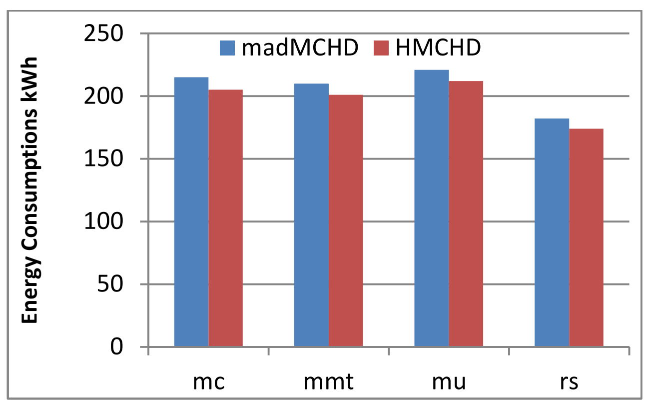
(a)



(b)



(c)



(d)

Fig 5. Comparison Host Detection Algorithm With Our Vm Selection Policies On Real Data

Workloads ,due decrease number of intermediate Vm migration, our algorithm HMCHD

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

increase its Perfomence over madMCHD 4.65%,4.29%,4.07% and 4.40% on reals data set and 4.65%,4.29%,4.07% and 4.40% in random data set for VM selection policies mc,mmt,mu and rs respectively

investigating our placeme algorithm ***MHP*** with combination of host detection algorithm madMCHD called as MHPD,[] in host selection and host detection name as madMCHD.MHPD improve Perfomence in term of SLA violation by 11.11%,14.29%,12.50% and 10.53% for real data set with VM selection polocies mc,mmt,mu and rs respectively

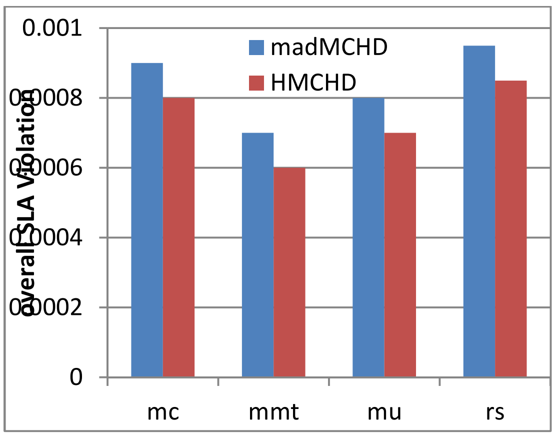


Fig 6. Comparison Host Detection Algorithm With Our VM Selection Policies On Real Data

5 CONCLUSION

With hidden markov prediction for detecting future host state .it avoid intermediate VM migration than markov model.in hidden morkov model the host in five state .so it predict host workload trends which leads to overload ,underload and normal host.

Our experiment results show that its improvement in term of SLA , number of VM migration ,host shout downs and energy consumption due the find trends of workload which leads to host state.

REFERENCES:

[1] E. Bauer and R. Adams, Reliability and Availability of Cloud Computing. Hoboken, NJ, USA: Wiley, 2012.

[2] A. Beloglazov and R. Buyya, ‘‘OpenStack neat: A framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds,’’ Concurrency Comput., Pract. Exper., vol. 27, no. 5, pp. 1310–1333, 2015.

[3] A. Beloglazov and R. Buyya, ‘‘Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers,’’ Concurrency Comput., Pract. Exper., vol. 24, no. 13, pp. 1397–1420, Sep. 2012.

[4] A. Beloglazov, J. Abawajy, and R. Buyya, ‘‘Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing,’’ Future Generat. Comput. Syst., vol. 28, no. 5, pp. 755–768, 2012.

[5] A. Beloglazov, ‘‘Energy-efficient management of virtual machines in data centers for cloud computing,’’ Ph.D. dissertation, Dept. Comput. Inf. Syst., Univ. Melbourne, Parkville, VIC, Australia, 2013.

[6] F. Farahnakian, P. Liljeberg, and J. Plosila, ‘‘LiRCUP: Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers,’’ in Proc. 39th IEEE EUROMICRO Conf. Softw. Eng. Adv. Appl. (SEAA), Sep. 2013, pp. 357–364.

[7] K. Maurya and R. Sinha, ‘‘Energy conscious dynamic provisioning of virtual machines using adaptive migration thresholds in cloud data center,’’ Int. J. Comput. Sci. Mobile Comput., vol. 3, no. 2, pp. 74–82, 2013.

[8] S. S. Masoumzadeh and H. Hlavacs, ‘‘An intelligent and adaptive threshold-based schema for energy and performance efficient dynamic VM consolidation,’’ in Proc. Eur. Conf. Energy Efficiency Large Scale Distrib. Syst., 2013, pp. 85–97.

[9] L. Salimian, F. S. Esfahani, and M. Nadimi-Shahraki, ‘‘An adaptive fuzzy threshold-based approach for energy and performance efficient

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

consolidation of virtual machines,'' Computing, vol. 98, no. 6, pp. 641–660, 2016.

[10] A. Horri, M. S. Mozafari, and G. Dastghaibyfard, ''Novel resource allocation algorithms to performance and energy efficiency in cloud computing,'' J. Supercomput., vol. 69, no. 3, pp. 1445–1461, 2014.

[11] E. Arianyan, H. Taheri, and S. Sharifian, ''Novel energy and SLA efficient resource management heuristics for consolidation of virtual machines in cloud data centers,'' Comput. Elect. Eng., vol. 47, pp. 222–240, Oct. 2015.

