Energy efficient virtual machine placement algorithm based on priority of resources

Amin Rahimi, Leili Mohammad Khanli, and **Saeid Pashazadeh**
Department of Electrical and Computer Engineering
Tabriz University
`{arahimi,l-khanli,pashazadeh}@tabrizu.ac.ir`

Abstract

The increasing energy consumption has become a major concern in cloud computing as it costs a lot and damages environment. Virtual Machine placement algorithms have been proven to be very effective in increasing energy efficiency and thus reducing the costs. In this paper we propose a new priority routing VM placement algorithm (`PRVP`) and compare it with `PABFD` method on COMON dataset using CloudSim for simulation. Our results show the superiority of the proposed method which decreases SLA violations by 68% and lowers energy consumption by 26% compared to `PABFD`. The results support that priority routing VM placement algorithms can be effectively utilized to increase energy efficiency in the clouds.

1 Introduction

The rapid growth in the need for ubiquitous computing utilities has resulted in the realization of a new computing model called *cloud computing* in which consumers from all over the world access the shared computing resources in an on-demand *pay-as-you-go* model. The computing resources are usually available in the form of Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The could computing model requires an adaptive resource allocation management to adopt with the workload of several applications with different resource allocation requirements to provide quality of service for each application and to utilize data center resources efficiently. One solution is to use Virtual Machines (VMs) over Physical Machines (PMs) for resource consolidation and environment isolation. The focus of cloud research is shifting from performance optimization to energy efficiency due to the rapid increase in energy bills and environmental damage. The total energy bill for data centers was $11.5 billion in 2010 and this cost is estimated to double every five year (Scroggie, 2008). Existing works (Beloglazov et al., 2012; Beloglazov and Buyya, 2012; Takouna et al., 2012) show that energy efficiency is highly dependent on the number of running PMs. (Barroso and Hölzle, 2007) reports that server utilization is usually about 10-50% resulting in lots of idle machines in the cloud. An idle machine consumes as much as 70% power as its peak so minimizing the number of active PMs by powering off or putting in low power mode the idle machines and utilizing the active ones more efficiently can reduce the power consumption (De Assuncao et al., 2012). Live migration of VMs partitions PMs into a highly utilized group and an idle group that can be put in low power mode (e.g. sleep, power off, hibernate) to eliminate idle power consumption . This results in huge savings in power consumption, cooling requirements, energy bills, Carbon footprint and in the same time assures quality of service for applications. The workload of applications highly varies in time so the dynamic resource consolidation should migrate VMs in order to adopt to this change. Excessive migration of VMs results in delays in response time of applications which damages the quality of service. To overcome this, a VM placement algorithm should maximize utilization on the least possible number of PMs in order to reduce energy consumption and in the same time should limit the number of VM migrations to assure quality of service which makes the task very challenging yet very rewarding (Beloglazov et al., 2012). VM consolidation problem can be divided into four subpromblems: Host overload detection, Host underload detection, VM selection and VM placement.

When host overload or underload detected, some VMs should be migrated from that host(s). If host overload detected, VM selection algorithm choose some VMs to migrated from that host to another. If host underload detected all the VMs should be migrated from that host so the host can be put in low power mode. The process of placing VMs from migration list to PMs is called VM placement.

In this paper we propose a new VM placement method (`PRVP`) based on priority routing. We compare our method with `PABFD` (Beloglazov and Buyya, 2012) in terms of power consumption, number of migrations, SLA (Service Level Agreement), load balancing and ESV metric described in 4.1. Our main contributions in this paper are: (1) We introduce a new priority routing based VM placement algorithm (`PRVP`) and compare it with `PABFD` in terms of power consumption, load balancing, number of migrations, SLA violations and ESV performance metric. Empirical evidence that `PRVP` are generally superior and can be used to achieve energy efficiency in the clouds. (2) We implement and publish the code of the proposed algorithm within CloudSim framework.

2 Related Work

Several methods have been proposed for VM placement. Hyser et al. (2007) proposed an architecture design for automatic VM placement leading to better resource utitization and less overloaded hosts. Piao and Yan (2010) proposed a network-aware VM placement to avoid network latancy and congestion situations caused by long data transfer time to maintain application perfomance. The main idea is to place VMs in PMs that minimize data access time. To achieve this purpose they calculate data access time of PMs for each data and create a data access time matrix and choose the PM with minimum data access time to host that application. Meng et al. (2010) proposed a traffic-aware VM placement by considering traffic patterns between VMs. VMs which have large mutual bandwidth usage are placed in PMs that are close together. Jiang et al. (2012) poroposed a VM placement algorithm to optimize the problem of joint VM placement and routing with the goal of minimizing the network congestion by exploiting multipath routing capability combined with VM placement.

