Virtual Disk based Centralized Management for Enterprise Networks

Yuezhi Zhou
Tsinghua University, China

Yaoxue Zhang
Tsinghua University, China

Yinglian Xie
Carnegie Mellon University

ABSTRACT

The rapid advances in hardware, software, and networks have made the management of enterprise network systems an increasingly challenging task. Due to the tight coupling between hardware, software, and data, every one of the hundreds or thousands of PCs that are connected in an enterprise environment has to be administered individually, leading to high Total Cost of Ownership (TCO). We argue that centralized management with distributed, diskless clients, yet centralized repositories of all software and data can reduce the management complexity with reduced software maintenance time, improved system availability, and enhanced security. We instantiate such paradigm with a diskless, thick client based system that supports heterogeneous OSes including Windows—the dominant commodity OS in the current market. The prototype requires no or minimum OS modification, nor application modification. Our initial deployment and experiment results demonstrate that our approach is a feasible and efficient solution for managing enterprise network systems.

Keywords

System management, enterprise networks, virtual disks

1. INTRODUCTION

The advent of desktop computers has greatly enhanced end-user productivity and flexibility by enabling a richer set of applications to execute locally. They have been ubiquitously deployed in enterprise network systems such as universities, corporations, and government organizations, where hundreds or thousands of desktop computers are networked with a few application servers.

This distributed thick-client model, although very successful in computing, has created a set of challenges for enterprise system management such as software maintenance and security. Due to the tight coupling of hardware, software, and data, each individual machine must be installed with OSes and applications, constantly patched or upgraded. The diversity of OSes and applications further increases the complexity of software support and maintenance. Meanwhile, the distributed storage of software and data may not only be lost or corrupted by malicious attacks, but also raise the risks of information leakage and data theft.

Various tools (e.g, Marimba [16]) have been created to ease the management tasks by automatically pushing patches and new software images across the enterprise. However, the existence of local copies of software and data inevitably creates local vulnerability for various errors and malicious attacks, making it difficult to ensure centralized control of distributed clients even with these tools. Once a client is faulty, offline, or infected by malicious attacks, administrators lose control of everything stored on the local host, in which case, manual intervention and local diagnostics have to come in.

We believe a centralized management paradigm with *distributed, diskless clients, yet centralized repositories of all software and data*, can significantly reduce the management complexity of enterprise systems. Without local disks, clients keep no persistent states. Administrators can ensure centralized control of all software and data at a small number of servers, hence effectively addressing various challenges associated with distributed, inconsistent system states.

We present a Transparent Computing model (*TransCom*) that instantiates the centralized management concept with diskless, thick clients for enterprise systems. TransCom decouples software, data, and states from the underlying hardware. TransCom clients perform all the computing tasks without local storage devices. All the required OSes, applications, and data will be located at centralized servers, and streamed to the clients on demand. Our approach is thus different from the traditional thin-client model of centralized management (e.g., [4, 2]), in that it leverages the cheap and powerful computing resources of individual clients, and hence is more scalable for supporting computation intensive applications.

We have developed and deployed a TransCom prototype system. Our key technique is the use of *virtual disks* that simulate physical block-based storage devices using disk image files located on the servers and accessed via network communication. Our system has the following desirable features:

- *Heterogeneous OS support:* TransCom supports heterogeneous OSes with no or minimum OS modification. TransCom clients can flexibly choose to boot the desired operating systems via the same remote OS boot process.
- *User transparency:* The use of virtual disks is transparent to users and applications, and requires no application modification. From the perspective of applications and users, there is no difference between accessing data from virtual disks and accessing data from local hard disks.
- *Flexible software and data sharing:* Our design enables both data and application software sharing. The system and application file sharing is transparent to users.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGCOMM'06 Workshops September 11-15, 2006, Pisa, Italy.
Copyright 2006 ACM 1-59593-417-0/06/0009 ...$5.00.

