Measuring the Multimedia Performance of Server-Based Computing

Jason Nieh
Department of Computer Science
Columbia University
nieh@cs.columbia.edu

S. Jae Yang
PC Magazine Labs
Ziff-Davis Media, Inc.
jae_yang@ziffdavis.com

Abstract

The server-based computing (SBC) model is becoming an increasingly popular approach for delivering computational services with reduced administrative costs and better resource utilization. In this paper, we examine how effectively SBC architectures support multimedia applications. We focus on the effectiveness of the remote display protocol used in three popular SBC platforms for supporting video applications, Citrix Metaframe, Windows Terminal Server, and AT&T VNC. Our results show that SBC can be a viable approach for delivering VCR-quality video in LAN environments, but that existing solutions are inadequate at network access bandwidths found in broadband environments. Our results also show that SBC can deliver video with comparable network efficiency as streaming media solutions. We show that there is wide variation in the performance of the remote display technologies and discuss the factors that influence their performance.

1 Introduction

Anticipating the growing demand for multimedia applications, researchers have designed networking and operating system mechanisms to support these applications in the distributed desktop computing environment that exists today. However in recent years, there is a growing trend away from the distributed model of desktop computing toward a more centralized model of server-based computing. In server-based computing (SBC), all computing is done by a set of shared server machines and the full persistent state of user sessions is maintained on the servers. Client machines connect to the server machines for all their computing needs. The only functionality required on the client is what is necessary for interacting directly with the user, such as sending keyboard and mouse input and receiving graphical display updates. A wide range of SBC products have been developed [1, 6, 3, 13, 15, 16]. As evident by the terminal services provided by Windows 2000, even desktop-based computing proponents are integrating these solutions into their operating systems. By using a more centralized computing model, SBC offers the potential of lowering the total cost of computational services through reduced system management cost and better utilization of shared hardware resources. This approach is being deployed to deliver computational services in a wide range of environments, from LAN-based workgroup environments [15] to Internet ASPs [4, 5, 12].

The key enabling technology underlying the SBC approach is a remote display protocol that enables graphical displays to be served across a network to a client device while applications and even window systems are executed on the server. Entire graphical user sessions are executed on the server, and this is done without storing any unrecoverable state on the client.

While SBC approaches have focused on their support for office productivity applications, little work has been done to examine how effective they are in supporting multimedia applications. Because the demand for multimedia applications will only continue to increase, it is important to consider how viable SBC approaches are for supporting these applications.

In this paper, we quantitatively evaluate how effectively SBC architectures support multimedia applications. We focus on the effectiveness of the remote display protocol used in three popular SBC platforms for supporting video applications. We focus on video because video is often the most demanding on system resources. We examine the underlying remote display technology because it is often the limiting factor in SBC performance. The SBC platforms included in this study are Citrix Metaframe [1], Microsoft Terminal Server [3], and AT&T VNC [13]. We evaluate their performance over a wide range of network bandwidths and across both Windows NT 4.0 Terminal Server Edition and Windows 2000 operating systems.

Our results show that in LAN environments, network bandwidth is not a limitation in delivering VCR-quality video via SBC. However, existing SBC solutions do not perform well for broadband network access bandwidths. We also show that some remote display protocols can deliver video with network efficiency comparable to streaming video solutions. Although some of the SBC platforms are similar in design, we show that there is a wide variance in the performance of the remote display technologies. We quantify the impact on video application performance of client caching and display encoding compression techniques used in SBC solutions. We found that client disk caching was not very effective in reducing bandwidth requirements and resulted in lower quality video application performance. On the other hand, we found that display encoding compression in SBC platforms was very effective in reducing bandwidth consumption during video playback and enabling good video application performance.

The paper is organized as follows. Section 2 describes the three SBC platforms evaluated in our study. Section 3 discusses the experimental setup used for our SBC testbed. Section 4 details our measurements of the effectiveness of SBC in supporting digital video. Section 5 discusses related work. Finally, we present some concluding remarks and directions for future work.

2 Server-based Computing Platforms

We used three of the most popular SBC platforms in our study, Citrix Metaframe, Microsoft Terminal Server, and AT&T VNC. The remote display protocols used in these platforms are Citrix ICA, Microsoft RDP, and AT&T VNC RFB, respectively. In this paper, we will refer to the SBC platforms by their product name and remote display protocol name interchangeably.

Each of the SBC platforms consists of three main components: a thin client, a server, and a remote display protocol. The thin-client application executes on a user's local desktop machine, and the SBC server executes on a remote server machine. The client and server communicate across a network connection between the desktop and server. On the server, applications run normally without any modification while the underlying SBC server redirects I/O to and from the server applications to the thin client on the user's local desktop machine. Using a remote display protocol, the client sends keystrokes and mouse clicks across the network to the server, and the server sends back screen updates in response to the user input. It is important for the screen updates to be encoded efficiently by the remote display protocol for good performance. Simply sending raw pixel values would be prohibitively expensive at current network bandwidths; even a 640x480 display with 8-bit pixel values would require over 100 Mb/s at typical monitor refresh rates.