`PABFD`: Beloglazov and Buyya (2012) proposed a power-aware VM placement named `PABFD` (power-aware best fit decreasing). In `PABFD` all VMs are first sorted (in decreasing order) based on CPU utilization then each VM is assigned to the PM that will increase the least power consumption after assignment. `PABFD` is the default placement algorithm in Cloudsim.

Takouna et al. (2012) investigated the VM placement algorithms for HPC data centers. Their poroposed algorithm is like `PABFD` with considering memory in the power model.

Our proposed VM placement algorithm tries to reduce resource (processing, memory and bandwidth) usage variances across PMs to achieve better load balancing by placing VMs in PMs based on their resource requirements. `PRVP` reduces the overlapping of resources by placing VMs with the same resource usage in separate PMs (e.g. by placing CPU intensive VMs besides RAM intensive VMs in the same PM).

3 Priority Routing VM Placement with Load Balancing

Priority routing VM palcement algorithms have proved to be very effective in energy efficient cloud computing. The main idea is to categorize VMs with regard to their resource utilization and categorize PMs based on their resource availability then place VMs to the PMs with the most similar category. Our proposed algorithm (`PRVP`) follows the general approach of priority routing VM placement algorithms but it differs with prior approaches in that it also load balances the VM to PM assignment.

VM and PM Categorization: One of the weaknesses of previous VM placement algorithms is that they don't consider the priority of resources. For example CPU has a higher priority compared to RAM, so a good VM placement algorithm should consider load balancing of resources based on their priority. Another problem of VM placement algorithms is the overlapping of resources: VMs with the same resource usage shouldn't be placed in same PM. To minimize resource overlapping of VMs we `PRVP` To address these problems. `PRVP` requires that VMs and PMs be categorized in terms of resource usage and resource availability respectively. We build 3 sorted list (in decreasing order) of VMs each associated with the last resource request for CPU, RAM and bandwidth. Then we partition each list into 3 equal parts (High, Medium, Low) labelling the VMS with H, M and L respectively. According to the category (H, M, L) of a VM in each of these 3 lists

(CPU usage list, RAM usage list and bandwidth usage list) a 3 character label is assigned to it. We call this 3 character label CRB (after CPU, RAM and Bandwidth). For example a VM with CRB label of 'HML' (VM_{HML}) has a high CPU utilization, medium RAM utilization and low bandwidth utilization compared to other VMs in the migration list (Figure 1). We do the same procedure for PMs based on their resource availability, so PM_{HML} has a high CPU availability, Medium RAM availability and low bandwidth availability compared to other PMs.

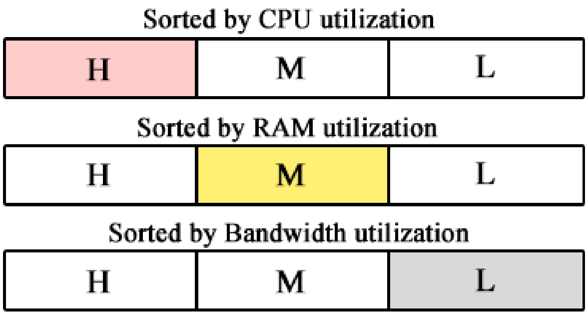

Figure 1: Position of VM_{HML} in sorted lists. List consist of all VMs sorted based on CPU, RAM and bandwidth. The sorted lists are partitioned into 3 equal sized categories each labeled with High, Medium and Low. The CRB label of a VM consists of 3 characters each assiciated with the categoriy of the VM within each list. The CRB label 'HML' here means that the VM has high CPU utilization, medium RAM utilization and low bandwidth utilization.

