

ระบบเคลื่อนที่แบบคลาวด์สำหรับการเล่นแฟ้มสื่อผสม

นางสาวแก้วขวัญ แก้วมิ่งสา

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2555

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository(CUIR)

are the thesis authors' files submitted through the Graduate School.

A MOBILE CLOUD SYSTEM FOR MULTIMEDIA FILE PLAYING

Miss Kaewkhwan Kaewmahingsa

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science and Information Technology

Department of Mathematics and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2012

Copyright of Chulalongkorn University

Thesis Title A MOBILE CLOUD SYSTEM FOR MULTIMEDIA FILE
PLAYING
By Miss Kaewkhwan Kaewmahingsa
Field of Study Computer Science and Information Technology
Thesis Advisor Assistant Professor Pattarasinee Bhattarakosol, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of
the Requirements for the Master's Degree

..... Dean of the Faculty of Science
(Professor Supot Hannongbua, Dr. rer. nat.)

THESIS COMMITTEE

..... Chairman
(Assistant Professor Nagul Cooharojananone, Ph.D.)

..... Thesis Advisor
(Assistant Professor Pattarasinee Bhattarakosol, Ph.D.)

..... External Examiner
(Assistant Professor Ohm Sornil, Ph.D.)

แก้วขวัญ แก้วมิ่งสา : ระบบเคลื่อนที่แบบคลาวด์สำหรับการเล่นแฟ้มสื่อผสม. (A MOBILE CLOUD SYSTEM FOR MULTIMEDIA FILE PLAYING) อ.ที่ปรึกษา
วิทยานิพนธ์หลัก : ผศ.ดร.ภัทรสินี ภัทรโกศล, 71 หน้า.

การใช้แฟ้มข้อมูลสื่อผสมบนเครือข่ายอินเทอร์เน็ตนับได้ว่าเป็นการบริการที่ได้รับความนิยมอย่างแพร่หลายจากผู้ใช้งานคอมพิวเตอร์ส่วนบุคคล อย่างไรก็ตามการใช้งานอินเทอร์เน็ตได้ย้ายรูปแบบไปสู่การใช้งานบนอุปกรณ์เคลื่อนที่ต่างๆ โดยเฉพาะอย่างยิ่งสมาร์ทโฟน ได้มีการนำเสนอและจำหน่ายซอฟต์แวร์หลายประเภทผ่านทางช่องทางของผู้ให้บริการระบบปฏิบัติการต่างๆ อาทิ ไอโฟนของบริษัทแอปเปิลจำกัด, แพลย์สโตร์สำหรับระบบแอนดรอยด์ และมาร์เก็ตเพลสของไมโครซอฟท์ จึงเห็นได้ว่าการใช้งานแฟ้มข้อมูลสื่อผสมผ่านโทรศัพท์เคลื่อนที่กลายเป็นสิ่งที่ได้รับความนิยมและมีแนวโน้มที่จะกลายเป็นสิ่งปกติระหว่างผู้ใช้งานทั้งหลาย แต่เป็นที่น่าเสียดายที่ระบบการดาวน์โหลดมีปัญหาพื้นฐานคือ ความล่าช้าอย่างมากที่เกิดขึ้นระหว่างการโหลด จากความล่าช้าในระบบแม่ข่าย-ลูกข่ายเพื่อการบริการการเข้าถึงแฟ้มข้อมูลสื่อผสมแบบเดิม ดังนั้นระบบเคลื่อนที่แบบคลาวด์สำหรับการเล่นแฟ้มสื่อผสม ได้ถูกนำเสนอเพื่อแก้ปัญหาดังกล่าว ด้วยสถาปัตยกรรมการประมวลผลแบบคลาวด์ช่วยให้เกิดประสิทธิภาพในการใช้งานหน่วยประมวลผลและหน่วยความจำร่วมกันอย่างมีประสิทธิภาพสูงของอุปกรณ์คำนวณต่าง ๆ บนระบบเครือข่าย ด้วยเหตุนี้ระบบเคลื่อนที่แบบคลาวด์สำหรับการเล่นแฟ้มสื่อผสม ใช้งานทรัพยากรการประมวลผลและหน่วยความจำที่ว่างงานของกลุ่มสมาร์ทโฟนต่าง ๆ เพื่อนำมาใช้ระยะเวลาในการดาวน์โหลดแฟ้มข้อมูลสื่อผสมน้อยลง ด้วยการทำงานของระบบประมวลผลแบบคลาวด์จะทำให้ผู้ใช้สามารถรับชมแฟ้มข้อมูลสื่อผสมได้รวดเร็วยิ่งขึ้น

ภาควิชา วิศวกรรมศาสตร์และวิทยาการคอมพิวเตอร์.....ลายมือชื่อ.....

สาขาวิชา..... วิทยาการคอมพิวเตอร์และ..... ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก.....

.....เทคโนโลยีสารสนเทศ.....

ปีการศึกษา..... 2555.....

5472602423 : : MAJOR COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

KEYWORDS : MULTIMEDIA FILE PLAYING, CLOUD COMPUTING, MOBILE PHONE.

KAEWKHWAN KAEWMAHINGSA : A MOBILE CLOUD SYSTEM FOR
MULTIMEDIA FILE PLAYING. ADVISOR : ASST. PROF. PATTARASINEE
BHATTARAKOSOL, Ph.D., 71 pp.

The multimedia file retrieval over the Internet has been counted as a popular service on the personal computer. However, the use of the Internet is shifting to the mobile devices especially smartphone. Various types of software have been proposed and sold via the operating system providers, such as iTunes of apple, Play Store of android, marketplace of Microsoft. As a result, the use of multimedia file via smartphone is becoming popular and tends to be general and widely used among users. Unfortunately, the loading system has a common problem which is a long delay during load. Due to the long delay time of the traditional client-server multimedia file retrieval service, the Mobile Cloud System (MCS) is proposed to solve the problem. Cloud computing architecture provides high efficiency of sharing the CPU and memories of different computing devices over the network. Therefore the MCS makes use of the vacant memory of a set of smartphones to shorten the downloading time. With the help of Cloud computing, users can receive multimedia file faster.

Department : Mathematics and Student's Signature.....

..... Computer Science.....

Field of Study : Computer Science and Advisor's Signature.....

..... Information Technology.....

Academic Year : 2012.....

ACKNOWLEDGEMENTS

I am certainly grateful to the Centre of Excellent in Mathematics or CEM who continuously supported this research.

I would like to express my thankfulness to the people who have helped me most throughout my thesis. I am sincerely grateful to my professor Assist Prof. Dr. Pattarasinee Bhattarakosol for continuously support for thesis; therefore, I am particularly obliged for the valuable assistance given contributions through constructive suggestions and recommendations.

Finally, my graduation would not have been achieved without best wish from my parents and my relation, who helped me in everything and always giving me greatest love, willpower and financial support until completion of my study and, lastly my heartfelt special thanks to my friends and my colleagues in INSET Lab. for their help and encouragement.

CONTENTS

	Page
ABSTRACT IN THAI	iv
ABSTRACT IN ENGLISH	v
ACKNOWLEDGEMENTS	vi
CONTENTS	vii
LIST OF TABLES	ix
LIST OF FIGURES	xi
CHAPTER I.....	1
1.1 Problem Statement	2
1.2 Objective	2
1.3 Scope of thesis	2
1.4 Definition	2
1.5 Structure of the Thesis	3
CHAPTER II.....	4
2.1 Literature Review	4
2.2 Cloud Computing	9
2.3 Mobile Computing	13
2.4 Agent based application	15
2.5 Peer to Peer sharing system	18
2.6 Android	19
CHAPTER III	24
3.1 Proposed Method	24
3.2 The design of the proposed system	26
3.3 The design of the Database	45
CHAPTER IV	48
4.1 Experiment Details	48
4.2 Results	56
4.3 Conclusion	61
CHAPTER V	63

	Page
5.1 Discussion	63
5.2 Limitation of Experiments	64
5.3 Conclusion	64
REFERENCES	66
BIOGRAPHY	71

LIST OF TABLES

	Page
Table 3.1 Specification of Client and Server application	31
Table 3.2 Multimedia_File_info table	46
Table 3.3 User_info table	46
Table 3.4 Cloud_info table	47
Table 4.1 Multimedia file Detail.....	49
Table 4.2 Devices Detail for Experiment 1	50
Table 4.3 File Part Detail for Experiment 1	50
Table 4.4 Devices Detail for Experiment 2	50
Table 4.5 File Part Detail for Experiment 2	50
Table 4.6 Devices Detail for Experiment 3	51
Table 4.7 File Parts Detail for Experiment 3	51
Table 4.8 Devices Detail for Experiment 4	51
Table 4.9 File Parts Detail for Experiment 4	52
Table 4.10 Devices Detail for Experiment 5	52
Table 4.11 File Parts Detail for Experiment 5	53
Table 4.12 Devices Detail for Experiment 6.1	54
Table 4.13 File Part Detail for Experiment 6.1	54
Table 4.14 Devices Detail for Experiment 6.2	54
Table 4.15 File Parts Detail for Experiment 6.2	54
Table 4.16 Devices Detail for Experiment 7.1	55
Table 4.17 File Part Detail for Experiment 7.1	55
Table 4.18 Devices Detail for Experiment 7.2	55
Table 4.19 File Parts Detail for Experiment 7.2	55
Table 4.20 The result of Experiment 1	56
Table 4.21 The result of Experiment 2	56
Table 4.22 The result of Experiment 3	57
Table 4.23 The result of Experiment 4	57
Table 4.24 The result of Experiment 5	58

Table 4.25 The result of Experiment 6.1	59
Table 4.26 The result of Experiment 6.2	59
Table 4.27 The result of Experiment 7.1	60
Table 4.28 The result of Experiment 7.2	60

LIST OF FIGURES

Figure 2.1 Cloud conceptual Diagram	9
Figure 2.2 Cloud Computing Layers	11
Figure 2.3 Mobile computing function	14
Figure 2.4 Agent-Oriented Software Development Process Model	16
Figure 2.5 Distributed Hash Table	19
Figure 2.6 Android Software Stacks [31]	22
Figure 3.1 Separation of Multimedia file	25
Figure 3.2 Function Module of the system	27
Figure 3.3 Class Diagram of Supporting Database	30
Figure 3.4 Foreground Use Case Diagram	32
Figure 3.5 Authentication Use Case Diagram	33
Figure 3.6 Select multimedia file Use Case Diagram	34
Figure 3.7 Download and Play Use Case Diagram	35
Figure 3.8 Background System Use Case Diagram	37
Figure 3.9 Define Location Use Case Diagram	38
Figure 3.10 Find proper seeder(s) Use Case Diagram	39
Figure 3.11 Download File Use Case Diagram	40
Figure 3.12 Delete parts of file Use Case Diagram	41
Figure 3.13 Class Diagram	42
Figure 3.14 Sequence Diagram for Download process	43
Figure 3.15 Sequence Diagram for Play process	45
Figure 4.1 The results of varying numbers of part file	61
Figure 4.2 Comparison among different file sizes under fixed delivery parts	62

CHAPTER I

INTRODUCTION

In the past ten years, users always obtain their enjoyable media from the Internet through PC-based system. Currently, most users obtain their entertainment through the mobile services. Unfortunately, not every user owns smartphone with high data rate transfer speed. Thus, these users, sometimes, have to wait for a long transmission from required multimedia servers.

In this decade, one technique to deduct the delay time of file retrieving over the Internet or over a network is the implementation of the cloud computing system. Presently, this cloud architecture turns to be a highly beneficial to many business organizations. This is because it provides high efficiency of sharing the CPU and memories of different computing devices over the network when these devices have low calling rates and small space usages. Thus, implementing a large size of computing equipments in one organization becomes an obsolete and useless concept.

Except from the smartphone, all new models of delivered mobile phones are built with a large memory space and high quality display screen. Therefore, most multimedia services are distributed to various locations or implemented under the cloud architecture to reduce the loading time of common mobile phone in the mobile market. However, the number of loaded multimedia files is continuously and rapidly increasing. As a result, the users cannot avoid the high delay during load.

According to the high needs of multimedia retrieval over the mobile devices, this paper proposes a new concept of applying cloud computing to serve the multimedia services of mobile users. The aim of this research is quite similar to the implementation of the cloud system with computers but this research presents a new concept of using mobile phones to act as clouds in the Mobile Cloud System (MCS).

This chapter states the interested problem in Section 1.1, and then the objective is described in Section 1.2. In Section 1.3, the scope and constraint of this thesis will be discussed, followed by definitions of technical terms in Section 1.4. Finally, the structure of this thesis is detailed in Section 1.6.

1.1 Problem Statement

In the present, the traditional client-server multimedia file retrieval application on mobiles has the long delay problem during download. The proposed system is developed to solve such problem. The proposed system also employs the member system to improve the security of the system and the cloud computing architecture to provide a better sharing CPU and memories of computing devices.

1.2 Objective

The objective of the proposed system is to implement the Mobile Cloud System for the multimedia file retrieval and presentation. The proposed system implements cloud computing and peer-to-peer combine.

1.3 Scope of thesis

Scope of work can be described as follows:

1. The experiment is performed on the Android operating system.
2. The presentation is in English language only.
3. The experiment must be applied to the smartphone only.
4. The communication is performed over 3G or Wi-Fi environment.

1.4 Definition

Member: The end user that already registered to the system. A member can be both seeders and the user depends on the action of that particular member.

User: A member of the system that wants to watch the multimedia file. The user has to log in to the system in order to request for a file.

Seeder: A member of the system that has been selected to has the part of the requested multimedia file stored in member's device. It is one of the sources for the part of requested file.

Private Cloud: The group of active seeders available in the system. Each active seeder contains the part of multimedia file that a user requested.

Download time: The time that the system uses from sending the request for a multimedia file until the file is played.

1.5 Structure of the Thesis

The rest of the thesis is organized as follows. The literature reviews and fundamental knowledge are presented in Chapter II. Chapter III describes the proposed method and its methodology. Follow by the experimental results in Chapter IV. Finally, discussions, limitations and conclusion are illustrated in Chapter V.