We can describe the remote display protocols based on four important characteristics that influence their performance: display encoding, encoding compression, display update policy, and client caching. The display encoding is the type of primitive used by the remote display protocol for transmitting screen updates. The encoding compression is the type of compression that is applied to the display encoding to reduce the amount of data transferred for screen updates. The update policy is the policy for determining when screen updates are sent from the server to the client, which may adapt based on the availability of network bandwidth. The client cache is a cache on the client that can be used to cache display primitives that are reused so that they do not need to be resent from the server. These four characteristics are summarized for each SBC platform in Table 1. We describe the SBC platforms in further detail in the following sections.

	Microsoft Terminal Server (RDP)	*Citrix Metaframe (ICA)*	*AT&T VNC (RFB)*
Display encoding	DDI graphics	DDI graphics	regions of pixels
Encoding compression	2D RLE plus others	2D RLE plus others	2D RLE for hextile
Update policy	server push, adaptive server buffer	server push, adaptive server buffer	client pull, framebuffer scraper
Client caching	cache for glyphs and bitmaps	cache for glyphs and bitmaps	local framebuffer only
Client cache size	1.5 MB RAM, 10 MB disk for RDP 5.0	3 MB RAM, percent of disk partition	RAM size of local framebuffer

Table 1 **SBC platform characteristics**

2.1 Microsoft Terminal Server and RDP

RDP is the remote display protocol built into Microsoft Windows NT 4.0 Terminal Server Edition (TSE) and Windows 2000. RDP 4.0 is used in Windows NT TSE and RDP 5.0 is used in Windows 2000. Unfortunately, no public specification for RDP is available, so we cannot say exactly how updates are encoded. However based on information that is available [3], it is likely that the display encodings are based on graphics primitives similar to the Windows DDI video driver interface. This choice of encoding is not surprising given that the platform was primarily designed for Windows environments. The encoding supports higher-level semantics and distinguishes among different graphic primitives such as glyphs, fills, boxes, etc. The higher-level semantics can potentially provide bandwidth savings at the cost of additional processing requirements for the RDP clients. The display encodings are compressed by the server before being sent over the network to the client. A combination of run-length encoding and other compression schemes is used.

To support the display update policy, the server contains buffers for the display output. The contents are flushed to the client at a varying rate depending on the amount of user input arriving from the client and the amount of graphical output being generated at the server. If display output is being generated at a high rate, updates are pushed out as quickly as they are produced, which would presumably lead to optimal performance for multimedia applications such as streaming video. Otherwise, if keyboard or mouse input data is coming in, updates are sent 20 times per second, and otherwise only 10 times per second, in both cases within the approximate limit of 100ms at which latency becomes human-noticeable.

The RDP client utilizes a local cache for graphics objects such as glyphs and bitmaps to speed the display of today's graphical user interfaces. In RDP 4.0, the client reserves 1.5 MB of RAM for caching glyphs and bitmaps. In RDP 5.0, the client uses a 1.5 MB RAM cache in conjunction with a 10 MB persistent disk cache in a manner similar to a paging system. Glyphs are not stored in the disk cache as they are generally small so that the overhead of re-downloading them is typically small.

It is important to note that the display protocol is entirely independent of the session management. In fact, Citrix Metaframe, another platform we studied, uses the same Terminal Server-enabled operating system as a base and replaces RDP with its own protocol.

2.2 Citrix Metaframe and ICA

Citrix Metaframe resembles Terminal Server in basic organization and actually runs on top of it on the server side. As mentioned above, the Metaframe server software runs on a Windows NT TSE or Windows 2000 Server system and simply uses the Independent Computing Architecture (ICA) protocol instead of RDP to communicate with the clients.

Like RDP, no public specification for ICA is available, so we cannot say exactly how updates are encoded. However based on information that is available [1, 9], it is likely that the display encodings are based on graphics primitives similar to the Windows DDI video driver interface and RDP. The display encodings are compressed by the server before being sent over the network to the client. A combination of run-length encoding and other compression schemes is used.

The screen update policy employs buffers on the server side that are flushed at a varying rate depending on user input and the quantity of display output being generated. An evaluator called Speedscreen examines buffered display commands and prunes those that would otherwise be overwritten by more recent display updates before the commands are sent to the client.

Graphics objects such as bitmaps and glyphs are cached on the client to improve performance just as with RDP clients. A 3.0 MB RAM cache and a disk cache whose size can be specified as a percentage of the disk size are used for this purpose. The minimum size bitmap that will be cached on disk can be configured to be 2 KB to 64 KB and defaults to 8 KB. It is interesting to note that Microsoft claims that a 1.5 MB RAM cache is optimal for its SBC platform [3] while Citrix uses a RAM cache twice as large.

2.3 AT&T VNC and RFB

Unlike the other two SBC platforms, VNC is an open-source SBC platform that employs a completely different remote display protocol design. The VNC RFB protocol uses a display encoding based on simple regions of pixels. It supports a number of compression techniques based on pixel region primitives, the most efficient of which is the hextile encoding, which is also the default encoding used. Hextile encoding can be thought of as 2D run-length encoding of pixel values.

The screen update policy in VNC is based on a framebuffer scraper model that is driven by client screen update requests. The VNC server keeps track of pixel regions that have changed since the last time a screen update was sent to the client. Overlapping pixel regions are collapsed together. A screen update is sent to the client when the client sends a screen update request to the server. Upon receiving a screen update request, the server compresses and sends only the screen changes that have occurred since the last screen update request. This client-driven method of display updates is intended to make the server adapt to low-bandwidth connections and slow clients, as in these cases, the client will presumably request fewer updates and less information will therefore be transmitted.

