Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 65 (2015) 682 – 690

Procedia Computer Science

International Conference on Communication, Management and Information Technology (ICCMIT 2015)

Hybrid Algorithm for Resource Provisioning of Multi-tier Cloud Computing

Marwah Hashim Eawna[a], Salma Hamdy Mohammed[b], El-Sayed M. El-Horbaty[a,b,*]

[a] *Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt*
[b] *Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt*
[ab*] *Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt*

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: marwa_safaar@yahoo.com, s.hamdy@cis.asu.edu.eg, shorbaty@cis.asu.edu.eg

Abstract

Cloud computing is a model for delivering information technology services in which resources are retrieved from the internet through web-based tools and applications. The most important challenges are influenced to adopt the cloud computing technology such as security, resources allocation and resources provisioning. The only existing resource provisioning mechanisms in cloud computing using meta-heuristic technique are based on single tier application. In this paper we propose dynamic resources provisioning in multi-tier application by using meta-heuristic technique such as Particle Swarm Optimization (PSO) algorithm, Simulated Annealing (SA) algorithm and hybrid algorithm that combine Particle Swarm Optimization (PSO) and Simulated Annealing (SA). The simulation results show that resource provisioning based on PSO-SA algorithm in multi-tier application is much faster than resource provisioning in multi-tier application based on PSO algorithm and SA algorithm, that is beneficial in the development of cloud computing.

Key Words: Cloud Computing, Resources Provisioning, PSO-Particle Swarm Optimization, SA-Simulated Annealing.

1. Introduction

Cloud computing is an evolving concept which makes better using of multiple distributed resources, cloud users use resource that is provide by the cloud service provider, the important aim in provisioning resource is maximum performance in minimum time, to minimize the total time taken, the scheduling principle should aim to reduce the amount of data transfer with minimum cost and ensure balanced distribution of tasks as per processing capability[1].There are entirely two generic architectures of resource provisioning, single tier and multi-tier, the single tier architecture has relatively simple structure and easy to setup and everything means database, application and presentation are at same place. Most modern Web sites use a multi-tier architecture, this architecture partitions the application process into multiple tiers, each tier provides a certain functionality. Most multi-tier Web applications use a 3-tier architecture: presentation tier, application tier and data tier. The basic feature of cloud computing is the ability to dynamically providing resources scaling, dynamic provisioning is a useful technique for handling the virtualized multi-tier applications in cloud environment[2]. A request in dynamic provisioning technique moves through the tiers, may visit a tier multiple times and get processed at the visited tier. Finally, the processing completes and returns to request senders from the front tier[3].

Some new search techniques involving nature-inspired meta-heuristics have become the new focus of resource provisioning and allocation research in single tier cloud computing such as Particle Swarm Optimization (PSO) which inspired by the social behavior of bird flocking and fish schooling, and Simulated Annealing (SA) that inspired by the way

Peer-review under responsibility of Universal Society for Applied Research
doi:10.1016/j.procs.2015.09.012

in which a metal cools into a minimum energy crystalline structure (the annealing process) and search for a minimum in a more general system. In this paper we propose dynamic resources provisioning in multi-tier cloud computing based on PSO algorithm and SA algorithm and hybrid algorithm that combine PSO-SA algorithms.

The rest of this paper is organized as follows: Section 2 an overview on the related work algorithm. In Section 3, we describe the proposed PSO, SA and PSO-SA algorithm to provide resources in multi-tier cloud computing. Experimental results are presented in Section 4. Section 5 contain conclusion and future work.

2. Related works

Resource provisioning problem has been an important topic in the cloud computing. To solve this problem we need to improve resource utilization and provisioning in minimum time and cost with effectiveness of system and meet Service Level Agreement (SLA). This problem has attracted a lot of attention from the research community in the last few years. In the following we provide a review of most relevant prior work.

