2012 IEEE 4th International Conference on Cloud Computing Technology and Science

On P2P Mechanisms for VM Image Distribution in Cloud Data Centers: Modeling, Analysis and Improvement

Di Wu[†], Yupeng Zeng[†], Jian He[†], Yi Liang[†], Yonggang Wen[‡]
[†]*Department of Computer Science, Sun Yat-Sen University, Guangzhou, China*
[‡]*School of Computer Engineering, Nanyang Technological University, Singapore*
Email: {*wudi27@mail, zengyp@mail2, hejian9@mail2, liangyi9@student*}*.sysu.edu.cn, ygwen@ntu.edu.sg*

***Abstract*—To provide elastic cloud services with QoS guarantee, it is essential for cloud data centers to provision virtual machines rapidly according to user requests. Due to bandwidth bottleneck of centralized model, P2P model is recently adopted in data centers to relieve server workload by enabling sharing among VM instances. In this paper, we develop a simple theoretic model to analyze two typical P2P models for VM image distribution, namely, *isolated-image P2P distribution model* and *cross-image P2P distribution model*. We compare their efficiency under different parameter settings and derive their corresponding optimal server bandwidth allocation strategies. In addition, we also propose a practical optimal server bandwidth provisioning algorithm for chunk-level cross-image P2P distribution mechanism to further improve its efficiency. Extensive simulations are conducted to validate the effectiveness of our proposed algorithm.**

***Keywords*-virtual machine (VM), image distribution, cloud data centers**

I. Introduction

Cloud computing is now reshaping the computing landscape by relieving users from the burden of managing their own IT infrastructure. An important feature of cloud computing is the capability of rapid and elastic resource provisioning. In the cloud data centers, different types of resources (such as CPU, memory, storage and bandwidth) should be automatically allocated in a fine granularity to meet the demand from end users timely. To this end, virtual machine (VM) technology is widely adopted by cloud data centers to virtualize different types of resources.

In most of today's cloud data centers, a central image server serves as a repository to host a catalog of diverse VM images. Those images can contain either a bare operating system (e.g., Linux, Windows, etc), or an operating system with additional applications/software (e.g., database). Upon receiving a request, the image server will create a number of VM instances to provide services by distributing requested VM images to multiple physical hosts. Due to the large size of VM images (normally in the order of GBs), the provisioning process of VM instances is time-consuming (e.g., it typically takes up to 10 minutes to provision one VM instance).

To eliminate the bottleneck of centralized model, P2P paradigm is adopted (e.g., [1], [2], [3], etc) to provide scalable and efficient VM image distribution. The basic idea is to allow VM instances started from the same image file to exchange obtained data chunks among themselves, and thus relieve the burden on the image server.

Generally, existing P2P-based solutions can be categorized into two models: 1) *Isolated-image P2P distribution model*, in which VM instances started from different image files form into different isolated swarms. Data exchange is only allowed among VM instances in the same swarm. 2) *Cross-image P2P distribution model*, in which data sharing among different images is allowed. It exploits the observation that different VM image files often have common chunks of data (e.g., different versions of Ubuntu OS). However, there lacks theoretical analysis to show *how those two P2P distribution models perform under different scenarios, how system parameters impact their efficiency*, and *whether it is possible to further improve their efficiency by proper server bandwidth provisioning*.

In this paper, we develop a theoretic framework to formally analyze the above two P2P distribution models, and compare their efficiency for VM image distribution under various situations. To the best of our knowledge, our paper is the first to theoretically analyze the efficiency of P2P distribution models for VM image distribution in cloud data centers. Our analysis shows that cross-image P2P distribution model can significantly reduce the distribution time if properly configured. To further optimize the efficiency of cross-image P2P distribution model, we propose a practical server bandwidth provisioning algorithm to minimize the average distribution time of all VM images.

Our main contribution in this paper can be summarized as below:

- We model and analyze two typical P2P models for VM image distribution in cloud data centers. Our model takes the dynamics of VM instances and image popularity into account. With the above theoretic model, we derive closed-form expressions for the average image distribution time, and show the corresponding optimal server bandwidth provisioning strategy.
- We further consider a more realistic cross-image P2P distribution scenario and propose an optimized server bandwidth provisioning algorithm based on subgradient

method. It enables fine-grain bandwidth allocation to optimize the average distribution time for all images. Our algorithm can be easily integrated into existing VM image distribution models.
- We conduct extensive simulations to validate the effectiveness of our proposed server bandwidth provisioning algorithm. Our simulation results show that our algorithm can greatly reduce the average distribution time for all VM images.

The remainder of this paper is organized as follows: In Section II, we review prior work on VM image distribution models. In Section III, we develop a simple model to analyze the efficiency of two P2P-based VM image distribution models. In Section IV, we propose a practical server bandwidth provisioning algorithm to minimize image distribution time based on subgradient method. In Section V, we conduct extensive simulations to evaluate our proposed algorithm. Finally, we summarize our work in Section VI.

II. Related Work

Rapid provisioning of VM images in data centers has great influence on the performance of cloud-based services. Researchers from both academic and industry area have made significant contribution in this field.

