The Journal of Industrial Technology, Vol. 8, No. 3 September – December 2012

การออกแบบและพัฒนาระบบค้นหาข้อมูลจราจรทางคอมพิวเตอร์ ด้วยวิธี Map/Reduce บนกรอบการทำงานของ Hadoop

ชูพันธุ์ รัตนโภคา

บทคัดย่อ

เนื่องด้วยพระราชบัญญัติว่าด้วยการกระทำความผิดเกี่ยวกับคอมพิวเตอร์ พ.ศ. 2550 มาตรา 26 ผู้ให้บริการเครือข่ายคอมพิวเตอร์ต้องเก็บรักษาข้อมูลจราจรทางคอมพิวเตอร์ไว้ไม่น้อยกว่า 90 วันนับแต่วันที่ข้อมูลนั้นเข้าสู่ระบบคอมพิวเตอร์ ซึ่งผู้ให้บริการจะต้องเก็บรักษาข้อมูลของผู้ใช้บริการเท่าที่จำเป็นเพื่อให้สามารถระบุตัวผู้ใช้บริการ ทำให้ผู้ให้บริการเครือข่ายคอมพิวเตอร์จำเป็นต้องเก็บข้อมูลเป็นจำนวนมากและต้องใช้เวลานานในการค้นหา ดังนั้นบทความวิจัยนี้นี้นำเสนอการออกแบบและพัฒนาระบบค้นหาข้อมูลจราจรทางคอมพิวเตอร์โดยการนำ Hadoop Distributed File System (HDFS) มาประยุกต์ใช้ในการเก็บข้อมูลจราจรทางคอมพิวเตอร์ และใช้วิธี Map/Reduce ในการค้นหาข้อมูลจราจรทางคอมพิวเตอร์ โดยระบบมีส่วนติดต่อกับผู้ใช้งานผ่านทางโปรแกรมที่พัฒนาด้วยภาษาจาวา จากการทดลองระบบค้นหาข้อมูลจราจรทางคอมพิวเตอร์ด้วยวิธี Map/Reduce บนกรอบการทำงาน Hadoop บนเครื่องคอมพิวเตอร์จำนวน 10 เครื่อง พบว่าในการค้นหาข้อมูลจราจรทางคอมพิวเตอร์ที่มีขนาด 50 กิกะไบต์ จะมีความเร็วเพิ่มขึ้นประมาณ 10 เท่า เมื่อเทียบกับการใช้เครื่องคอมพิวเตอร์เพียงเครื่องเดียวในการค้นหา

คำสำคัญ : กรอบการทำงานของฮาดูป, ระบบแฟ้มข้อมูลแบบกระจาย, แนวคิดแมพ-รีดิวส์, ภาษาจาวา

ภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์, วิทยาลัยเทคโนโลยีอุตสาหกรรม, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ผู้ติดต่อ อีเมล์ : choopanr@kmutnb.ac.th รับเมื่อ 25 สิงหาคม 2554 ตอบรับเมื่อ 15 กุมภาพันธ์ 2555

The Design and Implementation of Computer Traffic Log Searcher System using Hadoop Map/Reduce Framework

Choopan Rattanapoka

Abstract

Computer Crime Act B.E. 2550's section 26 requires Internet Service Provider (ISP) to keep users' traffic log for at least 90 days since the first log-on. ISP operator must keep essential information that can identify users and time of usage. For this reason each ISP has to store a vast amount of data log which takes quite long to search for needed log data. This paper proposes the design and implementation of computer traffic log searcher system by applying Hadoop Distributed File System (HDFS) to help in storing users' log and using Map/Reduce paradigm for searching users' log. The user interface is written in Java. Upon testing the system with 10 PCs by searching a log size of 50 Gigabytes, it is found that the search result is about 10 times faster than using a single PC.

Keywords : Hadoop framework, Distributed file system, Map/Reduce paradigm, Java language

Department of Electronic Engineering Technology, College of Industrial Technology, King Mongkut's University of Technology North Bangkok.
Corresponding author, E-mail: choopanr@kmutnb.ac.th Received 25 August 2011 ; Accepted 15 February 2012

1. บทนำ

เนื่องจากพระราชบัญญัติว่าด้วยการกระทำความผิดเกี่ยวกับคอมพิวเตอร์ พ.ศ.2550 มาตรา 26 ผู้ให้บริการเครือข่ายคอมพิวเตอร์ต้องเก็บรักษาข้อมูลจราจรทางคอมพิวเตอร์ไว้ไม่น้อยกว่าเก้าสิบวันนับแต่วันที่ข้อมูลนั้นเข้าสู่ระบบคอมพิวเตอร์ และในกรณีที่จำเป็นพนักงานเจ้าหน้าที่สามารถสั่งให้ผู้ให้บริการเก็บรักษาข้อมูลจราจรทางคอมพิวเตอร์ไว้เกินเก้าสิบวัน ซึ่งผู้ให้บริการจะต้องเก็บรักษาข้อมูลของผู้ใช้บริการเท่าที่จำเป็นเพื่อให้ระบุตัวผู้ใช้บริการได้ ดังนั้นภาระจึงเกิดกับผู้ให้บริการเครือข่ายคอมพิวเตอร์ขนาดเล็กและขนาดกลาง เนื่องด้วยการเก็บข้อมูลที่มีขนาดใหญ่จำเป็นต้องใช้อุปกรณ์ประเภท Network-attached storage (NAS) หรือ Storage area network (SAN) ที่มีราคาสูง และกรณีที่ต้องการค้นหาข้อมูลของผู้ที่กระทำผิดพระราชบัญญัติว่าด้วยการกระทำความผิดเกี่ยวกับคอมพิวเตอร์ ต้องใช้เวลานานในการสืบค้นเนื่องจากข้อมูลที่เก็บไว้มีขนาดใหญ่

ปัญหาเกี่ยวกับการจัดการข้อมูลขนาดใหญ่ กำลังได้รับความสนใจเป็นอย่างมากทั้งทางภาคธุรกิจและทางด้านงานวิจัย เนื่องจากข้อมูลต่างๆ ที่จัดเก็บมีความสำคัญ เพื่อใช้ในการสืบค้น วิเคราะห์ รวมทั้งประมวลผลข้อมูล อีกทั้งข้อมูลที่จัดเก็บยังมีขนาดใหญ่ขึ้นเรื่อยๆ ทำให้การจัดการกับข้อมูลเหล่านั้นมีความลำบากและยุ่งยากมากขึ้น ดังนั้นปัญหาที่สำคัญในการจัดการกับข้อมูลขนาดใหญ่ คือ 1) การเก็บข้อมูล และ 2) การค้นหาข้อมูล เพื่อแก้ปัญหาในการจัดเก็บข้อมูลขนาดใหญ่ จึงได้มีการนำเทคโนโลยีที่เรียกว่า Hadoop[1] โดยเฉพาะส่วน ที่ชื่อว่า Hadoop Distributed File System (HDFS) [2] มาประยุกต์ใช้งานในการเก็บข้อมูลขนาดใหญ่ ซึ่ง HDFS นั้นเป็นซอฟต์แวร์ที่สามารถนำมาใช้งานได้โดยไม่เสียค่าใช้จ่ายใดๆ หลักการทำงานของ HDFS คือแบ่งข้อมูลขนาดใหญ่ที่ต้องการจะเก็บออกเป็นส่วนย่อยๆ แล้วกระจายข้อมูลส่วนย่อยๆนั้นไปยังเครื่องคอมพิวเตอร์ต่างๆที่เชื่อมต่อ สำหรับการแก้ปัญหาในส่วนของการค้นหาข้อมูลภายในเทคโนโลยี HDFS วิธีที่เหมาะสมที่สุดเรียกว่าวิธี Map/Reduce [3] ซึ่งเป็นการส่งคำสั่งค้นหากระจายไปยังเครื่องคอมพิวเตอร์ทุกเครื่องในระบบ โดยที่ไม่จำเป็นต้องมีการย้ายข้อมูลระหว่างการประมวลผล

