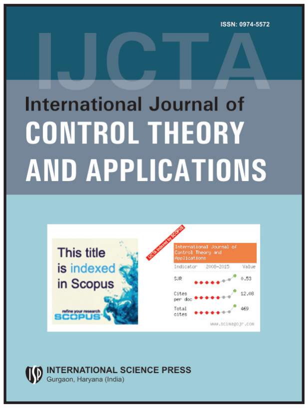

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 9 • Number 43 • 2016

A Comparison between PSO and GA Approach for Energy and Cost Effective Saas Placement on Cloud

Sony K.T.[a], Linda Sara Mathew[b] and Ani Sunny[c]

[a]*Department of Computer Science and Engineering, M.A College of Engineering, Kothamangalam, Kerala, India. Email: sonykt1991@gmail.com*
[b-c]*Asst. Prof. Department of Computer Science and Engineering, M.A College of Engineering, Kothamangalam, Kerala, India. Email:* [b]*lindasaramathew@gmail.com;* [c]*anisunny88@gmail.com*

Abstract: Software as a Service (SaaS) is a way of delivering applications over the Internet as a service. *SaaS* applications are sometimes called Web-based software, on-demand software, or hosted software. It is the very important services in cloud computing. The challenges in Saas placement problem depends resource requirements, cloud network size and communication among its components. This paper is a comparative study of two saas placement algorithms, the aim is to finds energy efficient cost effective solution for Saas Placement Problem (SPP). In this work, it find optimal Saas placement in Cloud based on service level agreement (SLA). Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) that have been applied to find the optimal placement of Saas component and find which algorithm approach minimize the total cost incurred to the Saas provider and minimize the total energy. Virtual machine placement approaches to the virtual machine placement problem consider the energy consumption by physical machines in a data center. Experimental results show that energy efficient Saas Placement using PSO generates better solutions than Saas Placement using GA.

Keywords: SaaS, Saas Placement Problem, Particle Swarm Optimization, Genetic Algorithm, Virtual machine placement.

1. INTRODUCTION

Cloud computing is an emerging technology provides internet based platform which are used for computer technology. And also provides various storage and computing services over the Internet. Based on the service that the cloud is offering [1] mainly classified into three types: (1) Infrastructure_as_a_Service (IaaS) providing computing infrastructure to the computing and storage problems of the user, (2) Platform_as_a_Service (PaaS) is a tool and service designed to develop, design and deploy test activities in Cloud platforms and (3) Software_as_a_Service (SaaS)[2]. Cloud service providers purchase data-center software and hardware to deliver storage and computing services through the Internet. Saas is service offered by clod computing, it is a way to deliver the application over the internet, it does not require client software installation and maintenance just a browser or other client device and network connectivity. In Saas service placing the software components is a problem, it is called saas placement problem. When software as services component placed on the cloud, performance of the software as services should be optimal based on the cost, and it should satisfy all the resource constraints without violating service level agreement. In saas service placing the virtual machine in physical server is also a problem. When placing virtual machine, energy efficiency of the virtual machine should be optimal based on the energy consumption. In each year user's demand for SaaS is increases, a report is presented by Dubey and Wagle [3]. It is billed based on usage of each consumer. So the software as services component should be place with maximum performance and energy efficiency with minimum cost.

Computational intelligence-based techniques, such as particle swarm optimization [4] and genetic algorithm used to find the solution for component placement problem. Genetic algorithms (GAs) are a heuristic search and optimisation technique inspired by genetic and natural selection used in computer science and engineering. Particle Swarm Optimization is a population-based search and optimization technique based on the intelligence and movement of swarms, it effectively solves large-scale nonlinear optimization problems. The aim of the project is a comparative study to find the cost effective energy efficient Saas component placement based on Particle swarm optimization approach and genetic algorithm approach. Customers pay for the services as per the use basis. So the software as services component should be place with maximum performance and energy efficiency with minimum cost.

2. RELATED WORK

In cloud computing environment, Saas components deploy onto the clouds as to satisfy the consumer's needs. Placement of the Saas components onto cloud server while satisfying all resource requirements, known as Saas Placement Problem (SPP), it is similar to existing problem Component placement problem (CPP). CPP is further categorized into two parts: (1) Online CPP, components are placed during run time, (2) Offline CPP, component placement is at initial stage.