CHAPTER II

LITERATURE REVIEW AND FUNDAMENTAL KNOWLEDGE

In this chapter, it provides fundamental knowledge and literature review. There are related works reviewed in Section 2.1. Followed by, cloud computing, mobile computing, agent-based application, peer-to-peer file sharing system and Android platform are described in Section 2.2 to Section 2.6 respectively.

2.1 Literature Review

In the present, the multimedia content plays an important role as the Internet sources. With better bandwidths, media encoding, and ease-of-publishing limitations, multimedia contents became more popular over the Internet than the past. There are many multimedia-rich applications and services over the Internet providing access to a huge amount of multimedia contents.

Zlatka Avramova et.al. [1] conducted the data collection from five popular video portals over the world. This paper proposed the model to analyze the data for the pattern of the multimedia content popularity. According to the result of this analysis, the usage of the multimedia contents is exponentially growing in a time.

According to the research of [2], the quality improvement on streaming and access bandwidth causes the increasing of multimedia demands over the Internet. Moreover, the study of [3] stated that Multimedia-rich applications and services, for example, video on demand and IPTV, consumed more than 32% of the HTTP traffic.

As a consequence of the wireless technology development such as Wi-Fi or IEEE 802.11, the communication for content transferring is much easier and widely used around the world. Moreover, using this technology can create various advantages. For example, this technology is low cost of investment and maintenance, ease of installation, configuration and maintenance. One significantly important advantage is that users use unlicensed spectrum in their operations [4].

Not only the wireless development such as Wi-Fi is rapidly implemented, but also the mobile technology and its subscribers have been growing outstandingly. At the

end of 2010, there are four times of mobile cellular subscribers which are much higher numbers than the fixed telephone [5]. Thus, the popularity of the mobile computing is going to be the trend for the future lifestyle. Its characteristics such as the mobility and the diversity of network connection conditions tend to overcome the physical limitations, for examples, lack of agility and convenience of current mainstream computing paradigm [6].

Recent advancement in the wireless network technology and the increasing computing capability of mobile devices lead to the multimedia streaming services on the mobile application. This can be a new trend of the application usage in the communication society. Instead of using personal computers at home, Internet users are able to use their mobile devices to play multimedia files anywhere-anytime through available wireless distributors. Therefore, there are numerous demands on the video applications over the mobile wireless network. Some previous studies also suggested that Internet users tend to consume more multimedia information in the real time manner than the simple text based or picture based information [7][8].

One example of the mobile computing technology and the wireless network usages is the Mobile Learning System. The definition of the Mobile Learning System, or the M-Learning, is the system that supports the self-learning anytime-anywhere based on a mobile device [9]. The device must be capable for presenting the learning content. Additionally, it must support the wireless two-way communication occurring between teacher(s) and student(s). There are many researches regarding to the mobile learning application using the multimedia file streaming. The mobile learning has been widely accepted as an advanced education solution. It helps reducing problems regarding to the lack of learning time and resources.

Mobile devices and mobile wireless networks also have some capability limitations [10][11]. The current mobile learning technology is based on the client-server architecture as an electronic learning. In the client-server model, all multimedia and other types of contents are managed and distributed by servers. Therefore, the increasing of the number of clients causes the task overload to the servers. Due to the failure of the management, it leads to the latency of streaming and network collision [12]. As a consequence, other applications on the same network are affected.

To solve the problems stated above, a peer-to-peer file sharing system is employed. The advantage of the peer-to-peer (P2P) multimedia streaming applications is the minimization cost of bandwidth usage by leveraging peer uploading capacities on streaming servers. The available bandwidth among peers is importance, especially when the number of peers in the streaming session reaches an enormous amount. The number of available peers directly affects the efficiency of P2P streaming applications. Server bandwidth cost savings are very significant when there are many participating peers contribute their bandwidths on the streaming session [13]. With the P2P file sharing system, users at the same location can form their own virtual networks to share multimedia contents.

As the fact that one popular software commonly used to download multimedia files over the Internet is BitTorrent. This software has implemented a specific protocol that handles the multimedia file loading process. The main benefits of BitTorrent are its efficiency and the transferring speed of files. There are two types of peers defined by the application: downloaders, and seeds. Downloaders are peers who own some parts of the required file, while seeds are the peers who own the entire file but stay in the system for other peers to download some other parts of the file from them. The concept of BitTorrent is all files that are distributed are divided in small pieces with the typical size of 256KB. The group of peers that downloads a particular file at a time is called a swarm. Each swarm is independent, but each peer is able to participate in more than one swarm at a time. Each user can concurrently download pieces of files from different seeds. However, BitTorrent application also needs the tracker server to track every seed and downloaders in the system. Nonetheless, the streaming of video and other multimedia files is guaranteed both speed and the loading volume. Unfortunately, there is a lack of security policy for the entire system [14].

One of outstanding examples for P2P file sharing or streaming applications on mobiles using a wireless network is [15]. This paper deployed P2P file sharing with a streaming service on mobiles over a Wi-Fi network. Hu et al. developed a multimedia file sharing system among peers over a wireless network on mobiles in the year 2012. In this research, the Android platform is the based system for applications to perform the live streaming video sharing between two users using Android phones or tablets. The application allows users to capture video with a built-in camera in their mobile devices then share it with

other people nearby via a free wireless network. Routing Protocols are required to ease the multi-hop communication so the content can be sent further. The result of the experiment indicates that performance of the streaming over the mobile devices is acceptable.

Although the multimedia file sharing system has been implemented for years, they were implemented on a single server or distributed servers over the network. These implementation techniques still have some defects according to the volume of the loading files. As the amount of multimedia file in the system is increasing, the data flow at the server may reach the maximum capacity which possibly leads to failure at the server side. Thus, the efficient multimedia file sharing system and the effective streaming system, altogether, are required to reduce the workload of the server in the streaming service. Therefore, a cloud computing architecture was proposed to serve various needs of the file sharing system. This new architecture uses the available resources over the network to support someone's request. As a result, performance to serve the users is increased and the resource usage is worthwhile.

Many researches pointed out that the cloud computing technology is a suitable solution to deal with a multimedia file sharing system which has a large amount of data in the system [16]. The Google Docs service is a well-known cloud-based service over the Internet. Multiple users are allowed to work on the same document at the same time. Google employed remote cloud storages to store documents in the system. Moreover, there are many websites that provide content sharing services with the implementation of cloud storages such as Facebook and Twitter. The system proposed by C. Gadea et.al. aimed to achieve a real-time collaboration among users who are in a particular collaborative session based on the social network environment. The collaborative session is mainly regarding multimedia streaming for both entertainment and educational proposes. Every member in collaborative session will perceive the multimedia file at the exactly same moment as the rest of members in the session. Similar to documents, images and presentation slides, every member will see those media at the same page, including sound and voice. In addition, the synchronization tasks between members are handled transparently; therefore, developers do not need to consider about the necessity in the synchronization messages among users within the session. In addition, the system itself also provides ease-of-use APIs for users.

The cloud computing technology provides platforms and resources management for streaming and sharing system. As an advance in development of cloud computing, computing resources in the mobile computing area are virtualized and managed to a group of distributed computing units instead of traditional local computers or servers especially mobile computing units such as smartphones, portable terminals [17]. In the concept of virtualization, the cloud can be considered as a pool of virtual resources where all lower layers and hardware devices are virtualized. Since the complexity of various computing resources is hidden from end users, end users can access to their desired resources through a browser or any user interfaces. Then, the data from cloud computing providers are delivered to users without any concern of the complexity of physical storages.

The agent based application is a new paradigm of computing. It is able to provide various types of applications, services, with better availability and automation. On the behalf of the agent based application model, unlike the traditional client-server application, the agent works at the client side. The agent is capable of real-time and non-real-time services, configurations and installations. It is the effective solutions for the bottle neck problem at the server in traditional client-server model. Agents are being used in a wide variety of applications, ranging from small application such as email spam fillers to large and complex mission critical applications [18].

The agent-based application has been integrated with many types of application such as location-based application and mobile learning application [19]. S. Abdel-Naby et. al. integrated the agent based application with the technology that is commonly used in daily life. They are motivated to improve the existing location based services with a better application paradigm and communication technologies. In this research, they use BarterCel and Andiamo platform which are agent-based mobile service applications. In addition, X. Shen et. al. proposed an agent-based information push technology and apply it to the M-Learning system [20]. The purpose of the study is to utilize the use of the agent-based model with the M-Learning system. Therefore, mobile learners can access the latest information anytime-anywhere using all available devices. The agent can help the classification and processing of the huge amount of information in the push application. In addition, it can also manage the difficulties in evaluation and processing of the information and the automatic information processing.

The concept of devices and locations independence encourages the proposed system to employ the agent-based application along with the client application installed into every device. So, users are able to access the service anywhere and anytime.

Based on the content mentioned above, performance of the file sharing system increases according to two main factors. One factor is related to the software and the protocol mechanism while another factor is related to the network architecture for installing servers. Therefore, the system that combined both factors is proposed in this research in order to obtain the most reliable system with high performance of services.

2.2 Cloud Computing

The history of cloud computing began in 1960s when John McCarthy introduced this new concept of cloud computing. His idea was that “The computation may someday be organized as a public utility”. The word “cloud” was from the telecommunication industry when the telecom company offered the Virtual Private Network (VPN) with the qualified quality of service with good cost. Using the VPN, the service provider can balance the utilization of the overall network. So, cloud computing applies this advantage to improve the network infrastructure [21].

Recently, many leaders in the industrial world offered the Internet services and applications. For examples, Amazon Web Service, Google, and IBM have been researching and developing many cloud services, applications, and Eucalyptus which is the first open source cloud platform for the private cloud.

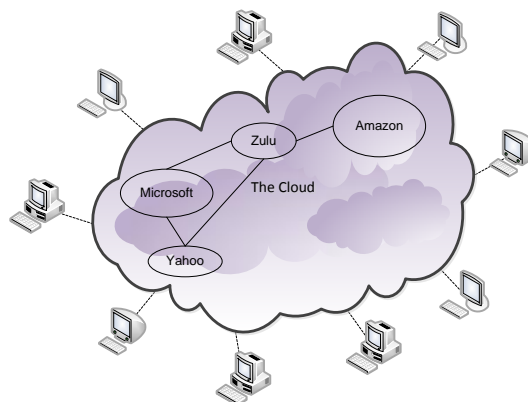


Figure 2.1 Cloud conceptual Diagram

Cloud computing is a virtual group of computing resources that consists of various computing devices. It provides software, platform, and environment for the Internet users. Cloud computing can manipulate all resources that includes system allocation, reallocation, and monitoring of the users' usages. In addition, all manipulations are performed automatically by the system software. Therefore, the user needs not to worry about the complexity of those tasks.

Cloud computing deploys the virtualization scheme so the system pictures all the resource as a pool of resource. The geographical data centers are not important for the cloud computing. It is so-called a flexible and extendable architecture network [22].

There are many unique and useful characteristics of cloud computing as listed below. Each of which is described as follow.

- Service oriented
- Strong fault tolerant
- Business model
- Ease use [23]

Service oriented

The concept of service oriented is driven by the abstraction and the accessibility. The cloud architecture is abstracted and transparent to the users. Abstraction helps users to use cloud computing without extra knowledge about the structure of the cloud. Therefore, users can easily access to the services.

Referring to Figure 2.2, the services provided by cloud computing are widely divided into three categories: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Details of each category are explained as follow.

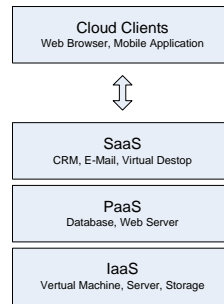


Figure 2.2 Cloud Computing Layers

- Infrastructure as a Service (IaaS) is a service that provides computing resources for clients, for example, processing unit, storage, and network. IaaS employs the on-demand scheme. Therefore, clients have to pay for the usage of their own. However, clients need not pay for the physical storages such as servers or network resources.
- Platform as a Service (PaaS) is a service that provides the development environment. Clients of this service are able to use the platform and the environment to create their own applications. PaaS acts like a middle man between the hardware and software.
- Software as a Service (SaaS) is a service that provides software for many users through the website. Users can access software anywhere anytime without concerning of installations and version updates. For service providers, they only need to maintain the software from servers. They are able to perform all updates or modifications from servers; users always use the latest version of the software.

Strong fault tolerant

In other types of computing such as parallel computing, there are still some fault tolerant problems. During the time users are using the application, there are chances of errors that can occur. For the huge or mission-critical application, there is a high cost for rerunning the application after an error causes the application unexpectedly terminated. Moreover, an

error from consuming huge computing resources leads to an error of the entire computing devices, storages, and the network. Fortunately, in the cloud computing services, each application almost has no effect to others.

Business model

Cloud computing is mainly developed for the IT industry. As a result, cloud computing supports the business model and grants profitable Return of Investment or ROI. The favorite case of the “how to pay” issue is the “pay per use”. The “pay per use” is the term described how users pay for services. The clients of cloud services will pay the service providers based on their usages of utilities services, for example, the processing power, storages, and networks.

This payment method of services is somehow economy for users. For users, they need not pay for all utilities by themselves since some utilities are unnecessary. The “pay per use” is also beneficial to the middle man. For small IT businesses, they are able to use the utilities of the cloud to create their own services and provide the services to the end users without worrying of the complexity of hardware and networks.

Ease use

User-friendly is another interesting advantage of cloud computing. Cloud computing user interface is the Internet-based interface that is simple and useful. Users are also familiar with the Internet interfaces for a certain of time; furthermore, the Internet is evolving. The Web 2.0 technology provides an effective and an interactive user interface for the Internet users. For example, the AJAX technology supports the website’s interface more effective and convenience as the application interface.

2.3 Mobile Computing

Mobile computing is a computing environment that can overcome the mobility issue. The user can access to the information anytime anywhere, and from any mobile devices. The user using the mobile computing environment is also able to interact with public websites or mobile web-applications via their mobile devices, such as retrieving information from websites, share their schedules with their co-worker through the collaborative workgroup application, and take a short memo in their mobile phones [24].

