Journal of Web Engineering, Vol. 15, No.5&6 (2016) 484-500

SYSTEM MODELING AND EVALUATION ON FACTORS INFLUENCING POWER AND PERFORMANCE MANAGEMENT OF CLOUD LOAD BALANCING ALGORITHMS

S SURESH

Department of Computer Science & Engineering,

Adhiyamaan College of Engineering, Hosur, Tamilnadu, India
ssuresh.siv.72@gmail.com

S SAKTHIVEL

Department of Computer Science & Engineering,

Sona College of Technology, Salem, Tamilnadu, India
sakvel75@gmail.com

Received February 1, 2016
Revised April 9, 2016

Cloud is an on-demand IT resource provisioning technology uses server virtualization and load balancing as the underlying techniques. Power and performance management are the major concern of cloud to achieve Total Cost Ownership (TCO) in terms of user acceptance and societal importance. In this concern, there is a need to investigate the power and performance influencing factors to design a novel cloud load balancing algorithms with respect to recent hardware and software advancements. Hence, the work studied these approaches to allocate only required amount of virtual servers for varying cloud workload. In this regard, the cloud system model is designed and evaluated for different scenarios like reactive system model, cloud workload and different scaling and sizing of Virtual Machine (VM) servers for various load balancing algorithms. The simulation results infer that the launching of an optimal number of virtual machines, the cost of VM setup time in the data centre, control considerations - dynamic regulation of frequency of controller invocation, adaptive algorithms instead of dynamic algorithms, and multi-core CPU architectures are to be considered while implementing cloud load balancing methods. Appropriate consideration of the above-mentioned parameters is required to make a powerful, flexible and cost-effective load balancing methods for power and performance management for cloud data centre.

Key words: Cloud computing, Server virtualization, Load balancing, Performance, Power management, Modeling and evaluation
Communicated by: D. Schwabe & S. Murugesan

1 Introduction

Cloud computing is a conceptual model for enabling on-demand network access to a shared pool of configurable IT computing resources that can be rapidly provisioned and released with minimal management effort [31, 10]. Server virtualization is an art of slicing the IT hardware resources into logical partitions i.e., Virtual Machine (VM) by implementing software virtualization technology i.e., Virtual Machine Monitor (VMM) on top of the IT hardware and converting physical infrastructure into virtual appliances. Server virtualization provides a means for server consolidation and allows for on-demand allocation and migration of these VMs, which run the applications on physical servers. It is

recognized that the dynamic consolidation of application workloads, through live migration of virtualized servers, helps to increase server utilization, allowing reducing the use of computing resources and the associated power demands. Specifically, the ability to dynamically move application workloads around in a virtualized server environment that enables some physical servers to be turned off during the periods of low activity, and when the demand increases, it allows for bringing them up back and distributing the application workloads across them. Thus, virtualization technologies promise great opportunities for reducing energy and hardware costs through server consolidation. This enables an efficient way of running a cloud data centre from a power management point of view [19]. Load balancing is an optimization technique that distributes the service requests to resources evenly across all the available virtual servers or nodes in the whole cloud to avoid a situation where some virtual servers or nodes are heavily loaded while others are idle or doing little work. It helps to increase utilization and throughput, lower latency, reduce response time, and avoid system overload to attain a high customer satisfaction. It further prevents bottlenecks of the system which may occur due to load imbalance [28].

Server virtualization is a technology that reduces power consumption by reducing the computing waste and improving the server resource utilization. Several commercial and open source solutions like VMware, XEN, Virtual Box and KVM offer software packages to enable the physical server into virtual appliances. Similarly, hardware vendors say Intel and AMD have also built virtualization enhancements to the x86 instruction set to support hardware assisted virtualization [30, 33]. The ability to migrate VMs at run-time enables the technique of energy efficient dynamic virtual server consolidation applied at the cloud data centre level. The proliferation of virtualization has a potential to drive wider adoption of the concept of terminal servers and thin clients, which have also been used in the Green IT practices. Generally, there are two ways in which a VMM can imply power management. First, VMM can monitor the overall system performance and appropriately apply Dynamic Voltage and Frequency Scaling (DVFS) or any Dynamic Component Deactivation (DCD) techniques to the system components. Second, VMM can leverage the power management policies applied by the VMs using the application level knowledge to enforce system-wide power limits in a coordinated manner [32].

1.1 Motivation

Cloud data centre continue to deploy virtualized services, as server virtualization allows new and better solutions to the problems of existing data centre by allowing rapid and flexible resource provisioning. However, cloud computing consists of many issues[7], chief among them is how to effectively manage the VM life cycle to manage Service Level Agreements (SLAs), and guarantee Quality of Service (QoS) satisfaction and minimize SLA violations to balance the energy consumption and performance. This is vital, because scaling capacity to match current demand, service providers can get additional work done by repurposing unneeded servers for other tasks [39].

In this regard, the work considered the performance constrained power management policy, such that the average power consumption of the servers is minimized and the average task response time does not exceed the given performance limit. The cloud system load balancer maintains the utilization of all virtualized servers and distributes the requests to virtual servers in a way that is power efficient. In the sense, the cloud load balancer has to maintain the availability of virtual servers while reducing the total power consumed by the cloud. Thus, the main objective of this work is to explore how server

virtualization with multi-core CPU hardware can allow for application agnostic solutions when dealing with challenges related to power and performance management in cloud data centre.

In this concern, the work, first study the existing scenario of power management, server virtualization solutions, and state-of-the-art work. Second the work, model, evaluate and characterize the power and performance tradeoff of the performance constrained power management policy; for various scenarios for different load balancing algorithms using CSIM simulation toolkit. Subsequently, it investigates the factors that impact the effectiveness of consolidation in cloud environments with respect to CPU hardware advancements.

The rest of the paper is organized as follows. Section 2 gives an overview of the power management techniques, server virtualization solutions on power management and reviews the state-of-the-art work in power and performance management in cloud computing data centre. Section 3 presents the system modelling and simulation, evaluation and analysis of the performance constrained power management cloud load balancing algorithm controller. Section 4 presents the statistical results conducted in a simulated environment and interpret results. Further, it elaborates power and performance influencing factors from the study. Section 5 concludes the work with the future scope.