บทความวิจัยฉบับนี้ต้องการนำเสนอการออกแบบและพัฒนาระบบค้นหาข้อมูลจราจรทางคอมพิวเตอร์โดยนำ HDFS มาประยุกต์ใช้ในการเก็บข้อมูลจราจรทางคอมพิวเตอร์ (Log) และใช้วิธีการค้นหาข้อมูลแบบ Map/Reduce เพื่อให้การค้นหาข้อมูลมีประสิทธิภาพทางด้านความเร็วในการค้นหามากยิ่งขึ้น

2. งานวิจัยที่เกี่ยวข้อง

การเก็บรักษาข้อมูลจราจรทางคอมพิวเตอร์ในปัจจุบันเป็นสิ่งที่ผู้ใช้บริการเครือข่ายคอมพิวเตอร์จำเป็นต้องทำเนื่องด้วยพระราชบัญญัติว่าด้วยการกระทำความผิดเกี่ยวกับคอมพิวเตอร์ พ.ศ.2550 มาตรา 26 ซึ่งในปัจจุบันสำหรับองค์กรขนาดเล็กและขนาดกลาง จะใช้เครื่องคอมพิวเตอร์แม่ข่ายที่ให้บริการเป็นตัวจัดเก็บข้อมูลลงในอุปกรณ์เก็บข้อมูลของเครื่องแม่ข่ายตัวนั้นเองผ่านโปรแกรม เช่น โปรแกรม syslogd ซึ่งจะทำให้เกิดปัญหา 2 อย่างคือ 1) เมื่อมีเครื่องแม่ข่ายหลายเครื่องทำให้ข้อมูลในการจัดเก็บกระจัดกระจาย ส่งผลให้ยากต่อการบริหารจัดการ 2) อุปกรณ์ในการจัดเก็บมีขนาดไม่เพียงพอในการรองรับข้อมูลที่มีขนาดใหญ่ขึ้นเรื่อยๆ

ผู้ให้บริการเครือข่ายคอมพิวเตอร์ที่มีเครื่องแม่ข่ายหลายเครื่องจึงมีการติดตั้งเครื่องแม่ข่ายหรืออุปกรณ์พิเศษ เพื่อใช้ในเก็บข้อมูลแบบรวมศูนย์ที่เรียกว่า Centralized Log Server ข้อดีของการจัดเก็บข้อมูลประเภทนี้คือ ง่ายต่อการบริหารจัดการ เนื่องจากมีจุดเก็บข้อมูลเพียงจุดเดียว แต่ข้อเสีย คือ เนื้อที่ในการเก็บข้อมูลที่อาจจะมีความจุไม่เพียงพอต่อการเก็บข้อมูลทั้งหมด และ ต้องมีการติดตั้งเครื่องแม่ข่ายเก็บข้อมูลแบบรวมศูนย์สำรองในกรณีที่เครื่องแม่ข่ายเก็บข้อมูลแบบรวมศูนย์หลักมีปัญหา ปัญหาเกี่ยวกับเนื้อที่ในการเก็บข้อมูลไม่เพียงพอสามารถแก้ไขได้ด้วยการติดตั้งอุปกรณ์เสริมประเภท NAS และ SAN แต่อุปกรณ์เสริมเหล่านี้มีราคาสูง

ผู้ให้บริการเว็บไซต์ใหญ่ๆ หลายเจ้า ก็มีปัญหาในการจัดเก็บและค้นหาข้อมูลขนาดใหญ่ เช่น Google ที่ให้บริการสืบค้นเว็บไซต์ เพื่อแก้ไขปัญหาในการจัดเก็บข้อมูล Google ได้วิจัยและนำเสนอ GoogleFS [4] ที่เป็นระบบแฟ้มข้อมูลแบบกระจายที่ Google ในพัฒนาขึ้นเอง เพื่อใช้ในการจัดเก็บ

ข้อมูลที่มีขนาดใหญ่มาก และได้นำเสนอการทำงานแบบ Map/Reduce (Google's Map/Reduce) เพื่อสืบค้นข้อมูลที่จัดเก็บอยู่ในระบบแฟ้มข้อมูล GoogleFS ซึ่งในปัจจุบันสามารถเห็นได้ชัดว่า Google สามารถให้บริการในการสืบค้นข้อมูลได้อย่างรวดเร็วและมีประสิทธิภาพ

หลังจากที่ Google พัฒนาและใช้งาน GoogleFS แต่ไม่มีการเผยแพร่โปรแกรมต้นฉบับ ทำให้มีกลุ่มโปรแกรมเมอร์พัฒนาโปรแกรมที่เรียกว่า Hadoop โดยได้แรงบันดาลใจมาจาก GoogleFS และ Googles' Map/Reduce โดย Hadoop แจกจ่ายในลักษณะของฟรีแวร์ และเป็นโอเพนซอร์ส ซึ่งใน Hadoop ประกอบไปด้วยส่วนการทำงานหลายส่วน แต่ส่วนที่สำคัญคือ HDFS ที่เป็นระบบแฟ้มข้อมูลแบบกระจายเพื่อเก็บข้อมูลขนาดใหญ่ และรองรับการจัดการข้อมูลแบบ Map/Reduce เว็บไซต์ที่จัดการกับข้อมูลขนาดใหญ่ได้นำ Hadoop ไปประยุกต์ใช้งาน เช่น Facebook และ Twitter นอกจากนี้ Hadoop ยังถูกนำไปประยุกต์ใช้ในหลายๆด้าน เช่น การเรียนรู้ของเครื่องจักร (Machine Learning) [5] และ การคำนวณทางสถิติ [6]

