INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Efficient Data Transfer in Clouds Using Multideployment

Kalpana R.
Department of CSE, PRIST University, Tamil Nadu, India
Bharathi R.
Assistant Professor, Department of CSE, PRIST University, Tamil Nadu, India

Abstract:
This paper aims at finding the shortest path from the multideployed nodes of the available nodes. After finding the shortest path the sender will encrypt the original message and generate the hash key for the encrypted message. The generated hash key for the encrypted message will reach the receiver through the intermediate node. During this transmission processes there may be a chance of hacking the message. If the message is hacked the transmission will dropped and the sender will get the acknowledgement of the hacked process. If it is not, the receiver will successfully get the original message.

Key words: *Virtual Machines, Infrastructure as a service, deployment, multi-path*

1. Introduction

This emerging model leads to new challenges relating to the design and development of IaaS systems. One of the commonly occurring patterns in the operation of Iaas is the need to deploy a large number of VMs on many nodes of a Data center at the same time, starting from a set of VM images previously stored in a persistent fashion. For example, this pattern occurs when the user wants to deploy a virtual cluster that executes a distributed application or a set of environments to support a workflow. We refer to this pattern as multi deployment. Such a large deployment of many VMs at once can take a long time. This problem is particularly acute for VM images used in scientific computing, where image sizes are large.

A typical deployment consists of hundreds or even thousands of such images. Conventional deployment techniques broadcast the images to the nodes before starting the VM instances, a process that can take tens of minutes to hours, not counting the time to boot the operating system itself. This can make the response time of the IaaS installation much longer than acceptable and erase the on-demand benefits of cloud computing.

2. Overview

In addition to incurring significant delays and raising manageability issues, these patterns may also generate high network traffic that interferes with the execution of applications on leased resources and generates high utilization costs for the user. For example, this pattern occurs when the user wants to deploy a virtual cluster that executes a distributed application or a set of environments to support a workflow. The role of virtualization in Clouds is also emphasized by identifying it as a key component. Moreover, Clouds have been defined just as virtualized hardware and software plus the previous monitoring and provisioning technologies. Cloud Computing is a "buzz word" around a wide variety of aspects such as deployment, load balancing, provisioning, data processing and outsourcing. It is not feasible to build a scalable, high- performance distributed data-storage service that facilitates data sharing at large scale.

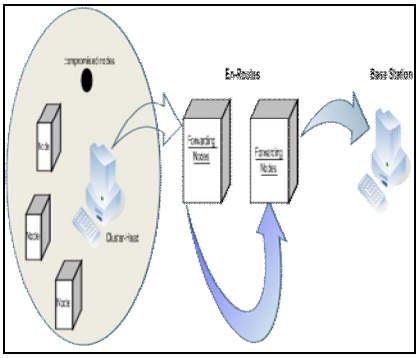

Figure 1: Architecture Model

Cloud computing provides many opportunities for enterprises by offering a range of computing services. It shares massively scalable, elastic resources (e.g., data, calculations, and services) transparently among the users over a massive network. These opportunities, however, don't come without challenges. Cloud computing has opened up a new frontier of challenges by introducing a different type of trust scenario. Today, the problem of trusting cloud computing is a paramount concern for most enterprises. It's not that the enterprises don't trust the cloud providers' intentions; rather, they question cloud computing capabilities. Yet the challenges of trusting cloud computing don't lie entirely in the technology itself.

The projected benefits of cloud computing are very compelling both from a cloud consumer as well as a cloud services provider perspective: ease of deployment of services; low capital expenses and constant operational expenses leading to variable pricing schemes and reduced opportunity costs; leveraging the economies of scale for both services providers and users of the cloud.

3. Related Work

This project proposes a distributed virtual file system specifically optimized for multideployment pattern. Since the patterns are complementary, we investigate them in conjunction. Our proposal offers a good balance between performance, storage space, and network traffic consumption, This introduce a series of design principles that optimize multideployment patterns and describe how our design can be integrated with IaaS infrastructures.

The project shows the design principles by building a virtual file system that leverages versioning-based distributed storage services. To illustrate this point, we describe an implementation on top of Blob Seer, a versioning storage service specifically designed for high throughput under concurrency.

To evaluate our approach in a series of experiments, each conducted on hundreds of nodes provisioned on the Grid'5000 tested, using both synthetic traces and real-life applications. Multideployment or Multipath, Aware of attacks by receiving the acknowledgement, Avoiding data loss, High Encryption methods.

