Indian Journal of Science and Technology, *Vol 8(15), IPL040, July 2015*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Backtracking Algorithm for Virtual Cluster Migration in Cloud Computing

T. Lavanya Suja[1*] and V. Savithri[2]

[1]Mother Teresa Women's University, Kodaikanal - 624102, Tamil Nadu, India; lavanyasuat@gmail.com
[2]Women's Christian College, Nungambakkam, Chennai - 600006, Tamil Nadu, India; dr.savithri.v@gmail.com

Abstract

Objectives: The aim of the paper is to propose a novel approach for Virtual Cluster (VC) migration in cloud computing which achieves better performance in less time and incorporating all Service Level Agreements. **Method:** Cloud Computing (CC) is rapidly gaining momentum and has spread its flavors and dimensions in wide discipline of Information Technology (IT) industry. When a new technology is spreading its wings to various places the functionalities realized by it throws challenges too. In an attempt to tap the whole potential of CC these challenges are addresses by professionals from IT industry and academia. Out of the myriad number of challenges, one is the need for migration of Virtual Clusters (VC) from one physical machine to another physical machine in spite of fault tolerance, load balancing, complying with Service Level Agreements (SLA). After studying and analyzing the various existing approaches for VC migration it is proposed that Backtracking algorithm outperforms the existing ones in terms of time and performance. **Findings:** The implementation of Backtracking algorithm proves the fact that the above objectives are achieved and the test results through line graphs below clarifies the fact further. The novelty of the algorithm and the compact lines of coding achieve the optimal solution for VC migration with satisfying SLA and time constraints. The comparison chart on feature analysis clearly list down the additional characteristics of the backtracking algorithm which adds value and uniqueness to the optimal solution. The flowchart aids to the development of algorithm and its easy implementation. The test data is run for various memory sizes of VC and VM therefore checked for all type of situations arising at real time. All the solutions are optimal in case of time and efficiency when compared with other exciting techniques like serial and greedy approaches. **Application/Improvement:** Though a handful of approaches are there in CC field for VC migration, not all techniques provides a balance between performance and time while the proposed backtracking algorithm does it with a unique solution. Also it is one of the few algorithms which satisfy all the SLA constraints and one of a kind to accommodate any number of SLA into the algorithm.

Keywords: Backtracking Algorithm, Cloud Computing, PM, VC, Virtual Cluster, VC Migration, VM, Virtual Machines

1. Introduction

In cloud computing (CC) there are three popular services called SPI model[9]. They are

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

The concern of the paper is IaaS where the consumers rent server, storage or network via internet from the provider who is responsible for maintenances of it. There is a Service Level Agreement (SLA) signed between the provider and consumer should be adhered by the former for which the latter pays on a metered basis.

While maintaining the infrastructure of the service and keeping up the SLA the provider has to do

- Load Balancing
- Fault Tolerance
- Migration
- Power up/down machines, etc.

**Author for correspondence*

The machine which is actually present is called a Physical Machine (PM) a highly configured server in which several Virtual Machines (VM) are created on the demand of the consumer. In order for easy maintenance several VMs are grouped to form a Virtual Cluster (VC). The rest of the paper is organized as follows. Section 2 surveys a similar literature and analyzes their working and approaches of the existing approaches. Section 3 gives the proposed approach and its working. Section 4 does the performance analysis of the existing and proposed approaches. The last section 5 concludes and adds the scope for future work.

The authors of [1] proposed a Decentralized approach for VM migration after giving a System and energy model. Their approach works on a Double threshold algorithm for VM migration. There is also another algorithm for selecting the destination for migration. They have implemented in a C#. Net platform and came up with the results and performance analysis saying that theirs is an energy and performance efficient one. In [2] the authors propose a genetic algorithm which schedules VMs to PMs in a best fit strategy. The only dimension under concern is load balancing and so they consider only the cost as a constraint, so the allocation leads to less cost to distribute is selected from the choices. The other constraints like migration time, down time, memory size are not considered is a place for improvement. Our newly proposed Backtracking algorithm takes into account those left out constraints of the above algorithm and hence proves it to be more meaningful and practical for cloud environment. The only fact missing here is that grouping of several VMs into VC. Hence VMs are migrated one by one which is rather time consuming. That is rectified in this newly proposed Backtracking algorithm as VCs are migrated one by one.