One thread of research is to speedup VM provisioning by reducing the distribution time of VM images. Schmidt et al. [3] discussed several methods for VM image distribution in data centers, including unicast, multicast, binary tree distribution, and BitTorrent-like P2P distribution. Chen et al. [4] proposed a BitTorrent-like P2P protocol to realize fast provisioning of virtual machine images in cloud data centers. Chowdhury et al. [5] presented an architecture called Orchestra that controls both inflow and outflow data transfers to optimize performance. Wartel et al. [2] proposed to use binary tree and BitTorrent-like schemes to distribute VM images among data centers in a flexible and efficient manner. The above mentioned BitTorrent-like distribution mechanisms all belong to the isolated-image P2P distribution model. Peng et al. [1] proposed a chunk-level VM image distribution scheme that enables data sharing among multiple images and takes the hierarchical network topology of data centers into account. Their proposed approach belongs to the cross-image P2P distribution model.

Another thread of research focuses on the reduction of VM initialization time by optimizing OS-related components. Zhu et al. [6] designed a fast provisioning mechanism called Twinkle for VM image initialization. By initializing a virtual machine from partial snapshot, provisioning time can be greatly reduced, especially in the case of flash crowd. Lagar-Cavilla et al. [7] presented the VM fork abstraction called SnowFlock, which supports sub-second VM file cloning and consumes few runtime resources. Shi et al. [8] used the VM image streaming method to decrease the initialization time of VM image distribution.

The research on content caching, source selection and data placement is orthogonal to our work and can be used together with the above techniques. Epstein et al. [9] proposed staging schedules to decrease the VM download time by reasonably placing data on centralized servers. Bjorkqvist et al. [10] proposed a scheme to decrease the retrieval time of VM images by taking both caching and retrieval capacity of the edge nodes and server into consideration.

Our work differs from previous work in that: 1) we formally analyze the distribution time of cross-image P2P distribution model for VM image distribution and compare it with the isolated-image P2P distribution model. Such work hasn't been done by others. 2) we theoretically derive the optimal server bandwidth provisioning strategies for two P2P-based VM image distribution models; 3) we also propose a practical server bandwidth provisioning algorithm to optimize the performance of chunk-level cross-image P2P distribution. Our algorithm is theoretically founded and able to improve the overall distribution performance.

III. Modeling and Analysis of P2P Models for VM Image Distribution

In this section, we develop a simple theoretic framework to formally analyze two typical P2P models for VM image distribution, namely, I) *isolated-image P2P distribution model*, and II) *cross-image P2P distribution model*.

In our framework, we consider a typical VM image distribution scenario, in which requests for cloning VM images arrive continuously. The arrival of VM image requests is assumed to follow a Poisson process, and the total arrival rate is denoted by λ. Suppose that there are K unique VM images hosted by the image server in the data center. The request rate of the i-th VM image file is denoted by λ_i $(i = 1, \cdots, K)$ and satisfies $\sum_{i=1}^{K} \lambda_i = \lambda$. Let F_i be the size of the i-th image file, $i = 1, \cdots, K$. For simplicity, we assume that all image files have the same size F, i.e., $F_i = F, (i = 1, \cdots, K)$. Suppose the upload bandwidth of the VM image server to be u_s. Among u_s, the server bandwidth allocated to distribute the i-th VM image is given by u_s^i, which satisfies $\sum_{i=1}^{K} u_s^i = u_s$. We assume that all the VM instances have the same upload capacity u_c.

Suppose that VM instances started from the same image file (say the i-th image) form into a distribution swarm, denoted by the i-th swarm. The sojourn time of a VM instance in a swarm is determined by the lifetime of that VM instance. Let $1/\gamma_i$ be the average sojourn time of an instance in the i-th swarm, where γ_i can also be interpreted as the departure rate of a VM instance.

Let x_i and y_i be the number of leechers and seeders in the i-th swarm. Here, leechers refer to the VM instances who haven't obtained a complete copy of the VM image, and seeders refer to the VM instances who have already received a full image copy. Denote η_i as the efficiency of data exchange among VM instances in the i-th swarm,

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

IEEE computer society

which represents the fraction of upload bandwidth of a VM instance that can be utilized.

Notation	Description
λ	the total arrival rate of VM requests;
λ_i	the request rate of the i-th VM image file;
K	the number of VM images;
F_i	the size of i-th VM image file;
u_s	the upload capacity of image server;
u_s^i	the server bandwidth allocated to the i-th swarm;
u_c	the upload capacity of a VM instance;
η_i	the efficiency of data exchange in the i-th swarm;
x_i	the number of leechers in i-th swarm;
y_i	the number of seeders in the i-th swarm.

Table I
NOTATION

Notations used in this section are summarized in Table I. Next, we will study different VM image distribution models based on the above model.

A. Isolated-image P2P Distribution Model

We first study the isolated-image P2P distribution model, in which VM instances started from the same image file creates an isolated swarm and data exchange is confined within the swarm. After obtaining the complete copy, the VM instance continues to stay in the swarm for a period until its running time is over. An illustration of the isolated-image P2P distribution model is shown in Figure 1.