In CPP placement is done at initial stage [5] so SPP is much similar to offline CPP. Existing work on CPP is related with data centre's resource allocation to the application components. Several existing studies formulated CPP as a resource optimization problem and also as a variant of the multiple knapsack problems.

Kichkaylo et. al., [5] defined the application by as set of interface and component types where each of the components for the execution through interface specifies required service.

In this paper, a general model for CPP is proposed and presented an algorithm based on efficient planning algorithms developed by artificial intelligence community. There was a drawback with this algorithm that if the resources are tight it may fail.

Karve et. al., [6] proposed a middleware clustering technology through which resources can be allocated to web application through dynamic Application Instance Placement (AIP). Karve divided the resources into two categories; one is load independent that do not completely depend on the intensity of application workload and another one is load dependent that depends on the intensity of application load.

Saas placement is also said to be NP-hard problem. Zimmerova et. al., [7] focused on the relational aspect between components of the problem presented in, and proposed a solution concerning both non-interaction and interaction properties as well. Urgaonkar et. al., [8] used the first-fit based approximation algorithm for placing component applications in an offline component placement problem. The algorithm proposed by Urgaonkar places the component at the first server found that can satisfy its demand.

Zhu et. al., [9] addressed the Application Component Problem (ACP). Application component ploblem is to decide which physical server should host the application component in order to minimize the processing, storage and communication requirement and resources are effectively used as well. Yusoh et. al., [10] proposed a Genetic algorithm with penalty, for composite Saas placement problem in cloud, considering both data component as well as software of Saas. The objective was to optimize the Saas performance based on its total execution time and optimization of resource consumption in each server.

Zhenzhang et. al., [11] proposed a method for solving Saas placement problem based on the Ant Colony Optimization technique and also, presented a model for deploying the Saas components in the cloud computing environment, and claimed to perform better than the genetic algorithm for the same. Zhipiao et. al., [12] established a request model for cloud service and proposed a cost-aware scheduling technique based on genetic algorithm for servicing request those too cost effective and not violating SLA constraints. Their approach was not only limited to reusing the resources but minimizing rental cost and maximizing providers profit.

Bhardwaj et. al., [13] proposed a particle swarm optimization approach for cost effective saas placement on cloud. It satisfies the resource requirements processing capacity, memory, storage capacity, and bandwidth. Grant Wu1 et. al., [14] proposed Energy-Efficient Virtual Machine Placement in Data Centers by Genetic Algorithm. It considers energy efficiency of the physical server and satisfies all resource requirements.

3. PLACEMENT PROBLEM

A. SaaS Placement Problem

Placement of each Saas component onto the virtual machines running on the servers, and performance of the Saas is optimal based on the cost, while satisfying all the resource constraints. Consider a possible scenario of saas components placement on virtual machine, as shown in Figure 1.

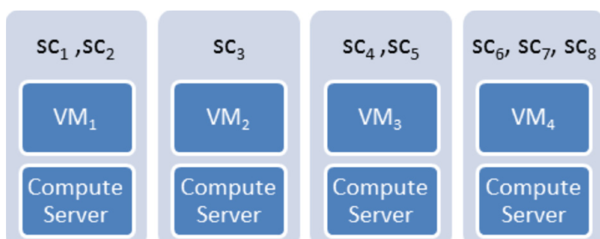

Figure 1: A possible scenario of Saas Components placement in Cloud

Virtualization is the ability to run multiple operating systems on a single physical system and share the underlying hardware resources. It Provides flexibility, scalability needed for a computing component. Virtual machines are deployed onto the set of server available.

