Energy-Efficient Virtual Machine Replication and Placement in a Cloud Computing System

BY: HADI GOUDARZI AND MASSOUD PEDRAM

UNIVERSITY OF SOUTHERN CALIFORNIA

DEPARTMENT OF ELECTRICAL ENGINEERING

Abstract:

Using Virtual Machines and server consolidation can reduce the total energy consumption for servicing clients with very little performance degradation.

The cloud provider can take advantage of dissimilar workloads by assigning these workloads to the same server and can utilize fewer resources to service clients.

Placing multiple copies of a VM on different servers and distributing the incoming requests among these VM copies can reduce the resource requirement for each VM copy and help the cloud provider utilize the servers more efficiently

Abstract:

In this paper, the problem of energy-efficient VM placement in a cloud computing system is solved.

This paper presents an approach that first creates multiple copies of VMs, then uses dynamic programming and local search to places these copies on the physical servers

Introduction:

Server consolidation provides a new way to improve the power efficiency of data centers.

Server consolidation enables the assignment of multiple VMs to a single physical server

Consolidation allows some of the available servers to be turned off or put into a sleep state, thereby lowering power consumption of the computing system(the technique works because modern servers tend to consume 50% or so of their peak power in idle state).

Introduction:

Consolidation involves a performance-power tradeoff.

If workloads are consolidated on servers , performance of the consolidated VMs may decrease because of reduction of available resources, but overall power efficiency is improved because fewer servers are needed to service VMs.

Low utilization of servers in a datacenter is one of the most important factors responsible for low power efficiency of datacenters

Due to the non-energy-proportional nature of the current servers, it is prudent from an energy efficiency viewpoint to have as few servers as possible turned on, with each ON server being highly utilized.

Introduction: SLAs

The IT infrastructure provided by the datacenter owners/operators must meet various Service Level Agreements (SLAs) established with the clients. The SLAs may be:

Resource related (e.g., amount of computing power, memory/storage space, network bandwidth)

Performance related (e.g., service time or throughput)

Quality of service related (24-7 availability, data security, percentage of dropped requests.)

To minimize the energy consumption using consolidation, these SLA constraints should be considered.

Introduction

A datacenter comprises of thousands to tens of thousands server machines working together to provide services to the clients.

In such a large computing system, despite non-energy-proportional characteristics of current server machines, energy efficiency can be maximized through system wide allocation and server consolidation.

The problem of resource provisioning is challenging due to the diversity in the hosted client applications.

For example, some applications may be computation-intensive while others may be memory intensive. Some applications may run well together while others do not.

Introduction

The energy cost and admission control policy in a cloud computing system are affected by its power and VM management policies

Power management techniques control the average and peak power in the data centers.

VM management techniques control the VM placement on the physical servers as well as VM migration from one server to the next

This paper focuses on the VM placement to minimize the energy cost in the cloud computing system.

Introduction: VM placement

Generating multiple copies of a VM and placing them on different servers is a basic way to increase service reliability. In this approach, only the original copy of the VM handles the requests while the other copies are idle

This paper proposes to exploit all the VM copies for servicing requests.

Resources provided to each VM copy should satisfy SLA requirements

The set of distributed VMs should be able to service all the incoming requests.

Therefore memory Bandwidth provided for each VM copy should be the same as the original VM

Introduction: VM placement

Total CPU cycles provided for all the VM copies should be equal to those provided to the original copy of the VM.

Increasing the number of VM copies increases the average utilization level of servers because by increasing the number of VM copies, we have more opportunity to use smaller VMs to fully utilize the servers, and thereby, avoid having under-utilized servers.

Using this approach and an effective VM placement algorithm, the energy cost of the system can be reduced by up to 20%

Introduction: VM placement

The paper's proposed VM placement algorithm is based on the dynamic programming and local search methods.

The dynamic programming method determines the number of copies for each VM and places them on servers

The local search tries to minimize the energy cost by turning off the under-utilized servers.

System model: table of notations and definitions.

Table I. NOTATION AND DEFINITIONS

Symbol name	Definition
c_i^m, c_i^p	Required memory BW and total processing capacity for the i[th] client
L_i	Max. # of servers allowed to serve the i[th] client
s_k	Set of servers of type k
C_j^p, C_j^m	Total CPU cycle and memory BW of the j[th] server, shorthand notation for $C_{S_k}^p$ and $C_{S_k}^m$
P_j^0	Constant power consumption of the j[th] server operation in the active mode. Same as $P_{S_k}^0$
P_j^p	Power of operating the j[th] server which is proportional to the utilization of processing resources, shorthand notation for $P_{S_k}^p$
T_e	Duration of a *decision epoch* in seconds
x_j	A pseudo-Boolean integer to determine if the j[th] server is ON (1) or OFF (0)
y_{ij}	A pseudo-Boolean integer to determine if the i[th] VM is assigned to the j[th] server (1) or not (0)
ϕ_{ij}^p, ϕ_{ij}^m	Portion of the processing and memory BW resources of the j[th] server that is allocated to the i[th] client
ϕ_j^p, ϕ_j^m	Portion of the processing and memory BW resources of the j[th] server that is allocated to any client

System model: Cloud computing system

A datacenter comprises of several potentially heterogeneous servers chosen from a set of known and well characterized server types.

