Citrix ICA®
&
Server Based Computing:
Elements of Performance

April 1999

A Quick Review

The Citrix Independent Computing Architecture (ICA) is a general-purpose distributed presentation services architecture. ICA is a Citrix-invented technology that shifts the application processing from client to server. A core technology of Citrix's WinFrame® and new MetaFrame™ server-based computing software, ICA helps I/S organizations reduce total cost of ownership by delivering business-critical applications over heterogeneous computing environments, while safeguarding application performance, data security and administrative control.

With over two million ports in use worldwide, Citrix's ICA technology has become a de facto industry standard for server-based computing and is being broadly adopted by leading vendors for building thin-client computing capability into a broad new range of devices.

Citrix ICA server-based computing technology includes the following equally important components:

- **Server Software Component** - On the server, ICA separates the application logic from the user interface and transports this to the client via standard network protocols—IPX, SPX, NetBEUI, NetBIOS, TCP/IP & PPP. The server also provides important user administration tools, application management and security management.

- **Client Software Component** - On the client, users see and work with the application's interface, but 100% of the application logic executes on the server. As a result, applications consume just a fraction of the network bandwidth usually required.

- **Protocol Component** - The ICA protocol transports keystrokes, mouse clicks and screen updates to and from the client consuming typically 10 kilobits-per-second of network bandwidth. The ICA protocol is comprised of many components – at the heart of it all *Thinwire*. Thinwire combined with things like Virtual Channels all work together to provide a *complete* user experience and improved system administration.

The flexibility of ICA brings manageable, high-performance deployment of Server-based applications to mobile professionals, telecommuters, branch office workers and extended enterprise users over any kind of network connection—dial-up, LAN, WAN and Internet—to any type of client or device. This distributed architecture allows applications to perform at very high speed over low and high bandwidth connections.

Key ICA Differentiators

- **Bandwidth Adaptability** – ICA adjusts dynamically to conditions on the network, client and server that affect performance. Intelligence is a basic component of ICA that takes advantage of performance conditions when bandwidth or system resources are available and economizes when they are not – without compromising the users experience and most importantly without the user ever knowing this is taking place. We can make this happen by caching images, reducing or eliminating unnecessary drawing instructions, and through advanced compression mechanisms.

- **Efficient Use of Client Resources** - ICA functions on devices requiring as little as an Intel 286 processor.

- **Thinness on the Wire** - ICA consumes, on average, 10kb of network bandwidth and is optimized for connections as low as 14.4Kbps. This allows it to operate consistently, even over dial-up and ISDN connections, regardless of bandwidth or application robustness.

- **Platform Independence.** ICA is inherently platform-independent. Its modularity makes it easily adaptable to a variety of operating systems and devices. By IDC estimates, ICA accommodates 95% of the desktop operating systems in use today. Citrix also supports various other real-time operating systems.

- **Universal Client.** ICA works with any Win16 or Win32 application, allowing applications to be developed with off-the-shelf Windows tools and deployed with only one piece of ICA-based client software.

- **Protocol Independence.** ICA is designed to run over industry standard network protocols including: TCP/IP, NetBEUI, NetBIOS and IPX/SPX; and over industry standard communications such as PPP, ISDN, Frame Relay and ATM.

Citrix ICA Clients

Citrix ICA clients support the following operating platforms. These clients are available for download from the Citrix website, located at http://www.citrix.com.

• DOS • OS/2 • Windows 3.1 & 3.11 WFWG • Windows 95 & 98 • Windows NT Workstation • Windows CE (HPC, Jupiter & WBT) • Java	• Microsoft Internet Explorer ActiveX Control • Netscape Navigator Plug-in • UNIX: Linux, Sun Solaris (SPARC & x86), HP-UX, IBM AIX, SunOS, Digital, SGI IRIX & SCO • Macintosh (Motorola and PowerPC)

Additionally, Citrix's Independent Computing Architecture has been ported to RISC OS and QNX OS embedded operating systems.

Citrix ICA Licensees

ICA is being broadly adopted by leading industry vendors. ICA licensees include Microsoft, IBM, Sun Microsystems, HP, Motorola, Sharp, Acorn, Wyse Technology, Boundless Technologies, and a rapidly growing number of other companies for inclusion in new and future hardware and software products that extend the reach of enterprise applications into new markets. These and future relationships are establishing ICA as the de facto standard for server-based computing.