The VNC client does not perform any form of caching except maintaining its local framebuffer. This is used for the copyrect feature which allows an existing rectangle of data saved in the client's framebuffer to be copied to a new location without re-requesting that information from the server. Moving a window can then be done by the client by copying the corresponding rectangular window of pixel values from one location to another.

Unlike Microsoft Terminal Server and Citrix Metaframe which can only display up to 1024x768 resolution with 8-bit color, VNC can display full-screen resolution with up to 32-bits per pixel. Since VNC uses a pixel-based encoding method, perhaps it is easier for VNC to support higher fidelity displays than the more complex graphics-based Microsoft Terminal Server and Citrix Metaframe solutions.

3 Experimental Design

The goal of our experiments is to quantify how effectively the different SBC platforms support multimedia applications, particularly digital video. In addition, we quantify the effectiveness of the platforms for different server operating systems and for different network access bandwidths between client and server. In order to do this, we need to determine how to measure and quantify the quality of the multimedia delivered on each of the platforms. We also need to be able to change network access bandwidths and operating systems in a controlled manner to ensure a fair comparison. We describe the application benchmark used for our experiments, our measurement methodology, and the hardware/software testbed we constructed for our experiments.

3.1 Video Application Benchmark

Because we are primarily concerned with remote display protocol multimedia performance, we used a digital video application for our benchmark. The application benchmark we used for our experiments was the Windows Media Video benchmark from the Ziff-Davis i-Bench version 1.02 [8], a comprehensive benchmark suite designed to measure web and multimedia performance. The video benchmark plays a 63 second MPEG-4 encoded video clip with an ideal frame rate of 30 frames/sec. The video clip has 1890 frames and is 23.74 MB in size. Each video frame is displayed at 320x240 resolution with 8-bit pixel values. The video is displayed using a Windows Media Player in-line plug-in that is launched within a Microsoft Internet Explorer 5 browser window. In addition to the video itself, the Windows Media Player also displays some simple graphics such as a digital clock that indicates the time that has elapsed during the display of the video clip.

To quantify video performance, standard metrics that are used include the video frame display rate and the display jitter. However, in an SBC architecture, the video application runs on the server while the display of the video is shown on the client in a manner that may be decoupled from the application. Hence, to measure the video frame display rate and jitter as perceived on the client, it would be necessary to modify and instrument the SBC clients. Since published specifications, source code, and measurement tools were not available for Metaframe, Terminal Server, or the Windows Video application, obtaining detailed frame rate and display jitter information was not possible.

As a result, we used three kinds of non-intrusive measurements to evaluate video application performance. First, we simply subjectively examined the visual quality of the video displayed on the client. Second, we measured how long the video clip took to be displayed on each platform. Third, using a network traffic analyzer, we measured the amount of data that was sent between server and client while the video clip was played. As we discuss in Section 4, these aggregate measures of display time and data transfer were generally correlated with our subjective perceptions of overall video quality.

3.2 Experimental Testbed

Our testbed was designed to allow us to perform well controlled experiments to measure remote display protocol performance between client and server over various network bandwidths. In addition to using a common multimedia application benchmark to ensure a fair comparison, we used a private network connection with network configuration and monitoring tools, a common set of operating systems, and a common hardware testbed for all of our experiments. The testbed configuration is listed in Table 2.

The testbed consisted of five PCs connected via a set of isolated 100 mbps Ethernet LANs. The machines included an SBC server and client. Each machine was connected via a separate isolated network to a PC with two network interfaces that ran The Cloud [2] network bandwidth simulator, which could then vary the network access bandwidth between the SBC server and client. The Cloud can connect the two nodes at various bandwidths, limiting the 100 mbps network bandwidth to as little as 2400 bps. We could then determine how network access bandwidth affected the performance of the three SBC platforms. A PC running an Etherpeek 4 [7] network traffic analyzer could be connected to either isolated network for real-time monitoring. Finally, one additional PC was connected to the SBC server and used for running the i-Bench benchmarking software.

Because Windows-based operating systems are the most common desktop operating system environments, we used them for both the client and server systems used in our experiments. For the client system, we used Windows NT 4.0 Workstation. For the server system, we used Windows NT 4.0 Terminal Server Edition (TSE) and Windows 2000, which are both designed for SBC servers. By using two server operating systems, we could consider the effect of operating systems on SBC performance. All three of the SBC platforms we measured ran on both Windows operating systems.In particular, we used Citrix Metaframe version 1.8, the Microsoft Terminal Server software that came with the operating systems used, and AT&T VNC version 3.3.3r2 for Windows NT/2000.

