Systems Software &
Operating Systems 88-4
(2001 July 26)

Checkpoint for Network Transferable Computer

Kuniyasu SUZAKI

k.suzaki@aist.go.jp
http://www.etl.go.jp/~suzaki/NTC

National Institute of Advanced Industrial Science and Technology
Information Technology Research Institute

"Network Transferable Computer" is a system which enables the transfer of the running OS image (Snapshot) to another machine through the use of a virtual machine. The previous version had to stop the virtual machine to get the snapshot because the snapshot was taken by hibernation. We developed Checkpoint on the OS (Linux) and the new version enables the taking of the snapshot without stopping the virtual machine.

Checkpoint for Network Transferable Computer

Kuniyasu SUZAKI

National Institute of Advanced Industrial Science and Technology,
Information Technology Research Institute
Tsukuba Central 2, Umezono 1-1-1, Tsukuba, Ibaraki, 305-8568, Japan

"NTC: Network Transferable Computer" is a system which enable to transfer the running OS image (Snapshot) to another machine using virtual machine. The previous version has to stop the virtual machine to get the snapshot, because the snapshot is taken by hibernation. We develop checkpoint on the OS (Linux) and the new version enables to get snapshot without stopping the virtual machine.

1 Introduction

"Network transferable computer" [1] is a system that makes it possible to continue working at home without physically bringing a computer from your office. This is not the same as accessing the office computer by using a network such as telnet. It is not projecting a mere image on the display monitor as in the case of VNC [2]. It is taking a snapshot of your office computer and recreating what you were doing on it on your home computer based on the executable image. An executable image is taken as a snapshot via a virtual machine. The mechanism for taking snapshots and recreating executable images is the hibernation function used on notebook computers.. The ordinary hibernation stops the machine and therefore cannot operate it anymore. But hibernation on a virtual machine does not stop the OS that is running the virtual machine and therefore makes it possible to operate the machine. Currently a "network transferable computer" is built on the combination of virtual machine software, VMWare [3], free OS, Linux, and hibernation software, SWSUP [4]. It is possible to pause a QuickTime movie on an X Window machine and continue to play it on another machine.

"Network transferable computer" serves two goals. One is to provide software developers with a platform where they can have a common execution/debugging environment, which is not possible with open source. The other is to provide end-users a function that enables them to transfer the condition of their machine's operation and use it for trouble-shooting in our information home appliance era.

"Network transferable computer" will give software engineers a common platform for collaboration. By distributing copies of execution images of a machine and sharing the same state, instead of merely transferring static information such as source code, "network transferable computer" makes it possible to provide more concrete information such as dynamic debugging information and graphical information on an X Window machine.. This will make software development easier and faster.

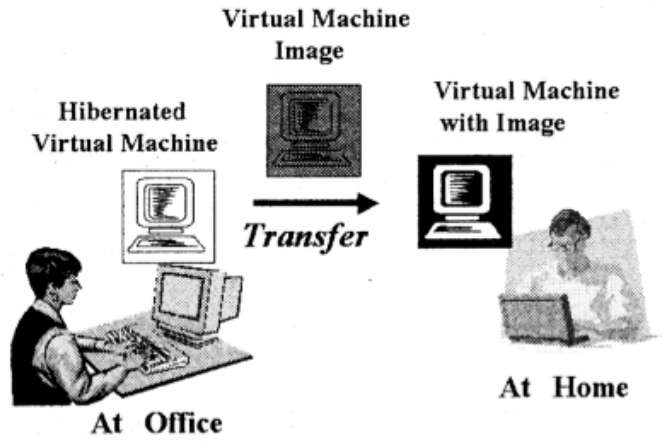


FIG. 1 Transfer execution image (transfer office environment to home)

Also end-users can send their software errors as is to a service center via network. In the past a computer needed to be brought into a service center and rebooted to recreate the problem. It is not only labor-intensive but also difficult to recreate the problem. As information home appliances are getting more and more popular, the labor-intensive issue and re-creation issue are expected to only get worse. "Network transferable computer" will provide a solution.