2 Background and Related Work

This section illustrates the existing scenario of power management, server virtualization solutions for power management and the state of the art work on power management.

Power management is an important consideration from an economic point of view since effective power management improves operational efficiencies and increases compaction. In recent years, researchers have proposed several techniques for managing power consumption in cloud data centre [24, 40]. These techniques are broadly classified as DVFS [13], power state transitioning and server consolidation based approach [11], workload management or task scheduling approach [38], thermal-aware power management approach [5]. Further, some techniques address the issues related to cooling in data centre [38].

2.1 Server virtualization solutions of power management

KVM supports hibernate and sleep / standby power state of VM. On hibernation, the guest OS dumps the memory state to a hard disk and initiates powering off the computer. The hypervisor translates this signal into the termination of the appropriate process. On the next boot, the OS reads the saved memory state from the disk, resumes from the hibernation, and reinitializes all the devices. During the next boot, the BIOS should recognize the sleep or stand by state, and instead of initializing the devices jump directly to the restoration of the saved device states [22]. The XEN hypervisor power management [2] is similar to the Linux on demand governor described for KVM. XEN supports ACPI P-states implemented in the ‘cpufreq’ driver. The system periodically measures the CPU utilization, determines the appropriate P-state, and issues a platform dependent command to make a change in the hardware power state. Similarly to the Linux power management subsystem, XEN contains four different governors, for setting the highest and lowest available clock frequency, setting the CPU frequency specified by the user, choosing the best P-state according to current resource requirements [37]. Apart from governors, XEN also supports offline and live migration of VMs, which can be leveraged by power-aware dynamic VM consolidation algorithms [23]. Similar to XEN, VMware

products namely VMware ESX Server and VMware ESXi supports host-level power management via DVFS. The system monitors the CPU utilization and continuously applies appropriate ACPI's P-states [36]. VMware VMotion and VMware Distributed Resource Scheduler (DRS) are other services that operate in conjunction with ESX Server and ESXi for live migration and load-balancing policy [35].

2.2 Related Work

Buyya et al [9] have introduced the energy-efficient virtual machines provisioning of cloud architecture to provide required QoS requirement with SLA between the consumer / broker and provider. In [4], authors proposed a fixed utilization thresholds energy-efficient resource allocation policies and scheduling algorithms for energy aware management in cloud computing. Similarly, in [3] authors proposed an adaptive heuristics for energy efficient dynamic consolidation of VMs, by mining the resource usage and historical data from VMs. The work studied for various host overloading detection policies for the VM selection. Buyya et al [8] propose cloud load management architecture comprises dispatcher, local and global managers. Local managers migrate the VMs in the case of SLA violation; Global managers receive information from local managers and issue commands for turning on/off servers, applying DVFS or resizing VMs. Ghosh et al [17] developed an out-of-band management processors model to save energy in the data centre. These strategies typically used for managing a server remotely, to satisfy the I/O requests from a remote server.

In [21] authors, proposed power and performance constrained load distribution strategy for multiple heterogeneous multi-core server processors across clouds data centre. The multivariable optimization problems are solved and demonstrated with some numerical examples for two different models of core runs for zero speed and a constant speed. Goudarzi et al [18] proposed a global load balancing strategy for a heterogeneous cloud data centre considering response time sensitive applications. The algorithm chooses VM's migration or VM assignment from one data centre to another by considering the heterogeneity of VMs and data centre, cooling system inefficiency, and peak power constraint in each data centre. Chen et al [12] evaluated energy consumptions for data centre using load balancing and server consolidation theoretically. They concluded that server consolidation helps to improve resource utilization by consolidating many VMs residing on multiple under-utilized servers and load balancing helps to decrease energy consumption by dispensing the load and decreasing the resource consumption, and decreasing energy consumption.

Gandhi et al [16], have studied the problem of obtaining the optimal power allocation by allocating power among the heterogeneous virtualized server farm (M/M/1 queuing model) by minimizing the mean response time of web applications so that performance also can be obtained optimally. Similarly, in [41], the authors developed a switch-on or switch-off energy proportional model which turns servers on and off to adjust the number of active servers (M/G/1 queuing system) based on the workload. The developed system provides controllable and predictable quantitative control over power consumption with theoretically guaranteed service performance. Furthermore, Suresh et al [34] carried out the qualitative and quantitative analysis of the multi-core impact on power and performance management over faster clock speed CPU processor in VM server cluster for the diverse cloud workloads. Analytic and simulation results showed that multi-core virtualized model yields the smallest mean delays results, over the faster clock speed CPU processor.

In [26] authors, proposed an Ant Colony meta-heuristic algorithm for load balancing of nodes in a cloud environment. In addition, Babu et al [1] proposed a honey bee foraging based global scheduling technique to balance the load and priorities of tasks to avoid heavily loaded VMs. Though this strategy reduces the response of time of VMs and improves the overall throughput, failed to investigate the

power consumption. Similarly, Dalapati et al [15] studied a meta-heuristics based green scheduling algorithm for power consumption management in cloud computing. This strategy uses bee colony for service rescheduling and ant colony for optimizing power consumption.

As mentioned above, there are many works carried out like resource management techniques, novel load balancing techniques, sizing and scaling of VMs for power and performance management. However, no work provided an evaluation and comparative study of power and performance tradeoff of load balancing mechanisms from a comprehensive point of view; leveraging an integration of many useful properties that can be utilized in cloud data centre to help in the design of new algorithms.

3 Research Approach - System Design, Model, and Implementation

3.1 System Specification

A simulation model is built with server clusters (each server is assumed a VM) and experimented with policies to switch on and switch off VMServer, based on average utilization of the VMServer CPU resources using a utilization threshold model. The VMServer clusters have 12 VMserver machines (for a single core VMServer case). The request service time is general and it's mean is 0.200 seconds for each VMServer in the cluster. A load balancer (VMM) controls the VMServer clusters which distributes arrived requests to VMServer cluster in a Stateless Server Selection {Round Robin and Random}, State-based Server Selection {Active Monitoring, Fastest Response, and Random Subset} and executes a policy to switch on / off the VMserver at the VMserver cluster machines. It can be noted that switch on server consumes 0.200 KWatts and a switch off server consumes 0.005 KWatts. The time to switch on and switch off a machine is 4 and 2 seconds respectively. Generally, a VM machine must be switched on / off for certain amount i.e., sample_interval of time before it changes its power state. The SLA for the system needs the VMServer cluster to maintain the mean response time not to exceed 0.250 seconds and the 99% response time not to exceed 0.500 seconds.

