Energy Aware Resource Allocation in Cloud Datacenter

Manasa H.B, Anirban Basu

Abstract- The greatest environmental challenge today is global warming, which is caused by carbon emissions. Energy crisis brings green computing, and green computing needs algorithms and mechanisms to be redesigned for energy efficiency. Green IT refers to the study and practice of using computing resources in an efficient, effective and economic way. Currently, a large number of cloud computing systems waste a tremendous amount of energy and emit a considerable amount of carbon dioxide. Thus, it is necessary to significantly reduce pollution and substantially lower energy usage. The proposed energy aware resource allocation provision data center resources to client applications in a way that improves energy efficiency of the data center, while delivering the negotiated Quality of Service (QoS). In particular, in this paper we conduct a survey of research in energy-efficient computing and propose: architectural principles for energy-efficient management of Clouds; energy-efficient resource allocation policies and scheduling algorithms considering QoS expectations and power usage characteristics of the devices. It is validated by conducting a performance evaluation study using CloudSim toolkit.

Green Cloud Computing aims at a processing infrastructure that combines flexibility, quality of services, and reduced energy utilization. In order to achieve this objective, the management solution must regulate the internal settings to address the pressing issue of data center over-provisioning related to the need to match the peak demand. In this context, propose an integrated solution for resource management based on VMs placement and VMs allocation policies. This work introduces the system management model, analyzes the system's behavior, describes the operation principles, and presents a use case scenario. To simulate the approach of organization, theory and implementation of migration policies and reallocation changes were made as improvements in the code of CloudSim framework.

Keywords- Energy efficiency, Green IT, Cloud computing, migration, resource management, virtualization.

I. INTRODUCTION

Cloud computing [1] delivers infrastructure, platform, and software (applications) as services, which are made available to consumers as subscription based services under the pay-as you- go model. In industry these services are referred to as Infrastructure as a Service (laaS), Platform as a Service (PaaS), and Software as a Service (SaaS) respectively.

Green computing [2] is defined as the study and practice of designing , manufacturing, using, and disposing of computers, servers, and associated sub systems such as monitors, printers, storage devices, and networking and communications systems efficiently and effectively with minimal or no impact on the environment.

Manuscript received on June, 2013.

Dr. Anirban Basu, Dept of Computer Science & Engg,- R&D Centre East Point College of Engineering & Technology, Bangalore, India.

Dr. Anirban Basu, Dept of Computer Science & Engg,- R&D Centre, East Point College of Engineering & Technology, Bangalore, India.

Research continues into key areas such as making the use of computers as energy efficient as possible, and designing algorithms and systems for efficiency related computer technologies. There are several approaches to green computing, namely:

- Product longetivity
- Algorithmic efficiency
- Resource allocation
- Virtualization
- Power management etc.

Need of green computing in clouds

Modem data centers [3], operating under the Cloud computing model are hosting a variety of applications ranging from those that run for a few seconds (e.g. serving requests of web applications such as e-commerce and social networks portals with transient workloads) to those that run for longer periods of time (e.g. simulations or large data set processing) on shared hardware platforms. The need to manage multiple applications in a data center creates the challenge of on demand resource provisioning and allocation in response to time-varying workloads.

Green Cloud computing is envisioned to achieve not only efficient processing and utilization of computing infrastructure, but also minimize energy consumption. This is essential for ensuring that the future growth of Cloud computing is sustainable. Otherwise, Cloud computing with increasingly pervasive front-end client devices interacting with back-end data centers will cause an enormous escalation of energy usage. To address this problem, data center resources [4] need to be managed in an energy efficient manner [5] to drive Green cloud computing. In particular, Cloud resources need to be allocated not only to satisfy QoS requirements specified by users via Service Level Agreements (SLA), but also to reduce energy usage.

