東 海 大 學 資 訊 工 程 學 系 研 究 所

碩士論文

指導教授：林祝興 博士

Dr. Chu-Hsing Lin

數位影片浮水印技術於雲端計算環境實現之研究

A Study of Digital Video Watermarking on Cloud Computing Environments

研究生：簡詩蓓

(Shih-Pei Chien)

中 華 民 國　一 百 零 一 年　六 月

東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 簡 詩 蓓 所提之論文

數位影片浮水印技術於雲端計算環境實現之研究

經本委員會審查，符合碩士學位論文標準。

學位考試委員會
召 集 人 ______ 簽章

委 員 ______

______

______

指 導 教 授 ______ 簽章

中華民國 101 年 6 月 26 日

ABSTRACT

It is expected that the digital era will be marked by the rapid development of the Internet and smart phones usage. Because of the need to rapidly distribute digital data files, the use of digital watermarking technologies to enhance digital copyrighted files has become extremely important as it relates to information security issues.

For the fast implementation of digital video watermarking, we proposed the use of the Hadoop distributed computing system, where the original video is split, for the different requirements to realize embedded digital watermarks. The Hadoop distributed file system together with task decomposition are discussed, and the results obtained using the software MapReduce are summarized. Both technologies are applied to the entire system and the Hadoop distributed computing features play a key role in reducing the required computing time.

Keywords: Digital information hiding、Hadoop、Grayscale images、Digital video watermark

中文摘要

在現今數位化時代當中，隨著網際網路的蓬勃發展，以及近年來智慧型手機廣泛使用，數位資訊檔案散佈的方式與速度日亦快速，因此，如何藉由利用數位浮水印技術，提升數位檔案之版權(所有權)，已成為極為重要的資訊安全議題。為了能使部屬數位浮水印的方式更為快速，已然變成了首要考慮的因素之一。在本論文中，我們所提出的方式是藉由 Hadoop 分散式運算系統，將輸入原始影片做裁剪分割，並對於分割完成後的圖片檔案，針對不同需求嵌入浮水印，以藉由利用 Hadoop 分散式運算的特性減低運算所需時間，其中，Hadoop 分散式檔案系統與(HDFS)，以及任務分解與結果匯總的技術(MapReduce)，兩者將扮演整個系統的重要關鍵。

關鍵字：數位資訊隱藏、Hadoop、灰階圖像、數位影片浮水印

致謝

本篇論文之所以能順利完成，首先要感謝我的指導老師 林祝興教授，在這兩年來的研究所學業上的悉心指導,使我在學術研究的領域中有所成長;而從 林教授身上學到的，不僅僅是學業上研究的方法，還有待人處事的態度與知識涵養的養成，這是除了學業之外最大的收穫。另外，感謝 劉榮春教授給予我在論文上的教導與啟發。再者，感謝口試委員 詹進科教授、張隆池教授以及江憲坤教授，對於論文提供的諸多改善建議。本文得以順利完成，得歸功於諸位老師的殷殷教誨，僅此向老師們致上最高的敬意與謝意，學生會永遠感激在心。

另外特別感謝李鎮宇學長對研究方向的指教與協助。在學期間也感謝資訊安全實驗室同學－信斌、泓彥、棠濰，與他們相處總是感覺輕鬆自在，也給予我在研究和課業上很多建議與幫助；學弟們－明泓、奇軒、侑廷與俊良，帶給了實驗室許多歡樂氣息，謝謝你們。也感謝遠在台北的摯友們，謝謝你們給予的包容與鼓勵，雖然大家現在各分處於不同的角落，但依然能維持我們深厚的友誼。謝謝怡伶平時的關心，讓我能平緩原本緊繃的神經。

兩年的研究生生涯使我成長許多，家人的鼓勵與支援是我最大的精神支柱，僅將此論文獻給我最敬愛的父親、母親、兄長，與所有關心我的親友們。在此，致上我最高的心意與感謝。

2012年7月 於ISLab　簡詩蓓謹上

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

The intellectual property (IP) of digital images has become an important topic in the digital age. However, because digital information can be easily copied or modified, the protection of digital data is a very important research topic in information security.

Digital watermarks can be divided into two categories: visible watermarks [1] [2] and invisible watermarks [3] [4] [5]. As the name suggests, an embedded visible watermark can be easily identified by the naked eye and its main purpose is the declaration of copyright or ownership [6] [7]. However, its disadvantage is that it destroys the appearance of the original image. In addition, advances in image processing techniques have made it easy to remove visible watermarks; therefore, visible watermarks are less suitable for the current digital environment.

On the other hand, an embedded invisible watermark cannot be easily recognized with the naked eye and a special method is required to remove it. Although this results in an increased watermark security, it is not ideal from a convenience perspective for the identification of images, where the visibility of the watermark is important.

In 2007, Huang et al. proposed unseen visible watermarking, which was a major breakthrough in the watermark field. The disadvantage of this technique is that the watermark itself must be a binary image, based on the previous restrictions of the UVW method, and therefore, its practical application is limited. In 2011, Lin et al. suggested that unseen visible gray watermarking based on the technical UVGW method is more easily applied to today's digital imaging technology. The other floating color watermark is converted to a grayscale watermark that can be easily embedded in its original, and the degree of recognition of the color watermark can be retained.

Hadoop reliability, scalability, and the characteristics of distributed computing, is also open source software. Here, we exploit these features. Hadoop distributed file systems (HDFS) exhibit low latency access to high throughput and data applications. Moreover, with MapReduce [8] [9], which is HDFS on a software development framework, distributed processing of large datasets is performed. There is no single point of computing power, and MapReduce is used to support watermark calculation.

In this study, Hadoop is used as a solution to provide customers with not only flexibility of computing capacity but also a methodology for solving high-performance computing that requires a high throughput IO. The rest of the thesis is organized as follows. Chapter 2 introduces the background of our implementation of requirements. Chapter 3 introduces the system architecture and the standard rules of the calculation for the implementation. Chapter 4 shows the experimental results, and finally, Chapter 5 presents the conclusions.

CHAPTER 2 BACKGROUND

2.1 Digital Watermarking

With the advent of the digital era, traditional media have gradually evolved into digital format storage devices and it is easy to duplicate and transmit convenient digital multimedia. Because of the use of digital multimedia data with simple storage, duplication, modification, and transmission, numerous unauthorized digital multimedia have been widely circulated on the Internet and can be downloaded free of cost by any user. To solve this problem, we can embed additional information or digital watermarks as original personal information containing the copyright for proof of ownership.

Digital watermarking technology [10] is a means of embedding digital information or digital watermarks in digital multimedia data. Different types of watermarks are applied to digital watermarking technologies, each of which have different advantages and disadvantages. It will be directly applied to various situations to prove the legitimacy of the copyright and ownership and thus the protection of intellectual property rights.

In recent years, because intellectual property rights have become an important issue of copyright protection, many researchers have proposed different digital watermarking methods to protect data, and depending on the requirements, the appropriate digital watermarking technology to achieve its purpose.

Based on human visual effects, digital watermarks can be divided into three categories: visible, invisible, and unseen visible watermarks. In this section, we will explain these watermarks in detail.

2.1.1 Visible watermarks

With visible watermarking technology [11] [12] [13] [14], the embedded watermark can be seen by the human eyes, as shown in Figure 1. It is mainly used for the declaration of copyright ownership and to prevent the illegal use of digital multimedia, but a different technology is used to prevent forgery and pirated watermarks. The main advantage of this type of watermarking technique is that it is not necessary to have any operator, although the human eye is used to identify the owner of the logo or mark. However, the main disadvantage of this watermarking technology is the destruction of the original digital multimedia data, resulting in the distortion of the original digital media.