2. CENTRALIZED MANAGEMENT WITH DISKLESS CLIENTS

We focus on the enterprise computing environments where all computers are connected with a low latency network, for example, a switch-based Ethernet. These computers have restricted usage in a limited local area such as a building. They are administrated within a single domain by dedicated IT professionals. To make our discussion more concrete, we consider the following example enterprise systems:

Educational Classrooms: Consider a high school e-learning classroom, where tens of computers are connected through a local area network. Given various course requirements, students need to use different OSes such as Linux, Windows, and Macintosh, and diverse applications such as office software (e.g., MS office or Open office), image/audio/video editors (e.g., Adobe Photoshop, Adobe Premiere, 3D MAX), and program developing tools (e.g., GCC or Microsoft C#). After each class, the computers should be in a clean and consistent state. Software failures or errors, if any, need to be corrected in time for the systems to be used for following up classes.

Government and Military Organizations: In military environments, availability and security have much higher priorities than other management goals. There should be functioning devices that can access the desired software and data at any time, but only in a limited area such as a specific building. No internal software or data should be available to any devices outside the enterprise network.

Commercial Enterprises: A commercial corporation network typically spans across several departments such as call centers, sales, marketing, accounting/fiance, etc. All the employees may use different types of business applications to finish their jobs. Flexibility and customizable environments are important to support business provisioning in addition to availability and security. The enterprise network system should have both the flexibility to quickly enable the deployment of new applications, and the flexibility of moving employees to different departments as necessary.

2.1 Challenges of Distributed Management

The distributed thick-client model has been very popular in the above example systems by leveraging the computing resources of distributed clients to achieve flexibility and enhanced productivity. Nevertheless, since software and data are distributed across individual clients, it has also created the following management challenges in such environments:

- **Software consistency:** Each machine has a local disk drive to store the required software and data. The tasks of software installation, patching, and upgrading have to be performed on every client to keep a correct, consistent, and up-to-date system state across the enterprise. Tools such as Marimba [16] can help reduce the manual effort of administrators by automatically pushing new software images to distributed clients. But they do not address the consistency problem fundamentally, as clients may fail to respond to these tools due to hardware, software, and user errors, or malicious attacks.

- **Security:** The distribute storage of software and data raises two security risks. The first security risk is associated with malicious attacks such as viruses and worms that target at disrupting the normal functions of individual machines. Once the corresponding client is damaged or compromised, the software and data may be lost or corrupted, requiring expensive distributed data backup and restoration service. A more serious security risk is information leakage and data theft, which is in particular a big threat to the military networks. If sensitive military data are fetched and cached at local disks, they will be potentially available to the public or to the attackers who have access to the machine.

- **Heterogeneous OS and application support:** Multiple types or versions of OSes and applications may need to co-exist, for example, to support educational requirements, or to support both legacy applications and new computing requirements. The diversity of software increases the management complexity. Administrators need accurate knowledge of the correct versions of software to update for each machine. And more sophisticated tools are required to automatically push various software in a heterogeneous environment.

- **Data backup and recovery:** Distributed data backup, although critical, is time consuming and not reliable due to the same difficulties of maintaining software consistency.

- **Flexibility and availability:** As usage scenario changes, for example, supporting new business applications or relocating employees as described in commercial enterprises, administrators need to re-provision and migrate client hosts with new software and data. System installation, software upgrades, and ongoing security threats impact the availability of client hosts and their data, which hampers end-user productivity.

2.2 The Power of Centralized Management

We believe a centralized management paradigm with *distributed, diskless clients, yet centralized repositories of all software and data*, can address the above challenges and significantly reduce the management complexity of enterprise systems. In this paradigm, clients have no local disks and keep no persistent states. All the software and data will be stored in various formats, such as regular files, disk images, or database entries, at one or more centralized servers. Instead of using scripts or tools to distribute management tasks to individual hosts, administrators perform management tasks at only centralized servers. For clarification, the computing tasks can be performed by either the centralized servers or distributed clients, and is less relevant to our discussion in this section.