Architecture of a green cloud computing platform Fig. shows the high-level architecture for supporting energy efficient service allocation in Green Cloud computing infrastructure [7]. There are basically four main entities involved:

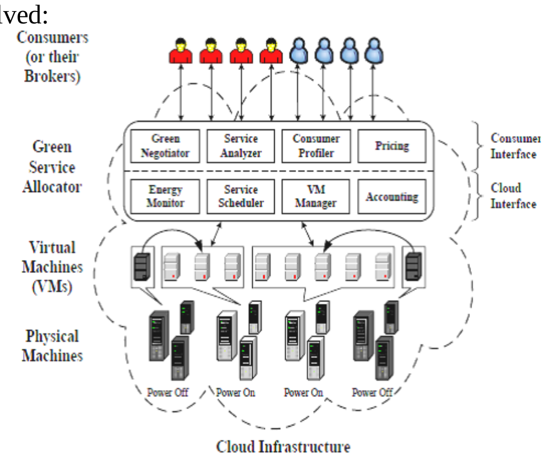


Fig.: Architecture of a green cloud computing environment

a) Consumers/Brokers: Cloud consumers or their brokers submit service requests from anywhere in the world to the Cloud. It is important to notice that there can be a difference between Cloud consumers and users of deployed services. For instance, a consumer can be a company deploying a Web application, which presents varying workload according to the number of users accessing it.

b) Green Resource Allocator: Acts as the interface between the Cloud infrastructure and consumers. It requires the interaction of the following components to support energy efficient resource management:

1. Green Negotiator: Negotiates with the consumers/brokers to finalize the SLA with specified prices and penalties (for violations of SLA) between the Cloud provider and consumer depending on the consumer's QoS requirements and energy saving schemes.

2. Service Analyzer: Interprets and analyses the service requirements of a submitted request before deciding whether to accept or reject it. Hence, it needs the latest load and energy information from VM Manager and Energy Monitor respectively.

3. Consumer Profiler: Gathers specific characteristics of consumers so that important consumers can be granted special privileges and prioritized over other consumers.

4. Pricing: Decides how service requests are charged to manage the supply and demand of computing resources and facilitate in prioritizing service allocations effectively.

5. Energy Monitor: Observes and determines which physical machines to power on/off.

6. Service Scheduler: Assigns requests to VMs and determines resource entitlements for allocated VMs. It also decides when VMs are to be added or removed to meet demand.

7. VM Manager: Keeps track of the availability of VMs and their resource entitlements. It is also in charge of migrating VMs across physical machines.

8. Accounting: Maintains the actual usage of resources by requests to compute usage costs. Historical usage information can also be used to improve service allocation decisions.

c) VMs: Multiple VMs can be dynamically started and stopped on a single physical machine to meet accepted requests, hence providing maximum flexibility to configure various partitions of resources on the same physical machine to different specific requirements of service requests. Multiple VMs can also concurrently run applications based on different operating system environments on a single physical machine. In addition, by dynamically migrating VMs across physical machines, workloads can be consolidated and unused resources can be put on a low-power state, turned off or configured to operate at low-performance levels (e.g., using DVFS) in order to save energy.

d) Physical Machines: The underlying physical computing servers provide hardware infrastructure for creating virtualized resources to meet service demands.

In this paper conduct a survey of research in energy efficient computing and energy efficient resource allocation policies and scheduling algorithm considering QoS expectations and power usage characteristics of the devices.

II. RELATED WORK

One of the first works, in which power management has been applied at the data center level, has been done by Pinheiro et al. [8]. In this work the authors have proposed a technique for minimization of power consumption in a heterogeneous cluster of computing nodes serving multiple web-applications. The main technique applied to minimize power consumption is concentrating the workload to the minimum of physical nodes and switching idle nodes off. This approach requires dealing with the power / performance trade-off, as performance of applications can be degraded due to the workload consolidation. Requirements to the throughput and execution time of applications are defined in SLAs to ensure reliable QoS. The proposed algorithm periodically monitors the load of resources (CPU,disk storage and network interface) and makes decisions on switching nodes on / off to minimize the overall power consumption, while providing the expected performance. The actual load balancing is not handled by the system and has to be managed by the applications. The algorithm runs on a master node, which creates a Single Point of Failure (SPF) and may become a performance bottleneck in a large system. In addition, the authors have pointed out that the reconfiguration operations are time consuming, and the algorithm adds or removes only one node at a time, which may also be a reason for slow reaction in large-scale environments. The proposed approach can be applied to multi-application mixed-workload environments with fixed SLAs.

