International Journal of Computing Sciences Research (ISSN print: 2546-0552; ISSN online: 2546-115X)
Vol. 6, pp. 857-876
doi: 10.25147/ijcsr.2017.001.1.78
https://stepacademic.net

Concept Paper

A Fat Client OS Architecture Supported by Semi-network Resources

Yin Sheng Zhang
Electrical and Computer Engineering, University of British Columbia, Canada
walterz862@hotmail.com

Date received: October 22, 2021
Date received in revised form: November 29, 2021; December 6, 2021
Date accepted: December 12, 2021

Recommended citation:

Zhang, Y. S. (2022). A fat client OS architecture supported by semi-network resources. *International Journal of Computing Sciences Research*, 6, 857-876. doi: 10.25147/ijcsr.2017.001.1.78

Abstract

Purpose – This study is to explore a way to retain the strengths and eliminate the weaknesses of the existing architecture of local OS and cloud OS, then create an innovative one, which is referred to as semi-network OS architecture.

Method – The elements of semi-network OS architecture includes network resources, local resources, and semi-mobile hardware resources; among them, network resources are the expanded portion of OS, which is used to ensure the scalability of OS; local resources are the base portion of OS, which is used to ensure the stability of local computing, as well as the autonomy of user operations; the semi-mobile hardware resource is OSPU, which is used to ensure the positioning and security of data flow.

Results – The fat client OS relies on the network shared resources, local exclusive resources, and semi-mobile hardware resources (OSPU), not relies solely on a single resource, to perform its tasks on a fat client, in this architecture, most of the system files of OS on a fat client is derived from OS server, which is a network shared resources, and the rest of system files of OS is derived from OSPU of a fat client, which is a non-network resource, so the architecture of OS has "semi-network" attribute, wherein the OSPU is a key subordinate component for data processing and security verification, the OS server is a storage place rather than operating a place of system files, and system files that stored on a server can only be downloaded to a fat client to carry out their mission.

Conclusion – A complete OS is divided into base portion and expanded portion, and this "portion" division of OS enables a fat client to be dually supported by remote network resources and local non-network resources, therefore, it is expected to make a fat client more flexible, safer and more reliable, and more convenient to be operated.

Keywords – base portion of OS, expanded portion of OS, fat client OS, thin client OS, Operating System Processing Unit (OSPU), semi-network OS

INTRODUCTION

This is a study on the sustainable development of fat clients and the utilization of semi-network resources; the concept of semi-network involved here refers to: if it belongs to "full network" that all data computing tasks are performed on the server-side, such as cloud computing, then it will be called “semi-network” that data computing tasks performed on fat client rely on only part of network resources; or if it belongs to “full network” that computing tasks of a fat client is performed completely dependent on network and strictly controlled by the network, such as thin client, then it will be called “semi-network” that computing task performed independently on fat client rely on part of network resources, and in this case, the fat client accepts network resource supplementation, but it can get rid of the control of the network to a great extent.

The mission

The missions undertaken by this study are: transform traditional OS architecture but make full use of the advantages of traditional one; try to use the method of architecture update instead of continuously adding new application software to enhance the reliability of system operation, or eliminate the inherent weaknesses of traditional architecture.

Today’s local OS is the most significant achievement of local system resources, and today’s cloud OS is the most significant achievement of network system resources, and their technical essence has been enough to become a solid foundation for the creation of semi-network OS architecture, specifically:

A. Under semi-network OS architecture, the shared network resources and the exclusive local resources are integrated into one, thereby improving the capacity and flexibility of data computing of fat clients, such as the space of fat client permanently occupied by junk software and other useless data in the traditional architecture will be automatically free up after device is shut down, while the network resources will act as a complement to local resources to ensure the flexibility of space occupation.

B. Under semi-network OS architecture, the data processing systems are integrated into one to ensure that the data at both ends of the network can communicate directly and smoothly.

There are different data processing systems in fat client and network server under traditional architecture, and the fat client has to be installed special data format conversion software to ensure the consistency of critical data in data transmission, but the semi-network OS architecture no longer has this requirement at all.

C. Under semi-network OS architecture, the security verification processes at both ends of the network are integrated into one or integrated under a unified system mechanism; thereby it is expected to greatly resist the mutual security threats between fat client and server.

As traditional architecture, the fat client is always in virus threat from the network without once-and-for-all response measures; on the other hand, the network server is also at the threat of viruses from the fat client all the time, in this case, virus attacks network server based on local platform, or spreads itself among different fat clients through the network.

The main points

The difference between a fat client and a thin client is: fat client is a network device installed with a wealth of local resources, such as hard drives, CD/DVD drives, software application, etc., while a thin client (a network-based application) disperses resources into a network and lacks its resources. The fat client OS discussed here relies on both network resources and local resources, not relies solely on a single resource, to perform its tasks, so it is referred to as semi-network OS.

The architecture of semi-network OS is designed for today's network environment, but its OS is not the so-called cloud OS known by people, because it will never perform computing tasks on a server but only act as an OS on a fat client; besides, this OS is not a traditional fat client OS, because it contains network resources, and its system files need to be re-processed by a hardware component of OSPU (operating system processing unit), then be able to run on a fat client; as a comparison, traditional fat client OS does not contain network resources, and as a local-only architecture, its system files do not need to be re-processed by any hardware component like OSPU in a fat client.