$VM = \{vm_1, vm_2, vm_3, ..., vm_m\}$. VM is represented with a tuple $(PC_{vmi}, M_{vmi}, SS_{vmi}, BW_{vmi})$

where, PC_{vmi} = Processing Capability of vm_i

M_{vmi} = Main memory Capacity of vm_i

SS_{vmi} = Storage Capacity of vm_i

BW_{vmi} = IO Capacity or Bandwidth of vm_i

$SC = \{sc_1, sc_2, sc_3, ..., sc_n\}$. The resource requirement with tuple $(TS_{sci}, MM_{sci}, S_{sci}, IO_{sci})$

where, TS_{sci} = Task Size of sc_i

MM_{sci} = Main memory Capacity of sc_i

S_{sci} = Size of sc_i

IO_{sci} = IO Requirement sc_i

1. Resource Constraints

While a Saas component placed on a virtual machine the total resource requirements for Saas components that are to be placed in either compute servers/storage servers or virtual machines must not exceed the VM's resource capacity.

$$\forall vm_i \in VM \sum_{sc_j} TS_{sc_j} \leq PC_{vm_i} \mid P(sc_j) = vm_i$$

$$\forall vm_i \in VM \sum_{sc_j} MM_{sc_j} \leq M_{vm_i} \mid P(sc_j) = vm_i$$

$$\forall vm_i \in VM \sum_{sc_j} S_{sc_j} \leq SS_{vm_i} \mid P(sc_j) = vm_i$$

$$\forall vm_i \in VM \sum_{sc_j} IO_{sc_j} \leq BW_{vm_i} \mid P(sc_j) = vm_i$$

2. Cost Calculation

Cost to place a saas component in a virtual machine is the sum of cost of processing, memory, storage and bandwidth cost and

$$\text{Total cost} = \sum_{i=1}^{m} \sum_{j=1}^{n} X_{i,j} \times C_{i,j}$$

where, $X_{i,j} = \begin{cases} 1 & \text{if } P(sc_j) = vm_i \\ 0 & \text{otherwise} \end{cases}$

$C_{i,j} = tet_{i,j} \times \cos t_{vm_i}$

$C_{i,j}$ = cost incurred due to j^{th} Saas component placement on i^{th} VM

$tet_{i,j}$ = total execution time of j^{th} Saas component when it is placed on i^{th} VM

$tet_{i,j}$ = calculated based on the processing and transferring of the component data

$cost_{vmi}$ = processing cost + memory cost + storage cost bandwidth cost

Virtual Machine Placement Problem

Placement of each Virtual machine onto the physical server, and performance of the virtual machine is optimal based on the energy. Saas components are placed on the virtual machines are optimal based on the cost, as shown in Figure 2.

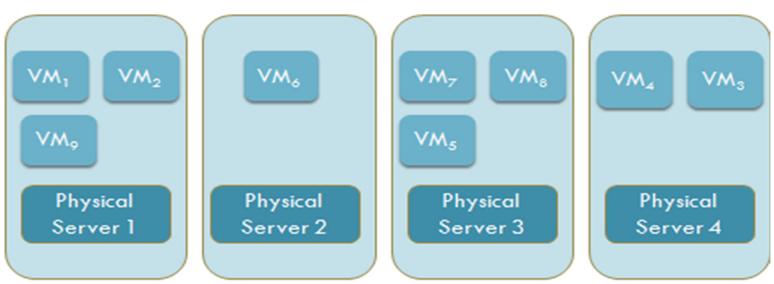

Figure 2: A possible scenario of virtual machine placement in server

The utilization rate of cpu in physical server p_j is

$$\mu_j = \frac{p_j^{w_{cpu}}}{p_j^{cpu}}$$

Energy consumption of physical server p_j when its cpu usage is μ_j is

$$E(pj) = k_j . e_{max_j} + (1 - k_j) . e_{max_j} . \mu_j$$

where, k_j is the fraction of energy consumed when p_j is idle; e_{maxj} is the energy consumption of physical server p_j when it is fully utilized; and μ_j is the CPU utilization of p_j. It uses Genetic Algorithm [14] to place the virtual machine in physical server. Energy consumption is considered as the fitness in genetic algorithm.

