ISSN:2320-0790

An International Journal of Advanced Computer Technology

Adaptive Workload Management for Efficient Energy Utilization on Cloud

M.Prabakaran[1], M. Parthiban[2]
[1]PG Scholar, Computer Science Department, Muthayammal Engineering College, Rasipuram, Tamilandu
[2]Assistant Professor, Information Technology Department, Muthayammal Engineering College, Rasipuram, Tamilandu

Abstract: Cloud infrastructures must accommodate changing demands for different types of processing with workloads and time constraints. Dynamic consolidation of Virtual Machines (VMs) is an effective way to improve the utilization of resources and energy efficiency in Cloud data centers. To find when it is best to rearrange VMs from an overloaded host is an aspect of dynamic VM consolidation that directly influences the resource utilization and Quality of Service (QOS) delivered by the system.

Index Term: Distributed systems, Cloud computing, virtualization, dynamic consolidation, energy efficiency, host overload detection

I INTRODUCTION

Cloud computing is a compute model, not an equipment. In this model customers plug into the cloud to access IT resources which are priced and provided "on-demand". Essentially, IT resources are rented and shared among multiple tenants much as workplace space, apartments, or storeroom spaces are used by tenant. Delivered more than an Internet link, the "cloud" replaces the company data centred or server providing the same service. The Cloud Computing is simply IT services sold and delivered over the Internet. Cloud Computing vendor unite virtualization (one computer hosting several "virtual" servers), automated provisioning Internet connectivity technology to provide the service. Traditional technologies but a new name applied to a collection of older (albeit updated) technologies that are package, sell and deliver in a new way. Cloud Computing involves aggregation of numerous computing, storage and network property into a single entity called cloud into which location-independent computing is perform. Cloud computing is a usual development of the virtualization, Service-Oriented Architecture (SOA) and Utility Computing. Virtualization in computing is the creation of a virtual version of a hardware platform, Operating system, a storage space or network property. A single physical server or machine can be sliced into various virtual machines or VMs, each embodying various property like memory, hard disk, CPU core, etc. Virtualization can be viewed as part of an overall in Information Technology (IT) enterprise that includes autonomic compute, a development in Which the IT environment will be able to manage itself based on efficacy computing. The plan of virtualization is to consolidate administrative tasks while improving Cloud Computing and virtualization are synonymous. Cloud computing is based upon vitalizing and allocate compute, storage and net services in a shared multi-tenant environment. Virtualization is a key enabler for cloud compute. At the similar time, cloud computing is also a dominant force pulling virtualization into the enterprise. One method to improve the utilization of data centres resources, which has been shown to be efficient, is dynamic consolidation of Virtual Machines (VMs). The VMs are periodically reallocated using live migration according to their current resource demand in order to minimize the number of active physical servers, referred to as hosts, required to handle the workload. The idle hosts are switched to low-power modes with fast transition times to eliminate the static power and reduce the overall energy consumption. The hosts are reactivated when the resource demand increases. This approach has basically two objectives, namely minimization of energy consumption and maximization of the Quality of Service (QOS) delivered by the system, which form an energy-performance trade-off.

II THE OBJECTIVE OF A HOST OVERLOAD DETECTION ALGORITHM

We show that to improve the quality of VM consolidation, it is needed to make the most of the time interval between VM migrations from overloaded hosts. Since VM consolidation is applied to reduce the number of active physical hosts, the excellence of VM consolidation is inversely proportional to H, the mean number of active hosts over n time steps.

$$H = \frac{1}{n}\sum_{i=1}^{n} ai$$

Where, ai is the number of active hosts at the time step i = 1,2...n. A lower value of H represents a better excellence of VM consolidation.