The architecture of semi-network OS contains three core elements, they are: an OS (system program), an OS server, and an OSPU, wherein, the OS server refers to a network server that specifically designed and provides services for a semi-network OS architecture, and the OSPU is a hardware component that directly dominates the semi-network OS, which can be installed on the motherboard of a fat client or is to be made as a plug-in component to connect to a device directly or indirectly.

Semi-network OS is not a cloud OS.

Cloud OS is never downloaded and running on a local platform, and it never directly manage specific hardware of fat client, but semi-network OS can only run on the fat client and perform computing task on a fat client, and once semi-network OS is downloaded to fat client, it will be running like a traditional OS on a fat client, and users do not have to change their long-established operating habits.

Semi-network OS is not a traditional OS of a fat client.

Traditional OS of a fat client has always been mixed installed on fat client's hard drive with a variety of software applications since it was born, regardless of those software applications are beneficial or harmful, however, as a comparison, semi-network OS is temporarily stored and running in RAM of a fat client, then hard drive and other external devices of a fat client will lose their original value in software installation, and will no longer to be the free place for malware and other uninvited programs to stir up trouble (Zhang, 2019b).

The Problems to be Solved

The study here focuses on solving the three weaknesses of traditional OS architectures: first, the fat client acts as a breeding place and footing base of viruses, and that the network platform can transmit viruses to a fat client and encounter little or slight obstacles; second, the excessively stacked application software and user data crowd up fat client resources, as well as the open model of software installation and data storage provides opportunities for virus flooding; third, the traditional model of an embedded OS that mainly used in mobile devices lacks the flexibility to scale function, and cannot adapt to or even hinder the evolution and transformation of mobile devices to some extent (Zhang, 2019c).

The Significant Changes

There will be some significant changes in the implementation of semi-network operating system architecture (Zhang, 2019a), such as:

A. Semi-network OS architecture can only be applied to a fat client, and never be applied to a thin client, and it is particularly suitable for ubiquitous computing and related intelligent devices, the reason is: the fat client here is already a network terminal and cannot be separated from network, the difference between it and thin client is only: thin client does not have a full set of hardware components to perform local computing.

B. In semi-network OS architecture, the “expanded portion” of OS is a resource generated on the server-side, which is an integral part of the overall design of the OS server rather than the data that generated on the fat client and be uploaded to a server for storage like traditional cloud storage; the expanded portion of OS can only be combined with the base portion of OS to form an over-all semi-network OS and running in RAM of a fat client, which means that the expanded portion of OS will not perform its function independently unless it is combined with the basic portion, and it only temporarily stays in a fat client in any case.

C. In semi-network OS architecture, the “base portion” of OS is embedded in OSPU chip, but it is not exclusively affiliated with OSPU, it is not dedicated to serving OSPU, and it is not tied to a specific device, its task is not only to manage OSPU itself, but also to manage the software and hardware of overall device of fat client; moreover, the base portion of OS will permanently stay in the fat client without uploading to a server, and never leaves OSPU.

D. In semi-network OS architecture, the “base portion” of OS is always the first step in process of transmission to RAM of the fat client (from OSPU); the “expanded portion” of OS is always the second step in process of transmission to RAM of the fat client (from a server by OSPU request and via OSPU channel); the integration of "portions" of OS and formation of complete semi-network OS (in RAM of fat client) is always the third step in the whole process; above sequence of "portions" transmission and integration are not allowed to be changed, nor is it allowed to be reversed in programming.

E. OSPU is a “semi-mobile” hardware component in a critical position in semi-network OS architecture; a tiny OSPU is carried by user, just like the entire set of semi-network OS is carried by user, or a complete set of a device with personalized settings is carried by user; therefore, those fat clients that are suitable for using semi-network OS architecture will become neutral devices serving different OSs, and OSPU gains “semi-mobile” attribute.

The Prominent Results

One of the most prominent results brought by semi-network OS architecture will be the change in the way of communication between fat client and server:

- The change from “indirect” to “direct”, that is, the fat client does not need to be installed an extra network connector or software to communicate indirectly with the OS server, and the OSPU can provide a channel for the fat client to directly access services on OS server as network-connected.
- The change from "open multi-channel" to "exclusive single-channel", that is, OSPU will be the only channel for fat clients to communicate with the OS server.
- The change from "unstable and rigid" to "stable and flexible", that is, the base portion of OS is used to ensure "stability", and the expanded portion of OS is used to ensure “flexibility” in communication.

- The change from “with trace” to “no trace”; semi-network OS architecture is based on local computing, so its operation traces are all in fat client, and "no trace" means that the fat client no longer keeps the traces of operations permanently, that is because the expanded portion of OS will automatically “disappears” after a device is shut down; base portion of OS and other data will automatically “disappears” from a device after OSPU is removed, in this cases, no trace of a previous operation (including traces of communication operation) will be left in a specific device.