Suraj Pandey et al.[4] presented a scheduling heuristic based on Particle Swarm Optimization (PSO). They used the heuristic to minimize the total cost of execution of scientific application workflows on Cloud computing environments.

```
Calculate average computation cost of all tasks in all
compute resources
Calculate average cost of (communication/size of data)
between resources
Set task node weight w_{kj} as average computation cost
Set edge weight e_{k1,k2} as size of file transferred be-
tween tasks
Compute PSO({t_i}) /* a set of all tasks i ∈ k*/
repeat
    for all "ready" tasks {t_i} ∈ T do
        Assign tasks {t_i} to resources {p_j} according
    to the optimized particle position given by PSO
    end for
    Dispatch all the mapped tasks
    Wait for polling_time
    Update the ready task list
    Update the average cost of communication between
    resources according to the current network load
    Compute PSO({t_i})
until there are unscheduled tasks
```

Fig. 1: The pseudo code of PSO algorithm in Single tier application

They calculate the average computation cost (assigned as node weight in Figure 1) of all tasks on all the compute resources. This cost can be calculated for any application by executing each task of an application on a series of known resources and scheduling the tasks depending on this cost. The experiment based on figure 1 show that when use PSO in task-resource mapping we can achieve at least three times cost savings as compared to Best Resource Selection.

Othman et al.[5] Proposed a novel Simulated Annealing (SA) algorithm for scheduling tasks in cloud environments. SA exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system.

```
Calculate the cost using Oposite () initial solution;
Initial temperature T:
    While (T>Tfinal and Tstop>0) do
   for I = 1 to I do
Inversion-move-set ()
       Calculate the new solution;
        If the new solution < = current solution then
the solution is accepted;
        Else
           R = Random Number (0<R<1); Y = exp(- E/T);
           If (R<Y) then accept the solution;
           Else reject the solution
            End if
            End if
        If we accept the solution then
           Current solution is set to the new solution;
        Else the old solution is kept;
       End if
       If we accept the new solution then Tstop = Ts;
      Else Tstop = Tstop-1;
      End if
      End for
Change the temperature by a factor x, T = T * x
      End while
```

Fig. 2: The pseudo code of SA algorithm in single tier application

They start at the initial temperature T. then execute the SA using the Inversion move set technique discussed in Section. Then, calculate the new cost and assign to it a variable called new gain. If the new gain is less than or equal to the current gain then the solution is accepted. If the new gain is more than the current gain then calculate exp (-E/T) and generate a random number. If the random number generated is less than e(-E/T) then the solution is accepted, otherwise the solution is rejected. If the solution is accepted then the new gain is assigned to the current gain and the new sorted vector is used for generating the next solution. If the solution is not accepted then the current gain does not change and the sorted vector will remain the same. I iterations are performed for each temperature value and after M iterations, the value of the temperature is changed to be T = T * x. The stopping condition T_s is chosen as 5 and is fixed. It is used to stop the algorithm when there is no change to the solution after T_s iterations. In this way the algorithm terminates either on reaching the final temperature T_{final} or after Ts iterations. Repeat the process if T is greater than the final temperature and if t_{stop} is greater than 0 ($t_{stop} > 0$). Finally, the algorithm stops after having met any of the two conditions. Experiment results show that this approach for job scheduling not only guarantees the Quality of Service (QoS) requirement of customer job but also ensures to make best profit of cloud providers. It has also concern about real execution time of jobs in different systems as well as deadline and penalty cost in the algorithm.

Talwinder Kaur et al.[6] Improved Particle Swarm Optimization (IPSO), Simulated Annealing (SA) Algorithm, and Hybrid Particle Swarm Optimization-Simulated Annealing based on utilization time are studied which were proposed to handle problems posed by network properties between user and resources.

```
PSO-SA algorithm
1.  Select N initial Solutions Randomly.
2.  Initialize Variables.
3.  Find the midpoint of present position and target.
4.  Find the nearest particle to the midpoint and initialize it as local Best.
5.  Implement SA Algorithm to generate global Best.
6.  At the end of iteration?
    {
    6.1.  Output the optima.
    }
7.  Update velocity and position of particle.
```

Fig. 3: The pseudo code of PSO-SA algorithm in single tier application

Randomly initialize variables for PSO then find the Center Position b/w the Present particle and target. Center position = (Present position+Target)/2 then find the nearest particle to the center position and initialize it as local Best. For each particle implement SA (Stimulated annealing) algorithm to generate global Best. If the end of iteration is reached, the optima will be returned otherwise next step will be followed. Then update velocity and position according to local Best using PSO. The experiment based on figure 3 show that when they use provisioning resource based on hybrid algorithm that combine PSO and SA algorithm obtain very less time to provide resource as compared to the existing algorithm.

Veena goswami et al.[3] Proposed a dynamic provisioning technique for multi-tier applications to allocate resources efficiently using queuing model. That an optimal autonomic virtual machine provisioning architecture for cloud data center to minimize the congestion in the network by varying the service rate of the virtual machines. The paper analyzes the performance of the virtual machines and dynamically increases the mean service rate of the VMs to avoid congestion in the multi-tier environments and improve the efficiency and flexibility in cloud environment for resource provisioning.