4. Proposed System

In this paper, a multi-path hybrid routing algorithm is proposed that reduces the total shortest path tree (SPT) execution time using the multi-path information; the proposed algorithm is called the Multi-Path Hybrid Shortest Path Tree (MP-HSPT) and is based on the HSPT. In order to efficiently compute the shortest paths using the HSPT, the times when the static and dynamic algorithms are applied are very important. The static routing algorithms should be applied when computing the shortest paths where some links have new weights near the root node; static routing algorithms are applied in this situation because there are many nodes that must be computed near the root node. In this case, using static routing algorithms is a faster method for computing the shortest paths rather than using the dynamic routing algorithms, which require more computation time for each node. Alternatively, dynamic routing algorithms should be used to compute the shortest paths when some links have new weights near the end node; they are applied in this case because there are only a few nodes that must be computed near the end node. In this case, using dynamic routing algorithms to only re-compute the nodes affected by the old shortest paths is faster than using the static routing algorithms that re-compute every node. In addition, the MP-HSPT uses multi-path information. If other minimum weight paths are found, the algorithm uses the multi-path information to reduce the computation time. If other paths of minimum cost are not found, the algorithm reduces the number of affected nodes using the multi-path information.

Step 1: Use multi-path information
Find the multi-path by using the multi-path information
If multi-path found **then**
Update the routing table
Else
Go to Step 2
End if

Step 2: Find the depth of whole network
Find the network depth by Depth First Search (DFS)
Determine D as 40% depth of whole network
For all nodes **do**
Find the link whose link cost is changed
Calculate the link depth by DFS
If the link depth < D **then**
Go to Step 3
Else
Go to Step 4
End if
End for

Step 3: Static routing
Find shortest paths using static routing
Update the routing table

Step 4: Dynamic routing
Find shortest paths using dynamic routing
Update the routing table

- Pseudo Code for the Proposed MP-HSPT

The multi-path hybrid routing algorithm is performed using the following procedures. First, the multi-path is found using the multi-path information. If a multi-path is found, it is included in the shortest path. If a multi-path is not found, the hybrid routing algorithm is applied. Second, in order to decide which algorithms should be applied, the depth of whole network is found using the Depth First Search (DFS) method. Third, when links with new weights and a depth of less than 40% are found, the static routing algorithms are applied to compute the shortest paths. Finally, when some links with new depth weights of more than 40% are found, the dynamic routing algorithms are applied to compute the shortest paths.

5. Performance Evaluation

The performance of the MP-HSPT is compared with previously published algorithms: the Dijkstra, Dynamic Dijkstra, and HSPT methods. The number of nodes, changed rate of link weights, and deviation of link weights were used as the input parameters in the simulations. The input parameters are presented in Table 1.

Parameters	Values
Number of nodes	50, 100, 150, 200
Changed rate of link weights (%)	100, 200, 300, 400

