社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－OS－101 (1)
2006／2／16

Virtual Machine Streaming and its Applications

RICHARD POTTER,† MIZUKI OKA†† and KAZUHIKO KATO†,††

Even if only a small part is used, bundling extra software in OS distributions is valuable, as is proven by the willingness of users to spend hours downloading them. Virtual machines, such as VMware's Player, now allow bundling in simulated hardware devices and snapshots of machine state. This paper will show how Virtual Machines and OS distributions can be streamed, analogously to video streaming. Users get all the advantages of large distributions with greatly reduced download times, so large distributions become practical for new applications such ubiquitous computing, sustainable computing, and flexible Web-based active content.

1. Introduction

Large general-purpose bundles of software, such as standard Linux distributions, have many advantages over software installations built up entirely from scratch. Since they are used by many users, there is more confidence that software is installed correctly. A certain level of software quality can be assumed based on the reputation of the distribution. The publisher of the software bundle can make sure that separate software packages are installed such that they work together properly.

In order to be general-purpose and thus be useful to many users, such bundles of software tend to grow large, mostly including features that few people use. Today's Linux distributions can be several gigabytes in size and take many hours to download, even on high-speed Internet connections. After downloading, installation can take over an hour.

Even though most of this time is spent installing software that will never be used, the value of using pre-selected and pre-integrated software justifies the extra cost in time and disk space. If the user can anticipate the need for the software many hours in advance, it can be downloaded and installed with low risk of problems with incompatibility or needed features not being installed. Unfortunately, if the user can not anticipate the need far enough in advance, the advantages of bundled software become out of reach.

This paper presents examples of where bundled software is advantageous, but the need for it can not be adequately anticipated. We propose that Virtual Machine Streaming can make these and similar examples practical. Virtual Machine Streaming has 5 basic aspects:

(1) Run bundled software inside a virtual machine,
(2) download only parts of the virtual machine that are needed,
(3) download the parts as early as possible,
(4) overlap execution with downloading, and
(5) download the parts in the order needed.

If all the Virtual Machine Streaming aspects can be achieved, the user can start using a large standardized software distribution at the earliest possible time and also complete a task at the earliest possible time. These times can be 10 to more than 100 times earlier than using the software by fully pre-copying and installing it from scratch. The behavior of the software itself is the same, except that there is some risk of some stall time during use, that is, that the virtual machine may freeze temporarily while waiting for needed data to arrive.

If stalls are unacceptable for the user, then successful streaming can offer an optimal tradeoff between starting use at a certain time and having a high likelihood of stall-free use. When stall-free use is a priority, then the user has the option of waiting some time before starting use. If starting and/or completing use is a higher priority for the user, then the user simply starts using the virtual machine as soon as it is restored and waits patiently during stalls, if they occur.

Virtual Machine Streaming offers a simple *autonomic* mechanism for balancing between these competing goals. Because the data for the virtual machine arrives in a continuous stream, any pause by the user during the task increases the chance of stall-free operation thereafter. Therefore, even if earliest task-completion is

† Japan Science and Technology Agency (JST) CREST
†† Graduate School of Systems and Information Engineering,, University of Tsukuba

most important to the user, the Virtual Machine Streaming will still take every opportunity to make operation stall-free as well.

The next section will explain each aspect and how our prototype system strives to achieve it. In Section 3, two examples are given for applications where Virtual Machine Streaming can make the use of large standardized distributions practical. The last two sections review related work and present our conclusions.

2. Prototype System

To explore some of the practical issues with Virtual Machine Streaming, we built a prototype system that includes techniques to address each of the aspects listed above. The following subsections describe the techniques and how they contribute to the overall goal of reducing the time to download, install, and begin using large standardized software distributions.

2.1 Use Virtual Machine

The use of virtual machines by itself makes using bundled software more practical by making the installation step unnecessary. This benefit is illustrated by the differences between **Fig. 1**, which shows an install step, and **Fig. 2**, which show no install step and thus shows use beginning earlier.

For our prototype, we use the User-Mode Linux (UML) virtual machine. UML is a Linux kernel ported to the system calls on a host Linux system. This means that one or more UML machines can be started on a single physical host machine. Each UML appears as a set of user-mode processes on the host, however inside it behaves as a standard Linux kernel with its own memory and process space. Hard disks for UML are simulated on the host by files, into which people have successfully installed standard Linux distributions such as RedHat, Fedora, Debian, etc. When these files are posted on the Internet, it becomes possible for other people to download the standard distributions and begin using them with UML without any time spent on installation.

