A PSO-GA based hybrid Algorithm for the composite SaaS Placement Problem in the Cloud

Yin Nyein Aye,Thinn Thu Naing
University of Computer Studies,Yangon
yinnyeinaye.ptn@gmail.com, *thinnthu@gmial.com*

Abstract

Cloud computing is an emerging computing paradigm in which applications, data and IT resources are provided as a service to users over the Internet. One kind of services that can be offered through the cloud is Software as a Service or (SaaS). For SaaS placement in the Cloud, the problem relates to how a composite SaaS should be placed in a Cloud by the Cloud's providers such that its performance is optimal based on its estimated execution time. The challenges in the SaaS placement process rely on several factors, including the size of the Cloud network, SaaS competing resource requirements, SaaS interactions between its components and SaaS interactions with its data components. This paper proposes a PSO-GA (Particle Swarm Optimizing-Genetic Algorithm) based hybrid algorithm to the composite SaaS placement problem in the Cloud storage for minimizing the execution time and execution cost of applications on the resources provided by cloud.

1. Introduction

With the rapid development of processing and storage technologies and the success of the Internet, computing resources have become cheaper, more powerful and more ubiquitously available than ever before. This technological trend has enabled the realization of a new computing model called cloud computing, in which resources (e.g., CPU and storage) are provided as general utilities that can be leased and released by users through the Internet in an on-demand fashion. [2] The emergence of cloud computing has made a tremendous impact on the Information Technology (IT) industry over the past few years, where large companies such as Google, Amazon and Microsoft strive to provide more powerful reliable and cost efficient cloud platforms and business enterprise seek to reshape their business models to gain benefit from this new paradigm. Cloud computing is broadly classified into three services: “Software”, “Platform” and “Infrastructure”. Each service serves a different purpose and offers different products for businesses and individual people around the world. The high cost of creating and maintaining software and hardware infrastructures for delivering services to business has led to a notable trend toward the use of third-party services which provide network presence, computation power and data storage space to clients with infrastructural needs. These third party services can act as data stores as well as entire software suites for improved availability and system scalability, thereby reducing business's burden of managing complex infrastructures. This provisioning is called information/application outsourcing or software as a service (SaaS). Among these, Software as a service (SaaS) provides an application on the basis of users' requirement. It is a type of model for software deployment in which an application is hosted as a service provided to the user across network. There is no need to install and run the application on the user's own computer. [5] For SaaS placement in the Cloud, the problem relates to how a composite SaaS should be placed in a Cloud by the Cloud's providers such that its performance is optimal based on its estimated execution time. The challenges in the SaaS placement process rely on several factors, including the size of the Cloud network, SaaS competing resource requirements, SaaS interactions with data components and SaaS interactions with data components. Generally speaking, the function of a cloud workflow system and its role in a cloud computing environment is to facilitate the automation of user submitted relationships defined by graph-based modeling tool such as DAG (direct acyclic graph). Meanwhile, application data can be hosted on different storage resources at the global cloud infrastructure. When one task needs to process data from different data centers, moving the data becomes a challenge. In order to efficiently and cost effectively schedule the tasks and data of applications among services, end user QoS-based scheduling strategies are implemented such as those for minimizing makespan, minimizing total execution cost and balancing the load of resources. The services are attached with their resources which are database, dataset, network, infrastructure, middleware, software, etc. SaaS execution time as the data is located at the cloud servers instead of local machines. Existing SaaS placement methods were not designed for cloud. In the SaaS placement case, the data as well as the components will be located at the Cloud provider's servers. As such, the Cloud providers must apply a strategic placement method in order to ensure the services and the data are well placed and the SaaS performance is optimized. Optimization is the process of making something better. It consists in trying variations on an initial concept and using the information gained to improve on the idea. It amounts to building, applying and studying algorithms based on Darwinian principles of natural selection. There are many algorithms for optimizing. This paper will propose a placement solution to address this gap. The particle swarm method for function optimization has been introduced by Kennedy and Eberhart in [8]. The ability of groups of some species of animals to work as a whole in locating desirable positions in a given area is simulated. It has better ability of global searching and has been successfully applied to many areas [1]. This algorithm is predominately employed to find solutions for continuous problem without prior information. Several approaches have been developed for PSO to solve discrete problem. This concept has been proved to be promising. Based on the set-based scheme, we use PSO-GA (particle swarm optimizing and genetic algorithm) to minimize the execution time and execution cost of applications on the resources provided by cloud. This research will address this problem by proposing an algorithm which considers not only the placement of the software components of a SaaS but the placement of data of the SaaS as well. The remaining paper is organized as follows. Section 2 discusses related work and section 3 describes the proposed algorithm. Finally, section 4 addresses the conclusion.