Other terms

Mobile computing is also known as many terms:

- Pervasive Computing: The computing environment that is pervasive and available in any environments.
- Ubiquitous Computing: Nobody can notice the presence of the computing environment. The user of the computing environment is able to use the local and remote services.
- Nomadic Computing: The computing environment is travelling along with the mobile users.

Characteristics

The characteristics of mobile computing are listed below:

- User Mobility: The user can move around physically from one place to another place while using the service. The services should be local and world-wide. For example, the same service should work both at home and in other countries.
- Network Mobility: Although users move from one network to another network, the service should perfectly work.
- Device Mobility: Users should be able to use any devices to access to the same application.

Functions

The function of mobile computing is divided into five major segments according to Figure 2.3.

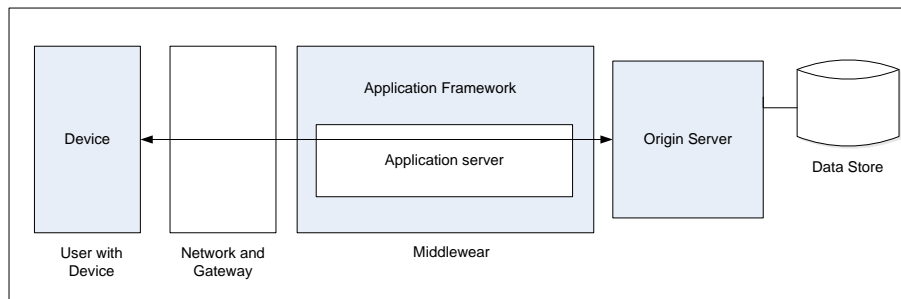


Figure 2.3 Mobile computing function

Referring to Figure 2.3, each segment has different characteristics and rules as follows.

- 1.) User's device: The devices using in mobile computing can be fixed or portable devices.
- 2.) Network: Since users are movable, the network should be wireless networks so they can use their mobile devices anywhere anytime.
- 3.) Gateway: The gateway provides interfaces for different bearers. It converts one transport bearer to another.
- 4.) Middleware: This function responsible for presenting data for each device. It also handles the security and privacy of each user.
- 5.) Origin server: It could be an application or a system. The origin server is able to access the storage devices.

Mobile computing Device

The device used in mobile computing devices could be either computing devices or communication devices. There are two main types of computing devices: fixed and portable computing devices. The example for fixed computing devices is personal computers and servers. In case of portable devices, the popular device is Personal Digital Assistant or PDA. However, communication devices could be either fixed devices or mobile devices, such as fixed telephone and mobile phone respectively.

When a computing unit is embedded into various equipment including communication devices such as mobile phones, smartphones was invented. It is the next generation of an effective device for communication. Human-Computer Interaction is the key to the effectiveness, efficiency and user experience. The smartphones and other mobile computing devices are becoming more user-friendly to the end users due to the effectiveness user interface and the smaller size of the devices.

Networks

- **Wired Networks:** This type of network is designed over the wired conductors. It is so called fixed network. For example, the fixed telephone network which is use the copper and fiber-optic as medium.
- **Wireless Networks:** Wireless Networks have another term as Mobile networks. There are many wireless networks connected by mobile devices. For example, PCS (Personal Cellular System), AMPS (Advanced Mobile Phone System), GSM, CDMA, DocoMo, and GPRS.
- **Ad-hoc Networks:** This type of network is the network that uses on a specific purpose. Sometimes it is a temporary connection in a small area. Ad-hoc Networks is established when mobile devices are in the same area using the plug-in module to connect to wireless connections, for example, Bluetooth or wireless LAN (802.11x).

2.4 Agent based application

Software application becomes more complex. Various environments are needed to support heterogeneous and distributed applications. Agent-oriented computing is an appropriate solution in the future software environment. The agent-oriented paradigm is capable of automatic and dynamic collaborations. It can handle the sophisticated interaction patterns [25].

The agent was first studied in 1950s by John McCarthy and was established by Oliver G. Selfridge several years later. In 1980s, the agent concept was widely spread and applied. The agent has three properties as follow:

- 1.) **Autonomy:** Agent itself can make the decision based on their own states without any interventions from software or human.
- 2.) **Adaptation:** Agent can recognize the environment and adapt itself to change in a suitable situation.
- 3.) **Cooperation:** Agent can interact with other agents through some communication. It is capable of cooperation among agents to accomplish their goals.

Since the agent-orientation paradigm is used in many areas of applications. The agent-oriented software model is important to develop the software. The agent-oriented software development process model is shown in Figure 2.4 [26].

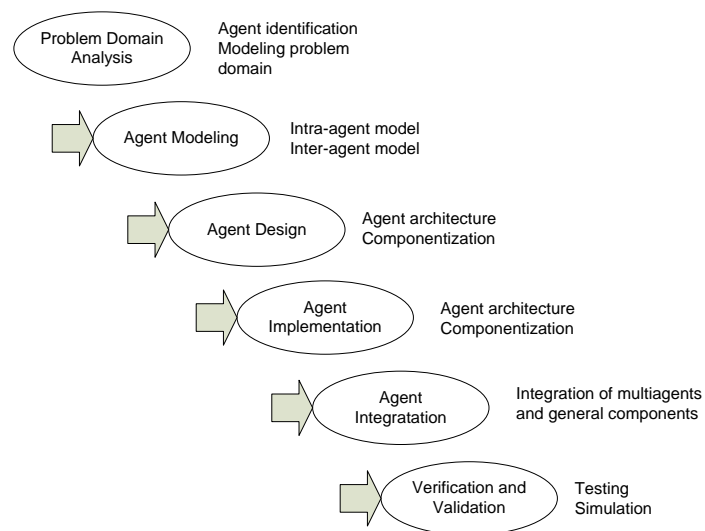


Figure 2.4 Agent-Oriented Software Development Process Model

Problem Domain Analysis

Problem analysis can be performed using the UML.

- **Use Case Diagram:** this diagram shows the set of use cases and actors and their relationships. It models the static aspect of the system.
- **Sequence Diagram:** this diagram shows the sequence of the interaction between object classes. It models the dynamic aspect of the system.
- **Class Diagram:** this diagram shows a set of classes, association and their relationships. It models the static design view of the system.

- Activity Diagram: this diagram shows the process flows from activity to activity. It models the dynamic aspect of system such as the system functions.

Intra-Agent Modeling

The intra-agent modeling is based on BDI model and Reticular Agent Mental Models. The characteristics of intra-agent are as follow.

- 1.) Goal model: Goal is an ultimate aim of an agent.
- 2.) Belief model: Belief is the information that an agent is already has. The information is regarding both an agent itself and the environment.
- 3.) Plan model: Plan is a behavior of an agent to succeed the goal. It changes when time flows.
- 4.) Capacity model: Capacity is all operations that an agent can perform.

Inter-Agent Modeling

For the multi-agent system, the mobility and exchanging messages are modeled in the inter-agent modeling. There are two models: Agent Mobile Model (AMM) and Agent Communication Model (ACM).

- 1.) Agent Mobile Model: this model shows how the agent migrates itself to perform their tasks. It also shows the mechanisms that decide the destination.
- 2.) Agent Communication Model: this model shows message exchanging among agents. It shows the log of communication between agents. It is also the information that is included in the message.

2.5 Peer to Peer sharing system

The term peer to peer file sharing system is the system that contains nodes which all nodes are equal. Nodes can communicate directly to other nodes in the network. The peer-to-peer nodes communicate with other nodes for sharing information or the computing resources. There are various examples of peer-to-peer system. Firstly, Seti-at-home is an Internet distributed computing application that uses the idle CPU of the personal computers to analyze the radio telescope data. Secondly, ICQ is an instant message application using peer-to-peer system. Finally, OceanStore focuses on sharing computing resources [27].

There are two types of architecture: structured system and unstructured system.

1.) Structured system

All peers are organized by specific criteria and algorithms. It uses worldwide protocol for nodes so they are able to find the node that requests for files or resources. The structured peer-to-peer system commonly uses the Distributed Hash Table (DHT) to assign the ownership of each file to the particular peers. This type of architecture is suitable for a large-scale implementation because of high scalability and the guaranteed performance of $O(\log N)$, where N is the number of nodes in the system.

2.) Unstructured system

Peers in unstructured peer-to-peer networks connect in an ad-hoc fashion based on a loose set of rules. In an ideal world, unstructured peer-to-peer systems have no centralized components or nodes. On the other than, in practical: there are three types of this architecture that employ various degrees of centralization.

- Pure peer-to-peer systems: the entire system consists of equipotent peers. There is no special node or infrastructure function.
- Centralized peer-to-peer systems: this type is used for indexing tasks and to initiate the entire system.

- Hybrid peer-to-peer systems: this type has the infrastructure nodes which is also called supernodes to help manipulate other nodes.

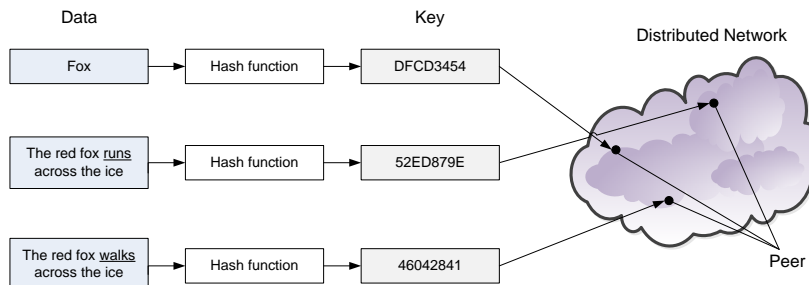


Figure 2.5 Distributed Hash Table

Peer-to-peer file sharing system has many advantages. First, all clients (known as peer) provide bandwidth, storages, and computing power which helps reducing the running costs of the original content distributors. Moreover, the entire capacity of the system also increases. Second, the peer-to-peer file sharing system increases robustness. When one peer in the system fails, the entire system is not damaged. Compare to the client-server file sharing system, if the system experiences any failures, the whole system will be collapsed. Finally, the peer-to-peer system needs no system administrator. The system is easy to initiate and run automatically by the effective protocol and its own management system. It does not require the staffs for the maintenances task.

Due to the decentralized network aspect, security issues are introduced. The peer-to-peer system is somehow vulnerable to the malicious pieces of codes which can remotely access to the file in users' computers. Furthermore, once the harmful intruders gain the access to one of the peers in the system, it may compromised the entirely system [28].

2.6 Android

Android is an open-source software platform for developing application for mobile devices. It is Linux-based operating system. Android is a combination of these three components: open-source operating system for mobile devices, open-source development platform for mobile application, and mobile devices such as mobile phones that run Android operating system and the application create for this operating system [29].

Android, Inc. was founded in 2003 by Andy Rubin. It was working on software for mobile phones. In 2005, Google purchased Android Inc. and used it as supplementary of Google since Google planned to move into mobile phone market. Android has been continuously developing their platforms for applications on mobile phones. Since 2008, Android developed a number of the improved Android operating system versions named after sugary treats, for example, Android 1.5 Cupcake and 1.6 Donut. The first commercial phone that runs Android operating system version 2.6 is HTC Dream. It is launched in October, 2008 [30].

Since Android is an open-source operating system, the applications developed on this operating system is an open-source as well. Android provides development environment called as the Software Development Kit (SDK). The features of Android are listed below.

- No licensing or development fee.
- Wireless connection, for example Wi-Fi, Bluetooth, and Near Field Communication are supported.
- GSM, EDGE, 3G and upper levels are supported for telephony.
- Many APIs for the basic functions such as location-based services, hardware sensors, and media file recorders and players.
- Capable of running background applications and processes.
- 2D and 3D graphic libraries are provided for application developers.

The application runs in Android operating system is written in JAVA programming language and executed by Dalvik virtual machine instead of JAVA Virtual Machine. Each of Android application runs in a separate process in their own Dalvik instance. It hands over the responsibility for the computing resources, memory and process management to the Android run time. Dalvik and the Android run time are on top of the kernel which handles low-level hardware operations. Android software stack, Software Development Kit, and Dalvik virtual machine are described below.

Android Software Stack

Android software stack is composed of components shown in Figure 2.6. The details of each component are described as follow.

- **Linux kernel:** It is the core elements, including hardware drivers, process and memory management, networks, security and power management. They are all handles by Linux 2.6 kernel. This component provides the abstraction layer from hardware and the rest of the software stack.
- **Libraries:** It is run above the kernel. The libraries are various of C/C++ core libraries such as libc and SSL. The examples of the libraries are:
 - Media library for the playback of video and audio file
 - Display management for the surface
 - Graphic libraries
 - SQLite for native database
 - SSL and Webkit for the Web browser and Internet security
- **Android run time:** This is the component supports the Linux-based environment to be the Android operating system so all Android applications can be run. Android run time is an engine that powers up the applications with the use of core libraries and the Dalvik virtual machine.
 - **Core libraries:** Android application is written in JAVA language but Dalvik virtual machine is not the JAVA virtual machine. Therefore, the Core Android Libraries are required to provide functions that available in JAVA libraries as well as in Android libraries.
 - **Dalvik virtual machine:** It is a register-based virtual machine. This virtual machine has been modified to ensure that the device is able to run multiple instances.
 - **Application framework:** It provides classes that are used to develop Android applications. It also provides the abstraction

for the hardware accessing, user interface and application resources.

- Application layer: Applications run on Android phone are built on the application layer using the same API libraries. It runs within Android run time and uses classes and services available in the application framework.