First, with full control of all software, administrators can enforce OS and application patching and upgrading, by updating the contents in the centralized repositories at the earliest available time. Only a small number of centralized servers have to be managed and maintained, as opposed to tens or hundreds of client machines. The savings of installation and configuration time to ensure software consistency can be significant.

As a comparison, let us assume we would like to install a new OS across an enterprise network with 100 computers. The typical time for installing a new OS at a host is about 30 minutes. With centralized management, administrators can finish the new OS installation by updating the disk image contents in maximum of 1.5-2 hours based on the TransCom prototype system that we have developed (see Section 3). In contrast, to automate the process with distributed management, administrators first need to manually deploy an OS installation agent (e.g., Ghost [11]) at each host, and then broadcast the new OS image to the distributed agents for actual OS installation. Now, suppose the time to manually install an agent takes 5 minutes per host, and 5% of the hosts will need a manual OS installation due to errors. The total time spent across 100 hosts would be 5 min/host $\times$ 100 host $+$ 90 min (configuration, OS image making and broadcasting) $+30$ min/host $\times$ 5 host $=$ 740 min, which is more than 5 times longer than the centralized installation.

Second, since software patching and upgrading can be performed in a more timely fashion, the time window of clients being vulnerable to malicious attacks will be shortened. Thus viruses and worms could have little chance of infecting end user computers. Information leakage will no longer happen and there is no need to scrub client hard drives at the end of their usage life. Although the system security now largely depends on one or more server computers, the management efforts are also more concentrated. Thus the servers can be better maintained. For example, they can be placed in more secure locations to reduce end users' opportunity of introducing attacks into the system.

Third, centralized storage of software potentially opens up great opportunities for sharing OSes and applications to reduce the complexity of managing heterogeneous software. Administrators can install and support only one copy of each software at the centralized repository, available to all users that have permission to use them. There is thus no need to keep track of the detailed configuration knowledge of each host, since users can launch their desired OSes and applications on demand.

Fourth, without the need of transferring data back and forth between clients and servers, centralized data backup and recovery is faster and more reliable by simply preserving and recovering the snapshots of server repository images.

Finally, software migration due to re-provisioning or employee relocation can be realized more easily by replicating disk image files or database entries at centralized repositories. Such process can take place in transparent to the users without the need of shutting down the client hosts. Client hosts can then reboot and use the migrated services after the management tasks are completed, achieving better system availability with reduced machine downtime.

In summary, by ensuring centralized control of software and data, centralized management with diskless clients is a promising solution to address the management challenges associated with distributed, inconsistent system states.

However, with centralized management, many issues of managing software at server repositories still need to be addressed, for example, developing scripts or tools to update OSes and applications across various disk image files or database entries. The centralized storage also creates a number of new challenges, for example, software sharing and customization, access control, and in particular performance isolation and guarantee. For some of these issues, we discuss our solutions to address them in the TransCom prototype system that we will describe in Section 3. For others, we leave them as open questions for future work.

2.3 Thin-client vs. Diskless Thick Client

A popular approach to perform centralized system management is the thin-client computing model, which has been deployed both academically and commercially (e.g., [4, 2]). In this model, both the computation and storage functions are performed by centralized servers. A thin client simply uses a remote display protocol to access its computing environment through a terminal. The thin-client model, while simplifying the management tasks, is not scalable by centralizing computing as well. Such performance disadvantage suggests that it cannot efficiently support CPU-intensive multimedia applications as the system scales. The thin-client model is more appropriate for enterprises where end users routinely use a limited number of CPU/memory non-intensive applications, or a small number of users who would like to share powerful server computing resources.

We favor a diskless, thick client model for centralized management in enterprise environments. In this model, clients carry out all computing tasks, but have no local disks. All the required software and data will be located at centralized servers, and streamed to the clients on demand through network communication. By leveraging the cheap and powerful computing resources of individual clients, such model can retain both the merits of centralized management and the system scalability.