4. ALGORITHM COMPARISON

A. PSO Approach

Placement of Saas component algorithm, Saas components Placement using Particle Swarm Optimization Approach (SPPSOA) [13], allocates a fixed number of Saas components to VMs. The position vectors (or particles) and velocities are updated based on the local and global best solutions obtained using the following equations:

$$X_k^{new}(m, n) = \begin{cases} 1 & \text{if } Vel_k^{new}(m, n) = \max Vel_k^{new}(m, n) \\ 0 & \text{otherwise} \end{cases}$$

$$Vel_k^{new}(m, n) = Vel_k^{old}(m, n) + c_1 \times r_1 \times \left(lbest_k^{old}(m, n) - X_k^{old}(m, n)\right) + c_2 \times r_2 \times \left(gbest_k^{old}(m, n) - X_k^{old}(m, n)\right)$$

Algorithm: SPPSOA

Input: VMs and SCs with their capacity and requirements respectively

Output: Sub-optimal solution for placement of Saas

```
Initialize particles, f (local_best) = ∞, f (globel_best) = ∞;
repeat
  for each particle i = 1... P do
    if sc resource requirements ≤ vm capacity and sc can be placed on vm then
      calculate cost for the Particle
    end
    if f (X_i) < f (local_best_i) then
      local_best_i ← X_i;
    end
    if f (local_best_i) < f (globel_best_i) then
      globel_best_i ← local_best_i;
    end
  end
```

```
    for each particle i = 1, ..., P do
      Update Velocity and Particles;
    end
until maximum iterations reached;
```

B. GA Approach

Genetic algorithms are used as a first choice to solve optimization problem. This algorithm allocates the Saas components to the VMs. Algorithm terminate with a fixed set of iterations.

Algorithm 2: SPGA

Input: SCs and VMs with their requirements and capacities respectively

Output: Sub-optimal solution for placement of Saas

```
Sort SC in descending order based on its processing requirement, SCsort;
  Sort VM in descending order based on its processing capacity, VMsort;
    for sc in SC_sort do
      min ← ∞;
      for vm in VM_sort do
        if sc resource requirements ≤ vm capacity and sc can be placed on vm
          Calculate ET_sc;
          if ET_sc < min then
            min ← ET_sc;
            P(sc) → vm;
          end
      end
    end
    if P(sc) → vm then
      Update vm;
    else
        foundsc ← false;
    end
    apply single point crossover operator to generate new individuals
    apply mutation operator to generate new individuals
    Replace the individuals of old population by new individuals
end
```

A chromosome in the SCPGA represents the placement for the Saas components. The chromosomes contains *m* number of genes, each of which corresponds to the Saas component, representing the ID of virtual machines where the Saas components should be placed and where *m* is the number of Saas components, as shown in Figure 3.

SaaS Components	SC_1	SC_2	SC_3	SC_4	SC_5	SC_6	SC_7	SC_8	…………………	SC_{M-1}	SC_M
Virtual Machine	42	1032	234	88	732	70	114	97	…………………	6	423

Figure 3: Chromosome representation

The crossover operation is single point crossover, as shown in Figure 4. The crossover operation depends on the crossover probability, which gives the information about number of chromosomes would be selected for the crossover operation.

Before Crossover:

sc_1	sc_2	sc_3	sc_4	sc_5	sc_6	sc_7	sc_8	………	sc_{m-3}	sc_{m-2}	sc_{m-1}	sc_m
42	1032	244	88	732	80	98	103	………	15	6	9	423

sc_1	sc_2	sc_3	sc_4	sc_5	sc_6	sc_7	sc_8	………	sc_{m-3}	sc_{m-2}	sc_{m-1}	sc_m
54	97	72	1024	244	4	12	16	………	21	32	13	449

After Crossover:

sc_1	sc_2	sc_3	sc_4	sc_5	sc_6	sc_7	sc_8	………	sc_{m-3}	sc_{m-2}	sc_{m-1}	sc_m
42	1032	244	88	732	80	98	103	………	21	32	13	449

sc_1	sc_2	sc_3	sc_4	sc_5	sc_6	sc_7	sc_8	………	sc_{m-3}	sc_{m-2}	sc_{m-1}	sc_m
54	97	72	1024	244	4	12	16	………	15	6	9	423

Figure 4: Crossover operation

The mutation operation is knowledge-based, which changes the VM to a particular Saas component with a new VM, such that the new VM is more appropriate for the placement of that particular component, as shown in Figure 5. This new VM reduces the overall resource cost and maximize the profit of Saas providers.

