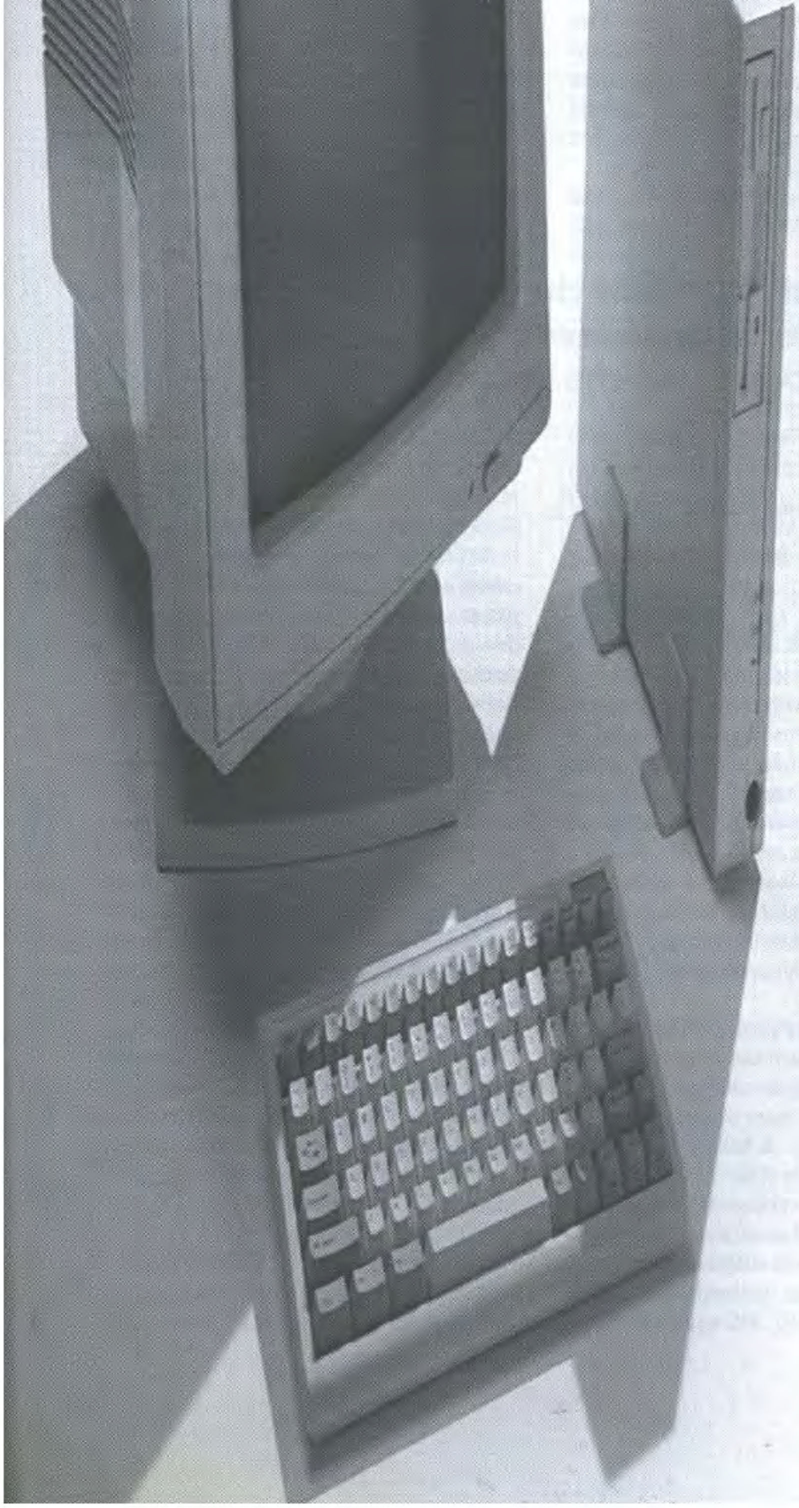


Client

and Network-Cent

Computi

by Mark Sheehan

There's a new presence on the desktop these
Unlike green-screen computer terminals, the
machine packs a screen full of graphics. And u
big, boxy "tower" PCs, the new machine may co
a slim, futuristic shape. At first glance, the newc
performs much like a PC, but offers an array of
tions drastically unlike the simple terminal.