THE BASE PORTION OF OS

The Feature of a Base Portion of OS

The concept of a base portion of semi-network OS contains meanings of "base" and "portion", where the "base" is defined as the system files is used to support most basic functions of a fat client, and " portion" is defined as the system files are only part of over-all OS stored in OSPU, base and portion are combined to form the technical features of "base portion" of semi-network OS, which includes the following attributes:

- The base portion of OS is pre-sealed and permanently integrated on the OSPU chipset, it is not allowed to be installed, uninstalled, and modified by the user and another third party.
- Base portion of OS combines expanded portion of OS to form an overall OS to provide fat client full functionality.
- When a network is disconnected, or in the absence of an expanded portion of OS, the base portion of OS will independently support the basic operation of a fat client, which include starting up OSPU and other components of a fat client, supporting the attempt to connect network; it is bundled with basic software applications and supporting basic software applications to operate.
- The base portion of OS can be transferred and applied to different kinds of fat clients together with OSPU.
- The base portion of OS combines the startup function and operation function of a fat client into one, to streamline OS in a fat client.
- The base portion of OS can independently perform a full set of basic operational tasks, which distinguishes it from a thin client because a thin client does not have its hard drive, CD/DVD drive, software application, and so on.

The Novelty of the Base Portion of OS

Traditional fat client OS is in a one-to-one installation mode when an OS is installed on a specified fat client, it cannot be transferred freely to another fat client to operate, so even if mobile devices has been quite popular, users are still not willing to give up their desktop and traditional notebooks.

To facilitate the free installation of OS, the traditional fat client has to allow users to install whatever software on the fat client according to their individual preference, which also leads to the fat client being kidnapped by swollen data, so "free installation platform" has been greatly reducing the work efficiency of a fat client.

Some techniques try to change this situation, one of the most representative alternatives is network thin client, but network thin client is not equipped with a permanent OS, and it will be downloaded from a server to enable it to operate; moreover, the thin client will be a waste if there is no network, or even in the case of network-connected and OS downloaded, the operation of a thin client is still under threat of interruption, that's because this operation is mainly for supporting network transmission rather than local processing of data, even network is temporarily disconnected, thin client will also be forced to stop work.

In contrast, the semi-network OS has a base portion, and this "base portion" can independently support the most basic operation of a fat client without a network, so it is more adaptable than a thin client. The base portion of OS not only acts as a supplement of the expanded portion of OS but also acts as the independent OS of a fat client in absence of an expanded portion of OS, thus forming the fat client's ability to run without a network. Moreover, the base portion of OS does not contain "non-basic" system files and un-related applications, so it will be able to reduce the consumption of fat client storage and other resources.

Therefore, the novelty of the base portion of OS lies in its " independent" attribute, which does not belong to a network resource, so it can independently support the smooth running of the fat client regardless of the fat client to be connected to the network or not.
The novelty of the base portion of OS also lies in its "mobile" attribute, which means the base portion of OS is not a resource dedicated to a specified fat client, in other words, it has no affiliation with a specific fat client, and only OSPU can determine its whereabouts; the base portion of OS is embedded on OSPU chipset and is strictly protected by OSPU security measures, in this regard, it somewhat similar to another type of embedded OS, but it is only a part of semi-network OS, once OS server connected, it will be automatically integrated with the expanded portion of OS, and it can be freely transferred from a device to another.

THE EXPANDED PORTION OF OS

The Feature of the Expanded Portion of OS

The concept of the expanded portion of OS contains meanings of "expanded" and "portion", where the "expanded" is defined as that the system files provided by the OS server will expand the operational capability of fat client and "portion" is defined as that the system files is only a part rather than "over-all", as well as it acts as a network shared resource to make up for the shortcomings of local resources; the " expanded " and " portion " are combined to form the following technical features :

- An expanded portion of OS will never run independently on a fat client, and it will never function until it combines with the base portion of OS to form an overall semi-network OS.
- An expanded portion of OS downloaded from the OS server will be adjusted in real-time according to the actual needs of the fat client.
- The expanded portion of OS supports data computing and user data generating on a fat client.
- The expanded portion of OS is not temporarily, nor permanently stored in OSPU or other external memory devices of a fat client, it is only temporarily stored in the RAM of a fat client, and it will automatically disappear after a fat client is shut down.
- OSPU is the guiding device for the expanded portion of OS to be downloaded, and the RAM of a fat client is the only platform for system files of the expanded portion of OS to be downloaded.
- An expanded portion of OS will never run on thin clients, however, as part of semi-network OS, it runs on the serving host of a thin client.

The Novelty of Expanded Portion of OS

The novelty of expanded portion of OS can be seen from the comparison with thin client OS, such as:

- The thin client OS cannot improve the performance of the device; a thin client is only used to upload user instructions from itself to a server, and download image data (generated on server) from server to itself, it cannot increase and decrease functions, or even improve its performance by itself, but can only accept the adjustment of server; moreover, when network traffic is in fluctuation or interruption, the working of all thin client will slower or even stop, and their OS can do nothing about it.
- The thin client OS cannot operate autonomously, it cannot directly generate user data, and its entire computing task is on a server, so the thin client OS performs a passive operation without any flexibility for scaling.

A specific thin client always needs to maintain the consistency of the system type and technical requirements with the specific server, so an OS of a specific thin client cannot adapt to all kinds of thin client, and more impossible to adapt to a fat client; in addition, it can only be downloaded via a single network channel, and it is a highly simplified one only being used within the scope of LAN, and what's more, a LAN's thin client OS may not be able to be applied to other LANs.

All in all, the “expanded portion” in the architecture of semi-network OS has attributes of "local computing", "multiple network channels", etc., it is expected to overcome the thin client's obstacles in its practice.

