31 Thin Client Technology for Windows-based Servers

If you are not familiar with the book check the introduction to "More Tasters" on the web site http://www.b.king.dsl.pipex.com/ to understand the scope and objectives of tasters.

Thin display technology has been around since the mid 1980s with the advent of X in the Unix world. Although there are marked differences in architecture and implementation Citrix introduced a Windows equivalent called WinFrame in 1995. A subsequent licensing agreement allowed Microsoft to produce Terminal Server – NT edition in 1998. The technology has proved to be very popular, particularly in the public sector. This taster concentrates on the Windows variant, basing the majority of the discussion on Terminal Services for Windows 2003. I may add observations on X in a later version.

31.1 Starting Terminology

RDC (Remote Desktop Client) is the software that runs at the client end

RDP (Remote Desktop Protocol) runs between the remote client and the Terminal Server

Roaming Profiles allow users to log on to and use different Terminal Servers, potentially in dispersed geographical locations. The user's profile is stored centrally and copied to the required Terminal Server during the log-in process

UDP (Universal Print Driver) runs on the server. It provides a "low common denominator" driver that can improve printing performance by reducing network bandwidth requirements

Working set is the subset of the total memory requirement for a process that currently resides in physical memory. The size of a working set will vary during the life of the process depending on the pressure on physical memory.

31.2 Technology Outline

In essence, thin client technology allows software that was primarily designed to run on a dedicated PC to be housed on a server that can support multiple users. The user requires only a simple and hence relatively cheap device that can provide the necessary means of input and output, e.g. keyboard input, mouse clicks, and screen display. These inputs and outputs are conveyed between client and server by means of the RDP protocol.

31.2.1 Server

The basic software architecture, as shown in Figure 31-1, comprises:

- The original vanilla-flavoured version of Windows NT was designed to support a single graphical user interface. Modifications were required to support multiple user interfaces; they included changes in the areas of memory management and object management. For those who are interested there is an excellent article on the

winntmag.com web site called *Inside Microsoft Terminal Server* by Mark Russinovich that provides a detailed rationale of the required changes.

- A Windows Service called Terminal Server Service constitutes the controlling core of the Terminal Server, keeping track of active and inactive connections and initiating the creation and disposal of connection contexts

- The Session Manager performs the hard work of creating the processes that are required by a user session, and it provides a user interface to control sessions and display statistics

- Users connect to the system via connection ports.

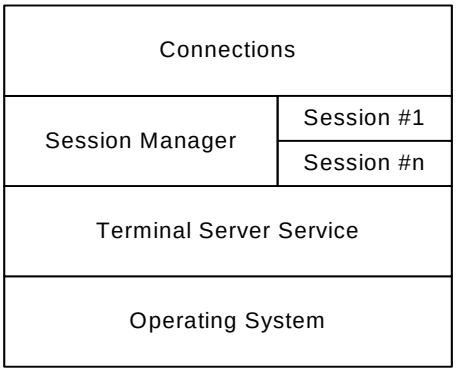

Figure 31-1 Outline Architecture for Terminal Server

There are other third-party Terminal Server offerings, e.g. Citrix MetaFrame and Canaveral iQ from Tarantella. It should be noted that these products sit on top of Microsoft Terminal Server, i.e. the Microsoft product is a prerequisite.

Figure 31-2 shows the various components that Terminal Server needs to communicate with. Much activity occurs at user logon. Domain controller, licensing server and session directory are short accesses, relatively speaking. Session Directory contains details of which user session is on what terminal server. Of particular note is the use of roaming profiles. A user's profile can include *inter alia:* individual desktop settings, My Documents folder, temporary Internet files and registry settings. On a system where a user's session may reside on different servers from day to day, or indeed from session to session, it is necessary to gain access to the profile with the latest consistent information. A typical method is to maintain a master version of the profile on a central server. At logon this profile is copied to the terminal server that has been assigned to the user for this session. At logoff the updated profile is copied back to the master. Apart from roaming there are in fact three other methods of storing profiles: local, i.e. permanently on one terminal server with a resultant lack of resilience, consistency and flexibility; mandatory roaming where updates to a profile are not copied back to the "master", resulting in the performance hit at logon but not at logoff; and finally, there is a hybrid mechanism, through the use of scripting, that allows specified parts of the profile to be updated and prevents other parts from being updated.