UML is open source, which has made it possible to include the Virtual Machine Streaming features. It's ability to run standard Linux distributions has made it possible to demonstrate the examples in the next section.

2.2 Minimize Download

Especially when used for a specific purpose, only a few of the software packages included in a standard distributions are used. Even in the packages that are used, not every feature is used. For example, the application may include help pages that are never referenced. When the user can easily avoid using features, the download requirements can be easily reduced.

For our system, we use snapshots so that users can easily avoid two types of features that would otherwise require large downloads: booting and application startup. Potentially, the need can be eliminated for any data demand fetched before the snapshots is taken. We use the snapshot feature in SBUML[1], our extended version of UML, to take snapshots of the virtual machine runtime state after booting and starting up needed application programs. To make sure that not-needed-again information is not saved in the snapshot, we use a simple memory ballooning technique[6] to flush the memory and disk cache of the virtual machine before saving the snapshot.

In order to accurately identify what information is needed after the snapshot is restored, our system builds on the demand fetching already built into the file system. To make the implementation flexible, a hook was placed into the UML disk driver to make it notify a separate utility program whenever one of the 4096-byte data blocks is about to be accessed during use of the virtual machine. The utility downloads the required data block from the SBUML Streaming Server and then signals the virtual machine that the data has arrived and that it can continue.

Because demand fetching waits until the last possible moment to decide what to download next, it can minimize the download with high accuracy. For such a simple technique, it is very effective. The downside of waiting so late is that the virtual machine is almost guaranteed to stall until the needed block of information arrives. The amount of stall can potentially be long because the file system and OS often delay the next request until the previous request has been filled. Delays such as network latency and process scheduling therefore acumulate for each of the potentially thousands of requests. To make matters worse, the stalls are likely to happen at frustrating times, because if the user initiates an action for the first time, the virtual machine might start downloading the information necessary to satisfy the action exactly when the user is expecting an immediate response.

Figure 3 summarizes the overall benefit of

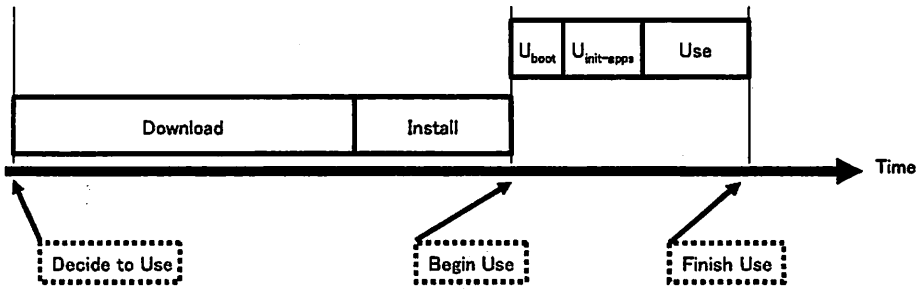

Fig. 1 Traditional way of preparing large standardized software distributions for use.

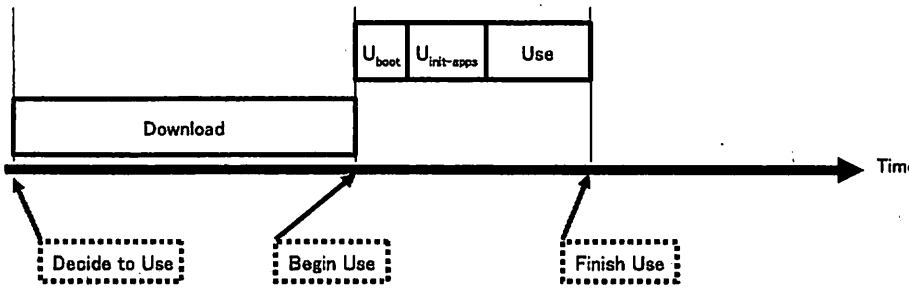

Fig. 2 Virtual Machines allow pre-install installations to be distributed.

snapshots, ballooning, and demand fetching. To represent that usage often consists of multiple steps, the use rectangle is divided into three smaller rectangles ($U_{1,2,3}$). Each of these steps require that certain parts of the distribution be downloaded ($D_{1,2,3}$) before the user can perform the cooresponding step. The network and other latency delays the download and adds to the stall expereinced by the user. Demand fetches tend to occur in bursts as the user begins to use additional software features. Therefore latency, stall, and download times represent the accumulated amount of time for the many steparate 4k-block requests and downloads.