Windows-based Terminal Manufacturers

- Acer
- Addonics (ART 2000)
- Boundless Technologies (Viewpoint TC & Capio)
- Bryant Computers (Sunnix 2000)
- Motorola Semiconductor Products Sector (WinCept 100/100 WBT reference designs)
- Neoware Systems, Inc. (NeoStation and netOS)
- Netier Technologies, Inc. (NetXpress SL and XL)
- Plexcom, Inc. (QuantumNet)
- TECO Information Systems (Relysis TR-3300)
- UMAX (NC-200 and NC-320)
- VXL (Winlinx)
- Wyse Technology (Winterm)

Network Computer Manufacturers

- Aranex (Internet Client Station Series 2000)
- IBM (Network Station)
- MTX, Inc. (MTX 1683 Network Computer Terminal)
- Sun Microsystems (Java Virtual Machine)

Handheld PC/Wireless Manufacturers

- Compaq (Series C)
- Hewlett-Packard (HP 620LX & 360LX Palmtop PCs)
- Motorola (Telephones)
- Sharp Electronics Corporation (Mobilon line of handheld PCs)
- Symbol Technologies (PPT 4300 portable pen terminals)
- Telxon Corporation (wireless, ruggedized pen-based computers)

Information Appliance Manufacturers

- Acorn Group plc (DeskLite reference design)
- Boca Research, Inc. (BocaVision set-top box)
- Clientec (Computer keyboards)
- Key Tronic Corporation (Computer keyboards)
- Websonic (universal communicators)

Operating System Suppliers

- Caldera
- NCI
- QNX
- Red Hat

This list constantly growing as more device and software manufacturers join the opportunities created by Citrix server-based computing.

Performance Measurement

Measuring performance is not quite the same as what might typically be considered under other legacy methods of computing. In the case of ICA, application activity does not occur on the user's desktop, but this is where the user experience takes place. Measuring this computing mechanism brings with it numerous challenges because it involves all of the components described on the first page of this document; the server, network, and the client desktop. Measuring any one of these items may actually be easier than the very interactive and dynamic combination of all of these components.

To help you understand the ICA performance spectrum, Citrix has created the material below to outline what needs to be considered. We begin by trying to outline what users experience, then defining what "performance" actually is using ICA or any other remote presentation protocol, like RDP. We conclude with some helpful information to improve the accuracy of measurements.

What's important to the Customer?

Citrix customers all share the following items in common relative to their expectations from our MetaFrame and WinFrame server-based computing solutions and the client-side ICA interaction with this software:

1. Usability

- Responsiveness and perceived speed of operation when performing typical tasks and features providing a PC-like experience.

2. Efficient use of the Network

- Minimizing the amount of data and number of packets generated.

3. Client-side Resource Utilization

- Minimizing client-side CPU and memory utilization.

4. Server-side Resource Utilization

- Minimizing server-side CPU and memory utilization. Gathering data to indicate server resources required and maximum supported users (scalability).

Does the Customer Environment Affect Performance?

YES. Remember that all of these items interact to provide the user with their "performance" experience. During measurement of performance, all points above need to be measured with different:

- Network bandwidths,
- Numbers of users,
- Applications,
- Interactive operations with other servers or services (e.g., back-end DB servers, the Internet, etc.)
- Target client types,
- Server configurations.

Elements of "Performance"

Expanding on the previous page, the following items elaborate on the elements of performance. Each and every one of these items contributes to the user experience. Depending on who you are (e.g., the person using the application, the IT manager, the network manager, etc.), the focus may be on different components. But the composition of all of these items is important to the collective business enterprise. For example, if perceived user performance being equal between a Citrix solution and an alternative occurs using a fraction of the network resources, then performance is not equal. Understandably, the same result using fewer resources does so more efficiently and therefore with greater performance.

Speed

Measured as ***time*** to visibly complete a given set of actions and benchmark reported figures. Time to visibly complete the operation is used because the server can be fooled into thinking that a set of display operations has completed but the client still has to display most of the data. This renders server-side measurement tools inaccurate.

Network efficiency

Network efficiency for a given set of operations are measured as:

- Total **KB** transmitted to and from the server. This determines when the network will saturate.
- Number of **packets** used to transmit that data. High numbers of packets will cause collisions and network saturation at a lower bandwidth and reduce the available bandwidth.