These new machines are called "thin clie
They're thin because they're often very mini
configured. Most of them have no disk drives
because they do no long-term local storage. So
them have little more memory than it takes to
a video screen because that's their only job. A f
them contain processors—286s for example—
familiar in computer museums than in the corp
cubicle maze.

computer specialist. System managers to run the machine and consultant/trainers to help neophytes use it. Overnight, the PC explosion transformed every computer user into a poorly-trained system manager. More than mere consultation and training were required to support the PCs and their networks. System managers with people skills—don't laugh, there were a few even then—took on the job of supporting PCs, LANs, printers, and all the other complex, buggy, failure-prone peripheral equipment and software that the market produced.

The PC may have hurt the mainframe market, but not because PC environments were cheap. True, with PCs a small company could get into computing fairly inexpensively. But *staying* in computing was expensive beyond anyone's expectations. It's a tribute to the resiliency of the human spirit that so many of us—computerphile and computerphobe alike—got into computing and have stayed there.

Most of us think of the cost of a new computer being $2,000-$3,000, with a useful life of three or four years. The Gartner Group in late 1997 [1] estimated the true, total cost of ownership (TCO), including purchase, administration, technical support, and end-user operations (i.e., "futzing"—not Gartner's term), to be more like $10,000 for a networked computer. That's $10,000 per year!

(Of course, a lot of output was diverted to the printer, too.) The expectation of access to the mainframe via terminal from remote offices or from home brought us the beginnings of computer networks.

Terminals started out "dumb." They put letters and numbers on the screen in fixed rows and columns.

...we began to resent the need to tie our terminals to mainframes through copper umbilical cords and the industry gave us th

Writing in Spanish? Sorry, no accent marks or upside down question marks available. Writing equations? Sorry, no math symbols available, let alone super or subscripting. Drawing pictures? Unthinkable!

Later, "smart" terminals overcame these obstacles. Soon thereafter, with our expectations raised and our individuality beginning to assert itself, we began to resent the need to tie our terminals to mainframes through copper umbilical cords and the industry gave us the PC. Suddenly the mainframe was obsolete as a general purpose computer. Sales fell off as the information producing and consuming markets tooled up to create islands of desktop automation—islands of PCs.

Immediately, of course—because we love to socialize almost as much as we love our independence, and because our managers love to economize—there sprang up a hundred-headed Hydra of network technologies aiming to link these isolated PCs back together. Local area networks sprang up within offices and buildings. Wide area networks sprang up too, linking PCs in distant

buildings, distant towns, a
tually distant countries.

To unify network users a
fountain of functionality, th
computer was born. Early
gave scattered PCs access t
printers and files. They in
ated, giving and denying
providing security and back

incidentally, creating one m
of failure. See "A Pig Ur
Hearthrug: The So-Called
Costs of PCs" sidebar.

All this time, the ma
continued to be a force
processing. As mainframe
down into economical mini
and as network servers sca
do more mainframe kinds c
the distinction blurred. It w
further by the fact that ma
of special-purpose personal
ers had become as powerfu
servers and some main
Mainframes of whatever siz
ued to do what mainfra
always done. They se
database repositories. The
data too sensitive or softv
complex or too expensive to
PC. Throughout the histor
PC, terminal emulation softv
a standard component; it wa
buy a $2,000 PC without als
ing software to turn it int
terminal.

CLIENT/SERVER COMPU

Well this relationship
PCs and mainframes was s

mainframe or something like it—was the server. The client had its own special software: software that crunched numbers, for example, or stored and managed a personal electronic mailbox. And the server had its special software too: software that managed multiple clients' access to a database of numbers, or that received incoming mail messages for a hundred users and held them until the client software came to fetch them. With client/server computing, the PC and the mainframe entered into a partnership, each doing what it did best: the PC interacting with the user, storing personal files, controlling local printers; and the mainframe managing data, storing shared files, and controlling shared peripherals. Linking the two partners was a data network. The more active the partnership, the bigger the data pipe needed to link the client and server. While networks had been important since the invention of terminals, they became absolutely pivotal in the client/server model of computing.

Most client software nowadays is written for "fat" or (my preference) "thick" client computers. In many cases, the thicker the client machine is, the better. At Montana State University, for example, we are implementing SCT's Banner2000 suite of administrative software packages. The desktop client computer SCT recommends for the average user is a 233MHz Pentium II with 32MB of memory, a 3.2GB hard disk, and a 100Mb Ethernet adapter. (This is a very thick client drinking data through a fire hose!) In an environment that has been based on cheap, dumb terminals connected to

network connections need to be upgraded to meet the application's need for speed. Total cost of ownership (TCO) figures are genuinely frightening.