OSPU AND ITS APPLICATIONS

OSPU as a Supporting Element in the Configuration of Fat Client

The fat client with semi-network OS architecture requires four elements, they are a fat client (specific device), OSPU, semi-network OS, and OS server. Semi-network OS is divided into two portions, these two portions contain different OS files, and they are stored in different places, wherein the system file of the base portion is stored in OSPU, and the system file of an expanded portion is stored in the OS server. OSPU is a portable hardware component in fat clients, only the fat client that carried OSPU can carry semi-network OS.

Each OSPU has its network connector and its own unique hardware identification code, wherein the hardware identification code of OSPU contains two subdivided codes, which are a series of numbers and kernel code. In addition to the storage of the base portion of semi-network OS, OSPU also provides other functions for fat clients, such as network connecting, OS start-up, system files flow guiding (semi-network OS into RAM of fat client), data encrypting or decrypting (network transmission), security code inspecting (software application), safety monitoring (send-out data), and device driver storing.

OSPU as a Supporting Element in the Fat Client Start-up Process

A user sends a request for starting up semi-network OS to OS server, the request will be encrypted by OSPU, then the OSPU try to connect OS server; in this case, there are two attempt results will be: successfully connected and connecting failed.

Connected successfully

OSPU releases system files of a base portion of OS to RAM of the fat client first, then OSPU sends start-up request that encrypted by OSPU to OS server, meanwhile, OSPU encrypt its hardware identification code and sent this encrypted information to the OS server. OS server receives these two kinds of encrypted data from a fat client, after data is successfully decrypted by the OS server, the OS server will confirm the series number and kernel code of OSPU.

After the hardware identification code of OSPU is successfully confirmed, the OS server will encrypt the system file of the expanded portion of OS and release it to OSPU that is consistent with the previously received hardware identification code. OSPU receives the encrypted system file of the expanded portion of OS and decrypts them, then guides this system file to the RAM of a fat client. In the RAM of the fat client, the base portion of OS and the expanded portion of OS are integrated, and then the startup process of semi-network OS is completed.

Connecting failed

OSPU cancels the previous start-up requests of semi-network OS, and then OSPU releases the system file of a base portion of OS to RAM of a fat client, and then the start-up process of semi-network OS is completed.

OSPU as a Supporting Element in the Accounting Process

After semi-network OS is successfully started up, a user account will face the login request, in this case, the hardware identity information of OSPU plays an important role in security verification.

In the architecture of semi-network OS, a user account is divided into public account and private account:

Public Account

The public account acts as a default login account after semi-network OS is successfully started up, the public account will be automatically logged in; the account configuration file of public account belonging to system files of a base portion of semi-network OS, and public accounts can be logged in regardless of network connection or not, but it can only be successfully logged in after the specific OSPU is used and the hardware identity information of specific OSPU is verified by system security.

A public account used in semi-network OS architecture does not require user registration, and user name and password may not be required as user logs in his account, as long as a specific OSPU is accessed, this account will be automatically logged in, but in this case, user data generated in the fat client will not be automatically stored to OS server.

Private Account

The configuration files of private accounts in semi-network OS architecture belonging to the expanded portion of OS, a private account can only be logged in when the specified OSPU is used and connected to a network. When a private account is used in semi-network OS architecture, the user data generated on a fat client will automatically default to be stored in a specified network server, but users have the option to close the automatic storage function. User registration is required before a private account is first used by a user, after a private account is registered, semi-network OS will ask a user to provide his personal information and login information, and automatically collect OSPU hardware identification code for registration, then specified network server will bind user's personal information, login information and OSPU hardware identification code together, and store them on the server after binding.

As a special account, once the OSPU hardware identification code is successfully bundled with personal and other login information provided by a user, the information, which does not include login password, will not be modified in future use. If the OSPU hardware identification code collected in a specified server is requested to be altered by the user, the user will have the option to rebind new OSPU hardware identification code information for replacement, but the specified server will first successfully confirm the user's additional pre-agreed login information.

METHODOLOGIES

Architecture Building

The building of semi-network OS architecture includes the following general steps:

System Data Division

In the new architecture, OS running on fat client is divided into local resources and network resources, that is, it is divided into the base portion of OS and the expanded portion of OS, wherein the base portion of OS is enclosed in a hardware component OSPU that with a semi-mobile attribute, so the base portion of OS is not specific to a specific device like usual embedded OS; moreover, the base portion of OS is a part of a complete OS, and the usual embedded OS is a complete OS, the base portion of OS has the attributes to be supplemented by network resources and combined with network resources (expanded portion) to form a unified OS, but the usual embedded OS lacks this attributes.

The Development of Hardware Components of OSPU

An important feature of semi-network OS architecture is that it requires a dedicated hardware component to process system data, however, in traditional OS architecture, whether local OS or cloud OS, they have never required special and dedicated hardware components like OSPU to join the system data processing.

In addition, OSPU is also an important security component in semi-network OS architecture (Zhang, 2020), its security performance and system data processing performance form a mutually supporting "dual verification" effect, which can replace and strengthen the traditional security mechanism, such as replacing and strengthening the traditional anti-virus software installation mechanism.

