COST OPTIMIZATION OF CLOUD USING EFFICIENT GREEN CONTROL ALGORITHM WITH VIRTUAL INSTANCES

Ms.W.Jenifer Sathya Bai M.E,
Gojan School of Business and Technology,
Redhills,
Chennai-52
jenifersathyaw@gmail.com

Mr.R.Anandh Assistant Professor /CSE
Gojan School of Business and Technology,
Redhills,
Chennai-52
raanandh37@gmail.com

Abstract

An energy efficient control, especially in mitigating server idle power has become a critical concern in designing a modern green cloud system. Algorithm works to find the idle mode systems and denied their process. That will solve the constrained optimization problems and making costs/ performances trade off's in systems with different policies. Power saving is done with virtual machine instance for the computing infrastructure managing the ISN policy, SN & SI policy. Ideally shutting down servers when they are left idle during low-load periods is one of the most direct ways to reduce power consumption. Virtual Instances with three operating modes Busy, Idle & Sleep. Jobs will be allocated to Virtual Instances. Instances will be allocated to Individual system. Power consumption of Virtual Instances will be calculated. Completion time of Virtual Instances will be fast than allocating all Virtual Instances in a single system.

Keywords

Efficient green control algorithm, ISN policy, SN & SI policy, Power saving ,Cost optimization, Response time.

I.INTRODUCTION

Cloud computing is a new service model for sharing a pool of computing resources that can be rapidly accessed based on a converged infrastructure. In the past, an individual use or company can only use their own serv-ers to manage application programs or store data. Nowa-days, resources provided by cloud allow users to get on-demand access with minimal management effort based on their needs. Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) are all exist-ing service models. For example, Amazon web services is a well-known IaaS that lets users perform computations on the Elastic Compute Cloud (EC2). Google's App Engine and Salesforce are public clouds for providing PaaS and SaaS,respectively

As cloud computing is predicted to grow, substantial power consumption will result in not only huge operational cost but also tremendous amount of car-bon dioxide (CO_2) emissions . Therefore, an energy-efficient control, especially in mitigating server idle power has become a critical concern in designing a modern green cloud system. Ideally, shutting down servers when they are left idle during low-load periods is one of the most direct ways to reduce power consumption. Unfortunately, some negative effects are caused under improper system controls

First, burst arrivals may experience latency or be unable to access services. Second, there has a power consumption overhead caused by awakening servers from a power-off state too frequently. Third, the worst case is violating a ser-vice level agreement (SLA) due to the fact that shutting down servers may sacrifice quality of service (QoS) . The SLA is known as an agreement in which QoS is a critical part of negotiation. A penalty is given when a cloud pro-vider violates performance guarantees in a SLA contract. In short, reducing power consumption in a cloud system has raised several concerns, without violating the SLA constraint or causing additional power consumption are both important .

To avoid switching too often, a control approach called N policy, defined by Yadin and Naor had been extensively adopted in a variety of fields, such as com-puter systems, communication networks, wireless multi-media, etc. Queuing systems with the N policy will turn a server on only when items in a queue is greater than or equal to a predetermined N threshold, instead of activat-ing a power-off server immediately upon an item arrival.

II. RELATED WORK

Power savings in cloud systems have been extensively studied on various aspects in recent years, e.g., on the virtual machine (VM) side by migrating VMs, applying consolidation or allocation algorithms, and on the data center infrastructure side through resource allocations, energy managements, etc.

A .Power-Saving in Virtual Machine

In Huang et al. studied the virtual machine placement problem with a goal of minimizing the total energy consumption. A multi-dimensional space partition model and a virtual machine placement algorithm were presented. When a new VM placement task arrived, their algorithm checked the posterior resource usage state for each feasible PM, and then chose the most suitable PM according to their proposed model to reduce the number of running PMs.

In Nathuji et al. considered the problem of providing power budgeting support while dealing with many problems that arose when budgets virtualized systems. They managed power from a VM-centric point of view, where the goal was to be aware of global utility tradeoffs between dif-ferent virtual machines (and their applications) when main-taining power constraints for the physical hardware on which they ran. Their approach to VM-aware power budget-ing used multiple distributed managers integrated into the virtual power management (VPM) framework.

