

Online Video Playing on Smartphones: A Context-Aware Approach Based on Cloud Computing

Jinlei Jiang, Yongwei Wu, Xiaomeng Huang, Guangwen Yang, Weimin Zheng

Department of Computer Science and Technology, Tsinghua National Laboratory for Information Science and Technology
Tsinghua University, China

{jjlei, wuyw, hxm, ygw, zwm-dcs}@tsinghua.edu.cn

Abstract

As the device holding great promise to realize the vision of pervasive computing, smartphones with Internet access and video playing capability have become commercial off the shelf products today. As one of the most popular applications on smartphones, online video playing, however, still faces some problems. From the perspective of service providers, they have to provide video files of different formats and qualities to meet the diverse needs of various consumers. From the perspective of service consumers, if one or more codecs required to open a video file supplied by a certain service provider are not available on their smartphones, they just cannot watch that video or have to install new codecs -- both cases give users bad experience. To deal with such a situation, this paper puts forward a context-aware approach for online video playing based on cloud computing. Our experience demonstrates that the combination of cloud computing and smartphones creates a lot of new opportunities for more advanced services.

Keywords: Context-aware computing, Mobile service, Cloud computing, Pervasive computing.

1 Introduction

As the advance of electronics and telecommunication technologies, smartphones with Internet access and video playing capability have become commercial off the shelf products today. Besides products from the traditional cell phone manufactures such as Nokia, Motorola and BlackBerry, IT giants Apple and Google also released their own cell phone products (iPhone and Nexus One).

Pervasive computing [1], or ubiquitous computing [2], was first envisioned by Mark Weiser in his seminal article "The Computer for the 21st Century" [3], which depicts a computing world where users can finish various tasks even without noticing the existence of the devices they use to fulfill their tasks. It highlights the thorough integration of computing capacity with our daily activities. Since smartphones have penetrated into our society and can be easily extended with new functionalities, it is generally accepted that they hold great promise to realize the vision of pervasive computing [4].

Online video playing is an application getting more and more popular on smartphones as the bandwidth for Internet access gets broader and broader. However, online video playing nowadays faces some problems. From the perspective of service providers, they have to buy more resources to cope with peak demands. In addition, they have to provide video files of different formats and qualities to meet the diverse needs of various consumers. Both cases result in a waste of resources. From the perspective of service consumers, they need various codecs to play videos encoded in different formats. If one or more codecs required to open a video file supplied by a certain service provider are not available on their smartphones, they just cannot watch that video or have to install new codecs -- both cases give users bad experience.

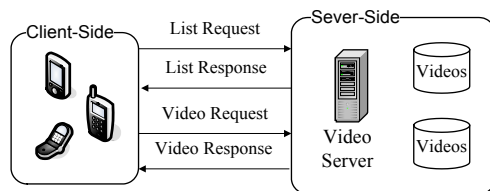
In our opinion, the problems stated above arise from two aspects: firstly, the inflexibility of the infrastructure adopted for online video playing; secondly, the information asymmetry between service providers and consumers, that is, service providers have no knowledge about the capability of the smartphones used by consumers and as a result, their services are designed to support the needs of common users and no consumer-specific services are supplied. This paper tries to solve these problems by giving an integrated solution. Contributions made are as follows.

- We devise a new online video playing service by combining the idea of context-aware computing [5] with cloud computing [6-7]. To the best of our knowledge, it is the first online video playing service that not only bases itself in clouds but also exhibits context-aware features.
- We demonstrate through our study that the interplay of cloud computing and smartphones can produce more advanced services.

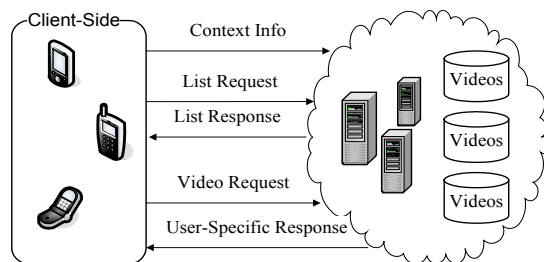
The rest of this paper is organized as follows. In the coming section, we give an overview of our new cloud-based context-aware online video playing system. Next, Section 3 explains the context model used in our system and details client-side software design, and Section 4 examines cloud-side services design and their interactions. Afterwards, we evaluate our system and discuss the related work on context-aware and cloud computing in Section 5. The paper ends in Section 6 with some conclusions as well as some future work.