3.2 Algorithm of the simulation model

The performance constrained power management policy for various load balancing algorithms is depicted as state transition diagram in figure 1. The policy assumes parallel VMServer cluster serving the incoming cloud workload requests. The incoming requests are routed to any of the available VMServers based on the load balancing algorithm. In a regular interval say Sample_Interval or Control Interval (CI), the monitor or controller of the load balance Manager collects statistics to determine the VMServer cluster CPU utilization for the last sample period. From the collected statistics, the controller determines the average CPU utilization for all switched on VMServers. If the average CPU utilization is greater than a high threshold then it switches on one more VMServer from the VMServer cluster. If the average CPU utilization is less than a low threshold then the load balance manager switch off one more VMServer. At the end of the simulation time, the controller prints avgResponseTime, avgActiveNodes, powerConsumption, avgUtilization.

3.3 Overview of CSIM simulator

The simulation model is developed using process-oriented discrete event simulator CSIM 20. A CSIM model is a C program that uses the functions and procedures in the CSIM 20 library to implement process-oriented, discrete-event simulation. CSIM provides a complete set of data structures that can

be used to construct models of almost any kind of system, at any level of complexity and detail. The data structures supported by CSIM are: Process to model simulation entities, Facility to model resources, Storage to model resources that are partially allocated to processes, Buffer to model buffers, Event to synchronize and control interactions between processes, Mailbox to exchange information between processes, Tables, Qtables, Meters, and Boxes to collect explicit statistics, Stream of random numbers to generate multiple streams of samples from specified probability distributions [27].

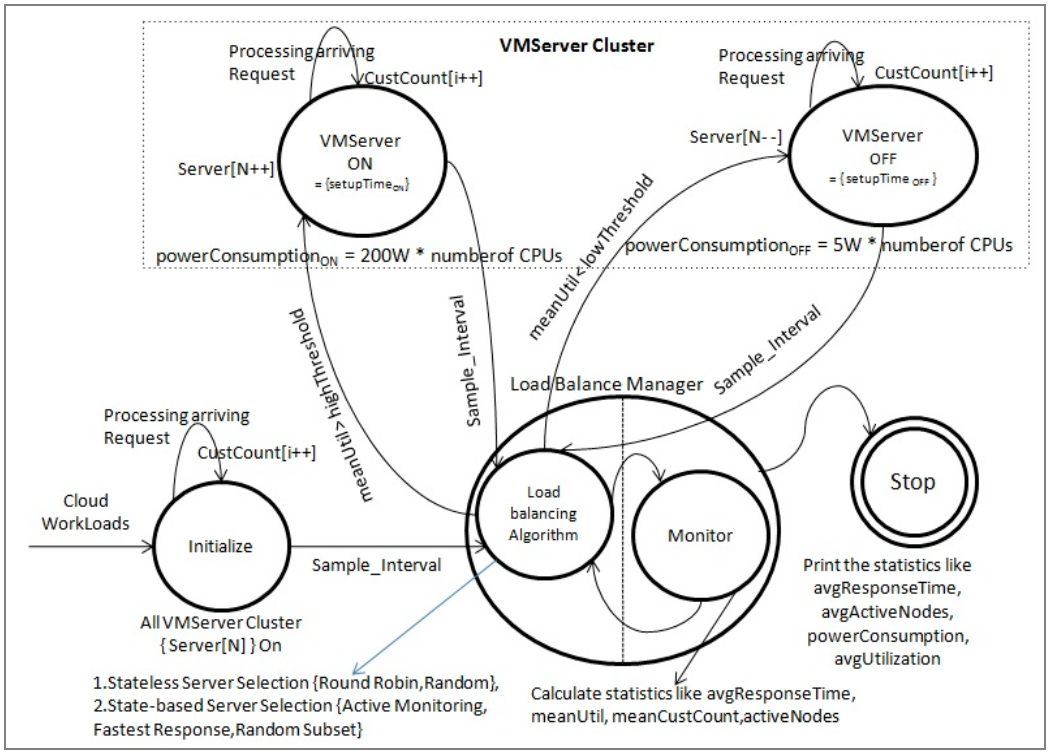

Figure 1 The state transition diagram of the performance constrained power management policy for various load balancing algorithms for the cloud.

3.4 Implementation of the simulation model

The model in the simulator consists of four main simulation entities, a client, the internet, a load balancing manager and VMServer Clusters. The clients make periodic requests and wait for a response to each request. Each request is submitted to the client's load balancing manager for selection of a VMServer Cluster through the internet. Then the client's load balancing manager selects the server that will be used to process the request and forwards the request to the selected server. When the server receives the request it is placed into a priority queue. The request waits in the queue until a VMServer connection becomes available. When a connection is available the server processes the request based on the priority of the request and sends the response to the requesting client. The complexity of the algorithm depends on the number of times the controller is invoked i.e., depends on 'sample_interval'.

In CSIM, processes appear to operate simultaneously with other active processes at the same points in simulated time similar like multi-threading. The CSIM process manager creates this illusion

by starting and suspending processes as time advances and as events occur. The function of a C program is straightaway converted into the process by simply using create () statement followed by the C function. For an example, the client process can be created by declaring create ("client") followed by the client function. Cloud consist of complex and heterogeneous user request that highly dynamic variations in its intensity as well as characteristics. The complex and heterogeneous user request is imitated by generating priority requests. It is implemented by including the set_priority (long new_priority) function. The value of the new_priority can be 1, 2, 3, 4 and so on. Higher the value, higher the priority. Similarly, the highly dynamic variation workload intensity is implemented by non-constant mean arrival rates using a vector of mean arrival rates; which is implemented using CSIM hold(exponential(mean of interarrival)) function. In CSIM, the simulation time is passed by using hold () statement. The cloud's server response or service time is imitated by setting the service discipline of the VMserver as pre-empt resume and implemented in CSIM by set_servicefunc(server, pre_res).