Role	Hardware	OSes loaded	Software
Benchmark server	550 Mhz Intel PIII, 128 MB SDRAM, 21 GB disk, 3COM 3C905 NIC, ATI Rage Pro AGP Turbo w/ 32 MB SDRAM	MS WinNT 4.0 Server SP6	ZD i-Bench 1.02, MS Internet Information Server
SBC server	550 Mhz Intel PIII, 128 MB SDRAM, 21 GB disk, 2 3COM 3C905 NICs, ATI Rage Pro AGP Turbo w/ 32 MB SDRAM	MS WinNT 4.0 Workstation SP6, MS WinNT 4.0 TSE, MS Win2000 Advanced Server	AT&T WinVNC 3.3.3r2, Citrix Metaframe 1.8, MS Internet Explorer 5
Network bandwidth simulator	300 Mhz Intel PII, 128 MB SDRAM, 6 GB disk, 2 3COM 3C905 NICs, Matrox G400 Max w/ 32 MB SDRAM	MS WinNT 4.0 Server SP6	Shunra Software The Cloud 1.1
Thin client	300 Mhz Intel PII, 128 MB SDRAM, 6 GB disk, 3COM 3C905 NICs, Matrox G400 Max w/ 32 MB SDRAM	MS WinNT 4.0 Workstation SP6	Win32 VNCviewer 3.3.3r2, Win32 ICA client, Win32 RDP 4.0 client, Win32 RDP 5.0 client
Network traffic analyzer	400 Mhz Intel PII, 128 MB SDRAM, 21 GB disk, 3COM 3C905 NIC, STB Velocity 128 w/ 16 MB SDRAM	MS WinNT 4.0 Workstation SP6	AG Group Etherpeek 4

Table 2 **Hardware/software used in experimental testbed**

4 Measurements and Results

We ran the video application using each of the three SBC client/server platforms. In running the video application, we used the standard settings on the Windows Media Player, which buffers 5 seconds of video before it begins playback. When running the video application using each of the SBC platforms, we used the default configuration settings for each platform. More specifically when applicable options were available, each client had display encoding compression turned on and client disk caching turned off. All SBC clients ran using Windows NT 4.0 Workstation Service Pack 6. We measured each SBC platform with its server running on both Windows NT 4.0 TSE and Windows 2000.

To obtain a measure of the application's performance in a traditional desktop environment as a basis of comparison, we also ran the video application by itself on the SBC server without the SBC software. In this measurement, the SBC server functions as the desktop client and the benchmark server functions as the server. There was no noticeable difference in the results for different operating systems for the desktop case, so we only report the Windows NT 4.0 Workstation measurements. We refer to the desktop baseline as the WinNT Server platform configuration.

A summary of all the platform configurations measured is given in Table 3. To provide a fair comparison, all of the clients were configured to display at 800x600 resolution with 256 colors with the browser window filling up the entire client display. All communication was done over TCP/IP using 100BaseT Ethernet, with available network bandwidth scaled by the network simulator. We first measured the performance of the SBC platforms with server and client connected in a LAN environment at 100 mbps. We then measured the performance of the SBC platforms with server and client connected at bandwidths from 128 kbps to 10 mbps to evaluate how the SBC platforms performed at different network bandwidths.

Name	Client	Server
WinNT Server	Win NT 4.0 server	i-Bench server
RDP WinNT	Win32 RDP 4.0	Win NT TSE server
RDP Win2K	Win32 RDP 5.0, disk cache off, compress on	Win 2000 server
ICA WinNT	Win32 ICA, disk cache off, compress on	Metaframe 1.8 on Win NT TSE server
ICA Win2K	Win32 ICA, disk cache off, compress on	Metaframe 1.8 on Win 2000 server
VNC WinNT	Win32 VNCviewer, hextile on, copyrect on	WinVNC 3.3.3r2 on Win NT TSE server
VNC Win2K	Win32 VNCviewer, hextile on, copyrect on	WinVNC 3.3.3r2 on Win 2000 server

Table 3 **Platform configurations measured**

4.1 Performance in 100 mbps LAN

At 100 mbps network bandwidth, all of the platforms we tested completed the playback of the video clip in roughly 63 seconds, so there was no slowdown in the video playback. However, there was substantial variation in the quality of the video playback across platforms.

As expected, the WinNT Server configuration displayed perfect quality video with no noticeable visual artifacts. Since the video clip is about 24 MB, the effective required bandwidth to display the video in 63 seconds is only 3 mbps on average, well within the bandwidth available for a 100 mbps network connection. Of the SBC platforms, VNC delivered the worst video application performance. The VNC results were equally bad on both Windows NT TSE and Windows 2000. Very little data was transferred from server to client and the video quality was terrible. Of the 1890 frames in the video clip, the VNC server only managed to send the first three frames to the VNC client. All of the other SBC platforms delivered good quality video with no perceivable degradation in video quality compared to using WinNT Server.

To determine if network bandwidth was a limiting factor in performance, we used the network traffic analyzer in our testbed to measure the data that was transferred for each platform. We measured the display data sent from server to client and the control data sent from client to server. The total display and control data transferred for each platform during video playback is shown in Figure 1. These measurements include the packet overhead of the network protocols, which is why the display data transferred during video playback using WinNT Server is larger than the size of the video clip. As expected, most of the data transferred is display data corresponding to the screen updates being sent from server to client. Since there was no user input during video playback, the control data largely consists of simple return acknowledgments for the display data.