Workflow and Configuration

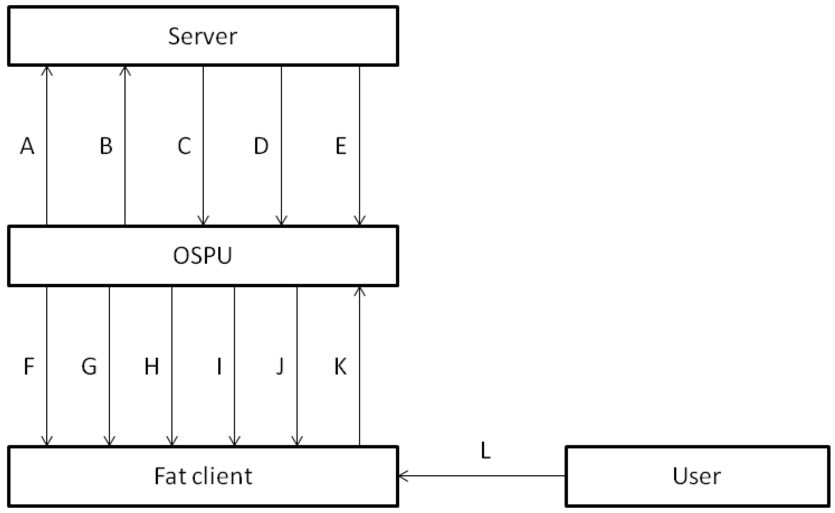

Figure 1. Workflow of semi-network OS architecture

In Figure 1, (A) is the process for fat client to connect to network, wherein the network connector is embedded in OSPU chip; (B) is the process for fat client to send instruction to OS server for downloading system file; (C)is the process for verifying OSPU hardware information by OS server; (D) is the data download process of expanded portion of OS;(E)is the application software downloading verification process (including hardware driver security verification), wherein the hardware driver is downloaded and stored in OSPU memory chip, and accepts server updates, wherein the application software runs in Random Access Memory (RAM)of fat client after download, but it will disappear together with expanded portion of OS after device of fat client shuts down; (F) is the process of OSPU to be verified and accepted by fat client's BIOS, wherein, the hardware driver stored in OSPU also requires OSPU to verify and accept the information of hardware configuration of device of fat client;(G)is the process of starting hardware device of fat client, wherein the fat client obtains files of base portion of OS from OSPU, obtains its stored hardware driver from OSPU, and starts hardware device of fat client; (H)is the process of system files to be downloaded from serve via OSPU and temporarily installed in RAM of fat client; (I)is the process of application software being downloaded via OSPU and temporarily installed in RAM of fat client; (J) is the process of data of expanded portion of OS and base portion of OS are combined by OSPU to form a whole OS running on fat client, wherein the expanded portion of OS is flexible part of OS, its data download volume can be adjusted and supplemented by request of OSPU to server in a timely manner according to actual operating needs of fat client;(K) is the process of user data generated by fat client to be uploaded to server after security verification by OSPU; (L) is the process where the user sends instructions to fat client.

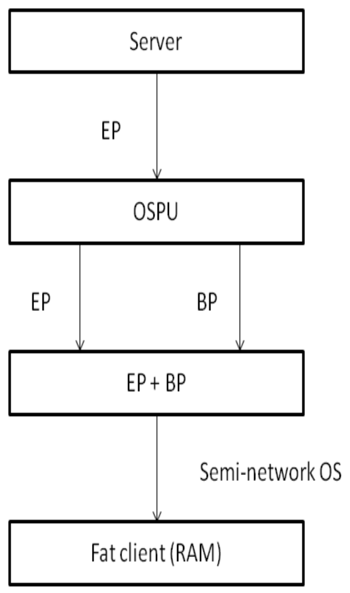

Figure 2. Configuration of semi-network OS architecture

In Figure 2, (EP) is the expanded portion of OS; (BP) is the base portion of OS; wherein, the semi-network OS, including (BP) and (EP), in the fat client will automatically disappear after a device is shut down. This is an important difference compared to thin client devices that whether it is (BP) or (EP), the downloaded data that runs in RAM is not permanently installed on a specific fat client. Because OSPU is not a permanent component dedicated to specific devices of fat clients, so the (BP) to be stored in OSPU does not mean it is to be a permanent OS in a device; moreover, the (BP) in OSPU is enclosed installed on-chip, its volume is small, and for OSPU chips, all foreign software is prevented from free installation.

The non-preset software, including systems and applications, are also prevented from being installed on it even if they always belong to the same software system; fat clients no longer need additional anti-virus and anti-hacking software; therefore, this architecture is expected to enable the OS to run easily in a simple environment.

DISCUSSIONS

There are a few points about the workflow of semi-network OS architecture that needs to be emphasized:

A. The expanded portion of semi-network OS architecture is a shared resource on the server-side and a temporary resource on a fat client, it has the attribute of scalability and flexibility, and this attribute is never found in traditional architecture.