3.5 Scenarios considered for the research work

As the work interested in investigating the factors influencing power and performance management of cloud load balancing algorithms, with respect to server virtualization and CPU hardware advancements, the work considers various scenarios shown in figure 2. In all the scenarios, the processing capacity of all the servers are assumed to be the same and there are '*n*' applications are running. In all the scenarios, '*m*' is the processing capacity of a single core CPU and there are '*n*' CPU cores are assumed.

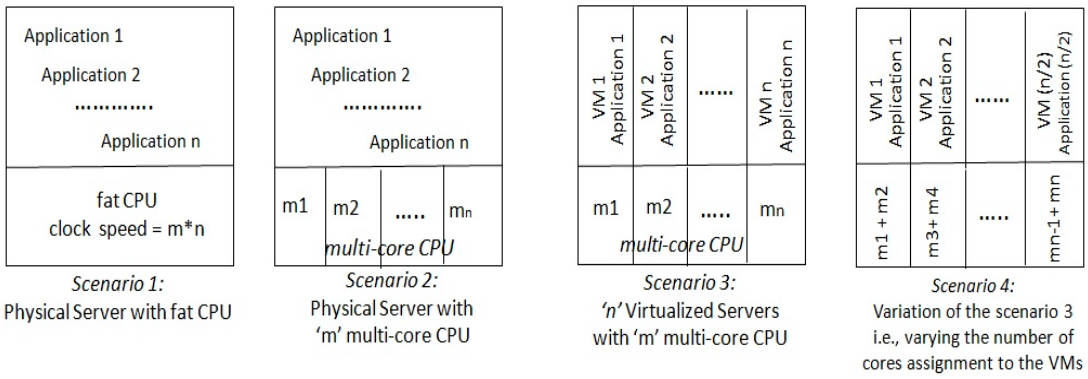

Figure 2 Various scenarios considered for the research work.

Scenario 1:- Physical server with higher frequency single CPU (or fat CPU): This system is modeled as M/G/1 queuing model with service rate $m*n*\mu$. It is implemented using CSIM facility, FACILITY physical_server = facility ("physical fat CPU"). A client request process typically uses a server for a specified interval of time which can be implemented in CSIM by use (physical_server, hyperx (mean service time, variance)).This scenario is considered as the base condition.

Scenario 2:- Physical Server with 'm' multi-core CPU: This system is modeled as $(M/G/1)^n$ where each CPU core service rate is $m*\mu$. It is declared using CSIM multiserver server facility which contains a single queue and multiple servers (where each server can be assumed be a core), using CSIM facility set FACILITY server = facility_ms ("multi-core physical server", n). The incoming requests are straight away forwarded to the available CPU core / server.

Scenario 3: - 'n' virtualized servers in 'm' multi-core CPU physical server: This system is modeled as n_*(M/G/1) where each CPU core service rate is $m_*\mu$. As far as this scenario is considered, each virtualized server (VMServer) is assigned a single core. This scenario is implemented using CSIM facility set, FACILITY server[n]; server = facility_set (server, "Virtualized Servers", n); where each facility has its own queue and its server. As this model consists an array of servers, the incoming requests manually forwarded to the appropriate server using the load balancing algorithm. In addition, the scenario is exercised by scaling down the VM and core together.

Scenario 4: - Varying virtualized servers for varying multi-core CPU physical server: This model is the variation of the scenario 3 in which for each VM the assigned core is varied from 2 to 3 and so on. Consequently, the number of virtualized servers are reduced to n/2 and n/3. For an example of virtualized servers with two core case, the number of consolidated virtualized servers are reduced to n/2.

3.6 Cloud load balancing algorithms

The designed system is to exercise the effective performance of five different load balancing algorithms that distribute the workload among a set of servers. These are (i) Round Robin Server Selection is the simplest method where each server takes a turn. The load balancer selects a server in linear order from a list of servers. When the end of the list is reached, it starts over at the first server in the list (ii) The Random Server Selection method selects a server randomly using a uniform pseudo-random variable. (iii) Active Monitoring Load Balancer implies that the processing element with the lightest workload is selected by considering server performance. (iv) Random subset server selection algorithm selects two or more servers randomly from the list of servers and then it selects the server with the lowest workload from the servers selected [25, 14]. (v) Fastest response time load balancer takes into consideration the time each server is taking to respond and then decides to send the request to the server that providing the fastest response time.

The factors that were identified for evaluating the algorithms are the cloud workload, the number of load balancer managers, type of servers or scenarios (non-virtualized single fat CPU server, non - virtualized multi-core CPU server, virtualized servers), number of servers (VM scaling) with {single CPU, multiple CPU}, algorithm invocation sample interval or control interval (CI). The experiments that were chosen were more as a matter of personal choice for comparing the performance of the various algorithms, as opposed to attempting to achieve any particular result. Also, there is a huge array of combinations that can be performed; therefore, a representative set of experiments were performed to satisfy the goals of this work. Furthermore, it is implied that the overall system workload is composed of multiple and independent heterogeneous applications which also correspond to a cloud environment [6].