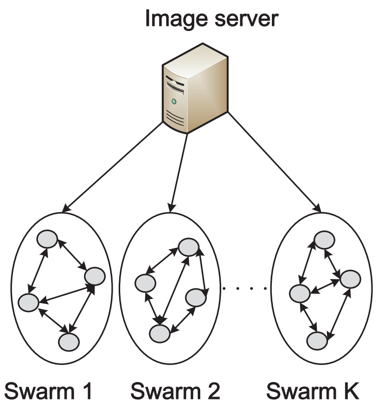


Figure 1. Isolated-image P2P distribution model

Suppose that the image server is responsible for serving the distribution of K VM images simultaneously, then we can use stochastic fluid model in [11] to analyze the image distribution process. The system evolution can be described by the following differential equations:

$$\begin{cases} \dot{x}_i &= \lambda_i - \frac{u_s^i + u_c(\eta_i \cdot x_i + y_i)}{F} \\ \dot{y}_i &= \frac{u_s^i + u_c(\eta_i \cdot x_i + y_i)}{F} - y_i \cdot \gamma_i \end{cases}$$

In the equilibrium state, we have $\dot{x}_i = \dot{y}_i = 0$. Then we can derive the average number of leechers and seeders in the i-th swarm as below:

$$\begin{aligned} \bar{x}_i &= \frac{\lambda_i}{\eta_i}\left(\frac{F}{u_c} - \frac{1}{\gamma_i}\right) - \frac{u_s^i}{\eta_i \cdot u_c} \\ \bar{y}_i &= \lambda_i / \gamma_i \end{aligned} \tag{1}$$

Let T_d^i be the average distribution time in the i-th swarm. By Little's Law, $\lambda_i T_d^i = x_i$, then we can obtain the average distribution time of the i-th image file by:

$$T_d^i = \frac{x_i}{\lambda_i} = \frac{1}{\eta_i} \cdot \left(\frac{F}{u_c} - \frac{1}{\gamma_i} - \frac{u_s^i}{u_c \cdot \lambda_i}\right) \tag{2}$$

From the system's perspective, it is expected to minimize the maximum distribution time of VM image files in all the swarms. Formally, the optimization problem can be stated as follows:

$$\begin{aligned} \min \max_i T_d^i &= \min \max_i \frac{1}{\eta_i} \cdot \left(\frac{F}{u_c} - \frac{1}{\gamma_i} - \frac{u_s^i}{u_c \cdot \lambda_i}\right) \\ \text{subject to:} & \quad \sum_{i=1}^{K} u_i^s = u_s \end{aligned} \tag{3}$$

In the above non-linear optimization problem, the only tunable parameter is u_s^i, which is the amount of server bandwidth allocated to the i-th swarm. By solving the above problem, we have the following theorem:

Theorem III.1 *In the isolated-image P2P distribution model, the optimal server bandwidth provisioning strategy that minimizes the maximum image distribution time in all swarms is to equalize T_d^i. Namely, the server bandwidth allocation $\mathbb{U} =< u_s^i, i = 1, \cdots, K >$ should satisfy the condition as follows:*

$$\frac{1}{\eta_1} \cdot \left(\frac{F}{u_c} - \frac{1}{\gamma_1} - \frac{u_s^1}{u_c \cdot \lambda_1}\right) = \cdots = \frac{1}{\eta_K} \cdot \left(\frac{F}{u_c} - \frac{1}{\gamma_K} - \frac{u_s^K}{u_c \cdot \lambda_K}\right)$$

Proof: First, let $M = \max_i T_d^i = \max_i \frac{1}{\eta_i} \cdot (\frac{F}{u_c} - \frac{1}{\gamma_i} - \frac{u_s^i}{u_c \cdot \lambda_i})$, then the above problem can be transformed to a LP problem:

$$\begin{aligned} \min & \quad M \\ \text{subject to:} & \quad \sum_{i=1}^{K} u_i^s = u_s \\ M \geq T_d^i &= \frac{1}{\eta_i} \cdot \left(\frac{F}{u_c} - \frac{1}{\gamma_i} - \frac{u_s^i}{u_c \cdot \lambda_i}\right), i = 1, ..., K \end{aligned} \tag{4}$$

Then we can solve this problem by using Lagrangian multiplier method,

$$L = M + v\left(\sum_{i=1}^{K} u_i^s - u_s\right) + \sum_{i=1}^{K} \alpha_i (T_d^i - M); \tag{5}$$

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

Based on KKT conditions [12], we know that $\sum_{i=1}^{K} \alpha_i = 1$, $\frac{\alpha_1}{\eta_1 \cdot u_c \cdot \lambda_1} = \frac{\alpha_2}{\eta_2 \cdot u_c \cdot \lambda_2} = ... = \frac{\alpha_i}{\eta_i \cdot u_c \cdot \lambda_i}, i = 1, ..., K$ and all $\alpha_i(T_d^i - M) = 0$, i=1,...,K. From the first two formulae, we know that all $\alpha_i > 0, i = 1, ..., K$, then we have all $T_d^i = M, i = 1, ..., K$. Thus, Theorem III.1 is proved. ■

In the extreme case, if we assume that η_i and γ_i are the same for all swarms, it can be easily proved that the server bandwidth allocated to the i-th swarm should be proportional to the request arrival rate λ_i.