B. There is a hardware component OSPU that is dedicated to processing OS in semi-network OS architecture, and it can also use the "dual verification" effect to strengthen the security mechanisms, and the "dual verification" effect is never found in traditional architecture.
C. The base portion of semi-network OS is also permanently installed on a fat client like traditional architecture, but it can be detached from a specific fat client by the user at will, and it can follow OSPU to be transferred from one device to another, while OS's personalized settings remain unchanged, this feature is never found in traditional architecture.
D. OSPU is a plug-in chip device, which usually does not permanently belong to a specific fat client (computing device), so it has semi-mobile attributes.
E. The network connector is pre-installed on a chip of OSPU, and at present, a variety of ready-made chips with network connectors have existed in the market, which greatly reduces the difficulty of making the hardware of OSPU.
F. The OSPU must first be accepted by the fat client's BIOS before it can start the fat client and perform other tasks, but the OSPU is made with chips, and it is difficult to find the ready-made chips that can be used to make OSPU and accepted by BIOS so far, fortunately, there is no technical difficulty to make this kind of chips, and the IT community has begun to make attempts in this regard, so it is expected that ready-made chip will be seen in the market soon, which can greatly accelerate the pace of OSPU development.
G. Discussed here is a new system architecture, not just a new program of OS, and in fact, various traditional programs of OSs can be transformed into new ones according to the new rules, therefore, most of the programs of the original OS only need to be modified with a limited workload without much more rewriting to form semi-network OS architecture, and the original OS's ecological environment, mainly the application software environment, can also be transplanted unimpeded to the architecture of semi-network OS environment; for example, Linux is an open-source system and its ecological environment is sufficiently diverse and complete, starting with such a transformation can greatly accelerate the formation and acceptance of semi-network OS architecture (Boyd-Wickizer et al., 2010; Grandl et al., 2016).

After continuous improvement year after year, the traditional fat client OS has formed a complete set of sophisticated and practical architecture and operational procedures, but it also has its fatal deficiencies, for example, as a local-only OS, it is completely separate from the network and has to use a specialized software application to contact network server, moreover, its "open and free" installation mode provides convenience for malware rooted fat client.

The architecture of semi-network OS is a solution for eliminating the deficiencies of the traditional architecture of fat clients, but it also absorbs the advantages of traditional architecture, such as local computing and cloud computing; in this solution, most of the system files of OS on a fat client will be derived from OS server, which is a network shared resources, and the rest of system files of OS is derived from a hardware component (OSPU) of a fat client, which is a non-network resource, so the overall OS has the "semi-network" attribute; wherein, the OS server is a storage place of system files rather than operating place in the architecture of semi-network OS, and system files can only be downloaded to a fat client to carry out their mission, which is the essential dividing line with cloud computing (Armbrust et al., 2010; Popa et al, 2012, Kim & Jung, 2013).

SIMILAR TECHNOLOGIES

Thin Client OS

The only similar technology suitable for discussion here is the thin client OS. The thin client OS is a tiny OS that downloads from a specific network server, and with its support, a thin client can log in to a remote OS on a server; wherein, the thin client refers to a computing dumb terminal that requires no dedicated software application in device for networking, and it communicates with the server through some protocols and then accesses the LAN; the thin client does not need a hard disk, and all data processing is done by the server, so it has a clear boundary with a fat client. Thin client OS is a technical outcome of combined use of network and client device, which enables resources of OS to be shared within a certain range through network server (Yang, Nieh, Selsku & Tiwari, 2002; Nieh, Yan, & Novik, 2000) but its shared OS resources are completely different from semi-network OS, the most obvious difference is:

- the base portion of semi-network OS is stored in OSPU on a fat client, not from the server, but thin client OS completely comes from the server, and
- the expanded portion of semi-network OS is stored in OS server, but it can only be functioning in fat client after it is combined with a base portion of semi-network OS, so it is not an independent OS, and never runs on any kind of fat client alone.

Weaknesses of Thin Client OS

Thin client OS is entirely a network source and its system data is not allowed to be downloaded to non-designated thin clients by the user's independent and free choice, nor can the thin client independently adjust the downloaded system file:

A. If thin client OS is not downloaded from a server to a thin client, the thin client will not be able to perform the tasks by a user; moreover, the thin client OS downloaded from the server is only to support data transmission and display.
B. Thin client OS can only be downloaded to a thin client that is compatible with the same OS, it cannot be downloaded to a thin client that is compatible with other types of OS.
C. The system files of thin client OS can only be completely downloaded to the thin client once, and its downloading cannot be divided and adjusted according to the operating needs of a thin client.

The thin client does not directly generate user data, and its data-generating are all on a server (host device), and the main function of thin client OS is to support the operation of transmission of user commands and download image data （results of server computing）, and display image data on a thin client; since the thin client OS cannot support the generation of user data, its application scope cannot jump out of the thin client. WAN server can also act as a resource provider of thin client OS in theory (Deboosere et al., 2007; Honda, Okagawa, Uchiyama, & Kurumatnai, 2012; Lai & Nieh, 2006), but the thin client is based on server computing, so it is unrealistic for WAN server to take on huge computing task from countless thin clients.

However, semi-network OS supports local computing on the fat client rather than server computing, and which overcomes the difficulties and shortcomings of thin client OS, and the following points can also enable the semi-network OS to make up for the shortcomings of thin client OS:

- When a network is disconnected, or there is no network, semi-network OS can support the basic operation of a fat client.
- When a network is connected but network traffic or speed is poor, the semi-network OS will reduce the impact of network instability.
- Semi-network OS applies to any type and model of the fat client regardless of what OS the fat client is originally carried, and the system files of semi-network OS downloaded from the server are adjustable according to actual operation needs of a fat client, or maybe downloaded zero.

PROSPECT ANALYSES

On the one hand, the technology using pure local resources and network resources has reached a fairly mature level, but on the other hand, a system built solely on local resources or network resources can never get rid of its inherent shortcomings, therefore, with the continuous emergence of modern new technologies, it is imperative to improve the traditional architecture.