A Heuristic based algorithm was devised by[3] in order to achieve minimum migration time and less number of VM migration. It can be seen from their results that there is an increase in SLA performance degradation due to migration than the previous approaches even though they have achieved less average SLA violations. The fact that there is degradation in SLA performance is not good news for both the consumer and provider because that will make the latter to incur more cost and less profit. Also the consumer is less likely to be attracted to that kind of cloud products. Here is where our algorithm comes to the rescue. Several migration techniques were surveyed in[4] and confirm the facts such as virtual clusters showed good scalability, in live migration of heterogeneous clusters, node by node migration is better than cluster by cluster. These are the approaches used in the proposed Backtracking algorithm and hence it is a better technique than the previous ones.

Presenting a novel approach for minimizing the migration time the authors of[5] employs three algorithm for finding out When, Where, Which to migrate. They have come up with the results proving better overall performance but left out the time complexity factor. The point of concern is about the 3 algorithms to be executed for a migration makes it clear the fact that total time taken will be far greater than the similar approaches for VM migration. The proposed Backtracking algorithm in this paper balance both the performance and Time factor better than the former ones. In a survey paper [6] techniques and issues in VM migration are studied and came out with the fact that there is less number of VM migration techniques which are network aware in nature. This gives a factor to be considered while developing algorithm for VM migration. In an attempt to propose a fine tuning algorithm for resource provisioning the authors of [7] try to theorize an improvisation to an already existing Resource Provisioning Manager by introducing another module called Resource Tuner which aids to the job of the former. Their approach tries to match the Quality of Service (QoS) profile of Client with that of Provider and assign a better solution based upon the prevailing conditions. The same approach is recommended for VM migration which is not implemented as such so it is too early to judge the performance of the approach.

In a survey of approaches for placing a VM the authors of[8] list several techniques like Heuristic based, Constraint based, Bin packing, and Stochastic Integer and Genetic approach. After surveying these techniques the authors find out that each approach is suitable for specific conditions and choosing a technique for VM placement is a hard task but important one. So it is proposed in this paper a novel algorithm for VM migration which is not surveyed or employed for the VC migration to make use of its inherent time saving and better performance nature.

2. Proposed Approach

Conducting a wide survey about VC migration and its techniques, we have found out that there is a need for better algorithms for VC migration which achieves energy

efficiency. This will attract more avenues of applications for CC and also pave way for Green Computing. The analysis of various algorithms proved the fact that the proposed one can be better than the existing ones[10]. In the previous research paper we found the backtracking approach has not been tried for VC migration despite the fact that it is a solution provider for popular problems like n-queen and Hamiltonian cycle[11]. Though in paper[12,13] the authors introduced a new technique called "Greedy Method" and proved it to be better than the "Serial Method", it failed to include several important constraints like minimum downtime, better performance etc. A popular algorithm "Backtracking"[14] was found as a potential candidate for this and a rough pseudo code is developed. In contrary to the "Greedy Method" this algorithm incorporates constraints like minimum downtime, selecting VMs from same VCs; hence better performance will be achieved than the former methods like "Serial" and "Greedy" approaches.

2.1 Backtracking Algorithm

2.1.1 Aim

Migrate VCs from one PM to another PM based upon the constraints.

2.1.2 Explicit Constraints

Select VMs from the same VCs.