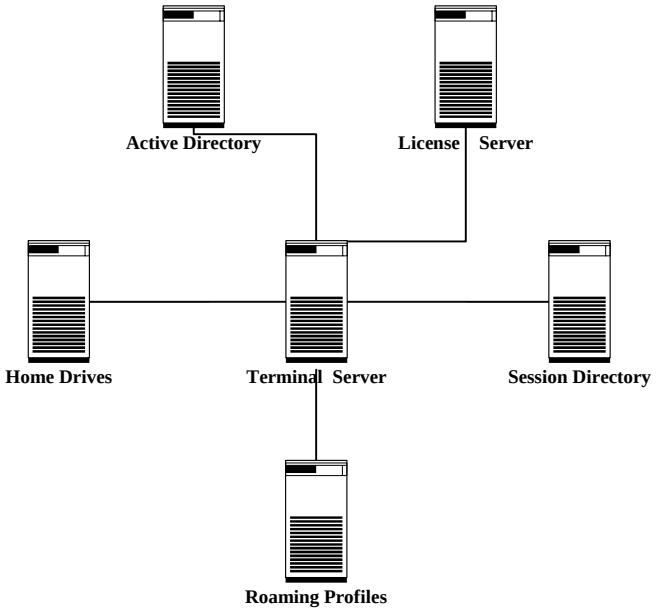

Figure 31-2 Logical View Of System Components Accessed

In a system with multiple terminal servers some form of load balancing sessions will be required. The Microsoft product has no inbuilt features in this area. Microsoft typically recommends the use of NLB (Network Load Balancing), its own general software-based load balancing facility or the use of a hardware load balancer (see Chapter 27 – Server Load Balancing for a general discussion of this technology). Citrix and Tarantella both have inbuilt load balancing features. No product currently supports or is likely to support fail-over clustering; this is mentioned purely because some vendors talk glibly about "clustering". The usual method of providing high availability is to configure N+1 servers. If a server fails, the users can logon to the spare server. However, the spare needs access to the failed server's data, which means that some form of shared access storage across the terminal servers is required, e.g. a SAN or a NAS.

31.2.2 RDP (Remote Desktop Protocol)

Communication between the client and the server is effected by RDP. This protocol, which runs on top of TCP/IP, is based on the International Telecommunications Union's (ITU) T.120 protocol, a standard for multimedia conferencing. As shown in Figure 31.3, it supports the concept of multiple virtual channels (up to 64,000). Additional protocols can be "Plugged-in", e.g. Citrix's ICA.

RDP has been gradually improved since its initial introduction, gradually closing the gap on ICA, which has been seen as a better performer. From a performance perspective, the improvements include: better compression (including print data compression); improved handling of low bandwidth connections by reducing the number of discrete interactions between client and server; and persistent bitmap caching (to a local client-side disk).

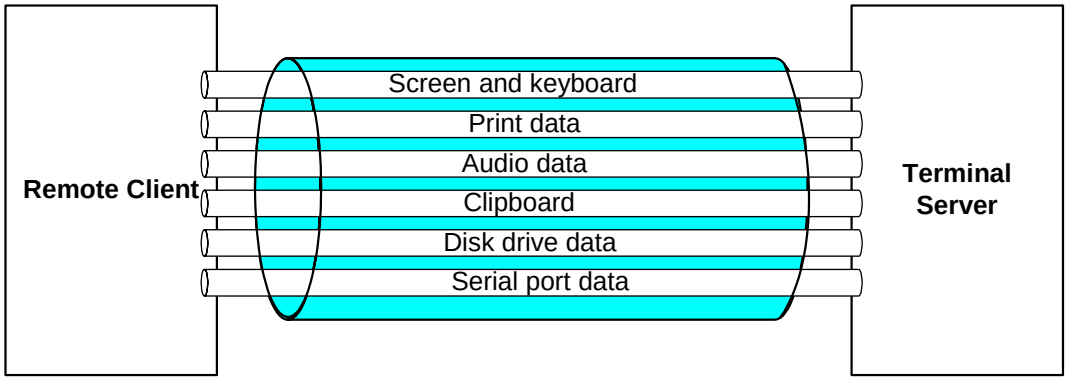

Figure 31-3 RDP and its Virtual Channels

31.2.3 Client

RDC software clients can come in two basic forms: the standard Microsoft RDC client that comes with Windows XP; and third-party offerings that are implemented in Windows, Linux or Java. They can be implemented on a standard PC, possibly to provide a mixed standalone PC / server-based solution; or on a low specification PC where the vast majority of the workload will be server-based; embedded in thin client hardware devices where usage is all server-based; or on portable devices, e.g. running under Windows CE. There are a plethora of options and vendors.