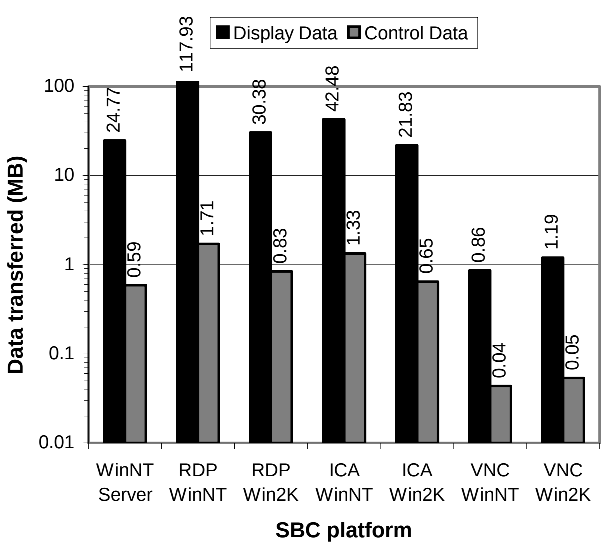

Figure 1 **Display vs control data transferred**

Of all the platforms considered, RDP WinNT transfers the most amount of data during video playback. For RDP WinNT, approximately 120 MB of display and control data was transferred over 63 seconds of video playback. This amounts to only about a 15 mbps average data transfer rate, well within the bandwidth available for a 100 mbps network connection. The other SBC platforms all transfer less data, so the 100 mbps network bandwidth was not limiting the video application performance for any of the SBC platforms.

Figure 1 also shows the relative efficiency with which each platform transmits the video data from server to client. VNC consumes the least amount of data, but it delivers poor video application performance. Both RDP and ICA consume much less bandwidth using the Windows 2000 server compared to the Windows NT TSE server. For the same video application performance, there is a 75 percent reduction in bandwidth consumption for RDP and a 50 percent reduction in bandwidth consumption for ICA. What is more impressive is that the video playback using ICA Win2K consumes less bandwidth than even WinNT Server, in which MPEG-4 encoding is used for transmission. With the exception of RDP WinNT, all of the other SBC platform configurations were within a factor of two of WinNT Server in the amount of bandwidth consumed. For the video clip used in our experiments, the SBC display encodings were surprisingly efficient at encoding the video data compared to MPEG, which was specifically designed for compressing digital video.

To determine if server or client speed was a limiting factor in performance, we used a performance monitor on the server and the client to measure the CPU load during video playback for each SBC platform. The average CPU load for the client and server for each SBC platform is shown in Figure 2. For all of the SBC platforms, the server CPU and client CPU are not fully utilized. The highest CPU load is for the client using the ICA platform, which is just below 90%.

For the SBC platforms, the video player runs on the server and decodes the MPEG-4 video stream before sending it to the client using its respective remote display protocol. Even though the video decoding is done on the server, the ICA client CPU load was higher than the server CPU load. This is in part due to the server being a more powerful 550 Mhz PIII PC while the client was only a 300 Mhz PII PC. However, it was still surprising to see how much work the ICA client needed to do to process the screen updates. Note that the WinNT Server client CPU load is less than the RDP and ICA platforms, but the client used in the WinNT Server configuration was the 550 Mhz PIII PC instead of the 300 Mhz PII PC used for the SBC clients.

As shown in Figures 1 and 2, the VNC platforms on Windows NT and Windows 2000 do not saturate the client, server, or network connection. We also measured memory usage and found that not to be a limiting factor either. The poor performance of VNC appears not to be due to hardware resource limitations, but is most likely due to a poor Windows implementation. In fact, the VNC online documentation confirms that the VNC Windows implementation is less robust than the VNC Unix implementation [18]. We have also measured other applications which show the VNC Unix implementation performs better than the VNC Windows implementation [11]. Unfortunately, the Windows Media Video benchmark does not run on Unix, so we could not directly compare the VNC Windows and Unix implementations using this benchmark.

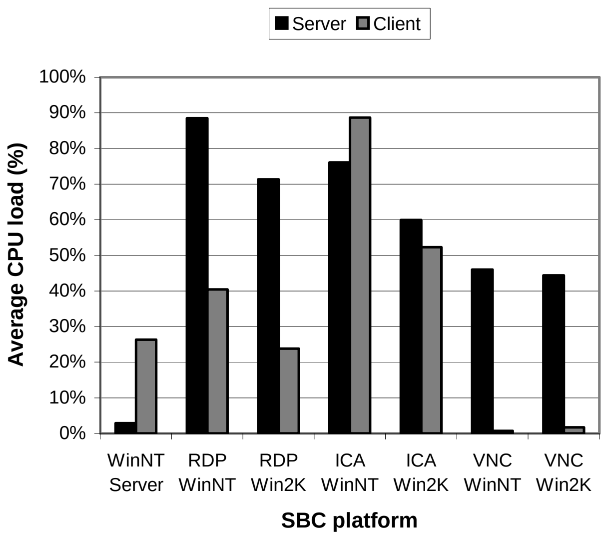

Figure 2 **Server and client CPU load**

To quantify the impact of client caching and display encoding compression, we also measured the performance of the SBC platforms on Windows 2000 with disk caching and display encoding compression turned on and disk caching and display encoding compression turned off. Figure 3 shows the differences in the display data transferred for the three SBC platforms running on Windows 2000 with different caching and compression combinations. For RDP Win2K and ICA Win2K, it was not possible to turn off the RAM cache so only disk caching was turned on and off for those platforms. For VNC Win2K, there was no disk cache, but we turned off the local framebuffer region copies for the no caching case.