Servers of a given type are modeled by their processing capacity or CPU cycles and memory BW as well as their energy cost, which is directly related to their average power consumption

The energy cost is calculated as the server power consumption multiplied by the duration of the epoch in seconds. The power of a server is modeled as a constant power cost plus another variable power cost, which is linearly related to the utilization of the server

System model: Client and Virtual Machines

Clients in the cloud computing system are represented with VMs.

Each VM may be copied onto different servers (i.e., requests generated by a single client can be assigned to more than one server).

If multiple copies of a VM are placed on different servers, these constraints must be satisfied:

Constraint (1) enforces the summation of the reserved CPU cycles on the assigned servers to be equal to the required CPU cycles for client i.

Constraint (2) enforces the provided memory BW on assigned servers to be equal to the required memory BW for the original VM

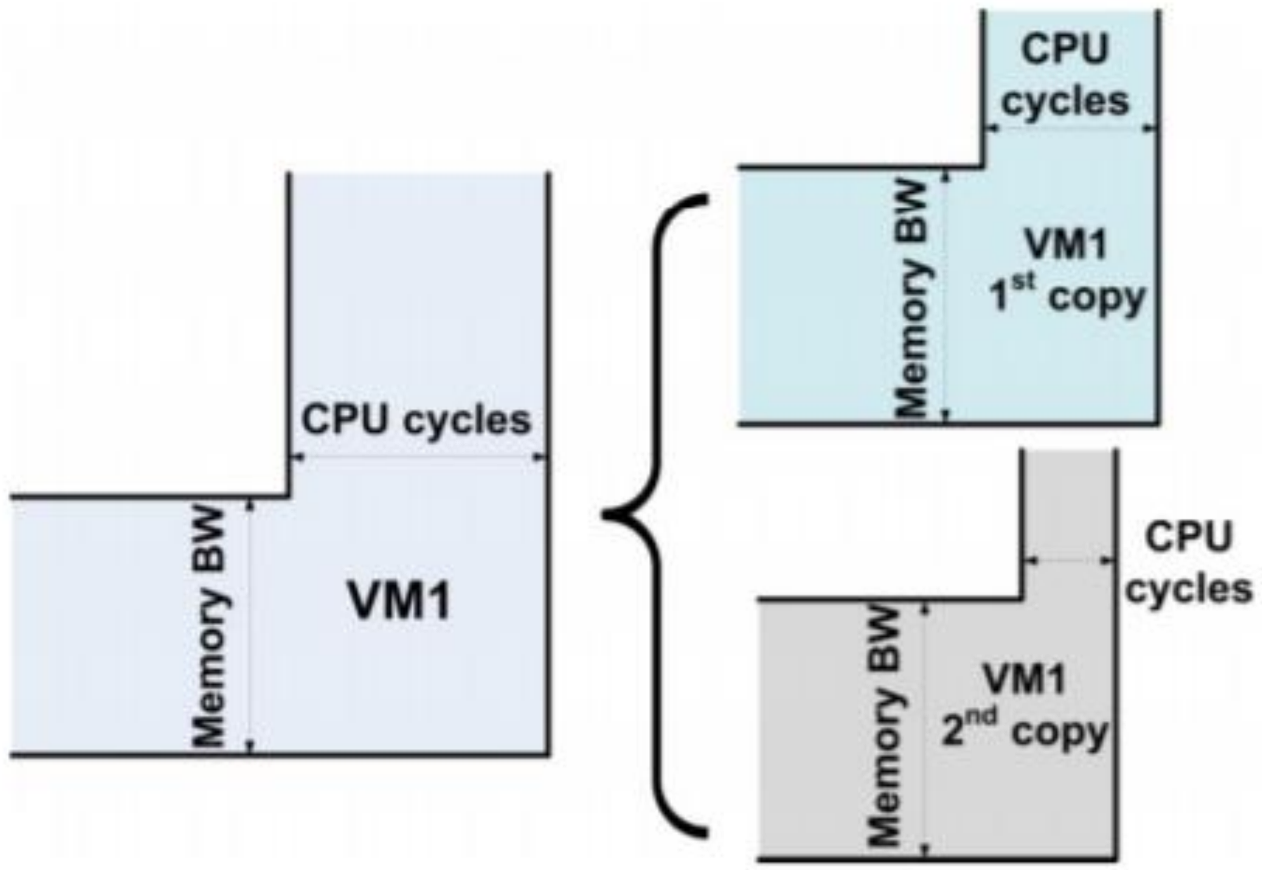

Figure 1. An example of multiple copies of a VM

Example of multiple copies of a VM:
The difference between heights of the horizontal bars shows the memory BW requirement while the difference between widths of the vertical lines show the CPU cycle requirement of each VM

System Model: VM management system

Datacenter management is responsible for admitting the VMs into the datacenter, servicing them to satisfy SLA requirements, and minimizing the energy cost of the datacenter.