Chase et al. [9] have considered the problem of energy-efficient management of homogeneous resources in Internet hosting centers. The main challenge is to determine the resource demand of each application at its current request load level and to allocate resources in the most efficient way. To deal with this problem the authors have applied an economic framework: services \bid" for resources in terms of volume and quality. This enables negotiation of the SLAs according to the available budget and current QoS requirements, i.e. balancing the cost of resource usage (energy cost) and the benefit gained due to the usage of this resource. The system maintains an active set of servers selected to serve requests for each service. The network switches are dynamically reconfigured to change the active set of servers when necessary. Energy consumption is reduced by switching idle servers to power saving states (e.g. sleep, hibernation). The system is targeted at the web workload, which leads to a \noise" in the load data. The authors have addressed this problem by applying the statistical \ip-op" filter, which reduces the number of unproductive reallocations and leads to a more stable and efficient control. The proposed approach is suitable for multi-application environments with variable SLAs and has created a foundation for numerous studies on power-efficient resource allocation at the data center level. However, in contrast to [8], the system deals only with the management of the CPU, but does not consider other system resources. The latency due to switching nodes on /off also is not taken into account. The authors have noted that the management algorithm is fast when the workload is stable, but turns out to be relatively expensive during significant changes in the workload. Moreover, likewise [8], diverse software configurations are not handled, which can be addressed by applying the virtualization technology.

Elnozahy et al. [10] have investigated the problem of power-efficient resource management in a single web-application environment with fixed SLAs (response time) and load-balancing handled by the application. As in [9], two power-saving techniques are applied: switching power of computing nodes on / off and Dynamic Voltage and

Frequency Scaling (DVFS). The main idea of the policy is to estimate the total CPU frequency required to provide the necessary response time, determine the optimal number of physical nodes and set the proportional frequency to all the nodes. However, the transition time for switching the power of a node is not considered. Only a single application is assumed to be run in the system and, like in [8], the load balancing is supposed to be handled by an external system. The algorithm is centralized that creates an SPF and reduces the scalability. Despite the variable nature of the workload, unlike [9], the resource usage data are not approximated, which results in potentially inefficient decisions due to fluctuations.

Chiaraviglio and Matta [11] have proposed cooperation between ISPs and content providers that allows the achievement of an efficient simultaneous allocation of compute resources and network paths that minimizes energy consumption under performance constraints. Koseoglu and Karasan [12] have applied a similar approach of joint allocation of computational resources and network paths to Grid environments based on the optical burst switching technology with the objective of minimization of job completion times. Tomas et al. [13] have investigated the problem of scheduling Message Passing Interface (MPI) jobs in Grids considering network data transfers satisfying the QoS requirements. Dodonov and de Mello [14] have proposed an approach to scheduling distributed applications in Grids based on predictions of communication events. They have proposed the migration of communicating processes if the migration cost is lower than the cost of the predicted communication with the objective of minimizing the total execution time. They have shown that the approach can be effectively applied in Grids; however, it is not viable for virtualized data centers, as the VM migration cost is higher than the process migration cost. Gyarmati and Trinh [15] have investigated the energy consumption implications of data centers' network architectures. However, optimization of network architectures can be applied only at the data center design time and cannot be applied dynamically.

III.ENERGY AWARE ALLOCATION OF DATACENTER RESOURCES

By supporting the movement of VMs between physical nodes, it enables dynamic migration [6] of VMs according to the performance requirements. When VMs do not use all the provided resources, they can be logically resized and consolidated to the minimum number of physical nodes, while idle nodes can be switched to the sleep mode to eliminate the idle power consumption and reduce the total energy consumption by the data center.

Currently, resource allocation in a Cloud data center aims to provide high performance while meeting SLAs, without focusing on allocating VMs to minimize energy consumption. To explore both performance and energy efficiency, three crucial issues must be addressed. First, excessive power cycling of a server could reduce its reliability. Second, turning resources off in a dynamic environment is risky from the QoS perspective. Due to the variability of the workload and aggressive consolidation, some VMs may not obtain required resources under peak load, and fail to meet the desired QoS. Third, ensuring SLAs brings challenges to accurate application performance management in virtualized environments.