Hadi Goudarzi el al.[7] Considered the problem of the resource allocation to optimize the total profit gained from the SLA contracts and lost from operational cost. They assume that servers are characterized by their maximum capacity in three dimensions: processing power, memory usage, and communication bandwidth. While guaranteeing SLAs for clients with applications that require multiple tiers of service to complete.

Heng et al. [8] Use a benefit-aware approach with feedback control theory to solve the problem of continuously guarantees the SLA in the new configuration in multi-tier application. This approach can reduce resource provisioning cost by as much as 30% compared with a cost oblivious approach, and can effectively reduce SLA violations compared with a cost-aware approach.

In this paper we propose dynamic resources provisioning in multi-tier cloud computing based on PSO algorithm and SA algorithm and hybrid algorithm that combine PSO-SA algorithm.

3. Proposed algorithms

Different approaches have been used in resource provisioning, the only existing resources provisioning approach in cloud computing using meta-heuristic technique are based on single tier application, In our proposed algorithm, use different algorithms of meta-heuristic technique such as PSO algorithm and SA algorithm and hybrid algorithm that combine PSO-SA algorithm to provide resources in multi-tier application of cloud computing.

3.1. Particle Swarm Optimization (PSO)

In our proposed algorithm, we use PSO as a local searching select local best position (*Lbest*) and global searching to select global best position (*Gbest*). To apply the PSO algorithm in multi-tier application the following steps should be taken and repeated in each tier

Step 1: Generate the initial population and initial velocity
Step 2: Calculate the objective function value for each individual depend on following equations:

ECT=ST+DU+EET (1)
EET= (ST-FT) + DU (2)

The ECT represent expected completion time, ST represent first time, DU represent duration time between arrival particles till to start, EET represent the estimated execution time and FT represent Finish time.
Step 3: Sort the initial population based on the objective function values. .
Step 4: Select the local best position (*Lbest*).
Step 5: Select the global best position (*Gbest*) based on *Lbest.*
Step 6: i=i+1, check if the end of iteration, else go back to Step 4.
Step 7: Check if *Gbest* need to migrate and get the output.
The pseudo code of the PSO algorithm is depicted as follows:

```
Procedure PSO algorithm
Generate initial population and initial velocity
 For each particle i=1,...,P do
     For each tier t=1,...,Tiers do
       Calculate the objective function
        Select the Lbest based on the objective function

   Select the Gbest based on the best local position
    End
End
If Gbest need to migration
  Finding out migration source machine
  Finding out migration object machine
  Finding out migration different value
  Finding out migration VM
  Migrate
Else
Output final Gbest.
End
```

Fig.4: The pseudo code of PSO algorithm in multi-tier application

In global PSO system, information is one-direction flow, *Gbest* transfer information to other particles, other particles search near *Gbest*, the whole particle swarm evolve to the optima with *Gbest*. *Gbest* has strong effect on optimal performance of PSO, poor searching ability of PSO for *Gbest* is one of the main reasons for the prematurity of algorithm. To improve optimal performance of PSO, *Gbest* can be sampled by SA after every iteration of particle swarm, whose result can be taken as new *Gbest* of PSO system. Application of SA increase the searching ability of PSO for *Gbest*, so increase the probability of jumping out of local optima[9].

3.2. Simulated Annealing (SA)

In our proposed algorithm, we use SA as global searching to select global best position (*Gbest*) and to search around *Gbest* to select best between them. To apply the SA algorithm in multi-tier application the following steps should be taken and repeat in each tier
Step 1: Generate the initial population and initial velocity
Step 2: Calculate the objective function value for each individual based on Equ.1 and Equ2.
The ECT represent expected completion time, ST represent first time, DU represent duration time between arrival particles till to start, EET represent the estimated execution time and FT represent Finish time.

Step 3: Sort the initial population depend on the objective function values. .
Step 4: Select the global best position (*Gbest*).
Step 5: Apply SA to search around the global solution *Gbest*. If the solution obtained by SA is better than previous *Gbest* then swap with new *Gbest*.
Step 6: i=i+1, check if the end of iteration else go back to Step 4.
Step 7: Check if *Gbest* need to migrate and get the output.