B. Cross-Image P2P Distribution Model

In the cross-image P2P distribution model, VM instances in different swarms are allowed to share common chunks of image files. It is based on the observation that many VM images have a significant fraction of common contents (e.g., Linux OS images in different versions). By allowing VM instances in different swarms to exchange common chunks, it is possible to improve the efficiency of data exchange and better utilize upload bandwidth of VM instances.

To simplify the problem, we assume that all K image files share the same set of chunks with the total size being $F \cdot \beta$, where β is the similarity factor among image files. Each peer joins two swarms to download common chunks and swarm-unique chunks separately (see Figure 2). The common chunks are shared and exchanged among all the peers; but for the swarm-unique chunks, they can only be exchanged with its own swarm. For convenience, the swarm to distribute common chunks is named as **Swarm 0**. A peer allocates θu_c bandwidth for common chunk distribution and $(1-\theta)u_c$ for swarm-unique chunk distribution, where θ is the fraction of upload bandwidth of a VM instance allocated to **Swarm 0**. We analyze the Cross-image P2P distribution model using stochastic fluid model similarly.

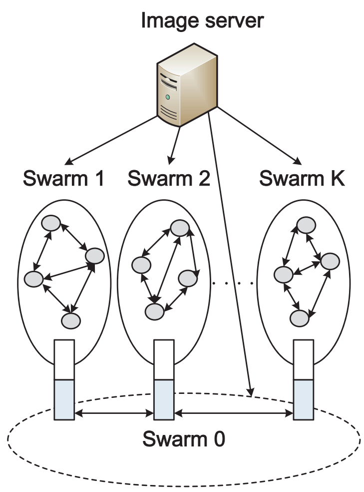


Figure 2. Cross-image P2P distribution model

For **Swarm** i, $(i = 1, \cdots, K)$, the system evolution can be described as follows:

$$\begin{cases} \dot{x_i} &= \lambda_i - \frac{u_s^i + u_c(1-\theta)(\eta_i \cdot x_i + y_i)}{(1-\beta)F} \\ \dot{y_i} &= \frac{u_s^i + u_c(\eta_i \cdot x_i + y_i)}{F} - y_i \cdot \gamma_i \end{cases}$$

Then we can derive the average number of leechers and seeders in the i-th $(i = 1, \cdots, K)$ swarm as below:

$$\begin{cases} \bar{x_i} &= \frac{\lambda_i}{\eta_i} \cdot (\frac{(1-\beta)F}{(1-\theta)u_c} - \frac{1}{\gamma_i}) - \frac{u_s^i}{\eta_i \cdot (1-\theta)u_c} \\ \bar{y_i} &= \lambda_i / \gamma_i \end{cases}$$

For the above equations, we can derive the average distribution time in **Swarm** i, $T_d^i, i = 1, \cdots, K$,

$$T_d^i = \frac{x_i}{\lambda_i} = \frac{1}{\eta_i} \cdot (\frac{(1-\beta)F}{(1-\theta)u_c} - \frac{1}{\gamma_i}) - \frac{u_s^i}{\lambda_i \eta_i \cdot (1-\theta)u_c} \quad (6)$$

For **Swarm** 0, the system evolution can be described as follows. The request arrival rate in Swarm 0 is the sum of all arrival rates in Swarm $1, ..., K$,

$$\begin{cases} \dot{x_0} &= \lambda - \frac{u_s^0 + u_c\theta(\eta_0 \cdot x_0 + y_0)}{\beta F} \\ \dot{y_0} &= \frac{u_s^0 + u_c\theta(\eta_0 \cdot x_0 + y_0)}{\beta F} - y_0 \cdot \gamma_0 \end{cases}$$

Thus, the average distribution time in **Swarm 0** is given by:

$$T_d^0 = \frac{1}{\eta_0} \cdot (\frac{\beta F}{\theta u_c} - \frac{1}{\gamma_0}) - \frac{u_s^0}{\lambda \eta_0 \cdot \theta u_c} \quad (7)$$

For the cross-image P2P distribution model, in order to minimize the distribution time of an image file, we need to consider the distribution process of both common and swarm-unique chunks. The distribution time is determined by the maximum of T_d^i and T_d^0. From the system's view, we need to solve the following min-max optimization problem,

$$\begin{aligned} \min \quad & \max\{T_d^0, T_d^1, ..., T_d^K\} = \\ & \max\{\frac{1}{\eta_0} \cdot (\frac{\beta F}{\theta u_c} - \frac{1}{\gamma_0}) - \frac{u_s^0}{\lambda \eta_0 \cdot \theta u_c}, \cdots, \\ & \frac{1}{\eta_i} \cdot (\frac{(1-\beta)F}{(1-\theta)u_c} - \frac{1}{\gamma_i}) - \frac{u_s^i}{\lambda_i \eta_i \cdot (1-\theta)u_c}\} \\ \text{subject to:} \quad & \sum_{i=0}^{K} u_i^s = u_s \\ & \sum_{i=1}^{K} \lambda_i = \lambda \end{aligned} \quad (8)$$

The above problem is also a nonlinear optimization problem, and the tunable parameters are u_s^i and θ. By solving the above problem, we have the following theorem:

Theorem III.2 *In the cross-image P2P distribution model, the optimal server bandwidth provisioning strategy is to equalize* $T_d^i, i = 0, \cdots, K$.

