Optimal Construction of Stroage Clusters for the Deployment of Cloud Platforms

Cong Xu
Institute for Network Sciences and Cyberspace
Tsinghua University
Beijing, China
Tsinghua National Laboratory for Information Science and Technology (TNList)
Tsinghua University
Beijing, China

***Abstract*—Improving the deployment efficiency for large-scale cloud computing platforms is critical for the performance of IaaS (Infrastructure-as-a-Service) cloud, especially under heavy workloads and the ever-changing demands of the tenants. Most of the state-of-the-art automatic deployment mechanisms use a single NFS server as the image source node and deploy the application software on the physical clusters based on the "PXE+NFS", which will suffers performance bottleneck and SpoF (Single point of Failure) problem. Some novel deployment mechanisms have also been proposed based on a storage network which consists of multiple NFS servers, however, they seldom consider the impact of storage network topologies on the performance of deployment processes. This paper focuses on optimizing the topologies of storage networks to improve the overall performance of the automatic deployment mechanisms. We formulate the throughput and deployment latency of a storage network under a specific topology, and then design a novel mechanism to optimize the topology of a specific storage network. Experimental results show that our mechanism improve the overall deployment latency on a specific physical cluster dramatically.**

Keywords-cloud computing; cluster; storage network; fast deployment

I. INTRODUCTION

With the rapid development of cloud computing technology and the immense proliferation of cloud-based services, the scales of cloud computing platforms are growing gradually, and deploying cloud system among large-scale physical clusters efficiently has become a challenge. The traditional deployment mechanism for IaaS (Infrastructure-as-a-Service) cloud is to setup system and application software on individual node of the physical cluster and modify relative configurations [1, 2]. However, this deployment manner will induce wasteful duplication of labour and configuration errors. Thus, automatic deployment mechanisms are needed to counteract the problems caused by the traditional deployment mechanism.

Currently, the most widely used automatic deployment mechanisms follow the "PXE+NFS [3]" mode: first deploy an NFS server to store the system image and a PXE server to load the OS images from the NFS server; and then set up the OS software on the destination cluster and finish the deployment of the IaaS cloud. However, following this deployment mode, only OS software has been setup, the complex configuration operations (e.g. configuration of network node and compute node) is still investable. Moreover, most of the state-of-the-art deployment mechanisms use only one NFS server as the image provider, and the performance of the deployment mechanisms are affected by the performance bottleneck and SpoF (Single point of Failure) problems of the single NFS server. To address this problem, some studies propose novel deployment mechanisms based on storage networks [4, 5, 6], which deploy multiple NFS servers and construct a storage network to mitigate the performance bottleneck as well as the SpoF problem caused by the single NFS server. However, to the best of knowledge, the existing studies seldom consider the impact of storage network topology on the performance of automatic deployment mechanism.

This paper focuses on optimizing the topologies of storage networks to improve the overall performance of the automatic deployment mechanisms. We present a performance model that precisely captures the throughput and deployment latency of a storage network under a specific topology. Based on the results of the performance model, we further design a novel mechanism to optimize the topology of a specific storage network and improve the deployment latency on a physical cluster.

II. AUTOMATIC DEPLOYMENT PROCESSES BASED ON STORAGE NETWORKS

The general architecture of a storage network and the deployment process are shown in Fig. 1. All the OS images and software images needed to deploy a cloud platform are stored in the storage network consists of multiple NFS and PXE servers. The PXE servers copy the source images from the NFS servers (Spawning process shown in the figure), and provide root-disk/rootfs for the compute nodes in a physical cluster. The management server in the architecture provides management services to the physical cluster (e.g. PXE service, DHCP service, TFTP service, denoted as *Bootloader and DHCP* in the figure), and meanwhile, the management node is responsible for building source images and uploading the images to the storage network.