Even with the extra stall time included, the user steps can be completed earlier than with the complete pre-copy in Fig. 2, because most of the distribution does not have to be downloaded and the user does not have to boot the virtual machine or startup applications.

2.3 Download Early

If the problem with demand fetching is downloading too late, the obvious solution is to download earlier. But this means the decision to download information must be made before the virtual machine requests it. The prediciton this requires is impossible to make perfect. So while in theory this could make it possible to finish use in less time, as illustrated in **Fig. 4**, in practice it depends on the accuracy of the prediction under the specific conditions. Downloading early also makes it possible to remove the stalls.

The SBUML Streaming Server performs predictions by comparing the virtual machine's recent disk activity with that of previous virtual machines. The previous disk accesses are grouped into sequences that represent disk accesses that occur within a short time period. When making a new prediction, the sequence that contains the highest number of the virtual machine's recent disk accesses gets streamed. Since it is possible that the sequence is streamed before new recent disk activity happens, the server also keeps a transition graph of how previous virtual machines transitioned from one sequence's disk activity to another. This transition graph is used to choose follow-on sequences to stream.

The three most important properties of the prediction system for the purpose of this paper are:

(1) It is accurate enough to demonstrate close-to-ideal streaming.

(2) Streaming is at the block level so that

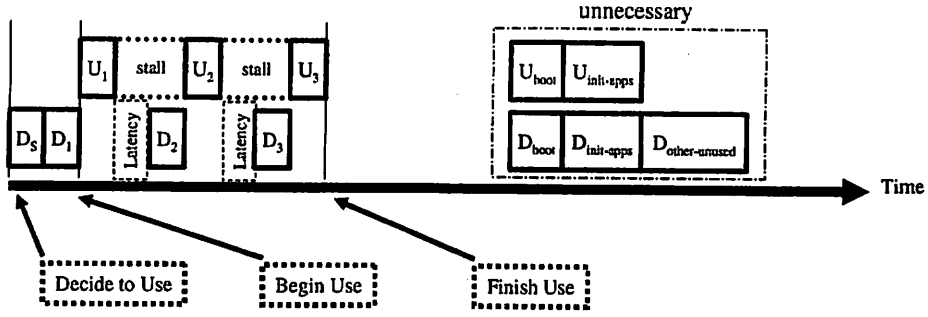

Fig. 3 Snapshots and demand fetching reduce the amount that must be downloaded, but guarantees stalls during use. The snapshot can make booting and application startup unnecessary, and therefore the corresponding downloads for these are unnecessary too.

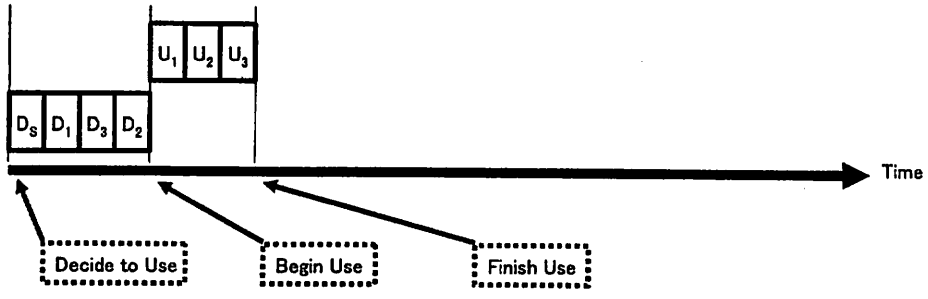

Fig. 4 Downloading earlier can enable faster use with no stalls.

subsets of files are usually sent.

(3) By traversing the sequence graph, large amounts of data can be streamed, so that data needs far into the future can be filled.

2.4 Overlap Use with Downloading

If early completion of use is the user's primary goal, then letting the user start using the software while the streaming continues improves performance. Unlike with demand fetching, this does not imply that the machine stalls during downloading. Still, if downloading is not fast enough, stalls are still possible, as illustrated in **Fig. 5**.

The streaming components of SBUML run in different processes than the virtual machine so that streaming can continue while the virtual machine is used. The server periodically checks recent disk activity from the virtual machine, so overlapping use with downloading can improve prediction accuracy with more up-to-date information.

2.5 Download In Usage Order

When execution is overlapped with downloading, the order that data is downloaded can affects the length and number of stalls. Downloading the same data in the order it is used can make earlier completion of use possible as illustrated in **Fig. 6**. The SBUML Streaming Server's transition graph of sequences records the time ordering of disk access of prevous virtual machine clients. Therefore it can often be used to demonstrate this aspect.