31.3 Performance

There are many areas where performance issues can arise:

- **Memory** is arguably the area where the first bottleneck is likely to occur. Physical memory can quickly become oversubscribed, although it obviously depends on the requirements of each user. At the lower end of the scale a Microsoft Word user may require a working set of ~25MB and an Internet Explorer user ~20MB. Real world applications are likely to require significantly more, possibly in the range 50-100MB. The working set size of an application instance will be reduced by the operating system in an attempt to fit all users into the available physical memory. However, these reductions will gradually but inexorably affect performance as the overheads of the memory management system increase while it tries to keep an ever increasing number of “balls” in the air. In theory, code elements can be shared, as they are read-only, i.e. only one physical copy is required in memory, and this can help to reduce the pressure on memory. However, in the Windows operating system items such as DLLs are not strictly read-only, they are written to, which leads to multiple copies of sections of a DLL. TScale is a third-party product that helps to reduce the impact on memory management of these multiple copies and thus allows more users to be supported. In addition, Terminal Server is quite hungry in its need for memory within the kernel; in part this is attributable to the requirements of the memory management system to support large user populations, e.g. the number of page table entries. On a 32-bit system the virtual memory is split 50-50 by default, 2GB for user mode and 2GB for kernel mode. There is a facility (/3GB) that allows the user side to have more than

50%; while this feature may be useful for software such as a DB Server that can have large data area requirements, e.g. buffer cache, it should not be used on Terminal Server. In general, performance is likely to be improved by appropriate kernel memory tuning, particularly on operating systems that predate Windows 2003

- **Network bandwidth** requirements will depend in the main on the frequency, size and complexity of screen drawing; an overall average of 20-30Kbits/second per concurrent user is not unusual although peaks of 100-200Kbits/second are possible, e.g. when using a web browser to look at a news site with pictures. It will obviously help if the application is tuned to be relatively parsimonious in its screen handling. RDP compression features can help to reduce the requirement. If overall bandwidth requirements are onerous the use of Bandwidth Management techniques, e.g. traffic shaping, may be beneficial

- **LAN Server Bandwidth**. Following on from network bandwidth it is important to follow the requirements through to the design of the server LAN. The standard solution is to have a single LAN connection from each Terminal Server, supporting both traffic to/from the client and traffic to/from the middle and backend tiers of the application. In many systems the application traffic is likely to be significantly greater than the network traffic. In larger systems it may be necessary to separate the two types of traffic, placing them on different LANs

- **CPU** usage will obviously vary by application and level of usage. As a quick rule of thumb (although it obviously depends on the frequency of use and application complexity) allow between 0.5 and 2% of a 3GHz Xeon CPU per concurrent user. As discussed above, on servers with 2-3GHz CPUs memory is likely to be the bottleneck long before the CPU. As discussed in Chapter 5 (Hardware Sizing: The Crystal Ball Gazing Act), it is sensible to use utilisation thresholds when sizing to ensure satisfactory performance (a 60% CPU threshold is suggested in lieu of any custom benchmarks)

- **Misbehaving applications.** It is almost certain that client side application code, which has been developed for freestanding PCs, will misbehave in some shape or form, affecting either resilience and/or performance. Some form of multi-user testing is essential to discover what problems may be lurking

- **Size of roaming profiles.** For reasonable logon and logoff performance it is essential that the amount of copying that is involved in transferring these profiles from and to the "master profiles" is kept to an acceptable level. Some form of monitoring and observation may be required to establish a pragmatic but satisfactory level. An example of limiting the copying is use of the folder redirection feature allows items such as My Documents to be stored outside the profile on a central file server (home drives as shown in Figure 31.2), helping to keep the size of the profile within bounds. Of course, a possible downside of this approach is that any significant volume of disk IO traffic during user sessions may affect performance as IOs are remote rather than local with the potential for network and file server bottlenecks

- **Use of client-side disk**. The facility is available to access disk on the client-side from a Terminal Server session. Limited observations indicate that this feature performs poorly; file access protocols running under RDP do not sound very pleasant. While it may be acceptable to use it spasmodically, e.g. to occasionally save a copy of a word processing document or spreadsheet, it is probably unwise to use it frequently as part and parcel of a production application

- **Printing.** The standard mechanism for client-side printing is to render the print job on the terminal server and send it over the network for printing. The rendered version is invariably much larger than the original base data, thereby using more network bandwidth. As a very simple example this taster occupies ~450KB (including the diagrams and Word overheads) while the rendered version (for printing on a deskjet is ~1.2MB). The recent (at the time of writing) RDP 5.2 supports compression of print images, which will alleviate some of this pain. There are a number of third party printing products. They fall into two camps: the job is rendered on the server using a Universal Print Driver, stored in PDF / PCL format, and sent over the network for client-side printing (the PDF version of this taster is ~60KB); or a common metafile format is created on the server, sent it over the network, and then rendered and printed at the client site

  Scanning documents as part of an application is another potential large-scale user of bandwidth.

