IEICE TRANS. INF. & SYST., VOL.E98–D, NO.5 MAY 2015

LETTER

A Deduplication-Enabled P2P Protocol for VM Image Distribution

Choonhwa LEE[†], ***Member***, **Sungho KIM**[†], ***and*** **Eunsam KIM**[††a)], ***Nonmembers***

SUMMARY This paper presents a novel peer-to-peer protocol to efficiently distribute virtual machine images in a datacenter. A primary idea of it is to improve the performance of peer-to-peer content delivery by employing deduplication to take advantage of similarity both among and within VM images in cloud datacenters. The efficacy of the proposed scheme is validated through an evaluation that demonstrates substantial performance gains.

key words: *virtual machine, provisioning, image similarity, peer-to-peer, deduplication*

1. Introduction

Cloud computing has been hailed as a great innovation to change the way computing resources and services are provided. According to the new paradigm, computing resources, such as computation, storage, and applications, can now be delivered over the network on an on-demand basis. Whatever resources it intends to provide, a cloud provider must be able to acquire and relinquish computing resources in a timely fashion, as if there exists an unlimited pool of resource instances that can be tapped at any time. While some early offerings have been enjoying growing popularity, their long provisioning latency is considered as a main obstacle to overcome for further cloud acceptance. On requests of IaaS services, a virtual machine instance is launched based on VM images from an image server of the network. This downloading of the images takes the lion's share of the provisioning delay, because the size of the images ranges up to several gigabytes and a datacenter easily scales to several hundreds of thousands of nodes. Therefore, an efficient dissemination scheme of VM images is key to ensuring satisfactory user experiences with IaaS offerings.

The sheer volume of VM provisioning traffic, which can easily overwhelm a datacenter network, calls for an innovative approach. Traditionally, both peer-to-peer content dissemination protocols and deduplication schemes have proven effective in delivering contents over the network [1], [2]. In this paper, we investigate the potential of integrating the P2P delivery and deduplication schemes to speed up virtual machine provisioning, with a focus on the synergetic effects of this integration.

Manuscript received September 5, 2014.
Manuscript revised January 23, 2015.
Manuscript publicized February 19, 2015.

[†]The authors are with the Division of Computer Science and Engineering, Hanyang University, Seoul, 133–791 Korea.
[††]The author is with the Department of Computer Engineering, Hongik University, Seoul, 121–791 Korea.
a) E-mail: eskim@hongik.ac.kr (Corresponding author)

DOI: 10.1587/transinf.2014EDL8180

2. Dedup-Enabled P2P Swarming Protocol

The idea of leveraging P2P swarming protocols for VM image distribution has been around for some time [3], [4]. Unlike those efforts where primary focus is on the adoption itself of P2P BitTorrent-like protocols for data dissemination in the cloud, our dedup-enabled VM provisioning scheme explores a possibility of deduplication combined with peer-to-peer content delivery protocols. Our approach is motivated by the recent studies of VM images for representative cloud offerings that reported a significant degree of similarity at image and block level [5], [6].

Figure 1 illustrates our approach of deduplicated VM image distribution in a peer-to-peer fashion. The scheme is designed to exploit image similarity in order to relieve the distress caused by an avalanche of VM image traffic. First, (1) having followed the usual bootstrapping procedure of BitTorrent-like swarming protocols, (2) peer 1 contacts a tracker also serving as an image server of the datacenter. The tracker replies with a list of peers, participating in the swarm of image A, along with an image similarity matrix. Then, (3) peer 1 figures out from the table that the target image A is 20% similar to image B that it happens to hold in a local disk. Starting off with overlapping of 20 percent, (4) the rest of the image is acquired by engaging itself in block trading with other peers. The protocol benefits from deduplication to eliminate redundant block transfers, while exchanging image blocks among peers, resultantly expediting VM image distribution process. The two components of our proposed scheme, i.e., inter- and intra-image deduplica-

