A Framework For Energy Aware Provisioning In Cloud Data Center

1Prof.N.R.RAJALAKSHMI and 2Dr.N.BALAJI

1Assistant Professor, Department of Computer Science and Engineering  SACS MAVMM Engineering College, Madurai-625301
2Professor & Head of Department, Department of Information Technology, K.L.N College of Engineering, Swarganga - 639611

ABSTRACT

Cloud computing data centers are the building blocks of IT organization to offer utility oriented services. These data centers consume huge amount of energy due to increasing utilities of cloud computing services, which leads to high operational costs and carbon footprints to the surroundings. Here, our work demonstrates the various energy aware provisioning models to lessen the energy cost and overhead of data centers. Among these proposed models, the hybrid energy aware provision model provisions the minimum figure of physical machines & releases ideal servers. This minimization of physical machines reduces overall energy consumption and overhead in the cloud computing environment.

INTRODUCTION

The cloud provides computing infrastructure that leverages system virtualization to allow multiple virtual machines (VM) to be consolidated on one physical machine (PM) where VMs often represent components of application environments. Clouds can make it possible to access applications and associated data from anywhere by pay-as-you-go model. Over the last few years, cloud computing facilities have become increasingly admired due to the growing data centers and parallel computing prototypes. The leading IT companies, such as Microsoft, Google, Amazon, Sales force and IBM, pioneered the cloud computing field and keep enhancing their services in data distribution and computational hosting.

Dzmitry Kliazovich (2010) discussed about the surveys of numerous governmental, academic and industrial; the surveys show that the energy consumed by computing and communication units within a data center gives to a considerable portion of the data center operational costs. Hence, the cost and operating expenses of data centers are overwhelming with the increase in computing capacity. Apart from the overwhelming operating costs due to high energy consumption, another rising concern is the environmental impact in terms of carbon dioxide (CO2) emissions caused by this high energy consumption.

Power and energy consumption of a large data center including the power consumption of processors, memories, storage and networking resources, and also other costly infrastructures, such as AC facilities. Actually, cloud data centers are electricity guzzlers especially if resources are switched on permanently even if they are not used. Naone (2009) said that, an unused server utilizes about 70% of its peak power. This waste of unused power is considered as a major root of energy inefficiency. Intel's cloud computing (2015) vision point outs to make better power efficiency in data centers by shutting down and putting to sleep idle servers. As said by the Gartner Report, “Between 2000 and 2006, the amount of energy consumed by data centers around the world has doubled and nowadays, the average data center consumes as much energy as 25,000 households”. According to a McKinsey report by Kaplan (2008), “Data centers in 2010 are expected to consume as much energy as 10 new major power plants”.

This ever increasing energy consumption of computing systems has begun to limit further performance growth due to vast energy consumption and carbon dioxide footprints by Akshat Dhingra (2013). Thus, the objective of the data center has been moved from performance improvements to power and energy efficiency. Here, we work to raise the power and energy efficiency by the various ways of provisioning PM to have the minimum number of servers into on mode.

Energy Aware Provision Algorithms:

The key objectives for energy aware provision algorithms proposed in our framework are...
minimizing energy and overhead. Here, we assume that, the waste power as booting additional PM plus idle power (no workload) of PMs in the data center.

We consider, the data center consists of a finite set of M homogeneous servers or physical machines with same maximum power consumption limit \( P_{j, \text{Max}} \). Each VM, has a maximum power consumption \( p_i \). Each server or hosting physical machine \( j \), from the data center, has a power consumption limit or power cap noted as \( P_{j, \text{Max}} \) that cannot be exceeded when serving or hosting the VMs. The following section describes the energy aware provision algorithms considered in this paper.

**Physical Machine Parallel Algorithm (Pmp):**

It will determine the number of physical machines \( N \) needed by the set of VMs request \( R \), as well as the number of currently free server or ideal server \( F \) (without any VMs) in the data center. The number of required PMs returned by the physical machine parallel algorithm will be equal to \( |R| - F \) or 0 if \( |R| \leq F \). All these required PMs will be booted in parallel. If there is no request in the queue of the data center, then all the ideal PMs will be released. It is expected that, the physical machine parallel algorithm may work well with large VM (in terms of required power as \( P_{j, \text{Max}} \)). It might not be the ideal solution for small VM arriving in the queue. Because, the requested small VM’s power might be less than the power of on demand PMs to ready for work.

**Physical Machine-Sequential (Pms):**

The PMS algorithm is similar to the PMP algorithm, except that if \( |R| > F \) then, it will decide to boot a single PM; otherwise no PM will be requested. If \( |R| = 0 \) then, it will release the free PM, if \( F > 0 \). This algorithm avoids the overhead of energy loss by booting the single PM; however, it may be slow to react to bulky workloads. The key advantage of the PMP and PMS algorithm is that, it does not require detailed information of power.