31.4 General Observations

Some common sense is necessary when deciding where terminal servers should be located in relation to the system and application services that they will access. In one particular example, users across the UK of one application accessed a single terminal server over WAN links with questionable bandwidth capacity. Siting the application and database servers in a data centre 200 miles away from the terminal server over the same network then compounded the network performance problem. Although it will obviously depend on the volume of traffic between terminal server and application server, in this case it would have been preferable to site the terminal server close to the application server and database server.

The published benchmarks that I have seen are fairly typical of vendor benchmarks; not surprisingly perhaps, they tend to overstate the number of users that can be supported on a given configuration. They tend to make significant use of office products, as they can frequently use modest amounts of resources: moderate levels of input; potentially minimal screen drawing; little disk IO, and relatively small memory requirements in comparison with real world applications. In addition, they are well-behaved items of software (e.g. if Microsoft cannot write well-behaved software for Terminal Servers, who can?). It is recommended that custom benchmarks be set up if there is any doubt surrounding likely performance.

Here are some observations on a benchmark that I was involved in (treat with care):

- The application consisted of three tiers (client, application and database)

- The client tier could be described as “fattish”; it performed a reasonable amount of local processing, had its own inbuilt client-side caching mechanisms and used compression techniques when communicating with the middle tier

- The middle tier could be described as “thinnish”, as some of the work was offloaded to the client and some to the database tier (heavy use of stored procedures)

- The users in the benchmark could be described as “medium”, typically submitting a transaction every 30-40 seconds

- The terminal server was a dual CPU (2.8GHz Xeons) with 4GB of memory

- 100 concurrent users used 46% of the CPU during the test proper (excluding logons and logoffs)

- With respect to memory each user required 80-90MB of memory. Memory performance was just about reasonable in spite of quite a lot of hard page faulting. However, it was definitely the constraining factor

- Network traffic averaged ~12Kbits/second per user (note that this application is reasonably lean and mean in its use of screen redrawing and the frequency of use is not particularly high)

- The above figures reflect metrics from the test proper, i.e. after logon and before logoff. Performance during logon and logoff was much more stressed: there was significant page write activity during the logon period (30 minutes), which in turn caused high disk IO traffic and additional CPU usage (the latter peaking at 90%+); and very high page read rates with associated disk IO traffic during the logoff period.

The above bullet points indicate that logon / logoff performance may be an issue on larger systems where these activities are performed in a relatively short period of time at the start or end of the working day, or more typically after a failure if users are forced to logon again.

In terms of hardware deployment for serious applications my personal preference is to use dual CPU systems with at least 2GHz Xeons and 4GB of memory; I remain to be convinced about the cost effectiveness of 4-CPU systems (a) on the grounds of general system scalability and (b) on the ability of memory to perform adequately to support the number of users that may come with such horsepower.

As discussed, any plans to make significant use of client-side printing needs careful investigation to minimise network bandwidth usage. The same observation applies to significant volumes of scanning.

31.5 Further Reading

Madden, B., *Terminal Services for Microsoft Windows Server 2003: Advanced Technical Design Guide*, BrianMadden.com Publishing, Washington DC 2004. This is an extremely useful, if lengthy guide.

Russinovich, M., *Inside Microsoft Terminal Server*, is an excellent, albeit dated, article to be found on the winntmag.com web site, which provides a detailed rationale of the required changes to the standard version of Windows NT to support Terminal Server.

Russinovich, M.E., *Windows Internals*, Microsoft Press International, ISBN 0735619174. If you are interested in the nitty-gritty of the Windows operating system this is the latest version of the "Inside Windows" series.

Information on TScale, the memory optimisation product, can be found at www.rtosoft.com.

Useful web sites include:

- www.brianmadden.com, Brian Madden's web site contains much related information on Windows-based thin client technology
- www.sysinternals.com, Windows operating system specific material written by Mark Russinovich and Bryce Cogswell
- there are various other web sites that focus on thin client technology, which may be worth browsing, viz. www.thinplanet.com, www.thin-world.com, www.thethin.net, and www.thinclient.net.