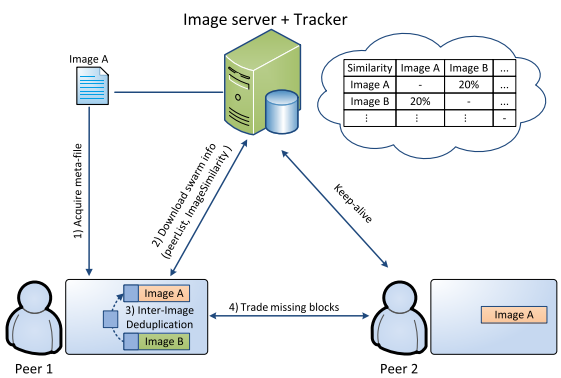

Fig. 1 Dedup-enabled P2P distribution of VM images.

tion, are explained below.

2.1 Inter-Image Similarity Detection

Our cross-image detection scheme makes use of Bloom filters to measure image similarity among related versions. It is noted that a new derivative can be created by installing additional packages on the base OS image. Every time a branched-out version of VM images is added to its repository, the image server updates the image similarity matrix. The new image is chopped into blocks of variable lengths using Rabin fingerprinting scheme, so that each block of the image can be fed into Bloom filter's hash functions. Bloom filters can be used for approximate matching of two sets of image blocks. More specifically, a bitwise logical AND operation of two Bloom filters can be performed to quickly estimate similarity among the contents they represent [7]–[9]. The dot product of two bit vectors can be used for a similarity measure of two VM images being represented by their corresponding filters. The AND operation results in a Bloom filter that approximates the intersection of the two images, i.e., blocks common to both images, enabling fast similarity detection of them.

This inter-image similarity index is stored in the matrix for a later use. Based on the indices, a peer can choose an image in its local storage closest to the target, if any. Common blocks can then be copied to the target image, so that the rest are to be acquired via our deduplication-enabled swarming protocol as described below.

2.2 Intra-Image Deduplication Protocol

Our deduplicated peer-to-peer swarming protocol intends to expedite virtual machine image distribution by eliminating redundant block transfers. So far, deduplication has been used mostly for C/S-based systems where dedup-related state information can be easily maintained in one-to-one fashion [2]. Conventional schemes assume sequentially-ordered transfer of image blocks between the two parties. However, this one-to-one communication model does not fit well to P2P swarming scenarios where blocks are exchanged with multiple peers in a random order. The deduplication scheme must be modified properly for P2P swarming settings.

Deduplication in our dedup-enabled protocol means that block hashes instead of blocks themselves are to be transferred from their second appearance on in the same way as in the conventional deduplication approach. But image blocks are now exchanged in a random order, which means that it is no longer guaranteed that a particular block always precedes its hash appearances. Therefore, block hashes should be replaced with their corresponding blocks at the end of a swarming session, not at the time of their receipt. The idea of our dedup-enabled P2P swarming proposal is illustrated in Fig. 2 where peer 1 trades image blocks with peer 2 and 3. First, peer 1 downloads block hash $h(B)$ for the second instance of block B in the image from peer 2,

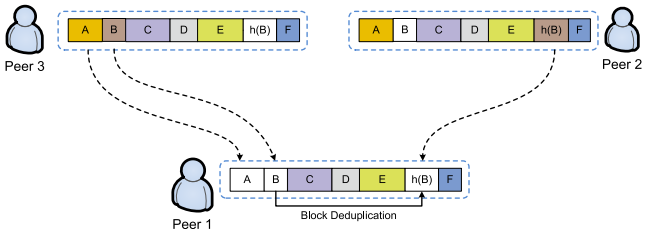

Fig. 2 Deduplication-enabled P2P swarming.

```
protocol Deduplicated-P2P-swarming
  receive(block)   // block = id + hash + data
  vmImage.update(block.getId(), block.getHash())
  if block.isDataPresent() then
      vmImage.write(block.getId(), block.getData())
  end if
  if vmImage.isComplete() then // if download is completed
    foreach block in vmImage do
      if block.isDataMissing() then
          pred = vmImage.getPredBlock(block.getHash())
          vmImage.write(pred.getId(), block.getData())
      end if
    end foreach
  end if
```

