A Just-in-Time Mechanism for Computer System

Qijian Chen\(^1\), Yuanchang Zhong\(^2\), Taohua Liang\(^1\)

\(^1\)Department of Software Engineering, Chengdu Polytechnic, Chengdu, Sichuan 610041, China.

\(^2\)School of Microelectronics and Communication Engineering, Chongqing University, Chongqing, 400044, China.

* Corresponding author’s e-mail: zyc@cqu.edu.cn

Abstract: As the installed programs will consume increasing space of hard disk, the computer becomes more and more bulky and costly. However, it’s harder and harder to make a larger yet still light and cheap hard disk. Therefore, computers used in most occasions are suggested to cancel their hard disks, and have their operating system programs and the application programs ‘installed’ on internet, instead of being installed on local hard disk. The data is accessed via supper high speed internet when the program booting. The operating system additionally serves as an open platform for registering the application software as well as the traditional role. This mechanism will bring us many gains. First, the user will no longer be restricted by the limited storage space, and will literarily have a never discarded and unlimited storage space that is shared by his or her all intelligent terminals. Second, the computers can become lighter and therefore easier to carry. Third, the user can get the latest version program once he or she launches the program. Last, the program provider can take the initiative to push the newest message to its users easily if they have subscribed.

1. Introduction

In this article, we talk about problems caused the present computer architecture based on hard disk and how to reform the present architecture to solve these problems. Unless specified, the computer in this article refers to desktop computer, laptop, mobile phone and any other mobile terminals.

With the development of computer technology, computer has successively passed through a series of stages with different characters. The following introduces the development and limitations of traditional computers in different stages.

1.1. Stage 1, OSless mode: No hard disk, no operating system (OSless)

The structure of a typical modern computer was first described in 1946 in a landmark paper by Barks, Goldstein, and Von Neumann \(^1\)-\(^2\). Since then, a classical computer mainly comprises a CPU, primary memories (ROM and RAM) and I/O peripherals.

However, there was no hard disk in the early-stage computers. Therefore, those computers had no standard operating system (OS for short) and can only run one application program each time \(^3\)-\(^4\). That is a terrible defect because of their limited capability.

1.2. Stage 2.0, Local-Stored-Application-on-OS mode: there are standalone applications based on OS and hard disk.
The early 1970s saw the first standard hard disk RK05 being used on the computer PDP-11, on which more than one application can be install based on OS\[4\]. Up to now, hard disk is one of the widely-used auxiliary memories. In fact, there are many kinds of auxiliary memories, including floppy disk, magnetic tape, optical disk and hard disk, to be used as warehouse of data. However, only hard disk survives as a kind of indispensable memory of computer, for the reasons that its capacity can be very large and it is easy to erase.

As we well know, both the operating system software and application software are installed on the hard disk of a traditional computer\[5-8\]. Once installed, the program can be accessed locally and launched an unlimited number of times. The operating system plays as an interface between application software and hardware\[9-12\]. It can make the users to do different things using different application programs. This kind of software almost doesn’t communicate with the internet. Actually, it is not an exception for mobile terminals, such as smart phones, which has the similar structure as the common computers. And the new technologies\[7,9,12,13\] haven’t changed this basic mechanism. So, the hard disk becomes one of the most indispensable parts of a common personal computer.

However, the hard disk also becomes an increasing burden and limitation for modern computer. The reason is that, with the users’ ever-growing needs for more capable computers, the operating system and the application software are designed larger and larger. As a result, the hard disk is filled with more and more programs and other data. In the last it inevitably becomes the bottleneck of computer. How about to equip the computer with a larger and larger hard disk? That sounds not a good idea for mobile terminals. Because the hard disks are also proved to be a considerable part of cost. The more capacity the hard disk has, the bulkier and more expensive it is. Furthermore, hard disk with a larger capacitor yet with still a tiny volume is quite hard to produce. That is to say, producing that kind of hard disk is either too expensive, or beyond our present ability. So, the capacity of hard disk literally became the ceiling of the number of programs that the computer can install.

![Fig. 1. Traditional computers with increasing Bulky HDD](image1)

![Fig. 2. The schematic layers of the traditional architecture](image2)

### 1.3. Stage 2.1, the browser mode based on low-speed internet

With the internet widely used in daily life, the user can visit the websites of the service provider by using “internet browser”. This makes computer to be quite concise because the users don’t need to install any other program of the service provider on their computer.

Unfortunately, this is a kind of passive communication mode between the service provider and the user, because the service provider cannot take the initiative to push their information to the clients before the users visit. What’s more, this kind of mechanism cannot use local hardware devices. As a result, the providers are required to provide the related hardware devices for these surfing visitors, which is extra burden for most service providers.
1.4. Stage 2.2, a mode of simple-App mainly depending on local data, with only a little data from internet.