Figure 1：Image with a visible watermark

In addition, with the recent advances in information technology, various image processing technologies have evolved owing to the numerous image processing tools, and we can easily remove visible watermarks in digital media along with other signs or marking. Therefore, visible watermarking technology is not suitable for the current digital environment. Table 1 illustrates the analysis and features of a visible watermark.

Table 1: Analysis and features of a visible watermark

Visible watermarking	
Features	1. Watermarks directly cover areas of the cover image. 2. Enhance ownership. 3. Trademarks and logos are mostly used as watermarks.
Advantages	1. Easy identification. 2. Fast processing.
Disadvantages	1. It can be easily destroyed. 2. It reduces the visual quality of the cover image.

2.1.2 Invisible watermarks

The main characteristic of invisible watermark technology is that stressed the watermark embedding of camouflage diagram shown in Figure 2. It cannot be directly recognized by the naked eye [15] [16] [17], and the direct recognition of this type of technology can be classified as information hiding technology.

However, invisible watermarking technology has more stringent requirements than visible watermarking technology, and it should be resistant to various destruction techniques such as those in fuzzy, clear, distortion compression reduction, zoom, and rotating processing environments. It should still be possible to remove clear and recognizable digital watermarking. Although there exists such technology to remove digital watermarks, we need to remove it using a special computing watermark algorithm. However, compared with the visible watermarking technology, the latter has a better ability to resist various types of destruction during image processing.

Figure 2：Image embedded with an invisible watermark

However, invisible watermarking technology has more stringent requirements than visible watermarking technology, and it should be resistant to various destruction techniques such as those in fuzzy, clear, distortion compression reduction, zoom, and rotating processing environments. It should still be possible to remove clear and recognizable digital watermarking. Although there exists such technology to remove digital watermarks, we need to remove it using a special computing watermark algorithm. However, compared with the visible watermarking technology, the latter has a better ability to resist various types of destruction during image processing.

Table 2: Analysis and features of an invisible watermark

Invisible watermarking	
Features	1. The most used watermarking method. 2. There are commercial considerations and requirements. 3. Use of the characteristics of the human visual system.
Advantage	Does not reduce the visual quality of the cover data.
Disadvantage	After destruction using image processing, the original content may be modified and hence the embedded watermark cannot be retracted correctly.

When there are ownership disputes, the hidden information within the camouflaged images or digital watermarks need to be viewed accurately and used as evidence. Table 2 lists the analysis and features of invisible watermarks.

2.1.3 Unseen visible watermarks

Unseen visible watermarking [18] was proposed by Huang et al. in 2007. Hidden watermarks can be displayed by making appropriate adjustments to the perspective, for example, setting a different elevation to view a screen allows an invisible watermark to be displayed directly in the human visual system.

Traditional unseen visible watermarking technology is used to remove its watermark via a special watermark removal algorithm program, and this is the only way to show its watermark as opposed to using the visible watermark method, which is not convenient. The former method does not affect the visual quality of the normal viewing angle or the watermark embedded in the image, and special procedures are required for its removal; hence, this technique highlights the practical value of the watermark technology.

Unseen Visible Watermark with the visibility of the watermark invisibility watermark merits of both approaches, and thus we believe that non-obvious visibility of the watermark will be many in fact applications. Table 3 illustrates the analysis and features of unseen visible watermarks.

Table 3: Analysis and features of an unseen visible watermark

Unseen visible watermarking	
Features	1. The most novel watermark embedding technology. 2. Use of the characteristics of the human visual system. 3. Has a high practical value.
Advantage	1. Does not reduce the visual quality of the cover data. 2. Without extra watermark extraction process, the hidden watermark can be revealed. 3. Has the characteristics of both visible and invisible watermarks.
Disadvantage	Presently, not very resistant to many watermark attacks.

2.2 Unseen Visible Gray Watermarking

Unseen visible gray watermarking [19] was proposed by Lin et al. in 2011. It is more easily applied to today’s digital imaging technology and is based on the technical UVGW. The other color floating watermark is easily converted to a grayscale watermark. The color watermark is embedded in its original image, and its degree of recognition can be retained.

2.2.1 Grayscale watermark embedding method

The watermark embedding process is described in the following steps:

Step 1： Perform a gamma correction on the original image I to obtain G(i), and find the largest gradient value i* of G(i).

$$G(i), 0 \le i \le 255, 0 \le G(i) \le 255 \tag{1}$$

$$i^* = \arg\max_i \nabla G(i) \tag{2}$$

Step 2：Perform a denoising operation to obtain I’j in Eq. (3), where the δj value is the denoising parameter for the maximum value of δmax.

$$I'_j = D(I_G, \delta_j), \delta_j \le \delta_{\max} \tag{3}$$

Step 3： Choose the best embedding region R.

Step 4：Again perform denoising until Eq. (6) is satisfied.

$$P = \arg\max \sum_{x=x_0}^{x_0+W_l-1} \sum_{y=y_0}^{y_0+W_w-1} \left| I'_j(x,y) - i^* \right| \tag{4}$$

$$R = \{ I'_j(x,y) \mid x_0 \le x \le x_0 + W_l - 1, \quad y_0 \le y \le y_0 + W_w - 1 \} \tag{5}$$

$$S = \sum_{x=x_0}^{x_0+W_l-1} \sum_{y=y_0}^{y_0+W_w-1} D\left(I_j^{'}(x,y) - i^{*}\right) \le W_l \times W_w \times T \quad (6)$$

In the RGB color mode, saturation has a certain proportion of R, G, and B values. One of the colors was chosen as the color index, which had the maximum value in the embedded region. If a pixel had the same R, G, and B values, the index was determined to be the color that appeared most around the pixel.

$$MAX = \max(R, G, B)$$
$$MIN = \min(R, G, B) \quad (7)$$

We performed the analysis on a grayscale watermark that is used to identify the distribution of gray watermarks, as in Eq. (8).

$$L = \left(avg \sum_{x=x_0}^{W_l-1} \sum_{y=y_0}^{W_w-1} W(x,y) \right) \times \sum_{x=x_0}^{W_l-1} \sum_{y=y_0}^{W_w-1} W(x,y) \quad (8)$$

The intensity adjustment of each pixel is according to its index in the watermark, which is embedded in the region. The dynamic adjustment prevents the adjusted color from exceeding 255. In the grayscale watermark, the pixel intensity value must be adjusted in accordance with the gray level; hence, the adjustment variable ω was determined by L. When the above was completed, the grayscale watermark was successfully embedded in the original image.

$$\tilde{I}(x,y) = \begin{cases} I_j^{'}(x,y) - \omega, & if \ \left(I_j^{'}(x,y) + \omega\right) > 255 \\ I_j^{'}(x,y) + \omega, & otherwise \end{cases}$$
$$\omega = W(x - x_0, y - y_0) \pm L \quad (9)$$

2.3 Hadoop Infrastructure

2.3.1 HDFS

HDFS [20] is a storage system that was developed by the Apache project and is used by Hadoop applications. It creates multiple replicas of data blocks, and distributes them on computing nodes through a cluster for reliable and extremely rapid computations. HDFS was created for distributed storage, and commodity hardware was designed for distributed processing. With the above characteristics, HDFS is simple to expand, fault-tolerant, and scalable.

HDFS has a master/slave architecture. An HDFS cluster includes a single namenode, a master server for managing the file system, and regulated file access by clients. Many datanodes are clustered and are attached to the nodes that run on them. HDFS creates a file system namespace in which a user can store data. A file is internally split into one or more blocks and is stored in a set of datanodes, as shown in Figure 3.