Fig. 3 Intra-image deduplication algorithm.

followed by its first instance B from peer 3. When it completes the download process, peer 1 has to scan the image to replace the hash $h(B)$ with its corresponding block B. There could be an incident of hash collisions by which different blocks are hashed to an identical hashed key. Hashes are replaced with their immediately precedent block to ensure the right mappings even in case of hash collisions. A simplified algorithm of our deduplicated peer-to-peer swarming is presented in Fig. 3.

A block message consists of its id, hash, and block data. On receipt of a block message from its neighbor, a peer checks whether there are both block hash and data in the message. First, the hash is recorded at its position of the image pointed to by the id index. Block data, if present, is then copied to its offset within the image, so that the block can be marked as complete. When all blocks are acquired, a scan is performed over the image. In the case of no data for a block, the algorithm looks for its immediate predecessor. The predecessor is defined as a block, with the same hash as the block in question, that appears earlier in the image and has its data part filled. Then, the block's data can be recovered from the predecessor. This way our algorithm can deal with hash collisions where two different blocks are hashed into the same hash value.

3. Performance Evaluation

In order to prove the efficacy of our dedup-enabled P2P swarming protocol, we performed a simulation study

that compares its performance with those of alternative schemes. Our simulator is built on PeerSim P2P simulator (http://peersim.sourceforge.net). The datacenter network for the simulation is structured as a three-layer hierarchy of $32 \times 8 \times 8$ nodes. We conducted an analysis study of popular images publicly available on the Internet (http://www.thoughtpolice.co.uk/vmware) to measure the redundancy rate of the blocks generated using Rabin fingerprinting scheme. It turned out that CentOS images have about 6% of duplicate blocks, while the rate goes up slightly for Ubuntu images. Our simulation study has been performed to use CentOS 5.8 image which is 1.6Gbytes long.

With the inter-image deduplication component disabled for fair comparison, our protocol is compared with alternative image distribution schemes for a datacenter such as unicast, P2P, and VDN. Unicast indicates typical server to client communication patters, while P2P represents a BitTorrent-like delivery protocol. VDN protocol is designed to promote data access locality and minimize redundant block transfers for VM image distribution [10]. A central piece of the protocol is an indexing server that tracks image blocks present at the descendent nodes within the subtree rooted at itself. A node has to update its indexing server on any changes in the set of image blocks it holds. Also, the node queries its index server for meta-information on which node to contact for a specific block. Then, the indexing server redirects the enquirer preferably to one of its descendants that has the requested block. A set of indexing servers are organized in a hierarchical fashion so as to reflect the datacenter network structure. As a result, block transfer traffic and time can be minimized by letting block requests be served by an inner most indexing server and ultimately one of its descendant nodes.

We first looked at average download time which is one of the most important performance indicators for our purpose. As plotted in Fig. 4, the unicast time grows in proportion to the number of virtual machines to be provisioned. For 30 VMs, it rises to 576 seconds which is much longer than 245, 182, and 171 seconds for P2P, VDN, and P2P-Dedup cases, respectively. Our deduplicated swarming protocol surpasses its plain P2P version by 30%. Also, due to the deduplication effect, P2P-Dedup protocol performs better than VDN scheme, which is a very encouraging result considering that VDN case represents a close-to-ideal performance. As shown in the result, Unicast's performance is not acceptable with a large number of VMs. Therefore, the unicast protocol is excluded in the further simulations.