This is an evolved mode of stage 2.0, Local-Store-Application-on-OS mode mentioned above. The difference is that programs of this stage depend greatly on interacting with the internet. With the development of internet, some applications of personal computer begin to access data from internet. Those applications, abbreviated to Apps, are especially popular on mobile terminals[13]. As a result, it will need a relatively big space to store those Apps, comparing to the Browser Mode.

There are two advantages of this mode. The first one is that the service provider can post the latest information to the users. The second advantage is that the Apps provide particular functions that can use local hardware devices. This is a great and important superiority to the Browser Mode.

As the users need to install a particular application from a particular service provider, the storage space of this kind of mode will inevitably be very crowd. Consequently, the computer or mobile terminal would be slower and slower.

1.5. Stage 2.3 Apps on App mode: service based on Open Platform Apps installed on local hard disks

With the emergence of new generations communication technology, a kind of more complex Apps developed from the simple App mentioned in section 1.5, which a few Apps become open commercial platforms for the other Apps. The typical commercial platforms are the Alipay, Wechat and the Amazon App. We named it 'TApps-on-App mode'. Any other service provider (the third party) can apply a “official account” or ID on those open platforms. Then the users can search and “add care” that official account, with which they can use the third-party programs on the open platform. These open platform Apps play role like internet browser, and are more capable than the latter. Because those third parties service can send information on their own initiative to their users. What’s more, as the App can use local devices, the providers needn’t to provide the related hardware devices for the users anymore. These are great superiorities to the internet browser. A typical example is that people can order food or shopping on a mobile payment platform or a social application. The data of the third party is not stored on local hard disk, but on internet. So, when the program of the third party is booted, its data is accessed temporarily from internet. The architecture of this kind of model is shown in Fig. 3.

The third advantage this kind of mechanism is that users don’t need to install too many programs on their computers or mobiles. As a result, the user doesn’t need much storage space of auxiliary memory, with just a high-speed internet and some high-performance devices.

On the other hand, the disadvantage of this “commercial platform” is that this kind of architecture is too complex. Since the platform App is based on the system software, and the “third party” application is based on the platform application, so there are too many layers in this structure, so it is quite complex and will consume too much storage of RAM.

2. A Just-in-Time mechanism

The well-known just-in-time (JIT for short) system is a zero-inventory management strategy that the purchaser organizes the supporter deliver the required quantity of raw-material at the right time they need. This strategy can reduce costs, as it can increase efficiency and decrease waste by receiving goods only as they need them for the production process. The suggested JIT mechanism for computer works like the JIT inventory management strategy. Briefly, all the programs, including the OS, driver and application programs, are suggested to be 'installed' on the internet, instead of on local hard disk.
So it is also named non-locally-installation mechanism. The proposed JIT mechanism makes the OS an open platform for temporary applications (TApps for short), which is named as ‘TApps-on-OS’ mode. The data of its program is downloaded from a registered website in time when a program is launched. Details are described below:

2.1. Details of design

2.1.1. Design of hardware
First of all, the computers of proposed mechanism get rid of local hard disk, which is replaced by internet disk. So, the computer is mainly made up of CPU, RAM, ROM and I/O peripherals. A super powerful wireless network card, like the high-performance CPUs and RAMs, is also the most important foundation of this Just-in-Time mechanism.

2.1.2. The enhanced BIOS
A reformed firmware has the information to visit routers. And it also has information to download the OS program. First of all, BIOS is enhanced to reserve and manage the information to routers, such as the name and password of WIFI, and the information of software, i.e., its website address and account numbers. So, the computer can access data from a given website address, for example that of OS. In addition, information of the OS, include the website address, account number (or ID) and password, are also managed by the enhanced BIOS. With them, the firmware can access data of OS program from internet, then launch it in an instant.

2.1.3. The temporary-application-program on OS mode (TApps-on-OS)
Besides the traditional functions, the OS of the proposed mechanism is also designed to be an open platform to manage applications programs. First, the service delivers apply to the OS manager for official accounts (or ID) of their applications program. Then the OS can help the users to search the official account and reserve the link to the application program. The application program is also stored on the internet, instead of being stored on local hard disk.

When user clicks the link to startup a program with an authorized code, the OS downloads the program data from the server of the service provider, and launches the program promptly. These user’s documents are stored on the internet, too. When the program is closed, the RAM can be cleared up at once.