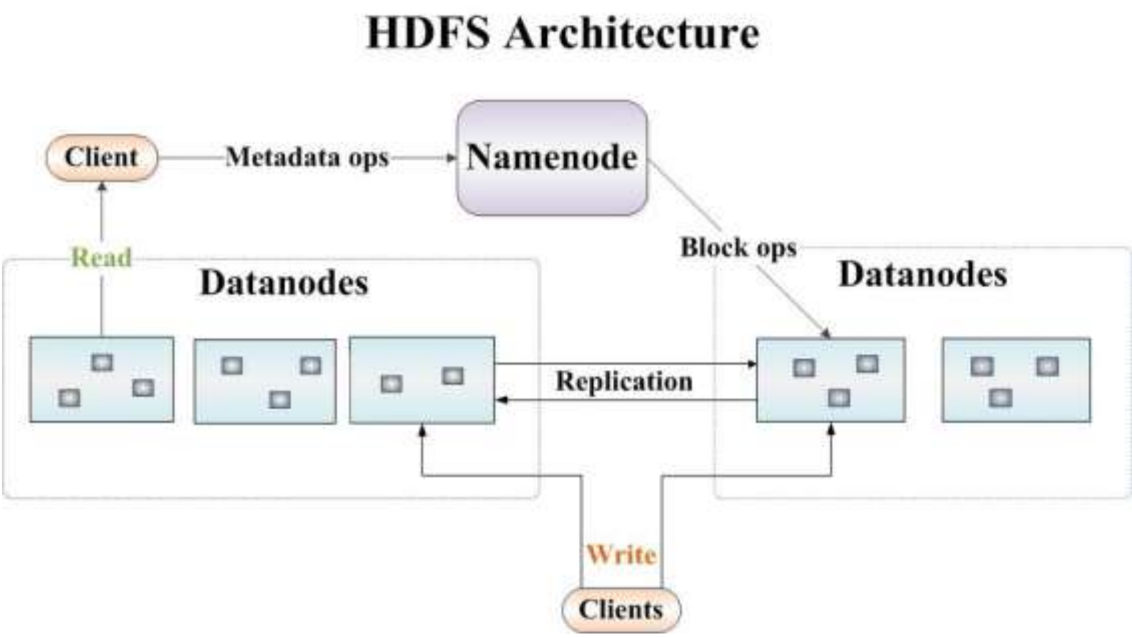

Figure 3：HDFS Architecture

Namenode manage the file system namespace, and managing file access permissions or a file is split into several small files stored in which Datanode called metadata information. Therefore, namenode does not actually store data but is the first step when users need to access files and when the namenode connection has been made in the actual location of the file. This method also prevents the namenode from becoming the reason for the performance bottleneck of HDFS.

The datanode is the data that is actually stored externally. The added datanodes perform the required instruction of the namenode, and there is also the need to meet the user in order to read and write multiple data files that are stored on multiple machines. The data read speed can therefore be sped up. For a write is because the relationship of the data a copy of, and subject to delay. A comparison between namenode and datanode is shown in Figure 4.

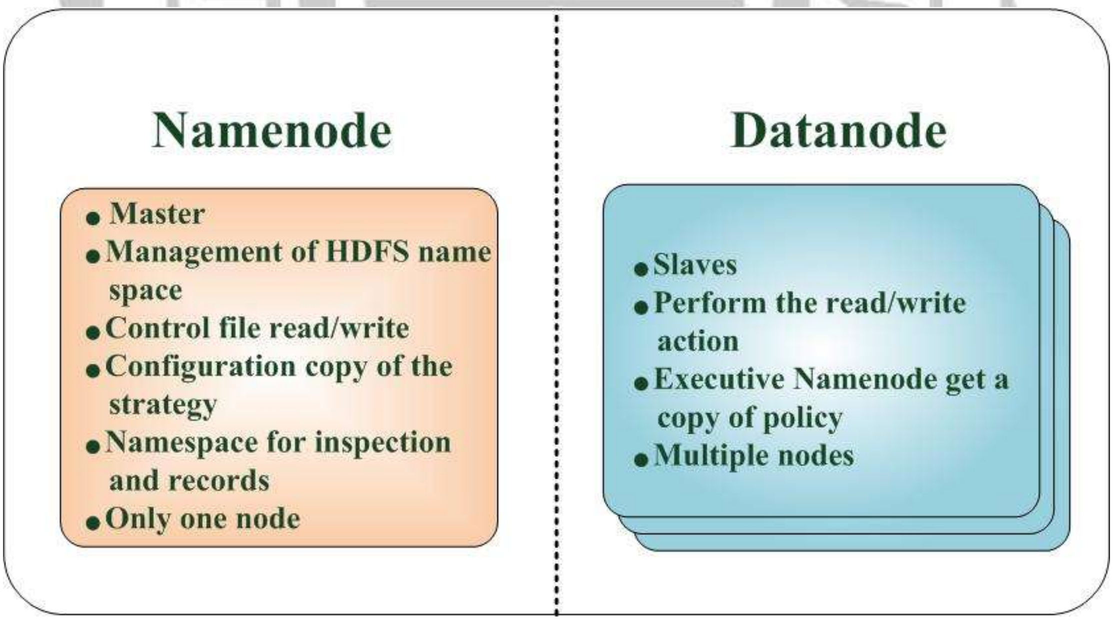

Figure 4：Namenode and datanode assignment

Namenode and datanode are designed to run on commodity machines that run on a Linux operating system. HDFS is built with the Java language; any computing node that supports Java can execute the software in the namenode or datanode. By using the Java language, HDFS can be easily deployed on a wide range of commodity machines, but it is typically deployed on a dedicated machine that runs only the namenode software.

2.3.2 MapReduce

MapReduce is an integral part of the system and is known for its simplicity, applicability, and its ability to handle a large set of distributed Hadoop applications. It is a software tool that is a framework for developing distributed computing cluster processing. MapReduce [20][21][22] has a parallel fault-tolerant file system. It provides reliable access to a large amount of data using large clusters of commodity hardware.

MapReduce jobs are split into independent blocks of data inputs under the map task parallel processing blocks. The output of the framework is for a variety of maps, and it is then input to the next step to reduce the tasks. The work input and output files will be stored in HDFS, and the framework will then perform task scheduling and monitoring tasks, and re-run the failed task. Storage-node cluster with datanode the same. This means that the MapReduce framework and HDFS are the same group on the cluster shown in Figure 5.