Figure 5 plots control message traffic, i.e., all the traffic except for actual block transfers, for the network to acquire 30 VM images. VDN control traffic includes update and lookup messages for indexing servers. In the cases of P2P and P2P-Dedup, the periodical exchanges of block maps among peers accounts for a major portion of their control overhead. VDN ends up using nearly twice as much control traffic as P2P-Dedup scheme, i.e., 1,739 vs. 910 messages on average. Lesser overhead of P2P-Dedup leads to an edge over VDN in download performance as in Fig. 4.

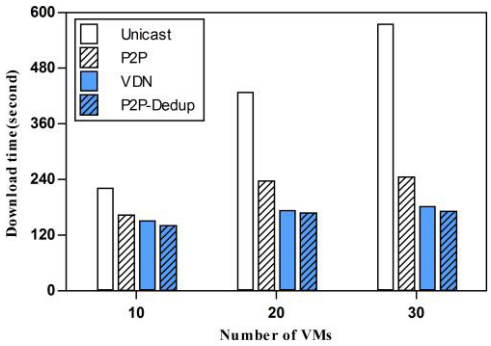

Fig. 4 Average download time.

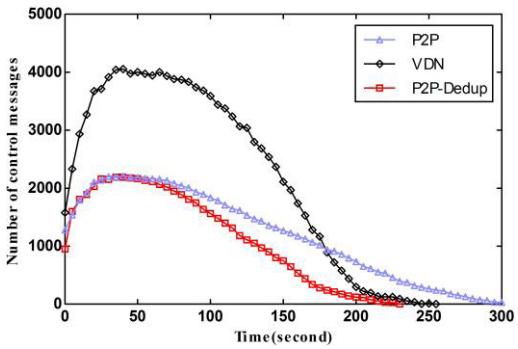

Fig. 5 Control message overhead.

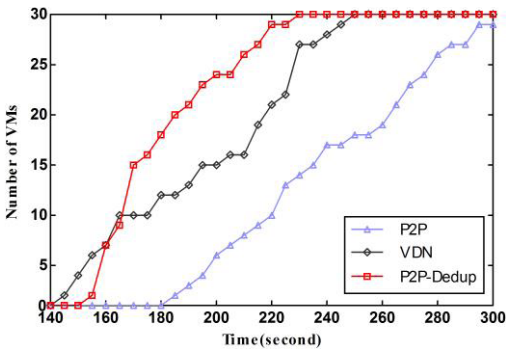

Fig. 6 Donwload progress of 30 VM images.

The outperformance of P2P-Dedup over its plain version can be traced to the eased performance bottleneck at the image server by deduplication and lessened dependency of the rest of the P2P network on the image server for block downloading.

Figure 6 compares the progress of 30 virtual machine image downloads by the three protocols. Being able to complete its first image downloading at around 180 seconds, P2P protocol suffers from the worst performance. VDN scheme has a slight advantage over P2P-Dedup protocol to the moment it acquires first 10 images. This is because some lead-time is required to prime the pump for P2P swarming protocols. More specifically, it takes some seeding time for chunks to propagate across the network so as to become available to peers in the network for swarming. Our P2P-

Dedup protocol finishes its download at 230 seconds, while it takes VDN 250 seconds and P2P 300 seconds.

4. Related Work

An efficient means of content delivery, e.g., VM image distribution, in a datacenter has recently been a keen interest to the cloud research community. Various delivery schemes explored can be categorized into either C/S- or P2P-based protocols. Many prominent efforts in the former group pursue data efficiency by avoiding or minimizing redundant data access or transfer [6], [11]. One notable approach is to perform deduplication with the help of block translation maps generated beforehand when VM images were created, at a higher level than the usual memory or storage device layer [11]. Deduplication integrated in the virtualization layer kicks in even before I/O requests for image blocks are issued, maximizing deduplication gains.

On the other hand, there have also been intensive research efforts on P2P-based schemes [3], [4], [12]. Since an early effort of adopting BitTorrent-like protocols for VM image distribution, the idea develops into further sophisticated schemes like cross-image distribution scheme [4]. According to the cross-image distribution model, image blocks are traded in two swarms: one devoted to blocks unique to the image and the second for blocks of the region shared with other versions of VM images. The work focuses on devising bandwidth allocation schemes of an image server across swarms.