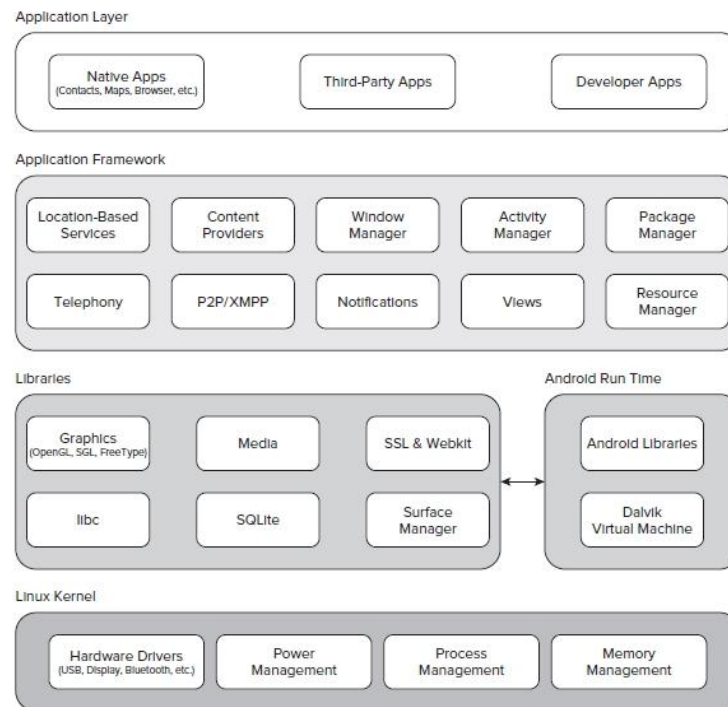


Figure 2.6 Android Software Stacks [31]

Software Development Kit

Software Development Kit, or SDK, provides everything that the developers need to use, test, and debug the application.

- **Android APIs:** The main component of SDK is the Android API libraries which provide the access to the Android stack. They are the libraries that Google uses for developing the native Android applications.
- **Development tools:** The SDK provides development tools that developer can write, compile, and debug the Android applications.

- The Android Virtual Device Manager and Emulator: It is a fully interactive Android device emulator. The emulator runs in Android Virtual Device. It simulates the hardware configuration for devices. The developers are able to test the application using the emulator.
- Full-documentation: The code-level reference information provides details for the packages and classes available in Android libraries. It also provides fundamental knowledge of Android applications required for the developers.

Dalvik Virtual Machine

Dalvik Virtual Machine is another key element of Android. It is a custom virtual machine for Android designed to assure that multiple instances run effectively on a single device. The Dalvik virtual machine uses the Linux kernel for the low-level functionality, for example, security, thread and process, and memory management. It is also support for C/C++ applications [31].

CHAPTER III

METHODOLOGY

Since the multimedia file sharing is a significant issue for all mobile users, this chapter describes the proposed method to enhance the downloading ability of such problem. The content includes the overall architecture, modules of the system, and the diagrams that have been designed for the Mobile Cloud System. Therefore, Section 3.1 describes the proposed method. Section 3.2 illustrates the design of the system. Finally, the database that has been used in the system is shown in Section 3.3.

3.1 Proposed Method

The proposed method, the Mobile Cloud System (MCS), is designed by combining two factors to enhance the security and speed of a multimedia file sharing system based on the cloud computing. This method integrates the software efficiency, a new protocol mechanism, and the cloud computing architecture to gain the suitable solution in the problem mentioned above.

3.1.1 Security issue

In order to overcome the security issue, a member system is employed. This is because the system requires the members' authentication before accessing to the Central Server (CS) every time users request for a multimedia file. As a result, the system is secure and tractable. The information regarding all members and their activities is stored in the CS; therefore, the proposed system can control, manage and manipulate source for all multimedia files available in the system.

3.1.2 Performance issue

Since the objective of this research is to solve the low download rate from the multimedia providers, the speed is the key performance indicator of the proposed mechanism. According to the fact that transferring a very small and fixed length of packet can increase the transfer speed, this technique is applied to this cloud system on mobile clients. Therefore, the requested multimedia file will be

divided into many small parts with a static length. However, the number of file separations will be depended on the total length of the required file, as shown in Figure 3.1. For example, to enhance the speed of the streaming and to reduce the buffering time, each multimedia file in Youtube.com with approximately 15 minutes long is divided into n parts according to the length of video file. When the target member requests for the multimedia file, the CS reroutes the sources of the streaming to n members' devices or seeder in the private cloud and stream the particular file concurrently to the target server.

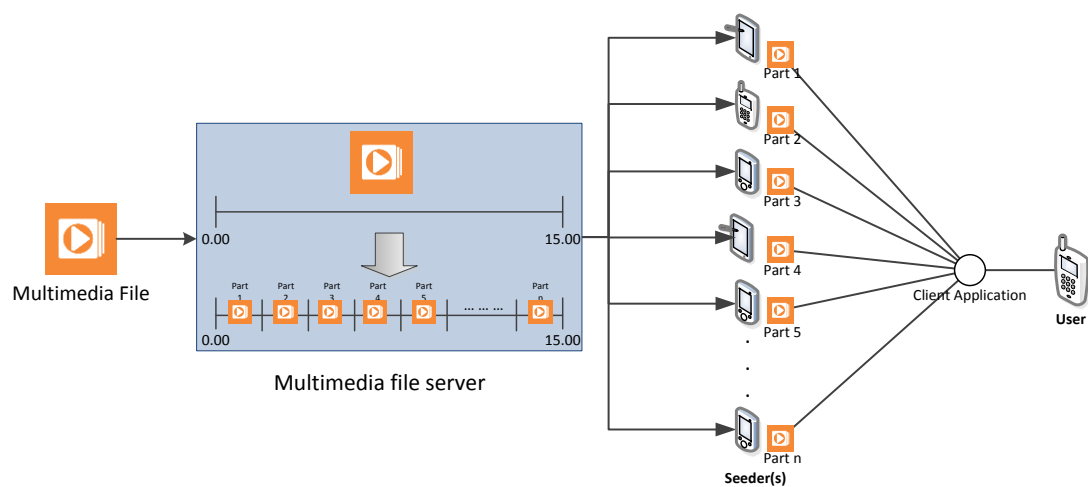


Figure 3.1 Separation of Multimedia file

Moreover, after the target member watched the requested multimedia file, the mobile agent destroys all parts of the file, except a small selected part. This small part is stored in the device's memory as a source for others to prevent the intellectual property problem. As a result, it also increases the number of members in the private cloud.

3.2 The design of the proposed system

The design of the system is divided in six parts: Software Specification, Architecture and Modules, System Implementations, Use case diagrams, Class diagrams, and Sequence diagrams. Details of each part are described as follow.

3.2.1 Software specification of the MCS

The Mobile Cloud System consists of 2-site applications: server-site application, and client-site application.

The server application is the server-site software mainly developed with the server-site script language called Hypertext Preprocessor or PHP. The server application performs various tasks as listed below.

- 1.) Authentication: The server application will receive the information from a client application. It uses the information to verify the identity of the user who wants to log in to the system.
- 2.) Database management: The server application is able to fetch, add, and update the information stored in the database of the system
- 3.) Information transfer: The server application is responsible for the communication between the application server and the client application.

The client application is an Android application. The client application is responsible for the following processes.

- 1.) Providing a user interface between a user and the system.
- 2.) Exchange the information regarding the authentication, location of user, seeder(s) information and the status of the download, and play task.
- 3.) Background tasks
 - a) Download all parts of the file concurrently from the seeder(s).
 - b) Destroy the unused parts of the multimedia file and display the multimedia file to the user.

3.2.2 Architecture and Modules of the MCS

Referring to Figure 3.2, the architecture of the MCS consists of four main components: the Private Cloud (PC), the Client Application (CA), the Central Server (CS), and the Supporting Database (SD). The CS is responsible for handling the request from the CA installed in the members' device and SD managing. It also monitors and manipulates the workload of the proposed system. The CA is the program that must be installed into every member's device to manage the client site's activities. In addition, it maintains the connection between the member's device and the CS. Furthermore, it is also used to connect between others in the PC. The PC consists of active members in the system that connected directly to the CS via the CA. The information that flows in the system will be stored in the SD, member's data, and multimedia file properties. Details of each module in the MCS are elaborated as follows.

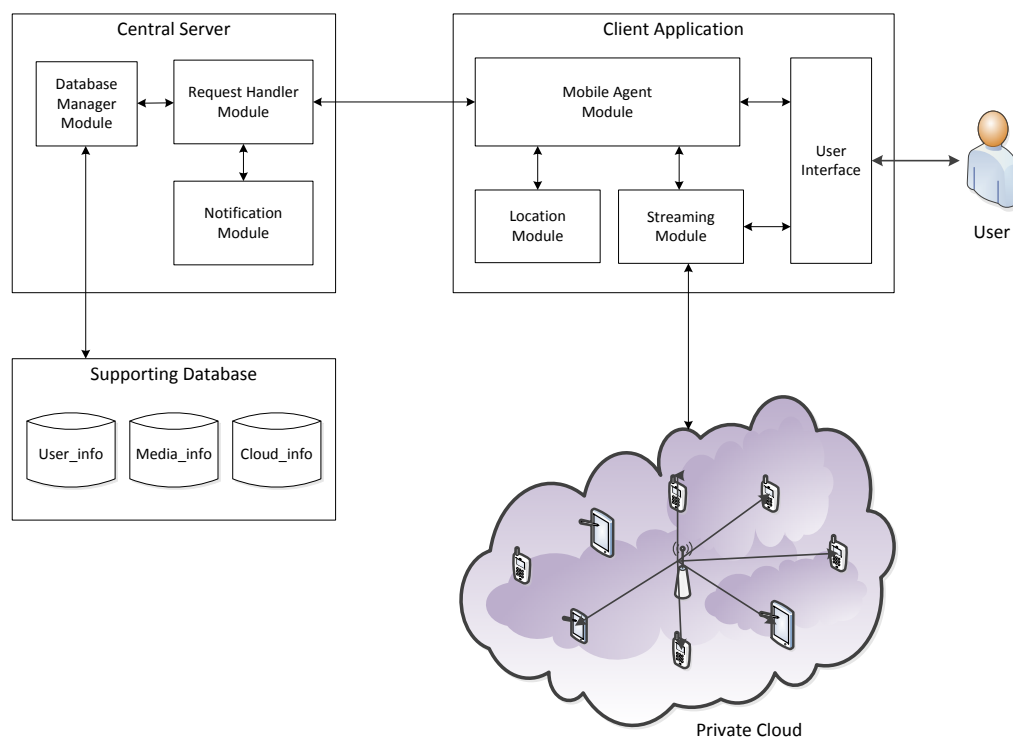


Figure 3.2 Function Module of the system

- **Private Cloud (PC)**

The private cloud consists of every active Android mobile device within the system. The mobile agent has been installed in each mobile device in order to store a part of multimedia files that the member already watched. It is also used to connect the device to the CS. When the target member wants to watch a multimedia file, other devices act as the multimedia sources and stream them concurrently to the target member.

- **Client Application (CA)**

The client application, or CA, is an interface between users and the central server. Members can log in to the server, view the list of multimedia files and manipulate the playback via the application. The application also works in the background to ensure that the streaming service run smoothly. It consists of three modules.

- **Interface Module (IM)**

The interface module handles every interaction between members and the central server. The IM module provides a log in interface for users to log in to the system and the multimedia file list available for members to choose. It also receives the information regarding the sources of the streamed multimedia file from the central server.

- **Streaming Module (SM)**

The streaming module manages the streaming service from the private cloud. Users can manipulate the playback using this module.

- **Mobile Agent Module (MAM)**

The mobile agent is a small part of the program embedded in the application required for every member in the system. It connects to the server as soon as the device

has connected to the Internet and works in the background in order to stay connecting to the server. The proposed system is assumed that it has fully access to the specified memory of member's device through the mobile agent. It is responsible for managing the streaming service, both upstream and downstream in the background.

- Location Module (LM)

The location module is a location-based application. The LM fetches the location information of the current coordinate in the form of latitude and longitude. This location information of the users will be sent to the CS; therefore, the CS can assign the best clouds for them.

- **Central Server (CS)**

All users need to authenticate themselves and send their locations to the CS before using the system. Therefore, the CS must be connected to the SD in order to check the users' statuses. If the user is an authorized member, then the CS will generate the list of available multimedia files to the user, and update the information of the user in the User_Adr_Lat and User_Adr_Long. Once a multimedia file is selected, all IP addresses and ports of the related private cloud components are sent to the user's device to retrieve the entire file. According to Figure3.1, there are three modules of the CS to perform the retrieving process: the Request Handler Module (RHM), the Database Management Module (DBMM), and the Notification Module (NM). Details of these modules are presented as follow.

- Request Handler Module (RHM)

This module handles all requests from members who want to watch the multimedia file. It is responsible for authenticating each user, generating the list of the available multimedia files in the system, and searching for available

members in the private cloud to be the sources of the streaming service.

- DataBase Management Module (DBMM)

The DBMM is an interface between the CS and the Supporting Database. It can fetch, search, and generate the result for the query of the database.

- Notification Module (NM)

This module is responsible for sending the notification to all clouds and the client in the streaming service.

- Supporting Database (SD)

According to the process of the CS, a database system must be implemented, named as the supporting database (SD). Referring to Figure 3.3, the SD contains three significant tables: the Multimedia_File_info, the User_info, and the Cloud_info.

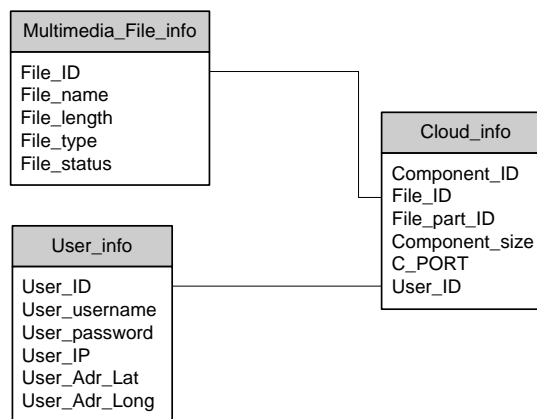


Figure 3.3 Class Diagram of Supporting Database

3.2.3 The Proposed System Implementations

In order to implement the proposed system, there are three subsystems to be implemented: server-site subsystem, client-site subsystem, and database subsystem. Details of these implementations are elaborated in Table 3.1.

1.) The server-site subsystem

The implementation at the server-site is the web-based application. The core software is written in PHP language along with the basic HTML language. It is run on the remote web server; the specification of the server is Dell PowerEdge 860 Xeon Pro 3050 2.13Ghz Dual Core processor.