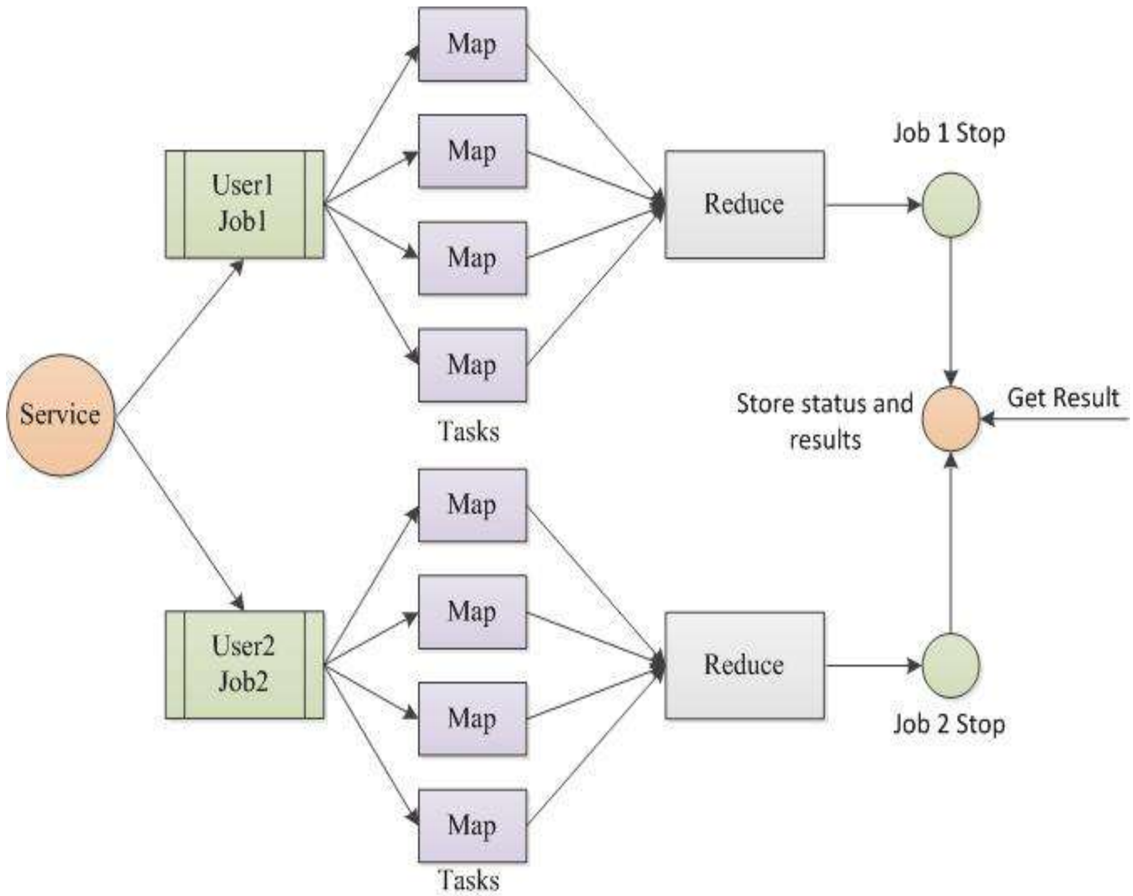

Figure 5： System components

It is effective at scheduling tasks for a configuration that allows the framework to access the nodes in which data are already present.

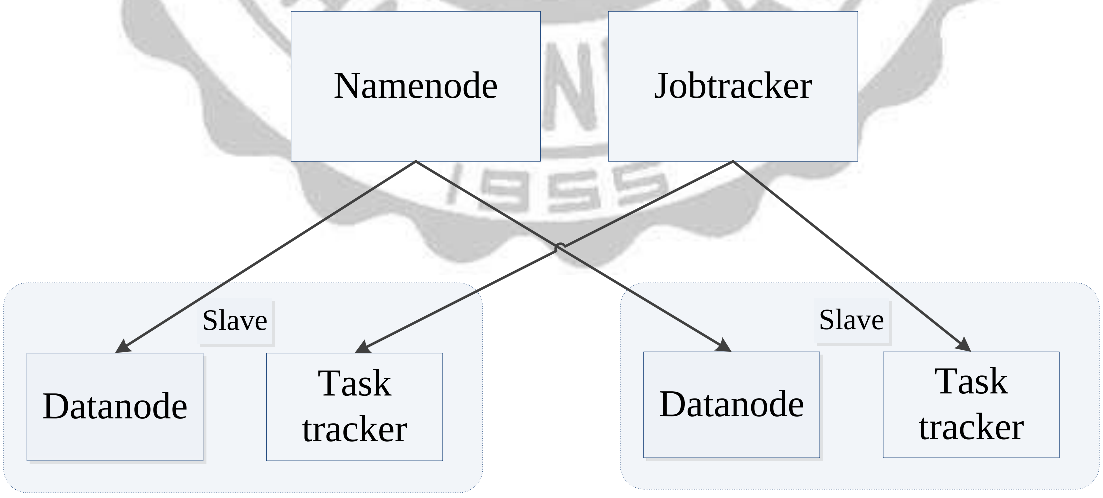

Figure 6： System components

As shown in Figure 6, the MapReduce framework consists of a single main jobtracker in each cluster node and a slave tasktracker. The main is an integral part of the task that is responsible for scheduling and monitoring the work of slaves, and if it fails the task, the process needs to be re-run. The application specifies the location of the input/output and provides a map and reduces the functions through appropriate interfaces. There are other operating parameters of the working configuration. Client work, and then submit our work with Java programming, assigned to the JobTracker and responsible for sales of software or configuration slave, and then schedule tasks, and monitor them, the JobTracker will provide the status and information about the client.

CHAPTER 3 IMPLEMENTATION

3.1 System Design

This section will introduce the system architecture and method of implementing its components. Of course, the Hadoop infrastructure plays a key role in the entire system. Because the watermarking scheme consumes a lot of computing resources, the greatest advantage of our design is that our system provides a high aggregate data bandwidth, which is scaled to many nodes in a single cluster. To attain this target, we implemented the following systems and performed the experiments described below.

3.2 System Architecture

The video portion of a general movie is composed of a large and continuous static image screen, the main movie screen in academia detection, and multi-static screen eigenvalue differences in the analysis of the film to identify points at which segmentation occurred in the video. The film is segmented or the main screen is removed to create the hierarchy of the video browsing framework.

Figure 7 shows that the navigation structure of the entire movie from the top down can be divided into the movie, scenes, clips, and screen. The static screen is a two-dimensional plane, and the film can be seen on the timeline, which is a three-dimensional cube consisting of multiple planes. Every piece of film is composed of a number of scenes, and each scene can be divided into several fragments.

The scene cut algorithm will enter the video segment of the image as shown in the following figure：

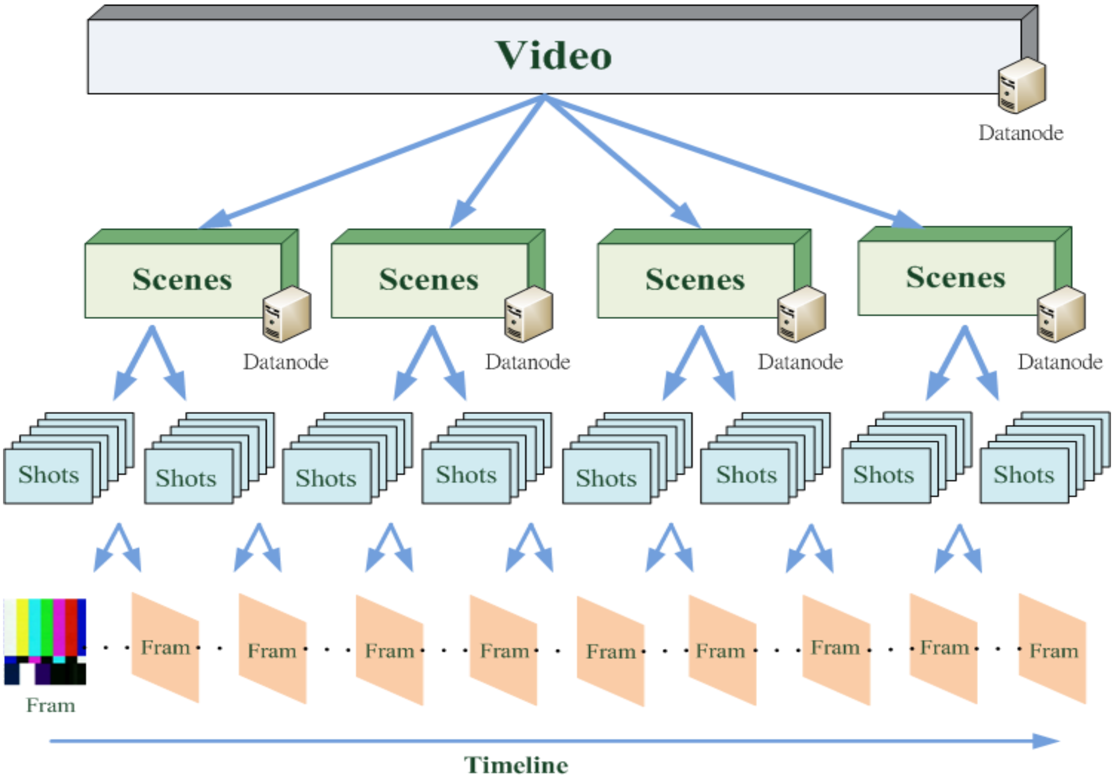

Figure 7：Video Segmentation

Using the Microsoft.NET Framework and the MediaDetClass category for users of video access, we can obtain the total length of the film by the StreamLength method at the bottom of the MediaDetClass category. This is because the film length can be split after the use of time to control a few seconds to obtain an image can also be set to obtain the size of the image, and video clips, and the end of the treatment of these pictures in the default folder.