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

Proof: Using the same method as that in Theorem III.1, we can transform the nonlinear optimization problem into a LP problem, then we can use Lagrangian multiplier and KKT conditions to solve it similarly. ■

In a simplified case, if we assume that η_i and γ_i are the same for all swarms, we just need to adjust θ and u_s^i according to the following equation:

$$\frac{F\beta}{\theta} - \frac{u_s^0}{\theta\lambda} = \frac{F(1-\beta)}{(1-\theta)} - \frac{u_s^i}{(1-\theta)\lambda_i}, \qquad \forall i = 1, \cdots, K \tag{9}$$

Let us further assume $\theta = \beta$, which implies that bandwidth allocation of VM instances is proportional to β, namely, the fraction of common chunks among different image files. We can derive the optimal server bandwidth allocation $u_s^0 = \beta u_s$ and $u_s^i = (1-\beta)\lambda_i u_s/\lambda$.

C. Numerical results

Based on the obtained analytical results, we compare the average distribution time of image files under different P2P models. The server bandwidth is allocated to different swarms according to the corresponding optimal provisioning strategy.

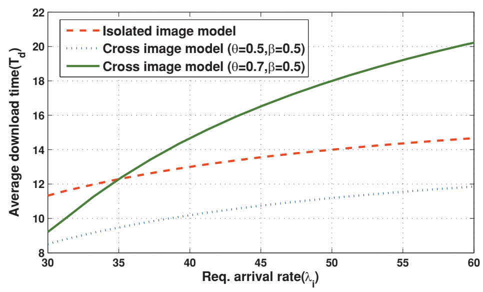


Figure 3. Average distribution time under different request arrival rates

In Figure 3, it is observed that if the fraction of upload bandwidth of a VM instance allocated to **Swarm 0** (namely, θ) is proportional to the fraction of common chunks (namely β), the cross-image P2P model can greatly reduce the average image distribution time under various request arrival rates. However, in case that the value of θ is not properly configured, cross-image P2P model may become worse than isolated-image P2P model when the request arrival rate increases. It is because too much bandwidth is allocated to distribute common chunks while the swarm-unique chunks don't have enough bandwidth for distribution. However, the distribution time of image files is determined by the maximum of the distribution time in all swarms.

In Figure 4, we investigate how the fraction of common chunks (i.e., β) and the fraction of upload bandwidth of a VM instance allocated to distribute common chunks (i.e., θ) impact the efficiency of cross-image P2P model. For different values of β, the average image distribution time

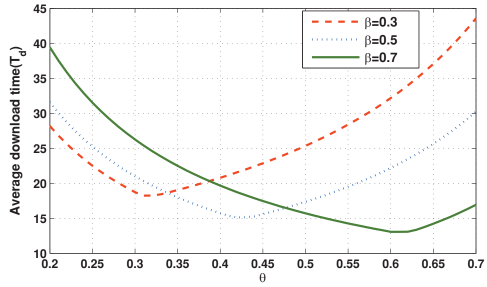


Figure 4. Average distribution time of cross-image model when varying the fraction of common chunks (i.e., β) and the fraction of upload bandwidth of a VM instance allocated to distribute common chunks (i.e., θ)

can be minimized when the value of θ is close to that of β, which is a direct consequence of Theorem III.2.

IV. Practical Server Bandwidth Provisioning Algorithm Towards Optimal Efficiency

In the previous section, we assume that all the VM image files share the same amount of common chunks to facilitate our analysis. In the realistic environments, it is possible that the set of common data chunks between any two VM image files is large, but the intersection among all the VM image files is still small.

In this section, we consider chunk-level cross-image distribution and propose a practical server bandwidth provisioning algorithm to optimize the distribution time for all VM images. We use K_c to denote the number of unique chunks across all the VM image files, each of which forms a distribution swarm. Therefore, there are K_c swarms in the system. The size of each chunk can be set as needed.

Let u_j denote the upload bandwidth of the j-th VM instance, and θ_j^i denote the fraction of upload bandwidth that VM instance j allocates to Swarm i. To track the download progress of VM instance j, we use R_j^i to denote the amount of un-downloaded bytes of the i-th chunk for VM instance j. Let $S(i)$ be the set of VM instances in Swarm i. Thus, the mean remaining time for all peers in Swarm i to complete the download of the i-th chunk, denoted by T_r^i, can be given by the following expression:

$$T_r^i = \frac{\sum_{j \in S(i)} R_j^i}{\sum_{j \in S(i)} u_j * \theta_j^i + u_s^i} \tag{10}$$

In order to minimize the distribution time of all VM images, we need to minimize the maximum of T_r^i, namely, $M = \max\{T_r^1, \cdots, T_r^{K_c}\}$. Then we only need to solve the following optimization problem.