2.) The client-site subsystem

The client-site implementation is the Android application. As mentioned in Android Platform in Chapter II, Android applications are developed using JAVA with the environment provided from Android Software Development Kit. The client application is run on the Android device with the Android operating system version 2.3 or upper version.

3.) The database implementation

The database used in the MCS is implemented by SQL language. The development tools is the phpMyAdmin web-based application.

Table 3.1 Specification of Client and Server application

No	Tools	Server Implementation	Client Implementation
1	Operating System	Linux	Android
2	Programming Language	PHP, HTML	JAVA
3	Device	Web Server	Android smart phone
4.	Developer Tools	Notepad++,Web Browsers	Eclipse, Android SDK

3.2.4 Use case diagrams

According to the architecture and modules of the system shown in Figure 3.1, the use case diagrams of the system are divided into two main diagrams: foreground process diagram and background process diagram. These diagrams can be drawn out to be many sub-diagrams.

- Use case of the foreground activities:

In order to describe a big picture of the foreground activities, the main use case of the foreground activities must be drawn, as shown in Figure 3.3. Then, all activities in this diagram can be extended to be the following use case diagrams

- Use case for the Authentication activity
- Use case for the Selecting file activity
- Use case for Download and play activity

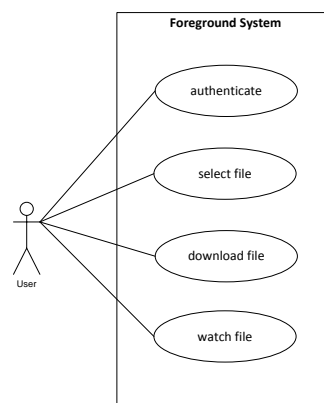


Figure 3.4 Foreground Use Case Diagram

Name: Foreground System

Participant actor: User

Entry Condition:

- 1.) User is a member of the system.
- 2.) User starts the Client Application on a mobile phone.
- 3.) User's mobile phone connects to a Wi-Fi network.

Flow of events:

- 1.) User authenticates with the system.
- 2.) The system verifies user as a member of the system.
- 3.) The system shows the available multimedia file list.
- 4.) User selects multimedia file.
- 5.) User downloads the multimedia file.
- 6.) User watches multimedia file.

Exit condition: User closes the application.

The use case of activities in the main use case of the foreground process can be shown in the following contents.

- Authentication Use Case Diagram: this diagram, as shown in Figure 3.4, presents the activities in the authentication process of the system.

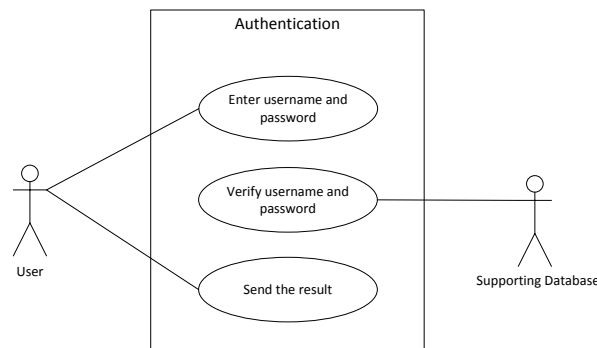


Figure 3.5 Authentication Use Case Diagram

Name: Authentication

Participant actor:

- 1) User
- 2) Supporting Database
- 3) Central

Entry Condition:

- 1) User is a member of the system.

- 2) User starts the Client Application on a mobile phone.
- 3) User's mobile phone connects to a Wi-Fi network.

Flow of events:

- 1) User enters username and password.
- 2) Username password and location information of the user are sent to the system.
- 3) The system verifies username and password with the Supporting Database.
- 4) The system stores location information of the user in the Supporting Database.
- 5) The system shows the result of information to the user.
- 6) The system notifies the CS the result of the authentication.

Exit condition: User close the application

- Select multimedia file Use Case Diagram: this diagram, as shown in Figure 3.5, presents the activities in the multimedia selection process of the system.

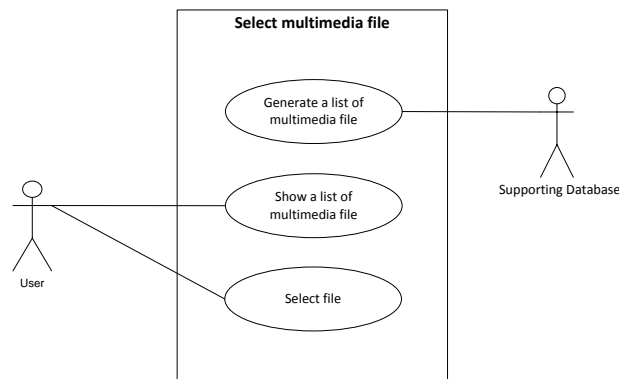


Figure 3.6 Select multimedia file Use Case Diagram

Name: Select multimedia file

Participant actor:

- 1) User
- 2) Supporting Database
- 3) Central Server

Entry Condition:

- 1) User is a member of the system.
- 2) User starts the Client Application on a mobile phone.
- 3) User's mobile phone connects to a Wi-Fi network.

Flow of events:

- 1.) The system generates list of available multimedia files in the system by getting the data from Supporting Database.
- 2.) The system shows a list of available multimedia file to user.
- 3.) User selects preferred multimedia file.

Exit condition: User selects the multimedia file.

○ Download and Play Use Case Diagram: this diagram, as shown in Figure 3.6, presents the activities in the download and play multimedia files.

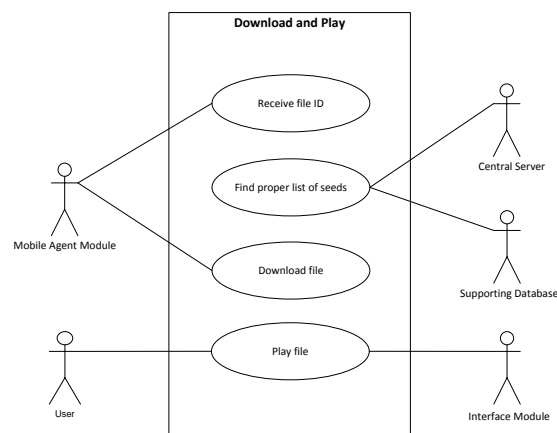


Figure 3.7 Download and Play Use Case Diagram

Name: Download and play

Participant actor:

- 1.) User
- 2.) Supporting Database
- 3.) Central Server
- 4.) Interface Module
- 5.) Mobile Agent Module

Entry Condition:

- 1.) User is a member of the system.
- 2.) User starts the Client Application on a mobile phone.
- 3.) User already selected the multimedia file.

Flow of events:

1. The system receives the selected file ID from the selecting file system.
2. Central Server finds proper list of seeders for the selected file from the Supporting Database.
3. Mobile Agent Module downloads the selected file.
4. User watches the file using the Interface Module.

Exit condition: User closes the application.

- Use case of the background activities:

Although the foreground activities are important in the users' view, the background activities are ignorable since these background activities support all activities of the foreground. Then, the entire picture of the background activities is explained using the main background use case, as shown in Figure 3.7. Consequently, all activities in the main use case can be elaborated in other four use case diagrams as follow.

- Use case for Define location
- Use case for Find proper seeder(s)
- Use case for Download and play
- Use case for Delete parts of file

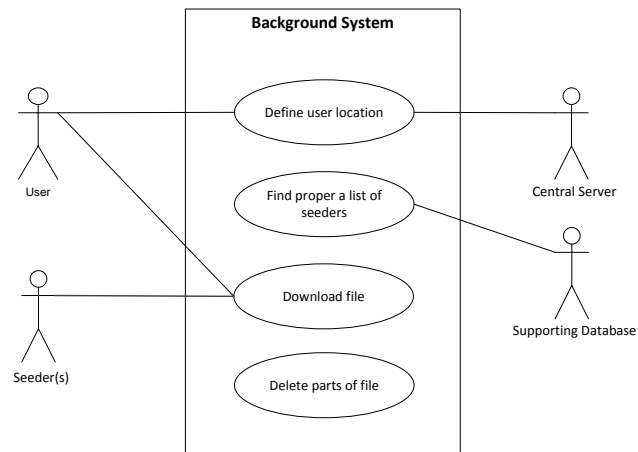


Figure 3.8 Background System Use Case Diagram

Name: Background System

Participant actor:

- 1.) User
- 2.) Central Server
- 3.) Supporting Database
- 4.) Seeder(s)

Entry Condition:

- 1.) User is a member of the system.
- 2.) User starts the Client Application on a mobile phone.
- 3.) User's mobile phone connects to a WiFi network.

Flow of events:

- 1.) The system retrieves user's location from the Supporting Database.
- 2.) The system searches for the nearest seeder(s) for the requested file from Supporting Database.
- 3.) The system notifies the Streaming Module of a list of seeder(s).
- 4.) The Streaming Module downloads file(s) from seeder(s).
- 5.) User watches the file.
- 6.) The system deletes every parts of the file except the selected part.

Exit condition: User closes the application.

The use case of activities in the main use case of the background process can be shown in the following contents.

○ Define location Use Case Diagram: The diagram in Figure 3.8 shows the activities to identify the location of the user.

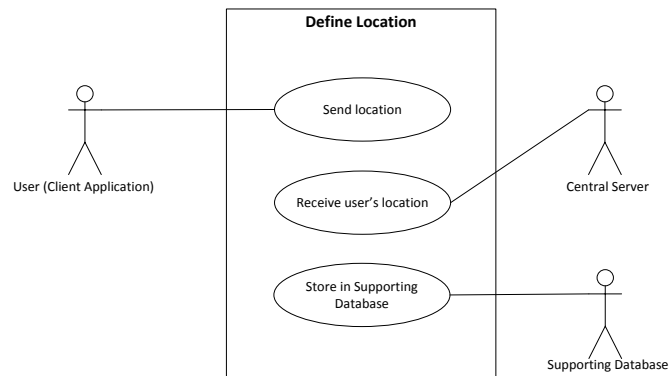


Figure 3.9 Define Location Use Case Diagram

Name: Define Location

Participant actor:

- 1.) User
- 2.) Central Server
- 3.) Supporting Database

Entry Condition:

- 1.) User is a member of the system.
- 2.) User starts the Client Application on a mobile phone.
- 3.) User's mobile phone connects to a Wi-Fi network.

Flow of events:

- 1.) Client application sends the location of the user to the system.
- 2.) The system receives the location of the user.
- 3.) The system stores the location of the user in the Supporting Database.

Exit condition: User closes the application.

- Find proper seeder(s) Use Case Diagram: This use case describes all activities for finding the proper providers for users.

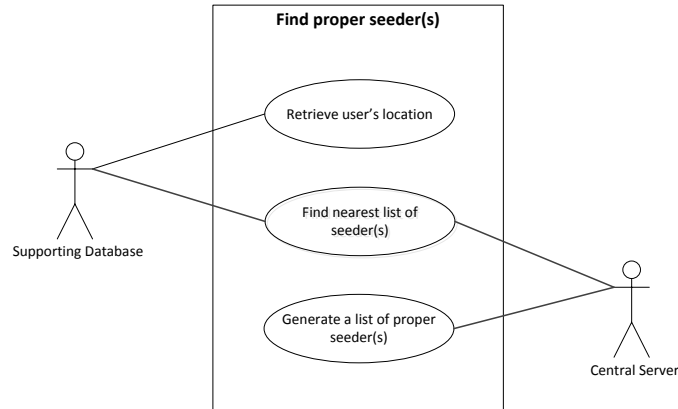


Figure 3.10 Find proper seeder(s) Use Case Diagram

Name: Find proper seeder(s)

Participant actor:

- 1.) Central Server
- 2.) Supporting Database

Entry Condition:

- 1.) User is a member of the system.
- 2.) User starts the Client Application on a mobile phone.
- 3.) User's mobile phone connects to a Wi-Fi network

Flow of events:

- 1.) The system retrieves user's location from the Supporting Database.
- 2.) The system retrieves possible seeders' location from the Supporting Database.
- 3.) The system calculates the distance between user and seeder(s) to find the nearest seeder(s) for the requested file.
- 4.) The system generates a list of seeder(s).

Exit condition: User closes the application

- Download file Use Case Diagram: The diagram in Figure 3.10 shows all activities to download the required file.

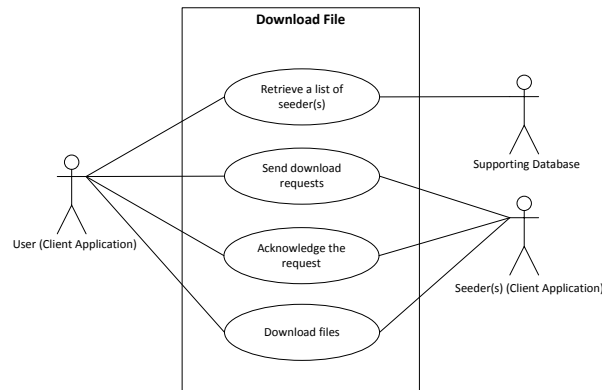


Figure 3.11 Download File Use Case Diagram

Name: Download File

Participant actor:

- 1.) User (Client Application)
- 2.) Seeder(s) (Client Application)
- 3.) Supporting Database

Entry Condition:

- 1.) User is a member of the system.
- 2.) User starts the Client Application on a mobile phone.
- 3.) User's mobile phone connects to a Wi-Fi network.

Flow of events:

- 1.) User (Client Application) retrieves a list of seeder(s)'s IP Address and Port from the Supporting Database.
- 2.) User (Client Application) sends requests to download files to all seeder(s).
- 3.) Seeder(s) (Client Application) send acknowledge back.
- 4.) User (Client Application) downloads files from all seeder(s).

Exit condition:

- 1.) User closes the application.
- 2.) The files are downloaded.

- Delete parts of file Use Case Diagram: When the file is completely downloaded, some parts of the file will be deleted from the user's mobile phone as drawn in Figure 3.11.