As shown in Figure 8, the masternode includes the secondary namenode and the jobtracker guardian, which play a role in the masternode architecture that is responsible for the distributed file name, the collection results, and the management of clusters. The slave node includes the tasktracker and the datanode. In this framework, the master node contains Hadoop cluster management, coordination, and data collection. The slave node includes HDFS, data processing, and storage functions.

The entire system is setup according to the official Hadoop manual.

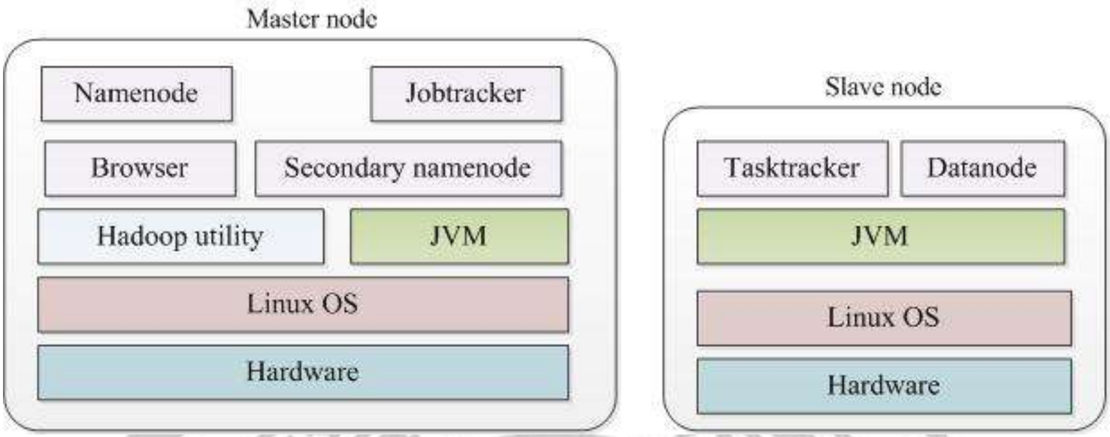

Figure 8：Node function

Our design includes two parts: HDFS and MapReduce. HDFS involves the storage of large datasets, and we need to address the data and provide a high throughput of the IO and reduce data access latency.

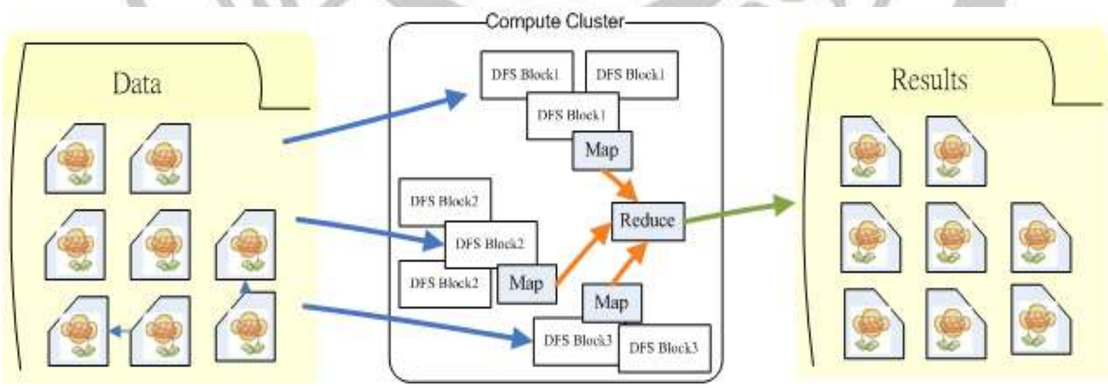

Figure 9：MapReduce flow

The components of MapReduce shown in Figure 9 are responsible for the distribution process. The system consists of a master JobTracker and one slave tasktracker for each cluster node. The main task is responsible for scheduling and monitoring the slaves. The master node instructs the slave nodes to perform tasks.

3.3 Application flow and algorithm

Our application specifies (shown in Figure 10) the input and output locations, supply map, and reduced functions. Then, the job client of Hadoop will submit the job from namenode and configure it to the jobtracker, which will distribute the software to the slaves, schedule the tasks, and monitor the statuses of the tasks. It will also provide status and related information to the job client. The job client will take the job, execute the assignment, and send feedback to namenode. The results of each job client will be sent to namenode and shown in the logs in the sub-directory named "logs" of the Hadoop directory.

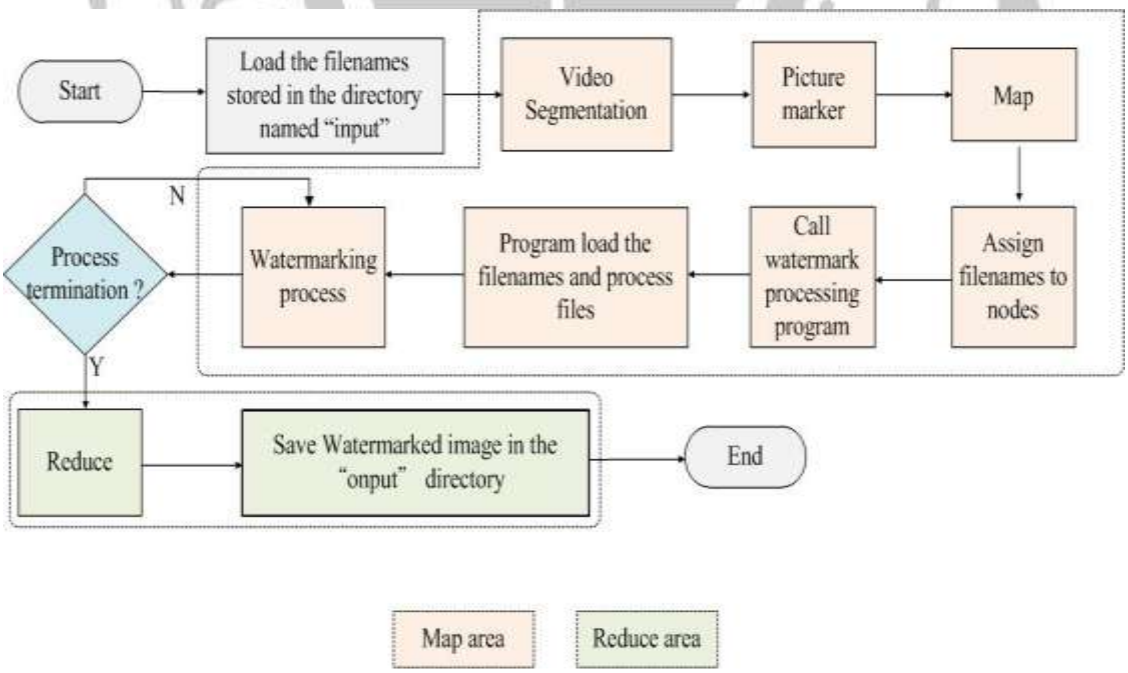

Figure 10：Application flow

3.4 Hardware and software

The hardware specifications of the computing nodes are shown in Table 4.