The previous version of "network transferable computer" had to stop a virtual machine to take a snapshot of the OS because a snapshot was taken by hibernation. When this is done on a server, it means the network connection is shut down. In order to solve this issue we developed a check point function, which makes it possible to take a snapshot in the middle of execution. We will now give a full report on this topic.

2 Network transferable computer

What "network transferable computer" does is the same as taking a hard drive out of a computer, connecting it to another computer, and booting it on this hard drive. In order to successfully boot the second computer, both computers have to have the same architecture as well as the same connecting device and BIOS.

"Network transferable computer" travels between virtual machines as shown in FIG. 2. Virtual computers run on virtual machine software. And they provide the same

environment with common devices regardless of the OS. For example, virtual machine software, VMware [3] can operate on Linux and Windows NT. The OS that executes the virtual machine software is called the 'host OS'. The OS that is installed on the virtual software is called the "guest OS". The Guest OS is installed on a virtual disk provided by the virtual machine software based on the common device. Because the host OS continues to operate after the guest OS stops, the disk image with the guest OS installed can operate after the guest OS is stopped. Further the disk image can be transferred and used to boot a virtual machine on a different machine. As long as the machine can execute the virtual machine software, you can use a desktop computer or a notebook computer.

"Network transferable computer" not only transfers a disk image but can also recreate the same OS environment on another computer. This is the same function as a notebook computer's hibernation. It enables it to pause an application without quitting, stop the OS, start the OS again, and continue working on the application. Combining the hibernation function and virtual machine together makes it possible to resume your application on any computer.

As long as data transfer is allowed either via network or removable memory media, "Network transferable computer" takes a snapshot of one computer's execution image and starts another computer in the same condition as when the snapshot was taken. Required functions for "network transferable computer" are the virtual machine and hibernation on the OS installed on a virtual machine. In the next chapter we will discuss the details and our current mounting condition.

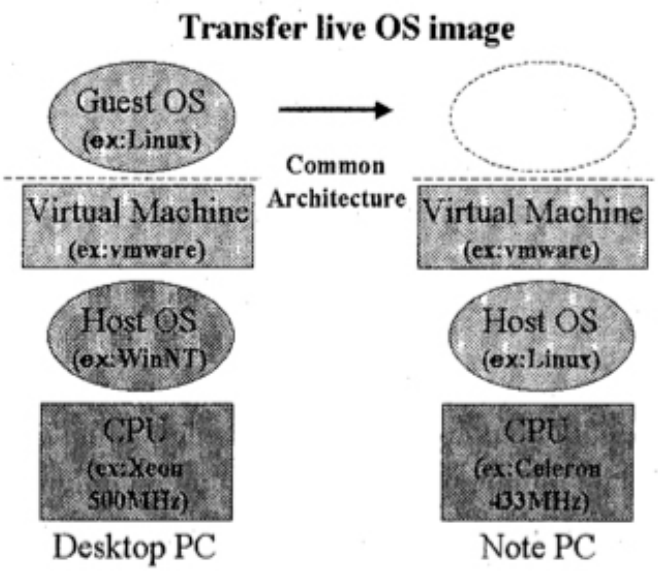


FIG. 2 How to realize a "Network transferable computer"

2.1 Virtual machine

Virtual machines were originally created for developing OS. Virtual machines trace a machine's condition and make it easy to debug. Lately they are used to run applications on a different OS. There are a lot of emulation software such as Sun Micro Systems' WABI and free software such as Wine and DOSEMU for this purpose. Virtual machine software is getting popular as the CPU's power improves and disk capacity gets bigger.

Software that only performs emulation is not suitable for "network transferable computer" because our purpose is to recreate execution images on different computers. Software that provides true virtual computing such as Connextix's Virtual PC [5] and VMware's VMware [3] is suitable. Virtual PC is virtualization software that runs on Macintosh computers while VMware is virtualization software that runs on Linux and Windows NT. "Network transferable computer" uses VMware that runs on Linux with open source code.