The pseudo code of the SA algorithm is depicted as follows:

```
Procedure SA algorithm
Generate initial population and initial velocity
 For each particle i=1,...,P do
     For each tier t=1,...,Tiers do
       Calculate the objective function

   Gbest=objective function
   Search around the Gbest solution by SA

  If the new solution better than the Gbest solution
      Replace Gbest with the new solution
     End
End
If Gbest need to migration
  Finding out migration source machine
  Finding out migration object machine
  Finding out migration different value
  Finding out migration VM
  Migrate
Else
Output final Gbest.
End
```

Fig.5: The pseudo code of SA algorithm in multi-tier application

A limitation in the proposed algorithm is that the requirements should be known in advance for initial allocation to take place. This information may not be known in certain scenarios. Hence, we aim to design such algorithms that offer a flexibility of not knowing the requirements in advance such that combined algorithm of PSO and SA (PSO-SA) algorithm.

3.3. Particle Swarm Optimization and Simulated Annealing (PSO-SA).

In our proposed algorithm, in each tier in multi-tier application we use PSO as a local searching select local best position (*Lbest*) and global searching to select global best position (*Gbest*), and use SA to search around *Gbest*; in other words, *Lbest* and *Gbest* changes in each iteration.

To implement the PSO-SA algorithm the following steps should be taken and repeated to each tier of multi-tier application.
Step 1: Generate the initial population and initial velocity.
Step 2: Calculate the objective function value for each individual based on Equ1 and Equ2.
The ECT represent expected completion time, ST represent first time, DU represent duration time between arrival particles till to start, EET represent the estimated execution time and FT represent Finish time.
Step 3: Sort the initial population depend on the objective function values. .
Step 4: Select the local best position (*Lbest*).
Step 5: Select the global best position (*Gbest*) based on *Lbest.*

Step 6: Apply SA to search around the global solution *Gbest*. If the solution obtained by SA is better than previous *Gbest* then swap with new *Gbest*.
Step 7: i=i+1, check if the end of iteration else go back to Step 4.
Step 8: Check if *Gbest* need to migrate and get the output.

The pseudo code of the POA-SA algorithm is depicted as follows:

```
Procedure PSO-SA algorithm
Generate initial population and initial velocity
 For each particle i=1,...,P do
     For each tier t=1,...,Tiers do
        Calculate the objective function
         Select the Lbest based on the objective function
   Select the Gbest based on the best local position
   Search around the Gbest solution by SA
  If the new solution better than the Gbest solution
      Replace Gbest with the new solution
     End
End
If Gbest need to migration
  Finding out migration source machine
  Finding out migration object machine
  Finding out migration different value
  Finding out migration VM
  Migrate
 Else
Output final Gbest
End
```

Fig.6: The pseudo code of PSO-SA algorithm in multi-tier application

PSO algorithm use to find local best particle and to find global best particle, it is poor searching ability of PSO for *Gbest* is one of the main reasons for the prematurity of algorithm.SA algorithm has problem with initial allocation to take place. So we combine PSO algorithm with SA algorithm, the PSO-SA algorithm can find the best particle that by use PSO algorithm to find *Lbest* particle and *Gbest* particle and use SA algorithm to search around *Gbest* and select best between them.

4. Results and discussions

To verify the efficiency and effectiveness of proposed algorithms, we use the CloudSim toolkit to provide resource based on the proposed algorithms. The experimental results of resource provisioning based on hybrid algorithm that combine PSO and SA algorithm in multi-tier application are compared with the experimental results of PSO algorithm and SA algorithm as alone in multi-tier application of cloud computing. We have checked the performance of the algorithms by fixed the number of customer owned twice number of virtual machine as host in each datacenter, the results are show in table1.

Table 1: Simulation results for resources provisioning multi-tier application based on PSO, SA and PSO-SA algorithm

Requests	Customers	Data Center	Finish Time(sec.) PSO	Finish Time(sec.) SA	Finish Time(sec.) PSO-SA
500	2	1	6	5	7
1000	3	1	8	6	8
1500	5	2	21	15	15
2000	7	3	33	25	28
2500	8	3	30	30	35
3000	10	5	75	87	76
3500	12	6	93	128	84
4000	13	6	115	142	85
4500	15	6	118	156	103

Based on above data, performance diagram is plotted separately for three resource provisioning algorithms, as shown as in figure 7 that resource provisioning based on PSO-SA algorithm in multi-tier application is less execution time than resource provisioning in multi-tier application based on PSO algorithm and SA algorithm.