This deployment mechanism constructs a storage cluster and enables the diskless deployment mode based on source

images. All the necessary OS images and configuration files are uploaded to the storage cluster by the management server. Each node in the physical clusters is mapped to a node image in the storage network. Supported by the COW (Copy or Write) technique, the image delivering process among the storage clusters is fast, moreover, using a cluster to store the source images, the performance of the deployment mechanism will increase linearly as the scale of the cloud platform.

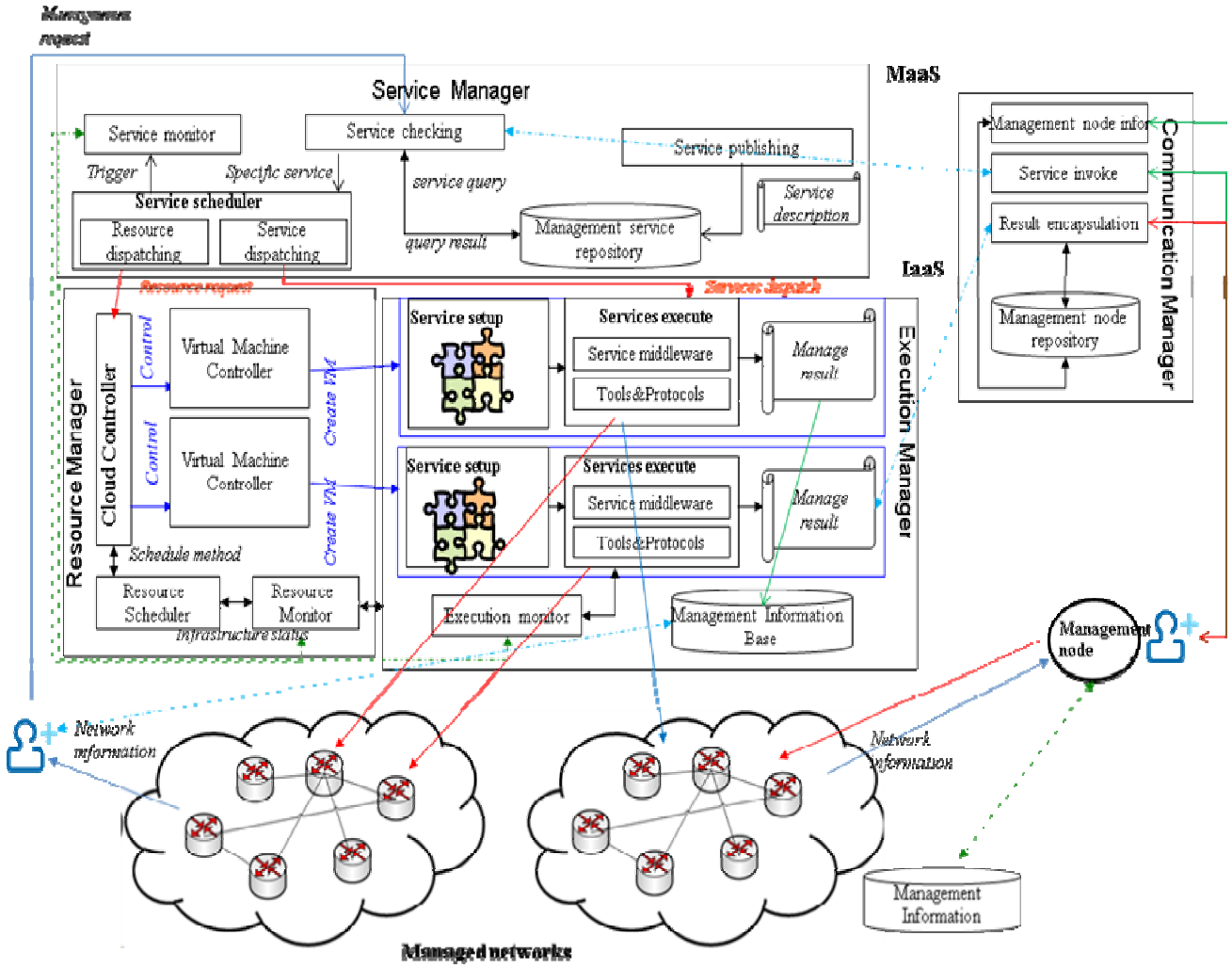

FIGURE I. DEPLOYMENT PROCESS OF IAAS CLOUD PLATFORMS USING STORAGE NETWORK

III. STORAGE NETWORK TOPOLOGY OPTIMIZATION MECHANISM

In this section, we first propose a model to calculate the throughput and deployment latency of a specific storage network. After that we design a novel mechanism to optimize the topology of the storage network. Some important notations and definitions used in the model are illustrated in Table 1.

To provide deployment performance guarantee, the overall throughput of the storage network should exceed the total read/write demand of all the nodes in the cloud platform. Thus, we get:

$$TP_{storage} > \sum_{j=1}^{N_{cloud}} RW_j$$

Suppose the number of image replicas in the storage network is $N_{replica}$, then the overall throughput of the storage network can also be formulated as:

$$TP_{storage} = \sum_{i=1}^{N_{storage}} TP_i / 2N_{replica} \quad (1)$$

The overall deployment process can be divided into 3 sub-phases: spawning process, PXE booting process and node image loading process. Therefore, the total deployment latency T_{total} can be formulated as:

$$T_{total} = T_{spawn} + T_{boot} + T_{load} \quad (2)$$

TABLE I. SUMMARY OF KEY NOTATIONS AND DEFINITIONS

Notations	Definitions
$TP_{storage}$	Overall IO throughput of the storage network
