ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

An Efficient Dynamic Virtual Machine Provisioning and Allocation in Clouds Using Gravitational Search Algorithm (GSA)

Hazel Ruby Saranya. J, Sudhakar. G

M.E, Research Scholar, Department of CSE. Ranganathan Engineering College, Coimbatore, India

HOD, Dept. of CSE., Ranganathan Engineering College, Coimbatore, India

ABSTRACT: A key challenging problem for cloud providers is manipulative efficient methods for virtual machine (VM) provisioning and allocation. On cloud policy, dividing resources are allocated on-demand dynamically and the framework hosted on a virtual machine(VM) usually has the delusion of total control of resources. Thus, a egotistical VM may deliberately compete for resource with other VMs to exploit its own advantage while at the cost of overall system performance. This problem creates a new challenges to cloud providers, who must prevent non-cooperative performance as well as allocating resource among egotistical VMs efficiently. In this propose a truthful optimal and gravitational search algorithm (GSA) which is one of the newest swarm based heuristic search technique for formulate the dynamic VM provisioning and allocation problem for the auction-based model. The proposed system is to maximize a suitable utility function which accounts for both the QoS delivered to their users and the associated cost. The proposed frameworks are truthful, that is, the users do not have inducements to control the system by untruthful about their requested packages of VM instances and their valuations.

KEYWORDS: Cloud Computing, Gravitational Search, Quality of Service, Cloud Auctioneer

I. INTRODUCTION

The number of enterprises and individuals that are outsourcing their workloads to cloud providers has increased rapidly in recent years. Cloud providers form a large pool of abstracted, virtualized, and dynamically scalable resources allocated to users based on a pay-as-you-go model. These resources are provided as three different types of services: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). IaaS provides CPUs, storage, networks and other low level resources, PaaS provides programming interfaces, and SaaS provides already created applications. In this paper, we focus on IaaS where cloud providers offer different types of resources in the form of VM instances. IaaS providers such as Microsoft Azure [6] and Amazon Elastic Compute Cloud (Amazon EC2) [2] offer four types of VM instances: small (S), medium (M), large (L), and extra large (XL).

Cloud providers face many decision problems when offering IaaS to their customers. One of the major decision problems is how to provision and allocate VM instances. Cloud providers provision their resources either statically or dynamically, and then allocate them in the form of VM instances to their customers.

Most of the past resource allocation works in literatures, such as [2], [3] discussed in cooperative environments, where VMs are cooperative with each others and honestly report their real resource requests to VMM. Based on predefined criterions, VMM allocates resource according to VMs' requests, or dynamically adjusts resource configuration according to the observation of workload or VM environment changes. VMs themselves just passively comply with the algorithm and accepts the resource control results from VMM. Whereas in some practical scenarios, different VMs may host different applications belonging to different users.

The rest of this paper is organized as follows. In Section 2 review the existing related work. The proposed models and descriptions are described in Section 3. Finally conclude the paper in Section 4.

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

II. RELATED WORK

In [1] authors proposed the mechanism design to allocate resource among selfish VMs in a non-cooperative cloud environment. Because the accurate relationship between VM's valuation function and allocated resource may not be available in practice and the valuation function parameters may not noisefree, it also proposeed to apply stochastic approximation methods to get stochastic solution for allocation and payment outcomes. In [2] authors addressed the model the service provisioning problem as a generalized Nash game and to show the existence of generalized Nash equilibria. Moreover, to proposed two solution methods based on the best-reply dynamics and to proved their convergence in a finite number of iterations to a generalized Nash equilibrium. In particular, to developed an efficient distributed algorithm for the run-time allocation of IaaS resources among competing SaaS providers. In [3] authors considered the several Software as a Service (SaaS) providers, that offer a set of applications using the Cloud facilities provided by an Infrastructure as a Service (IaaS) provider. To assume that the IaaS provider offers a pay only what you use scheme similar to the Amazon EC2 service, comprising flat, on demand, and spot virtual machine instances. To proposed a two stage provisioning scheme. In the first stage, the SaaS providers determine the number of required flat and on demand instances by means of standard optimization techniques. In the second stage the SaaS providers compete, by bidding for the spot instances which are instantiated using the unused IaaS capacity. In [4] authors addressed the challenge by proposing TRUST, a general framework for truthful double spectrum auctions. TRUST takes as input any reusability-driven spectrum allocation algorithm, and applies a novel winner determination and pricing mechanism to achieve truthfulness and other economic properties while significantly improving spectrum utilization. To our best knowledge, TRUST is the first solution for truthful double spectrum auctions that enable spectrum reuse. In [5] authors developed an auction-based framework that allows networks to bid for primary and secondary access based on their utilities and traffic demands. The bids are used to solve the access allocation problem, which is that of selecting the primary and secondary networks on each channel either to maximize the auctioneer's revenue or to maximize the social welfare of the bidding networks, while enforcing incentive compatibility. To first consider the case when the bids of a network depend on which other networks it will share channels with. When there is only one secondary network on each channel, to design an optimal polynomial- time algorithm for the access allocation problem based on reduction to a maximum matching problem in weighted graphs.