[12] X. Meng, V. Pappas, and L. Zhang, ''Improving the scalability of data center networks with traffic-aware virtual machine placement,'' in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–9.

[13] J. Xu and J. A. Fortes, ''Multi-objective virtual machine placement in virtualized data center environments,'' in Proc. IEEE/ACM Int. Conf. Green Comput. Commun. (GreenCom), Int. Conf. Cyber, Phys. Soc. Comput. (CPSCom), Dec. 2010, pp. 179–188. [14] E. Feller, L. Rilling, and C. Morin, ''Energy-aware ant colony based workload placement in clouds,'' in Proc. IEEE/ACM 12th Int. Conf. Grid Comput., Sep. 2011, pp. 26–33.

[15] F. Ma, F. Liu, and Z. Liu, ''Multi-objective optimization for initial virtual machine placement in cloud data center,'' J. Inf. Comput. Sci., vol. 9, no. 16, pp. 5029–5038, 2012. [16] D. Huang, D. Yang, H. Zhang, and L. Wu, ''Energy-aware virtual machine placement in data centers,'' in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2012, pp. 3243–3249.

[17] G. Wu, M. Tang, Y.-C. Tian, and W. Li, ''Energy-efficient virtual machine placement in data centers by genetic algorithm,'' in Neural Information Processing. Berlin, Germany: Springer, 2012, pp. 315–323.

[18] M. Tang and S. Pan, ''A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers,'' Neural Process. Lett., vol. 41, no. 2, pp. 211–221, 2015.

[19] C. T. Joseph, K. Chandrasekaran, and R. Cyriac, ''A novel family genetic approach for virtual machine allocation,'' Procedia Comput. Sci., vol. 46, pp. 558–565, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050915001544

[20] A. Beloglazov and R. Buyya, ''Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints,'' IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 7, pp. 1366–1379, Jul. 2013.

[21] S. K. Mandal and P. M. Khilar, ''Efficient virtual machine placement for on-demand access to infrastructure resourcesin cloud computing,'' Int. J. Comput. Appl., vol. 68, no. 12, pp. 6–11, 2013.

[22] E. Fosler-Lussier, ''Markov models and hidden Markov models: A brief tutorial,'' Int. Comput. Sci. Inst., Berkeley, CA, USA, Tech. Rep. TR-98-041, 1998.

[23] A. Beloglazov and R. Buyya, ''Energy efficient allocation of virtual machines in cloud data centers,'' in Proc. 10th IEEE/ACM Int. Conf. Cluster, Cloud Grid Comput. (CCGrid), May 2010, pp. 577–578.

[24] R. Buyya, R. Ranjan, and R. N. Calheiros, ''Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities,'' in Proc. IEEE Int. Conf. High Perform. Comput. Simulation (HPCS), Jun. 2009, pp. 1–11.

[25] S. Ray and A. De Sarkar, ''Execution analysis of load balancing algorithms in cloud computing environment,'' Int. J. Cloud Comput., Services Archit., vol. 2, no. 5, pp. 1–13, Oct. 2012.

[26] C. L. Dumitrescu and I. Foster, ''GangSim: A simulator for grid scheduling studies,'' in Proc. IEEE Int. Symp. Cluster Comput. Grid (CCGrid), vol. 2. May 2005, pp. 1151–1158. [27] A. Legrand, L. Marchal, and H. Casanova, ''Scheduling distributed applications: The SimGrid simulation framework,'' in Proc. 3rd IEEE/ACM Int. Symp. Cluster Comput. Grid (CCGrid), May 2003, pp. 138–145.

[28] R. Buyya and M. Murshed, ''GridSim: A toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing,'' Concurrency Comput., Pract. Exper., vol. 14, nos. 13–15, pp. 1175–1220, 2002.

[29] K. S. Park and V. S. Pai, ''CoMon: A mostly-scalable monitoring system for PlanetLab,'' ACM SIGOPS Oper. Syst. Rev., vol. 40, no. 1, pp. 47–65, 2006.

[30] S. B. Melhem, A. Agarwal, N. Goel, and M. Zaman, ''Selection process approaches in live migration: A comparative study,'' in Proc. 8th IEEE Int. Conf. Inf. Commun. Syst. (ICICS), Apr. 2017, pp. 23–28.

[31] S. B. Melhem, A. Agarwal, N. Goel, and M. Zaman, ''A Markov-based prediction model for host load detection in live VM migration,'' in Proc. 5th IEEE Int. Conf. Future Internet Things Cloud (FiCloud), Aug. 2017, pp. 32–38.

[32] S. Mustafa, B. Nazir, A. Hayat, and S. A. Madani, ''Resource management in cloud computing: Taxonomy, prospects, and challenges,'' Comput. Elect. Eng., vol. 47, pp. 186–203, Oct. 2015.

[33] N. Tziritas, C.-Z. Xu, T. Loukopoulos, S. U. Khan, and Z. Yu, ''Application-aware workload consolidation to minimize both energy consumption and network load in cloud environments,'' in Proc. 42nd IEEE Int. Conf. Parallel Process. (ICPP), Oct. 2013, pp. 449–457.

[34] X. Fan, W. D. Weber, and L. A. Barroso, ''Power provisioning for a warehouse-sized computer,'' ACM SIGARCH Comput. Archit. News, vol. 35, no. 2, pp. 13–23, 2007.

[35] A. Beloglazov and R. Buyya, ''Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers,'' in Proc. 8th Int. Workshop Middleware Grids, Clouds e-Sci., 2010, p. 4.

[36] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, N. T. Hieu, and H. Tenhunen, ''Energy-aware VM consolidation in cloud data centers using utilization prediction model,'' IEEE Trans. Cloud Comput., to be published.