2. Related Work

There are three main roles in a SaaS scenario of a Cloud. They are the SaaS vendor, Cloud infrastructure provider and SaaS customer. The SaaS vendor submits the composite SaaS with its data to the Cloud provider, where the Cloud provider is responsible for hosting and placing the SaaS in the Cloud based on an agreement between the SaaS vendor and the Cloud provider. The placement of the composite SaaS has to be done strategically, as the SaaS components and associated data are dependent upon each other. This is so called SaaS placement problem or SPP. The objective of the problem is to decide where each of the SaaS components and data chunks should be placed such that some requirements are satisfied and the SaaS performance is optimal based on its estimated execution time. The decision of SPP will be based on estimated SaaS execution time on a set of selected Cloud servers. The calculation of the estimated time is mainly based on the computing resources and data transfer time between the servers that have the data of the SaaS. The services are attached with their resources which are database, dataset, network, infrastructure, middleware, software, etc. The previous research, a penalty-based GA is scalable and it increases the computation time to linearly when the size of the Cloud increases and when the number of components is increased. [12] And second proposed algorithm, cooperative co-evolutionary algorithm can produce better quality solutions than the classical genetic algorithm for all the tested problems although its computation time is noticeably longer than that of the classical genetic algorithm. [13]

Another paper for composite SaaS is in[4] propose a framework for request-routing and load balancing horizontally and vertically in SOA-based enterprise cloud computing to find the best assignment of requests to implementations and service types instances to servers in order to minimize the running time of workload based on a genetic search heuristics. A placement that is concerned with SaaS is presented in [11]. In this research proposes a multi-tenant SaaS engineering approach for onboarding of tenants with custom needs, efficient testing for multiple tenants with a mix of common/custom behavior and refactoring techniques to increase the maintainability and economic value of multi-tenant SaaS systems.

In the figure 1, when user requests a query, the services are integrated their resources in data storage to finish the request query.

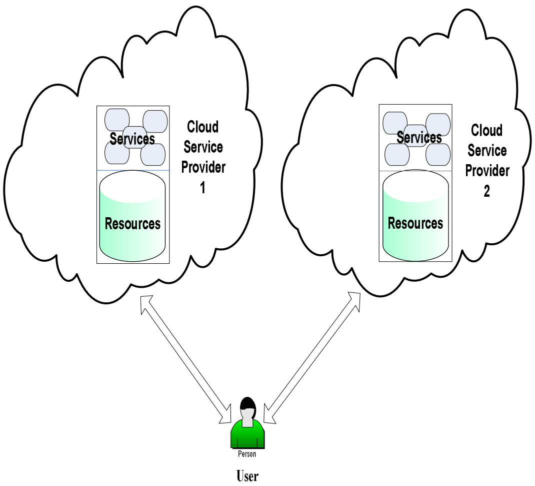

Figure 1. An Overview System

3. PSO-GA based hybrid Algorithm for SaaS placement problem (SPP)

Optimization is the process of making something better. It consists in trying variations on an initial concept and using the information gained to improve on the idea. It amounts to building, applying and studying algorithms based on Darwinian principles of natural selection. There are many algorithms for optimizing. Among them, this paper is proposed a PSO-GA (Particle swarm optimizing- Genetic Algorithm) based hybrid algorithm for SPP. Genetic algorithm is a family of computational models developed by Holland [9]. GA operates on a population of potential solutions by applying the principle of the survival of the fittest to produce successively superior approximations to a solution. At each generation of the GA, a new set of approximations is reproduced by the process of selecting individuals according to the fitness and some other operators from natural genetics. Particle Swarm Optimizing (PSO) also is an evolutionary computational model which is based on swarm intelligence. PSO is developed by Kenndy and Elberhart [8] who have been inspired by the search of the artificial livings. Similar to GAs, PSO is also an optimizer based on population. The system is initialized firstly in a set of randomly generated potential solutions and then performs the search for the optimum one iteratively. Whereas the PSO does not possess the crossover and mutation processes used in GAs, it finds the optimum solution by swarm following the best particle. Compared to GAs, the PSO has much more profound intelligent background and could be performed more easily. PSO can be easily implemented and is computationally inexpensive sine its memory and CPU speed requirements are low. Based on its advantages, the PSO is not only suitable for science research but also engineering applications in the fields of evolutionary computing, optimizing and many others areas. From the computational point of view, SPP is a large scale and complex combinatorial optimizing problem with constraints and an evolution approach would be suitable for it. The previous research is a classical GA [12] and a further improved the quality of solution proposed for a new co-evolutionary algorithm for SPP [13]. Aiming at further improve the quality of solutions, this section proposes a PSO-GA based hybrid algorithm.