Before Mutation

SC_1	SC_2	SC_3	SC_4	SC_5	SC_6	……………………	SC_{m-3}	SC_{m-2}	SC_{m-1}	SC_m
42	1032	234	88	732	70	……………………	21	32	13	449

After Mutation

SC_1	SC_2	SC_3	SC_4	SC_5	SC_6	……………………	SC_{m-3}	SC_{m-2}	SC_{m-1}	SC_m
42	1032	234	88	732	50	……………………	21	32	13	449

Figure 5: Mutation operation

5. EXPERIMENTAL RESULT

The SPPSOA was compared against the SPGA for different number of SaaS components. The results for different SaaS components are presented in Figure 6 and Figure 7.

For Virtual Machines capacities:

$$PC_{vmx} = 1 \text{ to } 10 \text{ Gbps},$$

$$MM_{vmx} = 1000 \text{ to } 20000 \text{ B},$$

$SSVM_x$ = 20 to 2000 MB, and

IO_{vmx} = 10 Mbps

For Saas Components requirements:

TS_{sci} = 20 to 200 MB,

M_{sci} = 1000 to 10000 B,

S_{sci} = 10 to 100 MB, and

IO_{sci} = 100 to 200 MB.

A. Performance Evaluation with Different Number of Saas Components

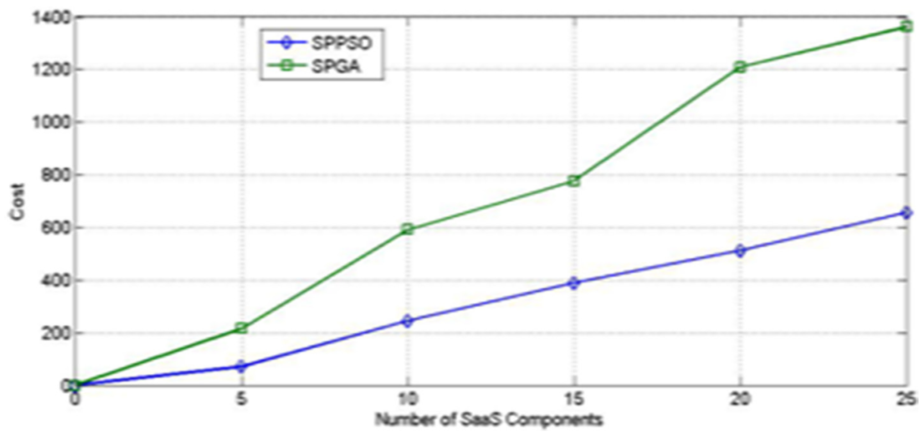

Figure 6: Experiment on number of Saas Components with for 100 VMs

Due to being stochastic nature of SPPSO experiments repeated several time. The SPPSO has always a lower cost value than SPGA, which implies SPPSO gives a better placement option for Saas components placement, as shown in Figure 6.

B. Performance Evaluation with Different Number of Saas Components

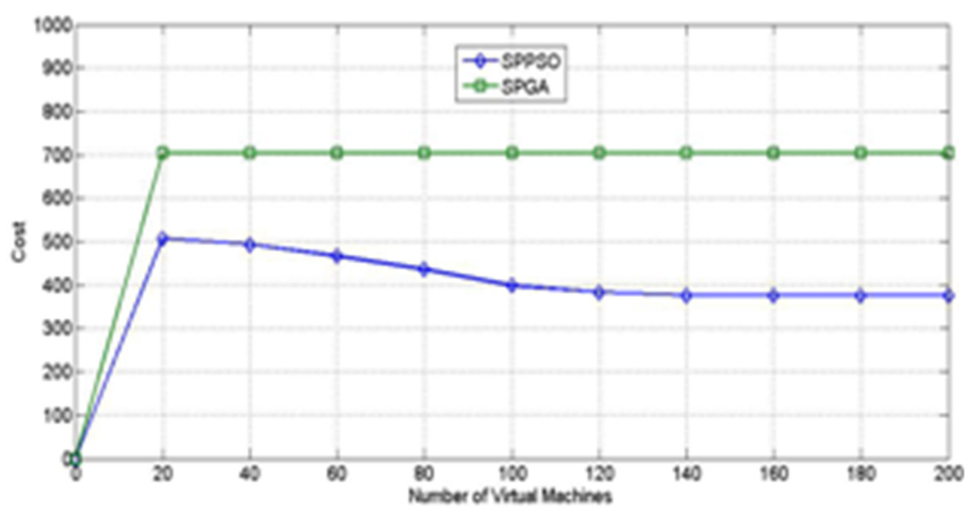

Figure 7: Experiment on number of VMs with 20 Saas Components

Due to being stochastic nature of SPPSO experiments repeated several times. The SPPSO has always a lower cost value than SPGA, which implies SPPSO gives a better placement option for Saas components placement, as shown in Figure 7. Deployment cost of Saas components can be minimized by using SPPSO.