Table 4：Hardware and software specifications

Node	Processor	Memory	Gflops	OS	JAVA
Node 1	Intel(R) Core(TM) i5 CPU650 × 2	1G	1.22e+01	ubuntu-11.10	jre 1.6.0_24
Node 2	Intel(R) Core(TM) i5 CPU650 × 2	1G	1.23e+01	ubuntu-11.10	jre 1.6.0_24
Node 3	Intel(R) Core(TM) i5 CPU650 × 2	1G	1.22e+01	ubuntu-11.10	jre 1.6.0_24
Node 4	Intel(R) Core(TM) i5 CPU650 × 2	1G	1.27e+01	ubuntu-11.10	jre 1.6.0_24
Node 5	Intel(R) Core(TM) i5 CPU750 × 2	1G	1.24e+01	ubuntu-11.10	jre 1.6.0_24
Node 6	Intel(R) Core(TM) i5 CPU750 × 2	1G	1.23e+01	ubuntu-11.10	jre 1.6.0_24
Node 7	Intel(R) Core(TM) i5 CPU750 × 2	1G	1.23e+01	ubuntu-11.10	jre 1.6.0_24
Node 8	Intel(R) Core(TM) i5 CPU750 × 2	1G	1.23e+01	ubuntu-11.10	jre 1.6.0_24

3.5 Computing Setting

We used a 166 MB video and imported it to the “input” directory that will be processed by the watermarking job. Then, we split it into 5240 pictures, where the size of each image was 480 × 360 pixels. These images will be executed over the Hadoop architecture, in which the watermark embedding process and extracting process are executed eight times and each time a different number of datanodes is used for performance measurement.The MapReduce administration information of the nodes is shown in Figure 11.

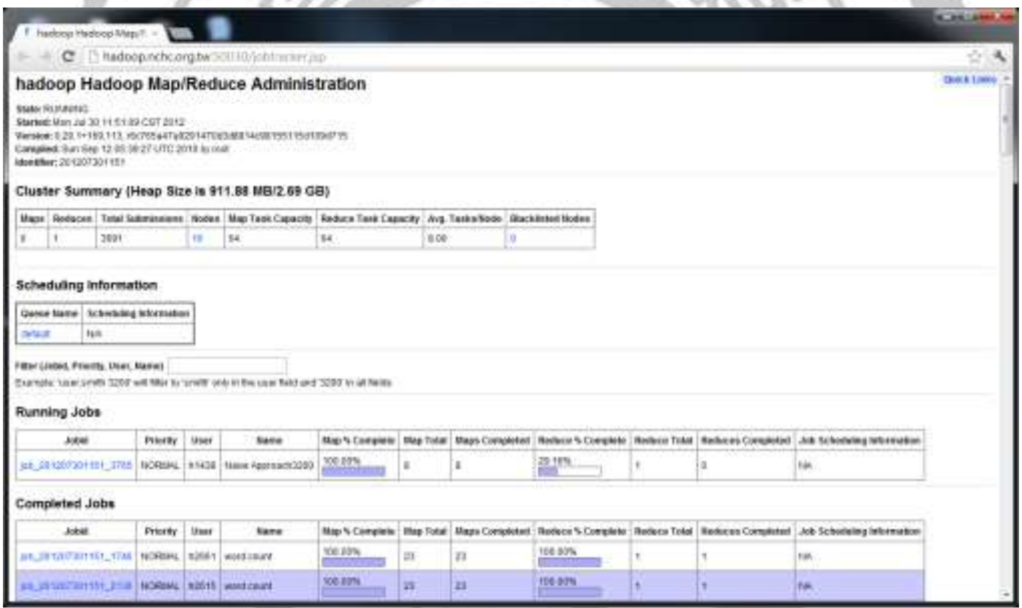

Figure 11：Node summary of the cloud environment

The node summary of the cloud environment provides the following information.

Cluster summary: The related information summarizes the number of nodes that are alive in this system, the task capacity, and the number of processes that are running on the Hadoop system. According to the document shown on the Hadoop site, the heap size uses the default value because our scale was not larger than 1000 nodes.

Scheduling information: Incoming jobs that are either waiting or being processed will be shown in this column.

Running jobs: Jobs that are still on the processing queue.

Completed jobs: Jobs that are finished and their results are being returned to the request process.

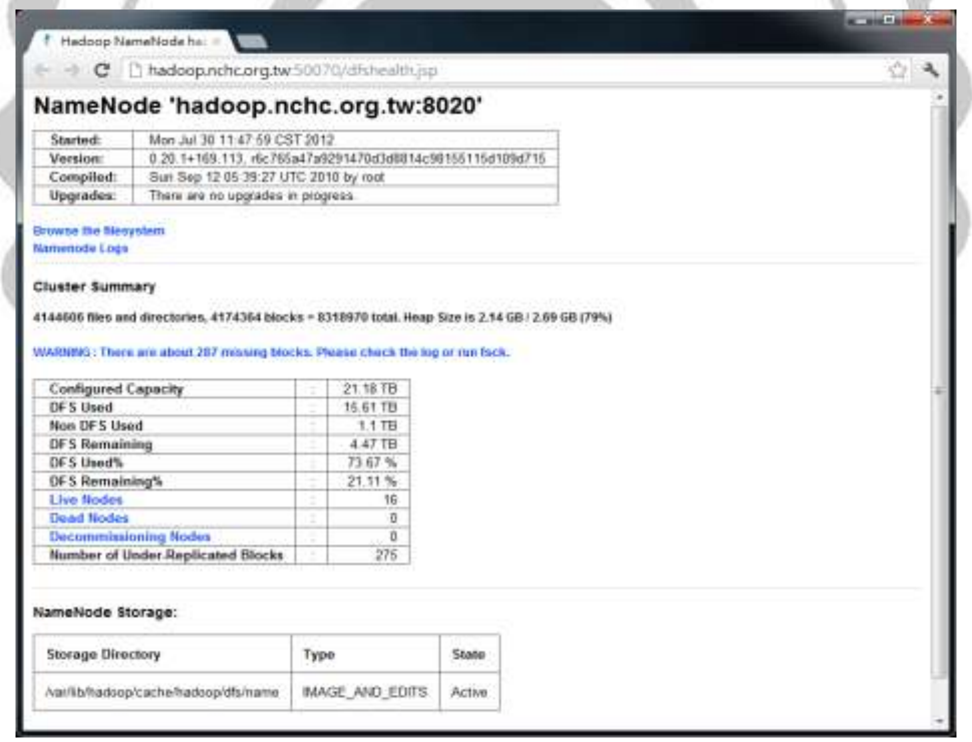

Figure 12：Cluster summary

CHAPTER 4 EXPERIMENTAL RESULTS

4.1 Calculation with Images Distributed Evenly

This section presents the experimental results. The time taken to execute the combined watermarking algorithm was recorded for each variation of the number of used datanodes. Two methods were used to measure the execution time. The first method divides the 5240 images evenly; in this way, each node will process the same number of images, and the result of the time consumed will include the wait for the slowest host to finish its job. The results are listed in Table 5.