1. Allocation of VM

The problem of VM allocation can be divided in two: the first part is the admission of new requests for VM provisioning and placing the VMs on hosts, whereas the second part is the optimization of the current VM allocation. The first part can be seen as a bin packing problem with variable bin sizes and prices. To solve it applied a modification of the Best Fit Decreasing (BFD) algorithm that is shown to use no more than 11/9. OPT + 1 bins (where OPT is the number of bins given by the optimal solution). In our modification, the Modified Best Fit Decreasing (MBFD) algorithms, sort all VMs in decreasing order of their current CPU utilizations, and allocate each VM to a host that provides the least increase of power consumption due to this allocation. This allows leveraging the heterogeneity of resources by choosing the most power-efficient nodes first. The pseudo-code for the algorithm is presented in Algorithm 1. The complexity of the allocation part of the algorithm is n.m, where n is the number of VMs that have to be allocated and m is the number of hosts.

<u>Algorithm 1: Modified Best Fit Decreasing (MBFD)</u>

```
Input:hostList,vmList Output:allocation of VMs
vmList.sortDecreasingUtilization()
foreach vm in vmList do
    minPower←MAX
    allocatedHost←NULL
    foreach host in hostList do
        if host has enough resource for vm then
            power←estimatePower(host, vm)
            if power < minPower then
                allocatedHost←host
                minPower←power
    if allocatedHost≠ NULL then
        allocate vm to allocatedHost
return allocation
```

2. Selection of VM

The optimization of the current VM allocation is carried out in two steps: at the first step select VMs that need to be migrated, at the second step the chosen VMs are placed on the hosts using the MBFD algorithm. To determine when and which VMs should be migrated, introduce three double-threshold VM selection policy. The basic idea is to set upper and lower utilization thresholds for hosts and keep the total utilization of the CPU by all the VMs allocated to the host between these thresholds. If the CPU utilization of a host falls below the lower threshold, all VMs have to be migrated from this host and the host has to be switched to the sleep mode in order to eliminate the idle power consumption. If the utilization exceeds the upper threshold, some VMs have to be migrated from the host to reduce the utilization. The aim is to preserve free resources in order to prevent SLA violations due to the consolidation in cases when the utilization by VMs increases. The difference between the old and new placements forms a set of VMs that have to be reallocated.

The Migration Policy

The Migration policy selects the minimum number of VMs needed to migrate from a host to lower the CPU utilization below the upper utilization threshold if the upper threshold is violated.

The pseudo-code for the Migration algorithm for the over utilization case is presented in Algorithm 2. The algorithm sorts the list of VMs in the decreasing order of the CPU utilization. Then, it repeatedly looks through the list of VMs and finds a VM that is the best to migrate from the host. The best VM is the one that satisfies two conditions. First, the VM should have the utilization higher than the difference between the host's overall utilization and the upper utilization threshold. Second, if the VM is migrated from the host, the difference between the upper threshold and the new utilization is the minimum across the values provided by all the VMs. If there is no such a VM, the algorithm selects the VM with the highest utilization, removes it from the list of VMs, and proceeds to a new iteration. The algorithm stops when the new utilization of the host is below the upper utilization threshold. The complexity of the algorithm is proportional to the product of the number of over-utilized hosts and the number of VMs allocated to these hosts.

Algorithm 2: Migration

```
Input: hostList Output: migrationList
foreach h in hostList do
  vmList ← h.getVmList ( )
  vmList.sortDecreasingUtilization ( )
  hUtil ← h.getUtil ( )
  bestFitUtil←MAX
  while hUtil > THRESH_UP do
     foreach vm in vmList do
         if vm.getUtil ( ) > hUtil - THRESH UP
         then
         t ← vm.getUtil ( ) - hUtil + THRESH_UP
         if t < bestFitUtil then
                  bestFitUtil ← t
                  bestFitVm ← vm
         else
                  if bestFitUtil = MAX then
                           bestFitVm ← vm
                  break
     hUtil ← hUtil - bestFitVm.getUtil ( )
     migrationList.add (bestFitVm)
     vmList.remove (bestFitVm)
  if hUtil < THRESH_LOW then
     migrationList.add (h.getVmList ( ))
     vmList.remove (h.getVmList ( ))
return migrationList
```

IV. PERFORMANCE ANALYSIS

The performance of the proposed method is measured and presented by using a CloudSim tool as a performance graph.