For all of the platforms, the use of client caching slightly reduced the data transfer requirements of the video application benchmark. We did not expect caching to provide much benefit as the screen updates for the video application are constantly changing and there is not much reuse. We did find though that both RDP and ICA platforms delivered slightly but visibly worse video quality when disk caching was turned on. Because RDP and ICA are both proprietary solutions, it is difficult to determine the exact reasons why turning disk caching off might improve the visual quality of the video. However, given that disk caching did not result in much bandwidth savings, it is likely that enabling disk caching results in the SBC platforms having to do more work. Instead of just obtaining the required screen data from the server, the client would have to check for the data each time in its local disk cache, realize the cache miss, then obtain the required screen data from the server. On a fast network, obtaining the data directly from the server may even be faster than determining whether an item is in the local disk cache and then reading it from the disk. The slightly worse video quality with disk caching enabled may also be due to a change in the SBC screen update policy that occurs with disk caching enabled. If the screen update policies buffer several video frames on the server side before sending an update to the client, this may result in some video frames being overwritten on the server side and never reaching the client. It is likely that any reduction in the amount of data transferred when disk caching is enabled signifies loss of video data rather than data transfer savings due to cache hits.

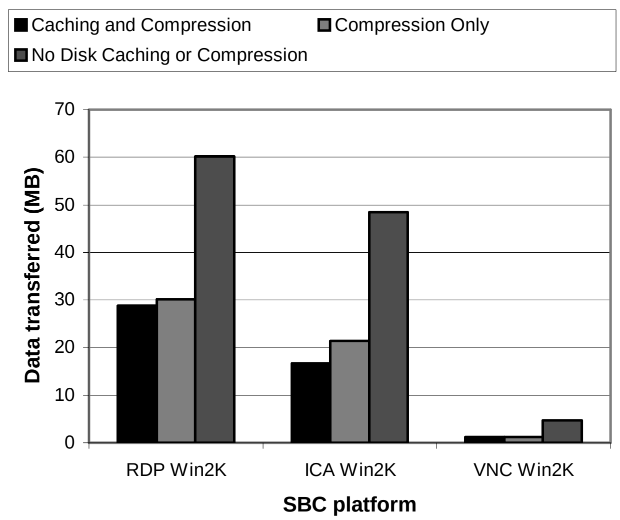

Figure 3 **Effect of caching and compression**

The biggest data transfer savings were from the use of display encoding compression. By default, all of the platforms use compression. To turn compression off, we checked the respective configuration option for RDP Win2K and ICA Win2K and used raw mode encoding instead of hextile encoding for VNC. Raw mode encoding simply sends uncompressed raw pixel values instead of using the 2D run length compression employed by hextile encoding. In all cases, using compression reduced the data transfer requirements by more than a factor of two compared to not using compression in the same SBC platform. The video quality for RDP Win2K and ICA Win2K was equally good when display encoding compression was turned on or off, so long as the disk caching was turned off. The video quality was still good with display encoding compression turned off because the bandwidth consumption with display encoding compression turned off is still well below the available bandwidth over a 100 mbps network. Just as before, the video quality for VNC Win2K was poor for all of the caching and compression combinations tested.

4.2 Performance at Various Network Bandwidths

Moving from a 100 mbps LAN environment to lower network access bandwidths, we observed greater variance in the video application performance for different SBC platforms. Figure 4 shows the video playback time for each SBC platform for different network access bandwidths between server and client. We observe that as the network access bandwidth decreases, the SBC platforms exhibit two kinds of behavior. They either playback the video at a slower rate or playback the video in constant time. A slowdown in the video playback over slower network connections is a clear indication of a degradation in video application performance. However, maintaining the same video playback speed over slower network connections can mean either that the video quality remains the same or that video frames are being skipped to maintain the same overall video playback time.

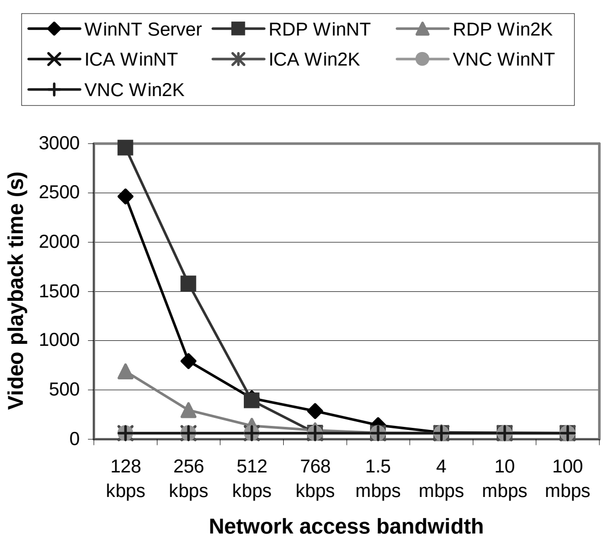

Figure 4 **Video playback time**

Figure 5 shows the corresponding server-to-client data transfer measurements for each SBC platform for different network access bandwidths. Suppose we assume that the data transfer over different network connections should be the same for a given platform if all the video frames are displayed. Then a reduction in data transfer at lower network bandwidths is effectively a measure of the number of video frames that are skipped. We can use this information in conjunction with the data in Figure 4 to evaluate if a constant video playback time over slower network connections is correlated with constant video quality or degraded video quality from video frames being skipped. Our results show that each SBC platform takes a different approach when running over slower network connections. We first describe the observed video performance for each SBC platform and then discuss how the remote display technologies used can result in the observed behavior.