2 System Overview

Our new cloud-based context-aware online video playing service, as shown in Figure 1, consists of two parts, where the architecture of traditional online video playing service is also given for comparison.



(a) Traditional Online Video Playing Service



(b) Cloud-Based Context-Aware Online Video Service

Figure 1 A Comparison of Traditional and Cloud-Based Context-Aware Online Video Playing Service

In traditional online video playing service, service providers deploy their own data centers to serve users and to store various video files. To guarantee service quality, they have to buy more resources to cope with peak demands. These resources become idle at normal time, thereby resulting in resource waste. It is in this sense that we say the infrastructure is inflexible. In addition, services are usually designed to support the needs of common users, with few characteristics of clients taken into account. As a result, users are obliged to utilize some codecs under certain conditions in order to watch a video.

Our new online video playing service improves existing solutions from two aspects.

Firstly, context information exchange between clients and service providers is introduced, as illustrated by the “Context Info” arrow line in Figure 1(b). The context, which will be explained in the next section, contains such information as the screen size, codecs installed, and bandwidth available. With context information got, service providers can then adapt their services to the special requirements of end users. The “User-Specific Response” in Figure 1(b) illustrates this point.

Secondly, all server-side services are moved to clouds. Thanks to the unique features of cloud computing [8]

-- unlimited resource available on demand, no up-front commitment and pay-as-you-go usage of resources, service providers need no longer to buy extra resources for peak demands, for they can rent more resources from the clouds whenever needed. In addition, based on the power of cloud computing, providers can develop more advanced services or functions, for example, recommending related films to a certain user, which is usually a compute-intensive task.

3 Context Model and Client Design

This section details the context model used in our system and client-side functional components design.

3.1 Context Model

Context and context-awareness have been hot topics in recent years especially in the fields of pervasive computing, computer-supported cooperative work (CSCW) and human-computer interaction (HCI) [9]. Due to space limitation, we will not explain existing work further. (Please refer to [10-11] for more information.) Instead, we will jump directly to the context model used by our system in the following.

According to Dey et al. [12], context is “any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves.” From this definition, it is easy to see that context involves nearly every aspect of an application and thus, it is not feasible to define a generic context model that covers all the information due to applications diversity. Bearing this in mind and taking into account the needs of online video playing as well as extensibility, we put forward a context model as follows.

Definition 1 (Context Model). A context model in this paper, denoted by CM , is defined as a tuple $CM = (Person, Tool, Environment, History)$ where

- *Person* describes a user profile in terms of preferences, personal features and other social relations (e.g., friends and communities one belongs to). It is an attribute reserved for future usage, e.g., notifying users that their friends get online and what they are watching.
- *Tool* is at the core of our context model. It records the software installed on the smartphone. For online video playing, we pay our attention to the codecs available and the Web browsers used.
- *Environment* keeps such information as screen size, screen resolution, CPU frequency, user location and the available bandwidth. These factors are critical for determining the quality (e.g., bit rate) of video to be played.

- *History* records the video-watching history of a certain user. It is also reserved for future usage, e.g., video recommendation.

In our model, whereas information about *Person* and *Tool* is relatively fixed and seldom changes, information about *environment*, user location and the available bandwidth in particular, is very dynamic. As for *History*, it can be got directly from the service side. This characteristic provides a good basis for incremental context information exchange between clients and servers.

To represent context information, we employ a hybrid way as explained below.