2.1.3 Implicit Constraints

2.1.3 Implicit Constraints

- Memory size of VM and PM
- Minimum down time
- Increase in performance after migration

2.1.4 Flowchart (Figure 1)

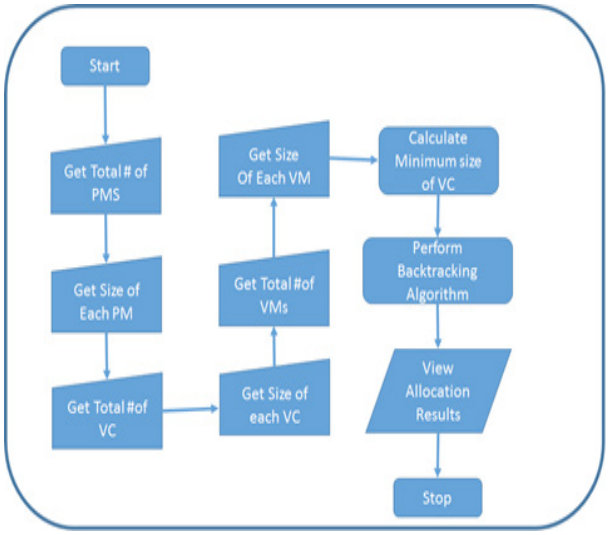

Figure 1. Flowchart for Backtracking Algorithm.

2.1.5 Pseudocode

Procedure VC_Migration( )
Begin
Vc list, pm list arranged in descending order;
VMs arranged in ascending order of down time;
For each PM in pm list do
While size of (PM) >minsize of (vc list) do
For each VC in vc list do
If (size of (VC)>size of (PM)) or (flag(vc) = 'done') then continue; //backtrack
else
For each VM in VC do
Size of (PM) = size of (PM) – size of VM);
Endfor
Endif
Print "VC is in migration to PM"
Flag [vc] = "done";
End for
End while
End for
End

3. Performance Analysis

The above algorithm is optimal when compared to serial and greedy methods (see Table 1) as it includes almost any constraints put forth in SLA. This feature of the proposed Backtracking algorithm makes it different from the existing algorithms.

The proposed backtracking algorithm works fine with all type of inputs and there is no limitation for the number of inputs (Figure 2). The if condition checking is to ensure the explicit constraint i.e. "VMs should belong to same VC for selecting to migrate". The continue statement is used to "Backtracking". By checking the flag of vc we are saving the time and avoiding to check the VCs who have already migrated to PM. After jotting down the test results of total migration time for existing and proposed

Table 1. Feature Analysis

Features → / Algorithm	Less Migration Time	Better Performance	Inclusion of all SLA factors	Adaptive to change	Effective usage of PMs
Powernap[15]	Yes	moderate	No	Difficult	Yes
Greedy[12]	No	No	No	No	Yes
Generic[16]	Yes	moderate	No	Moderate	moderate
Backtracking[11]	Yes	Yes	Yes	Yes	Yes