B. Power-Saving in Computing Infrastructure.

In Zhang et al. presented Harmony, a Heterogene-ity-Aware Resource Monitoring and management system that was capable of performing dynamic capacity provision-ing (DCP) in heterogeneous data centers. Using standard K-means clustering, they showed that the heterogeneous workload could be divided into multiple task classes with similar characteristics in terms of resource and performance objectives. The DCP was formulated as an optimization problem that considered machine and workload heterogeneity as well as reconfiguration costs.

A framework used to automatically manage computing resources of cloud infrastructures was proposed in to simultaneously achieve suitable QoS levels and to reduce the amount of energy used for providing services. Guazzone, Anglano and Canonico showed that via discrete-event system (DES) simulation, their solution was able to manage physical resources of a data center in such a way to significantly reduce SLO violations with respect to a traditional approach. The energy-efficiency of the infrastructure was defined as the amount of energy used to serve a single application request.

III .POWER MANAGEMENT IN CLOUDS

A.ISN Policy

A busy mode indicates that jobs are processed by a server running in one or more of its VMs'; and an idle mode indi-cates that a server remains active but no job is being proc-essed at that time. To mitigate or eliminate idle power wasted, three power-saving policies with different energy-efficient controls, decision processes and operating modes are presented. First, we try to make an energy-efficient con-trol in a system with three operating modes m ¼ {Busy, Idle, Sleep}, where a sleep mode would be responsible for saving power consumption. A server is allowed to stay in an idle mode for a short time when there has no job in the system, rather than switch abruptly into a sleep mode right away when the system becomes empty . An idle mode is the only operating mode that connects to a sleep mode. A server doesn't end its sleep mode even if a job has arrived; it begins to work only when the number of jobs in a queue is more than the controlled N value. According to the switching process (from Idle to Sleep) and the energy-effi-cient control (N policy), we have called such an approach the "ISN policy". Fig. 1 illustrates the step-by-step decision processes and job flows of the ISN policy.

Step 1. A server ends its busy mode when all current job requests have been finished.

Step 2. A server stays in an idle mode and waits for subsequently arriving jobs before switching into a sleep mode.

Step 3. If a job arrives during an idle period, a server can switch into a busy mode and start to work immediately. A server begins a next idle period until all job requests have been successfully completed.

Step 4. If there has no job arrival, a server switches into a sleep mode when an idle period expires.

Step 5. A server remains in a sleep mode if the number of jobs in the queue is fewer than the controlled N value. Otherwise, a server switches into a busy mode and begins to work.

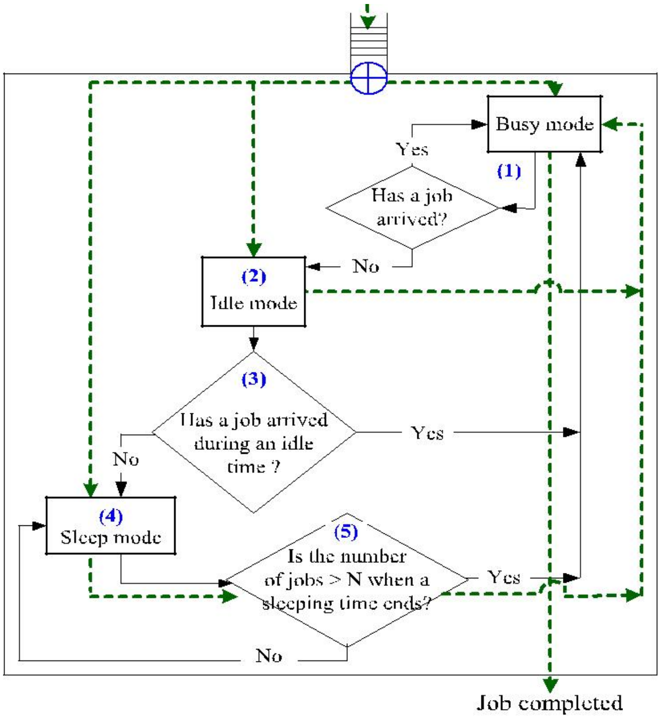

Busy mode
Yes
(1)
Has a job arrived?
No
(2)
Idle mode
(3)
Has a job arrived during an idle time ?
Yes
No
(4)
Sleep mode
(5)
Is the number of jobs > N when a sleeping time ends?
Yes
No
Job completed

Fig. 1. Decision processes of the ISN policy.