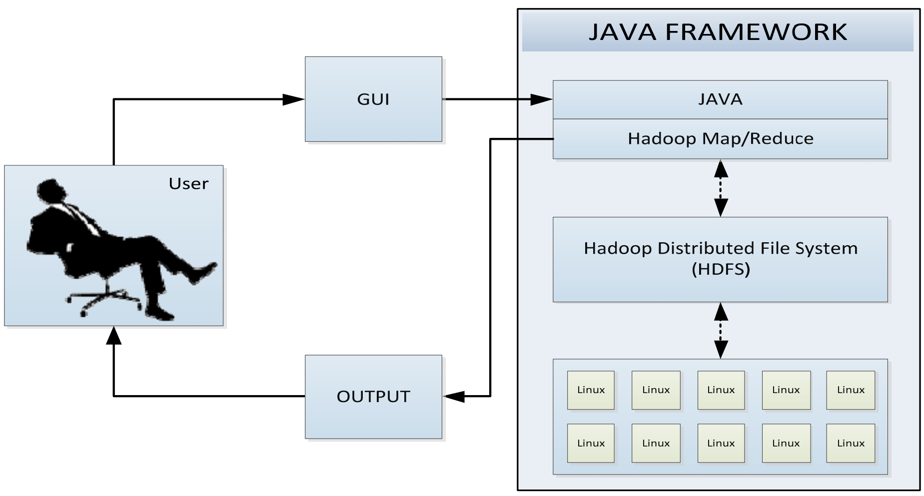
ดังนั้นจากงานวิจัยที่ผ่านมา ทำให้เกิดแนวคิดของการพัฒนาระบบที่ใช้ในการเก็บข้อมูลจราจรเครือข่ายคอมพิวเตอร์บนระบบแฟ้มข้อมูล HDFS ของ Hadoop เพื่อลดต้นทุนในการซื้ออุปกรณ์ราคาแพง และเพิ่มความเร็วในการค้นหาข้อมูลโดยการใช้วิธี Map/Reduce ในการจัดการกับข้อมูล

3. การออกแบบและพัฒนาระบบค้นหาข้อมูลจราจรทางคอมพิวเตอร์

ระบบค้นหาข้อมูลจราจรทางคอมพิวเตอร์ที่ได้ออกแบบและพัฒนาขึ้น ได้นำ HDFS มาประยุกต์ใช้ในการเก็บข้อมูลจราจรทางคอมพิวเตอร์ และได้ใช้วิธี Map/Reduce ในการค้นหาข้อมูลที่ต้องการ การทำงานโดยรวมของระบบแสดงดังรูปที่ 1 ผู้ใช้จะทำการค้นหาข้อมูลผ่านทางส่วนติดต่อกับผู้ใช้ (GUI) ที่พัฒนาขึ้นด้วยภาษาจาวา ซึ่งมีการเรียกใช้งานไลบรารีของ Hadoop Map/Reduce ที่จะไปเชื่อมต่อกับ HDFS ที่ได้ทำการติดตั้งบนระบบปฏิบัติการลีนุกซ์

3.1 รูปแบบของแฟ้มข้อมูลจราจรทางคอมพิวเตอร์

แฟ้มข้อมูลจราจรทางคอมพิวเตอร์ (Log) คือแฟ้มข้อมูลที่บันทึกรายละเอียดการติดต่อสื่อสารระหว่างคอมพิวเตอร์ผ่านระบบเครือข่าย ซึ่งแฟ้มข้อมูลจราจรทางคอมพิวเตอร์ที่ได้ใช้ในการวิจัยนี้ เป็นแฟ้มข้อมูลที่เกิดจากการทำงานของเครื่องคอมพิวเตอร์แม่ข่ายที่ทำหน้าที่เป็น NAT ของวิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ โดยภายในแฟ้มข้อมูลจราจรทางคอมพิวเตอร์ประกอบด้วยข้อมูลต่างๆ ดังรูปที่ 2



รูปที่ 1 การออกแบบระบบค้นหาข้อมูลจราจรทางคอมพิวเตอร์

```
2554-01-20 10:00:01 Local7.Debug 202.44.36.249 8975100: *Jan 20 03:00:20.209: %IPNAT-6-NAT_DELETED: Deleted tcp
10.10.63.25:1382 202.44.36.246:1382 69.63.181.66:80 69.63.181.66:80
2554-01-20 10:00:01 Local7.Debug 202.44.36.249 8975101: *Jan 20 03:00:20.209: %IPNAT-6-NAT_DELETED: Deleted udp
10.10.63.138:52925 202.44.36.246:52925 202.44.36.3:53 202.44.36.3:53
2554-01-20 10:00:01 Local7.Debug 202.44.36.249 8975102: *Jan 20 03:00:20.677: %IPNAT-6-NAT_CREATED: Created udp
10.10.42.51:64722 202.44.36.246:64722 202.14.164.9:53 202.14.164.9:53
2554-01-20 10:00:01 Local7.Debug 202.44.36.249 8975103: *Jan 20 03:00:20.721: %IPNAT-6-NAT_DELETED: Deleted tcp
10.10.63.25:1383 202.44.36.246:1383 69.63.181.66:80 69.63.181.66:80
2554-01-20 10:00:01 Local7.Debug 202.44.36.249 8975104: *Jan 20 03:00:20.721: %IPNAT-6-NAT_DELETED: Deleted tcp
10.10.63.25:2544 202.44.36.246:2544 69.63.181.66:80 69.63.181.66:80
```

2554-01-20	10:00:01	Local7.Debug	202.44.36.249	8975100:	*Jan	28	03:00:20.209:	%IPNAT-6-NAT_DELETED:	Deleted	tcp
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)

10.10.63.25:1382	202.44.36.246:1382	69.63.181.66:80	69.63.181.66:80
(12)	(13)	(14)	(15)

รูปที่ 2 รูปแบบแฟ้มข้อมูลจราจรทางคอมพิวเตอร์

โดยมีรายละเอียดต่างๆดังนี้

(1) ปี -เดือน-วัน ที่บันทึกข้อมูลจราจร

(2) เวลา ที่บันทึกข้อมูลจราจร

(3) ชื่อของบริการ Log ที่ส่งข้อมูลเข้ามาบันทึก

(4) หมายเลข IP ของเครื่องที่เก็บแฟ้มข้อมูล Log

(5) จำนวนครั้งของการเก็บ Log

(6) เดือนของเครื่องเราท์เตอร์

(7) วันที่ของเครื่องเราท์เตอร์

(8) แสดง เวลาของเครื่องเราท์เตอร์

(9) ส่วนของการทำ NAT IP และแสดงค่าสถานะของพอร์ต

(10) สถานะของ NAT IP

(11) โปรโตคอลที่ใช้งาน (TCP และ UDP)

(12) หมายเลข IP และพอร์ต ของเครื่องคอมพิวเตอร์ที่ใช้งานระบบเครือข่าย

(13) หมายเลข IP และพอร์ตที่ใช้ออกสู่ระบบเครือข่ายภายนอก ซึ่งถูกแปลงด้วยเราท์เตอร์ที่ทำหน้าที่เป็น NAT