Inevitability

The integration of network resources and local resources has become an inevitable trend of computing technology, and the innovation of OS architecture based on semi-network principles is just a matter of time. The semi-network OS architecture is specifically designed and built for the modern network environment, so it has great potential to avoid many weaknesses of traditional architecture, after all, in the era when the first generation OS architecture was created, the modern network technologies had not yet appeared, therefore, from a modern perspective, it is inevitable that the traditional architecture has some inherent weaknesses; now that things have changed, how to deal with these inherent weaknesses has always been the focus of attention in the field of computing technology (Wentzlaff & Agarwal, 2009; Engler et al., 1995;

Massalin & Pu, 1992), and the most important topic is how to adapt the traditional architecture to the modern network environment.

In traditional architecture, the fat client relies on more and more application software to implement network communication and ensure smooth data exchange on different platforms, which not only causes a large amount of software to be accumulated on fat clients but also may become a major security risk in network process due to the extremely high variability and human manipulation of application software.

Feasibility

Modern programming technology, network technology, and chip technology are currently recognized as the fastest-growing technologies with the most innovative results, and their current level and maturity are sufficient to allow the semi-network OS architectures to be successfully developed (Kivity et al., 2014; Colmenares et al., 2011); for example, the development of expanded portion and a base portion of OS depends on programming technology, or more simply, which can also be implemented by transforming some traditional open-source OSs, thereby achieving more results with less effort; the development of OSPU requires chip technology, and there are a variety of ready-made chips on the market for research and development of OSPU now, it can be expected that the ready-made chips used by OSPU will become more applicable and mature in near future; as for the network technology required by semi-network OS architecture, this is also developing rapidly. Besides, the semi-network OS architecture has the base portion of OS, which not only supports the basic local operation of a fat client but also provides conditions for the deployment of network technology.

Motivation

The wave of the ubiquitous computing comprising the Internet of Things and artificial intelligence has provided a strong impetus for the research of semi-network OS architecture, that is because the Internet of Things and artificial intelligence both rely on computing devices to achieve their goals, and the various devices they need also have higher and higher requirements for the operating environment, such as, "purification", "security", "lightweight computing" and "flexibility", and as the soul of computing devices, the traditional OS architecture has always been under pressure to keep up with the times; as a result, the fat client has to be installed more application software, sometimes even just for using one kind of software to deal with another kind of software, to enhance the adaptability of the fat client, which leads to an endless "software accumulation" race on fat clients.

Some of the inherent deficiencies of traditional architecture violate the requirements of ubiquitous computing, which forms the motivation for the transformation of system architecture of fat client, and also forms the most direct reason for the emergence of the concept of semi-network OS architecture.

CONCLUSIONS

Semi-network resources are resources formed by the integration of network resources and local resources for local computing, and the communication of different resources at both ends of the network is performed through OSPU or similar hardware components, which skips traditional dedicated application software such as browsers.
What is discussed here is one of the specific innovative forms of semi-network resources: semi-network OS architecture.

The fat client under traditional system architecture not only suffers from threats of a virus, but also suffers from system file swelling and junk file over-stacking, which frequently make the device run slower, or even frequently make the device run failure.
Now, existing technologies have begun to try to change the isolated state of some traditional OSs that are permanently installed on a local device, and get resources from network (Greenberg et al., 2011), such as a thin client. The thin client OS is completely dependent on a specified network server, network factors become the first requirement of the thin client to start up and operate, but in the existing technical level of networking, such a requirement is still unable to be fully and effectively guaranteed.

In addition, because a so-call thin client is the display device of server, in essence, thin client OS only supports the function that user instructions send from thin client to server, and image data downloaded from server to thin client, and the image data thereof is the user data generated by the server. The server's CPU cannot indefinitely undertake the computing task from countless local devices, therefore in foreseeable future, the thin client can only be limited to use within the local area network.

However, the semi-network OS can be transferred to different devices for operation along with the OSPU, so it is not restricted by a specific device; it has a "base portion of OS", so it uses a network but does not completely rely on a network; it obtains resources from the server, but it is a "local computing" rather than a "server computing" as a thin client to be, so it will not increase the server's "computing" burden, nor will it be restricted by server; it has an "expanded portion of OS", so it can make the fat client's operating capabilities show great flexibility, also minimizes the resource consumption of fat client.

In this new architecture, an overall OS is divided into “base portion” and “expanded portion”, this "portion" division of OS enables a fat client to be dually supported by both the remote network resource and the local non-network resource, which will inevitably provide many cutting-edge technologies with more choices for supporting devices; for example, the Internet of Things and artificial intelligence have higher and stricter technical requirements for the security, flexibility, and convenience of fat clients, and once the new OS concept is implemented, fat clients with a semi-network operating system architecture are bound to be one of the best choices for those technologies.

Lastly, when compared to the existing fat client: under traditional architecture, the semi-network OS architecture has huge potential to make fat clients get rid of threats of the common scourge of fat clients such as illegal software application, over-stacking of junk files, etc. Also, when compared to the thin client: the semi-network OS architecture can also make local devices share network resources like thin clients, but the difference is that it has huge potential to break through the control of specified network server, breaks through the scope of LAN, and breaks through the one-to-one affiliation between local device and server.

