Thin client

From Wikipedia, the free encyclopedia

A **thin client** (sometimes also called a **lean** or **slim client**) is a computer or a computer program which depends heavily on some other computer (its *server*) to fulfill its traditional computational roles. This stands in contrast to the traditional fat client, a computer designed to take on these roles by itself. The exact roles assumed by the server may vary, from providing data persistence (for example, for diskless nodes) to actual information processing on the client's behalf.

A public thin-client computer terminal inside a public library.

Thin clients occur as components of a broader computer infrastructure, where many clients share their computations with the same server. As such, thin client infrastructures can be viewed as the amortization of some computing service across several user-interfaces. This is desirable in contexts where individual fat clients have much more functionality or power than the infrastructure either requires or uses. This can be contrasted, for example, with grid computing.

The most common type of modern thin client is a low-end computer terminal which concentrates solely on providing a graphical user interface to the end-user. The remaining functionality, in particular the operating system, is provided by the server.

An Aleutia E3 thin client, with flash memory

Contents

History

Thin clients have their roots in multi-user systems, traditionally mainframes accessed by some sort of terminal computer. As computer graphics matured, these terminals transitioned from providing a command-line interface to a full graphical user interface, as is common on modern thin clients. The prototypical multiuser environment along these lines was Unix, and fully graphical X terminals were relatively popular thin clients in the 1990s. Modern Unix derivatives like BSD and GNU/Linux continue this multi-user tradition.

Windows NT became capable of multi-user operations primarily through the efforts of Citrix Systems,

which repackaged NT 3.5.1 as the multi-user operating system WinFrame. Microsoft licensed this technology back from Citrix and implemented it into Windows NT 4.0 Terminal Server Edition, under a project codenamed "Hydra." Windows NT then became the basis of Windows 2000 and Windows XP. Today, Windows allows graphical terminals via its Remote Desktop Services component.

IBM EXX thin client

The term *thin client* was coined in 1993 by Tim Negris, VP of Server Marketing at Oracle Corp., while working with company founder Larry Ellison on the launch of Oracle 7. At the time, Oracle wished to differentiate their server-oriented software from Microsoft's desktop-oriented products. Negris's buzzword was then popularized by its frequent use in Ellison's speeches and interviews about Oracle products.

The term stuck for several reasons. The earlier term "graphical terminal" was chosen to contrast such terminals with text-based terminals, and thus puts the emphasis on *graphics*. The term was also not well-established among IT professionals, most of whom had been working on fat-client systems. It also conveys better the fundamental hardware difference: thin clients can be designed with much more modest hardware, because they perform much more modest operations.


Size comparison - traditional Desktop PC vs Clientron U700

Thin clients as programs

The notion of a thin client extends directly to any client–server architecture: in which case, a thin client application is simply one which relies on its server to process most or all of its business logic. This idiom is relatively common for computer security reasons: a client obviously cannot be trusted with the logic that determines how trustworthy they are; an adversary would simply skip the logic and say "I'm as trustworthy as possible!"

However, in web development in particular, client applications are becoming fatter. This is due to the adoption of heavily client-side technologies like Ajax and Flash, which are themselves strongly driven by the highly interactive nature of Web 2.0 applications.

A renewed interest in virtual private servers, with many virtualization programs coming to a ripe stage, means that servers on the web today may handle many different client businesses. This can be thought of as having a thin-client "virtual server" which depends on the actual host in which it runs to do all of its computation for it. The end result, at least, is the same: amortization of the computing service across many clients.

Single point of failure

The server, in taking on the whole processing load of several clients, forms a single point of failure for those clients. This has both positive and negative aspects. On the one hand, the security threat model for the software becomes entirely confined to the servers: the clients simply don't run the software. Thus, only a small number of computers need to be rigorously secured, rather than securing every single client computer. On the other hand, any denial of service attack against the server will harm many clients: so, if one user crashes the system, everyone else loses their volatile data.

For small networks, this single-point of failure property might even be expanded: the server can be integrated with file servers and print servers particular to its clients. This simplifies the network and its maintenance, but might increase the risk against that server.

Although in practice redundancy is provided both in the form of additional connectivity from server to the

network as well as the servers themselves, using features like VMWare High Availability and Fault Tolerance or Citrix XenApp's load balancing.

Cheap client hardware

While the server must be robust enough to handle several client sessions at once, the clients can be assembled from much cheaper hardware than a fat client can. This reduces the power consumption of those clients, and makes the system marginally scalable: it is relatively cheap to connect additional client terminals. The thin clients usually have a very low total cost of ownership, but some of that is offset by requiring a robust server infrastructure with backups and so forth. This is also reflected in terms of power consumption: the thin clients are generally very low-power and might not even require cooling fans, but the servers are higher-power and require an air-conditioned server room.

On the other hand, while the total cost of ownership is low, the individual performance of the clients is also low. Thin clients, for example, are not suited to any real form of distributed computing. The costs of compiling software, rendering video, or any other computationally intensive task will be shared by all clients via the server.

Client simplicity

Gigabyte TA7 thin client

Since the clients are made from low-cost hardware with few moving parts, they can operate in more hostile environments than conventional computers. However, they inevitably need a network connection to their server, which must be isolated from such hostile environments. Since thin clients are cheap, they offer a low risk of theft in general, and are easy to replace when they *are* stolen or broken. Since they don't have any complicated boot images, the problem of boot image control is centralized to the central servers.

On the other hand, to achieve this simplicity, thin clients are generally highly integrated systems. This means that they may lag behind thick clients in terms of extensibility and accessibility. For example, if the server does not have support for independent audio streams, or the communication protocols don't transfer such streams, one simply cannot receive audio from the server. Similarly, if the client lacks USB ports, or if there is some communication failure of its USB signals over the network, the client might be wholly unable to support an unexpected USB peripheral.

Recent trends

Ultra-thin clients

A Sun Microsystems stateless S270 thin client,

Traditionally, a thin client ran a full operating system for the purposes of connecting to other computers. A newer trend is sometimes called an *ultra-thin client* or a *zero client*, which no longer runs a full operating system: the kernel instead merely initializes the network, begins the networking protocol, and handles display of the server's output.

Web thin clients

Web thin clients (running a Web OS) rely on the web-based software for the application and data storage, thus eliminating the single point of failure and the need for OS/application/data aggregation and licensing required by traditional thin client.

sometimes called an *ultra thin client*

Moderro Xpack Web-centric Thin Client

List of protocols used with thin clients

- Appliance Link Protocol
- Citrix ICA
- Ericom Blaze
- Remote Desktop Protocol
- Secure Shell or SSH, an encrypted replacement for telnet.
- Virtual Network Computing
- X11, central to Unix windowing
- XML, HTML, or JSON over HTTP (Ajax)
- DisplayLink over USB
- NFS

See also

- Blade PC
- Centralized computing
- Desktop Virtualization
- Dumb terminal
- Fat client
- Hybrid client
- Linux Terminal Server Project
- Multiseat configuration
- Nettop
- Smart client
- Terminal services
- Thinstation
- Time-sharing
- OpenThinClient
- Virtual Network Computing
- X11, central to Unix windowing

References