Green Cloud: An Algorithmic Approach

K.Mukherjee
Department of Computer Science & Engineering
Birla Institute of Technology
Mesra, Ranchi,India

G.Sahoo
Department of Information Technology
Birla Institute of Technology
Mesra, Ranchi, India

ABSTRACT

Cloud computing has opened a new era to a shared IT infrastructure in which large pools of systems are linked together via the internet to provide IT services, such as different utility services of our day to day lives like tax calculation web services, weather information web services etc; which are on pay per user basis**.** It provides a virtual resource and therefore is not limited by the power and capabilities of local or remote computers. But cloud computing is in its nascent stage of progression. It has some serious demerits. The data centre hosting cloud application needs huge consumption of energy, which is subsequently responsible for the emission of carbon dioxide gas, which aggravates global warming-a threat to existence of life on earth. In this paper, we propose eco-friendly cloud computing which will not only mitigate global warming but also minimize operational cost by reducing power consumption. Here we have introduced noble algorithm for proper utilization of energy for cloud computing. We have validated our study by a set of experiments using Ubuntu's 10.04 Server editions.

Keywords

Cloud computing, eco-friendly, bee colony algorithm, ant colony algorithm etc.

1. INTRODUCTION

Cloud computing [16] has given a new orientation of computing based on "pay as you use" model. It is capable to provide massive computing or storage resources without the need to invest money or face the trouble to build or maintain such huge resources. The consumers only need to pay for using the services just like they do in case of other day to day utility services such as water, gas, electricity etc, which are on pay per user basis and hence our cloud computing resources are also on pay per user basis. Actually Cloud computing provides a platform for the execution of massive tasks on cloud instead of the execution of tasks on users' Personal Computers, Servers etc. It is very beneficial for small organizations that cannot afford huge investment on their IT sector but at the same time expect maximum benefit from this supporting industry in order to survive in today's complex competitive business world. Cloud computing can help such organizations by providing massive computing power, unlimited storage capacity, less maintenance cost, availability of useful web-services etc. As per Buyya et. all [7], "A Cloud is a type of parallel as well as distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumer". The definition clearly implies that there is a Service Level Agreement (SLA) between the provider and the consumer for getting services from cloud on pay per user basis. Actually Cloud computing offers three layers of abstraction such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS). SaaS provides different types of applications as a Service for the end user. It includes different useful web-services. PaaS provides a standard platform for better execution of application with proper exploitation of physical resources using Middleware Services. PaaS includes Database services, Middleware Services etc. In this endeavor we intend to throw light on Platform as a Service (PaaS). IaaS provides the hardware infrastructure of cloud consisting of physical machines like clusters, datacenters etc to provide resources in order to meet the consumer's request on demand. Again middleware of PaaS consists of core middleware and user level middleware. The core middleware exploits IaaS [7] at its best level to fulfill consumer's demand, whereas the user level middleware provides the access point of services as delivered by the core middleware. It is observed that maintaining IaaS including their cooling involves a lot of energy consumption, thereby enhancing global warming by emission of huge carbon oxide and other related gases. There is a growing concern across the world to arrest the trend towards global warming by reducing these parameters which are responsible for it. Thus in an era of ecological degradation, the future of cloud computing is bright provided it is eco-friendly in nature. In this endeavor, we intend to throw some light on eco-friendly cloud computing.

Organization of this paper is as follows: Related Work is discussed in section 2. Proposed Frame Work of Eco-Friendly Cloud Computing is discussed in section 3. Proposed algorithm is discussed in section 4. Section 5 gives details of the Experimental Results. Section 6 gives the Conclusion with a direction of the Future Work.