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

$$\begin{aligned} \min \quad & M \\ \text{subject to:} \quad & \sum_{i=0}^{K_c} u_s^i = u_s \\ & M \geq T_i^r, i = 1, 2, ..., K_c \end{aligned} \tag{11}$$

Based on [12], the Lagrangian of the above problem can be derived as below:

$$\begin{aligned} L(\mathbf{M}, \mathbf{u}, \delta) \quad = \quad & M + \delta_0(\sum_{i=1}^{K_c} u_s^i - u_s) + \sum_{i=1}^{K_c} \delta_i(\\ & \frac{\sum_{j \in S(i)} R_j^i}{\sum_{j \in S(i)} u_j * \theta_j^i + u_s^i} - M), \delta_i \geq 0 \end{aligned} \tag{12}$$

With a given δ, we can obtain the optimal value of $\mathbf{M}, \mathbf{u}$ as below:

$$(\mathbf{M}^\star, \mathbf{u}^\star) = \arg \min_{M>0, u>0} L(\mathbf{M}, \mathbf{u}, \delta); \tag{13}$$

By applying duality theory, the original problem can be transformed into the following dual problem:

$$\max g(\delta) = L(\mathbf{M}^\star, \mathbf{u}^\star, \delta) \tag{14}$$

The dual problem can be solved efficiently by using subgradient method as below:

$$\delta_0(t+1) = [\delta_0(t) - \nu(\sum_{i=1}^{K_c} u_s^i - u_s)]^+ \tag{15}$$

$$\delta_i(t+1) = [\delta_i(t) - \nu(\frac{\sum_{j \in S(i)} R_j^i}{\sum_{j \in S(i)} u_j * \theta_j^i + u_s^i}) - M]^+, \quad \delta_i > 0 i = 1, ..., K_c \tag{16}$$

where t is the iteration index, ν is the sufficiently small positive step-size. The details of our proposed algorithm are provided in Algorithm 1.

Algorithm 1 Practical server bandwidth provisioning algorithm for chunk-level cross-image P2P distribution model

Notation:

M: the maximum of $\{T_r^1, \cdots, T_r^K\}$;
K_c: the number of unique chunks;
u_s^i : server bandwidth allocated to the i-th swarm;
$S(i)$: the set of VM instances in Swarm i;
R_j^i: the amount of un-downloaded bytes of the i-th chunk for VM instance j;
u_j: the upload bandwidth of VM instance j;
θ_j^i: the proportion of upload bandwidth that VM instance j allocates to Swarm i;
u_s: the total server bandwidth;
t: the iteration index;

Output:

the optimal server bandwidth provisioning strategy u_s^i.

1: Calculate $(\mathbf{M}^\star, \mathbf{u}^\star)$ as :
2: $(\mathbf{M}^\star, \mathbf{u}^\star) = \arg\min_{M>0,u>0} L(\mathbf{M}, \mathbf{u}, \delta(t)) = M + \delta_0(t)(\sum_{i=1}^{K_c} u_s^i - u_s) + \sum_{i=1}^{K_c} \delta_i(t)(\frac{\sum_{j \in S(i)} R_j^i}{\sum_{j \in S(i)} u_j * \theta_j^i + u_s^i} - M)$
3: Each peer i updates δ to be $\delta_0(t+1) = \delta_0(t) - \nu(\sum_{i=1}^{K_c} u_s^i - u_s)$; $\delta_i(t+1) = \delta_i(t) - \nu(\frac{\sum_{j \in S(i)} R_j^i}{\sum_{j \in S(i)} u_j * \theta_j^i + u_s^i} - M); i = 1, 2, 3, ..., K_c$
4: Set $t \rightarrow t+1$ and go to step 2 (until satisfying termination criterion).

V. Experimental Evaluation

In this section, we conduct extensive simulations to evaluate the performance of our proposed algorithm.

A. Simulation Setup

In default, there are ten VM images to be distributed in our simulations, each of which contains 16 chunks. Each chunk is composed of 64 blocks and the block size is 4Mb. The maximum upload bandwidth of the image server and VM instances is set as 5.12Gbps and 128Mbps respectively. In our experiment, the common content between any two VM image files is generated randomly and accounts from 20% of the image size based on the observations in [13]. Note that the percentage of common content among all image files may be much smaller than 20%. After obtaining the image file, the VM instance continues to run for a random period between 50–100 seconds.

In our experiment, we compare four different distribution models as below:

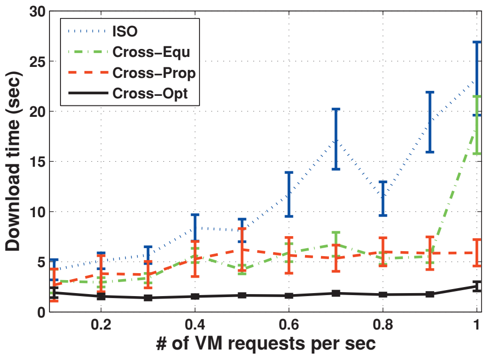
1) *ISO*, which refers to the isolated-image P2P distribution model with the server bandwidth being equally allocated to all image swarms.
2) *Cross-Equ*, which refers to the cross-image P2P distribution model with the server bandwidth being equally allocated to all image swarms.
3) *Cross-Prop*, which refers to the cross-image P2P distribution model with the server bandwidth being proportionally allocated to each image swarm according to its popularity.
4) *Cross-Opt*, which refers to the cross-image P2P distribution model with our proposed server bandwidth provisioning algorithm.