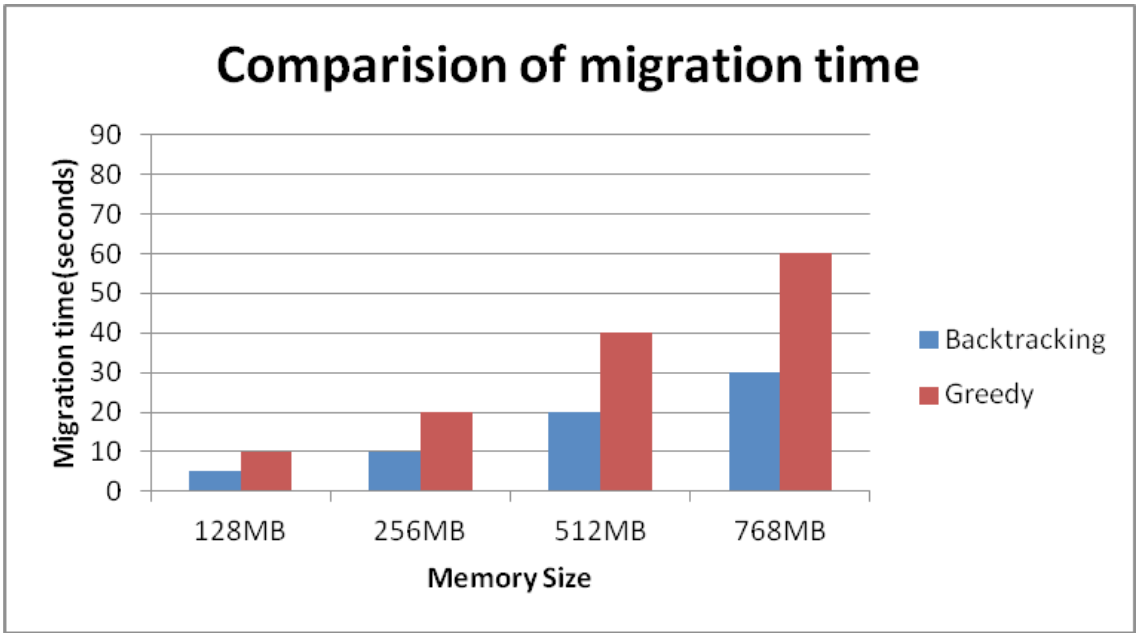

Figure 2. The Total Migration Time in Proposed and Existing Methods.

approach (Figure 3) it is clear that Backtracking consumes very less time than the greedy approach.

3.1 Advantages

After running the program with different kinds of inputs it is experienced by the nature of outputs the positive aspects of the algorithm,

- Efficient usage of PM's memory size
- Optimal assignment of VMs to PMs
- Cost of Implementation is very low
- Easy implementation in any kind of platform as developed in Java8
- Time complexity is less compared to greedy algorithm
- Minimum left over memory size of PM compared to greedy algorithm

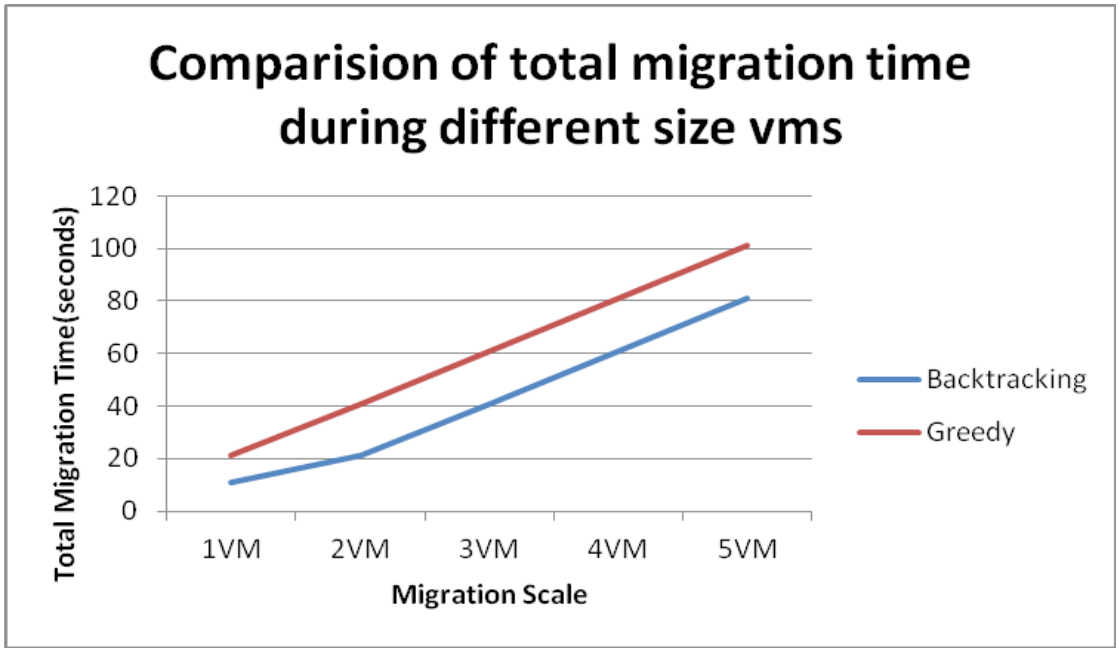

Figure 3. The Total Migration Time in different VM sizes.