3.1. Standard PSO algorithm

Suppose that the searching space is D-dimensional and with N randomly initialized particles in it. Each particle is represented by a D-dimensional vector X_i(i=1,2,….,d) which stands for its location (x_{i1},x_{i2},…,x_{in})in space and it is also regarded as a potential solution. The position of the best individual of the whole swarm is noted as the best previous position P_i (best local solution), and the global best position is noted as P_g (best global solution). The PSO algorithm could be performed by the following equations;

$$v_{id}^{n+1} = wv_{id}^{n} + c_1 r_1 (P_{id}^{n} - x_{id}^{n}) + c_2 r_2 (P_{gd}^{n} - x_{id}^{n}) \quad (1)$$

$$x_{id}^{n+1} = x_{id}^{n} + v_{id}^{n} \quad (2)$$

where d=1,2,…,D; i=1,2,…., s_i; w is called weight; c_1 and c_2 are two positive constants called acceleration coefficients; r_1 and r_2 are two random numbers uniformly from the interval [0,1].Let $T_{x_{id}}$ = total service instances and data storage, $T_{total\,(M)}$ =total completion time, $C_{exe(M)}$ = total execution cost and $C_{trans(M)}$ = total data transmission cost.

$$T_{total(M)} = \max (T_{x_{id}}) \quad (1)$$

$$C_{total(M)} = C_{exe(M)} + C_{trans(M)} \quad (2)$$

As the aim of the placement is to achieve the optimal performance of the SaaS that is being deployed, the main attribute for the fitness function evaluation is the $T_{total\,(M)}$=total completion time for the SaaS components. The objective function is to minimize ($T_{total(M)}$ +$C_{total(M)}$) where M= [<D_i, S_j>] (iϵ[1,m],jϵ[1,n]) where n is the total number of software components in the SaaS and m is the total number of data chunk in storage server. In recent years there have been a lot of reported works focused on the PSO which has been applied widely in the function optimization, artificial neural network training, pattern recognition, fuzzy control and some other fields. Some improved PSO algorithms have also been developed. Similar hybrid PSO algorithms are proposed by introducing operations of GAs into PSO systems [3, 6].GA and PSO are much similar in their inherent parallel characteristics and they have their specific advantages when solving different problems.

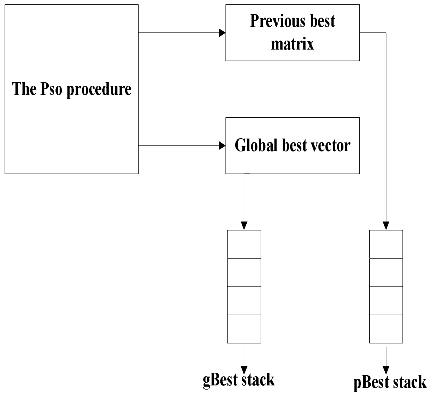

Figure 2. PSO strategy

After a hybrid probability is assigned to each particle, the algorithm selects a certain number of particles into a pool according to the hybrid probabilities at each stage of iteration. The particles in the pool are randomly separated into couples. Each couple reproduces two children by crossover. Then the children are used to replace their parents of the previous particles to keep the number of particles unchanged.

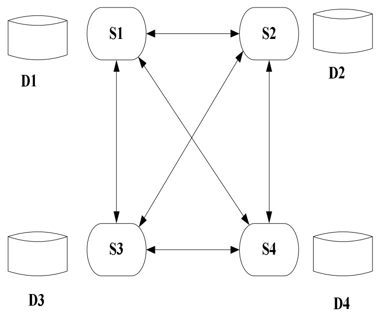

Figure 2. Service instances and data storage

There are a set of service instances $s_i = (s_i^1, s_i^2 \ldots, s_i^n)$ and a set of storage sites $D = (D_1, D2, \ldots, D_m)$ available ,where s_i^j $(1 \leq n_i)$ represents a service instance provided by a content and n_i is the total number of service instances. The properties of a service instance $s_i^j.C$, $s_i^j.R$, $s_i^j.S$ in which $s_i^j.C$ stands for the continents s_i^j while, $s_i^j.R$ and $s_i^j.S$ denote the region and server of s_i^j, respectively.