BUT, SOFT! WHAT LIGHT THROUGH YONDER WINDOW BREAKS?

Some see a glimmer of hope in the thin client. Stephen Bell introduced *ONLINE* readers to thin client technology last year [2]. Just to recap, a thin client is a kind of minimalist computer. There are two basic types:

When the network computer is switched on, it sends a boot request to the server. The server responds by downloading the operating system into the NC over the network.

the network computer (NC) and the Windows terminal (WT). They're so different that each deserves a description of its own. A third type of thin client, the NetPC, is no longer a player in the marketplace [3]. Because NCs and WTs require networks to function, their kind of computing is referred to as "network-centric" computing.

Network Computers

Network computers (NCs) are little more than a processor chip, some memory, a screen, a mouse, and a keyboard. They connect through a network cable to a server computer. The server houses the thin client's operating system and application software, as well as the files belonging to the user and the user's workgroup. When the NC is

user's task is completed, the files are saved on the server and NC's memory becomes available another application.

The applications that NCs run of two basic kinds: native applications written for the NC's proces or Java applications written to under Java Virtual Machine s ware that is downloaded to the over the network. Java applicati are by far the more common of two types. Once written they run on a variety of NCs. Nat applications, by contrast, must created in a different version each type of NC processor. Her they are more expensive for ma facturers to create and maintain are less flexible for server manag to deploy.

Most NCs are very thin inde with regard to disk resources. Bu the NC concept has matured, network, memory, and proces requirements to run fully featu Java applications have grown ou hand. At least one manufacture NC, configured with its optio hard disk drive, is physically dist guishable from a classical th client only by its lack of inter expansion ports (and definitely by its price!).

Manufacturers of NCs include S Microsystems (JavaStations), I and its partner NCD (IBM Netw Station, NCD Thinstar), Neow programs that are running on a server computer. There is no standard for what a WT's components are. It can be anything that supports RDP or ICA protocols. Some NCs do support those protocols, so that's how an NC can be both a Windows terminal and a network computer!

...the WT client simply sends keystrokes and mouse movements to the server... The programs...execute entirely on the server, using the server's processor, memory, and disk.

(NeoStations), and Acorn Computers Ltd. (Acorn Corporate NC) [4].

Windows Terminals

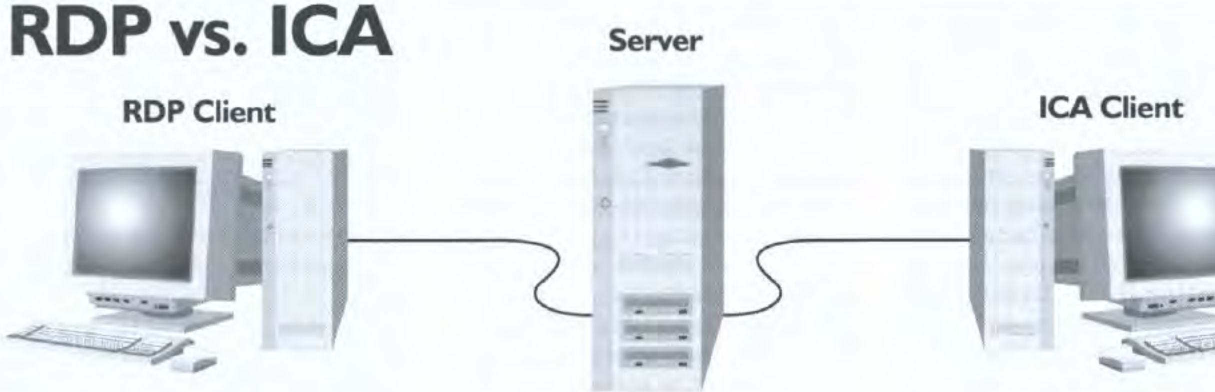
A completely different model of thin client computing is embodied in the Windows terminal (WT). Rather than downloading everything from a server and running it locally, the WT client simply sends keystrokes and mouse movements to the server. The client then displays what the server does with those inputs. It does no processing other than that needed to paint the screen. The programs a WT launches on the Windows server are ordinary Windows programs (newer 32-bit programs work best). They execute entirely on the server, using the server's processor, memory, and disk.

protocol (RDP) or the ind
computer architecture (ICA
RDP or ICA client softwa
brought to bear in three wa
reside on a chip inside the
as it does in the new, dedi
products and some NCs;
downloaded into the mem
NC and executed there; or
loaded onto the disk o
Macintosh, or UNIX works
execution on demand.