At the client side, we use extensible markup language (XML) to describe information about *Person*, *Tool* and *Environment*. Figure 2 shows a piece of context information where most attributes are omitted due to space limitation.

```

<Context>
  <Person>
    <ID>Cell Phone Number</ID>
    <Preferences> //one or more items
    <Item>Sports</Item>
    ...
  </Preferences>
  <Friends> //one or more friends represented by ID
  <ID>Friend's Cell Phone Number</ID>
  ...
</Friends>
</Person>
<Tools>
  <Tool>
    <Name>RealPlyer</Name>
    <Version>s60.31.42.01 A6</Version>
  </Tool>
  ...
</Tools>
<Environment>
  <OS>Symbian</OS>
  <CPUFrequency>389MHz</CPUFrequency>
  <ScreenSize>2.4Inch</ScreenSize>
  <ScreenResolution>240, 320</ScreenResolution>
  <Bandwidth>512Kbps</Bandwidth>
  ...
</Environment>
</Context>

```

Figure 2 A Piece of Context Information

At the server side, the information collected from clients is linked with an ontology. Such a design has two purposes: firstly, to depict complex context data, especially the relationships between various context entities; secondly, to facilitate context data sharing and reasoning. Ontology is selected because it formally represents a set of concepts within a domain and the relationships between those concepts [13] and there are quite some reasoning tools available. In our system, we exploit OWL-DL to

describe our context model because it has become a de facto standard in many application domains and a lot of reasoning services provide support for it. Figure 3 gives an overview of the proposed context ontology.

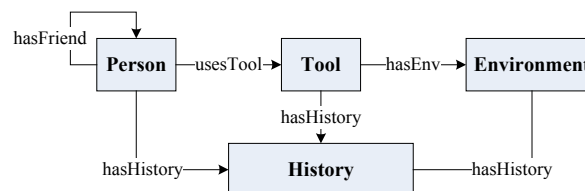


Figure 3 An Overview of Our Context Ontology

3.2 Client Design

To context-aware online video playing, the client mainly fulfills two functions: (1) collecting context information as formally defined by our context model; (2) navigating through the video list and playing the video selected. Details of them are as follows.

For context information collecting, different policies are adopted. Specifically, information about *Person* is got by asking users to fill in some forms when they access the online video playing service. To get other information about *Tool* and *Environment*, we developed some smartphone-specific helpers programs that can automatically retrieve the desired context using the APIs (Application Programming Interfaces) supported by the smartphone operating systems (OSs). To protect privacy, users are asked for the permission of context information collection.

For video list navigation and playing, since our purpose is to utilize existing video players rather than develop new ones, we just developed an interface for listing videos. Besides the traditional list-in-all view, this interface also provides list-by-month view to facilitate video navigation.

4 Cloud-Side Services Design

We base our server-side services design on Tsinghua Cloud. Details are as follows.

4.1 Tsinghua Cloud

Tsinghua Cloud is a test environment for cloud computing constructed using our own developed components. Figure 4 illustrates its architecture. Three key components, namely Carrier, Corsair and Nova, are explained below.

Carrier is a distributed file system that aims to deliver a high performance, high available and easy-to-use storage service. It adopts a loosely-coupled architecture consisting of five key modules: supervisors, clients, metadata servers (cluster), data servers and fuse. The metadata servers provide a global namespace of traditional tree structure.

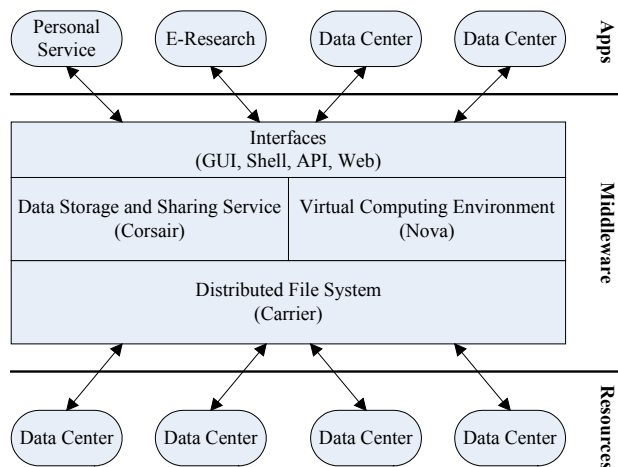


Figure 4 The Architecture of Tsinghua Cloud

The data servers, as the name implies, are used to store file data -- files in Carrier are divided into multiple 32MB chunks and then stored on several data servers. The clients provide operations to access file data and metadata. The fuse module enables users to use Carrier in the same way as using a local file system. The supervisors are designed to handle such issues as the failure of data and metadata servers, replication adjustment, load balancing, data integrity, garbage collection and so on. One point to highlight is that Carrier is fully POSIX-compliant.