This paper focuses on the VM controller.

The VMC is responsible for determining the resource requirements of the VMs and placing them on servers

The VMC should also perform VM migration to mimic workload changes

The VMC performs these tasks based on two different optimization procedures: semi-static optimization and dynamic optimization

System Model: VM management system

Semi-static optimization procedure is performed periodically

In this procedure the VMC considers:

The active set of VMs, previous assignment solution, feedbacks generated from power, thermal and performance sensors, and workload prediction to generate the best VM placement solution for the next epoch.

This paper focuses on the semi-static procedure of the VMC

The role of the semi-static optimization procedure in the VMC is to determine whether to create multiple copies of VMs on different servers and assign VMs to servers.

System Model: VM management system

Considering fixed payments by the clients for the cloud services they receive, the goal of this optimization is to minimize the energy cost of the active servers in datacenter.

An exemplary solution for assigning six VMs with different resource requirements on two heterogeneous servers is depicted in Figure 2

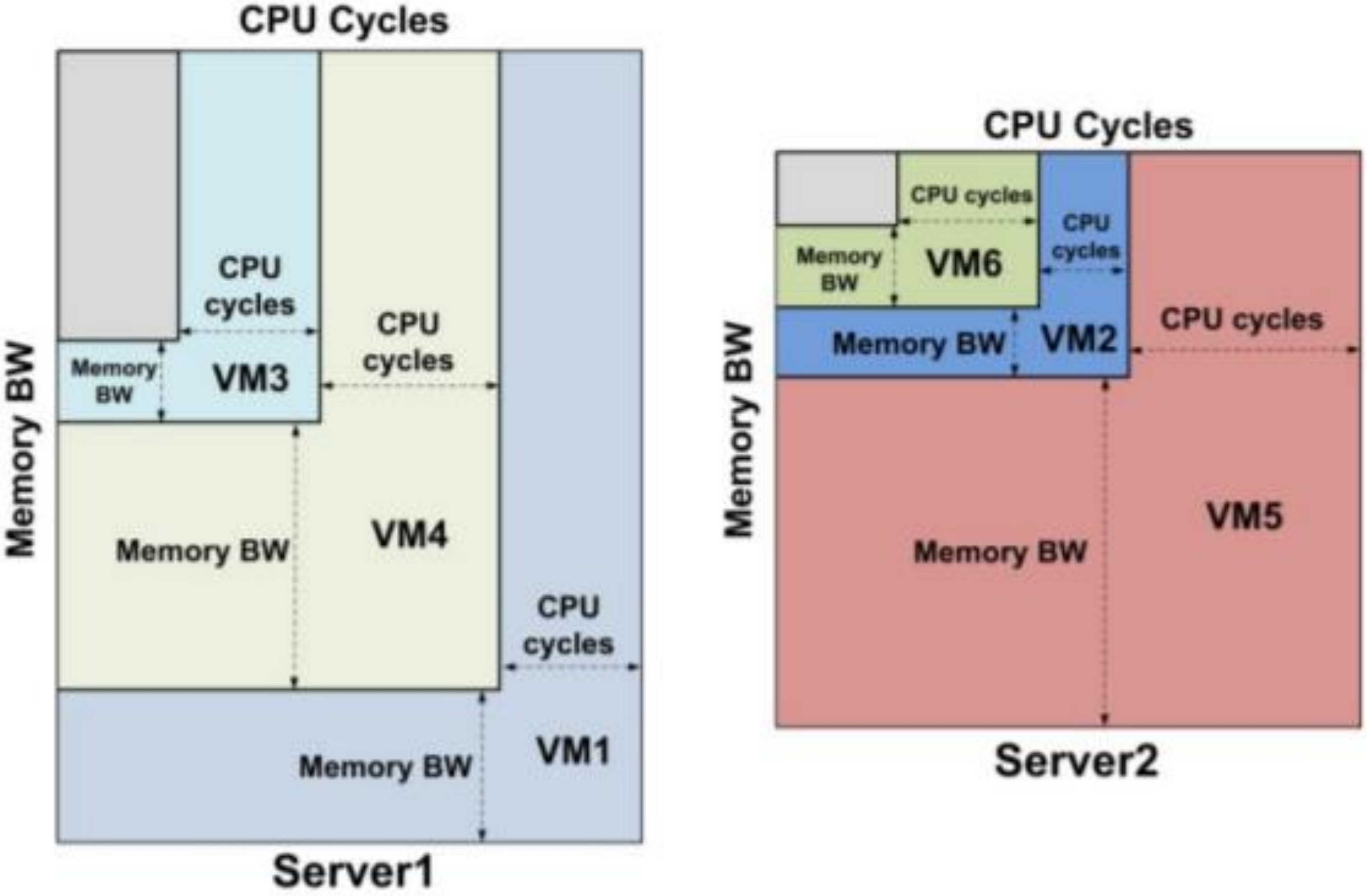

Figure 2. An exemplary solution for assigning six heterogeneous VMs on two heterogeneous Servers