B. Simulation Results

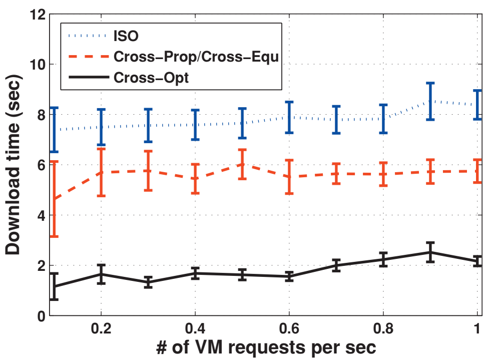
In Fig. 5, we compare the average download time of all image files under different image popularity distribution. In Fig. 5(a), the image popularity follows Zipf distribution. From the figure, it is observed that with the increase of VM request rate, the download time under ISO and Cross-Equ

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

strategies will increase significantly due to improper server bandwidth allocation. On the contrary, the performance under Cross-Prop and Cross-Opt is quite stable, and Cross-Opt outperforms Cross-Prop. The main reason is that under Zipf popularity, hot image files receive more requests, but the server bandwidth allocated to the hot swarm is not sufficient to support efficient distribution when using ISO and Cross-Equ strategies. Similarly, in Fig. 5(b), we consider a scenario in which the image popularity follows uniform distribution. Thus, Cross-Equ and Cross-Prop are actually the same. We also find that our proposed Cross-Opt performs the best in reducing the download time. It is largely because our Cross-Opt algorithm can allocate more server bandwidth to swarms with a longer remaining download time, thus the overall download time of all image files can be reduced.



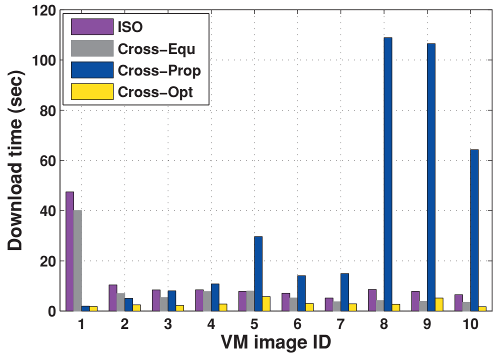
(a) Zipf image popularity



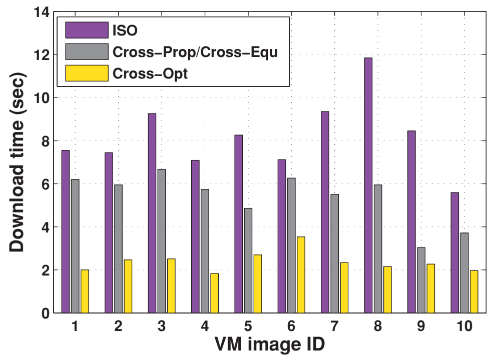
(b) Uniform image popularity

Figure 5. Download time of different VM image distribution models: (a) Zipf image popularity; (b) Uniform image popularity

To better understand the underlying reason, we further plot the average download time of each image file in Fig. 6. In Fig. 6(a), VM images are ranked according to their popularity with VM image 1 being the hottest. For the VM image 1, both ISO and cross-Equ perform much worse than the other two. It is due to insufficient server bandwidth allocated to that swarm. However, we also observe that Cross-Prop is the worst for VM image 8, 9, 10, which are three unpopular images. The intuition is that Cross-Prop allocates too much bandwidth to distribute popular images but sacrifices the performance of unpopular images. Our proposed Cross-Opt achieves stable download time for different image files regardless of their popularity. In Fig. 6(b), the download time under all algorithms is quite stable and our Cross-Opt algorithm still outperforms all the other three algorithms.



(a) Zipf image popularity

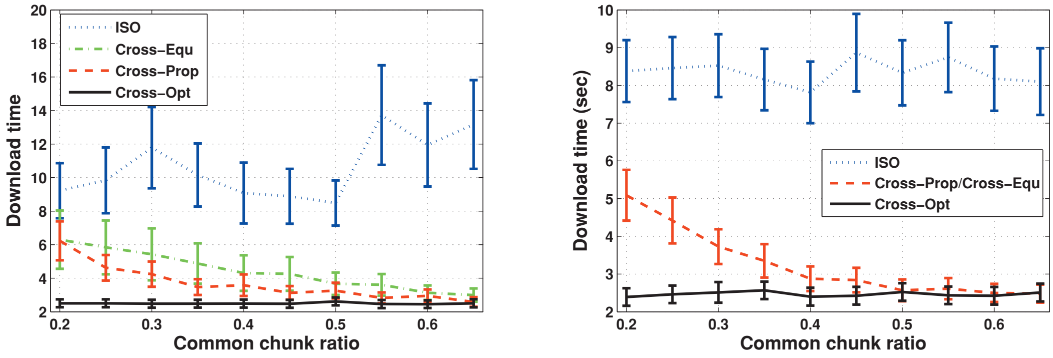


(b) Uniform image popularity

Figure 6. Average download time of each VM image: (a) Zipf image popularity; (b) Uniform image popularity