Routing Table: In order to place VMs in PMs a routing table is constructed. For each VM, PMs are sorted in a path based on the match between resource availability of PMs, resource utilization of VMs and resource priority. We assume that CPU has the most priority after which comes RAM and then bandwidth. The best match for VM_{HLM} is PM_{HLM} and the worst matches are PM_{LHH} and PM_{LHL}. Note that for VM_{HLM}, PM_{HHL} is a better match compared to PM_{LLM} because in the former CPU labels match and RAM/bandwidth labels don't while in the latter RAM/bandwidth labels match and CPU labels don't. Because CPU match produces more reward than combined RAM and bandwidth the former PM is better than the latter. For each VM a PM path is created from the best match to the worst. For example the routing path for VM_{HLM} is:

PM_{HLM}, PM_{HLH}, PM_{HLL},..., PM_{LHL}

VM Placement and Load Balancing: If any host overload or underload is detected, it'll be necessary to migrate some VMs to resolve the problem. The proposed VM placement algorithm (`PRVP`) is shown in 1. First all PMs and VMs are categorized and the corresponding CRB labels are calculated. Then for each CRB label the routing path consisting of sorted list of PMs with highest matches is created and put into a routing table. For each routing path a counter is used which points to the next PM. The counter is initialized to 0 for all routing paths. Given a VM to be assigned, the routing path corresponding to the VM's CRB label is acquired, then the PM which the path's counter points to is checked to have enough resources for that VM. If it has the resources then the VM is assigned to that PM and the counter is incremented by 1. If the PM doesn't have the required resources of the VM then counter is incremented by 1 and next PM is checked for availability of the resources. Every time an assignment happens the counter is incremented so that the next VM goes into the next PM in the routing path. This load balancing prevents resource congestion in the PMs at the begining of the routing path. Performance degradation due to the overlapping of VMs with similar demands is avoided by distributing the VMs into different PMs.

4 Performance Evaluation

In this section we compare the performance of `PRVP` with that of `PABFD`. We test the two algorithms by extensive simulations with real world traces of CoMon project.

Workload Data: We used the CoMon dataset which is gathered from the CoMon project workload traces. CoMon project is a monitoring system for PlanetLab (Park and Pai, 2006). The workload data consist of CPU utilizations of VMs from PMs located at different places in 10 random days. The data is collected every 5 minutes. Table 1 shows the workload datails.

4.1 Evaluation Metrics

We use several metrics to compare the performance of `PABFD` and `PRVP`.

```
getNewVmPlacement:
input  : vmsToMigrate, excludedHosts, hosts
output: A map of VMs to hosts
migrationMap ← emptyMap;
vCu ← sortByCpuUtilizationDec(vmsToMigrate);
vRu ← sortByRamUtilizationDec(vmsToMigrate);
vBu ← sortByBwUtilizationDec(vmsToMigrate);
hCa ← sortByAvailabeMipsDec(hosts);
hRa ← sortByAvailabeRamDec(hosts);
hBa ← sortByAvailabeBwDec(hosts);
typeHostListMap ← getMapByResAvail(hCa, hRa, hBa);
routingMap ← createRoutingMap(typeHostListMap);
for vm ∈ vCu do
    vmtype ←
    getVmTypeByUtilizations(vCu, vRu, vBu, vm);
    allocatedHost ←
    findHostForVm(vm, excludedHosts, vmType, routingMap);

    if allocatedHost ≠ NULL then
        migrationMap[vm]← allocatedHost;
    end
end
return migrationMap;
findHostForVm:
input  : vm, excludedHosts, vmType, routingMap
output: allocatedHost
allocatedHost ← NULL;
routingHosts ← routingMap[vmtype] ;
hostsNum ← sizeof(routingHosts) ;
counter ← 0 ;
while counter ≠ hostsNum do
    host ← getNextHost(routingHosts);
    if host ∈ excludedHosts then
        continue ;
    end
    if host has enough resource for vm then
        allocatedHost ← host;
        break;
    end
    counter ← counter + 1 ;
end
return allocatedHost;
```

Algorithm 1: Priority routing VM placement

Workload date	Number of VMs
03/03/2011	1052
06/03/2011	898
09/03/2011	1061
22/03/2011	1516
25/03/2011	1078
03/04/2011	1463
09/04/2011	1358
11/04/2011	1233
12/04/2011	1054
20/04/2011	1033

Table 1: Workload data details from CoMon project.