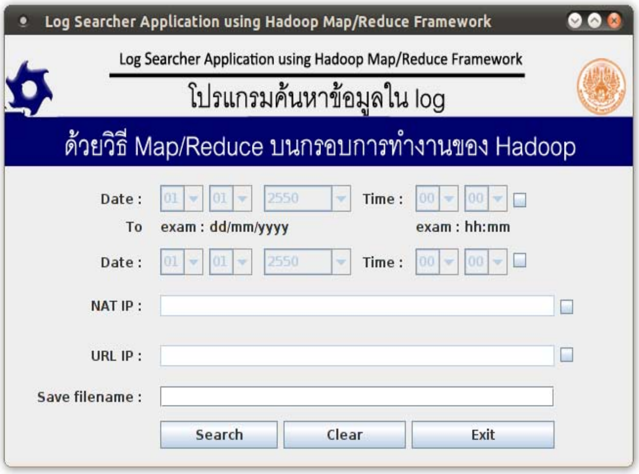
(14) และ (15) หมายเลข IP และพอร์ตของเครื่องคอมพิวเตอร์ปลายทางที่เชื่อมต่อด้วย

3.2 ส่วนติดต่อกับผู้ใช้งาน

ในส่วนติดต่อกับผู้ใช้งาน (GUI) สำหรับงานวิจัยนี้ได้พัฒนา ขึ้นเพื่อใช้ในการค้นหาข้อมูลจราจรทางคอมพิวเตอร์ โดยมีลักษณะดังรูปที่ 3 ซึ่งผู้ใช้งานสามารถเลือกใส่รายละเอียดของข้อมูลที่ต้องการค้นหาได้เอง โดยผู้ใช้สามารถใส่ข้อมูลที่ต้องการค้นหาหลักๆ คือ

- **วัน-เวลาเริ่มต้น** ถ้าใส่ข้อมูลนี้อย่างเดียว โปรแกรมจะทำการค้นหาข้อมูลที่ถูกบันทึกไว้ตั้งแต่วันที่กำหนดจนถึงวันปัจจุบัน
- **วัน-เวลาสิ้นสุด** ถ้าใส่ข้อมูลนี้อย่างเดียวโปรแกรมจะทำการค้นหาข้อมูลที่ถูกบันทึกตั้งแต่ข้อมูลแรกที่ถูกบันทึกไว้ในระบบจนถึงวันที่กำหนด แต่ถ้ามีการใส่ข้อมูลทั้ง วัน-เวลาเริ่มต้นและวัน-เวลาสิ้นสุด โปรแกรมจะค้นหาข้อมูล ตั้งแต่วัน-เวลาเริ่มต้นที่กำหนดจนถึงวัน-เวลาสิ้นสุดที่กำหนด
- **NAT IP** จะเป็นการกรองเฉพาะข้อมูลที่ถูกบันทึกที่มี ส่วนของ NAT IP (ส่วนที่ (13) ของรูปที่ 5) ตรงกับข้อมูลที่ผู้ใช้กำหนด
- **URL IP** จะเป็นการกรองเฉพาะข้อมูลที่ถูกบันทึกที่มีส่วนของหมายเลขไอพีปลายทาง (ส่วนที่ (14) ของรูปที่ 5) ตรงกับข้อมูลที่ผู้ใช้กำหนด
- **Save filename** เป็นส่วนที่ให้ผู้ใช้ระบุชื่อแฟ้มข้อมูลผลลัพธ์

ทั้งนี้ทำให้ผู้ใช้สามารถนำคำค้นหามาผสมกันได้ เช่น ค้นหาข้อมูล ตั้งแต่วันที่ 1 มกราคม 2554 ถึงวันที่ 1 มีนาคม 2554 เฉพาะข้อมูลที่ได้ NAT IP หมายเลข 202.44.36.246 และเชื่อมต่อไปยังเครื่องปลายทางที่มีหมายเลขไอพี 69.63.181.66



รูปที่ 3 ส่วนติดต่อกับผู้ใช้งานของระบบที่พัฒนา

3.3 การนำข้อมูลเข้าสู่ระบบแฟ้มข้อมูลแบบกระจาย HDFS

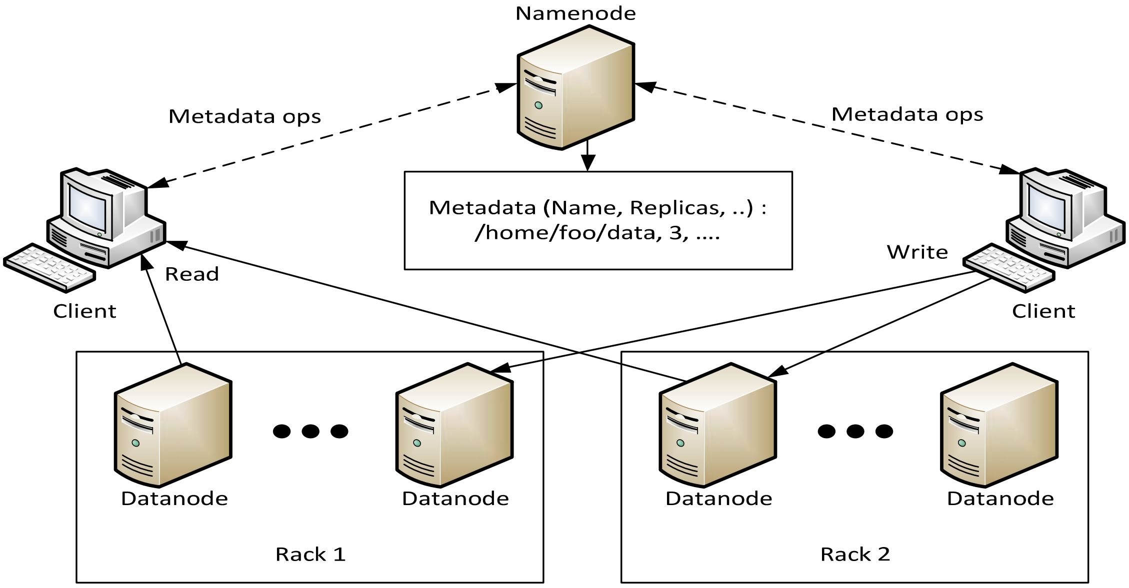
ขั้นตอนแรกของการใช้งานระบบคือ ผู้ใช้งานจำเป็นต้องนำแฟ้มข้อมูลจราจรทางเครือข่ายไปเก็บในระบบแฟ้มข้อมูล HDFS โดยการใช้คำสั่งในรูปแบบของ Command Line ที่มาพร้อมกับระบบ HDFS การทำงานของระบบ HDFS ประกอบด้วยเครื่องคอมพิวเตอร์จำนวนหลายเครื่องเชื่อมต่อการผ่านระบบเครือข่าย เมื่อผู้ใช้ป้อนแฟ้มข้อมูลที่มีขนาดใหญ่เข้าไปในระบบ ระบบจะทำการแตกแฟ้มข้อมูลที่ผู้ใช้ป้อนเข้ามา ออกเป็นชิ้นส่วนย่อยๆเรียกว่า Block (ซึ่งขนาดของ Block ผู้ติดตั้งระบบสามารถกำหนดเองได้ โดยปกติแล้วขนาดของ Block จะเท่ากับ 64 MB) แล้วส่งชิ้นส่วนย่อยๆนั้นกระจายไปเก็บยังเครื่องคอมพิวเตอร์ต่างๆ ที่เชื่อมต่ออยู่ในระบบ