Two technology trends suggest that the diskless, thick client model is potentially a high performance yet cost-effective solution. First, both the computing power and network bandwidth are becoming increasingly cheaper, with the prices between a regular desktop computer and a thin client getting closer. Second, disk bandwidth has been increasing with the RAID technology [7, 17]. Although the high performance storage technologies may be expensive to deploy on individual clients today, it is realistic to deploy them on servers by amortizing the costs over clients. These two trends together, may lead to a powerful server with faster I/O access that makes network-based software and data access faster than accessing low performance client local disks.

3. THE TRANSCOM SYSTEM

In this section, we present TransCom—a Transparent Computing system that instantiates the centralized management concept with diskless, thick clients for enterprise systems. In TransCom, a single server supports up to tens of client hosts connected in a network system. Figure 1 shows the high level architecture of a TransCom system. Each client machine is a bare-hardware like computing appliance without any local storage devices. The server can be either a regular desktop computer or a higher-end dedicated machine that stores all the software and data required for completing computing tasks at clients. In our current design, TransCom server uses the MAC address to identify a unique client. The delivery network is thus a local area network protected from other networks by NATs (Network Address Translator) or firewalls for security.

To use a TransCom client, a user just need to power it on for the client to boot remotely and load the selected OS and applications from a server. After this remote OS boot process, the user can access the client in the same way as a regular computer with local storage devices.

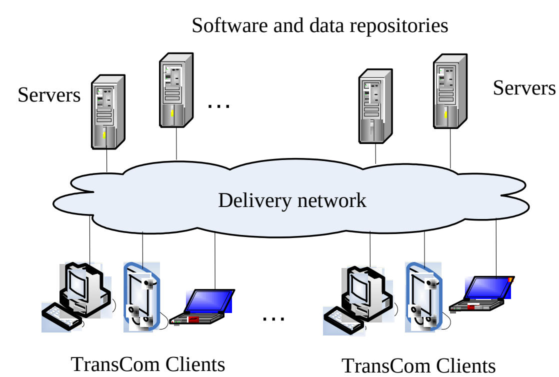

Figure 1: Overall architecture of a TransCom system

An important design goal to enable this software and data streaming is to support conventional, off-the-shelf software. Our key concept for achieving this goal is the use of *virtual disks* in replace of the traditional local disks. Without local disks, each client accesses data from one or more *virtual disks* (Vdisks) that simulate physical block-based storage devices. A Vdisk, in essence, is one or more disk image files located on the server and accessed by the client remotely via a UDP-based virtual disk access protocol. Because disk operations locate beneath file systems and applications, such disk block level virtualization can thus support both OSes and applications with no or minimum modifications.

3.1 System Overview

Figure 2 shows the various components for supporting the use of virtual disks in TransCom, using an example of a single server and a single client system. Access to Vdisks can be generally supported across different operating systems with a Vdisk driver, which is a specialized device driver running on the TransCom client.

TransCom clients issue separate requests for OS boot and disk access. When a client is powered up, it makes a boot request to the TransCom server, and uses a remote boot protocol to first download and enable Vdisk access functions. The client can then issue Vdisk access requests to launch the desired OS from its own virtual disks. During this OS loading process, the Vdisk driver will be loaded in replace of a normal hard disk driver. Further disk access requests will go through the Vdisk driver, and the OS will take control of the system to finish the boot process in a regular way, as if with a local hard disk.