3. Examples

Assuming the SBUML Streaming Server has seen a particular snapshot used in a certain way, future users should be able to use the snapshot in the same way at arguably optimal performance. The following examples show this in more concrete terms.

3.1 Active Content

Suppose a researcher wants to publish interactive content on a website to let readers explore relationships in some data. The researcher knows it is probably possible to do this in Java, and maybe even Flash or Javascript. However, the only tool that the research can

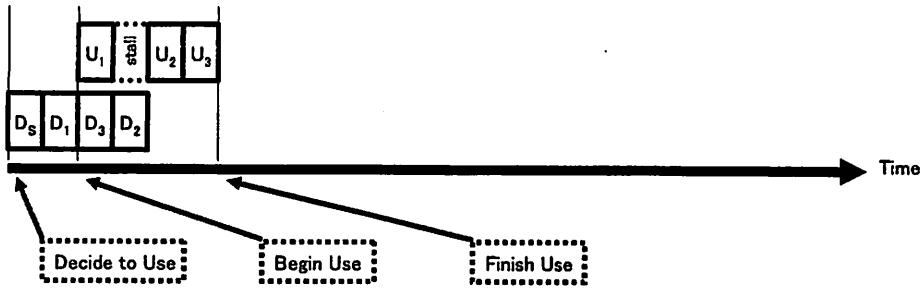

Fig. 5 Overlapping download with use allows early start and finish for use, but makes stalls possible.

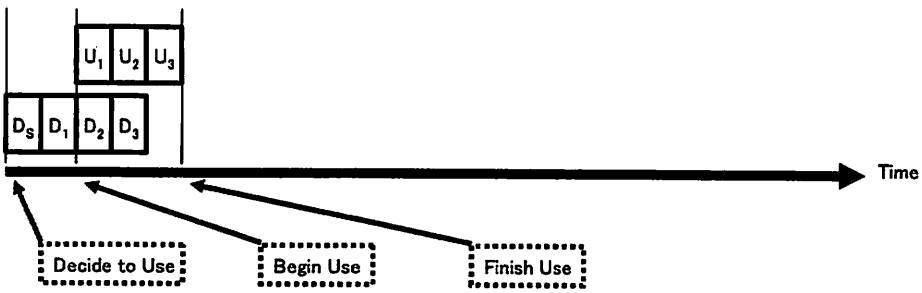

Fig. 6 Downloading information in the order needed can remove stalls and speed task completion.

use to confidently build this content is with interactive form element in a Gnumeric spreadsheet as shown in **Fig. 7**. One option is to post the spreadsheet document to the website. However, some readers might not have Gnumeric installed. In fact, after trying the spreadsheet on another computer, the researcher discovered that the spreadsheet was not portable to other versions of Gnumeric, so readers would have to have a specific version.

Virtual Machine Streaming makes its easy for the researcher to distribute a pre-installed version of a confirmed-to-work version of Gnumeric in a way that is practical for users to download from a website from scratch. The researcher starts by aquiring a virtual machine image with the correct Gnumeric. The easist way to do this is to pick a large distribution that already has it installed.

Without Virtual Machine Streaming, there would be an incentive to try to build a small distribution that included only Gnumeric. With virtual machine streaming, the large size of a standard distribution is not a problem. The snapshot in Fig. 7 was built by booting a full install of RedHat 8.0, which takes up over 3.7GB on disk. The virtual machine memory size was set to 32MB and Gnumeric was started inside X-windows using twm. After loading in the spreadsheet document, memory was ballooned to zero out unused memory pages. A snapshot of the virtual machine was saved, which required about 6MB of disk space. Just booting and starting X-windows accesses about 60MB, so in this case the snapshot saves considerable space. Next, the researcher has posts the 6MB snapshot to the website and has it reference an SBUML streaming server that hosts the 3.7GB distribution.