Corsair is a facility for data storage and sharing. It aims to provide a uniform view of the files managed as well as other useful functions like bookmarking, searching and user management. The storage in Corsair is divided into three spaces, namely public space, community space and personal space. While files within a personal space are only visible to that person and files within a community space are only visible to those who belong to the community, files within the public space are visible to anyone. By default, a user is allocated a personal space of 2GB after registration and a community is allocated a space of 100GB after created. In the end, anyone can create a community of one's own and has full control over who can join the community. In this way, file sharing becomes very easy.

Nova is a toolkit aiming to provide facilities for users to construct a virtual computing environment using the given physical machines in an easier and more productive way. To do so, it provides such functionalities as virtual machine (VM) and virtual cluster creation, on-demand software deployment and service provisioning, and VM monitor and migration. Compared with other cloud infrastructures such as Amazon EC2, our Nova-enabled cloud has the following features:

- Install-/Configuration-free client, that is, users only need a Web browser to use the services supplied by Nova.
- High productivity, for it provides quite some pre-virtualized software and users can easily customize a

computing environment only by one or several mouse clicks.

- Inherent integration with storage cloud, that is to say, after VMs are created, personal space in storage cloud will be attached automatically as an independent driver, which then acts as a source of inputs and as the destination of the data produced.

4.2 Services Design

To context-aware online video playing, services at the server side are mainly in charge of video file storage and management, user management, user requests handling and so on. With Tsinghua Cloud, these services can be easily constructed. Figure 5 shows the deployed functional components of our newly designed service as well as their interactions, where normal cloud users are also shown.

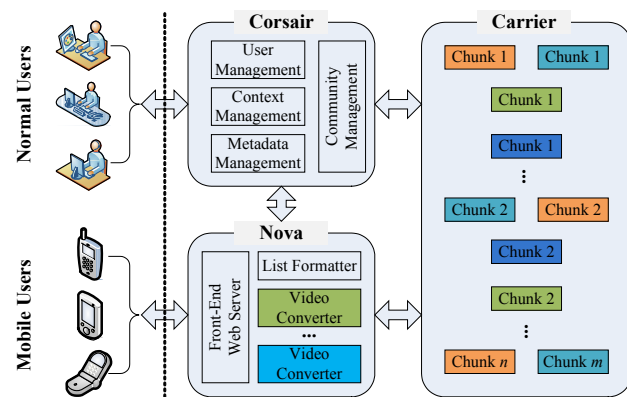


Figure 5 The Deployed Functional Components of Our Context-Aware Online Video Playing Service and Their Interactions

Carrier is deployed to store video files. Each video file is divided into one or more 32MB chunks and stored on multiple data servers. To guarantee data availability and reliability, for each file, at least three copies of it are maintained in Carrier by default. In addition, the location of files and the number of file replicas can be adjusted dynamically according to their usage. In this way, load balance is achieved and service throughput is guaranteed.

Corsair is deployed to manage users, context, metadata of videos, and community. Community management is an inherent function of Corsair. It is reserved here for future use. Metadata management maintains such information about a certain video as title, content description, snapshot and so on. It is unlike the metadata management in Carrier. Context management, as the name implies, is used to maintain the context information received from the clients. User management maintains user account, password, and other useful information related to a person.

Nova is deployed to run a front-end Web server that is in charge of handling users' requests, and other functions

where the List Formatter is used to adapt the video list (e.g., resize video icon and tailor the text displayed) according to the context information collected and the Video Converter is used to transform videos from one format to another. Due to the inherent scalability of cloud computing, new instances of Web server and Video Converter can be generated dynamically according to the real workload, thus ensuring the service performance and giving users better experience.

The full steps to use our cloud-based context-aware online video playing service are as follows.

- (1) A user visits the start page of online video playing service (interacting with the Front-End Web Server). Meanwhile, context information is collected and sent to the Front-End Web Server and then stored into Corsair.
- (2) The user issues a request to view the video list of a certain category.
- (3) The Web Server queries the Corsair service to get the full list, formats the list by the List Formatter according to the context got and sends the formatted list to the user.
- (4) The user navigates through the list and issues a request to play a certain video.
- (5) The Web Server checks if the video format is supported by the user's cell phone according to the context information collected. If not, a certain Video Converter is invoked to convert existing format to a desired one. Meanwhile, the context information is used to determine the bit rate, resolution and other parameters of the converted video. In the end, the video is sent to the client in a format that can be supported by the client.