6. CONCLUSION

This research is focuses on computing energy consumed by physical server and the cost of placement of Saas component on the Cloud. Particle Swarm Optimization and Genetic Algorithm implementation have been used. Both algorithms place the Saas component based on the resource constraints i.e. processing capability, main memory, bandwidth capacity and storage. PSO approach considered SLA constraint like response time for finding the cost of resources and profit of Saas providers. GA approaches used for place virtual machine in the physical server. The performance of the algorithm Saas components Placement using Particle Swarm Optimization is compared with GA based Saas components placement. Both energy efficient virtual machine placement and cost effective saas placement give better and efficient saas component placement. From the result comparison, we get PSO approach is the best for cost effective saas placement on cloud.

REFERENCE

[1] Minqi Zhou, Rong Zhang, Dadan Zeng, and Weining Qian. "Services in the cloud computing era: A survey". In *Universal Communication Symposium (IUCS), 2010 4th International*, pages 40–46. IEEE, 2010.

[2] Peter Mell and Tim Grance. "The nist definition of cloud computing," 2011.

[3] Abhijit Dubey and Dilip Wagle. "Delivering software as a service," *The McKinsey Quarterly*, 6(2007):2007, 2007.

[4] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. *Swarm intelligence*, 1(1):33–57, 2007.

[5] Tatiana Kichkaylo, Anca Ivan, and Vijay Karamcheti, "Constrained component deployment in wide-area networksusing ai planning techniques," In *Parallel and Distributed Processing Symposium, 2003 Proceedings. International*, pages 10–pp. IEEE, 2003.

[6] A Karve, Tracy Kimbrel, Giovanni Pacifici, Mike Spreitzer, Malgorzata Steinder, Maxim Sviridenko, and A Tantawi. "Dynamic placement for clustered web application". In *Proceedings of the 15th international conference on World Wide Web*, pages 595–604 ACM, 2006.

[7] Barbora Zimmerova et. al., "Component placement in distributed environment wrt component interaction". In *Proceedings of the 2nd Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, pages 260–267, 2006.

[8] Bhuvan Urgaonkar, Arnold L Rosenberg, and Prashant Shenoy. "Application placement on a cluster of servers". International Journal of Foundations of Computer Science, 18(05):1023{1041, 2007.

[9] Xiaoyun Zhu, Cipriano Santos, Dirk Beyer, Julie Ward, and Sharad Singhal. "Automated application component placement in data centers using mathematical programming". *International Journal of Network Management*, 18(6):467–483, 2008.

[10] Zeratul Izzah Mohd Yusoh and Maolin Tang. "A penalty based genetic algorithm for the composite saas placement problem in the cloud". In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[11] Zhenzhang Liu, Zhihui Hu, and Lee K Jonepun. Research on composite saas placement problem based on ant colony optimization algorithm with performance matching degree strategy. *Journal of Digital Information Management*, 12(4):225, 2014.

[12] Liu, Z., Wang, S., Sun, Q., Zou, H., and Yang, F, "Cost-aware cloud service request scheduling for saas providers," The Computer Journal, p. bxt009, 2013.

[13] Sumit Bhardwaj, Bibhudatta Sahoo National Institute of Technology Rourkela, India. A Particle Swarm Optimization Approach for Cost Effective Saas Placement on Cloud. International Conference on Computing, Communication and Automation (ICCCA2015).

[14] Grant Wu1, Maolin Tang1, Yu-Chu Tian1, andWei Li2. Energy-Efficient Virtual Machine Placement in Data Centers by Genetic Algorithm. School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, QLD 4001, Australia.