In Fig. 7, we investigate the impact of the common chunk ratio (namely, the percentage of common chunks between images) on the download time of images. In the experiment, we vary the common chunk ratio from 20% to 90% and compare the average download time under four distribution strategies. For both Zipf and uniform image popularity distribution, we have a similar observation that the gap between Cross-Equ, Cross-Prop and Cross-Opt becomes smaller with the increase of the common chunk ratio. Our proposed Cross-Opt still achieves the lowest download time of all images. It is not difficult to understand the above observation. Under the cross-image P2P distribution model, when the common chunk ratio increases to a higher level, multiple image swarms will look more like a single swarm that exchange the same set of chunks. In this case, the server

978-1-4673-4510-1/12/$31.00 ©2012 IEEE



(a) Zipf image popularity (b) Uniform image popularity

Figure 7. Download time of different common chunk ratios: (a) Zipf image popularity; (b) Uniform image popularity

bandwidth provisioning strategy plays a less important role.

VI. Conclusion

Fast provisioning of virtual machine images in data centers is a challenging problem. In this paper, we developed a simple theoretic model to analyze the efficiency of different P2P mechanisms for VM image distribution and derived the corresponding optimal server bandwidth provisioning strategies. For chunk-level cross-image P2P distribution model, we also proposed a practical server bandwidth provisioning algorithm to optimize the average distribution time of all images. In our future work, we plan to implement our algorithm in the real prototype and conduct more realistic experiments by taking the topology of data centers, image cache placement into account.

Acknowledgment

This work has been supported by NSFC (61003242, 61272397), NSF Guangdong (10451027501005630), Guangzhou Pearl River Sci. & Tech. Rising Star Project (No. 2011J2200086), the Fundamental Research Funds for the Central Universities (12LGPY53), and Program for New Century Excellent Talents in University (NCET-11-0542).

References

[1] C. Peng, M. Kim, Z. Zhang, and H. Lei, "VDN: Virtual Machine Image Distribution Network for Cloud Data Centers," in *IEEE INFOCOM*, 2012.

[2] R. Wartel, T. Cass, B. Moreira, E. Roche, M. Guijarro, S. Goasguen, and U. Schwickerath, "Image distribution mechanisms in large scale cloud providers," in *IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, 2010, pp. 112–117.

[3] M. Schmidt, N. Fallenbeck, M. Smith, and B. Freisleben, "Efficient distribution of virtual machines for cloud computing," in *18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, 2010, pp. 567–574.

[4] Z. Chen, Y. Zhao, X. Miao, Y. Chen, and Q. Wang, "Rapid provisioning of cloud infrastructure leveraging peer-to-peer networks," in *29th IEEE International Conference on Distributed Computing Systems Workshops*, 2009, pp. 324–329.

[5] M. Chowdhury, M. Zaharia, J. Ma, M. Jordan, and I. Stoica, "Managing data transfers in computer clusters with orchestra," *SIGCOMM-Computer Communication Review*, vol. 41, no. 4, p. 98, 2011.

[6] J. Zhu, Z. Jiang, and Z. Xiao, "Twinkle: A fast resource provisioning mechanism for internet services," in *IEEE INFOCOM*. IEEE, 2011, pp. 802–810.

[7] H. Lagar-Cavilla, J. Whitney, A. Scannell, P. Patchin, S. Rumble, E. De Lara, M. Brudno, and M. Satyanarayanan, "Snowflock: rapid virtual machine cloning for cloud computing," in *Proceedings of the 4th ACM European conference on Computer systems*. ACM, 2009, pp. 1–12.

[8] L. Shi, M. Banikazemi, and Q. Wang, "Iceberg: An image streamer for space and time efficient provisioning of virtual machines," in *International Conference on Parallel Processing-Workshops*, 2008, pp. 31–38.

[9] A. Epstein, D. Lorenz, E. Silvera, and I. Shapira, "Virtual appliance content distribution for a global infrastructure cloud service," in *IEEE Conference on Computer Communications (IEEE INFOCOM)*, 2010, pp. 1–9.

[10] M. Bjorkqvist, L. Chen, M. Vukolic, and X. Zhang, "Minimizing retrieval latency for content cloud," in *IEEE Conference on Computer Communications (IEEE INFOCOM)*, 2011, pp. 1080–1088.

[11] D. Qiu and R. Srikant, "Modeling and performance analysis of bittorrent-like peer-to-peer networks," in *ACM annual conference of the Special Interest Group on Data Communication (SIGCOMM)*, 2004.

[12] D. Bertsekas, *Convex optimization theory*. Athena Scientific, 2009.

[13] K. R. Jayaram, C. Peng, Z. Zhang, M. Kim, H. Chen, and H. Lei, "An empirical analysis of similarity in virtual machine images," in *Proceedings of the Middleware 2011 Industry Track Workshop*, ser. Middleware '11, 2011.

978-1-4673-4510-1/12/$31.00 ©2012 IEEE