2. RELATED WORK

The potentiality of computing utility was first pointed by Leonard Kleinrock[1], one of the pioneer researcher, who seeded the idea of internet. He had predicted long back, regarding the wide use of "computer utilities" as our day to day utilities like "Electrical Utilities", "Telephone Utilities" etc. Today, his vision came into existence in the form of cloud computing. Cloud computing provides different utilities in the form of web-services, which are on pay per user basis. Most of the existing work in web service performance focuses on the latest trend of technologies and standards. Andreozzi et al. [2] present a model for rigorous representation of service characteristics. Gouscos et al. [3] presents a simple approach to model certain web service management attributes. Thomas et al.[4] represent distributed web service by modeling the flow of messages and methods in a web service transaction. Tu et al.[5] discuss design strategies to improve the performance of web service. Levy et al.[6] present an architecture and prototype implementation of performance management system for cluster based web service. Cardellini et al. [15] consider different categories of web applications, and evaluate how static, dynamic and secure web service requests affect performance and quality of service at distributed web sites. Buyya et. al.[7] have proposed the use of market oriented resource management in order to regulate the supply and demand of cloud resources at market equilibrium. Again Buiya et. al.[7] have proposed the architecture for market oriented allocation of resources within clouds. Xinhui et. al. [9] have proposed total cost of ownership of cloud computing using a suite of metrics and formulas. Again, H.Aydin et.al [19] have proposed minimizing the energy consumption and subsequently the cost for the static system. Again K.Mukherjee and G.Sahoo[20] [21] [22] [23] have conducted research on different issues of cloud computing regarding its security aspects, mathematical models and others. Again K.Mukherjee and G. Sahoo[24] have proposed a frame work for achieving better load balancing and job scheduling in Grid environment.

3. PROPOSED FRAME-WORK OF ECO-FRIENDLY CLOUD COMPUTING

In order to develop a eco-friendly cloud computing, we propose a better energy management for IaaS by core middleware. We propose that the core middleware should have “Power Consumption Policy Maker(PCPM)” “Virtual Machine Turn-on-of Decider(VMTD)”, “Service Assigner to Virtual Machines(SAVM),“Mapping of service from Virtual Machine to IaaS(MVMI)”, “Resource using Recorder(RUR)”, “Tracker of Virtual Machines (TVM)”,”Virtual Machines(VMs)”,“Service Execution Manager(SEM)” etc.. We propose that the user-level middleware should consists of “Graphical User Interface for Consumer (GUIC)”, “Quality of Service Checker(QoSC)”, “Service Accepter(SA)”, “Billing System(BS)” etc. We propose that the working of overall process should be as follows:

Step1: At first, consumer would submit services to GUIC, from where service is submitted to SA. SA decides whether to accept the service or not after getting information of Virtual Machine status from TVM. TVM keep tracks about the status of VMs (i.e. whether VMs are available or not to execute the new service) as well as from the information of power consumption policy from PCPM.

Step 2: If service is accepted then it is assigned to QoSC, which fixes up the SLA and the price of the requested service after getting information from BS. BS estimates the cost of a service using the historical data at RUR or by using empirical cost estimation technique. It also decides the penalties for violation of SLA.

Step 3: After QoSC, services are submitted to SEM, which interacts with PCPM. PCPM interacts with VMTD and turns the required number of VMs on to carry out services and then assign services to VMs using SAVM. RUR keeps resource using record.

Step 4: Finally the active VMs(which are turned on), execute the services on CPUs of IaaS. The active VMs turns the powers of the corresponding CPUs of IaaS on and the rest are made off and then send the services from active VMs to active CPUs of IaaS using MVMI.

Thus all CPUs of IaaS are not turned on, only those required are turned on and the rest are turned off. The overall process is shown by the sequence diagram in Figure 1, (Figure 1 is shown at last page). In this way power consumption of energy is minimized along with minimization of operational cost and the emission of carbon dioxide gas to the environment. Thus cloud computing is a very promising technology due to its eco-friendly nature with minimized operational cost.