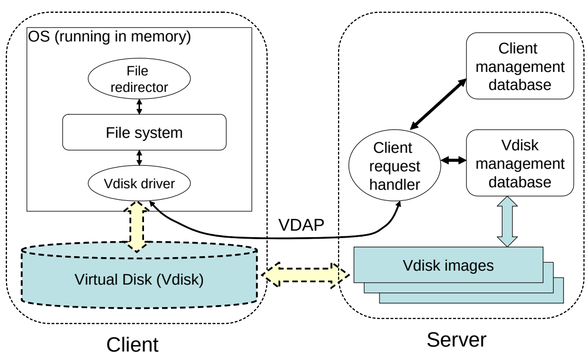

Figure 2: TransCom system modules

Each virtual disk is mapped to a virtual disk image in the server repositories. The virtual image holds the actual disk contents and is the basic management unit. A Vdisk image is created by administrators as an empty image or as a replication of an existing hard disk with software and data. Each image has a number of attributes for management, including image name, type, access mode, and access control list.

A Vdisk seen by users can be mapped to different images. This feature provides flexibility for sharing OS and application software among different users for easy management. Specifically multiple user-perceived Vdisks on different clients can be mapped to the same Vdisk image on the TransCom server. Each client uses a software agent called *file redirector* to redirect the access of files on user-perceived disks to the access of files on server-perceived disks. Eventually, every file access request will be converted into one or more low level disk access requests, which will be handled by the corresponding Vdisk driver by communicating with the server.

The TransCom server is running as an application daemon. It maintains a client management database and a Vdisk management database. Given a disk request, the server first looks up the Vdisk management database to find the corresponding Vdisk images. It then performs the requested operation before sending replies back to the client. For fairness, the TransCom server maintains a request queue per client, and serves queues in a round robin fashion.

3.2 Sharing, Isolation, and Recovery

In order to facilitate managing centralized disk image files and to support heterogeneous OSes and applications with reduced complexity, TransCom classifies client Vdisks into four different categories to enable sharing and isolation. First, TransCom separates software from data based on the observation that many users will use the same OS and application software and thus can share them, while data are often user-specific and cannot be shared.

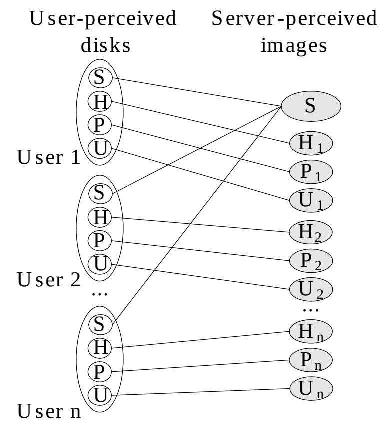

Figure 3: Categories of Vdisks and their virtual images

Second, TransCom maintains a "golden image" of a clean system that contains the desired OS and a common set of applications. This "golden image" is thus immutable and can be shared by all clients. However, some applications must write to the disk directories where they are installed to function properly, e.g., create temporary files. To support these applications, TransCom adopts a copy-on-write (COW) approach by having a COW Vdisk for each client. Figure 3 illustrates this idea by mapping different categories of client Vdisks onto different virtual images at the server:

- *System Vdisk (S):* It is used to store the "golden image" of OS and applications. The corresponding system virtual images are created by the administrator, and shared by all clients. It can be modified by only administrators.
- *Shadow Vdisk (H) :* It is a user-specific COW disk of the system Vdisk to enable write access to the system Vdisk contents. The copy-on-write semantics, however, are supported at the granularity of files through a file redirector, which is a file system level software agent. When a user needs to modify a file on the system Vdisk, a COW copy of the file will be created on the shadow disk for any subsequent access. The use of shadow Vdisks is transparent to the end users.
- *Profile Vdisk (P):* Each client also has a profile Vdisk to store user-specific persistent data such as customized user settings for OS and applications. Similar to shadow Vdisks, the existence of profile Vdisks is also transparent to the users.
- *User Vdisk (U):* Each client has one or more user Vdisks that are used to store private user data. Each user Vdisk will be mapped to a user-specific image.