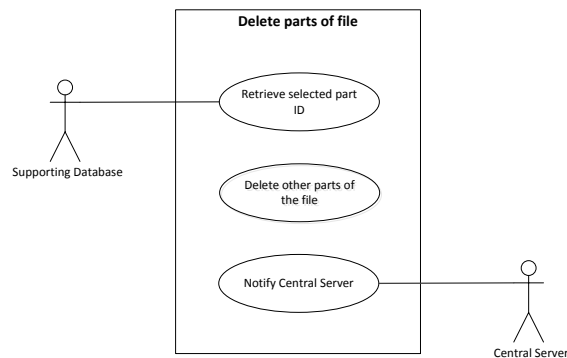


Figure 3.12 Delete parts of file Use Case Diagram

Name: Delete parts of file

Participant actor:

- 1.) Central Server
- 2.) Supporting Database

Entry Condition:

- 1.) User is a member of the system.
- 2.) User starts the Client Application on a mobile phone.
- 3.) User's mobile phone connects to a WiFi network.
- 4.) User watched the file.

Flow of events:

- 1.) The system retrieves the selected part ID from the Supporting Database.
- 2.) The system deletes all parts of file downloaded except the selected part.
- 3.) The system notifies the Central Server.

Exit condition:

- 1.) User closes the application.
- 2.) Delete parts of file.

3.2.5 Class Diagram

According to the use case diagrams of the system, there are eight classes that can be generated and the relations among classes are presented in Figure 3.13.

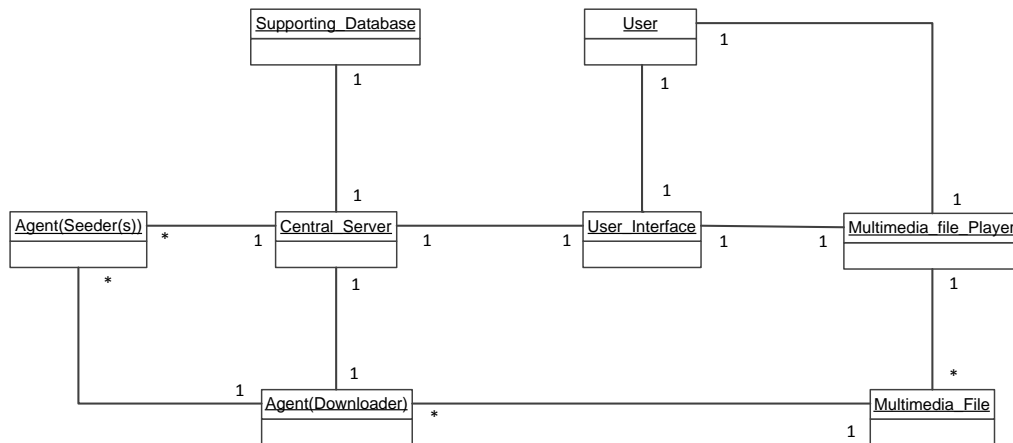


Figure 3.13 Class Diagram

According to Figure 3.13, eight classes are listed below.

- 1.) Class Central_Server
- 2.) Class Supporting_Database
- 3.) Class Agent(Downloader)
- 4.) Class Agent(Seeder(s))
- 5.) Class User
- 6.) Class User_Interface
- 7.) Class Multimedia_file_player
- 8.) Class Multimedia_File

Based on Figure 3.12, Class User uses Class User_interface to interact with the system both sending request for the multimedia file to Class Central_Server and watching the multimedia file from Class Multimedia_file_player. Class Multimedia_file_player receives the multimedia file from Class Multimedia_file, and stores multimedia file in the user's device. Class Central_Server is responsible for fetching the information from Class Supporting_database to authenticate the authorized user and fetching the information from Class Supporting_database to find the proper list of seeder(s) for the requested multimedia file. Moreover, Class Central_Server manages Class Agent(Downloader) in the transferring task with other Agent(Seeder(s)) in the seeder(s) of the requested multimedia file. After finishing downloading the multimedia file, Class Agent (Downloader) delivers the multimedia file to Class Multimedia_file to be stored in the user's device.

3.2.6 Sequence Diagram

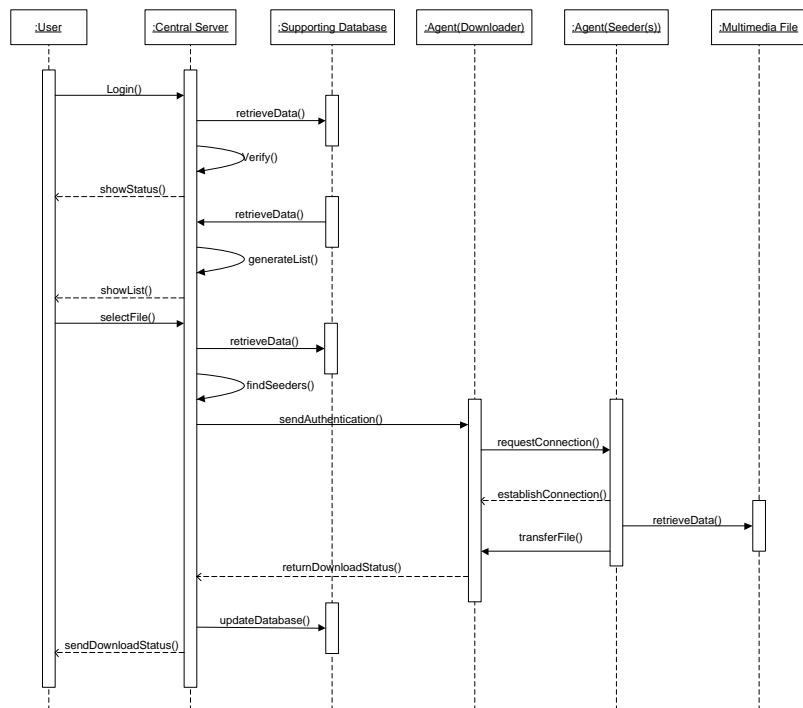


Figure 3.14 Sequence Diagram for Download process

Based on the use case of the background activities, the sequence of the download process for the required multimedia file is illustrated by Figure 3.14. The process begins when a user, an object in the Class User, (**:User**), logs into the system using the client application in the mobile phone. The user needs to enter the username and the password to authenticate the required access while the modules location is fetched by the application. The authentication information and the location are sent to the object of Class Central_Server (**:Central_Server**). Then, the CS verifies the username and the password using information from objects in the Class Supporting_Database (**:Supporting_Database**). In addition, the server stores information of user's authentication and location in the SD. As the user is authorized by the CS, Class Central_Server retrieves the list of multimedia files from the SD and generates the list of available multimedia file. The list of all available multimedia files is presented to the user.

After presenting the list, the user selects the multimedia file as requested. The information of selected multimedia file is sent to the CS. Then, the CS finds the nearest seeders of the selected file using the information from the SD. After the CS found the proper list of seeders, the CS sends the information about the seeders to an object in Class Agent(Downloader) (**:Agent(Downloader)**) that is the user's Agent. This object sends the request for connecting to all objects of Class Agent(Seeder) (**:Agent(Seeder)**) that is the seeders' Agent of the requested file. When the objects of Class Agent(Seeder) receive the request, the acknowledge is replied and the connections are established. After the connections are established, seeders start to concurrently transfer objects of Class Multimedia_files (**:Multimedia_File**) to the object of Class Agent(Downloader). When the download completed, the object of Class Agent(Downloader) sent a notification to the CS. As a result, the CS updates the SD and notifies the user.

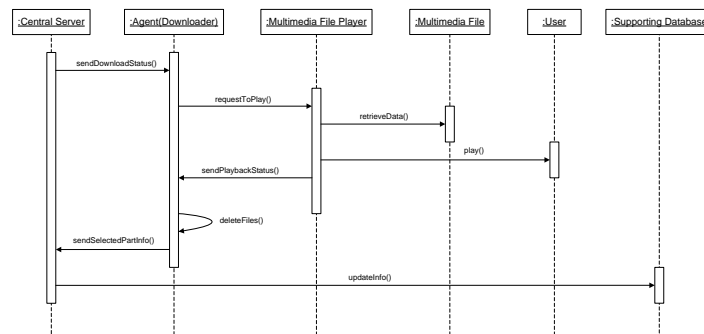


Figure 3.15 Sequence Diagram for Play process

After finishing the download process, the user can watch the multimedia file through the client application. The sequence of this process is illustrated in Figure 3.15. When the download succeeds, the object of Class **Central_Server (:Central_Server)** sends the download status to an object of Class **Agent(Downloader) (:Agent(Downloader))**; this object sends a request to the object of Class **Multimedia_File_Player (:Multimedia_File_Player)** to play the file. The object of Class **Multimedia_File_Player** retrieves the multimedia file from an object of Class **Multimedia_File (:Multimedia_File)** to play the object of Class **User (:User)**.

After the multimedia file is played, the object of Class **Multimedia_File_Player** notifies the playback status to the object of Class **Agent(Downloader)**. Therefore, the object of Class **Agent(Downloader)** deletes all parts of the multimedia file except the selected part. Then, this object reports the CS regarding the stored part of the multimedia file. The CS updates the information of the new seeder of the multimedia file using the object of Class **Supporting_Database (:Supporting_Database)**.

3.3 The design of the Database

The Supporting Database in the system consists of three tables: **Multimedia_File_info**, **User_info**, and **Cloud_info**

3.3.1 **Multimedia_File_info**: This table, Table 3.2, contains all characteristics of multimedia files, such as the file name, keywords, type of the retrieved file, and file length.

Table 3.2 Multimedia_File_info table

Field	Datatype	Explanation	note
File_ID	INT(40)	Identification code of multimedia file	Primary key, not null, Auto Increment
File_Name	VARCHAR(40)	Name of multimedia file	Not null
File_Length	VARCHAR(40)	Length of multimedia file	Not null
File_Type	VARCHAR(40)	Type of multimedia file	Not null
File_Status	VARCHAR(40)	Status of available file	Not null
File_Size	INT(40)	Size of multimedia file	Not null

3.3.2 User_info: This table, Table 3.3, contains the information regarding all members in the system. For example, ID, username, password and reference address.

Table 3.3 User_info table

Field	Datatype	Explanation	note
User_ID	INT(11)	Identification code of user	Primary key, not null, Auto Increment
User_username	VARCHAR(40)	Username of user	Not null
User_password	VARCHAR(40)	Password of user	Not null
User_Adr_Lat	VARCHAR(40)	Latitude value of current location of user	Not null
User_Adr_Long	VARCHAR(40)	Longitude value of current location of user	Not null
User_IP	VARCHAR(40)	IP of user's device	Not null

3.3.3 Cloud_info: This table, Table 3.4, stores the information regarding to the parts of the multimedia file that have been stored in the members' devices.

Table 3.4 Cloud_info table

Field	Datatype	Explanation	note
Component_ID	INT(11)	Identification code of component	Primary key, not null, Auto Increment
File_ID	VARCHAR(40)	Identification code of the multimedia file of the particular component	Not null
File_part_ID	VARCHAR(40)	Identification code for unique part of file	Not null
User_ID	VARCHAR(40)	Identification code of user that has the particular component	Not null
Component_size	VARCHAR(40)	Size of component	Not null
C_PORT	VARCHAR(40)	The port that the particular element will be transfer	Not null

CHAPTER IV

EXPERIMENTAL RESULT

This chapter will illustrate experimental results from the proposed method and provide the trend of the time consumed in the downloading file process. The experimental details are described in Section 4.1, the results and the conclusions are demonstrated in Section 4.2 and Section 4.3 respectively.

4.1 Experiment Details

The experiments for the proposed system can be categorized in two aspects. Firstly, the number of parts to deliver the multimedia file is varying while another varies the sizes of the multimedia file when the number of parts is fixed. Details of these two different experiments have objectives to confirm the performance of the proposed method under variety environments.

4.1.1 The performance measurement under various delivery parts

The experiments are conducted five times with various numbers of the file's parts. All experiments are set up in the INSET Laboratory, Faculty of Science, Chulalongkorn University with the same Wi-Fi connections. The CS and the SD are located in the remote Web-Server called "<http://www.insetlab.com/CS>".

The experiments consist of two significant participants. The first participant is the CS that runs the server application. The second participant, running the client application, is mobile devices that acting as seeders and users. On condition in this experiment is the stable Wi-Fi connection.

Fibonacci numbers is a well-known sequence of integer. The first two numbers of the sequence is 0 and 1. The other numbers are the sum of the previous two numbers. Fibonacci numbers is used in many studies and researches. One of the applications of Fibonacci number is the merge sort algorithm. An unsorted list is divided into two lists which each list has length according to Fibonacci numbers. By dividing a list using Fibonacci numbers, each list's length is approximately equal to the proportion of ϕ (32).

The various numbers of parts of the file in the experiments are determined according to the Fibonacci numbers. Therefore, the sequence of the number of the parts of file is increasing in the proportion of ϕ . As a result, the results of the experiments are able to state the significant trend of the increasing of total download times.

The multimedia file is divided using the Fibonacci number:

$$F_n = F_{n-1} + F_{n-2}$$

n is an positive integer, where

$$F_1 = 0, F_2 = 1$$

The number of parts of the file in Experiment 1, 2 and 3 will be 3, 5, and 8 parts respectively. The details of each experiment are as follow.

The device used for the user in the experiments is Samsung Galaxy Note II with Android Operating System version 4.2.2.

The detail of the multimedia file used in the experiment is as followed in Table 4.1 Multimedia file Detail.

Table 4.1 Multimedia file Detail

File Name	Ponprom Ep.1-1 08 Feb 2013.3gp
File ID	1
File Size	10.87 MB
File type	3gp
File Length	13.01 min
Source	Youtube.com

- Experiment 1

In this experiment, the requested multimedia file is a full multimedia file.

Experiment Detail

Place: INSET Laboratory, Department of Mathematics and Computer Science,
Faculty of Science, Chulalongkorn University

Network: Science_wifi_e26

Table 4.2 Devices Detail for Experiment 1

Device No.	Device	OS	User_ID	File_Name
1	Samsung : Galaxy Note 10.1	Android 4.2.2	2	1_1.3gp

Table 4.3 File Part Detail for Experiment 1

Part No.	File_part_ID	Part_Name	Part_Size (MB)	Part_Length (min)
1	1	1_1.3gp	10.87	13.01

- Experiment 2

In this experiment the requested multimedia file is divided into two parts.