4. PROPOSED ALGORITHM

In nature, we find a lot of interesting collective behavior of bees, ants, etc. It is observed that honey bees [17] work in a decentralized and self –organized manner. The honey colonies practise division of labor between the forgers, who work in the field collecting the nectar and the food-stores who work in the hive to store the nectar. The forger bees search the nectar at random and after getting the nectar they return back to the hive and start dancing on the dance floor in order to motivate the follower forgers to access the nectar. The duration of this dance is closely correlated with the search time experienced by the dancing bees. In this paper, we have used this behavior of bees to enunciate cloud computing minutely. In cloud computing environment, we observe that some CPUs of IaaS are overloaded for processing consumers’ services, some are under loaded and some are totally idle. We can save the consumption of energy by turning these idle CPUs off and rescheduling services from overloaded CPUs to under loaded CPUs. Our proposed technique will not only save energy but will also prevent potential SLA violation. But the problem lies in managing these idle CPUs, overloaded CPUs and under loaded CPUs effectively; for this we have proposed Bee-Ants colony system. At the beginning, we divide our jobs into two parts; the first part, which looks after the proper management of overloaded and under loaded CPUs (service rescheduling) and the second part, which helps to manage the idle CPUs (power consumption management). We propose bee colony algorithm for service rescheduling and ant colony algorithm for power consumption management. Thus for service rescheduling, we consider TVM as hive which consist of bee agents, from where agents start to forge for nectar. Here nectar implies the threshold value of CPU. Poor quality of nectar implies lower threshold value of CPU whereas good quality of nectar implies the higher threshold value of CPU. Thus good quality of nectar implies overloaded CPU and poor quality of nectar implies under loaded CPU, in addition the dance floor represents the service scheduling table, where the information about the status of the CPUs that the bee agents have visited, are stored . Thus the forging bees after finding the nectar, communicate directly via dancing on the dance floor. There are two types of dance; waggle dance which implies poor quality of nectar and tremble dance (round dance) which implies good quality of nectar. After dancing, the old bee agents die. If the dance is tremble dance, then the new born bee agents fly to collect the good quality nectar and store them in the hive. After this operation the old bee agents die and the new born bee agents start to fly with the good quality nectar stored in the hive, and finally mix them with those sources which are holding poor quality nectar. This process of distribution goes on until there is a uniform quality of nectar in all the sources. Actually this process is implemented for rescheduling the services from overloaded CPUs to under loaded CPUs in order to provide good QoS(Quality of Services) to consumers. Again if the dances of the bee agents are waggle dance then hive TVM states SA to accept new services from consumer. After this step, we apply ant colony based algorithm to find the idle CPUs and then turning them off in order to minimize the consumption of power and hence lower the operational cost. The ant colony from VMTD start to find the source of food, in our context it is idle CPUs. The ant colony algorithms were introduced by Dorigo(1992)[18] as a distributed meta-heuristic, which mimics the social forging behavior of real ants. The algorithm involves artificial agents called ants who show a cooperative behavior to find the shortest path to the food source from their nest. Ant System [18] is best suited in solving problems which run in dynamic and varied environments without the help of any central control. This is why it is very much applicable to distributed problem solving. As cloud computing is dynamic in nature, hence we propose to apply ant system in cloud computing in finding idle CPUs. Due to their restricted visibility, ants interact with one another by dropping a chemical known as pheromone for their indirect and global communication. The pheromone concentration is distributed on the path traversed by the ants and is inclined to move over the path with higher pheromone concentration. Thus, ants based algorithm construct a solution in an iterative manner by deriving information from their short visibility (myopic information) and pheromone concentration distributed on path (long term information). An evaporation mechanism is introduced in order to avoid stagnation.

Based upon the above discussion, the algorithms based eco-friendly cloud computing are given below:

```
Procedure GUIC (service)
Begin
Boolean flag
flag ← SA(services)
if (flag = 0)
then
  write(“Request Service can not be carried  out”)

 else
       write(“Request Service is accepted”)
 End

Function Boolean SA (service)
Begin
   Boolean flag.
  flag ← TVM(Request for checking of  availability of
                                  Virtual Machines)
If (flag =0)
                Return(false)
else
       do
             call Procedure VMTD()
         Return (true)
        end
End
Function Boolean TVM (Request_for_
                                      _Checking_of_
                                      _Resource_Avablity)
Begin
         Boolean flag
         Bee stores in hive.
      Forger Bees from hive start to search for nectar
      i.e. Availability of VMs.
       Forger Bees return back and start dancing on
       the dancing floor.
       If (dance be Waggle Dance)
                        then Return (true.)
      else
             do
                Call QoSC (service)
             Return (True)
   End

Procedure QoSc (service)
Begin
```

Ψ is upper utilization of thresholds of CPUs representing good quality of nectar.