REFERENCES

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50-58. Retrieved from https://dl.acm.org/doi/pdf/10.1145/1721654.1721672

Boyd-Wickizer, S., Clements, A. T., Mao, Y., Pesterev, A., Kaashoek, M. F., Morris, R. T., & Zeldovich, N. (2010, October). An Analysis of Linux Scalability to Many Cores. In *OSDI* (Vol. 10, No. 13, pp. 86-93). Retrieved from https://pdos.csail.mit.edu/papers/linux:osdi10.pdf

Colmenares, J. A., Bird, S., Eads, G., Hofmeyr, S., Kim, A., Poddar, R., ... & Kubiatowicz, J. (2011, August). Tessellation operating system: Building a real-time, responsive, high-throughput client OS for many-core architectures. In *2011 IEEE Hot Chips 23 Symposium (HCS)* (pp. 1-1). IEEE Computer Society.

Deboosere, L., De Wachter, J., Simoens, P., De Turck, F., Dhoedt, B., & Demeester, P. (2007, June). Thin client computing solutions in low-and high-motion scenarios. In *International Conference on Networking and Services (ICNS'07)* (pp. 38-38). IEEE. Retrieved from https://ieeexplore.ieee.org/document/4438287

Engler, D. R., Kaashoek, M. F., & O'Toole Jr, J. (1995). Exokernel: An operating system architecture for application-level resource management. *ACM SIGOPS Operating Systems Review*, *29*(5), 251-266. doi:10.1145/224057.224076

Grandl, R., Chowdhury, M., Akella, A., & Ananthanarayanan, G. (2016). Altruistic scheduling in multi-resource clusters. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)* (pp. 65-80). Retrieved from https://www.usenix.org/system/files/conference/osdi16/osdi16-grandl-altruistic.pdf

Greenberg, A., Hamilton, J. R., Jain, N., Kandula, S., Kim, C., Lahiri, P., ... & Sengupta, S. (2011). VL2: a scalable and flexible data center network. *Communications of the ACM*, *54*(3), 95-104. doi:10.1145/1897852.1897877

Honda, Y., Okagawa, T., Uchiyama, K., Kurumatnai, S., Toyama, M., Chen, E. Y., ... & Takada, J. (2012, March). Thin client technology for mobile service platform. In *2012 World Telecommunications Congress* (pp. 1-6). IEEE. Retrieved from https://ieeexplore.ieee.org/document/6170467

Kim, N. U., Jung, S. M., & Chung, T. M. (2013). A remote control architecture for thin-client in mobile cloud computing. In *2013 International Conference on Information Science and Applications* (ICISA) (pp. 1-2). IEEE. Retrieved from https://ieeexplore.ieee.org/document/6579335

Kivity, A., Laor, D., Costa, G., Enberg, P., Har'El, N., Marti, D., & Zolotarov, V. (2014). OSv—optimizing the operating system for virtual machines. In *2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14)* (pp. 61-72). Retrieved from https://www.usenix.org/system/files/conference/atc14/atc14-paper-kivity.pdf

Lai, A. M., & Nieh, J. (2006). On the performance of wide-area thin-client computing. *ACM Transactions on Computer Systems (TOCS)*, *24*(2), 175-209. doi: 10.1145/1132026.1132029

Massalin, H., & Pu, C. (1992). A lock-free multiprocessor OS kernel. *ACM SIGOPS Operating Systems Review*, 26(2), 108. doi: 10.1145/142111.993246

Nieh, J., Yang, S. J., & Novik, N. (2000). A comparison of thin-client computing architectures. Retrieved from https://www.researchgate.net/publication/2453810_A_Comparison_of_Thin-Client_Computing_Architectures

Popa, L., Kumar, G., Chowdhury, M., Krishnamurthy, A., Ratnasamy, S., & Stoica, I. (2012, August). FairCloud: Sharing the network in cloud computing. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication* (pp. 187-198). doi:10.1145/2377677.2377717

Wentzlaff, D., & Agarwal, A. (2009). Factored operating systems (fos) the case for a scalable operating system for multicores. *ACM SIGOPS Operating Systems Review*, *43*(2), 76-85. doi: 10.1145/1531793.1531805

Yang, S. J., Nieh, J., Selsky, M., & Tiwari, N. (2002, June). The Performance of Remote Display Mechanisms for Thin-Client Computing. In *USENIX Annual Technical Conference, General Track* (pp. 131-146). Retrieved from https://pdfs.semanticscholar.org/e60a/892427b03bfbcbbcfb058ef310708006276c.pdf

Zhang, Y. S. (2019a). Applicability analysis of semi-network operating system. *Computer and Information Science, 12*(1), 93-97. doi: 10.5539/cis. v12n1p93.

Zhang, Y. S. (2019b). Semi-network OS and new mode of application software sharing. *Journal of Multidisciplinary Engineering Science and Technology (JMEST),* 6(5), 10083-10088. Retrieved from http://www.jmest.org/wp-content/uploads/JMESTN42352949.pdf

Zhang, Y. S. (2019c). Semi-network OS and Embedded OS for Co-mobile Computing. *International Journal of Computing Sciences Research,3*(2), 189-198. doi: 10.25147/ijcsr.2017.001.1.32.

Zhang, Y. S. (2020). Data Management System for Ubiquitous Multi-Task Mobile Devices on Semi-Network OS Architecture. *EJECE: European Journal of Electrical Engineering and Computer Science*, *4*(4), 1-8. doi: 10.24018/ejece.2020.4.4.236