Table 5：Execution results

Node number	**Loading per node (Image numbers)**	**Time consumed (minutes)**
1	5240	8318
2	2620	6106
3	1747	5027
4	1310	5341
5	1048	4847
6	873	4216
7	749	2736
8	655	2105

Each datanode will have a different loading, depending on the number of datanodes that are used. Table 5 shows the number of segments that each datanode must possess along with the segment size for each slave.

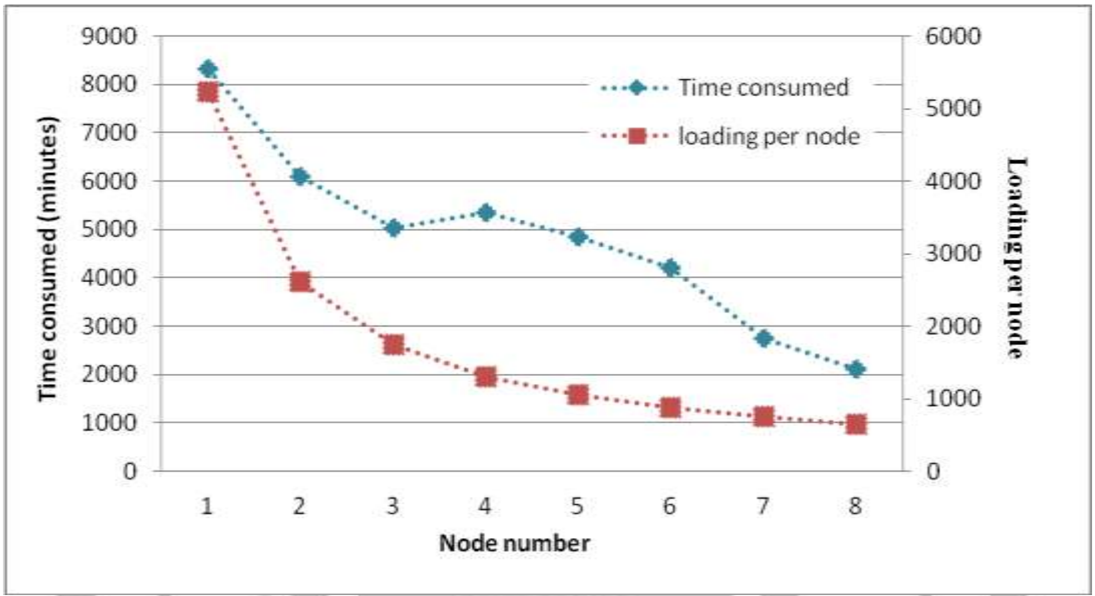

Figure 13：Execution result chart of images distributed evenly

It is apparent that all images were processed in a single node, and it took 8318 minutes to finish the watermarking job, whereas the execution time for the cluster with 8 nodes was 2105 minutes.

Figure 14 also indicates that because the result for the time consumed for each calculation process includes the wait for the slowest host to return its consumed time to the program, when the slower one is added into the computing cluster, the time consumed will increase, and the slope of the time-consumed line will become positive, as slower node 4 was added to the computing cluster.

4.2 Calculation with Optimized Computing Ability

The second method distributes the 5240 images into different numbers to each host according to the HPC challenge benchmark (HPCC) [23] and presents the results in the Gflops format (See Table 4). Gflops or gigaflops represents the measurement in billions of floating point operations per second (FLOPS) that a computer's microprocessor can handle. In this method where image numbers are distributed by the host's Gflops, we optimized and sorted the computing node from high to low for efficient distribution. The results of their computing abilities are listed in Table 4-2.

Table 6：Execution results optimized by the HPC challenge benchmark

Computing nodes	Node number	Node Loading (Image numbers)	Time consumed (minutes)
Node 1	1	5240	8318
Node 1,2	2	2620	6106
Node 1,2,5	3	1747	5027
Node (1,2,5),4	4	(1421×3),977	4627
Node (1,2,5,6), 4	5	(1121×4),756	3561
Node (1,2,5,6),(4,7)	6	(1028×4),(564×2)	2732
Node (1,2,5,6),(4,7),3	7	(962×4),(512×2),368	2163
Node (1,2,5,6),(4,7),(3,8)	8	(902x4),(427×2),(389×2)	1762

After the adjustment using Gflops values generated from the HPC challenge benchmark, the node loading was optimized by computing ability. The time spent shown in Table 6 was better than that in Table 5, in which the images were distributed evenly.

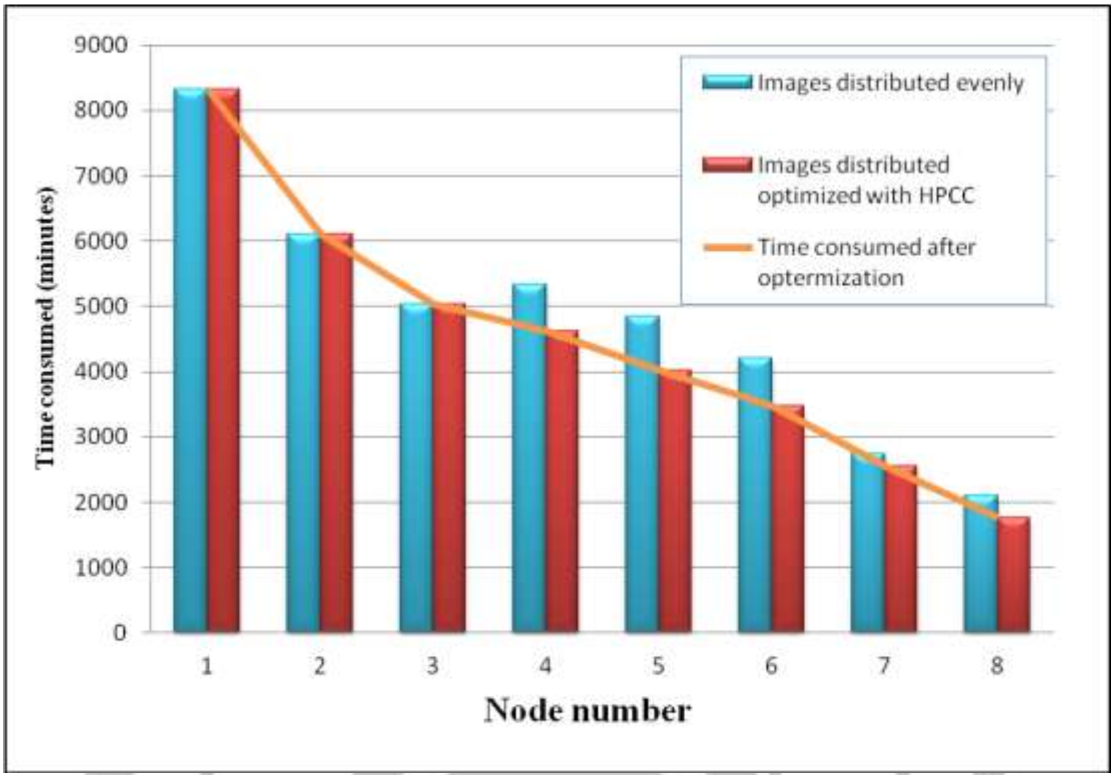

Figure 14：Comparison chart

Obviously, in the first method, computing from node1 to node 3 took the same amount of time as the second method to finish the watermarking job, whereas the best execution time for a cluster with 8 nodes was 2105 minutes. In Figure 15, when the distributed number of images was optimized by the HPC challenge benchmark, the best execution time for a cluster with 4 nodes decreased from 5341 minutes to 4627 minutes and the time taken for the cluster computing with 8 nodes decreased from 2105 to 1762 minutes. This means that the number of images distributed by the optimization of the HPC challenge benchmark will enhance the efficiency of this implementation.