This paper proposes a novel hybrid approach through crossing over the PSO and GA, called PSO-GA-based hybrid algorithm (PGHA). The proposed algorithm executes the two systems simultaneously and selects P individuals from each system for exchanging after the designated N iterations. The individual with larger fitness has more opportunities of being selected. The performance of the algorithm is described as follows:

Algorithm Description for PSO-GA

Step1: Initialize GA and PSO sub-systems, respectively.

Step2: Execute GA and PSO simultaneously.

Step3: Memorize the best solution as the final solution and stop if the best individual in one of the two sub-systems satisfies the termination criterion.

Step4: Perform hybrid process if generations could be divided exactly by the designated iterative times N. Select P individuals from both sub-systems randomly according to their fitness and exchange. Go to step 2.

Step5: Repeat this process until final solution is reached.

Suppose that the n particle, each of which corresponds to a software component, representing the computation server where the software component would be placed in the placement plan of the SaaS, where *n* is the total number of software components in the SaaS. Each *m* particle corresponds to a data chunk that is used in the SaaS, standing for the storage server where the data chunk would be stored in the placement of the SaaS.

Table 1.Definition of symbols

symbol	definition
$s_i = (s_i^1, s_i^2 \ldots, s_i^n)$	a set of service instances
$D = (D_1, D_2, \ldots, D_m)$	a set of storage sites
n	total number of service instances
m	total number of data chunk for storage
T_{xid}	total service instances and data storage
$T_{total\ (M)}$	total completion time
$C_{exe}(M)$	total execution cost
$C_{trans}(M)$	total data transmission cost
$C_{total}(M)$	total execution cost and total data transmission cost

4. Conclusion

This paper has proposed a PSO-GA based hybrid algorithm for the composite SaaS placement problem in the Cloud. Different from the previous two approaches, it executes the two systems simultaneously and selects P individuals from each system for exchanging after the designated N iterations. The individual with larger fitness has more opportunities of being selected. A PSO-GA based hybrid algorithm is used that it considers not only the placement of software components of a SaaS, but the placement of data of the SaaS as well.

References

[1] D. Bratton and J. Kennedy, "Defining a Standard for Particle Swarm Optimization," in Swarm Intelligence Symposium, 2007. SIS 2007. IEEE, 2007, pp. 120-127.

[2] K.S Candan, W.S Li, T. Phan, & Zhou, M., “Frontiers in information and Software as Services”, 2009, IEEE: Shanghai, China. p. 1761-1768.

[3] H.Y. Fan, “A modification to particle swarm optimization algorithm”, Eng. Computer. 19 (8) (2002) 970–989.
[4] B. Furht , A. Escalante, Handbook of Cloud Computing, ISBN 978-1-4419-6523-3, pp. 277-308.
[5] I. Foster, Z. Yong, I. Raicu, & S. LuCloud “Computing and Grid Computing 360-Degree Compared”, in Grid Computing Environments Workshop. 2008, IEEE: Austin, Texas. p. 1-10.
[6] X.H Shi , Y.C. Lianga, H.P. Lee, C. Lub, L.M. Wang ," An improved GA and a novel PSO-GA based hybrid algorithm", 13 December 2004,www.elsevier.com/locate/ipl.
[7] A. Kundu, C. Banerjee, P. Saha, “Introducing New Services in Cloud Computing Environment” International Journal of Digital Content Technology and its Applications Volume 4, Number 5, August, 2010.
[8] J. Kennedy and R. Eberhart, "Particle swarm optimization”, Pro. The IEEE International Conference on Neural Networks, pp. 1942-1948, Perth, Australia, 1995.
[9] R. L Haupt and S. E. Haupt, Practical genetic algorithm, second edition, published by John Wiley & Sons, Inc., Hoboken, New Jersey 2004.
[10] J. Robinson, S. Sinton, R.S. Yahya, “Particle swarm, genetic algorithm, and their hybrids: Optimization of a profiled corrugated horn antenna”, vol. I, 2002, pp. 314–317
[11]B.Sengupta&A.Roychoudhury, “Engineering Multi-Tenant Software-as-a-Service Systems”, In ICSE’11, May 21–28, 2011, Waikiki, Honolulu, HI, USA.
[12] Z.Yusoh, & M .Tang, “A penalty-based genetic algorithm for the composite SaaS placement problem in the cloud July, 18-23, 2010-CCIB, Barcelona, Spain.
[13] Z. Yusoh, & M. Tang, “A Cooperative coevolutionary Algorithm for the Composite SaaS Placement Problem in the Cloud”. 22-25 November 2010, Sydney. (In Press)

□□□□□□□□□□