Power consumption metric: It has proven that the power consumption of a host has a linear relationship with its CPU utilization (Fan et al., 2007). For multi-core CPUs, CPU utilization of host is the sum of utilization of cores. The power consumption of a few servers is modelled in Cloudsim (Calheiros et al., 2011) based on the real power consumptions in different CPU load levels (e.g. HP ProLiant G4, HP ProLiant G5). Each host with CPU utilization greater than zero is considered for power consumption. Table **??** shows the power consumption of HP ProLiant G4 and HP ProLiant G5 by the amount of CPU utilization.

SLA violation metric: Service-level agreements (SLAs) represent the QoS requirements in cloud computing environments. Cloudsim combines two metrics for SLA violation metric (a) Performance degradation due to Overload Time Fraction (OTF) and (b) performance degradation due to migration of the VMs (PDM). The SLA violation metric (SLAV) is formulated as:

$$SLAV = OTF \cdot PDM$$

ESV metric Besides SLA violations, energy consumption is also an important evaluation metric. Beloglazov and Buyya (2012) proposed a performance metric (ESV) by multiplying energy consumption and SLA violations. They define ESV as:

$$ESV = E \cdot SLAV \quad (1)$$

4.2 Simulation Setup

For simulation we used the latest version of the Cloudsim (3.0.3). We used 1000 servers from two PM types, HP ProLiant ML110 G4 and HP

ProLiant ML110 G5. We used stochastic utilization model for RAM and bandwidth so we choose 80 GB for RAM and 10 Gbit/s for bandwidth for each PM. Table 2 shows the PMs characteristics. For VMs we use four Amazon EC2s VM instances (Amazon,) (High-CPU medium, extra-large, small, micro). Table 3 shows VM instance types. We used IQR (Inter Quartile Range) with safety parameter of 1.5, for VM allocation policy and MMT (Minimum Migration Time) for VM selection policy. Figure 2 shows the number of VMs used during the simulation period.

Host	HP ProLiant G4	HP ProLiant G5
CPU	1x Xeon 3040 (1860 MHz)	1x Xeon 3075 (2660 MHz)
Cores	2	2
RAM (GB)	80	80
Bandwidth(Gbits/s)	10	10

Table 2: Physical Machine (PM) characteristics.

VM Type	High-CPU Medium	Extra Large	Small	micro
MIPS	2500	2000	1000	500
RAM (GB)	0.85	3.75	1.7	0.633
Bandwidth (Mbit/s)	100	100	100	100

Table 3: VM instance type.

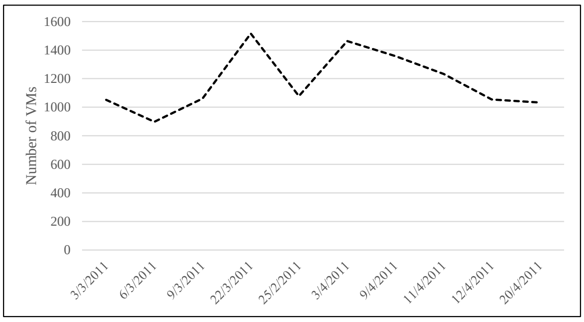

Figure 2: Number of VMs in workload traces dates.

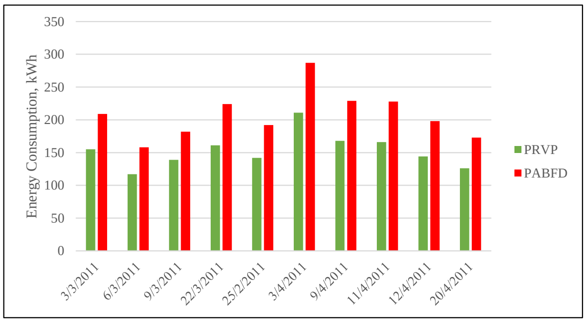

Figure 3: Energy consumption of the datacenter (kWh).

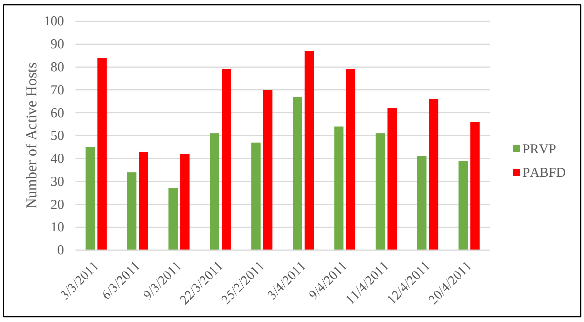

Figure 4: Number of active hosts after consolidation.