NOTE: It is important to realize that the amount of data and packets sent/received will change depending on the other performance elements. There are frame or graphics dropping techniques that will reduce data transmission but sometimes at the expense of jerky graphics presentation. The recommendation is that a varying number of configurations be tested before making statements on bandwidth usage. The number of packets transmitted could be lower but possibly at the expense of responsiveness

Server load

Server load includes memory usage and CPU usage. This should change as the number of active sessions increases. For instance, it may be fine for a single session to utilise 50% of the server CPU, when it is readily available, but not okay when many sessions are active. *Don't assume linear usage of server resources based on single session data.* Especially due to multi threading on multi CPU's and the memory caches. Also, low CPU usage may not be efficient if the time it takes to complete a task consequently increases.

Both WinFrame and MetaFrame leverage the economy of scale achieved by using shared memory resources. By benefiting from existing memory contents, duplicate images are reduced. Memory usage rarely scales linearly with the number of users.

Client load

Client load includes memory & CPU usage. These can be effectively measured on some devices; for instance Windows NT & UNIX but not all.

Responsiveness	This refers to how quickly the results of a given user action produce the resultant change in displayed graphics (latency). A good example is after clicking on a menu button how long it takes for the entire menu to be displayed. A bad example is how long it takes a file to be loaded and displayed since the loading time is not greatly affected by the client/server solution. Some aspects of responsiveness are objective and not easily measured. For instance the way in which a large bitmap is displayed may be pleasing or displeasing and lead the user to "feel" that one client is more responsive than another even though they both continue on to display the next graphical element after the same elapsed delay. As mentioned above under *Network Efficiency*, some techniques involve throwing away data in order to meet low bandwidth requirements but these lead to jerkiness in interactive operation although producing good benchmark figures.
Total user experience	The "feel" experienced when using applications software should be similar to, or better than, that experienced when running the same application on the server console. To what extent is the entire server console environment reproduced for the client user? This includes device support, port support, audio and integration features such as cut/copy/paste.
Effect of Network	How do all of the above factors vary between high bandwidth LANs and low bandwidth WANs? Also, what affect do various settings have using a client that can be tuned for the bandwidth utilized?

Common Performance Measurement Pitfalls

- ***IMPORTANT!*** Many of the typical benchmarking programs available today were originally designed to test the performance of a single user PC system (e.g., WinTach, WinBench, Winstone, Sysmarks, etc.). ***However,*** In a server-based computing environment where client, network, and server are all elements being measured, these benchmarking applications are not very well suited for measurement of remote display protocols. Instead, these may actually be a better measure of server performance (which is quite variable) rather than client or protocol performance. At this time, there are no real server-based computing benchmarking applications available to help overcome this issue.

- All figures tend to suffer from "noise". This is partly due to the fact that operating systems such as NT will use resources differently dependent on what was run on the system previously or even the exact point in time after a reboot that a given operation is performed, mostly due to caching. It is therefore important to take an average of multiple runs. It is also important to reboot and perform the same series of operations for each run. When enough multiple runs are taken you can then use statistics to determine if the difference seen in the different setup is due to chance or not. Please note that even minor differences in configurations can make a dramatic difference in the results due to caching algorithms.

- In real user environments users often do perform the same operations time after time. It is therefore also important that repetitive tasks are performed to see if the system "learns" from experience.

- Short benchmarks are very easy to fool with large buffering systems. A minimum test of at least 5 minutes is recommended.

- Benchmarks that use normal applications often have a fixed overhead of time during which files are being loaded or calculations being performed. These are not affected by the client or protocol used and can therefore make 2 clients appear closer in performance than they actually are.

- Performance can come at some cost to other parameters. For example, a client that performs twice as fast as another might consume a disproportionate amount of additional server CPU in doing so. One would consider a system superior that gives an improvement in performance without a proportionate increase in resources usage.

- *Never* rely on benchmark statistics. The benchmarking software often thinks that operations have finished as soon as the server based client driver tells it. It may well be some time before the client user sees the end of the test. Check visually whether the test has finished on the client and use hand timings to check correlation with reported benchmark numbers

Performance Testing Set-up Recommendations

The information below is intended to help conduct performance testing in a manner that provides consistent, reliable and credible results.