To examine the impact of decision made by host overload discovery algorithms on the quality of VM consolidation, we consider an experiment, where at any time step the host overload detection algorithm can initiate a migration from a host due to an overload. There are two possible consequences of a decision to migrate a VM relevant to host overload detection: Case 1, when a VM to be migrated from an overloaded host cannot be placed on another active host due to insufficient resources, and therefore, a new host has to be activated to accommodate the VM; and Case 2, when a VM to be migrated can be placed on another active host. To study host overload detection in isolation, we assume that no hosts are switched off during the experiment, i.e., once a host is activated, it remains active until n. Let p be the probability of Case 1, i.e., an extra host has to be activated to migrate a VM from an overloaded host determined by the host overload detection algorithm. Then, the probability of Case 2 is (1-p). Let T be a random variable denoting the time between two subsequent VM migrations initiated by the host overload detection algorithm. The expected number of VM migrations initiated by the host overload detection algorithm over n period steps is n=E[T], E[T] is the expected inter-migration time.

Based on the definitions given above, we can define X ~ B(n=E[T]; p), a binomially distributed random variable denoting the number of extra hosts switched on due to VM migrations initiated by the host overload detection algorithm over n time steps. The expected number of extra hosts activated is E[X] = np=E[T]. Let A be a random variable denoting the time during which an extra host is active between the time steps 1 and n.The expected value of A can be defined as follows:

$$\begin{aligned} E[A] &= \sum_{i=1}^{\lceil \frac{n}{E[T]} \rceil} (n-(i-1)E[T])p \\ &= \left\lceil \frac{n}{E[t]} \right\rceil \frac{p}{2}\left(n+n-\left(\frac{n}{E[t]}1\right)E(T)\right) \\ &\le \frac{np}{2}\left(1+\frac{n}{E[T]}\right) \end{aligned}$$

Let us rewrite H as follows:

$$\begin{aligned} H &= \frac{1}{n}\sum_{i=1}^{n} ai \\ &= \frac{1}{n}\sum_{i=1}^{n} a1 + \frac{1}{n}\sum_{i=1}^{n}(ai-a1) \\ &= a1 + \frac{1}{n}\sum_{i=1}^{n}(ai-a1) \end{aligned}$$

The first term a1 is a constant denoting the number of hosts that have been initially active and remain active until the end of the experiment.

$$H* = \frac{1}{n}\sum_{i=1}^{n}(ai-a1)$$

Since the objective is to improve the quality of VM consolidation, it is necessary to minimize E[H_]. The only variable that can be directly controlled by a host overload detection algorithm is E[T]; therefore, to minimize E[H_] the objective of a host overload detection algorithm is to maximize E[T], i.e., to maximize the mean time between migrations from overloaded hosts.

III OPTIMAL OFFLINE ALGORITHM

It is necessary to maximize the mean time between VM migrations initiated by the host overload detection algorithm, which can be achieved by maximizing each individual inter-migration time interval. Therefore, we limit the problem formulation to a single VM migration, i.e., the time span of a problem instance is from the end of a previous VM migration and
to the end of the next.

Algorithm 1 An Optimal Offline Algorithm (OPT)
Input: A system state history
Input: M, the maximum allowed OTF
Output: A VM migration time
1: **while** history is not empty **do**
2: **if** OTF of history _ M **then**
3: **return** the time of the last history state
4: **else**
5: drop the last state from history
6: **end if**
7: **end while**

Proof: Let the time interval covered by the system state history be [t0; tn], and tm be the time returned by Algorithm 1. Then, according to the algorithm the system states corresponding to the time interval (tm; tn] do not satisfy the constraint. Since tm is the right bound of the interval [t0; tm], then tm is the maximum possible time that satisfies the constraint. Therefore, tm is the solution of the optimization problem and Algorithm 1 is an optimal offline algorithm for the problem of host overload detection.

IV A MARKOV CHAIN MODEL FOR THE HOST OVERLOAD DETECTION PROBLEM

Each VM allocated to a host at each point in time utilizes a part of the CPU capacity determined by the application workload. The CPU utilization created over a period of time by a set of VMs allocated to a host constitutes the host's workload. For the initial analysis, we assume that the workload is known a priori, stationary, and satisfies the Markov property. The CPU utilization of a host measured at discrete time steps can be described by a single time-homogeneous DTMC. There is a controller component, which monitors the CPU utilization of the host and according to a host overload detection algorithm decides when a VM should be migrated from the host to satisfy the QOS requirements, while maximizing the time between VM migrations. We limit the problem formulation to a single VM relocation, i.e., the time span of a trouble instance is from the end of a previous VM migration to the end of the next.