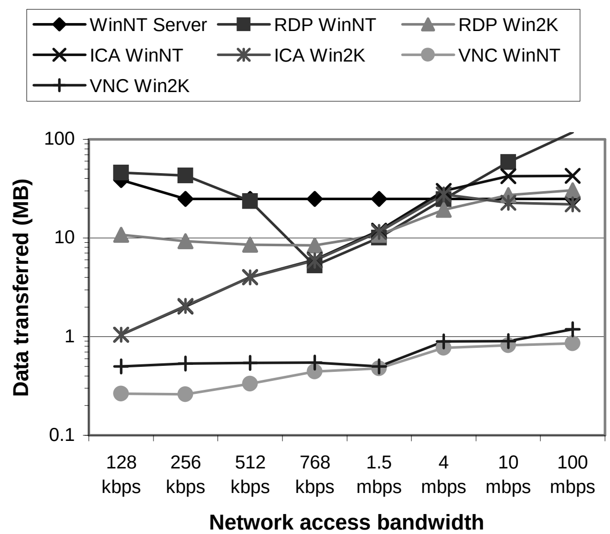

Figure 5 **Display data transferred**

Figures 4 and 5 show that VNC delivers fairly constant video quality for different network bandwidths. Unfortunately, the video quality was poor over all network bandwidths. VNC performed the worst of all three platforms at all network access bandwidths. The VNC results were equally bad on both Windows NT TSE and Windows 2000. Very little data was transferred from server to client and the video quality was terrible. Although the video playback completed with almost no delay in roughly 63 seconds, almost none of the frames were displayed on the client. As discussed in Section 4.1, the performance problems with VNC most likely reflect a poor Windows implementation of VNC more than anything else.

Figures 4 and 5 show that video application performance using RDP degrades in two ways at lower network bandwidths. First, video frames start being skipped while the overall video playback time remains constant. Then when the network access bandwidth drops to 768 kbps, the video playback slows down. RDP WinNT exhibits the unusual behavior of not only slowing down the video playback at low bandwidths, but also increasing the amount of data that is transferred. At 128 kbps, the video playback takes almost 50 minutes and is agonizingly slow, but a substantial percentage of the video frames appear to be displayed. Despite the large reduction in bandwidth consumption going from a 100 mbps network connection to a 10 mbps network connection, the video performance for RDP WinNT at 10 mbps is still quite reasonable, though there is visible tearing in the video clip every few seconds. The video performance is much poorer for RDP WinNT at 4 mbps and is unusable at lower bandwidths. The video performance using RDP Win2K at 10 mbps is quite reasonable though slightly worse than at 100 mbps. The video performance for RDP Win2K is somewhat better than RDP WinNT at 4 mbps, but it is also unusable at lower bandwidths. Overall, RDP on Windows 2000 provided better video application performance with better network efficiency than RDP on Windows NT TSE.

Figures 4 and 5 show that video application performance using ICA WinNT and ICA Win2K is smooth at both 10 and 100 mbps network bandwidths, but degrades at lower bandwidths by skipping video frames to maintain the same video playback time. Citrix ICA had the best overall performance of all three SBC platforms. It outperformed Microsoft RDP despite the fact that RDP is bundled with the Windows server operating systems, which some might argue should allow for better performance. The video quality using ICA degrades slightly over 4 mbps network connections, but the visible hesitations in the video are not easily noticeable. At 1.5 mbps, there is noticeable tearing and frame skipping in the video playback, although the video quality was still usable. When the network access bandwidth dropped to 768 kbps, there were too many skipped frames and the video quality became unusable. Overall, ICA on Windows 2000 provided better video application performance with better network efficiency than ICA on Windows NT TSE.

Figure 6 is derived from Figures 4 and 5 and shows the average rate at which display data was sent from the server to the client for each platform at each network access bandwidth. A curve representing the maximum bandwidth available for each network access bandwidth is also shown for comparison. Figure 6 shows that although VNC performed poorly, it was not limited by network bandwidth in any of the experiments. For the other SBC platforms, Figure 6 indicates that the average bandwidth consumed during video playback was close to the network access bandwidth available at lower network bandwidths. We can see that for RDP and ICA, available network bandwidth was the primarily limiting factor in video application performance for network bandwidths at 4 mbps and lower.

The differences in the behavior of RDP and ICA when network bandwidth is the primarily limiting factor for the video application performance are most likely due to differences in their screen update policies. Both SBC platforms exhibit the ability to skip video frames. This is probably due to the corresponding screen updates being buffered then overwritten on the server side when the rate at which screen updates are generated is greater than what can be sent through the network to the client. ICA appears to use this policy of overwriting old screen updates for all the bandwidths we tested. RDP however appears to slow down the rate at which the server processes screen updates when the network connection between the server and the client gets too overloaded.