Our classification of Vdisks can greatly simplify enterprise software management tasks, especially for system recovery. If a client is corrupted by accidental errors, software bugs, or malicious attacks such as viruses, worms, and spyware, system administrators need to simply clear up the corresponding shadow Vdisk or profile Vdisk, and then reboot the client to resume its normal operation.

4. IMPLEMENTATION AND EVALUATION

We have implemented a prototype of TransCom that supports both Windows 2000 Professional and RedFlag 4.1 Desktop (Linux kernel 2.4.26-1) [19]. Our implementation uses the Intel PXE as the remote boot protocol for sending boot requests. Because device drivers are platform dependent, we implemented two different Vdisk drivers, customized for Windows and Linux, respectively. The implementations are in C++. Since Windows 2000 is a modified microkernel, we modified the corresponding Windows Registry files for the OS to load the Vdisk driver. Thus there is no need to change or recompile the kernel. However, Linux is a monolithic kernel. Thus we compiled the Vdisk driver into the kernel by modifying the related kernel source code before recompilation.

4.1 Deployment Experience

Our Windows based system has been deployed across 30 clients in a university e-learning classroom for daily usage for 14 months. In our deployment, TransCom clients are Intel Celeron 1GHz machines, each with 128 MB DDR 133 RAM and 100 Mps onboard network card. The server is an Intel Pentium IV 2.8GHz PC with 1G RAM, an 1Gbps network card, and an 80 GB 7200rpm soft RAID0 hard disk. The clients and the server are connected by an Ethernet switch with 48 100Mbps interfaces (used for clients) and two 1 Gbps interfaces (used for the server). The server OS is Windows 2003 Server (SP1) edition. TransCom clients use Windows 2000 professional (SP2).

During our initial deployment, TransCom has been running stably most of time and has achieved the following benefits:

- *Reduced system maintenance time:* Previously, administrators on average spent 4 to 8 hours a week to regularly clear every machine even with the help of automatic tools to fix problems caused by user faults or malicious attacks. Using TransCom, the system cleaning and upgrading time is reduced to 30 minutes per week, due to both the reduced number of malicious attacks and the centralized operations.

- *Improved availability:* Before using TransCom, the 4-8 hour system maintenance took place every Thursday. No class can be arranged to use the classroom during this maintenance window. After deploying TransCom, the classroom has been in operation without weekly service interruption.

- *Improved security:* After deploying TransCom, there has been no reported viruses or worms in the system. One physical theft happened in the classroom of our deployment, where in addition to the 30 TransCom clients, there are also 30 other regular desktop computers. All the memory slots and hard disk drives of the 30 regular clients were stolen in this incident, resulting significant data loss. As a contrast, all the TransCom clients remained intact, except for one that suffered loss of memory slot after the thief opened this single computer box and discovered no disk. No data were lost, and the TransCom system resumed operating the very next day.

4.2 Testbed Experiments

In our testbed experiments, we used the same hardware configurations as our real deployment but with a more powerful server of AMD Athlon64 3000+ machine. The server is configured with 2 GB Dual DDR 400 RAM, two 80 GB Seagate Barracuda 7200rpm soft RAID0 hard disks, and 1 Gbps onboard network card.

We first vary the number of clients supported by a TransCom server, and compare the performance with a regular PC (with the same hardware configuration, and has an additional local hard disk of 80GB Seagate Barracuda 7200rpm). Table 1 lists the client access latency measured by concurrently running on all clients an operation in the following four categories: OS booting, launching office applications, launching image applications, and file copying. The OS boot latency refers to the time elapsed from powering on the client to the login window displayed on screen. The application launch time refers to the time elapsed from starting the application till it is ready for use.

For all four categories of performance, we observe that TransCom outperforms the regular PC in the single client scenario. The latency increases with the numbers of clients within our range and is on the order of tens of seconds. The worst case latency is for the 20 clients to concurrently copy a 50 MB file, which is a Vdisk intensive operation. Thus the network communication overhead was higher and the server was more likely to become a bottleneck. Such worst case scenario rarely occurs in our real deployment, where we find there is strong locality of disk access patterns by studying the traces collected from the server.