โครงสร้างภายในของระบบ HDFS ประกอบด้วยเครื่องคอมพิวเตอร์ที่ทำหน้าที่ต่างกัน 2 ประเภท คือ NameNode และ DataNode โดยปกติแล้วใน HDFS จะมีเครื่องที่ทำหน้าที่เป็น NameNode อยู่ 1 เครื่อง ซึ่งจะทำงานร่วมกับเครื่องที่ทำหน้าที่เป็น DataNode จำนวนหลายๆเครื่อง เนื่องด้วย HDFS ทำงานกับเครื่องคอมพิวเตอร์หลายเครื่องในระบบ ดังนั้นเมื่อมีเครื่องคอมพิวเตอร์เครื่องใดเครื่องหนึ่งล้มเหลวขณะใช้งานอาจจะทำให้ชิ้นส่วนของแฟ้มข้อมูลที่เก็บอยู่ในเครื่องนั้นอยู่ในสถานะไม่พร้อมใช้งาน HDFS จึงแก้ไขด้วยการทำสำเนาชิ้นส่วนย่อยลงไปในเครื่องคอมพิวเตอร์เครื่องอื่นในระบบด้วย โดยจำนวนการทำสำเนานั้นผู้ติดตั้งระบบ HDFS สามารถกำหนดได้เอง ว่าต้องการจะทำสำเนาชิ้นส่วนข้อมูลย่อยเป็นจำนวนกี่ชุดในระบบ ซึ่งปกติค่าโดยปริยายจะอยู่ที่ 3 สำเนา (ผู้ใช้สามารถปรับแต่งจำนวนสำเนาได้ตามต้องการ)

Namenode เป็นเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเครื่องมาสเตอร์ (master) ซึ่งจะจัดการเกี่ยวกับ Namespace ของระบบแฟ้มข้อมูลใน HDFS เช่น เปิด ปิด เปลี่ยนชื่อแฟ้มข้อมูลและ ไดเร็กทอรี และยังทำหน้าที่ในการจดจำบล็อกที่เก็บข้อมูลว่าได้กระจายไปเก็บไว้ยัง Datanode ตัวไหน

Datanode เป็นเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเครื่องแสลฟ (slave) สามารถมีได้หลายตัว หน้าที่หลักของ Datanode ก็คือจัดการเกี่ยวกับเนื้อที่ในการเก็บข้อมูลบล็อคลงในเนื้อที่เก็บข้อมูลของเครื่อง และรับผิดชอบในการบริการการเขียนอ่านข้อมูลตามคำขอของผู้ใช้งาน รวมทั้งมีหน้าที่สร้าง ลบ และทำสำเนาบล็อกตามคำสั่งของ Namenode

รูปที่ 4 แสดงโครงสร้างและตัวอย่างการทำงานของ HDFS ซึ่งประกอบไปด้วยเครื่องคอมพิวเตอร์ที่อยู่ต่างที่กัน (Rack1 และ Rack 2) โดยแต่ละ Rack มีเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Datanode อยู่ และมี Namenode ตัวเดียวที่ใช้จัดการการทำงานทั้งหมด ซึ่ง Namenode จะทำการเก็บข้อมูล Metadata ที่เกี่ยวกับชื่อของแฟ้มข้อมูลใน HDFS และจำนวนสำเนาของแฟ้มข้อมูลนั้น เช่น /home/foo/data คือชื่อของแฟ้มข้อมูลใน HDFS และหมายเลข 3 คือจำนวนสำเนาของแฟ้มข้อมูลนั้นใน HDFS เมื่อผู้ใช้ต้องการอ่านข้อมูลจาก HDFS โปรแกรมที่ผู้ใช้พัฒนาผ่านไลบรารีของ HDFS จะทำการติดต่อกับ Namenode โดยอัตโนมัติ (Metadata Ops) เพื่อหาตำแหน่งที่อยู่ของบล็อกที่เก็บแฟ้มข้อมูลนั้นว่าได้ถูกเก็บไว้ที่ Datanode เครื่องใดบ้าง จากนั้นไลบรารี HDFS จะติดต่อไปยัง Datanode เหล่านั้นเพื่ออ่านข้อมูลของแฟ้มข้อมูลเหมือนกับการอ่านแฟ้มข้อมูลแบบปกติสำหรับการเขียนข้อมูลก็เช่นเดียวกัน ไลบรารีของ HDFS จะทำการส่งข้อความไปยัง Namenode เพื่อขอให้ Namenode ติดต่อกับ Datanode เพื่อสร้างบล็อกจากเนื้อที่ว่างของ Datanode นั้นๆ จากนั้นผู้ใช้สามารถเขียนข้อมูลลง HDFS ผ่านทาง Datanode ได้โดยตรง ส่วนการทำงานต่างๆที่ซับซ้อนเหล่านี้จะถูกซ่อนในมุมมองของผู้ใช้งาน จึงทำให้ผู้ใช้งานเขียนโปรแกรมเหมือนกับการอ่านและเขียนแฟ้มข้อมูลแบบปกติ



รูปที่ 4 โครงสร้างและการทำงานของ Hadoop Distributed File System

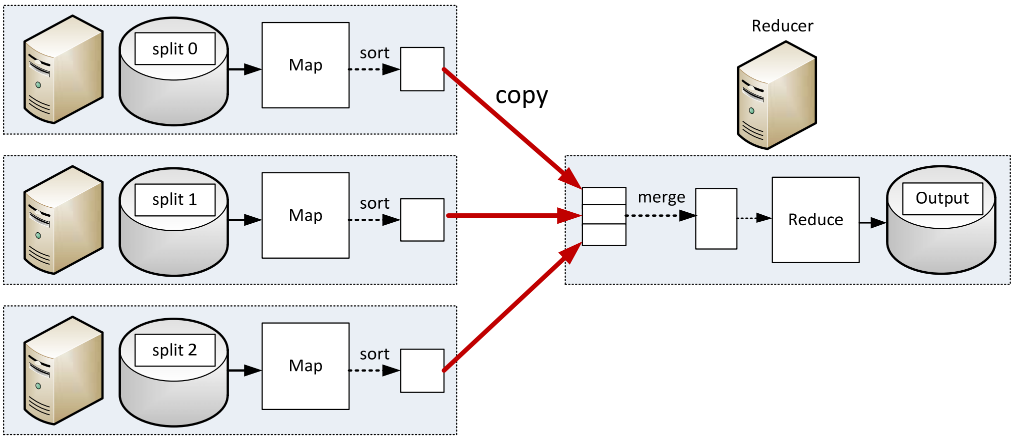
3.4 วิธีการค้นหาข้อมูลจราจรทางคอมพิวเตอร์