4 Results and Discussion

The simulation experiments and the results are presented here. Experiments were performed using the designed simulator discussed in section 3 by varying the factors. The initial analysis of the simulation data showed that the results of the experiments between the different server configurations were very similar. The difference in magnitude of the server utilization was inversely proportional to the magnitude of difference in the number of servers. Therefore, since the results of the experiments are so close for each server group, only one set of the data is presented. The data set that was chosen is the data set for scenario specific. The statistical result for scenario1 (non-virtualized physical fat CPU) is: mean response time (0.017 sec), 99% response time (0.03267 sec), utilization (0.102) and power consumption = 1728 KWH. Similarly, the statistical result for scenario2 (non-virtualized single

physical multi-core CPU) is: mean response time (0.2000 sec), 99% response time (0.2000 sec), utilization (0.2041), power consumption =1728 KWH. For both the non-virtualized case the resource utilization ratio is poor. However in the multi-core case, the utilization rate is higher than the fat CPU case. As there isn't power management policy is enforced for both the cases there is no power improvement is achieved. This lower utilization encouraged the need for virtualized servers. Table 1(a) gives the statistical values for the stateless load balancing algorithm for the single core VMserver clusters. In the stateless algorithms, Round Robin policy satisfied SLA for various scenarios (highlighted). However, the SLA metric is satisfied only for when CI=30 sec and for all other CI values the SLA metric is violated; Consequently, The value CI=30 sec makes Setup Time in hours high. In addition, depending on the parameters like number of VMs and CI the SLA and power saving differs for Round Robin. In addition, the simulation reveals that even for the lesser number of VM Server (VMs=7) good power savings (425.18 KWH) is possible; further, there isn't a huge difference in Setup Time. However, Random policy did not satisfy the SLA for any scenario. The reason is, over a small number of requests the load may not be balanced exactly evenly i.e., the possibility that the load may not be evenly balanced over a small number of requests.

Table 1(a) Parametric comparisons of stateless algorithms for single core VM

Scenario		Round Robin				Random			
VMs	CI (sec)	respTime in seconds		Setup Time in hours	Power (KWH)) usage	respTime in seconds		Setup Time in hour	Power (KWH)) usage
		Avg.	99%			Avg.	99%		
12	**30**	**.2249**	**.4544**	**78.34**	**521.26**	.2378	.5510	78.33	521.26
	60	.2456	.5718	39.34	517.80	.2526	.6662	39.34	517.92
	90	.2624	.7075	26.30	520.54	.2763	.8059	26.30	520.85
11	**30**	**.2253**	**.4579**	**78.46**	**508.14**	.2383	.5529	78.46	508.10
	60	.2393	.5673	39.39	506.44	.2524	.6624	39.39	506.50
	90	.2612	.7093	26.31	509.48	.2743	.8037	26.32	509.35
10	**30**	**.2259**	**.4658**	**78.58**	**491.92**	.2393	.5607	78.57	492.04
	60	.2392	.5639	39.45	491.75	.2528	.6622	39.45	491.83
	90	.2644	.7123	26.36	495.40	.2780	.8120	26.36	473.61
9	**30**	**.2263**	**.4702**	**78.75**	**473.53**	.2403	.5654	78.75	474.88
	60	.2480	.5795	39.54	474.97	.2557	.6796	39.53	474.88
	90	.2619	.7180	26.42	478.16	.2757	.8212	26.42	478.36
8	**30**	**.2272**	**.4783**	**79.02**	**451.33**	.2420	.5778	79.02	451.34
	60	.2463	.5723	39.67	453.72	.2547	.6781	39.67	453.74
	90	.2668	.7493	26.49	456.85	.2813	.8560	26.49	456.98
7	**30**	**.2291**	**.4845**	**79.31**	**425.18**	.2443	.5886	79.29	425.21
	60	.2562	.6234	39.79	428.02	.2661	.7284	39.79	427.98
	90	.2697	.7628	26.59	432.73	.2856	.8802	26.60	432.85

Table 1(b) gives the statistical values for the stateful load balancing algorithms for single core VMs. The results revealed that Random Subset, Active Monitoring stateful Load Balancing Algorithms satisfied SLA for all scenarios with slight power saving variations based on the parameters like CI and number of VMs. The interesting observation is stateful load balancing algorithm performs much better than a stateless algorithm. Specifically, Random subset algorithm satisfies the SLA for different CI i.e., CI=30 and 60 sec. However, comparing with Random Subset, Active Monitoring performance

degrades and it satisfies the SLA only when CI=30. This could be improved if more than one parameter is used for load estimation and the estimation will be more reliable than cases where only one parameter is used. In addition, the CI interval decides the setup Time of the VMs. Furthermore, as the number of deployed VM servers are reduced, the load balancing algorithm needs the frequent invocation of the control algorithm to satisfy the SLA leads to high setup time (Table 1(a) and Table 1(b) for 7 VMs scenario). As the Fastest response server selection did not satisfy SLA for any of the Scenarios, it is not shown in Table 1(b). The reason may be, the static Threshold values did not allow a smooth transition from under loaded to overloaded conditions, lead to high fluctuations and the performance degradation. Figure 3 gives Power usage comparisons of stateful and stateless load balancing algorithms for single core VMs. From the figure 3, it is evident that random subset gives good power savings for different VM scaling and CI; however, active monitoring gives overall good power savings of 424.94 KWH for VM=7 and CI=30. This leads to the implication that load balancing algorithm should consider cloud workload variability and the adaptive algorithm for the appropriate parameter settings.

Table 1(b) Parametric comparisons of stateful load balancing algorithms for single core VMs

Scenario		Random Subset				Active Monitoring			
VMs	CI (sec)	respTime in seconds		Setup Time in hours	Power (KWH)) usage	respTime in seconds		Setup Time in hour	Power (KWH)) usage
		Avg..	99%			Avg..	99%		
12	**30**	**.22127**	**.3844**	**78.333**	**521.25**	**.2176**	**.3891**	**78.33**	**520.57**
	60	**.23435**	**.50064**	**39.34**	**517.84**	.2311	.5186	39.35	517.59
	90	0.2571	.64415	26.297	520.58	.2531	.6512	26.29	520.40
11	**30**	**.2213**	**.38488**	**78.463**	**508.11**	**.2177**	**.3892**	**78.46**	**507.73**
	60	**.23416**	**.50052**	**39.391**	**506.41**	**.2309**	**.5017**	**39.40**	**506.24**
	90	.25543	.64043	26.314	509.26	.2518	.6469	26.31	509.27
10	**30**	**.22178**	**.3880**	**78.575**	**491.98**	**.2180**	**.3912**	**78.57**	**491.65**
	60	**.23392**	**.4982**	**39.455**	**491.78**	**.2307**	**.5010**	**39.46**	**491.66**
	90	.25809	.64814	26.36	495.21	.2538	.6567	26.36	495.29
9	**30**	**.2220**	**.38915**	**78.74**	**473.61**	**.2183**	**.3921**	**78.74**	**473.40**
	60	0.2364	.51801	39.54	474.82	.2328	.5293	39.54	474.60
	90	.25585	.65127	26.42	478.20	.2522	.6572	26.42	478.15
8	**30**	**.22266**	**.39328**	**79.023**	**451.30**	**.2189**	**.3960**	**79.01**	**451.05**
	60	**.23424**	**.50043**	**39.67**	**453.72**	**.2310**	**.5013**	**39.67**	**453.68**
	90	.26003	.6849	26.49	456.89	.2560	.6971	26.49	456.87
7	**30**	**.22355**	**.39241**	**79.304**	**425.13**	**.2195**	**.3957**	**79.30**	**424.94**
	60	.24397	.55378	39.79	427.96	.2392	.5650	39.78	427.81
	90	.26593	.7019	26.59	432.82	.2582	.7141	26.59	432.67