	PC	TransCom client		
		1 client	10 clients	20 clients
OS boot	53.13	48.73	70.62	92.79
MS Word 2003	2.23	1.26	2.28	6.35
MS PPT 2003	5.21	3.04	6.58	9.98
Photoshop V7.0	13.29	11.08	16.48	27.51
Flash V 6.0	18.62	7.16	31.41	74.30
3D MAX V4.0	29.71	25.68	34.24	54.18
Copy a file (20 MB)	11.59	8.95	19.75	37.51
Copy a file (50 MB)	28.24	24.33	49.48	109.52

Table 1: TransCom client access latency (seconds)

We also compared the TransCom performance with the performance achieved by thin-client systems including Citrix [4], Microsoft RDP [10], and VNC [20] using both Web browsing and Video playback applications. In our experiments, TransCom can reduce the client access latency by 2-3 times and improve the video playback quality by 2-20 times compared with the thin-client systems. In particular, we find that when the number of clients is small (≤ 4 in our setup), thin-client systems can achieve lower client latencies than TransCom since the servers perform all computing tasks. However, as the system scales (number of hosts > 4), TransCom achieves almost constant access latency as opposed to the thin-client systems where the client latencies grow linearly. These results suggest that TransCom is a promising cost-effective solution for scalable real world use.

5. RELATED WORK

The power of centralized management has been recognized for long to reduce the cost and complexity by using diskless or thin clients. Despite the overwhelming potential advantages of this architecture, so far none of the existing systems has become a prevalent paradigm in practice. Network computers, such as the Java Station by Sun [12], are proposed as a low-cost desktop terminal. Such solution supports WWW & Java applications only, and does not work with general commodity OSes or other applications such as Microsoft Office. Thin-client systems (e.g., [10, 4, 23, 20, 3]), while very popular, have only limited scalability, with a centralized server performing all the storage and computing tasks.

Our idea of centralizing storage while distributing computing is similar to the concept of diskless computers (e.g., [8, 15, 9]) in early years. A diskless computer usually downloads an OS kernel image from the remote server. It thus cannot support OSes that do not have clear kernel images, such as Windows. In addition, these systems typically use remote file systems such as NFS [21] or AFS [13] as their secondary storage. Without copy-on-write support, they cannot support sharing OSes or applications that need to customize files or directories at fixed locations.

The concept of resource virtualization has been introduced long ago and recently has been adopted to address security, flexibility, and user mobility. Example systems include VMware [24], ISR [14], SoulPad [5], and Collective [6]. These systems virtualize not just storage, but all computing resources using virtual machines, and thus can support heterogeneous OSes as well. On the other hand, they also introduce additional performance overhead, and need to rely on some form of local storages (e.g., a USB disk) for supporting the launch of virtual machines. As a contrast, in TransCom, client OSes are running directly on top of the hardware resources without the need of local storage devices. Such model thus completely eliminates the local client management tasks.

Commercial products such as [1, 22] also support diskless clients for commodity OSes such as Windows. They share similar motivations of TransCom, but do not provide detailed technology or implementation specifics. We believe TransCom will be complementary and applicable to these systems.

6. SUMMARY AND FUTURE WORK

In this paper, we have argued the advantages of centralized management with distributed, diskless clients for today's enterprise network systems. We have developed and deployed TransCom, a virtual-disk based prototype system for enterprise computing and management. In TransCom, clients have no local disks, and all software and data are stored on virtual disks that correspond to disk image files located on a centralized server. By using disk-level remote data access, TransCom supports running heterogeneous OSes including Windows. Our initial experiment results and real world deployment have suggested that TransCom is a feasible and cost-effective solution for managing enterprise network systems.

Future work includes supporting a broader range of OSes (e.g., Open Solaris [18]), optimizing TransCom performance, increasing the system robustness, and enhancing the system security with data encryption and user-level access control.