Basically, there have two cases of starting a busy mode:

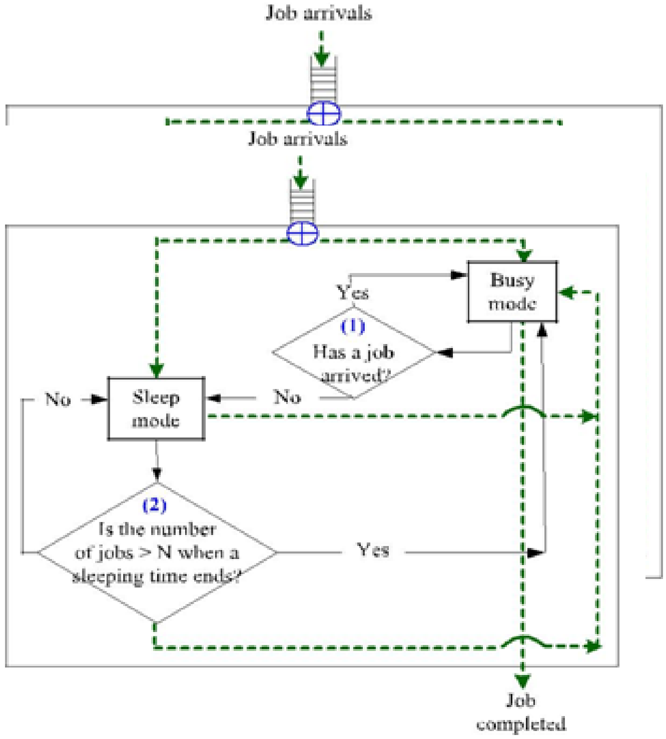
Case 1. Starting a busy mode when a job arrives in an idle mode;

Case 2. Starting a busy mode if the number of jobs in a wait-ing queue is more than the N value when a sleep period expires.

Although power is wasted in allowing a server to stay in the idle mode during a non-load period, the benefits are that an arrival job has more possibilities to get immediately service and the server startup cost can be reduced.

B. SN and SI Policies

To greatly reduce idle power consumption, non-idle mode operating is considered in another approach, where it only holds {Busy, Sleep} operating modes. Instead of entering into an idle mode, a server immediately switches into a sleep mode when the system becomes empty. Similarly, a server switches into a busy mode depending on the number of jobs in a waiting queue to avoid switching too often. According to the switching process (directly to Sleep) and the energy-efficient control (N policy), we have called such an approach the “SN policy”. Fig. 2 illustrates the step-by-step decision processes and job flows of the SN policy.



Step 1. A server switches into a sleep mode immediately when no job is in the system.

Step 2. A server stays in a sleep ,ode if the number of jobs in the queue is less than the N value; otherwise a server switches into a busy mode and begins to work.

For comparison, the other policy is designed with no mode-switching restricton and performed under the other energy – efficient control. A server switches into a sleep mode right away rather than an idle mode when there has no job in the system. This is similar to the SN policy but a server only stays in a sleep mode for a given time. When a sleeping time expires, it will enter into an idle mode or a busy mode depending upon whether a job has arrived or not. According to the switching process(from sleep to idle), we have called such an approach “SI policy”.

Fig 2 illustrates the step-by-step decision processes and job flows of the SI policy.

Step 1. A server switches into a sleep mode immediately instead of an idle mode when there has no job in the system.