The user’s documents are stored on internet disk, which can be shared by different intelligent terminals of the same owner. So, when a computer is claimed to be useless, the internet disk would not be discarded together.
To sum up, the OS plays as both an interface and a special open platform to application software, which is an open platform of OS level. It is used to replace the commercial platform of the App level, such as the Alipay, or the Amazon App, etc. That is to say, the APP management platform sinks down to the OS level.

2.2. Feasibility analysis

There are two necessary conditions are needed for this mechanism: plenty of communication base stations or relays and the fast enough transmission rate.

2.2.1. Distribution of Communication infrastructure

The first precondition is that users can surf the internet anytime anywhere they live. As some countries or corporations are developing the technology of IP-over-satellite\(^{[14-15]}\), which will make it the possible to communicate almost everywhere.

2.2.2. Transmission rate

A further precondition of this mechanism is that the transmission rate should be high enough. A basic criterion is users’ psychological endurance of the total time to download and launch the program. That is:

\[
t_{download} + t_{launch} \leq t_{tolerate}\tag{1}
\]

\(t_{tolerate}\): total time that most users can tolerate.

Suppose that the total time that most users can tolerate is \(t_{tolerate}\) second to start a program, and the downstream time should be no more than \(t_{tolerate}/2\).

Suppose the data volume is \(D_{total}\) Byte and there are N step to download and launch the program. So, the total data is divided into N parts,

\[
D_{total} = Dp_1 + Dp_2 + \ldots + Dp_N.\tag{2}
\]

And the maximum part of data is:

\[
D_m = \max(Dp_1, Dp_2, \ldots, Dp_N) = \frac{1}{M} D_{total} (M \geq 1).\tag{3}
\]

Then the speed of downstream is no less than:

\[
speed \geq \frac{D_m}{t_{tolerate}/2} \quad (\because D_k \leq D_m, t_k \leq t_{tolerate}/2)
\]

\[
= \frac{2D_{total}}{M t_{tolerate}} (M \geq 1) \quad \text{(Byte)} \tag{4}
\]

If the initial value of \(t_{tolerate}\) is 0.1 second, and the total volume of data is 1GByte, and initial part of data is about 1/2 of the total data of the program, then the speed to download data. Then:

\[
\text{speed} \geq \frac{2D_{total}}{M t_{tolerate}} (M \geq 1) = \frac{2 \times 1Gb}{1 \times 0.1s} = 20Gb/s
\]

According to ITU guidelines, 5G internet speeds should have a peak data rate of 20 Gb/s for the downlink and 20 Gb/s for the uplink. So, the future internet \(^{[16-17]}\) can afford a more promising speed for the proposed mechanism. It is feasible to put this mechanism to practice.
2.3. The performance of the proposed mechanism

2.3.1. The advantage of the proposed mechanism

Compared to the traditional computer, the proposed computer has the following advantages:

1) The computer of the proposed architecture is lighter and tinier than the traditional ones. This is good news for users of mobile terminals.
2) As the auxiliary memory of the proposed mechanism is internet disk, so the owner of the computer will literally have a permanent and unlimited storage space. Even if he or she purchases a new computer, he or she can share the same internet disk.
3) The proposed TApps-on-OS mode has fewer layers than the TApps-on-App mode, which makes the computer more efficient.
4) Meanwhile, people don't have to spend too much energy and cost to design a tiny hard disk with large storage capacity. The cost cut down can be used to enhance other parts of computer, such as CPU, RAM, high-speed network board.
5) The subscription platform of OS can deliver support to authorize the service deliver to push news to the users, while the explorer can't do so.
6) Then the users can access the latest updated program once they boot the program.

2.3.2. The disadvantage of the proposed mechanism

The suggested mechanism depends entirely on a widely covered high-speed network. Computer of this kind of architecture will out of work without the internet. The present network can't cover everywhere of the global. So, people who want to use computers of the proposed architecture have to stay in the areas with high performance communication facilities.

Nevertheless, the restriction will be broken before long with the development of new communication technology [14, 15, 17, 19].

3. Summary

As proved in the paper, with the help of the coming new generation communication, a computer can be reformed to have its OS and application programs installed on the internet. First, this just-in-time mechanism, or non-locally-installation mechanism, cancels the increasing bulky hard disk, then a firmware can be enhanced to guide the computer to launch the OS from a designated website. The OS is reformed to be an open platform for the applications, with which the data of the application programs to be boot is downloaded from internet.

There are many gains. Since the hard disk is cancelled, the weight of computer can be cut down to an appreciable degree. Computers will be relatively thinner, more cost saving, and easier to carry. The saved cost of local hard disk can be spent on getting higher performance CPUs and RAMs. What's more, the service delivers can push the latest information to their users if the users have subscribed. And at the same time, the user can access the latest updated version of program once he or she boots it. At last, the owner of the computer will no longer be restricted by the limited storage space, and will literally have a never discarded and unlimited internet disk that can be shared by his or her all intelligent terminals.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities (No. 2020CDCGTX055, No. 2019CDCGTX302). The work described in this paper was performed at the Innovation and Practice Base for Post doctors, Chengdu Polytechnic. The authors would like to thank our respected colleague Shiqing Yang for his guidance and support.