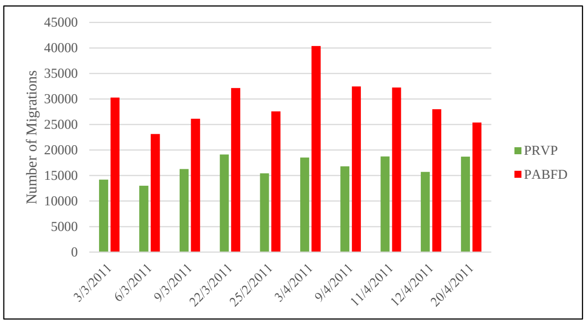

Figure 5: Number of migrations.

5 Results

We compared the performance of PABFD with that of PRVP. Figure 3 shows the total power consumption of the datacenter which show that in average the energy consumption of PRVP is 26% less than PABFD.

Figure 4 shows number of active hosts after consolidation which shows that in average the number of active hosts using PRVP is 33% lower than PABFD.

Figure 5 shows number of migrations which shows that the number of VM migrations is 44% lower in PRVP compared to PABFD.

Figures 6, 7 and 8 show the resource utilization variance of hosts after consolidation by using PRVP and PABFD. The results show that PRVP has lower resource utilization variance compared to PABFD and so better load balances the resources.

The load balancing of resources are relative to their priority. CPU utilization is load balanced well and bandwidth is not because it has the lowest priority in PRVP. The order of priorities in PRVP is CPU, RAM and then bandwidth which completely matches with the amount of load balancing shown in Figures 6, 7 and 8. It is though possible that in cloud environments with more variety of VMs, bandwidth will load balance better due to more host resource availability categories.

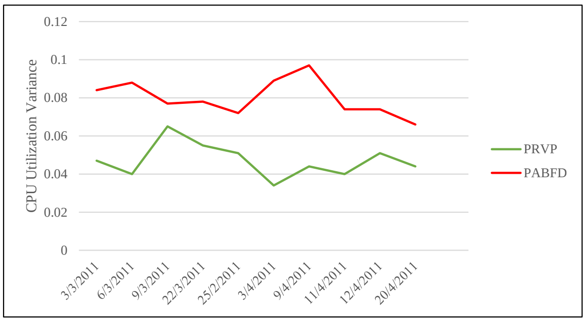

Figure 6: CPU utilization variance.

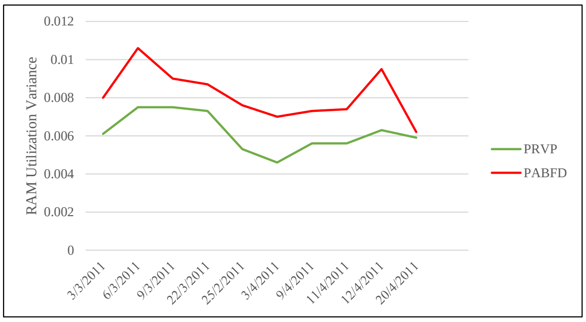

Figure 7: RAM utilization variance.

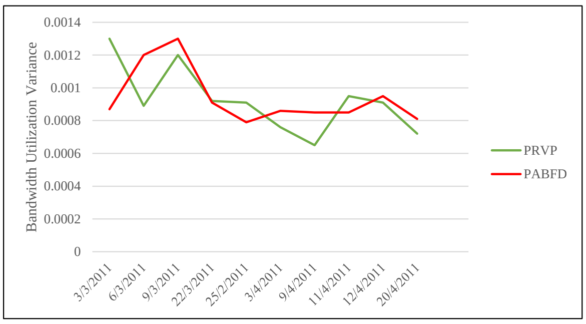

Figure 8: Bandwidth utilization variance.

Figure 9 shows comparison of algorithms with regard to SLA violations. Results show that the number of SLA violations in `PRVP` is 32% lower than `PABFD` Figure 9 as the better load balancing

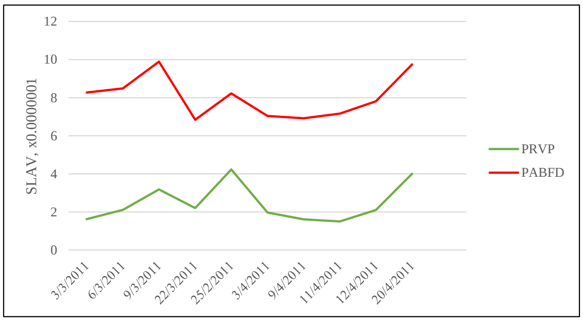

Figure 9: SLA violation metric.