δ is lower utilization of thresholds of CPUs representing poor quality of nectar.

Repeat until $\Psi = \delta$ do

After observing the Tremble dance on the dancing floor, new born bee agents start to collect good quality nectar from the source and start to store them in the hive.

Gradually previous sources of good quality nectar, loose their quality, hence $\Psi = \Psi - 1$

After this operation the old bee agents die and the new born bee agents start to fly with the good quality nectar stored in the hive, and finally mix them with those sources which are holding poor quality nectar. This process of distribution goes on until there is a uniform quality of nectar in all the sources, hence $\delta = \delta + 1$.

```
done.
End.

Procedure VMTD ( )
Begin
/* Initialization*/
For every edge (i, j) do
$ι_{ij}(0) = ι_0$
$η_{ij} = η_0$
End For.
For   k =1 to m do
     Place  k on nest M
 End   for
/* Main Loop */
  For t = 1 to $t_{max}$ do
 For k = 1 to m do
Build tour $T^k(t)$
Choose the next CPU j with probability
   $p^k_{ij} = \frac{[ι_{ij}]^α \, [η_{ij}]^β \, r_{ji}(d)}{\sum_{l \, ε \, N^k_i} [ι_{ij}]^α \, [η_{ij}]^β}$ , if   j ε $N^k_i$
            = 0, otherwise.
Find the threshold $ξ_j$ of the $j^{th}$ CPU
If $| ξ_i | < ε$  then turn that CPU off.
Where i is the current CPU.
          $η_{ij} = 1/d_{ij}$ , called visibility
          $d_{ij}$ = Euclidian distance.
          $ι_{ij}$ =  Pheromone Trials.
          α = Parameter which determines the relative
               influence of the pheromone trial.
          β = Parameter which determines the relative
           influence of the visibility.
           ε  is very very small quantity.
           $N^k_i$   is the set of all CPUs that the ant still has to visit when it is on CPU i.   By exploiting $N^k_i$   an ant k can avoid visiting a CPU more than once.
M is the Nest from where ants start to turn off those CPUs whose thresholds are very small.              .
End For.
/* Updating Pheromone Trails*/

For every stage (i , j )  do
Update pheromone trails by using following:
$τ_{ij} = (1 – ρ) \, . \, ι_{ij}(t) + \sum_{k=1}^{m} Δι^k_{ij}$
Where  $Δι^k_{ij} = 1/C^k$ , if stage ( i, j )  ε $T^k$
             = 0 , otherwise
$c^k$   is the sum of lengths of arc belonging to  $T^k$
ρ   is  the  pheromone evaporation rate.
End.
```

5. EXPERIMENTAL RESULT

A private cloud has been setup based on Ubuntu's 10.04 Server edition, that consists of two Servers – Server A and Server B. Server A acts as the cloud, cluster, warehouse and storage controller and Server B acts as node controller. We configured Machine A on a Core2duoX6800 processor based machine with 2GB DDR 2 RAM and 80 GB Hard disk. Machine B is running on an AMD PhenomeII X4 965 processor with 4 GB DDR 3 RAM and 250 GB Hard disk. The nodes communicate through a fast local area network. This paper demonstrates the results of local accessing of multiple web services running on the web servers. Experiments have been carried out on above mentioned cloud environment. In order to execute the web services on Machine A, we have used the following commands:

```
If [ ! –e ~/.euca/mkey.priv]; then
   mkdir –p –m 700 ~/.euca
   touch ~/.euca/mykey.priv
   chmod 0600 ~/.euca/mykey.priv
   euca-add-keypair mykey > ~/.euca > ~/.euca/mykey
fi
```