เมื่อแฟ้มข้อมูลการจราจรทางเครือข่ายได้ถูกกระจาย ไปเก็บไว้ยังเครื่อง Datanode ต่างๆ ในระบบแล้ว ผู้ใช้งานจึงสามารถดำเนินการค้นหาผ่านทางส่วนติดต่อกับผู้ใช้ได้ โดยงานวิจัยนี้ได้ประยุกต์ใช้เทคนิค Map/Reduce ซึ่งเป็นรูปแบบการเขียนโปรแกรมและการดำเนินงานที่เกี่ยวข้องกับการประมวลผลข้อมูลขนาดใหญ่ การใช้งาน Map/Reduce จะต้องเตรียมข้อมูลที่ต้องการจัดการก่อน โดยนำข้อมูลขนาดใหญ่นั้นแบ่งส่วนออกเป็นข้อมูลขนาดย่อย แล้วกระจายไปเก็บยังเครื่องคอมพิวเตอร์ต่างๆ ในระบบ โดยจะเห็นได้ว่าการเก็บข้อมูลลงในระบบแฟ้มข้อมูล HDFS ทำให้ข้อมูลต่างๆ อยู่ในรูปแบบที่พร้อมใช้งานสำหรับวิธี Map/Reduce จากนั้นการทำงานของ วิธี Map/Reduce เป็นการกระจายคำสั่งในการประมวลผลกับข้อมูลขนาดย่อยที่ถูกแบ่งไปเก็บไว้ยังเครื่องคอมพิวเตอร์ต่างๆ แบบขนาน ทำให้เกิดการจัดการที่มีความเป็นอิสระ และลดเวลาในการประมวลผลข้อมูลได้เป็นอย่างมาก การทำงานของวิธี Map/Reduce จะแบ่งออกเป็น 2 ขั้นตอนคือ ขั้นตอนการ Map และขั้นตอนการ Reduce

ขั้นตอนการ Map คือ การสั่งคำสั่งไปยังเครื่องคอมพิวเตอร์ต่างๆ ที่เก็บส่วนของข้อมูลที่ต้องการจัดการ (Datanode) เพื่อให้เครื่องคอมพิวเตอร์เหล่านั้นทำการดำเนินการกับข้อมูลที่เครื่องคอมพิวเตอร์นั้นๆ เก็บอยู่ สำหรับขั้นตอนการ Reduce คือ การนำข้อมูลที่เครื่องคอมพิวเตอร์ต่างๆ ได้ดำเนินการเสร็จแล้วมารวมกันเพื่อดำเนินการข้อมูลโดยรวม ซึ่งผู้พัฒนาโปรแกรมสามารถกำหนดจำนวนเครื่องที่ต้องการทำ Reduce เองได้

รูปที่ 5 แสดงตัวอย่างการทำงานของวิธีการ Map/Reduce ร่วมกับ HDFS โดยกำหนดแฟ้มข้อมูลขนาดใหญ่ที่ต้องการจัดการ ได้ถูกแบ่งออกเป็นส่วนย่อยๆ คือ split 0, split 1 และ split 2 ในระบบแฟ้มข้อมูลแบบกระจาย HDFS จากนั้นขั้นตอนการ Map จะทำการดำเนินงานกับข้อมูลของแต่ละเครื่องแบบอิสระ ผลลัพธ์ที่ได้จากการ Map จะถูกเรียงข้อมูล (sort) ก่อนจะนำมารวมกัน (merge) ที่ตัว Reducer เพื่อทำการ Reduce เป็นผลลัพธ์สุดท้าย

สำหรับการค้นหาข้อมูลจราจรทางคอมพิวเตอร์ที่พัฒนาขึ้นโดยใช้วิธีแบบ Map/Reduce นั้น เนื่องจากลักษณะการค้นหาข้อมูลจราจรทางคอมพิวเตอร์ เป็นการนำเฉพาะข้อมูลที่ตรงกับรายละเอียดที่ผู้ใช้กำหนดออกมาเป็นผลลัพธ์เท่านั้น ไม่ได้มีการนำข้อมูลที่ได้ไปทำการประมวลผลต่อ เช่น นับว่ามีการเข้าไปเว็บไซต์ไหนเป็นจำนวนกี่ครั้ง เป็นต้น ดังนั้นระบบที่พัฒนาขึ้นจึงไม่มีขั้นตอนของการ Reduce แต่อย่างไร



รูปที่ 5 หลักการทำงานของวิธี Map/Reduce

วิธีการทำงานของระบบในขั้นตอนการ Map คือ เมื่อเครื่อง Datanode แต่ละเครื่องได้รับคำสั่งในการค้นหาข้อมูล เครื่องเหล่านั้นจะทำการอ่านส่วนของข้อมูลที่มีอยู่ในเครื่องของตน โดยทำการอ่านข้อมูลขึ้นมาทีละหนึ่งบรรทัด แล้วนำข้อมูลที่อ่านขึ้นมานั้นมาแตกออกเป็นข้อมูลย่อย 15 ส่วน (ตามรูปที่ 2) และเปรียบเทียบข้อมูลในส่วนต่างๆ ว่าตรงกับที่ผู้ใช้กำหนดหรือไม่ ถ้าตรงโปรแกรมก็จะ บันทึกข้อมูลนั้นเป็นผลลัพธ์จากการดำเนินงานของเครื่อง เมื่อการประมวลผลแฟ้มข้อมูลของเครื่องนั้นๆ เสร็จสิ้น ข้อมูลผลลัพธ์ของเครื่องนั้นๆ จะถูกส่งกลับมารวมกันที่เครื่องผู้ใช้ เพื่อบันทึกเป็นแฟ้มข้อมูลผลลัพธ์รวมของการค้นหา

4. ผลการทดลอง

ในการทดลองผู้วิจัยได้ติดตั้ง HDFS บนเครื่องทั้งหมดจำนวน 11 เครื่อง โดยแบ่งออกเป็น Namenode จำนวนหนึ่งเครื่อง และ Datanode จำนวนสิบเครื่อง โดยเครื่องทั้ง 11 เครื่องเชื่อมต่อกันผ่านระบบเครือข่ายภายในความเร็ว 1 Gbps ที่เป็นระบบปิด (ไม่มีการรบกวนจากเครือข่ายภายนอก)

เครื่องคอมพิวเตอร์ที่นำมาใช้ในการทดลองมีคุณลักษณะเหมือนกันดังนี้

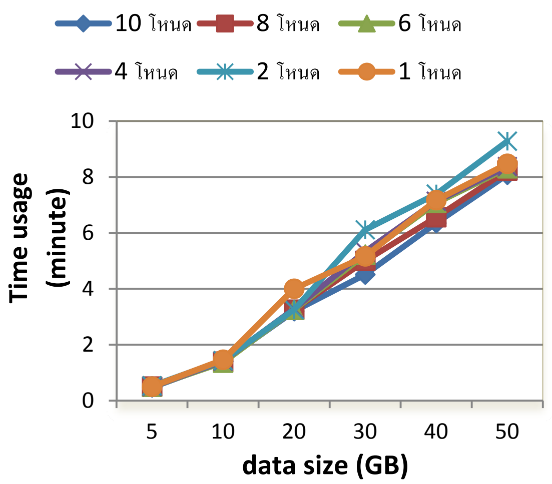
- หน่วยประมวลผลกลาง Core 2 Quad Q9500
- หน่วยความจำหลัก 4096 MB ประเภท DDR3
- ฮาร์ดดิสก์ขนาด 1 TB