1. Implementation Environment

The CloudSim toolkit has been chosen as a simulation platform as it is a modern simulation framework aimed at Cloud computing environments. We have simulated on a single machine with a static numbers of hosts and VMs, it needs a CPU with the performance equivalent to 1000, 2000 or 3000 MIPS, 4 GB of RAM, 40 GB of disk 15 inch colour monitor. Each VM requires one CPU core with 250, 500, 750 or 1000 MIPS, 128 MB of RAM and 1 GB of storage. Initially, the VMs are allocated according to the requested characteristics assuming 100% CPU utilization.

2. Simulation Improvements and Results

Due to the difficulty of replicating experiments in real environments and with the goal of performing controlled and repeatable experiments, we opted to validate the proposed scenarios using simulation. For this task we used the CloudSim framework, a tool developed in Java at the University of Melbourne, Australia to simulate and model cloud-computing environments. Extended CloudSim to simulate the organization-theory approach and implemented the migration and reallocation policies using this improved version. In this way, one can evaluate the scenarios proposed in Section IV and reuse the implemented models and controls in further simulations.

The basic characteristics of the simulated environment, physical machines and virtual machines using data extracted from production equipments located at our university. The data was used to represent the reality of a data center, and is based on a data center into production at the university. It consists of different physical machines and applications that require heterogeneous virtual machine configurations.

The main components implemented in the improved version at CloudSim are as follows:

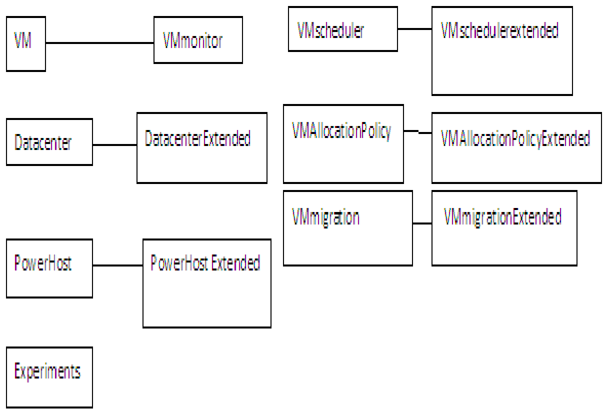
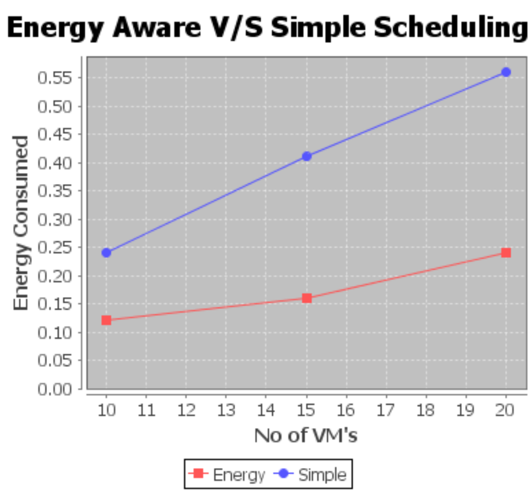


Fig.2. Classes implemented in the ClodSim framework

- PowerHost: controls the input and output of physical machines;
- VmMonitor: controls the input and output of virtual
- machines;
- VmAllocationPolicyExtended: allocation policy;
- VmSchedulerExtended: allocates the virtual machines;
- DatacenterExtended: controls the datacenter.

V. RESULTS

For the experimental results we have used the Non Power Aware policy (Simple Scheduling). This policy does not apply any power aware and implies that all hosts run at 100% CPU utilization and consumes maximum power all the time. Energy aware scheduling is compared with the non power aware policy.



The above graph shows the simulation results for 3 inputs of host and VMs firs the number of host=5 and VMs=10, second host=10 and VMs=15, third host=15 and VMs=20.