This implementation shows that when processing these data with a single computer, completion of this job will require a long period of time. In the cloud environment, the impact will not be significant because all jobs are distributed to multiple computing nodes.

CHAPTER 5 CONCLUSIONS

In the implementation, we offer a better way to reduce the time consumed for such a large computing load. In addition, this method can be used daily by all users over the cloud environment similar to the other applications over the cloud environment powered by Hadoop, such as Google and Yahoo. They have provided many cloud services such as platforms, infrastructure, and systems. All these services, such as storage, calculations, and data mining in the cloud, have already deeply penetrated our daily lives.

Cloud storage has become more popular and reliable. Issues regarding copyright infringement will be accompanied by a large number of personal files, such as Facebook pictures or YouTube videos. On the related social network website, the method we proposed can be applied for all digital files, such as pictures, videos, audios, and the documents in the cloud environment.

The computation of digital watermarking technology in cloud environments will play a very important role in the protection of intellectual property in the age of digitalization.

From this implementation, we showed that the cloud computing environment offers a better solution than a single node to satisfy the requirements for high computing resources, and our proposed architecture has a wide range of uses for data encryption. The cloud environment has been implemented for the public cloud as well as for many private clouds. Most of them are powered by Hadoop, which is the same cloud environment that we used in this thesis. From the perspective of cost, performance, and data security, business always aims to achieve economies of scale to balance among lower cost, higher performance, and increased reliability. The proposed method and architecture will be helpful in enhancing data security in private companies, such as online encryption for design drawing, and may be used to design a security gateway to detect illegal attempts to upload the watermarked files on the Internet. Therefore, the proposed method will play an important role in the future.

REFERENCES

[1] Ruizhen Liu, and Tieniu Tan, “An SVD-based watermarking scheme for protecting rightful ownership,” Proceedings of IEEE Transactions on Multimedia, vol. 4, no. 1, pp. 121-128, 2002.

[2] Xianghong Tang, Lianjie Yang, Hengli Yue, and Zhongke Yin, “A watermarking algorithm based on the SVD and hadamard transform,” Proceedings of International Conference on Communications Circuits and Systems (ICCCAS 2005), Kowloon, Hong Kong, China, vol. 2, pp. 877, 2005.

[3] Xiuhui Ge, and Tian Hao, “Research on application of immune digital watermark algorithm,” Proceedings of International Conference on Computer Science and Software Engineering (CSSE 2008), Wuhan, China, vol. 3, pp. 806-809, 2008.

[4] Raymond B. Wolfgang , Christine I. Podilchuk and Edward J. Delp, “Perceptual watermarks for digital images and video,” Proceedings of IEEE Proceedings, vol. 87, no. 7, pp. 1108-1126, 1999.

[5] Liangbin Zheng, Kai Xie, Yeli Li, Huaqun Liu, and Tong Li, “A digital watermark scheme for vector graphics,” Proceedings of International Conference on Image Analysis and Signal Processing (IASP 2010), Dallas, Texas, America, pp. 699-702, 2010.

[6] Fu-Cheng You, Yang Li, and Xu-Yan Tu, “Research on mobile agent-based digital watermark tracking system and communication security,” Proceedings of International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR 2007), Beijing, China, vol. 2, pp. 835-839, 2007.

[7] Fu-Cheng You, and Fan-Zhang Li, “The research of digital copyright protecting

mechanisms base on mobile agent and digital watermark," Proceedings of International Conference on New Trends in Information and Service Science (NISS 2009), Beijing, China, pp. 466-469, 2009.

[8] Ekanayake Jaliya, Pallickara Shrideep, and Fox Geoffrey, "Mapreduce for data intensive scientific analyses," Proceedings of IEEE Fourth International Conference on eScience, pp. 277-284, Indianapolis, Indiana, Dec. 2008.

[9] Colby Ranger, Ramanan Raghuraman, Arun Penmetsa, Gary Bradski, and Christos Kozyrakis, "Evaluating mapreduce for multi-core and multi-processor systems," Proceedings of IEEE 13th International Symposium on High Performance Computer Architecture, pp. 13-24, Phoenix, Arizona, Feb. 2007.

[10] Ingemar J. Cox, Matthew L. Miller, and Jeffrey A. Bloom, "Digital watermarking," Journal of Electronic Imaging, vol. 11, no. 3, pp. 50-55, 2002.

[11] Tsung-Yuan Liu, and Wen-Hsiang Tsai, "Generic Lossless Visible watermarking — a new approach," Proceedings of IEEE Transactions on Image Processing, vol. 19, no. 5, pp. 1224-1235, May 2010.

[12] Reuben A. Farrugia, "A reversible visible watermarking scheme for compressed images," Proceedings of 15th IEEE Mediterranean Electrotechnical Conference, Valletta, Malta, pp. 212-217, April 2010.

[13] Huicheng Yang, and Ting Zhang, "A new algorithm of compound image watermarking based on DDWT," Proceedings of MultiMedia and Information Technology (MMIT'08), Three Gorges, China, pp. 268-271, Jun. 2008.

[14] Jung-Chun Liu, Chu-Hsing Lin, and Li-Ching Kuo "A robust full-band image watermarking scheme," Proceedings of 10th IEEE Singapore International Conference on Communication Systems (ICCS'06), Singapore, pp. 1-5, Oct. 2006.

[15] Fida Ullah Khattak, Abdul Bais, and Yahya M. Khawaja, "An invisible dual watermarking scheme for authentication and copyrights protection," Proceedings of International Conference on Emerging Technologies (ICET 2009), Islamabad, Pakistan, pp. 247-251, Oct. 19-20, 2009.

[16] Saraju P. Mohanty, Parthasarathy Guturu, Elias Kougianos, and Nishikanta Pati, "A novel invisible color image watermarking scheme using image adaptive watermark creation and robust insertion-extraction," Proceedings of Eighth IEEE International Symposium on Multimedia (ISM'06), San Diego, America, pp. 153-160, Dec. 2006.

[17] Soumik Das, Pradosh Bandyopadhyay, Shauvik Paul, Arindam Sinha Ray, and Monalisa Banerjee, "A new introduction towards invisible image watermarking on color image," Proceedings of IEEE International Advance Computing Conference (IACC 2009), Patiala, India, pp. 1224-1229, Mar. 6-7, 2009.

[18] Shang-Chih Chuang, Chun-Hsiang Huang, and Ja-Ling Wu, "Unseen visible watermarking," Proceedings of IEEE International Conference on Image Processing (ICIP 2007), vol. 3, San Antonio, America, pp. III-261-III-264, Sep. 16-Oct. 19, 2007.

[19] Chu-Hsing Lin, Chen-Yu Lee, Shu-Yuan Lu, and Shih-Pei Chien, "Unseen visible watermarking for gray level images based on gamma correction," Proceedings of Communications in Computer and Information Science, vol. 265, pp. 236-243, Dec 8-10, 2012.

[20] Apache Hadoop Project, http://hadoop.apache.org/

[21] Ekanayake Jaliya, Pallickara Shrideep, and Fox Geoffrey, "Mapreduce for data itensive scientific analyses," Proceedings of IEEE Fourth International Conference on eScience, pp. 277-284, Indianapolis, Indiana, Dec. 2008.

[22] Colby Ranger, Ramanan Raghuraman, Arun Penmetsa, Gary Bradski, and Christos Kozyrakis, "Evaluating mapreduce for multi-core and multi-processor systems," Proceedings of IEEE 13th International Symposium on High Performance Computer Architecture, pp. 13-24, Phoenix, Arizona, Feb. 2007.

[23] HPC Challenge Benchmark, http://icl.cs.utk.edu/hpcc/index.html