Server Setup:
The server should be setup to be as independent as possible from other machines (i.e., it should be a stand alone server-using local users not domain users, it should not require any files from a file server, where possible all the default options should be taken in the setup). Each client connected should be their own user. Another server component of performance affecting testing outcome is the presence of applicable hotfixes and service packs – the server should be as up to date as possible.

Benchmarks to use:
As stated earlier, the proper benchmarking tools do not yet exist from the major benchmarking labs like Ziff-Davis or National Software Test Laboratories (NSTL, a division of CMP). So these recommendations are made for utilizing the alternatives (e.g. WinTach, WinBench, ISAT, etc.).

Which networks to use:
We recommend using a variety of connection mechanisms as part of performing any tests because results will vary for each of them. Connections should include at least the most common:

- 10Mbps LAN
- Modem
- WAN (if available, but please note the issues involved here as discussed in the first item above relative to introducing "noise.")

How to measure speed:
If automatic client and server side time stamping is possible for determining this would naturally be more accurate. If not, then a stopwatch based on the client display actions, noting reaction times. Benchmarks that are long enough mitigate most reaction time discrepancies by making any margin for error a relatively small percentage.

How to set up the client:
Default options should be used were ever possible. If tests are applicable to low bandwidth environments, then appropriate WAN mode settings should be applied.

How to ensure that you have the latest ICA client version:
The Citrix "Download" website contains the most recent versions of all of the ICA clients available. Please use this to obtain any clients used for benchmarking analysis. Please include references to this version number in any written material.

How to measure server load:
Server load can be measured on the server itself. This adds a small extra load but that can be measured and subtracted if necessary. This can be better than trying to measure it remotely and generating extra network traffic. The Citrix Resource Management Service product can provide an excellent source for this kind of information particularly over long-term test periods.

The Server load data should not be recorded to disk as this puts a heavy load on the server. Ideally the data should be collected automatically over the period of time for the benchmark. It should also be an average for the whole benchmark. NT's performance monitor is not suitable to collect the data because of the significant CPU overhead this tool adds on its own. Citrix uses a modified version that collects individual values and keeps a running total and averages the collection period. Data should be collected every second to capture short-term peaks in the average. Data should be collect on a per user basis. This permits analysis separate of the load associated with a user's action versus the load on the console due to the data measurements, and the system processes load.

The items measured should be measured using a "total system CPU load percentage" measurement so that a correct measurement is made even when there are multiple processors in a system.

How to measure client load:
This is best done in a similar manner to the server load measurement and therefore can be done best on an NT based client.

How to measure network traffic without affecting the actual figures:
This is best done on a separate machine with the network card in promiscuous mode. This permits receiving all network traffic on the network segment without interference. You should not try and save or real-time decode the network data, instead just keep a running total of the amount of data/packets sent during the benchmark. This is best achieved with something like Network Generals Basic Sniffer Network Analyser.

How to run the benchmarks:
Ideally there should be a reboot of the server and client after every run, with the mouse pointer in the client as it would be if a user were using the client. The mouse pointer should also be moved by the benchmark **and all user input should be generated from the client side**. However, this makes it difficult to synchronise the client with the server automatically (e.g., responses to items like yes/no, menu pull-downs, etc.).

How to measure responsiveness:
Responsiveness should be measured by inputting user actions at the client and timing how long the associated graphics change takes to happen. This should measure very basic types of user actions like typing, menu operation, scrolling, maximizing, and minimizng within real applications.

How to measure non-graphics performance:
Disk I/O is easily measured with the Ziff Davis Winbench tool which has built-in disk tests that simulate real applications that can be used on drive remotely mapped to the clients, network drives and the servers local drives. Other types of non-graphical performance is difficult to measure as there are no standard techniques defined and the performance varies by the type of printer and printer driver used.

Reporting Recommendations

A time will come to document any extensive work conducted as a result of this document. As indicated earlier on, the variability of configurations makes every set-up unique. As a result, details on the configurations should include the following:

- Network configuration (protocol, speed, hardware components, servers attached, clients attached, etc.)
- Server configuration (CPU, memory, bus architecture, cache, processors, subsystems, etc.)
- Client configuration (Hardware configuration, operating system, version, etc.)
- ICA client type used, version number and build number