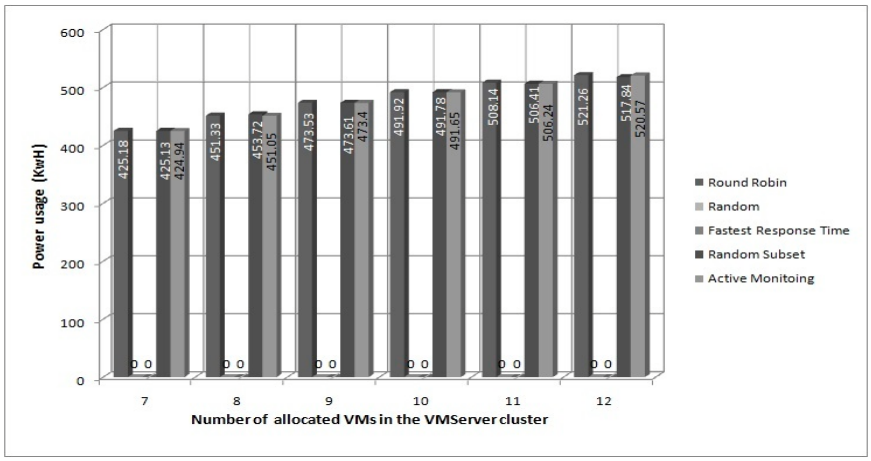

Figure 3 Power usage comparisons of stateful and stateless load balancing algorithms for single core VMs.

Table 2 gives a parametric comparison of stateful and stateless Load Balancing Algorithms for the multi-core VM Servers case. All algorithms satisfy the SLA, irrespective of the experimental factors. Especially the Random Subset algorithm works better in all the cases. The specific feature of the Random Subset algorithm condition is that many nodes simultaneously select a certain node for running a process is avoided. As the setup time is highly reduced for multi-cores, this motivates the need of multi-cores usage for the VM servers instead of single core CPU. However, the power savings highly differs comparing with the single core CPU. Especially when the VM has many cores the power savings could not be reduced. This makes sense of adaptive techniques in the load balancing algorithm.

Table 2 Comparisons of stateful and stateless load balancing algorithms for multi-core VMs

Scenario		Round Robin				Random			
VMs / Core	CI /sec	respTime in seconds		Setup Time in hours	Power (KWH)	respTime in seconds		Setup Time in hour	Power (KWH)
		Avg.	99%		usage	Avg.	99%		usage
6/2	**30**	**.2007**	**.2000**	**79.67**	**788.39**	**.2018**	**.2749**	**79.67**	**788.27**
	60	**.2010**	**.2000**	**39.97**	**796.30**	**.2021**	**.2881**	**39.97**	**796.25**
	90	**.2010**	**.2079**	**26.70**	**806.24**	**.2023**	**.2929**	**26.70**	**806.17**
4/3	**30**	**.2001**	**.2000**	**80.52**	**951.28**	**.2003**	**.2000**	**80.52**	**951.30**
	60	**.2001**	**.2000**	**40.47**	**962.06**	**.2003**	**.2000**	**40.47**	**961.98**
	90	**.2002**	**.2000**	**27.05**	**973.21**	**.2003**	**.2000**	**27.05**	**973.12**
		Random Subset				Active Monitoring			
6/2	**30**	**.2005**	**.2000**	**79.67**	**788.13**	**.2005**	**.2000**	**79.67**	**788.38**
	60	**.2007**	**.2000**	**39.97**	**796.15**	**.2007**	**.2000**	**39.97**	**796.33**
	90	**.2008**	**.2000**	**26.70**	**805.95**	**.2008**	**.2000**	**26.70**	**806.16**
4/3	**30**	**.2001**	**.2000**	**80.53**	**951.30**	**.2001**	**.2000**	**80.52**	**951.33**
	60	**.2001**	**.2000**	**40.47**	**961.92**	**.2001**	**.2000**	**40.47**	**962.07**
	90	**.2002**	**.2000**	**27.05**	**973.12**	**.2002**	**.2000**	**27.05**	**973.12**

4.1 Findings from the study of various algorithms

Our empirical characterization generates fundamental understandings of power and performance tradeoff, in the context of cloud computing that suggests engineering insights for power and performance efficient data centre operations. In-depth analysis of the simulation results reveals a few fundamental insights about the impact of server consolidation with respect to CPU hardware advancements on QoS-aware SLA constrained power management, including,

Server under Utilization: Companies that manage their own servers and data centre often only use 10 to 20 percent (non-virtualized physical fat CPU case and physical multicore CPU case utilization is 0.102 and 0.204142 respectively) of their available computing cycle. The rest of those cycles go to waste. Server and data centre underutilization are the primary reasons of waste and inefficiency in computing. There are several challenges to this approach like Performance Tradeoff because of dynamic resource management for unexpected load, Load Unpredictability, Short Idle Times and energy cost of switching VM to lower power modes is often not worth the potential energy saving from the server on / server off.

The Importance of multi-core CPU*:* Permanently altering the course of computing, the multi-core processor technology provides new levels of performance and energy efficiency. Ideal for multitasking, multimedia, and networking applications, multi-core technology delivers exceptional energy efficient performance for the ultimate computing experience. Non-virtualized and virtualized scenarios show that incorporating multiple processor cores in a single package for delivering parallel execution of multiple software applications enables higher levels of performance and less power consumption typically required by a higher frequency single core processor with equivalent performance.