Figure 2

Problem formulation

In this paper, a VM placement problem is considered with the objective of minimizing the total energy consumption in a decision epoch while servicing all VMs in the cloud computing system

The formulation of this problem is called MERA for Multi-dimensional Energy-efficient Resource Allocation

$$Min \quad T_e \sum_j x_j \left(P_j^0 + P_j^p \sum_i \phi_{ij}^p \right) \quad (3)$$

subject to:

$\phi_j^p = \sum_i \phi_{ij}^p \leq 1$	$\forall j$	(4)
$\phi_j^m = \sum_i \phi_{ij}^m \leq 1$	$\forall j$	(5)
$\sum_j C_j^p \phi_{ij}^p = c_i^p$	$\forall i,j$	(6)
$y_{ij} \geq \phi_{ij}^p, \quad y_{ij} \leq 1 + \phi_{ij}^p - \varepsilon$	$\forall i,j$	(7)
$\phi_{ij}^m y_{ij} C_j^m = c_i^m$	$\forall i,j$	(8)
$\sum_i y_{ij} \leq L_i$	$\forall i$	(9)
$x_j \geq \sum_i \phi_{ij}^p$	$\forall j$	(10)
$y_{ij} \in \{0,1\}, x_j \in \{0,1\}, \phi_{ij}^p \geq 0, \phi_{ij}^m \geq 0$	$\forall i,j$	(11)

where ε is a very small positive value, and, x_j is a pseudo-Boolean integer variable to determine if the j^{th} server is ON (x_j=1) or OFF (x_j=0).

Exact formulation of MERA

Problem Formulation: Theorems

This paper proposes and proves two theorems

Theorem I: Generalized Assignment Problem (GAP) can be reduced to the MERA problem

Theorem II: Bin Packing Problem (BPP) can be reduced to MERA problem

Problem formulation

Theorem I and II show that MERA is a combination of two NP-hard problems.

MERA is an NP-hard problem

This paper considers a case in which the required resources for VMs are smaller than the available resources in the datacenter.

This means we consider energy minimization with a fixed set of VMs instead of maximizing the number (or the total profit) of VMs that are served in the datacenter

So, we assume that a simple greedy algorithm will find a feasible solution to MERA for the specified inputs in the problem definitions

Problem formulation

In this paper, the authors examine the effect of multiple active copies of VMs

An effective algorithm to reduce the energy consumption in the cloud computing system is proposed.

Proposed Algorithm

In this section, a heuristic for solving the MERA problem is presented.

An algorithm based on dynamic programming is presented to determine the number of copies of each VM and assign these VMs to the servers

The goal of this algorithm is to minimize the total energy cost of the active servers.

To improve the results, a local search is considered to minimize the number of active servers as much as possible.

Algorithm 1: Energy-efficient VM Placement

Algorithm 1: Energy Efficient VM Placement

Inputs: C_j^m, C_j^p, P_j^0, P_j^p, c_i^m, c_i^p, L_i

Outputs: ϕ_{ij}^p, ϕ_{ij}^m (i is constant in this algorithm)

P={}
For (k = 1 to number of server types)
    ON=0; OFF=0;
    For ($\alpha = 1$ to L_i)
        $\phi_{ij}^p = (\alpha c_i^p / L_i) / C_j^p$
        $c_{ik}(\alpha) = \phi_{ij}^p P_j^p + P_j^0 c_i^m / C_j^m$
    End
    $J^{ON} = \{j \in s_k | (1 - \phi_j^m) \geq c_i^m / C_j^m\}$
    $J^{OFF} = \{j \in s_k | \phi_j^p = 0, (1 - \phi_j^m) \geq c_i^m / C_j^m\}$
    Foreach ($j \in s_k$)
        If ($j \in J^{ON}$ & ON< L_i)
            $P = P \cup \{j\}$, ON++, $cost_{ij}(\alpha) = c_{ik}(\alpha) - \varepsilon$
        Else if ($j \in J^{OFF}$ & OFF< L_i)
            $P = P \cup \{j\}$, OFF++, $cost_{ij}(\alpha) = c_{ik}(\alpha)$
    End
End
X = L_i, and Y = size (P)
Foreach ($j \in P$)
    For (x = 1 to X)
        D[x,y]= infinity; //Auxiliary X×Y matrix used for DP
        For (z = 1 to x)
            D[x,y]=min(D[x,y],D[x-1,y-z]+ $cost_{ij}$(z))
        D[x,y]=min(D[x,y], D[x-1,y])
    End
End
Back-track to find the best assignment and update ϕ_j's

Proposed algorithm

In the beginning of the VM placement, clients are ordered based on their processing requirement.