$N_{storage}$	Number of servers in the storage cluster
TP_i	IO throughput of the *i*th server in the storage network
TP_{avg}	Average IO throughput of a storage node
$SP_{storage}$	Overall read/write speed of the storage network
B	Average data transmission speed between two servers
S	Size of source image
S_{node}	Size of node image
$N_{replica}$	Number of source image replica in a storage network
N_{cloud}	Number of servers in the IaaS cloud
RW_j	Read/Write performance demand of node *j* in the IaaS cloud
T_{total}	Total deployment latency
T_{spawn}	Spawning latency of the deployment process
T_{boot}	PXE booting latency of the the deployment process
t^i_{load}	Node image loading latency of the *i*th server in IaaS cloud

In the spawning phase, the total deployment latency is determined by the image size, replica number, bandwidth, as well as the IO performance of each storage server. Specifically, the overall deployment latency of the spawning phase is:

$$T_{spawn} = S \cdot N_{replica} / Min\{B \cdot N_{storage}, TP_{storage}\} \qquad (3)$$

The expression of T_{spawn} is determined by which is the bottleneck performance factor: the data transmission performance between the storage nodes, or the I/O performance of each storage node. Substitute (1) into (3), we get more specific expression of T_{spawn}:

$$T_{spawn} = \begin{cases} \dfrac{S \cdot N_{replica}}{B \cdot N_{storage}}, & if\ N_{storage} \le \sqrt{\sum_{i=1}^{N_{storage}} TP_i / 2B} \\ \dfrac{2S \cdot N_{replica}^2}{\sum_{i=1}^{N_{storage}} TP_i}, & if\ N_{storage} > \sqrt{\sum_{i=1}^{N_{storage}} TP_i / 2B} \end{cases} \qquad (4)$$

In the booting phase, the total deployment latency is determined by the image size as well as the IO performance of each storage server. Thus, T_{boot} can be expressed as:

$$T_{boot} = \frac{S_{boot} \cdot N_{storage}}{TP_{storage}} = 2S_{boot} \cdot N_{storage}^2 / \sum_{i=1}^{N_{storage}} TP_i \qquad (5)$$