In order for the server to stream the correct blocks in the correct order, it must first record the disk activity generated by the virtual machine after the snapshot is restored. Therefore, the researcher restores the snapshot and uses the spreadsheet in the way users are expected to use it. For example, the key point of the spreadsheet is to use the sliders to explore many values quickly. Using the slider causes the virtual machine to download the specific pieces of code for Gnumeric, X-windows, etc., that are involved in adjusting the slider and updating values and the chart on the spreadsheet. This

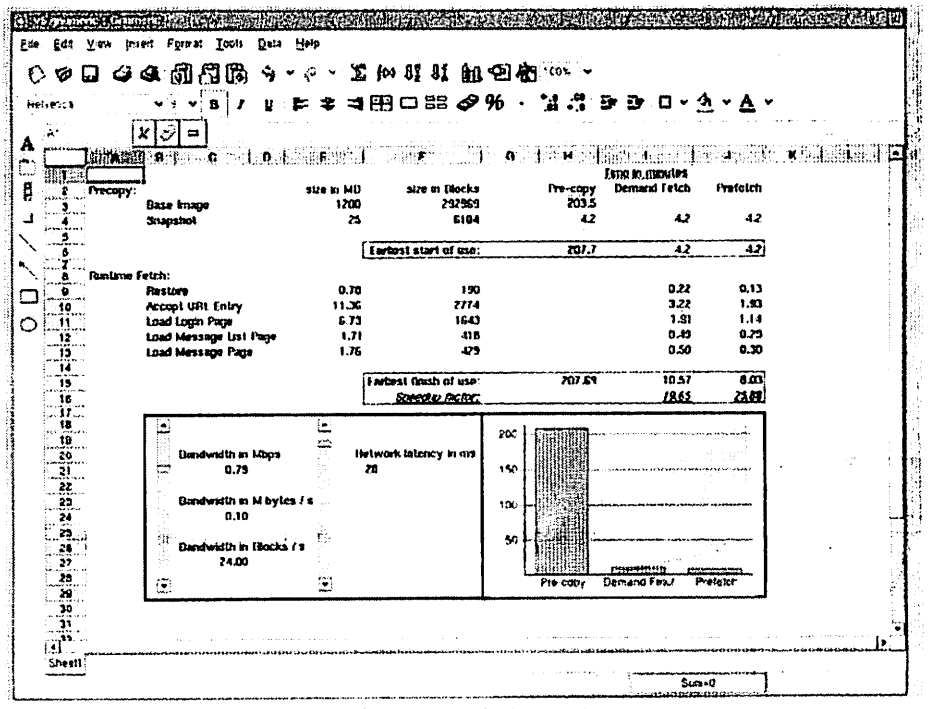

Fig. 7 Spreadsheet to publish along with application.

requires over 1000 blocks to be downloaded, during which the virtual machine stalls and after which the spreadsheet finally begins to respond to the slider adjustments.

Now the server is ready to give future users all the benefits of Virtual Machine Streaming. After a user decides to download and use the snapshot, the system first downloads the 6MB snapshot. As soon as the snapshots is restored, the streaming server automatically begins to download all the blocks needed to make the sliders work and the spreadsheet update. This requires streaming an additional 6MB for the disk blocks. On a moderate speed ASDL line, all downloading and streaming takes about 3 minutes. On a direct line, the time drops to about 20 seconds from the decision to use the spreadsheet until actually using it.

In this case, the standardized software distribution is 3.7GB or about 1.2GB compressed. Downloading 12MB instead fof 1.2GB means that virtual machine streaming can cut the time required to begin using the researcher's spreadsheet by a factor of 100. This brings the times down to a range that can be practical for many users.

However, a perhaps more-important benefit is that virtual machine streaming allows the researcher essentially to create a mini-distribution that is speciaized for showing the spreadsheet without having to learn anything about installing Gnumeric or understanding its dependancies on other software packages. The resulting size of 12MB is quite resonable considering that the rpm installation file for Gnumeric is 7MB, and the rpm for XFree86 by itself is 11MB. Therefore, creating a smaller download with the same functionality would be challenging.

3.2 Sustainable Computing

Virtual machine streaming could make it easier to more flexibly assign servers for backup and recovery operations for sustainable computing. While main servers should be on fast links, the same is not necessary true of backup servers. For allocation of new servers during major network outages or peaks in demand, it may be necessary to allocate new servers on fast links and get them up running from scratch. However, the connection between the backup image and the newly allocated server may be slow. **Figure 8** shows examples where a major