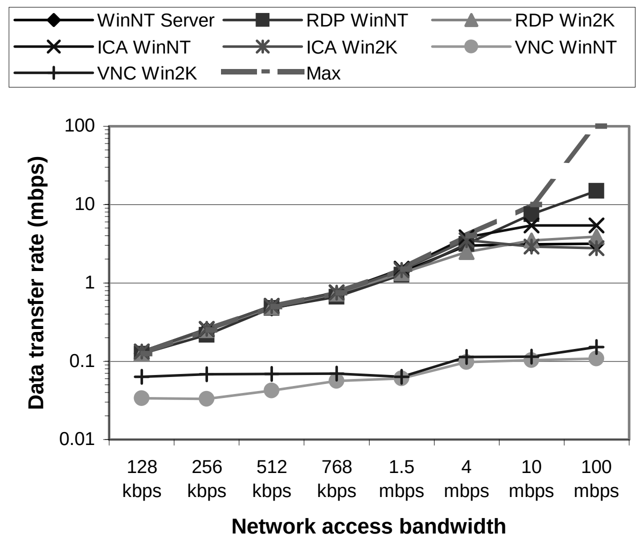

Figure 6 **Display data average transfer rate**

5 Related Work

Many SBC platforms have been designed, but quantitative performance data on these platforms is limited. Little previous work has been done to compare the performance of SBC platforms against one another in supporting multimedia applications. SBC platform vendors such as Citrix and SCO have conducted internal performance testing of their products, but it is unclear how much comparative evaluation they have done, especially for multimedia application performance. Schmidt, Lam, and Northcutt examined the performance of the Sun Ray platform in comparison to using the X protocol [14]. Microsoft has published several white papers on Terminal Server that discuss its performance for purposes of capacity planning [10, 19]. Wong and Seltzer have also studied the performance of Windows NT Terminal Server, focusing on office productivity tools and web browsing performance [20]. Tolly Research has conducted similar studies for Citrix Metaframe [17]. In our own work, we have also compared web browsing performance using both Unix and Windows-based SBC platforms [11, 21].

6 Conclusions and Future Work

Server-based computing platforms offer the potential of lower total cost of ownership. We have shown that the performance of these platforms can vary widely, but that the best of them can already support multimedia video in LAN environments with comparable quality and network resource usage as desktop computing based approaches. However, existing solutions do not perform well for broadband network access bandwidths, which may limit attempts by ASPs to use existing SBC approaches over the Internet.

We have also measured the impact of remote display technologies such as display encoding compression and client disk caching. Our results show that remote display encoding compression techniques can yield significant savings in network bandwidth requirements. However, client disk caching does not provide any bandwidth savings for video applications and may instead degrade video application performance.

We have used video playback to measure the multimedia performance of server-based computing. However, video playback is only one type of multimedia application. We are evaluating server-based computing using a range of multimedia application workloads to determine their effectiveness in a broader context. We are also examining ways in which remote display protocol technologies can be improved to support multimedia applications at the lower network access bandwidths available in broadband environments.

References

1. "Citrix MetaFrame 1.8 Backgrounder", Citrix White Paper, Citrix Systems, June 1998.
2. The Cloud, Shunra Software, http://www.shunra.com.
3. B. C. Cumberland, G. Carius, A. Muir, *Microsoft Windows NT Server 4.0, Terminal Server Edition: Technical Reference*, Microsoft Press, Redmond, WA, Aug. 1999.
4. Desktop.com, http://www.desktop.com.
5. Futurelink, http://www.futurelink.net.
6. GraphOn GO-Global, http://www.graphon.com.
7. Etherpeek 4, AG Group, Inc., http://www.aggroup.com.
8. i-Bench version 1.02, ZD, Inc., http://i-bench.zdnet.com.
9. T. W. Mathers, S. P. Genoway, *Windows NT Thin Client Solutions: Implementing Terminal Server and Citrix MetaFrame*, Macmillan Technical Publishing, Indianapolis, IN, Nov. 1998.
10. "Microsoft Windows NT Server 4.0, Terminal Server Edition: An Architectural Overview", Technical White Paper, Microsoft Corporation, 1998.
11. J. Nieh, S. J. Yang, N. Novik, "The Network Performance of Thin-Client Computing", submitted for publication.
12. Personable.com, http://www.personable.com.
13. T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper, "Virtual Network Computing", *IEEE Internet Computing*, 2(1), pp. 33-38, Jan.-Feb. 1998.
14. B. K. Schmidt, J. D. Northcutt, and M. S. Lam, "The Interactive Performance of SLIM: A Stateless, Thin-Client Architecture", *17th ACM Symposium on Operating Systems Principles*, pp. 32-47, Dec. 1999.
15. Sun Ray 1 Enterprise Appliance, Sun Microsystems, http://www.sun.com/products/sunray1.
16. "Tarantella Web-Enabling Software: The Adaptive Internet Protocol", A SCO Technical White Paper, The Santa Cruz Operation, Dec. 1998.
17. "Thin-Client Networking: Bandwidth Consumption Using Citrix ICA", IT clarity, Tolly Research, Feb. 2000.
18. Virtual Network Computing from AT&T Laboratories Cambridge, http://www.uk.research.att.com/vnc.
19. "Windows 2000 Terminal Services Capacity Planning", Technical White Paper, Microsoft Corporation, 2000.
20. A. Y. Wong, M. Seltzer, "Evaluating Windows NT Terminal Server Performance", *Proceedings of the USENIX 2000 Annual Technical Conference*, San Diego, CA, June 2000.
21. S. J. Yang, J. Nieh, "Thin Is In", *PC Magazine*, 19(13), p. 68, July 2000.