Finally, the latency of the node image loading process is determined by the total size of node images and the bandwidth. Hence, we get:

$$T_{load} = \frac{S_{boot} \cdot N_{cloud}}{N_{storage} \cdot B} \qquad (6)$$

For analytic tractability, we assume that the IO performance of a storage server is much better than the transmission performance between storage servers. Substitute (4), (5) and (6) into (2), we get the final expression of T_{total}:

$$\begin{aligned} T_{total} &= T_{spawn} + T_{boot} + T_{load} \\ &= \frac{SN_{replica} + S_{boot}N_{cloud}}{BN_{storage}} + \frac{2S_{boot}N_{storage}^2}{\sum_{i=1}^{N_{storage}} TP_i} \end{aligned}$$

Our goal is to minimize the overall deployment latency while maintain the throughput guarantee of the storage system. Then, the performance optimization model is:

$$Min \quad T_{total} = \frac{SN_{replica} + S_{boot}N_{cloud}}{BN_{storage}} + \frac{2S_{boot}N_{storage}^2}{\sum_{i=1}^{N_{storage}} TP_i}$$

s.t.

$$\begin{cases} TP_{avg} \cdot N_{storage} > \sum_{j=1}^{N_{cloud}} RW_j \cdot 2N_{replica} \\ N_{storage} \le \sqrt{\sum_{i=1}^{N_{storage}} TP_i / 2B} \end{cases}$$

Suppose the physical topology of the storage network satisfies centralized architecture, then our topology optimization mechanism is to determine the optimal number of source image replicas ($N_{replica}$) and the optimal number of PXE servers ($N_{storage}$).

First, we assume that the number of image replicas is fixed, and then the optimal number of PXE services can be calculated by solving the optimization model:

$$\begin{cases} \dfrac{\partial T_{total}}{\partial N_{storage}} = 0, \quad \dfrac{\partial T_{total}}{\partial N_{replica}} = 0 \\ \dfrac{\partial^2 T_{total}}{\partial N_{storage}^2} \cdot \dfrac{\partial^2 T_{total}}{\partial N_{replica}^2} - \left(\dfrac{\partial^2 T_{total}}{\partial N_{storage} \partial N_{replica}}\right)^2 > 0 \end{cases}$$

$$\Rightarrow N_{storage} = \sqrt{\frac{(SN_{replica} + S_{boot}N_{cloud})TP_{avg}}{2SB}}$$

Then, the optimal number of source image replicas can be calculated using the following algorithm.

Algorithm 1 Optimal deployment of storage network

1: **Input**: S S_{boot} N_{cloud}, B $TP_{storage}$ $N_{storage}$
2: **Output**: Optimal number of image replicas: $N_{replica}$
3: Initialize: $N_{\max} \leftarrow TP_{avg} / 2B - S_{boot}N_{cloud} / S$
4: Initialize: $T_{Min} \leftarrow \infty$
for N:= 1 to N_{Max} **do**
Calculate T_{total}
if $T_{total} < T_{Min}$ **then**

Algorithm 1 Optimal deployment of storage network

$T_{Min} = T_{total}$
else break
end for
5: $N_{replica}$:=N-1
6: **return** $N_{replica}$

Our mechanism optimizes the topology of the storage network by determining the optimal number of service replicas and PXE servers in a centralized cluster.

IV. EXPERIMENT

This section constructs a storage network based on our topology optimization mechanism, and shows some experimental results to validate our model. Our storage network is deployed using 25 physical servers with the same configuration (2 Intel(R) Xeon(R) CPU E5-2680 @ 2.70GHz,

64GB Memory, and 1TB disk). Using this storage network, we try to setup OpenStack cloud software (release Havana) on a cluster consists of 50 physical servers to build an IaaS cloud. To validate our mechanism, some other deployment mechanisms are also used as comparisons.

Fig. 2A and 2B show the average IO throughput of a storage server and average in/out traffic between storage servers in the spawning phase respectively. We can see from the Fig. 2A that the storage servers are in little demand of IO performance since most of the operations in this phase are to read/write configuration files. Thus, the throughput guarantee can be ensured. By comparing the results shown in the two figures, we find that the bottleneck performance factor is the transmission performance, which is in accordance with our previous assumption.