Step 2. A server can stay in a sleep mode for a given time in an operating period. If there has no job arrival when a sleeping time expires, a server will enter into an idle mode. Otherwise , it switches into a busy mode without any restriction and begins to work.

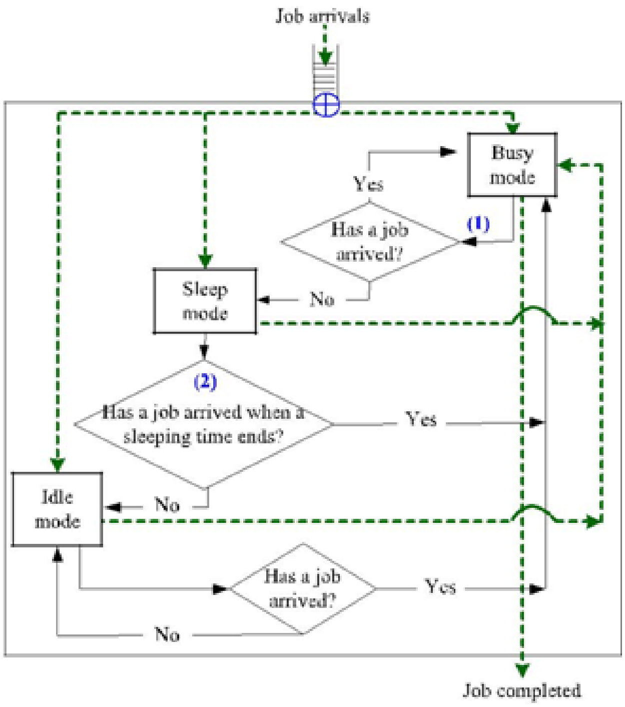


Fig 3. Decision processes of the SI policy.

IV. Performance Of Efficient Green Cloud (EGC) Algorithm.

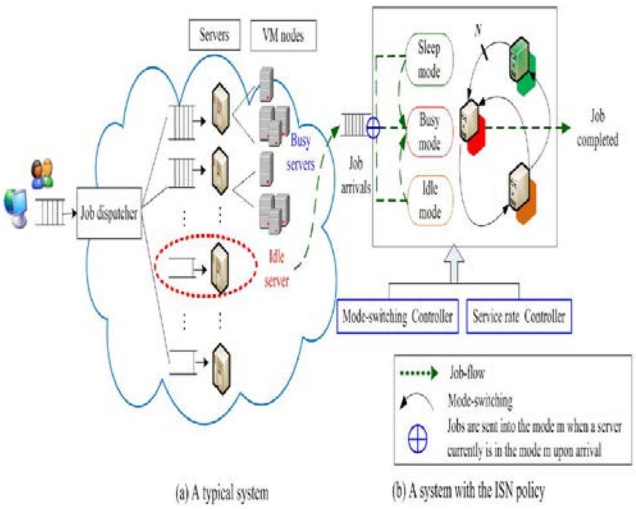
Efficient Green Cloud Algorithm is (i) illustrate the relationship between the mode-switching restriction and traffic-load intensity on power consumption cost and system congestion cost; (ii) examine the idle and sleep probability distributions under different service rates and (iii) compare response times and total operational costs with a typical system, where it doesn't have any energy-efficient control

EGC Algorithm:

Input:

1. An arrival rate _.
2. Upper bound of the server rate and the waiting buffer, denoted by m_u and N_b.
3. Cost parameters [C_o; C_1; . . . ; C_6].
4. A response time guarantee x.
5. System parameters {Q_i; Q_d; Q_s} used by the ISN policy.
6. System parameter {k} used by the SN policy.
7. System parameters {Q; N ¼ 1} used by the SI policy.

Output: m–; N– and $F_c$ðm–; N–Þ



(a) A typical system

(b) A system with the ISN policy

.