Reference:

[1] Von Neumann, J. (1945). First draft of a report on the EDVAC. doi:10.5479/sil.538961.39088011475779
[2] Arthur W. Burks, The invention of the universal electronic computer—how the Electronic Computer Revolution began, Future Generation Computer Systems, Volume 18, Issue 7, 2002, Pages 871-892.

[3] M. Stojčev. Computer Architecture: A quantitative approach. Microelectronics Journal, 1997, 28(5); 599-600.

[4] Andrew S. Tanenbaum, Albert S. Woodhull: Operating Systems: Design and Implementation, 3rd ed. London: Pearson Education, Inc., 2006

[5] Milojicic, G. Dejan S. Milojicic. Operating Systems: Now And In The Future. IEEE Concurrency, 1999, 7(1); 12-21.

[6] T.Z. Kalambokis, Communication performance of a message passing MIMD computer, Future Generation Computer Systems, Volume 10, Issue 4, 1994, Pages 403-409

[7] Yousef Bin Zikria, Heejung Yu, Muhammad Khalil Afzal, Mubashir Husain Rehmani, Oliver Hahn, Internet of Things (IoT): Operating System, Applications and Protocols Design, and Validation Techniques, Future Generation Computer Systems, Volume 88, 2018, Pages 699-706

[8] Andrzej M Goscinski, Towards an operating system managing parallelism of computing on clusters, Future Generation Computer Systems, Volume 17, Issue 3, 2000, Pages 293-314

[9] Quentin Rouland, Brahim Hamid, Jason Jaskolka, Formal specification and verification of reusable communication models for distributed systems architecture, Future Generation Computer Systems, Volume 108, 2020, Pages 178-197

[10] Chao-ze Lu, Guo-sun Zeng, Ying-jie Xie, Bigraph specification of software architecture and evolution analysis in mobile computing environment, Future Generation Computer Systems, Volume 108, 2020

[11] Steven J. Johnston, Philip J. Basford, Colin S. Perkins, Herryerry, Fung Po Tso, Dimitrios Pezaros, obert D. Mullins, Eiko Yoneki, Simon J. Cox, Jeremy Singer, Commodity single board computer clusters and their applications, Future Generation Computer Systems, Volume 89, 2018, Pages 201-212

[12] Andrey Getmanisky, Semen Sechenev, Igor Ryadchikov, Alexander Gusev, Nikita Mikhalkov, Dmitry Kazakov, Alexey Simulin, Dmitry Sokolov, Real-time system architecture design practices, Procedia Computer Science, Volume 186, 2021, Pages 769-776

[13] Abdel-moniem El-Shorbagy, 5G Technology and the Future of Architecture, Procedia Computer Science, Volume 182, 2021, Pages 121-131

[14] Daniel M. Kohn, Providing global broadband Internet access using low-earth-orbit satellites, Computer Networks and ISDN Systems, Volume 29, Issue 15, 1997, Pages 1763-1768

[15] Satoshi Utsumi, Salahuddin Muhammad Salim Zabir, Norio Shiratori, TCP-Cherry: A new approach for TCP congestion control over satellite IP networks, Computer Communications, Volume 31, Issue 10, 2008, Pages 2541-2561

[16] R. Ferrús, H. Koumaras, O. Sallent, G. Agapiou, T. Rasheed, M.-A. Kourtis, C. Boustie, P. Géland, T. Ahmed, SDN/NFV-enabled satellite communications networks: Opportunities, scenarios and challenges, Physical Communication, Volume 18, Part 2, 2016, Pages 95-112

[17] Karan Sheth, Keyur Patel, Het Shah, Sudeep Tanwar, Rajesh Gupta, Neeraj Kumar, A taxonomy of AI techniques for 6G communication networks, Computer Communications, Volume 161, 2020, Pages 279-303

[18] Satellite could make Internet reach further faster, Computer Law & Security Review, Volume 13, Issue 3, 1997, Page 203. https://doi.org/10.1016/S0267-3649

[19] Akila Sairete, Mohamed El Amin Mousa, Zain Balfagih, Tayeb Brahim, Aziza I. Hussein, Miltiades Lytras, Anna Visvizi, Editorial: 5G and Beyond; Paving the way for 6G, Procedia Computer Science, Volume 182, 2021, Pages 1-4