A virtual computer's VMware provides the equivalent of an Intel Pentium CPU, memory up to 512MB, SVGA graphics, IDE hard disk, AMD PCnet network card, Sound Blaster 16, and PhoenixBIOS 4.0R6. The OS that can run this virtualization software such as Linux and

that is installed on a virtual computer is called the "guest OS." The Guest OS can be Linux, FreeBSD, Windows 95, 98, NT, and 2000. The Guest OS is installed on a virtual disk that VMware provides. The SWAP area is also created in the same virtual disk. "Network transferable computer" uses this virtual disk as a snapshot for an execution image.

2.2 Hibernation

Hibernation is a function that saves energy by saving a computer's state information. This function was created to save energy usage on notebook computers by putting the CPU and/or a hard disk to sleep and yet enabling them to wake up and resume the operation where it was. In this article we define hibernation as below since different terms are used depending on different OS.

Standby mode: Saves state information in memory and puts CPU and hard disks to sleep. Power must be provided to keep the content of memory.

Suspend mode: Moves all the state information to nonvolatile memory such as hard disks and turns off a computer. There is no need to provide power.

"Network transferable computer" must use suspend mode where all the data exists in a hard disk during hibernation in order to transfer state information.

Usually hibernation controls the power consumption of each device by a BIOS energy saving interface standard such as APM (Advanced Power Management) or ACPI (Advanced Configuration and Power Interface) and the OS working together. Linux uses the APM command but it doesn't always achieve hibernation. Fortunately there's software available called SWSUSP (Software Suspend) to make up for AMP's command flaws. It enables the power to be safely turned off without depending on BIOS.

SWSUSP consists of Linux kernel patch and patch to software related to shut-down/boot

hibernation using SWSUSP is not a transition to suspend mode. In between the shutdown process and boot (init process) execution process state is evacuated to the SWAP area and memory is restored.

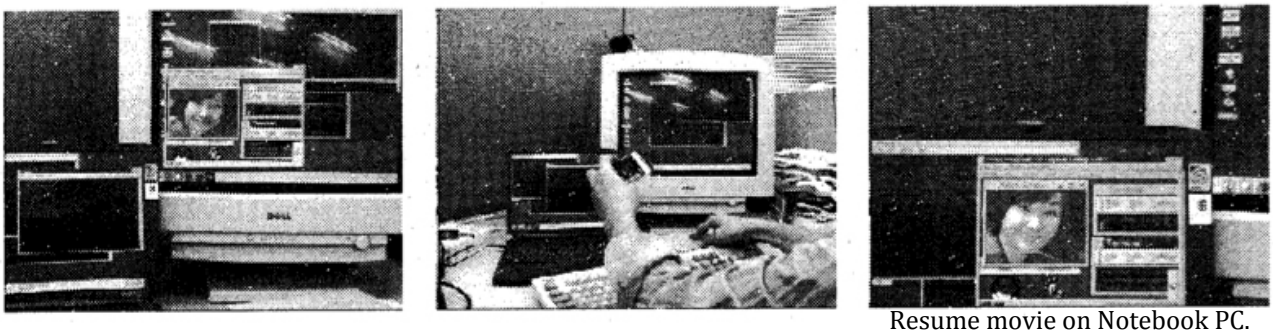
The shutdown process patched with SWSUSP calls the bdflush process and evacuates the buffer marked dirty for all the executing processes to the SWAP area before killing the executing process by sending SIGKILL. When the evacuation is completed, shutdown process executes sync and halts the OS. The next time it is booted, init process patched with SWSUSP enables the SWAP area to be available (when swapon is executed), and the execution image that was evacuated to the SWAP area is returned to memory to resume the process.

SWSUSP imitates a special shutdown/boot as hibernation. Therefore if a computer works with multiple OS's, Linux can execute hibernation by SWSUSP, boot a different OS, and shut down the OS, and resume Linux paused by SWSUSP.