V MODELING ASSUMPTIONS

The introduced model allows the computation of the optimal control policy of a host overload detection controller for a given stationary workload and a given state configuration. It is essential to take into report that this result is based on a few fundamental modelling assumptions. First of all, it is assumed that the system satisfies the Markov property, or in other words, the sojourn times (i.e., the time a CTMC remains in a state) are exponentially distributed. Assume an exponential Distribution of sojourn times may not be exact in many systems. For instance, state transition delays can be deterministic due to a particular task scheduling, or follow other than exponential statistical distribution, such as a bell-shaped distribution. Another implication of the Markov property is the assumption of memory less state transitions, which means that the expectations state can be predicted solely based on the knowledge of the current state. It is possible to envision systems, in which future states depend on more than one past state. Another assumption is that the workload is stationary and known a priori, which does not hold in typical computing environments. We show how the introduced model can be heuristically adapted to handle unknown non-stationary workloads. The proposed heuristically adapted model removes the assumption of stationary and known workloads; however, the assumptions implied by the Markov property must still hold.

We evaluate the proposed heuristically adapted model and test the assumptions through a simulation study using real workload traces from more than a thousand PlanetLab VMs. The simulation results show that the model is efficient for this type of mixed computing workloads. With a correct understanding of the basic model assumptions and careful assessment of the applicability of our model to a particular system, an application of the model can bring substantial performance benefits to the resource management algorithms. As demonstrated by our simulation our approach outperforms the benchmark algorithms in terms of both the mean inter-migration time and the precision of meeting the specified QOS goal.

VI THE CONTROL ALGORITHM

The control algorithm based Optimal Markov Host Overload Detection (MHOD-OPT) algorithm. We refer to the MHOD-OPT algorithm adapted to unknown non stationary workloads using the Multi size Sliding Window workload estimation technique introduced the Markov Host Overload Detection (MHOD) algorithm. A high-level view of the MHOD-OPT algorithm. In the online setting, the algorithm is invoked periodically at each time step to make a VM migration decision.

Algorithm 2 The MHOD-OPT Algorithm
Input : Transition probabilities
Output: A decision on whether to move a VM
1: Build the idea and constraint functions
2: Call up the brute-force search to find the **m** vector
3: **if** a feasible solution exists **then**
4: Extract the VM migration probability
5: **if** the probability is < 1 **then**
6: **return false**
7: **end if**
8: **end if**
9: **return true**

Closed-form equations for L1(1); L2(1); : : : ; LN(1) are pre computed offline from (13); therefore, the runtime computation is not required. The values of transition probabilities are substituted into the equations for L1(1); L2(1); : : : ; LN(1), and the objective and constraint functions of the NLP problem are generated by the algorithm. To solve the NLP problem, we applied a brute-force search algorithm with a step of 0.1, as its performance was sufficient for the purposes of simulations. In MHOD-OPT, a decision to migrate a VM is made only if either no feasible solution can be found, or the migration probability corresponding to the current state is 1. The justification for this is the fact that if a feasible solution exists and the migration probability is less than 1, then for the current conditions there is no hard requirement for an immediate migration of a VM.

VII PROPOSED WORK

A technology that is able to improve the utilization of server resources, and thus, reduce power consumption, is virtualization of computing resources. Virtualization introduces an abstraction layer between an OS and hardware. Physical resources can be split into a number of logical slices called Virtual Machines (VMs). Each VM can accommodate an individual OS creating for the user a view of a dedicated physical resource and ensuring the performance and failure isolation between VMs sharing a single physical machine. The problems of host overload detection as a part of dynamic VM consolidation. Find when it is best to reorganize VMs from an overloaded host is an aspect of dynamic VM consolidation that directly influences the resource utilization and Quality of Service (QOS) delivered by the organization is shown Figure 2.