การทดลองระบบได้แบ่งออกเป็น 3 ส่วน ส่วนแรกคือการวัดประสิทธิภาพทางด้านความเร็วในการนำแฟ้มข้อมูลเข้าสู่ระบบ HDFS ส่วนที่สองคือการวัดประสิทธิภาพทางด้านความเร็วในการค้นหาข้อมูลด้วยวิธีการ Map/Reduce จากข้อมูลที่เก็บไว้ในระบบ HDFS โดยการทดลองทั้ง 2 ส่วน ทำการวัดเวลาที่ใช้ในการดำเนินงานกับขนาดของข้อมูลที่แตกต่างกันคือ 5 GB, 10 GB, 20 GB, 30 GB, 40 GB และ 50 GB และมีการปรับเปลี่ยนจำนวนเครื่องที่ทำหน้าที่เป็น Datanode เป็นจำนวน 2, 4, 6, 8 และ 10 เครื่องตามลำดับ และส่วนที่สามเป็นการศึกษาผลกระทบต่อความเร็วในการค้นหาต่อจำนวน Datanode โดยกำหนดให้แต่ละเครื่องเก็บข้อมูลที่มีขนาดเท่ากัน

4.1 การทดลองความเร็วในการนำแฟ้มข้อมูลเข้าสู่ระบบ HDFS

การทดลองนี้ทำขึ้นเพื่อศึกษาผลกระทบของจำนวนเครื่องที่ทำหน้าที่เป็น Datanode ว่าส่งผลกระทบต่อการนำแฟ้มข้อมูลเข้าสู่ระบบ HDFS หรือไม่ และความเร็วในการนำแฟ้มข้อมูลเข้าสู่ระบบ HDFS มีความเร็วแค่ไหน

ผลการทดลองดังรูปที่ 6 แสดงเวลาที่ใช้ในการนำแฟ้มข้อมูลขนาดต่างๆ เข้าสู่ระบบ HDFS ที่มีจำนวน Datanode ต่างกัน จากผลการทดลองจะเห็นได้ว่าจำนวนเครื่อง Datanode นั้นไม่มีผลกระทบต่อความเร็วในการนำแฟ้มข้อมูลเข้าสู่ระบบ โดยการนำแฟ้ม ข้อมูลขนาด 5 GB, 10 GB, 20 GB, 30 GB, 40 GB และ 50 GB เข้าสู่ระบบ HDFS จะใช้เวลาประมาณ 30, 84, 200, 313, 417 และ 507 วินาทีตามลำดับ



รูปที่ 6 เวลาที่ใช้ในการนำแฟ้มข้อมูลเข้าสู่ระบบ HDFS

ความเร็วของการนำแฟ้มข้อมูลขนาด 5 GB เข้าสู่ระบบอยู่ที่ 167 MB/s ในขณะที่ความเร็ว ในการนำแฟ้มข้อมูลขนาด 50 GB เข้าสู่ระบบอยู่ที่ 98 MB/s ความเร็วในการนำแฟ้มข้อมูลตกลงเนื่องมาจากจำนวนของบล็อกที่ Namenode ต้องติดต่อให้ Datanode จัดสรรให้นั้นมีจำนวนเพิ่มมากขึ้น

4.2 การทดลองความเร็วในการค้นหาข้อมูล

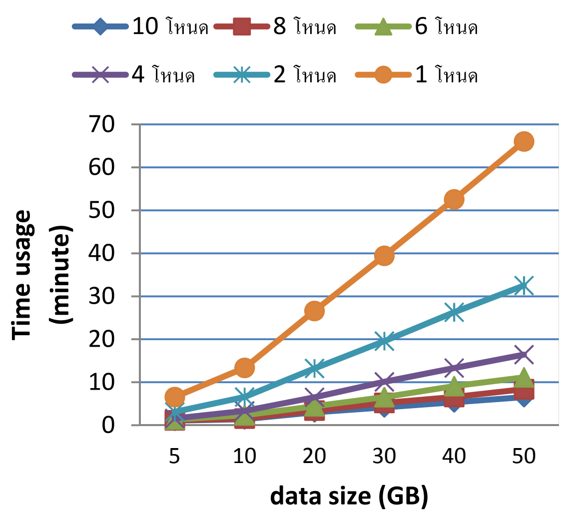
การทดลองนี้เป็นการวัดประสิทธิภาพในการค้นหาข้อมูลด้วยวิธี Map/Reduce ที่พัฒนาขึ้นบนระบบ HDFS เพื่อศึกษาผลกระทบของจำนวนเครื่องที่ทำหน้าที่เป็น Datanode ต่อความเร็วในการค้นหาข้อมูล

รูปที่ 7 แสดงเวลาที่ใช้ในการค้นหาข้อมูลต่อจำนวนเครื่อง Datanode ที่ใช้งาน ซึ่งจะเห็นได้ว่าเมื่อมีการเพิ่มจำนวน Datanode ในระบบ เวลาที่ใช้ในการค้นหาข้อมูลจะน้อยลง สำหรับข้อมูลขนาด 50 GB เวลาที่ใช้ในการค้นหาข้อมูลด้วยเครื่อง Datanode จำนวน 1, 2, 4, 6, 8, และ 10 เครื่อง คือประมาณ 66 นาที, 32 นาที, 16 นาที, 11 นาที, 8 นาที และ 6 นาที ตามลำดับ

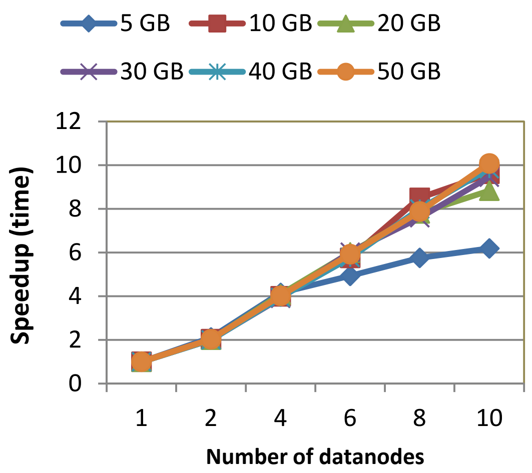
รูปที่ 8 แสดงความเร็วเป็นจำนวนเท่าในการค้นหาข้อมูลเมื่อเทียบกับเวลาที่ใช้ในการค้นหาข้อมูลด้วยเครื่องคอมพิวเตอร์เพียงเครื่องเดียว จะเห็นได้ว่าความเร็วที่เพิ่มขึ้น (Speed up) เป็นสัดส่วนแปรผันตรงกับจำนวนเครื่องที่ทำหน้าที่เป็น Datanode สำหรับข้อมูลขนาด 50 GB ความเร็วเป็นจำนวนเท่าที่เพิ่มขึ้นในการค้นหาข้อมูลเมื่อเปรียบเทียบกับการค้นหาข้อมูลด้วยเครื่องคอมพิวเตอร์เพียงเครื่องเดียว โดยการใช้เครื่อง Datanode จำนวน 2, 4, 6, 8, และ 10 เครื่อง คือประมาณ 2 เท่า, 4 เท่า, 6 เท่า, 8 เท่าและ 10 เท่า ตามลำดับ