Thus creating keypair, we now log into our instance as root. After that, we can monitor state of instance using the command:

```
watch –n5 euca-describe-instances
```

Thus we start to access different web services.In order to evaluate our algorithm, we have conducted several experiments with different values of the energy consumption and SLA violation. We observe that our experiments shows that with rise of consumption of energy, there is a sharp fall in SLA violation but the beauty of our algorithm is that it estimates some cases in which there are optimal values between energy consumption and SLA violation. The fact is clearly shown in Figure 2. In Figure2, from left to right, we are having some lines; indicating minimum energy consumption, mean energy consumption, standard deviation and maximum energy consumption. .

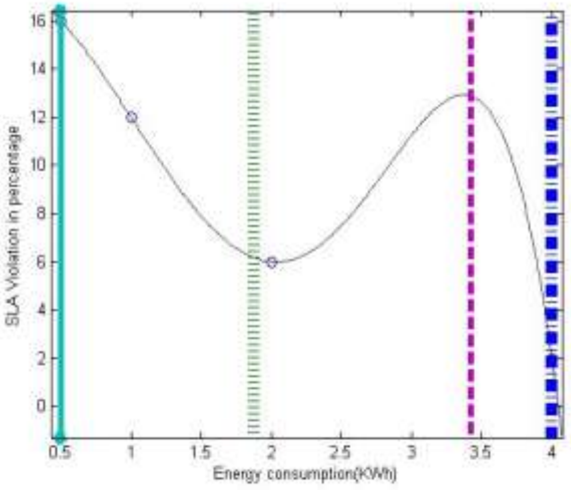

Figure 2.

Thus the algorithm states some cases in which we can have better utilization of energy with less SLA violation

6. CONCLUSION WITH FUTURE DIRECTION OF WORK

In this endeavor we have emphasized on developing an eco-friendly cloud computing by turning the switches of idle CPUs of IaaS off, simultaneously we have focused on avoiding SLA violation for minimized energy consumption and operational cost. For the overall process, we have proposed algorithm. We feel that a proper security management is needed for eco-friendly cloud computing, which we intend to elucidate in our forthcoming endeavor.

7. REFERENCES

[1] L.Keleinrock. A vision for the Internet. ST Journal of Research, Nov.2005, pages 4-5

[2] S.Adreozzi, P. Ciancarini, D. Montesi, R. Moretti, " Towards a model for quality of web and grid service" In Proc 13th IEEE international Workshops on Enabling Technologies: Infrastructure for Colloborative Enterprises(WET ICE'04), 2004, page 271-276,.

[3] D. Gouscos, M. Kalikakis, and P. Georgiadis. "An approach to modeling Web service QoS and provision price", in Proceding 3rd International Conference on Web Information Systems Engineering Workshops, 2003,pages 121-130.

[4] J.P Thomas, M.Thomas and G.Ghinea "Modeling of Web service flow", in Proceeding IEEE International Conference on E-Commerce (CEC 03), 2003, pages 391-398,.

[5] S.Tu , M.Flanagin Y. Wu, M. Abdelguerfi, E. Normand, V. Mahadevan, " Design Strategies to improve performance of GISWeb services", in Proceding International Conference on Information Technology : Coding and Computing(ITCC04), 2004, pages 444-448,.

[6] R.Levy, J Nagarajaro, G. Pacifici, M. Spreitzer , A. Tantawi, and A. Youssef, " Performance management for cluster based Web services", in Proceding IFIP/IEEE 8th International Symposium on Integrated Network Management, 2003, pages 247-261 ,.

[7] Rajkumar Buyya, Chee Shin Yeo and Srikumar Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities", The 10th IEEE International Conference on High Performance Computing and Communications, IEEE Computer Society, 2008, pages 5-13.

[8] Foster I., C. Kesselman, J. Nick, and S. Tuecke, "Grid Services for Distributed System Integration", IEEE Computer, June 2002, pages 37-46..