Experiment Detail

Place: INSET Laboratory, Department of Mathematics and Computer Science,
Faculty of Science, Chulalongkorn University

Network: Science_wifi_e26

Table 4.4 Devices Detail for Experiment 2

Device No.	Device	OS	User_ID	File_Name
1	Samsung : Galaxy Note 10.1	Android 4.2.2	2	1_1.3gp
2	Samsung : Galaxy Y	Android 2.3.6	3	1_2.3gp

Table 4.5 File Part Detail for Experiment 2

Part No.	File_part_ID	Part_Name	Part_Size (byte)	Part_Length (min)
1	1	1_1.3gp	5,695,450	6.30
2	2	1_2.3gp	5,630,297	6.31

- Experiment 3

In this experiment the requested multimedia file is divided into three parts.

Experiment Detail

Place: INSET Laboratory, Department of Mathematics and Computer Science,
Faculty of Science, Chulalongkorn University

Network: Science_wifi_e26

Table 4.6 Devices Detail for Experiment 3

Device No.	Device	OS	User_ID	File_Name
1	Asus : Nexus 7	Android 4.2.2	2	1_1.3gp
2	Samsung : Galaxy S III	Android 4.1.2	3	1_2.3gp
3	Samsung : Galaxy Nexus	Android 4.1.1	4	1_3.3gp

Table 4.7 File Parts Detail for Experiment 3

Part No.	File_part_ID	Part_Name	Part_Size (MB)	Part_Length (min)
1	1	1_1.3gp	3.14	04.20
2	2	1_2.3gp	3.69	04.21
3	3	1_3.3gp	3.54	04.20

- Experiment 4

In this experiment the requested multimedia file is divided into five parts.

Experiment Detail

Place: INSET Laboratory, Department of Mathematics and Computer Science,
Faculty of Science, Chulalongkorn University

Network: Science_wifi_e26

Table 4.8 Devices Detail for Experiment 4

Device No.	Device	OS	User_ID	File_Name
1	Asus : Nexus 7	Android 4.2.2	2	1_1.3gp
2	Samsung : Galaxy S III	Android 4.1.2	3	1_2.3gp
3	Samsung : Galaxy Nexus	Android 4.1.1	4	1_3.3gp
4	Oppo : Find my guitar	Android 2.3.6	5	1_4.3gp
5	Samsung : Galaxy Y	Android 2.3.6	6	1_5.3gp

Table 4.9 File Parts Detail for Experiment 4

Part No.	File_part_ID	Part_Name	Part_Size (MB)	Part_Length (min)
1	1	1_1.3gp	2.061	2.35
2	2	1_2.3gp	2.356	2.41
3	3	1_3.3gp	2.248	2.35
4	4	1_4.3gp	2.310	2.35
5	5	1_5.3gp	2.296	2.35

- Experiment 5

In this experiment the requested multimedia file is divided into eight parts.

Experiment Detail

Place: INSET Laboratory, Department of Mathematics and Computer Science,
Faculty of Science, Chulalongkorn University

Network: Science_wifi_e26

Table 4.10 Devices Detail for Experiment 5

Device No.	Device	OS	User_ID	File_Name
1	Samsung : Galaxy Note 10.1	Android 4.2.2	2	1_1.3gp 1_2.3gp
2	Samsung : Galaxy Y	Android 2.3.6	3	1_3.3gp 1_4.3gp
3	HP Pavilion HPE : Window 7	Android 4.1.1	4	1_5.3gp 1_6.3gp 1_7.3gp 1_8.3gp

Table 4.11 File Parts Detail for Experiment 5

Part No.	File_part_ID	Part_Name	Part_Size (byte)	Part_Length (min)
1	1	1_1.3gp	1,454,391	1.37
2	2	1_2.3gp	1,041,117	1.37
3	3	1_3.3gp	1,466,425	1.38
4	4	1_4.3gp	1,532,627	1.37
5	5	1_5.3gp	1,547,165	1.37
6	6	1_6.3gp	1,518,459	1.38
7	7	1_7.3gp	1,479,175	1.37
8	8	1_8.3gp	1,551,583	1.40

4.1.2 The performance measurement under various file sizes

The experiments are conducted three times with the three file sizes. In these experiments, there are two numbers of parts of multimedia files which are one and three parts of multimedia file.

Each size of the multimedia file used in these experiments is according to the quality of the video file in Youtube.com. The first size is from “144p” quality option which is 10.87 MB. The second size of the file is from “240p” quality option which is 21.7 MB. Finally, the last size of the file is from “480p” quality option which is 36.67 MB.

- Experiment 6 (“240p”)

In this experiment, the requested multimedia file is a “240p” quality option which its size is 21.7 MB.

Experiment Detail

Place: INSET Laboratory, Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University

Network: Science_wifi_e26

1.) One part of multimedia file

Table 4.12 Devices Detail for Experiment 6.1

Device No.	Device	OS	User_ID	File_Name
1	Samsung : GalaxyNote 10.1	Android 4.2.2	2	1_1.3gp

Table 4.13 File Part Detail for Experiment 6.1

Part No.	File_part_ID	Part_Name	Part_Size (MB)	Part_Length (min)
1	1	1_1.3gp	21.7	13.01

2.) Three parts of multimedia file

Table 4.14 Devices Detail for Experiment 6.2

Device No.	Device	OS	User_ID	File_Name
1	Asus : Nexus 7	Android 4.2.2	2	1_1.3gp
2	Samsung : Galaxy S III	Android 4.1.2	3	1_2.3gp
3	Samsung : GalaxyNexus	Android 4.1.1	4	1_3.3gp

Table 4.15 File Parts Detail for Experiment 6.2

Part No.	File_part_ID	Part_Name	Part_Size (MB)	Part_Length (min)
1	1	1_1.3gp	8.9	04.20
2	2	1_2.3gp	8.9	04.21
3	3	1_3.3gp	8.9	04.20

- Experiment 7 (“480p”)

In this experiment, the requested multimedia file is a “240p” quality option which its size is 36.65 MB.

Experiment Detail

Place: INSET Laboratory, Department of Mathematics and Computer Science,
Faculty of Science, Chulalongkorn University

Network: Science_wifi_e26

1.) One part of multimedia file

Table 4.16 Devices Detail for Experiment 7.1

Device No.	Device	OS	User_ID	File_Name
1	Samsung : Galaxy Note 10.1	Android 4.2.2	2	1_1.3gp

Table 4.17 File Part Detail for Experiment 7.1

Part No.	File_part_ID	Part_Name	Part_Size (MB)	Part_Length (min)
1	1	1_1.3gp	36.65	13.01

2.) Three parts of multimedia file

Table 4.18 Devices Detail for Experiment 7.2

Device No.	Device	OS	User_ID	File_Name
1	Asus : Nexus 7	Android 4.2.2	2	1_1.3gp
2	Samsung : Galaxy S III	Android 4.1.2	3	1_2.3gp
3	Samsung : Galaxy Nexus	Android 4.1.1	4	1_3.3gp

Table 4.19 File Parts Detail for Experiment 7.2

Part No.	File_part_ID	Part_Name	Part_Size (MB)	Part_Length (min)
1	1	1_1.3gp	12.2	04.20
2	2	1_2.3gp	12.2	04.21
3	3	1_3.3gp	12.2	04.20

4.2 Results

4.2.1 The performance measurement under various delivery parts

- Experiment 1

Table 4.20 The result of Experiment 1

File_name	File_Size (byte)	File_length	Experiment #1 (ms)	Experiment #2 (ms)	Experiment #3 (ms)
1_1.3gp	11,133,707	13.01	138561	101591	128024
Start time			43:24.1	47:12.7	49:28.7
End Time			45:44.6	48:54.2	51:36.7
Total (sec)			138.561	101.591	128.024
Average (sec)			122.725		

- Experiment 2

Table 4.21 The result of Experiment 2

File_name	File_Size (byte)	File_length	Experiment #1 (ms)	Experiment #2 (ms)	Experiment #3 (ms)
1_1.3gp	5,695,450	6.3	89311	93719	109112
1_2.3gp	5,630,297	6.31	65285	62487	92816
Start time			06:35.8	10:42.3	13:19.4
End Time			08:05.1	12:16.0	15:08.5
Total (sec)			89.311	93.719	109.112
Average (sec)			97.38		

- Experiment 3

Table 4.22 The result of Experiment 3

File_name	File_Size (byte)	File_length	Experiment #1 (ms)	Experiment #2 (ms)	Experiment #3 (ms)
1_1.3gp	3,828,847	4.2	30324	93135	47426
1_2.3gp	3,780,694	4.21	60932	113077	46490
1_3.3gp	3,622,960	4.2	72086	91176	29748
Start time			14:45.9	21:40.4	41:52.6
End Time			15:58.0	23:33.4	42:40.0
Total (sec)			72.085	113.08	47.525
Average (sec)			77.563		

- Experiment 4

Table 4.23 The result of Experiment 4

File_name	File_Size (byte)	File_length	Experiment #1 (ms)	Experiment #2 (ms)	Experiment #3 (ms)
1_1.3gp	2,110,134	2.35	26879	31795	27638
1_2.3gp	2,412,435	2.41	27031	42749	41112
1_3.3gp	2,301,125	2.35	49711	22170	37962
1_4.3gp	2,364,890	2.35	35540	19654	12234
1_5.3gp	2,351,064	2.35	43266	38548	43991
Start time			41:51.0	49:31.2	13:48.0
End Time			42:40.7	50:14.0	14:32.0
Total (sec)			49.715	42.75	43.995
Average (sec)			45.487		

- Experiment 5

Table 4.24 The result of Experiment 5

File_name	File_Size (byte)	File_length	Experiment #1 (ms)	Experiment #2 (ms)	Experiment #3 (ms)
1_1.3gp	1,454,391	1.37	37744	34361	17142
1_2.3gp	1,041,117	1.37	34075	28921	36727
1_3.3gp	1,466,425	1.38	37290	32121	37973
1_4.3gp	1,532,627	1.37	38087	36804	38370
1_5.3gp	1,547,165	1.37	15364	18933	31788
1_6.3gp	1,518,459	1.38	24428	19434	21065
1_7.3gp	1,479,175	1.37	23924	20729	30252
1_8.3gp	1,551,583	1.4	26313	21527	31259
Start time			52:22.2	00:15.7	24:01.2
End Time			53:00.3	00:52.5	24:39.5
Total (sec)			38.12	36.805	38.375
Average (sec)			37.767		

4.2.2 Varying the sizes of the multimedia file

- Experiment 6 (“240p”)
 - Experiment 6.1

Table 4.25 The result of Experiment 6.1

File_name	File_Size (byte)	File_length	Experiment #1 (ms)	Experiment #2 (ms)	Experiment #3 (ms)
1_1.3gp	22,220,093	13.01	161698	145565	169100
Start time			52:20.3	58:03.9	00:43.3
End Time			55:01.9	00:29.5	03:32.4
Total (sec)			161.698	145.565	169.1
Average (sec)			158.787		

- Experiment 6.2

Table 4.26 The result of Experiment 6.2

File_name	File_Size (byte)	File_length	Experiment #1 (ms)	Experiment #2 (ms)	Experiment #3 (ms)
1_1.3gp	9,147,093	4.2	107021	95124	43650
1_2.3gp	9,120,469	4.21	80490	67812	113123
1_3.3gp	9,138,254	4.2	63662	60727	115314
Start time			13:48.2	18:42.9	24:00.7
End Time			15:35.2	20:18.1	25:56.1
Total (sec)			107.021	95.124	115.314
Average (sec)			105.819		

- Experiment 7
 - Experiment 7.1

Table 4.27 The result of Experiment 7.1

File_name	File_Size (byte)	File_length	Experiment #1 (ms)	Experiment #2 (ms)	Experiment #3 (ms)
1_1.3gp	37,543,963	13.01	223934	187801	227086
Start time			07:41.7	18:01.7	29:28.4
End Time			11:25.6	21:09.5	33:15.4
Total (sec)			223.934	187.801	227.086
Average (sec)			212.94		

- Experiment 7.2

Table 4.28 The result of Experiment 7.2

File_name	File_Size (byte)	File_length	Experiment #1 (ms)	Experiment #2 (ms)	Experiment #3 (ms)
1_1.3gp	12,510,379	4.2	79358	181325	179183
1_2.3gp	12,518,104	4.21	180685	180808	142522
1_3.3gp	12,544,417	4.2	182744	98129	85256
Start time			27:54.3	42:00.6	46:28.7
End Time			30:57.0	45:02.0	49:27.9
Total (sec)			182.744	181.325	179.183
Average (sec)			181.084		

4.3 Conclusion

4.3.1 The conclusion of The performance measurement under various delivery parts

The results from the Experiment 1 to 5 show the trend of the decreasing download time. According to the various numbers of parts of the file from 1 part to 5 parts, the download time used for downloading multimedia file is decreased when the number of the part of the file is increased as shown in Figure 4.1.