Based on this ordering. The optimal numbers of copies of the VMs are determined and these copies are placed on servers using dynamic programming.

Proposed Algorithm: Local Search method

In the local search method, servers are turned off based on the utilization and VMs are placed on the rest of the servers(if possible) to minimize the energy consumption.

To minimize the total energy consumption in the system, all servers with utilization less than a threshold are examined

To examine these under-utilized servers, each of them is turned off one by one and total energy consumption is found by placing their VMs on other active servers using the proposed DP placement method

Proposed Algorithm: Local Search method

If the total cost of the new placement is less than the previous total cost, the new configuration is selected, and the remaining under-utilized servers are examined; otherwise the option of turning off that server is rejected, and other candidate servers are examined.

Simulation Results: Heuristics for comparison

The authors implemented the min Power Parity (mPP) heuristic as one of the state of the art energy-aware VM placement techniques.

This heuristic is based on first fit decreasing heuristic for the bin-packing problem.

The heuristic tries to minimize the overall power consumed by the active servers in datacenter servicing the VMs.

To show the effectiveness of their proposed approach for placing multiple copies of VMs on servers, along with mPP, a version of their algorithm in which the solution is prohibited from using more than a single VM per client is considered(baseline)

Simulation Results

Moreover, to show the effect of distributed resource assignment and constant power cost for active servers, the authors implement a procedure to find the lower bound on the total energy cost with relaxation of these obstacles

Numerical Results: Normalized total energy cost in the system using the EVRP algorithm, baseline method, and mPP algorithm

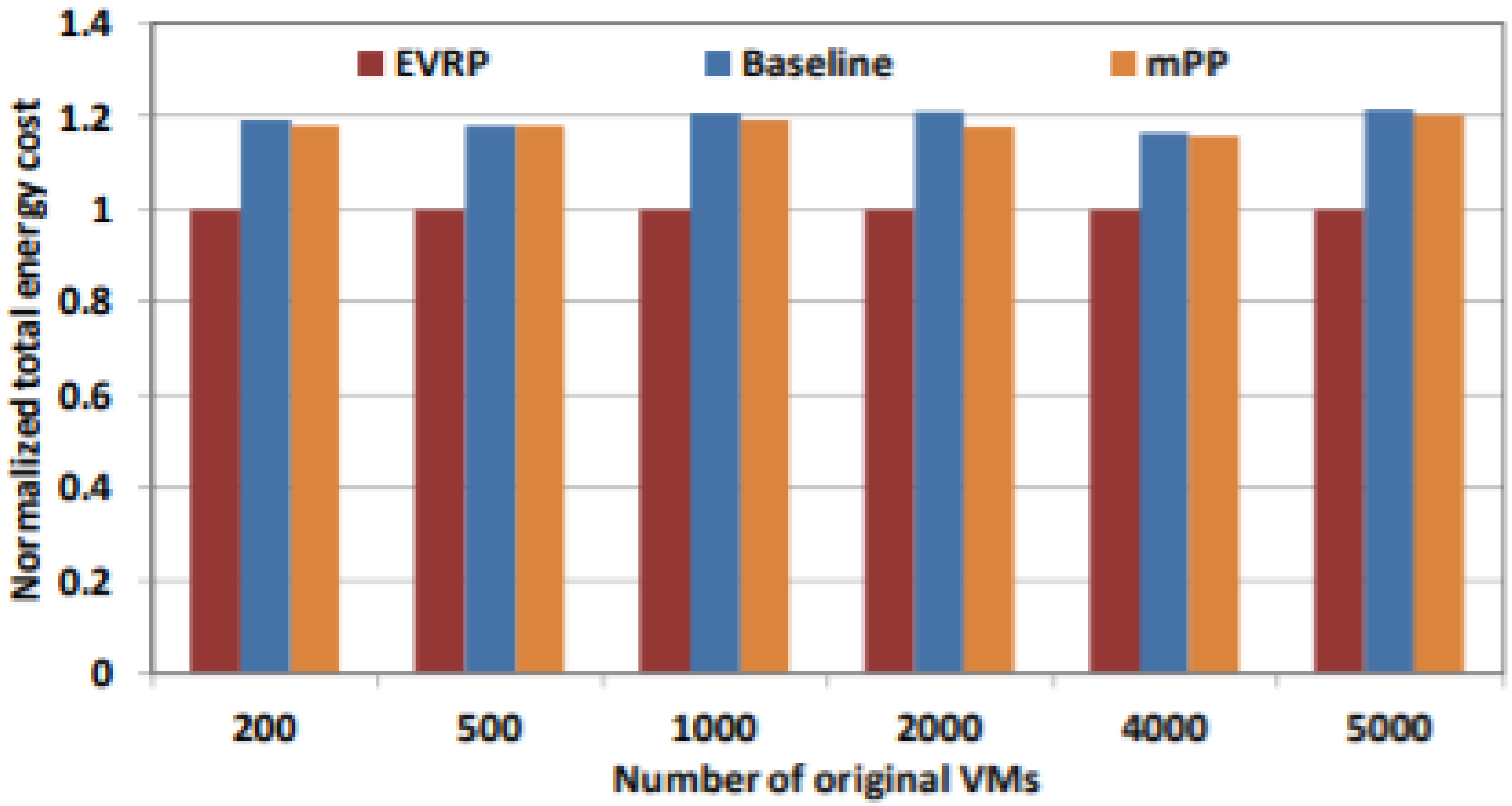

Figure 3. Normalized total energy cost of the system

Figure 3

EVRP reduces the total energy cost of VM placement solution by 16 to 20% with respect to mPP algorithm.