[9] Xinhui Li, Ying Li, Tiancheng Liu, Jie Qui, Fengchun Wang "International Conference on Cloud Computing" IEEE Computer, 2009 , pages 93-100,

[10] Furlinger K., M. Gerndt, "A Lightweight Dynamic Application Monitor for SMP Clusters", HLRB and KONWIHR joint Reviewing Workshop 2004, March 2004.

[11] Furmento N., A. Mayer, S. McGough, S. Newhouse, T. field, and J. Darlington, "ICENI: Optimization of Component Applications within a Grid environment", Parallel Computing, 2002, pages 1753-1772,.

[12] H.FWedde and M.Farooq, "A comprehensive review of nature inspired routing algorithm for fixed telecommunication networks", Journal of System Architecture, 2006, vol. 52, no. 8-9, pages 461-484,.

[13] C. Blum and D. Merkle, Eds., Swarm Intelligence: Introduction and Applications, Natural Computing Series. Springer, 2008, pages 344-349.

[14] A. Abraham, C. Grosan, and V. Ramos, Stigmergic Optimization Studies in Computational Intelligence. Secaucus, NJ, USA: Springer-Verlag New York., 2006, pages 124-129

[15] V. Cardellini, E. Casalicchio, and M. Colajanni, "A performance study of distributed architectures for the quality of Web services", in Proceding 34th Annual Hawaii International Conference on System Sciences,2001

[16] L.Vaquero, L.Rodero-Marino,J.Caceres, M.Linder, "A break in the clouds: towards a cloud definition", SIGCOMM Computer Communication Review,2009, pages 137-150

[17] T.D. Seely, "The Wisdom of the Hive", Harvard University Press, 1995, pages 295-296.

[18] Marco Dorigo and Thomas Stutzle "Ant Colony Optimization", Prentice-Hall of India, India, 2001

[19] H. Aydin, R.G Melhem, D Mosse, and P. Mejia-Alvarez. "Power-Aware Scheduling for Periodic Ral-Time Task", IEEE Transactions on Computers, May 2004, , 53(5)

[20] K.Mukherjee and G.Sahoo " Mathematical Model of Cloud computing framework using Fuzzy Bee Colony optimization Technique", International Conference on Advances in Computing, Control and Telecommunication Technologies, IEEE Xplorer, 2009

[21] K.Mukherjee and G.Gahoo “ A Secure Cloud Computing” , International Conference on recent trends in Information, Telecommunication and computing, IEEE Xplorer, 2010

[22] K.Mukherjee and G. Sahoo “ Performance Analysis of Cloud Computing using Multistage Ant System” , International Journal of Computer Applications(0975- 8887), volume 1- No 20, 2010

[23] K.Mukherjee and G.Sahoo, “Cloud Computing: Future Framework for e-Governance” , International Journal of Computer Applications(0975- 8887), volume 7, No 7, October 2010.

[24] K.Mukherjee and G. Sahoo, “A framework for achieving better load balancing and job scheduling in Grid environment”, International Journal of Information Technology and knowledge Management, volume 2, No - 1, pp- 199- 202, January- June, 2009.

