Middle-East Journal of Scientific Research 20 (12): 2301-2304, 2014
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2014.20.12.328

Mapping Through ILS Using VNE Algorithm

P. Gayathri

Bharath University, Chennai, India

Abstract: Cloud computing promises to provide high performance, on-demand services in a flexible and affordable manner, it offers the benefits of fast and easy deployment, scalability and service oriented architecture. It promises substantial cost reduction together with increased flexibility than the traditional IT operation. Cloud service providers typically come with various levels of services and performance characteristics. In addition, there are different types of user applications with specific requirements such as availability, security and computational power. A simple modification consists of iterating calls to the local search routine, each time starting from a different initial configuration. This is calledrepeated local search and implies that the knowledge obtained during the previous local search phases is not used. Learning implies that the previous history, for example the memory about the previously found local minima, is mined to produce better and better starting points for local search.

Key words: Calledrepeated local search · Providers typically come with

INTRODUCTION

Cloud computing is the delivery of computing services over the Internet. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. Examples of cloud services include online file storage, social networking sites, webmail and online business applications [1]. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available. Cloud computing provides a shared pool of resources, including data storage space, networks, computer processing power and specialized corporate and user applications.

The cloud computing service models are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). In a Software as a Service model, a pre-made application, along with any required software, operating system, hardware and network are provided. In PaaS, an operating system, hardware and network are provided and the customer installs or develops its own software and applications [2]. The IaaS model provides just the hardware and network; the customer installs or develops its own operating systems, software and applications.

Relatedwork

Virtual Network Embedding: Up to now, most algorithms dealing with the VNE problem operate in a cen-tralized manner (a notable exception being the algorithm by Houidi *et al*. [3]). Virtualized network environments, however, are envisioned to be highly dynamic, with resources being frequently added, moved and removed on demand. This has to be taken into account by VNE algorithms. It follows, that these algorithms have to operate in an online manner (as opposed to computing an embedding in an arbitrary amount of time beforehand). For centralized algorithms it follows that there is a single point of failure - namely the instance that computes the embedding. Moreover, the scalability of a centralized solution is questionable. Thus, developing new algorithms that operate in a distributed manner is an important goal. A centralized embedding algorithm uses an external resource to calculate the actual result. However, a distributed approach uses the hosts of the physical network itself for calculations. Work is distributed to the physical nodes and the nodes have to cooperate to _nd an appropriate mapping. A key requirement of a distributed algorithm is that the physical nodes involved in an embedding operation have to cooperate and exchange noti_cation messages to update

Corresponding Author: P. Gayathri, Bharath University, Chennai, India.

their local state of the network. This is similar to peer-to-peer systems where nodes have to send and receive administrative messages, too. Some approaches, like the Kazaa network, try to reduce message and management overhead by partition-ing the network into smaller clusters. These clusters maintain their information autonomously [4]. A special node inside the cluster, the supernode, is selected as the leader of the cluster.

Analogous to the Kazaa approach, the substrate network can be clustered into multiple, non-overlapping parts, too. These parts can be used independently to embed virtual network requests in a distributed manner, in parallel. A node inside a cluster that has the most available CPU resources can then be selected as the clusterhead. This node is responsible for computing the actual network embedding for its part of the network. Depending on the size of the virtual network, an appropriate cluster is selected, where the virtual network should be integrated.

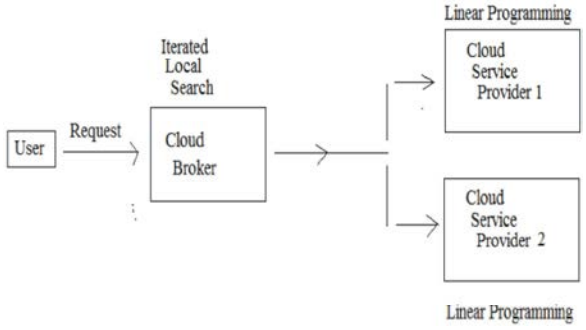
We use an heuristic that estimates the potential of a partition to embed a virtual network request. This is done depending on the amount of resources the virtual network request demands. For this reason, two weight values and _ are introduced. is used to weight the demands of CPU resources and _ for the weight of bandwidth resources demands. A higher or _ value results in a more pessimistic estimation of physical resources [5]. This means that using higher values, the algorithm will try to embed a virtual network into bigger physical partitions, ignoring smaller ones.

Iterated Local Search: The ILS metaheuristic iteratively builds a sequence of solutions generated by the repeated application of an embedded heuristic and perturbation of the local optimum found . The embedded heuristic is most commonly a problem-specific local search technique. The starting point for the search may be a randomly constructed candidate solution, or one returned by a greedy construction heuristic. Usually, employing a greedy solution results to better performance over less improvement steps. Local Search refers to the local improvements of the solution produced from the perturbation. Essentially, local search consists of moving from one solution to another according to some well defined rules related to a neighborhood of solution, accepted moves within the neighborhood and termination criterion.Regarding acceptable movements in the search space, first improvement or best improvement can be used [6]. Finally common stopping criteria for local search algorithms are definition of a maximum number of iterations or terminating the search when no improvement has been observed for a number of iterations.Perturbation generates new starting points of the local search by modifying some local optimum solution.Perturbation guides the algorithm to escape from local optima hence to investigate other parts of the search space. The strength of the perturbation plays a significant role in the efficiency of the algorithm, where strength isusually defined as the number of solution components modified. For a very strong perturbation, ILS behaves like random restart while for a weak perturbation the algorithm's behavior is greedy [7]. Finally, the solution from which the walk is continued is selected according to an appropriately defined Acceptance Criterion. The new local optimum is accepted usually with a fixed rule e.g. accept only better solution. To enhance diversification, memory is incorporated on the decision process by keeping track of previous solutions and restarting the ILS when no improvement is noticed over a defined number of iterations.ILS metaheuristic is adopted for request partitioning.