None of the previous efforts do not support deduplication in the context of P2P swarming. It is a central, key piece of our proposed protocol, acting as a performance booster; in addition to the inter-image similarity detection, the protocol is also able to deduplicate redundant blocks in the process of peer-to-peer block swarming.

5. Conclusion

This paper proposes a P2P-based VM image distribution protocol for the cloud which leverages deduplication to speed up virtual machine provisioning. The main novelty of our approach is that deduplication is incorporated in the swarming-based P2P content delivery protocol to be able to further expedite VM image distribution in a datacenter. A simulation study demonstrates that our proposals outperform a plain P2P protocol by 30% in terms of average download time and by 23% in terms of download completion time.

Acknowledgments

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2014-H0301-14-1017) supervised by the NIPA (National IT Industry Promotion Agency) and by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2013R1A1A2007616).

References

[1] C. Ashwin, R. Bharambe, and V. Padmanabhan, "Analyzing and improving a BitTorrent network's performance mechanisms," Proc. IEEE INFOCOM 2006, pp.1–12, April 2006.
[2] A. Muthitacharoen, B. Chen, and D. Mazieres, "A low-bandwidth network file system," Proc. ACM SOSP'01, pp.174–187, Oct. 2001.
[3] J. Reich, O. Laadan, E. Brosh, A. Sherman, V. Misra, J. Nieh, and D. Rubenstein, "VMTorrent: Scalable P2P virtual machine streaming," Proc. ACM CoNEXT, pp.289–300, Dec. 2012.
[4] D. Wu, Y. Zeng, J. He, Y. Liang, and Y. Wen, "On P2P mechanisms for VM image distribution in cloud data centers: Modeling, analysis, and improvement," Proc. IEEE CloudCom, pp.50–57, Dec. 2012.
[5] K.R. Jayaram, C. Peng, Z. Zhang, M. Kim, H. Chen, and H. Lei, "An empirical analysis of similarity in virtual machine images," Proc. ACM/IFIP/USENIX Middleware, doi: 10.1145/2090181.2090187, Dec. 2011.
[6] S. Bazarbayev, M. Hiltunen, K. Joshi, W.H. Sanders, and R. Schlichting, "Content-based scheduling of virtual machines (VMs) in the cloud," Proc. IEEE ICDCS 2013, pp.93–101, July 2013.
[7] A. Broder and M. Mitzenmacher, "Network Applications of Bloom Filters: A Survey," Internet Mathematics, vol.1, no.4, pp.485–509, 2005.
[8] N. Jain, M. Dahlin, and R. Tewari, "TAPER: Tiered approach for eliminating redundancy in replica synchronization," Proc. USENIX FAST 2005, pp.281–294, Dec. 2005.
[9] G. Koloniari, Y. Petrakis, and E. Pitoura, "Content-based overlay networks for XML peers based on multi-level Bloom filters," Lect. Notes Comput. Sci., vol.2944, pp.232–247, 2004.
[10] C. Peng, M. Kim, Z. Zhang, and H. Lei, "VDN: Virtual machine image distribution network for cloud data centers," Proc. IEEE INFOCOM 2012, pp.181–189, March 2012.
[11] Z. Shen, Z. Zhang, A. Kochut, A. Karve, H. Chen, M. Kimand, and H. Lei, "VMAR: Optimizing I/O performance and resource utilization in the cloud," Proc. ACM/IFIP/USENIX Middleware, pp.183–203, Dec. 2013.
[12] X. Leon, R. Chaabouni, M. Sanchez-Artigas, and P. Garcia-Lopez, "Smart cloud seeding for BitTorrent in datacenters," IEEE Internet Computing, vol.18, no.4, pp.47–54, July-Aug. 2014.