Performance of the baseline algorithm which is based on assigning the VMs using DP method is 1 to 4% worse than mPP method because baseline method does not place the VM on the server with least resource availability and instead choose the host server randomly in a selected server type

TABLE II. PERFORMANCE OF THE PROPOSED SOLUTION W.R.T. LOWER BOUND COST AND AVERAGE NUMBER OF VM COPIES

# of original VMs	Performance w.r.t Lower bound	average # of VM copies
200	1.13	1.33
500	1.14	1.32
1000	1.10	1.29
2000	1.14	1.31
4000	1.16	1.30
5000	1.10	1.35

Table II

Table II

Table II shows the relative performance of EVRP with respect to the derived lower bound on the total energy cost

There are two reasons behind the difference between the result of EVRP and the lower bound:

i) imperfection of the algorithm, and

ii) constant power consumption of the servers (independent from their utilization) and effect of the distributed resources in the datacenter.

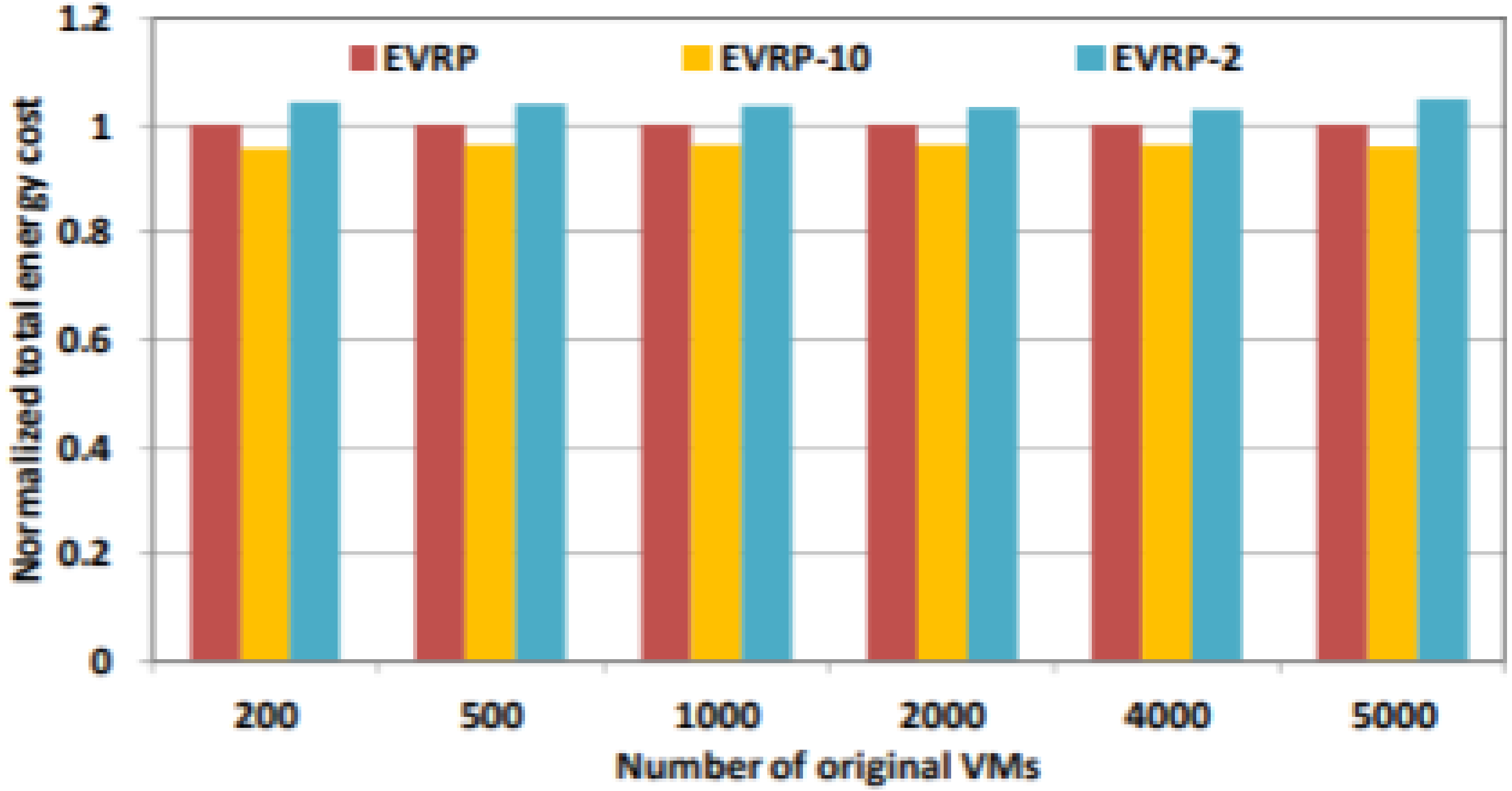

Figure 4. Normalized total energy cost of the VM placement solution using for different L_i