2.3 Mounting

Currently "network transferable computer" is used on Linux. We used virtualization software, VMware. We installed Linux patched with hibernation software, SWSUSP, as a guest OS on a virtual machine provided by VMware. This guest OS, Linux, including the SWAP area is installed on a virtual disk created by VMware. A snapshot of the execution image taken by SWSUSP is stored in the SWAP area of the virtual disk. This virtual disk is delivered to another computer by the host OS. Then VMware on the second computer boots Linux on the virtual disk from the snapshot taken by SWSUSP, resuming the last state of the operation.

In this environment it is possible to pause a QuickTime movie played by XAnim on a desktop computer, transfer the execution image via network or a PC card to a notebook computer, and resume the movie (FIG. 3). You can watch the demonstration of the operation and the installation process at our website here (http://www.etl.go.jp/~suzaki/NTC).



Movie on desktop PC. Saves execution image.

Transfer on a removable disk.

Resume movie on Notebook PC.

FIG. 3 Demonstration

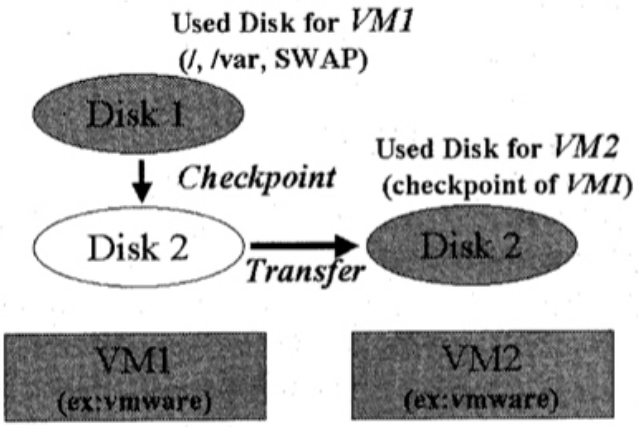


FIG.4 "Network transferable computer" using Checkpoint

3 Checkpoint function

Checkpoint function makes it possible to save current state information without stopping the execution of the process. Checkpoint was primarily developed for fault tolerance while hibernation is a function which enables a notebook computer to pause OS execution to save on power consumption. It can pause and resume the operation but it cannot save it.

Because "network transferable computer" was originally developed from hibernation, a virtual computer has to be stopped to take a snapshot of the OS. The process can be executed again when the computer is started but the network connection is lost. To solve this issue we developed a checkpoint function which makes it possible to take a snapshot of the state information without stopping the virtual

The checkpoint was developed by modifying SWSUSP (Software SUSPend), Linux hibernation software. Basically we use two hard disks, Disk 1 for the original OS, and Disk 2 for snapshots (FIG. 4). When the OS executes Checkpoint, a copy of Disk 1 is taken by Disk 2. After Checkpoint takes a snapshot, the original OS continues to run and enables another virtual computer to resume the operation from the point at which Checkpoint was executed.

The detailed disk operation of Checkpoint is shown in FIG. 5. The two disks consist of identical partitions with a read only partition, a read/write partition, and a SWAP area partition. Disk 1's SWAP area is used as ordinary SWAP while Disk 2's SWAP area is used for snapshots. When Checkpoint is executed, memory and the process being executed in Disk 1's SWAP area is saved in Disk 2's SWAP area. The area that does not need to be saved (pages that are not dirty) is eliminated so that snapshots can fit within Disk 2's SWAP area.

When Checkpoint is executed, the file system on Disk 1 and Disk 2 must be exactly the same. To be exact even the i-node number must be the same. When Checkpoint is created, the i-node number of Disk 1's file system is copied to memory. Checkpoint then stores the snapshot in Disk 2's SWAP area. Later the execution is resumed by the snapshot of Disk 2's SWAP area and file system. If there is no consistency between the i-nodes, the file system will be destroyed. Checkpoint does not operate correctly when we copy and paste each individual file from Disk 1 to Disk 2 because it does not maintain its consistency. Fortunately VMware provides a virtual disk, which is an image of a hard disk. By copying and pasting it

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.