Table 1: Input Parameters

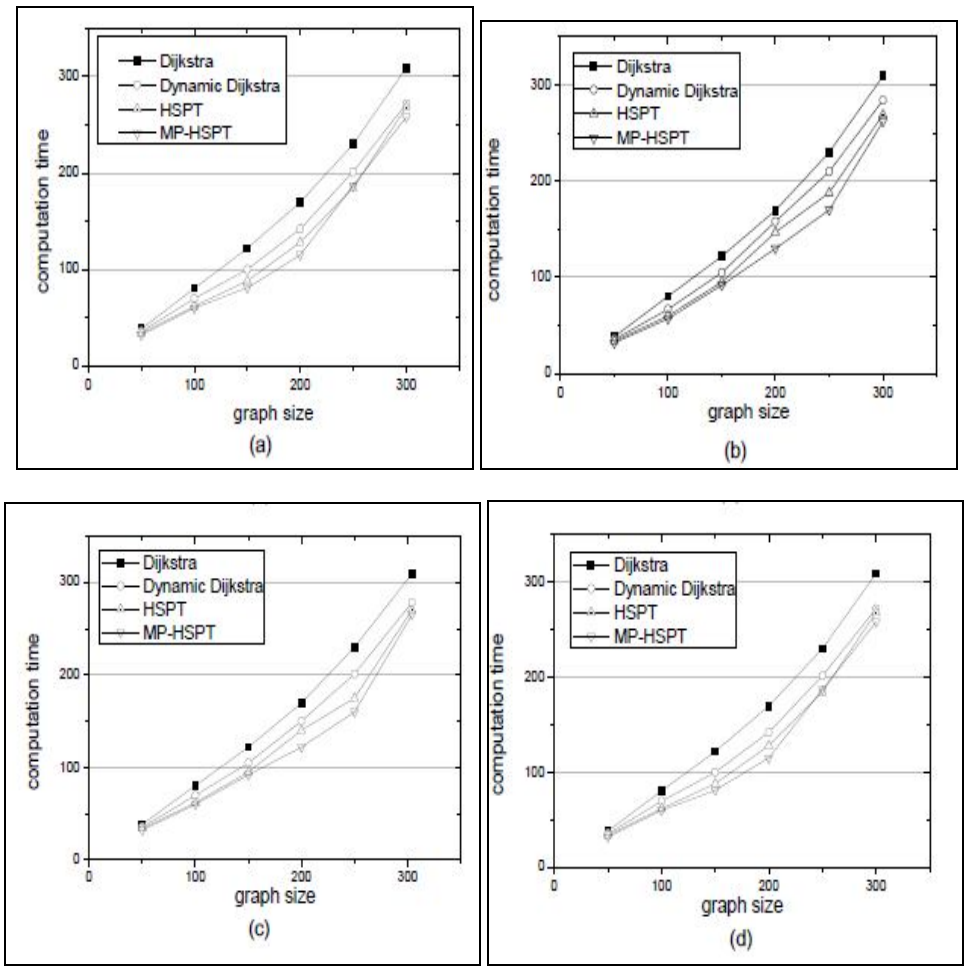

Figure 2: (a, b, c, d)

Triple DES uses a "key bundle" which comprises three DES keys, K_1, K_2 and K_3, each of 56 bits (excluding parity bits).
The Encryption algorithm is:
ciphertext = $E_{K3}(D_{K2}(E_{K1}(\text{plaintext})))$
I.e., DES encrypts with K_1, DES decrypt with K_2, then DES encrypt with K_3.
Decryption is the reverse:
Plaintext = $D_{K1}(E_{K2}(D_{K3}(\text{ciphertext})))$
I.e., decrypt with K_3, encrypt with K_2, then decrypt with K_1.
Each triple encryption encrypts one block of 64 bits of data.
In each case the middle operation is the reverse of the first and last. This improves the strength of the algorithm when using keying option 2, and provides backward compatibility with DES with keying option 3.
The standards define three keying options:

- Keying option 1: All three keys are independent.
- Keying option 2: K_1 and K_2 are independent, and $K_3 = K_1$.
- Keying option 3: All three keys are identical, i.e. $K_1 = K_2 = K_3$.

Keying option 1 is the strongest, with $3 \times 56 = 168$ independent key bits.
Keying option 2 provides less security, with $2 \times 56 = 112$ key bits. This option is stronger than simply DES encrypting twice, e.g. with K_1 and K_2, because it protects against meet-in-the-middle attacks.
Keying option 3 is equivalent to DES, with only 56 key bits. This option provides backward compatibility with DES, because the first and second DES operations cancel out. Each DES key is nominally stored or transmitted as 8 bytes, each of odd parity, so a key bundle requires 24, 16 or 8 bytes, for keying option 1, 2 or 3 respectively.

6. Future Work and Conclusion

In this paper several techniques that integrate with cloud middleware to efficiently handle pattern multideployment. The Multi-Path Hybrid Shortest Path Tree (MP-HSPT) algorithm was presented and it offers an efficient shortest path decision that can be used to reduce the total execution time using the multi-path information. The decreased total execution time also leads to reductions in packet

losses. As shown in the comparison results, the proposed MP-HSPT algorithm provides better performance when compared with the Dijkstra, Dynamic Dijkstra, and HSPT methods in terms of computation time for the shortest path.

7. Acknowledgement
The author would like to thank the anonymous reviewers for their valuable comments and suggestions.

8. References

1. Bogdan Nicolae, John Bresnahan, Kate Keahey, Gabriel Antoniu, "Going Back and Forth: Efficient Multideployment and Multisnapshotting on Clouds", IEEE/ACM CLOUD COMPUTING June 2011 (HPDC 2011).
2. K.Nageswara Reddy. Rapid effective system doubling for cloud computing, et al International Journal of Computer and Electronics Research [Volume 2, Issue 4, August 2013].
3. A Performance Study on the VM Startup Time in the Cloud. Preliminary version. Final version appears in Proceedings of the IEEE CLOUD 2012 5th International Conference on Cloud Computing (Cloud 2012), June 24-29 2012, Honolulu, Hawaii, USA.
4. X. Wu, Z. Shen, R. W and Y. Lin, "Jump-start cloud: efficient deployment framework for large-scale cloud applications," In Proceedings of the 7th International Conference on Distributed Computing and Internet Technology (ICDCIT 11).
5. M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. Commun. ACM, 53:50–58, April 2010.
6. Jinzhu Kong, "A Practical Approach to Improve the Data Privacy of Virtual Machines" 2010 IEEE 10th International Conference on Computer and Information Technology (CIT), June 29 -July 1, 2010, pp. 936-941.
7. Xiao-Yong Li , Li-Tao Zhou ,Yong Shi and Yu Guo, "A trusted computing environment model in cloud architecture", in 2010 International Conference on Machine Learning and Cybernetics (ICMLC), July 2010, Volume 6, pp. 2843-2848