### Algorithm

```
Initialize P ← 0, S ← 0, F ← 0, N ← 0;
U=\sum P_{j, \text{current}} / P_{j, \text{Max}};
for each request VMs power \( p_i \) do
  \( P = P + p_i \);
  if 0 ≤ P ≤ \( \sum P_{j, \text{Max}} \sum P_{j, \text{current}} \) then
    S ← 0;
  else
    S = P / P_{j, \text{Max}}
end
for each server \( M \in \text{datacenter} \) do
  if \( M \) has VM then
    V = V + 1;
end
end
foreach server \( M \in \text{datacenter} \) do
```

**Physical Machine -Power-Parallel (Pp):**

The PP algorithm uses the power consumption of queued VMs to determine the number of servers or physical machines needed. The algorithm calculates the required servers by dividing the total power \( P \) of queued VMs by the maximum power consumption of the server \( P_{j, \text{Max}} \) in the cloud. All these required PMs will be booted in parallel. If \( 0 < p_i < P_{j, \text{Max}} \), a single server will be requested. If there is no request of VM in the queue, the algorithm releases all free PMs in the data center.

**Physical Machine-Power-Sequential (Ps):**

The PS algorithm is similar to the PP algorithm, except that it will boot only one server if \( P > P_{j, \text{Max}} \). If there is no request of VM in the queue, it will release free PM.

**Hybrid Energy Aware Provision Algorithm (Hpp):**

This algorithm combines the above algorithms. It minimizes the energy consumption of data centers. Such as, all the requested VM instances (e.g., small, medium, large) could be placed in minimum number of servers; then, the unused or ideal servers are shut down or put into sleep mode.

All the variables used in this model are listed below.

- \( N \) is the size of the servers to be switched on for the \( R \) number of VMs request.
- \( M \) is the total quantity of servers in the data center.
- \( V \) denotes servers which have VM.
- \( U \) is the number of utilized servers.
- \( p_i \) denotes the power consumption of request VM.
- \( P_{j, \text{Max}} \) represents the maximum power consumption of server \( j \).
- \( P_{j, \text{current}} \) represents the power consumption of server \( j \) \( (P_{j, \text{current}}=P_{j, \text{idle}}+\sum p_k \) with VMs \( k \) hosted by the server \( j \)).
- \( P_{j, \text{idle}} \) represents the power consumption of server \( j \) when it is idle.
The above algorithm determines the number of new servers or physical machines to be booted parallel & the number of idle servers is to be shut down to save the energy. The algorithm will need to carefully evaluate the information obtained from resource manager to make a decision. Such as, the currently utilized power of the server, the power of request VMs, the number of running server & the idle server in the data center to have a decision.

The pseudo-code of the algorithm is explained below. It has the following steps:

- **Step 1**: Get the currently utilized power of all the physical machines and the maximum power consumption of server to calculate the number of currently utilized servers.
- **Step 2**: Calculate the total power consumption of request VMs.
- **Step 3**: Note that, if \(0 \leq P \leq \sum_{j} \max(p_i) \sum_{j} p_i \text{ current}\) then required server is zero. Else, calculate the total physical machines required for requesting VMs.
- **Step 4**: Get the numbers of free Server F which have no workloads & number of servers R which have VMs in the data center.
- **Step 5**: Calculate the number of servers N to be booted for incoming request by using \(S-(V+F-U)\); then, it releases all free servers.

**Experiment:**

The evaluation scenario corresponds to data center with 100 servers or nodes for the experiments. All servers have a power consumption cap \(p_i\) max, set to 250 watts (the peak power of a typical server is around 250 watts [8]). To perform per-VM power estimation, we refer to a power estimation model proposed in [9, 10]. We assume energy consumption \(p_i\) associate to each VM type (small, medium and large) equal to 10 watts (low), 20 watts (medium) and 30 watts (high) respectively.

Users submit request for provisioning of 10, 50, 100, 150, 200..., heterogeneous VMs (small, medium and large) that have an energy consumption \(p_i\) respectively equal to 10 watts (low), 20 watts (medium) and 30 watts (high). Here, we have changed VMs request at all iterations. Figure 1 shows the numbers of server are able to be allotted by the proposed energy aware provisioning methods for queued VMs. We note that, the parallel algorithm PMP, PP, HPP perform worse in terms of power than their sequential equivalents; such as PMS and PS method boot the minimum physical machines in the data center during simulations. However, PMS and PS method may be slow to react to the bulky request. But, HPP places all the VM instances (e.g., small, medium, large) into a minimum number of servers and releases all unused or ideal servers.

**Conclusion:**

We experiment the various energy aware provisions of physical machine to serve the incoming request of virtual machines in the cloud data center. Our HPP algorithm rectified energy overhead by reducing provision of more physical machines. Then, it also released the unused or ideal servers to save the energy. We concluded that, the best decision might be utilizing the currently running physical machine’s capacity and the idle machines than initiating more servers in the cloud.

```
if M has no VM then
    F = F + 1;
end
N← (S-(V+F-U));
Return N, F
```

**Fig. 1:** The number of servers is to be allotted by the proposed energy aware provisioning methods for queued VMs.
REFERENCE


Gartner Group, available at: http://www.gartner.com