RDP is used when th
running a recent version of
soft operating system (Wi
and 98, Windows NT Wor
Windows 3.11) or when the
has been designed specif
support RDP (as some NCs
cated WTs do). ICA is used
machines running DOS, UN
Macintosh operating system.

RDP and ICA server softv
on a dedicated Window

RDP vs. ICA



PC running:
- Windows 3.x, Windows 95/98, or Windows NT client
- Some network computers
- Some Windows terminals

Microsoft NT server running:
- NT v.4 Terminal Server Edition, or
- NT v.3 with Citrix WinFrame for RDP clients and Citrix MetaFrame for ICA clients

- PC running DOS
- Macintosh
- UNIX workstation
- Some network compute
- Some Windows terminal

host-based applications on dedicated X terminals and on computers running X terminal software under UNIX, Macintosh, and Microsoft operating systems, among others.

The difference, of course, between X and RDP or ICA is that when you use X to access host applications, what you access are UNIX applications. While not quite mainstream for most of us, UNIX applications are common enough in a number of industries that many NC and Windows terminal manufacturers are now building X Windows System support into their products.

machine. This server's only job is to serve WTs, offering a selection of Windows applications to all WTs authorized to use them. Because these servers run Windows applications for their client machines, they must be Intel or Intel-like computers. The server software that supports RDP access under Microsoft Windows NT is called Microsoft Windows NT 4.0, Terminal Server Edition. The server software that supports RDP under earlier versions of NT is called Citrix WinFrame.

Citrix also supplies add-on NT server software, called MetaFrame, which supports ICA access. (For a while, Insignia Solutions, Inc. sold an ICA-based product called NTRIGUE! that delivered Windows applications to NCs and other terminal devices. In the first quarter of 1998, Insignia sold its NTRIGUE! Technologies to Citrix Systems, Inc.)

MetaFrame also provides such functions as load-balancing when a large farm of NTs is set up to share several NT servers, and provides a Windows95 taskbar to clients not running Windows95.

Microsoft considers the top five vendors of WTs to be Wyse (Winterm series), Tektronix (ThinStream), Boundless (Viewpoint series), NCD (Thinstar), and Neoware (@workstation series) [6].

COSTS AND BENEFITS

NCs and WTs can both address the high total cost of PC ownership. This is less true if we use Citrix MetaFrame to make WTs out of Macintoshes, fully configured new or old PCs, or UNIX workstations, because those machines can be tinkered with more by their owners—thought to be a major TCO factor in desktop computing. But if we stick to NCs as NCs or if our WTs are simple, limited devices such as NCs or dedicated WTs, they can definitely help us minimize ownership costs.

First, NCs and dedicated WTs are less expensive to purchase than most PCs. NCs cost from $500 to a little over $1,000 plus monitor. Dedicated

The greatest virtue NCs and WTs have in the management and support arena is that their software environments are stored entirely on centrally controlled servers.

numbers, the trend is clearl
Why is this so?

The greatest virtue NCs
have in the managem
support arena is that their
environments are stored e
centrally controlled servers
most NCs have no disk d
average NC user can't b
ware or data from home on
and thus can't introduce
other system complications
expensive to combat. Nor
download software (other t
Applets) from the Internet
it on their desktop compu
the attendant risks of vir
system software conflicts.
WTs are similarly constrain

With NCs and WTs, w
enterprise decides to upgra
next version of a software
the change occurs in one
the server. There's no nee
every desktop to install the
(and then visit it again and
figure out why the new
doesn't work with that p
set of hardware and so
Because NCs and WTs
with a controlled, limited s
ware, training and end-use
requirements are reduc
because they are fairly id
these devices are hard for
to mess up [8]. It's difficul
emphasize this benefit. I
education [9], and almos
where else computers are u
to reduce the cost of suppor
vidually managed PC envi
have been the subject of
debate.

Another common benefi
and WTs is that they hel
around the obsolescence iss

DOCKET ALARM

Explore Litigation Insights

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

Real-Time Litigation Alerts

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

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

Advanced Docket Research

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

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

Analytics At Your Fingertips

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

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

API

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

LAW FIRMS

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

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

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.