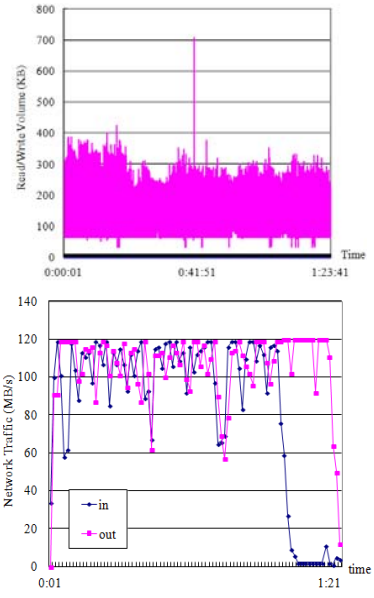

FIGURE II. AVERAGE STORAGE NODE THROUGHPUT AND NETWORK TRAFFIC IN THE SPAWING PROCESS

After that, we deploy IaaS cloud platform of different scales (no more than 50 servers) using this storage network. To validate our mechanism, another two deployment mechanisms are used as comparisons: one is the automatic deployment mechanism based on a single NFS server as aforementioned; the other is Cobbler [7].

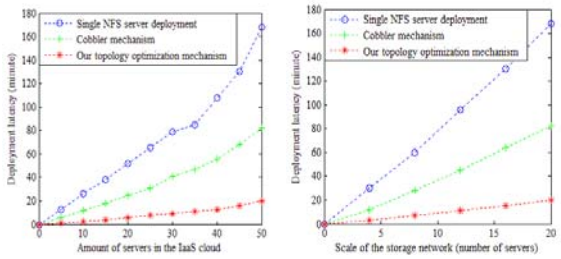

FIGURE III. AVERAGE STORAGE NODE THROUGHPUT AND NETWORK TRAFFIC IN THE SPAWING PROCESS

First, we evaluate the deployment latency by using the same storage network (consists of 20 storage nodes) to deploy different sized IaaS cloud platforms; the experiment results are shown in Fig. 3A. Next we evaluate the deployment latency by deploy the same IaaS cloud (consists of 50 servers) using different sized storage networks; the experiment results are shown in Fig. 3B. Since our model has calculated the optimal replica number based on the scale of the storage network ($N_{storage}$) and the IaaS cloud (N_{cloud}), it improves the deployment latency dramatically, especially when the scale of the physical cluster is large.

V. CONCLUSION

This paper optimizes the topologies of storage networks to improve the overall performance of the automatic deployment mechanisms. We present a performance model that precisely captures the throughput and deployment latency of a storage network under a specific topology. Based on the results of the performance model, we further design a novel mechanism to optimize the number of service replicas and PXE servers in a centralized storage cluster. Experimental results show that our mechanism improve the overall deployment latency on a specific physical cluster dramatically.

REFERENCES

[1] Dong, X., Sun, F., Design and Implementation of Image Based Cluster Deployment System, Computer Engineering, 31(24), pp. 132–134, 2005.

[2] Wu, W., Liu, A., Cheng Y., Fast Deployment and Dynamical Configuration of Large-scale Computer Cluster System, Application Research of Computers, 25(6), pp. 1911-1913, 2008.

[3] Frye, Jr. J. F., Embedded OS PXE server, U.S. Patent No.7085921.1, 2006.

[4] Fiorese, A., Paulo, S., Fernando, B., Assessment of multi-domain network management through P2P, *IEEE ACM* Transactions on networking, 2005.

[5] Gao, C., Yu, H., Shi, G., et al, SMON: Self-Managed Overlay Networks for Managing Distributed Applications. Proc. of *NOMS 2010*, Japan, Apr. 2010.

[6] Wendell, P., Jiang, J. W., Michael. J., DONAR: Decentralized Server Selection for Cloud Services. Proc. of *SIGCOMM 2010*, India, 2010.

[7] Cobbler Manual, http://www.cobblerd.org/manuals/2.4.0/. 2013.6.