

Although it is not exactly a new concept, Cloud computing architectures are rapidly spreading over the world of IT, supporting the idea of provisioning various computing capabilities “as-a-service”, in a transparent way for users. Information is stored in servers on a network and cached temporarily on clients, such as desktops, entertainment centers, table computers, notebooks, wall computers, handhelds, etc. (see: "ORGs for Scalable, Robust, Privacy-Friendly Client Cloud Computing"). Reliable services are delivered to clients from next-generation data centers based on virtualization technologies. Some of the most relevant issues brought about by this paradigm are whether or not this is really feasible on a geographical scale, where network latency matters, and, more generally, whether or not a browser can really substitute every kind of computer application. Finally, big privacy issues rise: users data and work are given away in the hands of third parties, without any control and any real guarantee.

Without necessarily dealing with these “cloudy” aspects, it is always possible to reason about the more general and solid concept of Utility computing, according to which computing resources, such as computation and storage, can be precisely metered and packaged, similarly to what happens with a traditional public utility, apart from the fact that the distribution of such a service happens to be in “The Cloud”.

Utility computing can occur at different levels. As long as applications are concerned, one talks about Software-as-a-Service (SaaS): applications are hosted as a service provided to users across a network (e.g., the Internet). If systems are concerned, one can talk about Desktop-as-a-Service (DaaS): desktops can be transformed into a cost-effective, scalable and comfortable subscription service.

As it is described [here](#) , it is possible to identify three types of virtualized client computing (VCC):

1. Application: Encapsulating and isolating a specific application from its underlying host operating system and running it in the client environment, isolated from other applications running locally. Examples: Citrix Presentation Server (version 4.5+), Altiris Software Virtualization Suite, Thinstall, Microsoft SoftGrid, Trigence AE, Endeavors;
2. Desktop: Using virtualization to decouple the client environment (including operating system, application and data) from its host hardware and isolating it from other software or systems running on the client. It can be server-hosted or client-hosted. Server-hosted examples: VMware VDI, Virtual Iron VDI, Citrix XenDesktop, Qumranet Solid ICE. Client-hosted examples: VMware ACE/Player/Workstation/Fusion, SWsoft Parallels, Kidaro Managed Workspace, Sentillion;
3. Virtual user session: Creating multiple user sessions on the server, within a single operating system, that can be accessed concurrently. Examples: Citrix Presentation Server, Microsoft Terminal Services, Sun Secure Global Desktop Software.

Within these three types of VCC, two delivery models can be adopted (again in [6]):

- Remote interaction: I/O operations between a client device and a server through specific (and sometimes proprietary) protocols and software;
- Remote streaming: delivering executable blocks of data from a server to a client device, through specific (and sometimes proprietary) protocols and/or software.

VDD is in between type 2 and 3, as desktop virtualization software is used to host multiple unique and isolated client environments aboard a single server (or a group of servers in a cluster). Interaction with these remote virtual desktops is performed through virtual user (graphical) sessions. VDD exploits network transparency of X-Window-System: the machine where an application program (the client application) runs can differ from the user's local machine (the display server). X-Window-System clients run on virtual servers and create multiple user sessions within multiple virtual environments. X- Window-System display servers run on thin clients (terminals). VNC protocols can be used for OSs which lack of X11 server (e.g. Windows and ReactOS), so both delivery models listed above are available.