5 Evaluation and Related Work

We tested our system both by simulation and real-world operation. In simulation, many processes are deployed to issue requests to our service. The result shows that the response time only changes slightly as the number of requests increases. This is because more resources can be leased on the fly from the cloud when the workload becomes heavy. In real-world operation, we use HTC Magic G2 to access a video in an unsupported format. The result shows that the video can still be played without installing new codecs. These tests demonstrated the viability and benefits of our design at least to some extent.

Context-aware computing is a hot topic drawing more and more attention. Below is some relevant work on smartphones.

ContextPhone [14] is a software platform for users to create context-aware applications on mobile phones using Symbian OS. The platform consists of four interconnected

modules that cover almost all aspects of context-aware computing. One amazing application in ContextPhone is ContextMedia, which enables users to capture, annotate and share media.

Contory [15] is a middleware specially designed for context provisioning on smartphones. It took into consideration the resources constraints of mobile devices and provided support for multiple context provisioning strategies as well as various context query approaches. Contory is also implemented on top of Symbian OS.

Using cloud computing to break down the barrier of resource constraints on mobile devices also draws much attention with some examples seen.

Byung-Gon & Petros [16] put forward a novel architecture named CloneCloud with which users can (partially) off-load the execution of an application from a smartphone to its clones in the cloud. Indeed, our work is inspired by theirs.

Larry [17] described a virtualization infrastructure for deploying pervasive computing applications. Based on virtualization technology, which is at the core of cloud computing, such an infrastructure provides an isolated operating environment for users to utilize I/O devices on remote machines.

Dejan & Ralf [18] discussed the opportunities created by cloud computing on context-aware mobile multimedia services and proposed to utilize cloud computing to support complex multimedia tasks on the mobile clients. Like in our system, they also suggested using ontologies to model context.

Compared with the work above, our system is an integrated solution of both context-aware and cloud computing features. We think it can enrich the related studies and give some hints to similar applications.

6 Conclusion and Future Work

This paper presented a new cloud-based context-aware online video playing service. Combining the features of context-aware and cloud computing, our service can benefit both service providers and end users. Our study also showed the great opportunity created by cloud computing to smartphones.

The work reported here is only our first step towards a more comprehensive media solution for smartphones. Our future work will focus on the following aspects. First, we will investigate more effective video recommendation algorithms based on the context information got and cloud computing. Second, we will provide support for sharing videos captured by users themselves using Tsinghua Cloud.

Acknowledgements

The work reported here is co-sponsored by National Basic Research Program of China (2007CB310900), Natural Science Foundation of China (60773145, 60911130371), National High-Tech R&D (863) Program of China (2009AA01A130, 2006AA01A101, 2006AA01A117), MOE-Intel Foundation, and Tsinghua National Laboratory for Information Science and Technology (TNLIST) Cross-discipline Foundation.