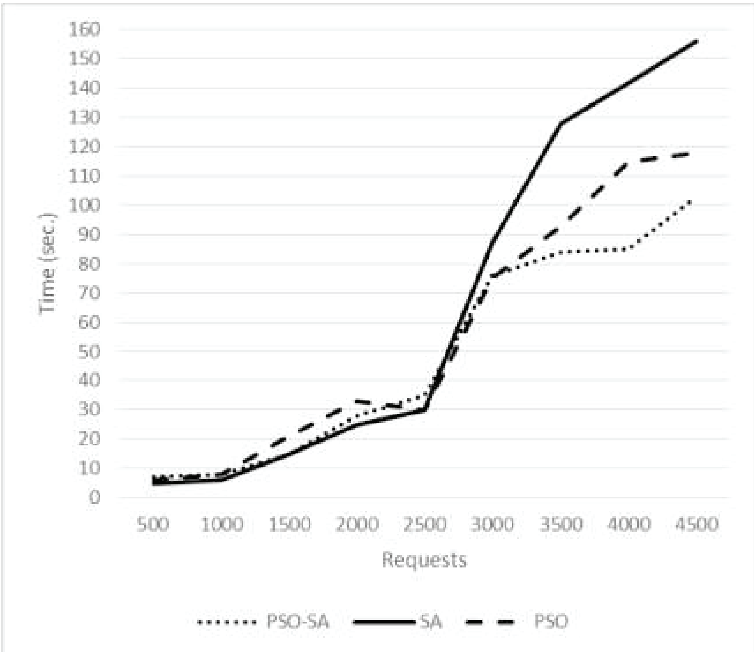

Fig 7: Performance diagram

When the number of requests are less than 2500 there is no difference between execution time of PSO and SA and PSO-SA algorithms, but when exceeded more than 2500 requests the execution time of PSO-SA algorithm is less than PSO and SA algorithms.

5. Conclusion and future work

The work presented in this paper provides a valuable result provisioning resources for a cloud computing. A new method is proposed for resource provisioning in multi-tier cloud computing based on PSO and SA and hybrid algorithm that combine PSO and SA algorithm, Simulation of our proposed algorithms shows that provisioning resource based on hybrid PSO-SA algorithm are good that take less average execution time as compared with resources provisioning based PSO and SA algorithms as alone in multi-tier cloud computing.

As future work, the authors of the paper plan to use other meta-heuristic technique based on provisioning resources and to consider not only execution time but also the cost and memory space share as resources.

References:

[1] Guo, Zhao, Shen and Jiang, "Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm", *JornalOf Networks,* vol. 7, March 2012, p. 547-553.

[2] Jing Bi, Zhiliang Zhu, Ruixiong Tian, Qingbo Wang, "Dynamic Provisioning Modeling for Virtualized Multi-tier Applications in Cloud Data center", *IEEE 3rd International Conference on Cloud Computing*, 2010*,* p. 370-377.

[3] Veena Goswami , S. S. Patra Y, g. B. Mund, "dynamic provisioning and resource management for multi-tier cloud based Applications", *foundations of computing and decision sciences,* vol. 38, 2013, p. 175-190.

[4] Suraj Pandey, LinlinWu, Siddeswara Guru, Rajkumar Buyya, "A Particle Swarm Optimization (PSO)-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments", *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, 2010*),* April 2010, p.20-23.

[5] Monir Abdullah and Mohamed Othman, "Simulated annealing approach to cost-based multi- quality of service job scheduling in cloud computing enviroment", *American Journal of Applied Sciences 11 (6),* 2014, p. 872-877.

[6] Talwinder Kaur, Seema Pahwa, "An Upgraded Algorithm of Resource Scheduling using PSO and SA in Cloud Computing", *International Journal of Computer Applications ,* vol. 74, July 2013, p. 28-32.

[7] Hadi Goudarzi and Massoud Pedram, "Multi-dimensional SLA-based Resource Allocation for Multi-tier Cloud Computing Systems", *IEEE International Conference on Cloud Computing (CLOUD)*, 2011*,* p. 324 - 331.

[8] Heng WU, Wenbo ZHANG, Jianhua ZHANG,JunWEI and Tao HUANG, "A benefit-aware on-demand provisioning approach for multi-tier application in cloud computing", *Frontiers of Computer Science,* vol. 72013, p. 459–474,.

[9] Shaobin Zhan, Hongying Huo, "Improved PSO-based Task Scheduling Algorithm in Cloud Computing", *Journal of Information & Computational Science*, November 1, 2012 *,* p. 3821–3829.