The Importance of Adaptive algorithms: Reactive algorithm includes knowledge of the communication prior performance but does not consider continuous monitoring of the nodes. Reactive algorithms cannot consider load changes during run time. The proactive algorithm would be more accurate and more efficient scheduling techniques as it includes the VM capabilities and network characterization. The proactive algorithm depends on the combination of knowledge based on all gathered information about the VM resources and different properties of the selected nodes process and task on that node in the public and private cloud. By using gathered information and calculation, proactive algorithms assign the task and for some condition, it should be reassigned them. Some proactive algorithms require the current status of the node and task current situation and progress. Such algorithms are usually harder to implement. Similarly, adaptive algorithms [29] constitute a subset of dynamic algorithms, which go further in their use of system state information.Such information may be used to modify the parameters of the algorithm, or even to choose which workload distributing strategy is used.

The Cost of VM Setup time in the data centre: Table 1(a), Table 1(b) and Table 2, Setup Time field shows that dynamic power management reconfiguration actions come with associated costs. Server switching by addition and removal of a virtual server introduces non-negligible latency to a service that affects the perceived end-to-end response time of users. It means that sleeping servers incur a high setup time to get them back on again. Consequently for the given high setup Time, it is not at all obvious whether sleep states are useful or not.

Control Considerations: In terms of control algorithm invocation, the setup time in hours field of the Table 1(a), Table 1(b) and Table 2 gives an idea of VMM design and deployment need to consider

workload forecasting and frequency of control into account to have a significant impact on the efficiency of the controller and on the performance of the entire system. An adequate choice of workload forecasting techniques to be used by the load balancer control component would result in a higher system performance and power management. The use of effective forecasting algorithms enables the controller to make better configuration decisions to accommodate the future. A dynamic regulation of frequency of controller invocation, the overall system power, and performance stability could be improved. For a sudden workload surge, an adaptive controller algorithm can respond quite early to such a change in the external environment. Consequently, the controller being able to position the system in a more convenient configuration before performance seriously degrades. Subsequently, when the workload becomes more stationary, controller algorithm needs to run occasionally that may contribute to a higher overall stability of the entire system.

The studies suggest some operational optimizations toward performance efficient data centre design and operations. To Summarize,

(1) Virtualized Server architecture is still far from being power aware SLA constrained performance-proportional in that a significant amount of power and performance is lost when the server is virtualized, thus it needs improvement for server consolidation in the data centre for reducing power and performance improvement. Server consolidation for effective cloud data centre should aim to balance a fundamental tradeoff between performance and the energy saving from least usage idle servers and energy overhead and the throughput reduction from hypervisor due to server virtualization.

(2) Virtualized servers power and performance overhead is higher than physical ones, for the computing intensive traffic. The performance overhead from virtualized servers increase as the utilization of physical resources increases.

(3) The power and performance overhead resulted from server virtualization highly depends on the scheduler of the hypervisor used, which in turn is determined by CPU scheduling and load balancing of the hypervisor. Hypervisors should be architect with power and performance objectives while providing the maximal flexibility in server resource management. Resources should be allocated adaptively than dynamically according to the real-time demand, with an objective to optimize the power and performance overhead.

(4) From a given cloud workload the power and performance can be optimized by launching an optimal number of virtual machines.

(5) In a multicore server running multi-processing applications, physical servers, if a multi-core optimization mechanism is absent, performance overhead resulted than virtualized servers. Multicore scheduling algorithms should be incorporated in hypervisor design for virtualized servers and OS design for physical servers, to minimize the performance overhead and power management.

5 Conclusions and Future Work

Power and performance management have become the primary focus of cloud data centre for achieving TCO. Server virtualization and software and hard advancements contribute in easing power and performance management, thereby, reducing the operational costs of the cloud data centre. This work first explained the existing scenario of power management algorithms in the traditional data centre. It then, overviewed the server virtualization solution of power management algorithms and reviewed the state of the art work in cloud power management. The work, then formulated the problem

of power minimization while maintaining the required performance SLAs. Further, it described a study on cloud VMserver Clusters load balancing methods based on simulation methodology to solve the problem of how to take load balancing into virtualized cloud data centre. As part of this, the work designed and developed the concept of stateful and stateless load balancing for various scenarios for cloud computing workload. The results infer that the launching of an optimal number of virtual machines, the cost of setup time in the data centre, control considerations - dynamic regulation of frequency of controller invocation, adaptive algorithms instead of dynamic algorithms, and multi-core CPU architectures are to be considered while implementing load balancing methods. Appropriate consideration of these parameters is required to make a powerful, flexible and cost-effective load balancing methods for power and performance management.

Considerable work still remains to be carried out in this area, both in terms of real web workload traces as input and some more load balancing algorithms for validation of the underlying hypothesis, and in the more detailed exploration of the simulation process.

References

1. Babu LD and Krishna PV, "Honey bee behavior inspired load balancing of tasks in cloud computing environments", Applied Soft Computing journal, vol. 13, no. 5, pp. 2292–2303, 2013.

2. Barham P, Dragovic B, Fraser K and Hand S et.al., "XEN and the art of virtualization", 19th ACM Symposium on Operating Systems Principles (SOSP ' 03), pp.16–177, 2003.

3. Beloglazov and Buyya R, "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 experiments, vol.24, no.13, pp.1397-1420, 2012.

4. Belolazov J, Abawajy and Buyya R "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing", Future Generation Computer Systems, vol.28, no.5, pp.755-768, 2012.

5. Bergamaschi RA, Piga L, Rigo S, Azevedo R and Araújo G, "Data center power and performance optimization through global selection of p-states and utilization rates, Sustainable Computing: Informatics and Systems; vol.2, no.4, pp.198–208, 2012.

6. Bodik P, Fox A, Franklin M J, Jordan M I and Patterson DA, "Characterizing, Modeling, and Generating Workload Spikes for Stateful Services", 1st ACM symposium on Cloud computing, pp.241-252, 2010.