จากการทดลองที่ได้จะเห็นได้ว่าประสิทธิภาพในการค้นหาข้อมูลด้วยวิธี Map/Reduce ในการค้นหาข้อมูลเร็วขึ้นเป็นสัดส่วนตามจำนวนเครื่อง Datanode ทั้งนี้เนื่องจากเครื่อง Datanode ที่ใช้ในการเก็บข้อมูลแต่ละเครื่อง สามารถดำเนินการกับข้อมูลที่เก็บในเครื่องของตัวเองอย่างอิสระต่อกัน และเครื่อง Datanode แต่ละเครื่องดำเนินการกับข้อมูลในลักษณะการทำงานแบบขนาน (ค้นหาข้อมูลพร้อมๆ กัน)

แต่อย่างไรก็ตามเมื่อดูความเร็วที่ใช้ในการค้นหาข้อมูลขนาด 5 GB จะเห็นได้ว่า เมื่อใช้จำนวนเครื่อง Datanode เป็นจำนวน 2, 4, 6, 8, และ 10 เครื่อง ความเร็วจำนวนเท่าที่เพิ่มขึ้นจากการค้นหาข้อมูลด้วยเครื่องคอมพิวเตอร์เครื่องเดียว คือประมาณ 2 เท่า, 4 เท่า, 5 เท่า, 5.75 เท่า และ 6.2 เท่าตามลำดับ ซึ่งสามารถสรุปได้ว่า เมื่อข้อมูลที่ต้องการดำเนินการด้วยมีขนาดไม่ใหญ่พอ ความเร็วในการค้นหาจะเริ่มถึงจุดอิ่มตัวที่การเพิ่มจำนวนเครื่อง Datanode เข้าไปไม่สามารถทำให้ความเร็วในค้นหาข้อมูลมากขึ้น และอาจจะไปลดความเร็วในการค้นหาอีกด้วย ซึ่งมีผลมาจากเวลาที่ใช้ในการส่งข้อมูลเพื่อรวมผลลัพธ์จากการค้นหาจะเริ่มมากกว่าเวลาที่ใช้ในการค้นหาข้อมูลจริง



รูปที่ 7 เวลาที่ใช้ในการค้นหาข้อมูลจากระบบ HDFS



รูปที่ 8 ความเร็วในการค้นหาข้อมูลที่เพิ่มขึ้นเป็นจำนวนเท่า

4.3 การศึกษาหาผลกระทบของจำนวนเครื่อง Datanode

การศึกษาหาผลกระทบของจำนวน Datanode ที่เพิ่มขึ้นต่อการค้นหาข้อมูลทำเพื่อตรวจสอบค่าใช้จ่ายในการดำเนินการค้นหาข้อมูล (Overhead) ต่อจำนวน Datanode ที่เพิ่มขึ้นในระบบ โดยกำหนดให้ Datanode แต่ละเครื่องเก็บข้อมูลขนาด 5 GB เท่ากัน จากการทดลองพบว่าเวลาที่ใช้ในการค้นหาข้อมูลของเครื่อง Datanode จำนวน 1, 2, 4, 8 และ 10 เครื่อง คือ 6.56 นาที, 6.57 นาที, 6.51 นาที, 6.55 นาที, 6.57 นาที และ 6.55 นาที ตามลำดับ ซึ่งจะเห็นได้ว่าจำนวนของเครื่อง Datanode ที่เพิ่มขึ้นถึง 10 เครื่องในทดลอง ไม่ได้เป็นการเพิ่มค่าใช้จ่ายในการดำเนินการค้นหาข้อมูลแต่อย่างไร

5. บทสรุป

บทความวิจัยนี้นำเสนอการออกแบบและพัฒนาระบบค้นหาข้อมูลจราจรทางคอมพิวเตอร์ ด้วยวิธี Map/Reduce บนกรอบการทำงานของ Hadoop โดยระบบที่ออกแบบและพัฒนาขึ้นนี้ช่วยทำให้การบันทึกและค้นหาข้อมูลที่มีขนาดใหญ่มีประสิทธิภาพมากขึ้น โดยส่วนของการบันทึกข้อมูลได้ประยุกต์ใช้ HDFS ซึ่งทำหน้าที่ในการเก็บข้อมูลแบบกระจาย อีกทั้งยังสามารถรองรับการขยายตัวของจำนวนเครื่องในระบบสำหรับส่วนของค้นหาข้อมูลได้ประยุกต์ใช้วิธีการค้นหาแบบ Map/Reduce เพื่อให้การค้นหาข้อมูลที่ถูกจัดเก็บใน HDFS มีความเร็วในการค้นหามากขึ้น

ผลการทดลองชี้ให้เห็นว่าเมื่อติดตั้งระบบลงบนเครื่องคอมพิวเตอร์จำนวน 10 เครื่อง และทำการค้นหาข้อมูลจราจรทางคอมพิวเตอร์ที่มีขนาด 50 กิกะไบต์ พบว่าเวลาที่ใช้ในการค้นหาข้อมูลจะมีความเร็วเพิ่มขึ้นประมาณ 10 เท่า เมื่อเทียบกับการใช้เครื่องคอมพิวเตอร์เพียงเครื่องเดียว

6. กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณ นายคงวิศิษฐ์ จันทรภิบาล นายกฤษณลักษณ์ เครือยศ และนายอนุชา ย่อมกระโทก นักศึกษาภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ที่ช่วยพัฒนาระบบค้นหาข้อมูลจราจรทางคอมพิวเตอร์ ตลอดจนวิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือที่ให้ใช้สถานที่ในการทำวิจัย

7. เอกสารอ้างอิง

1. Apache Hadoop, Available: http://hadoop.apache.org, 20 August 2011.
2. D. Borthakur, “The Hadoop Distributed File System: Architecture and Design”, The Apache Software Foundation.
3. J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters”. Commun. ACM 51, 1 January 2008, pp. 107-113.
4. S. Ghemawat, H. Gobioff, and S.T. Leung, “The Google File System”. Proceedings of the nineteenth ACM symposium on Operating systems principles (SOSP '03). ACM, New York, USA, pp. 29-43.
5. Z. Weizhong, M. Huifang, and H. Qing, “Parallel K-Means Clustering Based on MapReduce”, Cloud Computing, Lecture Notes in Computer Science, 2009, pp. 674-679.
6. Z.D. Zhao, M.S. Shang, “User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop”, International Workshop on Knowledge Discovery and Data Mining, 2010, pp. 478-481.