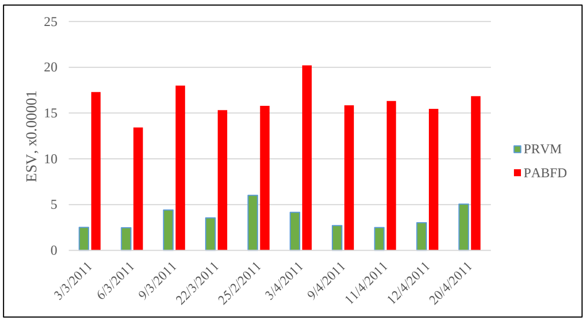

Figure 10: ESV metric.

lead to less host overload and more resource availability decreasing both OTF and DPM metrics.

6 CONCLUSIONS

In this paper we proposed `PRVP`, an online VM placement algorithm which simultaneously reduces power consumption and keeps the performance degradation low. The main idea was to place VMs in PMs where the resource demands of the VM matches with resource availability of the VM. We simulated a real cloud environment by Cloudsim using CoMon project data and showed that `PRVP` outperforms `PABFD` by 26% decrease in energy consumption, 44% less migrations and 32% less SLA violations. Our algorithm performs significantly better than `PABFD` in terms of ESV metric as it has less energy consumption and less SLA violations. We also showed that the amount of load balancing is relative to (and controllable by) the amount of priority given to VM placement in `PRVP`.

References

Amazon. Amazon ec2 instance types. `http://aws.amazon.com/ec2/`.

Luiz André Barroso and Urs Hölzle. 2007. The case for energy-proportional computing. *IEEE computer*, 40(12):33–37.

Anton Beloglazov and Rajkumar Buyya. 2012. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice and Experience*, 24(13):1397–1420.

Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. 2012. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 28(5):755–768.

Rodrigo N Calheiros, Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. 2011. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1):23–50.

Marcos Dias De Assuncao, Jean-Patrick Gelas, Laurent Lefèvre, and Anne-Cécile Orgerie. 2012. The green grid5000: Instrumenting and using a grid with energy sensors. In *Remote Instrumentation for eScience and Related Aspects*, pages 25–42. Springer.

Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. 2007. Power provisioning for a warehouse-sized computer. In *ACM SIGARCH Computer Architecture News*, volume 35, pages 13–23. ACM.

Chris Hyser, Bret Mckee, Rob Gardner, and Brian J Watson. 2007. Autonomic virtual machine placement in the data center. *Hewlett Packard Laboratories, Tech. Rep. HPL-2007-189*, pages 2007–189.

Joe Wenjie Jiang, Tian Lan, Sangtae Ha, Minghua Chen, and Mung Chiang. 2012. Joint vm placement and routing for data center traffic engineering. In *INFOCOM, 2012 Proceedings IEEE*, pages 2876–2880. IEEE.

Xiaoqiao Meng, Vasileios Pappas, and Li Zhang. 2010. Improving the scalability of data center networks with traffic-aware virtual machine placement. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE.

KyoungSoo Park and Vivek S Pai. 2006. Comon: a mostly-scalable monitoring system for planetlab. *ACM SIGOPS Operating Systems Review*, 40(1):65–74.

Jing Tai Piao and Jun Yan. 2010. A network-aware virtual machine placement and migration approach in cloud computing. In *Grid and Cooperative Computing (GCC), 2010 9th International Conference on*, pages 87–92. IEEE.

C. Scroggie. 2008. Predictions 2009: Symantecs craig scroggie. Website. `http://searchstorage.techtarget.com.au/articles/28102-Predictions-2-9-Symantec-s-Craig-Scroggie`.

Ibrahim Takouna, Wesam Dawoud, and Christoph Meinel. 2012. Energy efficient scheduling of hpc-jobs on virtualize clusters using host and vm dynamic configuration. *ACM SIGOPS Operating Systems Review*, 46(2):19–27.