Figure 4

Figure 4

This figure shows the effect of different Values of Li(VMs per client)

In this figure the normalized total energy cost of the VM placement solution by using EVRP for different Li values are shown.

As can be seen, the difference between EVRP and a version of EVRP that restricts the number of VM copies to 2 is 4% (average).

This shows that the idea of using multiple copies of VM is effective even if the number of these copies is limited to 2 for big VMs.

This difference for a version of EVRP that considers at most 10 copies of VM for a VM with the biggest CPU cycle requirement is 3% (average).

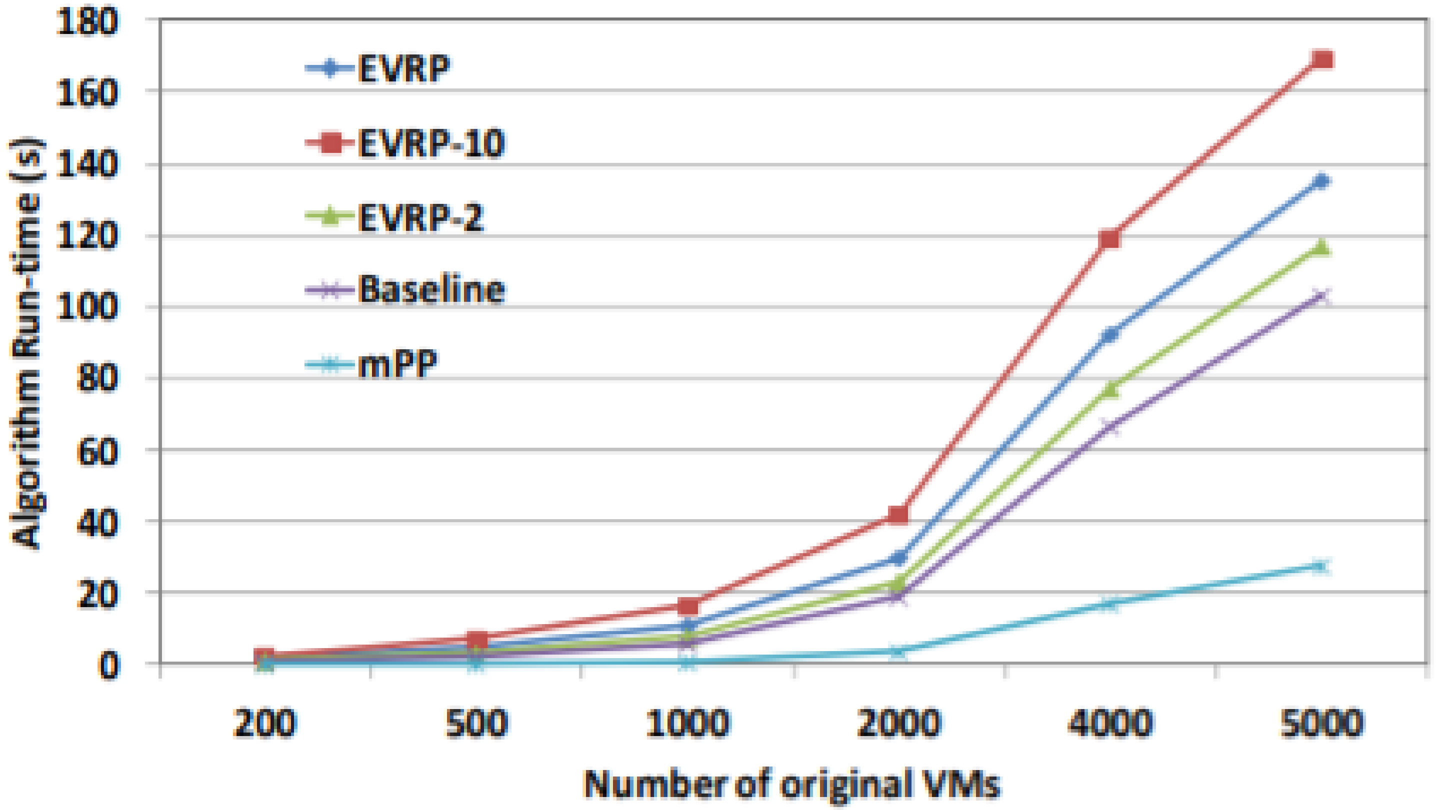

Figure 5. Run-time of EVRP for different number of VMs on 2.4GHZ E6600 server with 3GB of RAM from Intel

Figure 5 shows the average run-time of the EVRP, baseline, and mPP methods for different number of VMs

Figure 6

Figure 6 shows the average utilization of the servers for different Pp/P0 and for different VM placement methods. As can be seen, the utilization level increases when Pp/P0 decreases. Smaller value for Pp/P0 means that the server is less energy-proportional.