7. ACKNOWLEDGEMENTS

We would like to thank the following people for their contributions to the development and experiments of the TransCom system: Guanfei Guo, Xiaohui Wang, Li Wei, Huajie Yang, Pengzhi Xu, Nan Xia, Yukun Peng, Xingrong Chen, and Yu Chen. Special thanks to Paul Francis for his valuable comments and suggestions. We thank the people and organizations who have deployed TransCom and allowed us to measure the deployed systems.

8. REFERENCES

[1] Ardence Desktop Edition. http://www.ardence.com/enterprise/products.aspx?ID=83.

[2] R. Baratto, L. Kim, and J. Nieh. THINC: A Virtual Display Architecture for Thin-Client Computing. In *Proc. of the Twentieth ACM Symposium on Operating Systems Principles*, 2005.

[3] R. A. Baratto, S. Potter, G. Su, and J. Nieh. MobiDesk: Mobile Virtual Desktop Computing. In *Proc. of the International Conference on Mobile Computing and Networking*, 2004.

[4] I. Boca Research. Citrix ICA Technology Brief, Technical White Paper. Boca Raton, 1999.

[5] R. Caceres, C. Carter, C. Narayanaswami, and M. Raghunath. Reincarnating PCs with Portable SoulPads. In *Proc. of ACM/USENIX MobiSys*, 2005.

[6] R. Chandra, N. Zeldovich, C. Sapuntzakis, and M. S. Lam. The Collective: A Cache-Based Systems Management Architecture. In *Proc. of NSDI*, 2005.

[7] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson. RAID: High-performance, Reliable Secondary Storage. *ACM Computing Surveys*, 26(2):145–185, 1994.

[8] D. R. Cheriton and W. Zwaenepoel. The Distributed V Kernel and its Performance for Diskless Workstations. In *Proc. of the 9th ACM Symposium on Operating Systems Principles*, 1983.

[9] B. Croft and J. Gilmore. Bootstrap Protocol (BOOTP). RFC 951, 1985.

[10] B. Cumberland, G. Carius, and A. Muir. Microsoft Windows NT Server 4.0, Terminal Server Edition: Technical Reference. Microsoft Press, 1999.

[11] Norton Ghost. http://www.symantec.com/home_homeoffice/products/overview.jsp?pcid=br&p%vid=ghost10.

[12] R. G. Herrtwich and T. Kappner. Network Computers – Ubiquitous Computing or Dumb Multimedia? In *Proc. of Third International Symposium on Autonomous Decentralized Systems*, 1997.

[13] J. H. Howard, M. L. Kazar, and S. G. Menees. Scale and Performance in a Distributed File System. *ACM Transactions on Computer Systems*, 6(1):51–81, 1988.

[14] M. Kozuch and M. Satyanarayanan. Internet Suspend/Resume. In *Proc. of the 4th IEEE Workshop Mobile Computing systems and Applications*, 2005.

[15] R. Linlayson. Bootstrap Loading Using TFTP. RFC 906, 1984.

[16] Marimba. http://www.marimba.com.

[17] R. J. T. Morris and B. J. Truskowski. The Evolution of Storage Systems. *IBM Systems Journal*, 42(2):205–217, 2003.

[18] Open Solaris. http://www.opensolaris.org/os/.

[19] RedFlag Linux. http://www.redflag-linux.com/eindex.html.

[20] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper. Virtual Network Computing. *IEEE International Computing*, 2(1):33–38, 1998.

[21] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon. Design and Implementation of the Sun Network Filesystem. In *USENIX Association Conference Proceedings*, 1985.

[22] The SoftGrid Platform. http://www.softricity.com/products/softgrid.asp.

[23] Sun Ray Overview, White Paper, Version 2. http://www.sun.com/sunray/whitepapers.html, 2004.

[24] VMware GSX server. http://www.vmware.com/products/gsx.