References

- [1] Mahadev Satyanarayanan, *Pervasive Computing: Vision and Challenges*, *IEEE Personal Communications*, Vol.8, No.4, 2001, pp.10-17.
- [2] Gregory D. Abowd and Elizabeth D. Mynatt, *Charting Past, Present, and Future Research in Ubiquitous Computing*, *ACM Transactions on Computer-Human Interaction*, Vol.7, No.1, 2000, pp.29-58.
- [3] Mark Weiser, *The Computer for the 21st Century*, *Scientific American*, Vol.265, No.3, 1991, pp.66-75.
- [4] George Roussos, Andy J. Marsh and Stavroula Maglavera, *Enabling Pervasive Computing with Smart Phones*, *IEEE Pervasive Computing*, Vol.4, No.2, 2005, pp.20-27.
- [5] Gregory D. Abowd, Maria Ebling, Guerny Hunt, Hui Lei and Hans-Werner Gellersen, *Context-Aware Computing*, *IEEE Pervasive Computing*, Vol.1, No.3, 2002, pp.22-23.
- [6] Rajkumar Buyya, Chee S. Yeo, Srikumar Venugopal, James Broberg and Ivona Brandic, *Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility*, *Future Generation Computer Systems*, Vol.25, No.6, 2009, pp.599-616.
- [7] Marios D. Dikaiakos, Dimitrios Katsaros, Pankaj Mehra, George Pallis and Athena Vakali, *Cloud Computing: Distributed Internet Computing for IT and Scientific Research*, *IEEE Internet Computing*, Vol.13, No.5, 2009, pp.10-13.
- [8] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica and Matei Zaharia, *Above the clouds: A Berkeley view of cloud computing*, February, 2009. University of California Report No. UCB/ECS-2009-28.
- [9] Guiling Wang, Jinlei Jiang and Meilin Shi, *Modeling Contexts in Collaborative Environment: A New Approach*, *Lecture Notes in Computer Science*, Vol.4402, 2007, pp.23-32.
- [10] Matthias Baldauf, Schahram Dustdar and Florian Rosenberg, *A Survey on Context-Aware Systems*, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol.2, No.4, 2007, pp.263-277.
- [11] Jong-Yi Hong, Eui-Ho Suh and Sung-Jin Kim, *Context-Aware Systems: A Literature Review and Classification*, *Expert Systems with Applications: An International Journal*, Vol.36, No.4, 2009, pp.8509-8522.
- [12] Anind K. Dey, Gregory D. Abowd and Daniel Salber, *A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications*, *Human-Computer Interaction*, Vol.16, No.2, 2001, pp.97-166.
- [13] Mike Uschold and Michael Grüninger, *Ontologies: Principles, Methods, and Applications*, *Knowledge Engineering Review*, Vol.11, No.2, 1996, pp.93-155.
- [14] Mika Raento, Antti Oulasvirta, Renaud Petit and Hannu Toivonen, *ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications*, *IEEE Pervasive Computing*, Vol.4, No.2, 2005, pp.51-59.
- [15] Oriana Riva, *Contory: A Middleware for the Provisioning of Context Information on Smart Phones*, *Proc. ACM/IFIP/USENIX Middleware*, Melbourne, November, 2006, pp.219-239.
- [16] Byung-Gon Chun and Petros Maniatis, *Augmented Smartphone Applications through Clone Cloud Execution*, *Proc. the 12th Workshop on Hot Topics in Operating Systems*, Monte Verità, Switzerland, May, 2009.
- [17] Larry Rudolph, *A Virtualization Infrastructure that Supports Pervasive Computing*, *IEEE Pervasive Computing*, Vol.8, No.4, 2009, pp.8-13.
- [18] Dejan Kovachev and Ralf Klamma, *Context-Aware Mobile Multimedia Services in the Cloud*, *Proc. the 10th International Workshop of the Multimedia Metadata Community on Semantic Multimedia Database Technologies*, Graz, Austria, December, 2009.

Biographies



Jinlei Jiang is an Assistant Professor of Computer Science and Technology, Tsinghua University, China. He received a PhD degree in computer science and technology from Tsinghua University in 2004 with an honor of excellent dissertation. His research interests mainly focus on grid and cloud computing, CSCW, and workflow management.



Yongwei Wu received a PhD degree in applied mathematics from the Chinese Academy of Sciences in 2002. He is presently a Professor of Computer science and Technology, Tsinghua University, Beijing, China. His research interests include grid and cloud computing, distributed processing, and parallel computing.



Xiaomeng Huang is an Assistant Professor with Department of Computer Science and Technology, Tsinghua University, Beijing, China. He received a PhD degree in computer Science and technology from Tsinghua University, China in 2007. He is especially interested in large-scale distributed system, computer networks, etc.



Guangwen Yang is a Professor of Computer Science and Technology, Tsinghua University, Beijing, China. He received a PhD degree in computer architecture from Harbin Institute of Technology in 1996. His research interests are mainly on parallel computing, grid computing, distributed systems and machine learning.



Weimin Zheng is a Professor of computer science and technology, Tsinghua University, China, and the managing director of the Chinese Computer Federation. He received the BS and MS degrees from Tsinghua University in 1970 and 1982 respectively. His research interests include computer architecture, operating system, storage networks, and distributed computing.