P0 is constant power consumption of a server operation in the active mode

Pp is Power of operating a server, which is proportional to the utilization of processing resources

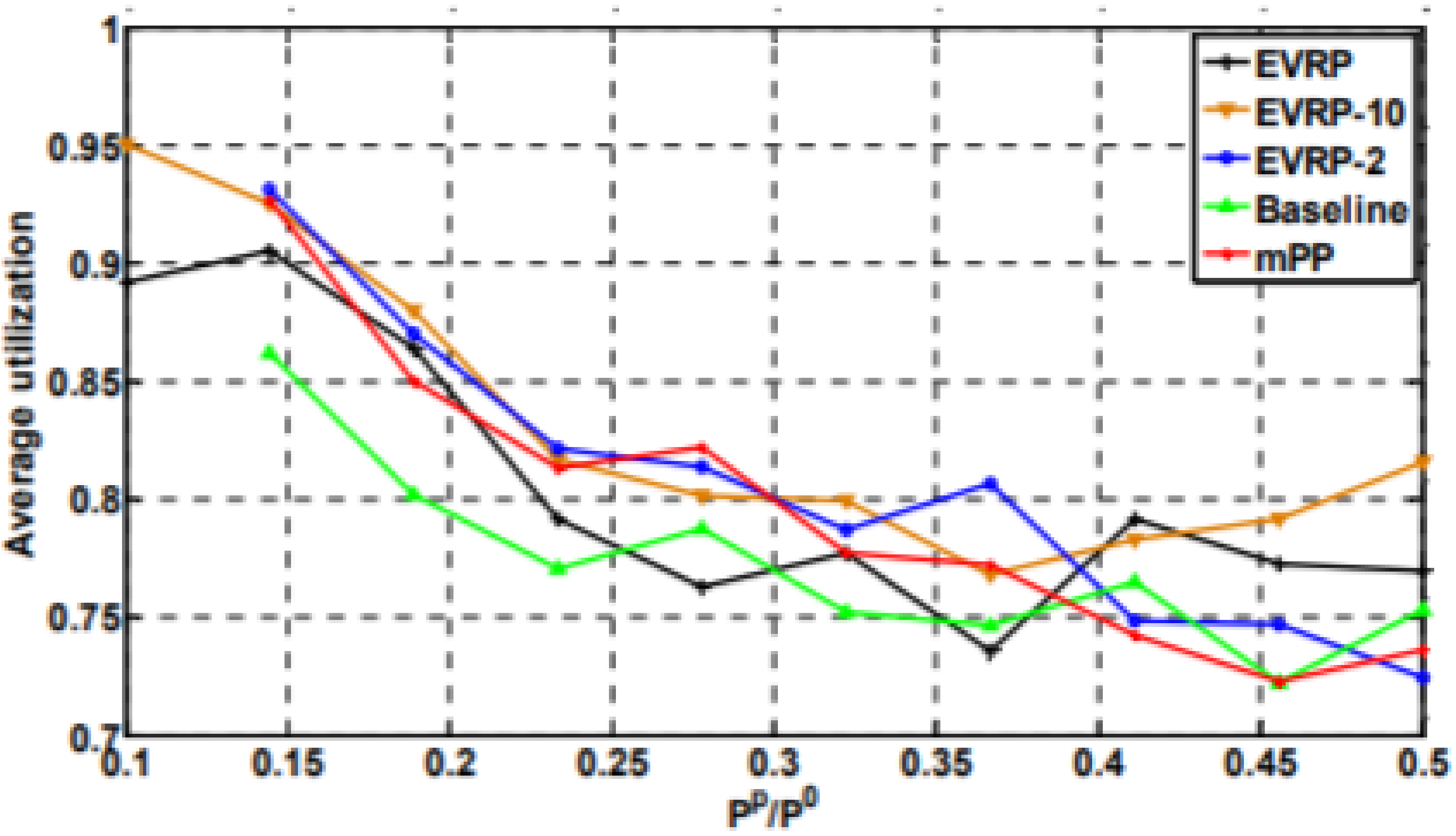

Figure 6. Ratio of expected percentage of the response time constraint's violation to the maximum allowed percentage of violation

Figure 6

Conclusion

The authors presented an approach to generate multiple copies of VMs without sacrificing QoS

An algorithm based on dynamic programming and local search was provided to determine the number of VM copies, and then place them on the servers to minimize the total energy cost in the cloud computing system.

This approach reduces the energy cost by up to 20% with respect to prior VM placement techniques.