In this paper, mainly due to its intrinsic simplicity and general applicability, overcoming at the same time performance/scalability issues that arise when addressing the resource mapping problem in real time and for large incoming requests [8].

System Model

System Architecture



Modules Description

Request to Cloud Broker: At first create the user to access the resource in the cloud [9-13]. User must get the password to access the resource. For password the user must register in the cloud service provider [14]. And then the request is send to the cloud broker.

Iterated Local Search: The cloud broker plays important role between the user and the cloud service provider.

Cloud broker gets request from the user and forward it to the cloud service provider [15]. The cloud broker uses the Iterated Local Search to partition the request from the user.

Linear Programming: The cloud service provider provides the resource to the user. Cloud service provider gets request from the cloud broker and mapping the resource by using the linear programming.And then the cloud service provider provides the available resource to the user.

Resource Utilization and Cost Calculation: Then the user access the resource provided by the cloud service provider. According to the size of available resource the user can access it [9]. The cost is calculated on the basis of the available resource and the resource utilization by the user.

CONCLUSIONS

In this work the cloud resource mapping problem over a networked cloud computing environment is studied,by providing high performance algorithms in terms of embedding effectiveness and run time complexity [10]. Within the proposed framework, an ILS-based heuristic is employed to provide a cost efficient resource allocation realizing the partitioning of the user request. Numerical results obtained through modeling and simulation demonstrate the efficiency of the proposed approach.The proposed algorithm is compared against an exact method that provides minimal cost partitioning. Detailed evaluations have shown a great cost efficiency over a large number of VN request instances and networked cloud sizes, with minimum computation time. Following VN partitioning, a sophisticated intra VNE algorithm based on linear programming is utilized so that each sub-VN is properly allocated in the selected cloud [11]. The objective is to minimize the cost of embedding a request by performing load balancing and enhancing the cloud scalability [13]. Finally, it should be noted that resource provisioning cost has been define in this article based on node scarcity and average utilization. Thus, utilization in this case depends linearly on the computing resources. An interesting approach would be to derive utilization based on more enhanced queuing models [13], that could better reflect potential non-linear behaviors.

REFERENCES

1. Abdul-Razaq, T.S., C.N. Potts and L.N. Van Wassenhove, 1990. A survey of algorithms for the single machine total weighted tardiness scheduling problem. Discrete Applied Mathematics, 26: 235-253, CrossRef
2. Baum, E.B., 1986. Towards practical “neural” computation for combinatorial optimization problems. In J. Denker, editor, Neural Networks for Computing, pp: 53-64, AIP conference proceedings.
3. Congram, R.K., C.N. Potts and S.L. Van de Velde, 2000. An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem. INFORMS Journal on Computing, to Appear.
4. Crauwels, H.A.J., C.N. Potts and L.N. Van Wassenhove, 1998. Local search heuristics for the single machine total weighted tardiness scheduling problem. INFORMS Journal on Computing, 10(3): 341-350. CrossRef
5. Den Besten, M., T. Stützle and M. Dorigo, 2000. Ant colony optimization for the total weighted tardiness problem. In M. Schoenauer et al., editor, Proceedings of PPSN-VI, 1917 of LNCS, pp: 611-620. Springer Verlag, Berlin, Germany.
6. Hong, I., A.B. Kahng and B.R. Moon, 1997. Improved large-step Markov chain variants for the symmetric TSP. Journal of Heuristics, 3(1): 63-81. CrossRef
7. Hoos, H.H. and T. Stützle, 1998. Evaluating Las Vegas algorithms - pitfalls and remedies. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence 1998, pp: 238-245. Morgan Kaufmann Publishers.
8. Kreipl, S., 2000. A large step random walk for minimizing total weighted tardiness in a job shop. Journal of Scheduling, 3(3): 125-138. CrossRef
9. Lenstra, J.K., A.H.G. Rinnooy Kan and P. Brucker, 1977. Complexity of machine scheduling problem. In P.L. Hammer *et al*., editor, Studies in Integer Programming, volume 1 of Annals of Discrete Mathematics, pp: 343-362. North-Holland, Amsterdam, NL, CrossRef
10. Lourenço, H.R., 1995. Job-shop scheduling: Computational study of local search and large-step optimization methods. European Journal of Operational Research, 83: 347-364. CrossRef

11. Lourenço, H.R. and M. Zwijnenburg, 1996. Combining the large-step optimization with tabu-search: Application to the job-shop scheduling problem. In I.H. Osman and J.P. Kelly, editors, Meta-Heuristics: Theory and Applications, pp: 219-236. Kluwer.
12. Udayakumar, R., V. Khanna, T. Saravanan and G. Saritha, 2013. Retinal Image Analysis Using Curvelet Transform and Multistructure Elements Morphology by Reconstruction, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 16(12): 1798-1800.
13. Udayakumar, R., V. Khanna, T. Saravanan and G. Saritha, 2013. Cross Layer Optimization for Wireless Network (Wimax), Middle-East Journal of Scientific Res., ISSN: 1990-9233, 16(12): 1786-1789.
14. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. A frame work for modelling task coordination in Multi-agent system, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 15(12): 1851-1856.
15. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. An Integrated Agent System for E-mail Coordination using Jade, Indian Journal of Science and Technology, ISSN: 0974-6846, 6(6): 4758-4761.