Computer Communications 34 (2011) 1539–1548

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Separating computation and storage with storage virtualization

Yaoxue Zhang, Yuezhi Zhou *

Key Laboratory of Pervasive Computing, Ministry of Education, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, People's Republic of China

ARTICLE INFO

Article history:
Available online 3 July 2010

Keywords:
Transparent computing
PC management and security
Virtual storage
Virtual disk

ABSTRACT

Recent advances of hardware, software, and networks have made the management and security issues increasingly challenging in PC usage. Due to the tight coupling of hardware and software, each one of the hundreds or thousands of PCs connected in a networked environment has to be managed and administrated individually, leading to a high Total Cost of Ownership (TCO). We argue that a centralized storage of software and data, while distributed computation in clients, i.e., transparent computing, can address these challenges potentially and reduce the complexity with reduced software maintenance time, improved system availability, and enhanced security.

This paper presents a novel approach, named StoreVirt, to realize transparent computing, which separates computation and storage from inside a single physical machine to different machines with a storage virtualization mechanism. With virtualization, all the OSes, applications, and data of clients are centered on the servers and scheduled on demand to run on different clients in a "block-streaming" way. Therefore, due to the central storage of OSes and applications, the installation, maintenance, and management are also centralized, leaving the clients light-weighted. Further, due to timely patching and upgrading, the system security can be improved. Experimental and real-world experiences demonstrate that this approach is efficient and feasible for real usages.

1. Introduction

The advent and advance of desktop/personal computers has greatly improved end user productivity and flexibility by enabling a richer set of applications to be installed and executed locally. Now, they have been ubiquitously deployed in enterprise network environments (typically LANs), such as universities, corporations, and governmental organizations. However, the great success of the distributed PC has also brought many challenges for system management and security.

Consider a typical scenario of PC usage in educational classrooms, 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 Solaris, and diverse applications such as office software (e.g., MS Office or Open Office), image/audio/video editors (e.g., Adobe Photoshop, Adobe Premiere, 3dMAX), and program developing tools (e.g., Microsoft C# or GCC). Thus, to satisfy these diverse requirements, various types of OSes and applications have to be installed on each PC. Other PC-based systems in governmental organizations or enterprises are similar to the above usage scenario, as illustrated in Fig. 1(1).

In such typical scenarios, although end users can leverage the computation and storage resources of distributed client computers to achieve flexibility and enhanced productivity, there are mainly two categories of challenges: management and security. With respect to management, there are at least two following challenges:

- *Software consistence:* Since each PC has a local hard disk to store all the required software and data, the tasks of installation, patching, and upgrading have to be carried out on every client to keep a correct, consistent, and up-to-date system state across the entire environment. Automatic management tools such as Marimba can help reduce the manual efforts of administrators by automatically pushing new software images or patches to distributed clients. However, as clients may fail to respond to these tools due to hardware, software, and user errors, or malicious attacks, these tools cannot address the consistency problem fundamentally.
- *Heterogeneous OS and application support:* As described in the above scenario, multiple types or versions of OSes and applications may need to co-exist to support educational requirements, or to support legacy applications and other new requirements. This diversity of software further increases the management complexity. Administrators need accurate knowledge of the correct versions of software to update for each machine. Thus more sophisticated tools are required to push packages automatically in a heterogeneous environment.

* Corresponding author. Tel./fax: +86 10 62782118.
E-mail addresses: zyx@moe.edu.cn (Y. Zhang), zhouyz@mail.tsinghua.edu.cn (Y. Zhou).