III. PROPOSED ALGORITHM

A. *CLOUD USERS MODEL*

A cloud user submits a bid defined by $b_i = (d_i, l_i, v_i)$ where $d_i = (d^1{}_i, d^2{}_i, \ldots, d^K{}_i)^T$ is a demand, and $d^k{}_i$ indicates the number of instances of type *k*. $l_i = (l_i, t^s{}_i, t^e{}_i)^T$ is a demand period where l_i indicates a length of time that the user *i* wants to reserve a bundle of the instances between starting time $t^s{}_i$ and ending time $t^e{}_i$. For example, $l_i = (1, 2, 5)$ indicates that user *i* requests instances for 1 time slots (that do not have to be consecutive) between time 2 and time 5, each of the time slots reserves d_i instances. We define $t^d{}_i \equiv t^e{}_i - l_i$ and call it the deadline by which the instances should be assigned to satisfy the demand. v_i is user *i*'s valuation for the demand as a bidding price, which indicates the maximum price that is acceptable for the user to buy the requesting instances. In the proposed system, users and providers can form coalitions to gain benefits in terms of resource price and resource utilization efficiency.

For the multiple cloud users are allowed to form a group, i.e., we call a user coalition, and request instances of the same provider in order to buy the instances at a discounted price. For example, when two users request 10 instances as a coalition, a provider's offering price changes. However, cooperation may incur a delay of the instance allocation because different users have different demand periods. Multiple cloud providers are also allowed to form a group, i.e., we call a provider coalition. The providers in the same coalition can share users' demands to meet their requests. For doing this, users' instances can be migrated from one provider to another as indicated. This cooperation enables the effective use of the residual resources and improves their resource utilization. However, the cooperation may incur a cost for the instance migration. For example, during migration, the degradation of application performance (e.g., throughput and response time) can be considered as the cost.

B. *CLOUD RESOURCE PROVIDER*

The cloud resource provider is the entity that has resources available with him. Memory, CPU, disk, storage space etc all resources are available with it. Provider wants to sell these resources as to gain maximum revenue. It searches

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

for user as to provider services. For this provider will participate in auction process when auctioneer sends messages. If provider has enough resources it can participate and bid for resources. If it wins the auction, will provide service to users and will earn profit. But if after winning auction it is unable to provide service, it will be penalised. Provider can participate in any number of auction processes according to his available resources as there is no limit. But the only condition is that if it commits it will have to provide service.

The user will request resources from auctioneer who will then go to Cloud resource providers for bidding. After receiving bids from Cloud resource provider, auctioneer will calculate winners of the auction. Auctioneer will first calculate weighted total of resources requested by user which can be represented by wt as described in Eq. 1.

$$Rt_n = \sum_{i=1}^{k} (a_i R_i) \quad eq. (1)$$

Cloud provider is the lowest bidder in the auction. If he has enough resources for first user, mapping will be done of first user and the corresponding Cloud provider. But if not, then check for next Cloud provider. Iterate through all providers until user's requirements are satisfied. Same process will be followed for all users.

C. *CLOUD AUCTIONEER*

The cloud auction entity is the centre of this complete auction process. It is responsible for assigning particular resource provider to user. Auctioneer will finalize the price required to be paid by the user. User will raise requirement their request to auctioneer and it will send messages to Cloud resource providers for bidding. The complete auction process will be controlled and managed by him. Auctioneer is responsible for determining auction winners and informing them. If Cloud provider is unable to complete user's task after auction winning, he will be penalised by auctioneer. Auctioneer will calculate penalty for Cloud resource provider and then assign task to next provider.