7. Buyya R., "Introduction to the IEEE Transactions on Cloud Computing", IEEE Transactions on Cloud Computing, vol.1, no.1, pp.3-21, 2013.

8. Buyya R and Beloglazov A, "Energy efficient resource management in virtualized cloud datacenters", 10th IEEE / ACM International Conference on Cluster, Cloud and Grid Computing, IEEE Computer Society, pp. 826-831, 2010.

9. Buyya R, Beloglazov A and Abawajy J, "Energy-Efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges", 2010 International Conference on Parallel and Distributed Processing Techniques and Applications, 2010.

10. Buyya R, Yeo CS, Venugopal S and Broberg I Br, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems 2009, vol.25, no.6, pp.599–616.

11. Chang J, Meza J, Ranganathan P and et al., "Totally green: evaluating and designing servers for lifecycle environmental impact", ACM SIGARCH Computer Architecture News, vol.40, no.1, pp.25–36, 2012.

12. Chen F, Grundy J, Schneider J, Yang Y and He Q, "Automated analysis of performance and energy consumption for cloud applications, 5th ACM / SPEC international conference on Performance engineering, ACM, pp.39-50, 2014.

13. Chetsa GLT, Lefevre L, Pierson J and Stolf P, "Beyond CPU frequency scaling for a fine-grained energy control of hpc systems", 24th IEEE International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'12), pp. 132–138, 2012.

14. Dahlin M, "Interpreting Stale Load Information", UTCS Technical Report TR98-20, University of Texas at Austin, 1998.

15. Dalapati P and Sahoo G, "Green Solution for Cloud Computing with Load Balancing and Power Consumption Management", International Journal of Emerging Technology and Advanced Engineering, vol.3, no.3, pp.353–359, 2013.

16. Gandhi A and et al., "Optimal power allocation in server farms", 11th International Joint Conference on Measurement and Modeling of Computer Systems, pp.157–168, 2009.

17. Ghoshc S, Redekopp M and Annavaram M, "Knightshift: shifting the i/o burden in datacenters to management processor for energy efficiency", Computer Architecture, Springer, pp.183–197, 2012.

18. Goudarzi H and Pedram M, "Geographical Load Balancing for Online Service Applications in Distributed Datacenters", IEEE international conference on cloud computing, IEEE Computer Society 2013, pp. 351-358.

19. Graubner, P and et al., "Energy-Efficient Virtual Machine Consolidation", IT Professional, vol.15, no.2, p.28-34, 2013.

20. Jonathan K, "Growth in data center electricity use 2005 to 2010", CA: Analytics Press, 2011.

21. Junwei Li K and Stojmenovic I, "Optimal Power Allocation and Load Distribution for Multiple Heterogeneous Multicore Server Processors across Clouds and Data Centers Qualitative performance Study", IEEE Transactions on Computers,vol.63, no.1, pp.45-58, 2014.

22. Y. Kivity and et al., “KVM: The Linux virtual machine monitor”, Linux Symposium, Ottawa, pp.225–230, 2007.

23. Lef`evre L and Orgerie A C, “Designing and evaluating an energy efficient Cloud”, The Journal of Supercomputing, vol.51, no.3, pp.352–373, 2010.

24. Mittal S, “Power Management Techniques for Data Centers: A Survey”, Technical Report, 2014.

25. Mitzenmacher M, “On the Analysis of Randomized Load Balancing Schemes”, 9th Annual Symposium on Parallel Algorithm and Architectures, pp. 292-301, 1997.

26. Nishant K. and et al, “Load Balancing of Nodes in Cloud Using Ant Colony Optimization”, 14th International Conference on Modelling and Simulation, pp.3-8, 2012.

27. Schwetman H, “CSIM19: A Powerful Tool for Building System Models”, 2001 Winter Simulation Conference, pp.250-255, 2001.

28. Shirazi BA, Hurson AR and Kavi KM, “Scheduling and Load Balancing in Parallel and Distributed Systems”, IEEE CS Press, 1995.

29. Shivaratri NG and Krueger P, “Two Adaptive Location Policies for Global Scheduling Algorithms”, Proceedings of 10th International Conference on Distributed Computing Systems, pp. 502-509, 1990.

30. Smith JE and Nair R, “Virtual Machines: Versatile Platforms for Systems and Processes”, Elsevier, 2005.

31. Sosinsky B, Cloud Computing Bible, New Delhi, WILEY – INDIA, 2012.

32. Stoess J and et al., “Transparent, power-aware migration in virtualized systems”, GI/ITG Fachgruppentreffen Betriebs system, pp. 1–6, 2007.

33. Suresh S and Kannan M, “A Performance Study of Hardware Impact on Full Virtualization for Server Consolidation in Cloud Environment” Journal of Theoretical and Applied Information Technology, vol. 60, no.3, pp.556-567, 2014.

34. Suresh S and Sakthivel S, “A Qualitative and Quantitative Analysis of Multi-core CPU Power and Performance Impact on Server Virtualization for Enterprise Cloud Data Centers”, Research Journal of Applied Sciences, Engg. and Tech,. vol.9, no.6, pp.471-477, 2015.

35. VMware Inc., “VMware distributed power management concepts and use”, Technical Report, 2010.

36. VMware Inc., “vSphere resource management guide”, Technical Report, 2009.

37. Wei G and et al., “The on going evolutions of power management in XEN”, Intel Corporation, Technical Report, 2009.

38. Zapater M, Ayala JL and Moya JM, “Leveraging heterogeneity for energy minimization in data centers”, 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, (CCGrid'12), Ottawa, pp. 752–757, 2012.

39. Zhang Q, Cheng L and Boutaba R, “Cloud computing: state-of-the-art and research challenges”, Journal of Internet Services and Applications, vol.1, no.1, pp.7-18, 2010.

40. Zhang Y and Ansari N, “Green data centers”, Handbook of Green Information and Communication Systems, 2012.

41. Zheng X and Cai Y, “Achieving Energy Proportionality in Server Clusters”, International Journal of Computer Networks, vol.1, no.2, pp.21-35, 2010.