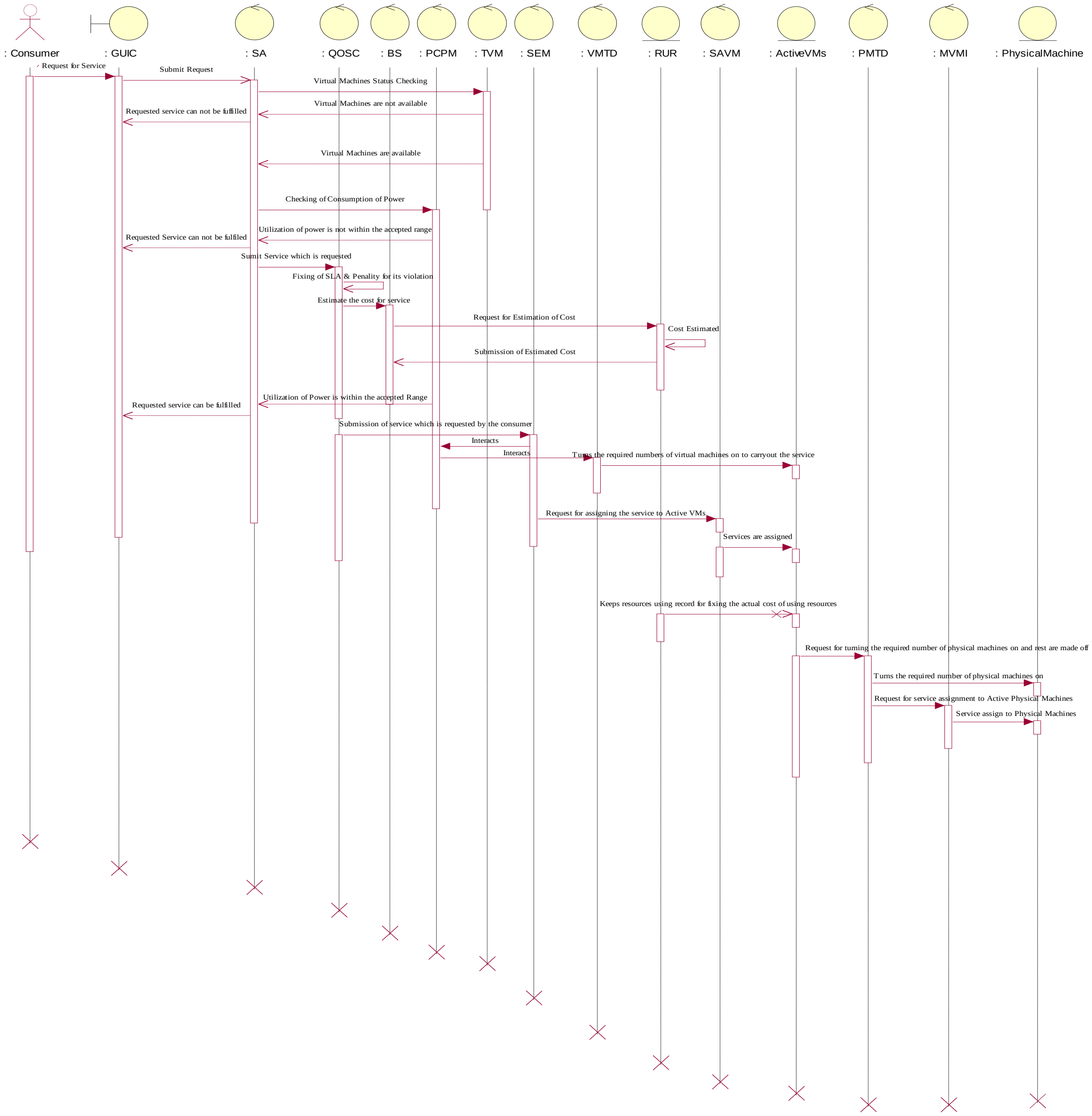
: Consumer
: GUIC
: SA
: QOSC
: BS
: PCPM
: TVM
: SEM
: VMTD
: RUR
: SAVM
: ActiveVMs
: PMTD
: MVMI
: PhysicalMachine
Request for Service
Submit Request
Virtual Machines Status Checking
Virtual Machines are not available
Requested service can not be fulfilled
Virtual Machines are available
Checking of Consumption of Power
Utilization of power is not within the accepted range
Requested Service can not be fulfiled
Sumit Service which is requested
Fixing of SLA & Penality for its violation
Estimate the cost for service
Request for Estimation of Cost
Cost Estimated
Submission of Estimated Cost
Utilization of Power is within the accepted Range
Requested service can be fulfilled
Submission of service which is requested by the consumer
Interacts
Interacts
Turns the required numbers of virtual machines on to carryout the service
Request for assigning the service to Active VMs
Services are assigned
Keeps resources using record for fixing the actual cost of using resources
Request for turning the required number of physical machines on and rest are made off
Turns the required number of physical machines on
Request for service assignment to Active Physical Machines
Service assign to Physical Machines

Figure 1