4. Conclusion and Future Work

In an attempt to propose a new algorithm we have come up with the coding and found satisfying enough to proceed further, also the manual calculation of test data gave an optimal solution hence the feasibility study proved useful. Proposed algorithm migrates one or more VCs from one PM to another PM after taking into account of criteria in SLA such as minimum downtime, memory size and VMs in same VC. Hence it is concluded from the above results and discussion that the proposed Backtracking algorithm provides better performance, less time than Greedy algorithm, includes all SLA constraints and easy to implement which is evident from Java implementation. Future work includes fine tuning the code for the algorithm to include more implicit constraints and studying the results. We propose to do this by Cloud Sim simulation software and come up with the simulation results in future. In future we would like to implement this Backtracking algorithm with CLOUDSIM a simulating tool under Java platform and successfully come out with results. While working with the algorithm we found that after migration few memory GBs left unused which resembled the problem of internal fragmentation in Paging Memory. Similar solution like Compaction of Memory has to be thought for Consolidation of PMs in our future work.

5. References

1. Wang X, Liu X, Fan L, Jia X. A decentralized virtual machine migration approach of data centers for cloud computing. Mathematical Problems in Engineering. 2013 Jun; 2013:1–10.
2. Gu J, Hu J, Zhao T, Sun G. A new resource scheduling strategy based on genetic algorithm in cloud computing environment. Journal of Computers. 2012 Jan; 7(1):42–52.
3. Mofolo T, Suchithra R. Heuristic based allocation using virtual machine migration: a cloud computing perspective. IRJES. 2013 May; 2(5):40–5.
4. Palta R, Jeet R. Load balancing in the cloud computing using virtual machine migration: A review. International Journal of Application or Innovation in Engineering and Management (IJAIEM). 2014 May; 3(5):437–41.
5. Das L, Shameem PM. Enhancing minimal virtual machine migration in cloud environment. IJRET. 2014 May; 3(7):375–9.
6. Chawda RM, Kale Q. Virtual Machine Migration Techniques in Cloud Environment. IJSRD. 2013; 1(8):1–5.
7. Verma D, Somani RK. Tune the resource provisioning and virtual machine migration for cloud environment. IJESIT. 2014 Nov; 3(6):1–9.
8. Gupta RK, Pateriya RK. Survey on virtual machine placement techniques in cloud computing environment. IJCCSA. 2014 Aug; 4(4):1–7.
9. Sosinsky B. Cloud Computing Bible. 1st edition. USA: Wiley Publishing Inc; 2011.

10. Suja TL, Savithri V. Analyzing a novel approach for virtual cluster migration in cloud computing. IJCOA. 2015 Mar; 4(special issue):1252–6.
11. Suja LT, Savithri V. Feasibility study of Backtracking algorithm for virtual cluster migration in cloud computing. IJCA. 2015 Apr; 116(19):26–30.
12. Baghshahi SS, Jabbehdari S, Sahardabi. VM migration based on greedy algorithm in cloud computing. IJCA. 2014 Jun; 96(12):32–5.
13. Beegom A, Rajasree MS. Resource provisioning and management for IaaS providers in cloud computing. IJCA. 2014 Oct; 104(17):1–4.
14. Horrowitz E, Sahni S. Fundamentals of Algorithms. 3rd editon. USA: Computer Science Press; 1989.
15. Yadav V, Malik P, Chauhan AS. Energy efficient VM optimization. IJCA. 2014 Nov; 106(7):23–8.
16. Beloglazov A, Buyya R. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science. 2010 Nov; 10:1–4.