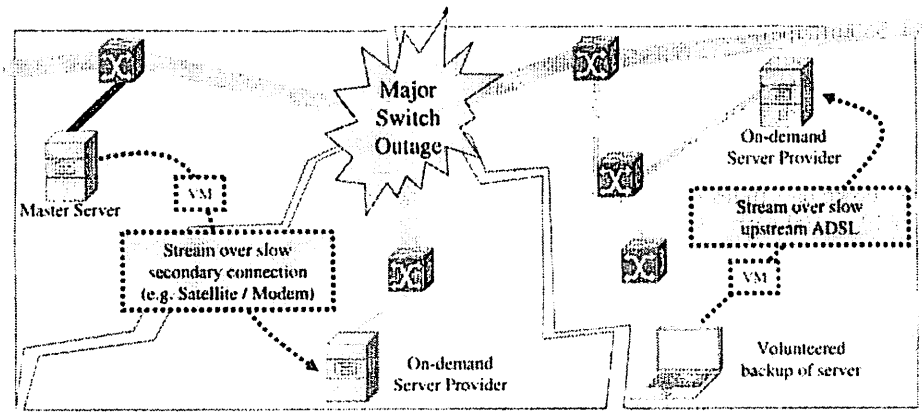

Fig. 8 Virtual Machine Streaming can be used to initialize servers over slow links.

network outage could require slow links such as upstream-ASDL or modems to initialize newly allocated servers. Virtual Machine Streaming techniques could be used to stream the most likely needed parts of the image until the risk of stall is low enough to bring the server on-line. Since the streaming can continue while the server is on-line, in time the risk of stalls could be reduced to zero.

4. Related Work

Virtual Machine Streaming is a form of virtual machine migration, which is gaining popularity as a conceptually simple and practical means of moving computation between physical hardware. Techniques for optimizing virtual machine migration have been developed to address various performance goals given particular underlying assumptions. For load balancing applications, VMware's VMotion product[5] and Xen's machine migration[3] are optimized to minimize downtime, under the assumption that most of the memory image can be pre-copied. Block storage devices are assumed to be on the local network and thus do not have to be migrated. For workspace migration, the systems in 6) and 4) are optimized to minimize startup time at a destination across some WAN, but assume that the destination is known in advance and that most of the state can be pre-copied. In contrast, the optimization goal for the research in this paper is to minimize startup time for virtual machines migrated on-demand over wide area networks such that no pre-copy phase can be assumed.

Some aspects of Virtual Machine Streaming have been explored in projects such as HTTP-FUSE-KNOPPIX[2] by using demand fetching to download only the needed parts of a 3.8GB Knoppix Live-CD software bundle, which contains over 2500 software packages. Instead of downloading the whole 3.8GB image, parts of the image are downloaded as Knoppix software requests them. Booting the images requires about 250MB and is performed on actual hardward using a CD with only 5MB of content. Execution is overlapped with downloading, because additional pieces are downloaded as specific applications are used. Since it is unlikely a user will every use all 2500 packages, the overall download remains much less than the 3.8GB total.

VMware now supports the publishing of virtual machine images with its VMware Player product. One example image is the Browser Appliance, a pre-installed Linux distribution with a software bundle suitable for Web browsing. The Browser Appliance bundles in Macromedia Flash Player for Linux, an addition that the average might not think to include if installing everything from scratch. Everything in the software bundle still must be downloaded. For example, even though the Browser Appliance is specialized and relatively small, it is still over 200MB compressed. Therefore, on slower internet connections, its usefulness might need to be anticipated by an hour or more, because all 200MB must be downloaded to use any of the bundled software.

5. Conclusions

Virtual Machine Streaming can bring the advantages of large standardized software distributions to situations where the time of use

can not be anticipated adequately in advance. The streaming additions to the SBUML project have allowed us to experiment with how well these ideas perform with standard software and realistic examples. By using snapshot to minimize the amount of data required by the virtual machine, download times can be reduced by a factor of 10 to 100 without risking loss of functionality.

References

1) : http://sbuml.sourceforge.net/.
2) : http://unit.aist.go.jp/itri/knoppix/http-fuse/.
3) Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I. and Warfield, A.: Live migration of Virtual Machines, *Proc. 2nd USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2005)* (2005).
4) Kozuch, M. and Satyanarayanan, M.: Internet Suspend/Resume, *WMCSA '02: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*, Washington, DC, USA, IEEE Computer Society, p. 40 (2002).
5) Nelson, M., Lim, B.-H. and Hutchins, G.: Fast Transparent Migration for Virtual Machines, *Proceedings of the USENIX 2005 Annual Technical Conference*, Anaheim, California, pp.391–394 (2005).
6) Sapuntzakis, C. P., Chandra, R., Pfaff, B., Chow, J., Lam, M. S. and Rosenblum, M.: Optimizing the Migration of Virtual Computers, *Proceedings of the 5th Symposium on Operating Systems Design and Implementation* (2002).