Fig 4. Architecture Diagram For EGC

```
Step 1. For i ¼ 1; i ¼ u; i þ þ
       Set $m_i$   a current service rate;
 Step 2. For j ¼ 1; j ¼ b; j þ þ
       Set $N_j$   a current N parameter;
cStep 3. Calculate the system utilization.
       If the current test parameters satisfy the constraint
       of (i) 0 _ r _ 1, then
              Calculate the response time;
       Else
              Return to step 1 and begin to test a next index;
       End
 Step 4. If the current test parameters satisfy the
          constraint of (ii) W _ x, then
            Record the current joint values of ($m_i$; $N_j$)
            and identify it as the approved joint
            parameters;
       Else
            Return to step 1 and begin to test a next index;
       End
 Step 5. When all the test parameters have been done,
          then fðm$_{iþa}$; N$_{jþa}$Þ; . . . ; ðm$_u$; N$_b$Þg current set
          of the approved parameters;
```

```
        Bring cost parameters into the objective
        function by using Eq. (6) and test all approved
        joint parameters;
Step 6. If the joint values of ð$m_{iþa}$; $N_{jþa}$Þ can
        obtain the minimum cost value in all
        testing, then,
            Output ð$m_{iþa}$; $N_{jþa}$Þ and F ð$m_{iþa}$; $N_{jþa}$Þ
    Else
 Return to step 5 and begin to test a next approved parameter.
   End
```

To gain more insight into systems with different power saving policies, experiments are conducted to (i) illustrate the relationship between the mode switching restriction and traffic – load intensity on power consumption cost and system congestion cost (ii) examine the idle and sleep probability distributions under different service rates

V CONCLUSION

The growing crisis in power shortages has brought a concern in existing and future cloud system designs. To mitigate unnecessary idle power consumption, three power-saving policies with different decision processes and mode-switching controls are considered. Our proposed algorithm allows cloud providers to optimize the decision-making in service rate and mode-switching restriction, so as to minimize the operational cost without sacrificing a SLA constraint. The issue of choosing a suitable policy among diverse power managements to reach a relatively high effec-tiveness has been examined based on the variations of arrival rates and incurred costs. Experimental results show that a system with the SI policy can significantly improve the response time in a low arrival rate situation. On the other hand, applying others policies can obtain more cost

REFERENCES

[1] G. Wang and T. E. Ng, “The impact of virtualization on network performance of amazon ec2 data center,” in Proc. IEEE Proc. INFO-COM, 2010, pp. 1–9.

[2] R. Ranjan, L. Zhao, X. Wu, A. Liu, A. Quiroz, and M. Parashar, “Peer-to-peer cloud provisioning: Service discovery and load-bal-ancing,” in Cloud Computing. London, U.K.: Springer, 2010, pp. 195–217.

[3] R. N. Calheiros, R. Ranjan, and R. Buyya, “Virtual machine provi-sioning based on analytical performance and QoS in cloud com-puting environments,” in Proc. Int. Conf. Parallel Process., 2011, pp. 295–304.

[4] Server virtualization has stalled, despite the hype [Online]. Available: http://www.infoworld.com/print/146901, 2010.

[5] Y. C. Lee and A. Y. Zomaya, “Energy efficient utilization of resources in cloud computing systems,” J. Supercomput., vol. 60, no. 2, pp. 268–280, 2012.

[6] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, “A taxonomy and survey of energy-efficient data centers and cloud computing systems,” Adv. Comput., vol. 82, pp. 47–111, 2011.

[7] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility,” Future Genera-tion Comput. Syst., vol. 25, no. 6, pp. 599–616, 2009.

[8] L. Wang, G. Von Laszewski, A. Younge, X. He, M. Kunze, J. Tao, and C. Fu, “Cloud computing: A perspective study,” New Genera-tion Comput., vol. 28, no. 2, pp. 137–146, 2010.

[9] R. Ranjan, R. Buyya, and M. Parashar, “Special section on auto-nomic cloud computing: Technologies, services, and applications,” Concurrency Comput.: Practice Exp., vol. 24, no. 9, pp. 935–937, 2012.

[10] M. Yadin and P. Naor, “Queueing systems with a removable ser-vice station,” Operations Res., vol . 14, pp. 393–405, 1963.