0140-3664/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.06.024

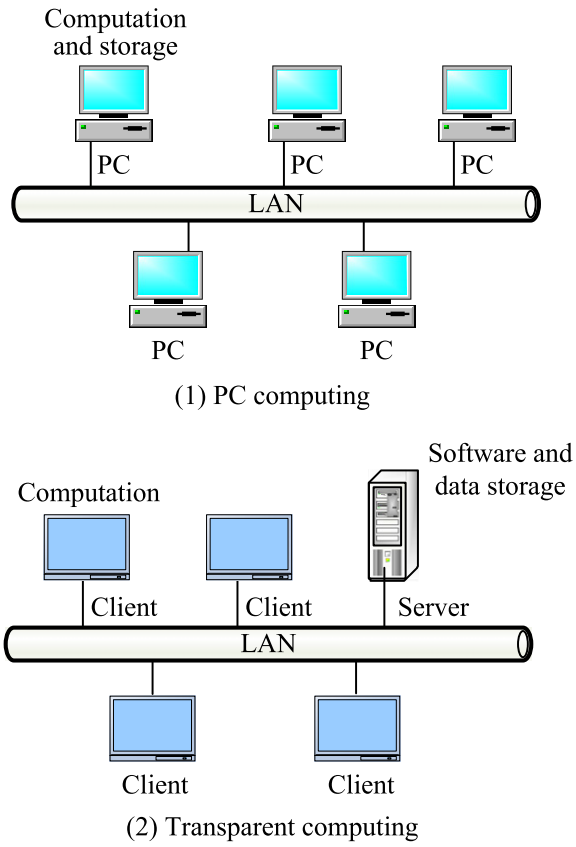

Fig. 1. (1) Each PC machine is installed with all needed software and data and executes them locally. (2) Each client machine does not hold any desired software and data locally, which resides on the central server, while being streamed to and executed with the client's local resources.

The next category of challenges concerns the security issue:

- *Malware threat:* The first security risk is associated with malicious attacks such as virus, worms, spyware, and other malware that target the normal functions of individual machines. Once the corresponding client is compromised or damaged, the installed software and data may be lost or corrupted, requiring expensive distributed backup and restoration services.
- *Data protection:* The second security issue is concerned with the data protection. As the data is distributed in the typical scenario, thus the distributed data backup, is time consuming and not reliable due to the same difficulties as that in maintaining software consistency. A more serious data security risk is information leakage and data theft, which is in particular a big threat to the governmental or military organizations. If sensitive data are fetched and cached at local disks, they will be potentially available to the errant end users or intended attackers who have access to the client machines.

Due to these difficulties and challenges in PC management and security, much money and manpower are involved in dealing with these issues. As estimated in a typical scenario, the annual Total Cost of Ownership (TCO) of a PC has been around five times the purchase cost of the PC [1].

To address these challenges, a variety of approaches have been proposed, which can be classified into two categories: distributed client management tools and centralized computing paradigms.

Various client management tools (e.g., BMC BladeLogic Client Automation [2]) have been produced in the past years. These tools use two sets of software: one installed on the server that help administrators to monitor and update the other set of software, often called agents, distributed on all client machines. These agents can report client status to the server and carry out management tasks assigned by the server, such as patching, updating, and scanning. As mentioned before, it is challenging for these management tools to tackle the management and security problems fundamentally. With the constant change in the increasingly distributed and heterogeneous environment, multiple operating system images and hundreds of applications have to be maintained and the patch and system security assured. Moreover, if the connectivity of the client to the server is destroyed in any way, the management tools cannot function anymore, resulting in manual effort or other data or information lost.

To overset the distributed model of PC, new centralized computing paradigms, such as thin client [3–5] in the past decade and new emerging virtual desktop [6,7], try to get PC off the desktop by centralizing both computation and storage on the server and only delivering the keyboard and mouse input and display output between the client and server, which is similar to the mainframe computing a long time ago, but can support desktop operating systems and applications. Due to the centralization of computation and storage, both management and security tasks are also centralized on the server, reducing the overall management efforts in half [7]. However, the large video display data transferred from the server to the client will consume much network bandwidth, it is very limited for these computing paradigms to support multimedia applications, such as video playback and 3D games. Further, the computing power of clients will be underutilized and wasted.

We believe that a new Transparent Computing paradigm with distributed computing, while centralized storage of all software and data, can achieve the best benefits of both the distributed computing model of PC and the above centralized computing paradigm. Without local storage, clients keep no persistent states and execute programs with local resources, while administrators can ensure centralized control of all software and data at a small number of servers, hence effectively addressing various challenges associated with distributed and inconsistent system states.

In this paper, we present a transparent computing system, namely StoreVirt, that can provide desirable features, such as heterogeneous OS support, user transparency, and flexible software and data sharing, by separating computation from storage. StoreVirt decouples software, data, and states from the underlying client hardware. The StoreVirt clients perform all the computing tasks, while all required OSes, applications, and data will be located at centralized servers and streamed to the clients on demand. The key technique is the virtual storage/disk mechanism which simulates the physical block-based storage devices using disk images located on the server and accesses them via network communication.

The remainder of this paper is organized as follows. In Section 2, we introduce the concept of transparent computing as a background and discuss some related work. In Section 3, we provide the detailed ideas and design of StoreVirt. In Section 4, we present the implementation and real experiences of StoreVirt. In Section 5, we study the performance of StoreVirt through several experiments and compare it with other similar approaches. Section 6 discusses possible extensions and optimizations. In Section 7, we conclude this paper.

2. Background and related work

2.1. Concept of transparent computing

To address the challenges faced by today's personal computers as mentioned above, we proposed a new computing paradigm, termed as transparent computing [8,9]. Its aim is to realize the vision advocated by ubiquitous or pervasive computing [10,11], in