VI. CONCLUSION

This work advances the Cloud computing field in two ways. First, it plays a significant role in the reduction of data center energy consumption costs, and thus helps to develop a strong and competitive Cloud computing industry. Second, consumers are increasingly becoming conscious about the environment. A recent study shows that data centers represent a large and rapidly growing energy consumption sector of the economy and a significant source of CO2 emissions. Reducing greenhouse gas emissions is a key energy policy focus of many countries around the world. Have presented and evaluated our energy-aware resource allocation algorithms utilizing the dynamic consolidation of VMs. The experiment results have shown that this approach leads to a substantial reduction of energy consumption in Cloud data centers in comparison to static resource allocation techniques. Aiming at putting in a strong thrust on open challenges identified in this paper in order enhance the energy-efficient management of Cloud computing environments.

In this paper, proposed an organization theory model for resource management of Green Clouds and demonstrated that the proposed solution delivers both reliability and sustainability, contributing to our goals of optimizing energy utilization and reducing carbon emission.

Concepts related to cloud computing and green cloud computing were presented. We also described the simulator employed in the practical part of the experiments and detailed improvements undertaken on it to validate the green cloud computing approach. The simulator we used is called CloudSim and was developed at the University of Melbourne in Australia. The improvements we implemented relate to services-based interaction and policies for migration and relocation of virtual machines, which are based on system monitoring and control.

REFERENCES

[1] A. Berl, E. Gelenbe, M. di Girolamo, G. Giuliani, H.de Meer, M.-Q. Dang, and K. Pentikousis. EnergyEfficient Cloud Computing. The Computer Journal,53(7), September20 1 O,doi: 10.10931 comjnl! bxp08

[2] The green grid consortium (2011).

[3] Ministry of Economy, Trade and Industry, Establishment of the japan data center council, Press Release.

[4] EMC 2008 Annual overview Releasing the power of infonnation, http://www.emc.comJ digital_universe.

[5] S.Albers, "Energy -Efficient Algorithms," Communications of the ACM, vo1.53, no.5, pp.86-96,May 2010.

[6] Anton Beloglazov, Jemal Abawajy, Rajkumar Buyya for "Energy-Aware Resource Allocation Heuristics for Effcient Management of Data Centers for Cloud Computing" preprint submitted to Future Generation Computer Science.

[7] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems 25 (6) (2009) 599-616.

[8] E. Pinheiro, R. Bianchini, E. V. Carrera, T. Heath, Load balancing and unbalancing for power and performance in cluster-based systems, in: Proceedings of the Workshop on Compilers and Operating Systems for Low Power, 2001, pp. 182-195.

[9] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, R. P.Doyle, Managing energy and server resources in hosting centers, in: Proceedings of the 18th ACM Symposium on Operating Systems Principles, ACM New York, NY, USA, 2001, pp. 103-116.

[10] E. Elnozahy, M. Kistler, R. Rajamony, Energy-efficient server clusters, Power-Aware Computer Systems (2003) 179-197.

[11] L. Chiaraviglio, I. Matta, GreenCoop: cooperative green routing with energy-efficient servers, in: Proceedings of the 1st ACM International Conference on Energy-Efficient Computing and Networking (e-Energy 2010), Passau, Germany, 2010, pp. 191-194.

[12] M. Koseoglu, E. Karasan, Joint resource and network scheduling with adaptive offset determination for optical burst switched grids, Future Generation Computer Systems 26 (4) (2010) 576-589.

[13] L. Tomas, A. Caminero, C. Carrion, B. Caminero, Network-aware meta-scheduling in advance with autonomous self-tuning system, Future Generation Computer Systems 27 (5) (2010) 486-497.

[14] E. Dodonov, R. de Mell, A novel approach for distributed application scheduling based on prediction of communication events,Future Generation Computer Systems 26 (5) (2010) 740-752.

[15] L. Gyarmati, T. Trinh, How can architecture help to reduce energy consumption in data center networking?, in: Proceedings of the 1st ACM International Conference on Energy-Efficient Computing and Networking (e-Energy 2010), Passau, Germany,2010, pp. 183-186.