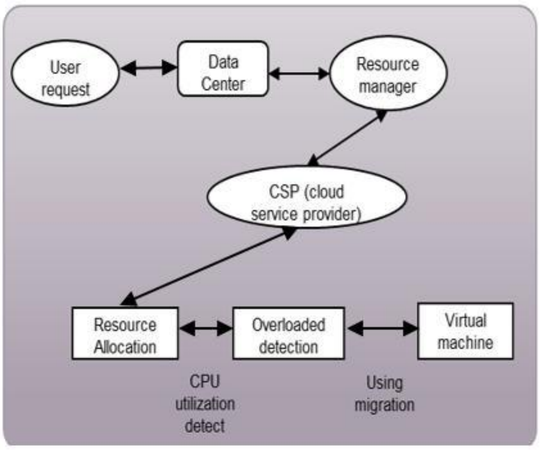

Fig 1: Virtualized servers with live movement capability

The authority on the QOS is explained by the fact that server overloads cause resource shortages and performance degradation of applications. In present solutions to the problem of host overload detection are generally heuristic-based, or rely on statistical analysis of past data. The limits of these approaches are that they lead to sub-optimal results and do not allow explicit specification of a QOS goal.

A novel approach that for any known stationary workload and a given state configuration optimally solves the problem of host overload detection by maximizing the mean inter-migration time under the specified QOS goal based on a Markov chain model. The algorithm is adapted to handle unfamiliar non-stationary workloads using the MultiSize Sliding Window workload estimation technique.

1. An analytical model showing that to improve the quality of VM consolidation, it is necessary to maximize the mean time between VM migrations initiated by the host overload detection algorithm.
2. An optimal offline algorithm for host overload detection, and proof of its optimality.
3. A novel Markov chain model that allows a derivation of a randomized control policy that optimally solves the problem of maximizing the mean time between VM migrations under an explicitly specified QOS goal for any known stationary workload and a given state configuration in the online setting.
4. A heuristically adapted algorithm for handling unknown non-stationary workloads using the Multisize Sliding Window workload estimation approach, which leads to comparable to the best benchmark algorithm performance in terms of the inter -migration time, while provide the advantage of precise specification of a QOS goal.

VIII CONCLUSION

Virtualization, in computing is the creation of a virtual i.e., rather than actual version of a storage device or network resources. By using some interfaces we can access the data in cloud. This paper gives about the cloud data management interface by using storage virtualization mechanism. The open cloud computing interface is an emerging standard for interoperable interface management in the cloud. To implement the MHOD algorithm as an extension of the VM manager within the Open Stack Cloud platform to evaluate the algorithm in a real system as a part of energy-efficient dynamic VM consolidation.

IX REFERENCES

[1] Anton Beloglazov and Rajakumar Buyya., “Managing Overloaded Hosts for Dynamic Consolidate of Virtual Machines in Cloud Data Centers Under Quality of Service Constraint”, July 2013.

[2]. N. Bobroff, A. Kochaut, and K. Beaty, “Dynamic placement of virtual machines for managing Service Level Agreement violation,” in 10th IFIP/IEEE Intl. Symp. on Integrated Network Management (IM), 2007, pp. 119–128.

[3]. D. Gmach, J. Rolia, L. Cherakasova, G. Belrosie, and A. Kemper, “An integrated approach to resource pool management: Policy,efficiency and quality metric,” in Proc. of IEEE Intl. Conf. on Dependable System and Network (DSN), 2008, pp. 326–335.

[4]. Christian Strack ”Performance and Power Management for Cloud Infrastructures” Philipps-

University Marburg, Germany -Department of Mathematics and CSE, Distributed System Group.

[5] J. Koomey,Growth in data center electricity use 2005 to 2010.Oakland, Analytics Press, 2011.

[6] Gartner, Inc.,Gartner estimates ICT industry accounts for 2 percent of global CO2 emissions. Gartner Press Release (April 2007).

[7] R. Nathuji and K. Schwan, "Virtualpower: Coordinated power management in virtualized enterprise system,"SIGOPS Operating System Review, vol. 41, pp. 265–278, 2007.

[8] A. Verma, P. Ahuja, and A. Nieogi, "pMapper: Power and migration cost aware application placement in virtualized system," in 9th Intl. Conf. on Middleware, 2008,pp. 243–264.