The three key players involved in auctions are: resource owners (providers), auctioneers (mediators), and buyers (consumers). The auctioneer sets the rules of auction which is agreed by both consumers and the providers. Auctions basically use market forces to negotiate a clearing price for the service. Usually auctions are used particularly for selling goods/items within a set duration. Auctions can be classified into two types, single auctions and double auctions. The single auction model supports one-to -many negotiation, between a provider (seller) and consumers (buyers), and reduces negotiation to a single value (i.e. price). In the double auction model, buyers (bids) and sellers (asks) may be submitted at anytime during the trading period. If at any time there are open bids and asks that match or are compatible in terms of price and requirements (e.g., quantity of goods or shares), a trade is executed immediately. In this auction orders are ranked highest to lowest to generate demand and supply profiles. From the profiles, the maximum quantity exchanged can be determined by matching asks (starting with lowest price and moving up) with demand bids (starting with highest price and moving down). All auctions can be classified as open or closed (sealed) auctions.

D. *GRAVITATIONAL SEARCH ALGORITHM (GSA)*

A GSA is a heuristic approach that trades reductions in computation time against potentially sub-optimal solutions. The principle idea is to initially determine the equilibrium prices for all VM types, such that the expected profit from the served bids is maximized. Subsequently, these served bids – or more specifically, the VM instances that have been requested in these bids – are cost-efficiently distributed across the physical hosts. Accordingly, the approach is split into two phases, VM pricing and VM distribution.

To describe the GSA consider a system with s masses in which position of the i-th mass is defined as follows:

$$S_i = s_i^1, \dots s_i^d, \dots, s_i^n; i = 1,2, \dots, s \quad \text{eq. (2)}$$

where s^d_i is position of the i-th mass in the d-th dimension and n is dimension of the search space. Based on mass of each agent is calculated after computing current population's fitness as follows:

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

$$q_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad eq. (3)$$

$$M_i(t) = \frac{q_i(t)}{\sum_{j=1}^{s} q_j(t)} \quad eq. (4)$$

where Mi(t) and fiti(t) represent the mass and the fitness value of the agent i at t, respectively, and worst(t) and best(t) are defined a minimization problem.

IV. Pseudo Code

Step 1: Search space identification.
Step 2: Randomized initialization.
Step 3: Fitness evaluation of agents.
Step 4: Update G(t), best(t), worst(t) and Mi using eq. (4)
Step 5: for i = 1,2,. . .,*N*.
Step 6: Calculation of the total force in different directions.
Step 7: Calculation of acceleration and velocity.
Step 8: Updating agents' position.
Step 9: Repeat steps c to g until the stop criteria is reached.
Step 10: End

V. Conclusion and Future Work

In this paper proposed a gravitational search algorithm called GSA to solve the Virtual Machine Provisioning and Allocation in Clouds scheduling problem of works in workflow workshop environment considering the criterion of minimizing the sum of works delays and on times for in advance reservation of resources. The advantages of this algorithm are velocity, implementation time, and optimal scheduling of works in workflow workshop environment.

In future work, we intend to improve the proposed algorithm to develop real time cloud with auction methods for optimization to control the resources of the result data.

References

1. Z. Kong, C.-Z. Xu, and M. Guo, "Mechanism Design for Stochastic Virtual Resource Allocation in Non-Cooperative Cloud Systems," Proc. IEEE Fourth Int'l Conf. Cloud Computing, pp. 614-621, 2011.
2. D. Ardagna, B. Panicucci, and M. Passacantando, "Generalized Nash Equilibria for the Service Provisioning Problem in Cloud Systems," IEEE Trans. Services Computing, vol. 6, no. 4, pp. 429-442, Oct.-Dec. 2013.
3. V. Di Valerio, V. Cardellini, and F. Lo Presti, "Optimal Pricing and Service Provisioning Strategies in Cloud Systems: A Stackelberg Game Approach," Proc. IEEE Sixth Int'l Conf. Cloud Computing, pp. 115-122, 2013.
4. X. Zhou and H. Zheng, "Trust: A General Framework for Truthful Double Spectrum Auctions," Proc. IEEE INFOCOM, pp. 999-1007, 2009.
5. G.S. Kasbekar and S. Sarkar, "Spectrum Auction Framework for Access Allocation in Cognitive Radio Networks," IEEE/ACM Trans. Networking, vol. 18, no. 6, pp. 1841-1854, Dec. 2010.
6. WindowsAzure, http://www.windowsazure.com/en-us/pricing/calculator/, 2014.