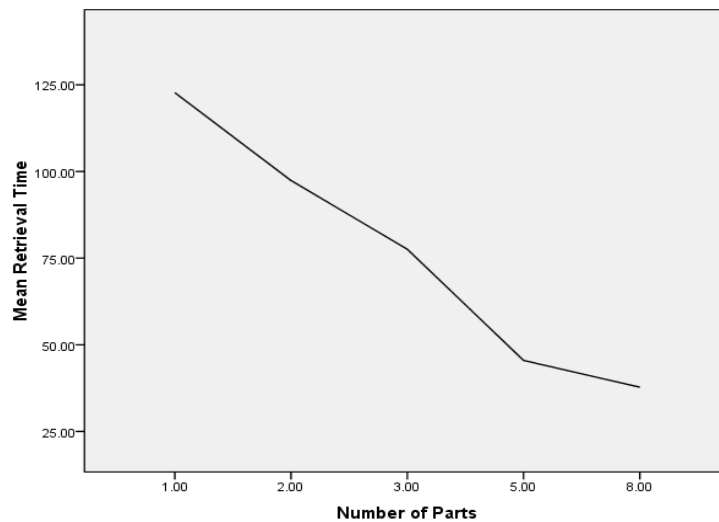


Figure 4.1 The results of varying numbers of part file

4.3.2 The conclusion of the performance measurement under various file sizes

The results from the experiment 6 and experiment 7 elaborate trends. In addition, comparisons among three experiments show that the time consumed for downloading a multimedia file decreases when the number of parts of the required file is increasing. Moreover, for a larger multimedia file, the trend of the download time is still the same as the smaller multimedia file as shown in Figure 4.2. From Figure 4.1, it showed that the download time has potential to be reduced when the number of the part of the file is increasing. Referring to Figure 4.2, the dash line represents the time for only one part to be downloaded so the delivery time of the larger file size can be reduced when the number of parts of the file is three.

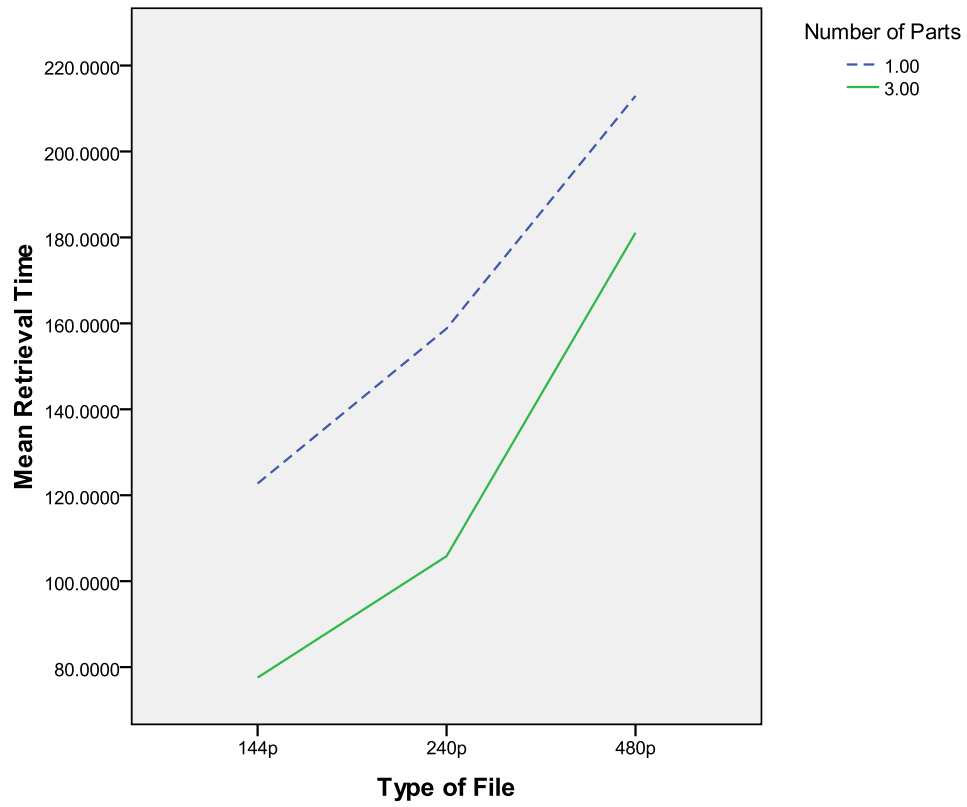


Figure 4.2 Comparison among different file sizes under fixed delivery parts

CHAPTER V

DISCUSSTION AND CONCLUSION

In this chapter, the discussion will be discussed in Section 5.1, limitation of the experiment stated in Section 5.2 and finally conclusion will be drawn in Section 5.3.

5.1 Discussion

Multimedia file retrieval and playing is becoming more and more popular due to the increasing popularity of the Internet. Because of better bandwidths, media encoding, and ease-of-publishing limitations, there are many multimedia-rich applications and services are available in the market for Internet users to access to a numerous multimedia content.

The improvement in the wireless network technology and the increasing computing capability of mobile devices lead to the multimedia file retrieval and play over mobile applications. The traditional client-server multimedia file retrieval application on mobiles may experience the long delay problem during download when users of the system are dramatically increasing.

Many techniques and aspects are discussed earlier in Chapter II to solve the problem stated in Chapter I. One of the computing technique that is employed to solve this problem is the use of Cloud computing. The cloud architecture is a highly beneficial to many business models. Cloud computing provides high efficiency of sharing CPU s and memories of different computing devices over the network when these devices have low calling rates and small space usages.

The peer-to-peer file sharing system is another key to the solutions of the long delay problem. Since multimedia file shared in the system is distributed to various locations or implemented under the cloud architecture to reduce the loading time of common mobile phones in the mobile market.

The Mobile Cloud System proposed in this thesis is a solution for the multimedia file retrieval and play on mobile phone. It makes use of many techniques to solve the problem. For the security issue, a member system is employed to help ensuring the security of the system. It builds the system secured and tractable. The requested multimedia file used in the experiment was divided into many small parts with static length to help solving the performance issue. As transferring a very small and fixed length of packet can increase the transfer speed.

Thus, the result in this research shows that the concept of the Mobile Cloud System is useful and effective for multimedia file retrieval and playing in mobile phone. However, there are some limitations in the experiments that has been stated in the next section

5.2 Limitation of Experiments

The main limitation of the experiment is the lack of testing environment. In order to test the entire system with full scales, it requires many android devices to be the seeders of the system. Another limitation is the wireless connection. The Wi-Fi connection used in the experiment is not much stable as expected so the result may not be precise.

5.3 Conclusion

According to the implementation of the Internet, the browse of multimedia files can be counted as a popular service on any personal computers. In the same way, people also call for multimedia files using their smartphones or mobile devices. However, these users must wait for a long delay until the entire file is stored and ready to play. Thus, the enjoyment cannot be perfectly fulfilled according to the limitation of the service that the provider granted to smartphones.

In order to solve the transferring problem of a multimedia file through smartphones, this proposed method the use of the cloud computing technology where a set of smartphones is a component of the entire system, called as Mobile Cloud System (MCS). The main concept of the MCS is based on the fact that people might not use the entire phone's memory. Thus, this remaining memory can be shared for others in the same group of the

service system. So, once a multimedia file is retrieved from the main server, a small particle of this file will be extracted and saved into the calling member mobile device. As a consequence, the next user can use this particle to shorten the retrieval time in the case that the previous users are nearby the new user. The result of using the proposed architecture for the MCS is the fast retrieval process such that members support members to gain the service satisfaction.

REFERENCES

- [1] Avramova, Z., Wittevrongel, S., Bruneel, H., and Vleeschauwer, D. D. Analysis and Modeling of Video Popularity Evolution in Various Online Video Content Systems: Power-Law versus Exponential Decay. The First International Conference on Evolving Internet (INTERNET 2009) (Aug 2009) : 95-100.

- [2] Klasen, B. Social, Fast, Efficient: Content Distribution in Hybrid Networks. The Sixteenth IEEE Symposium on Computers and Communications (ISCC'11) (July 2011) : 61-67.

- [3] Maier, G., Feldmann, A., Paxson, V., and Allman, M. On dominant characteristics of residential broadband internet traffic. Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference (2009) : 90–102.

- [4] Elianos, F. A., Plakia, G., Frangoudis, P. A. and Polyzos, G. C. Structure and Evolution of a Large-Scale Wireless Community Network. 2009 IEEE International Symposium on a World of Wireless, Mobile, and Multimedia Networks (WoWMoM 2009) (June 2009) : 1-6.

- [5] Mshvidobadze, T. Evolution Mobile Wireless Communication and LTE Networks. Application of Information and Communication Technologies (AICT) 6th International Conference (October 2012) : 1-7.

- [6] Ibrahim, I.K. Mobile multimedia: Bringing users into focus. 49th International Symposium ELMAR-2007 (September 2007) : 7.

- [7] Wong, C. F., Fung, W. L., Tang, C. F. J., and Chan, S.-H. G. TCP Streaming for Low-Delay Wireless Video. Second International Conference on Quality of Service in Heterogeneous Wired/Wireless Network (Aug 2005) : 6-41.

- [8] Yang, F., Zhang, Q., Zhu, W., and Zhang, Y. Q. End-to-end TCP-friendly streaming protocol and bit allocation for scalable video over wireless Internet. IEEE Journal on Selected Areas in Communications 22, 4 (2004) : 777-790.
- [9] Dye, A., Solstad, and K'Odingo, J. A. Mobile Education - A Glance at The Future. NKI Distance Education (Norway) Report [Online]. 2009 Available from : http://www.dye.no/articles/a_glance_at_the_future [2013, March 28]
- [10] Jiugen, Y., Ruonan, X., and Jianmin, W. Applying Research of Mobile Learning Mode in Teaching. 2010 International Forum on Information Technology and Applications 3 (July 2010) : 417-420.
- [11] Yi, C. Research and implementation of multiserving-oriented mobile learning application based on eLML. The 2010 International Conference on Computer Application and System Modeling (ICCSM 2010) 8 (October 2010) : 619- 624.
- [12] Hayakawa, T., Higashino, M., Takahashi, K., Kawamura, T., and Sugahara, K. Management of Multimedia Data for Streaming on a Distributed e-Learning System. 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA 2012) (March 2012) : 1282-1285.
- [13] Wu, C., Li, B., and Zhao, S. Characterizing Peer-to-Peer Streaming Flows. IEEE Journal on Selected Areas in Communications 25, 9 (2007) : 1612-1626,200.
- [14] Liu, P., Wang, W., and Tang, Y. An Effective Approach for Video-on-Demand Streaming Services Based on BitTorrent. International Symposium on Computer Network and Multimedia Technology (CNMT) (2009) : 1-4.

- [15] Hu, P., Symons, N., Indulska, J., and Portmann, M. Share your view : wireless multi-hop video streaming using Android phones. 2012 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops) (2012) : 782-787.
- [16] Gadea, C., Solomon, B., Ionescu, B., and Ionescu, D. A Collaborative Cloud-Based Multimedia Sharing Platform for Social Networking Environments. 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN) : 1-6.
- [17] Qi, H. Research on mobile cloud computing: Review, trend and perspectives. 2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP) (May 2012) : 195-202.
- [18] Miao, Y., Li, B., and Gay, R. An agent based application service providing model. 8th International Conference on Control, Automation, Robotics and Vision (ICARCV 2004) 1 (December 2004) : 120- 125.
- [19] Abdel-Naby, S., Sandygulova, A., Zlydareva, O., and O'Hare, G. M.P. Investigating Agent-based Mobile Service Applications Integration with Reliable Wireless Communications. Proceedings of the IFIP Wireless Days Conference 2012 (November 2012) : 1-3.
- [20] Shen, X., and Li, F. The Application of Agent-based Information Push Technology in Mobile Learning. International Conference on Information Technology and Computer Science (ITCS) (July 2009) : 192-195.
- [21] Jadeja, Y., and Modi, K. Cloud Computing - Concepts, Architecture and Challenges. 2012 International Conference on Computing, Electronics and Electrical Technologies [ICCEET] (2012) : 877-880.

- [22] Zhang, S., Zhang, S., Chen, X., and Huo, X. Cloud Computing Research and Development Trend. 2010 Second International Conference on Future Networks (2010) : 93-97
- [23] Gong, C., Liu, J., Zhang, Q., Chen, H., and Gong, Z. The Characteristics of Cloud Computing. 2010 39th International Conference on Parallel Processing Workshop (2010) : 275-279
- [24] Talukder, A. K., and Yavagal, R.R. Mobile Computing. India : Tata McGraw-Hill, 2005.
- [25] Shenl, W., Ghenniwa, H., and Li, Y. Agent-Based Service-Oriented Computing and Applications. 2006 1st International Symposium on Pervasive Computing and Applications (2006) : 8-9.
- [26] Kim, M., Lee, S., Park, I., Kim, J., and Park, S. Agent-Oriented Software Modeling. Sixth Asia Pacific Software Engineering Conference (APSEC '99) (1999) : 318-325.
- [27] Neild, L. L., and Pargas, R. P. Investigating Peer-to-peer Systems for Resource Sharing within a Small Group of Nodes. Proceeding of the International Conference on Information Technology: Coding and Computing (ITCC'04) 2 (2004) : 400-404.
- [28] Wikipedia Foundation Inc., Peer-to-peer. Wikipedia The Free Encyclopedia. 2013. Available from : <http://en.wikipedia.org/wiki/Peer-to-peer> [2013, April 15]
- [29] Kim, H., Lee, M., Han, W., Lee, K., and Shin, I. Aciom: Application Characteristics-aware Disk and Network I/O Management on Android Platform. 2011

Proceedings of the International Conference on Embedded Software
(EMSOFT) (2011) : 49-58

- [30] Wikipedia Foundation Inc., Android Operating System. Wikipedia The Free Encyclopedia. 2013. Available from : [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system)) [2013, April 15]

- [31] Meier R., PROFESSIONAL Android™ 2 Application Development. United State of America : Wiley Publishing, Inc, 2010.

- [32] Wikipedia Foundation Inc., Fibonacci Numbers. Wikipedia The Free Encyclopedia. 2013. Available from : http://en.wikipedia.org/wiki/Fibonacci_sequence [2013, April 29]

BIOGRAPHY

Kaewkhwan Kaewmahingsa was born in Bangkok on 28th April 1989. She is graduated in Bachelor in Computer Science from Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University – Thailand, in the year of 2010.

Master in Computer Science and Information Technology, from Chulalongkorn University, Bangkok, Thailand, in 2012 under the scholarship of Center of Excellent Mathematics or CEM, Bangkok, Thailand.

The proceeding article of this thesis is: “Mobile Cloud System: A Solution for Multimedia Retrieval via Mobile Phones,” The proceeding of 2012 7th International Conference on Computing and Convergence Technology (ICCIT, ICEI and ICACT), 2012, pp. 36-40.