P2P-DFS: ระบบแฟ้มข้อมูลแบบกระจายบนเครือข่ายเพียร์ทูเพียร์
P2P-DFS: Distributed File System on Peer-to-Peer Network

ชูพันธุ์ รัตนโภคา[1] อธิวัตน์ หงษ์ทอง[2] รุ่งโรจน์ เบญจานุกร[3] และ ภูวรุฬห์ พันธ์เครือบุตร[4]

ภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์
วิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
choopanr@kmutnb.ac.th[1], theking_k59[2], tictace[3], puwarun[4] @hotmail.com

บทคัดย่อ

ปัจจุบันมีการใช้งานคอมพิวเตอร์กันอย่างแพร่หลายทั้งในองค์กรและสถานศึกษาแต่อย่างไรก็ตามเครื่องคอมพิวเตอร์เหล่านี้ไม่ได้ถูกใช้งานกันอย่างเต็มที่โดยเฉพาะอย่างยิ่งเนื้อที่ในการเก็บข้อมูล ในทางกลับกันเนื้อที่สำหรับเก็บข้อมูลในเครื่องแม่ข่ายเพื่อทำการสำรองข้อมูลหรือให้บริการเกี่ยวกับแฟ้มข้อมูลนั้นกลับไม่เป็นที่เพียงพอ บทความนี้ต้องการนำเสนอระบบแฟ้มข้อมูลแบบกระจายที่พัฒนาขึ้นโดยภาษาจาวา โดยผู้ใช้สามารถส่งแฟ้มข้อมูลเข้าไปในระบบเพื่อสำรองข้อมูลและดึงแฟ้มข้อมูลออกจากระบบจากที่ใดก็ได้ โดยแฟ้มข้อมูลที่ต้องการจัดเก็บในระบบจะถูกแบ่งออกเป็นส่วนเล็กๆและกระจายไปเก็บตามเครื่องต่างๆในระบบพร้อมทั้งมีการทำสำเนาข้อมูลตามความต้องการของผู้ใช้ ระบบแฟ้มข้อมูลแบบกระจายนี้ทำงานบนเครือข่ายเพียร์ทูเพียร์เพื่อหลีกเลี่ยงปัญหาคอขวดที่อาจเกิดขึ้นกับเครื่องในการรับส่งข้อมูล จากการทดลองระบบแฟ้มข้อมูลแบบกระจายบนเครือข่ายเพียร์ทูเพียร์พบว่าระบบสามารถทำงานได้อย่างมีประสิทธิภาพและสามารถทนทานต่อปัญหาการล้มเหลวของระบบได้เป็นอย่างดี

คำสำคัญ: ระบบแฟ้มข้อมูลแบบกระจาย เครือข่ายเพียร์ทูเพียร์ การสำรองข้อมูล

Abstract

Recently, Computers are widely used in organizations and campus. However, computer usage is under-utilized, especially on storage. In the opposite way, the storage in computer server for backing up data or serving files is not enough. This paper presents a simple distributed file system implemented by using Java language. Users can upload or download files anywhere. Uploaded files are divided into small pieces and distributed among machines in network. In addition, the system provides file replication by user demand. The infrastructure of this system uses peer-to-peer network for avoiding the network congestion that can occur on server. From the experiment, we found that our peer-to-peer distributed file system is quite efficient and well tolerate on node failure.

Keyword: Distributed file system, Peer-to-peer network, Data backup

1. บทนำ

การเก็บข้อมูลในปัจจุบัน ส่วนใหญ่จะเป็นการเก็บข้อมูลลงในฮาร์ดดิส ซึ่งเมื่อฮาร์ดดิสมีปัญหาข้อมูลสำคัญที่ถูกเก็บเอาไว้อาจจะสูญหาย ซีดีหรือดีวีดีก็เป็นอีกทางเลือกหนึ่งของการสำรองข้อมูล แต่อย่างไรก็ตามข้อมูลที่ต้องการสำรองนั้นบางครั้งมีขนาดใหญ่ทำให้พื้นที่ในการจัดเก็บนั้นไม่เพียงพอต่อความต้องการ ปัจจุบันระบบเครือข่ายมีความเร็วเพิ่มขึ้นทำให้สามารถนำมาใช้สำหรับการสำรองแฟ้มข้อมูลได้ วิธีที่มีใช้กันอย่างแพร่หลายคือการนำแฟ้มข้อมูลไปจัดเก็บในเครื่องคอมพิวเตอร์แม่ข่ายเช่น เครื่องแม่ข่าย FTP จากนั้นเมื่อผู้ใช้ต้องการแฟ้มข้อมูลดังกล่าวจะทำการดึงแฟ้มข้อมูลจากเครื่องแม่ข่ายออกมาใช้งาน แต่อย่างไรก็ตามการทำงานในลักษณะนี้จะสร้างภาระให้แก่เครื่องแม่ข่ายเป็นอย่างมาก ถ้ามีผู้ใช้บริการเป็นจำนวนมากหรือแฟ้มข้อมูลมีขนาดใหญ่ อีกทั้งเนื้อที่ในการเก็บข้อมูลยังอาจไม่เพียงพอต่อการใช้งาน ในทางกลับกันพื้นที่ในการเก็บข้อมูลของเครื่องคอมพิวเตอร์ส่วนบุคคลส่วนใหญ่ไม่ได้ถูกใช้งานอย่างเต็มที่ ซึ่งน่าจะนำมาใช้ประโยชน์ในการสำรองข้อมูล

ปัจจุบันเครือข่ายเพียร์ทูเพียร์เข้ามามีบทบาทสำคัญมากขึ้น โดยเฉพาะอย่างยิ่งทางด้านแบ่งปันแฟ้มข้อมูลระหว่างผู้ใช้ ตั้งแต่โปรแกรม Napster[1] จนกระทั่งโปรแกรม Bittorrent[2] เนื่องจากระบบเพียร์ทูเพียร์มีความทนทานต่อการล้มเหลวของระบบสูง เครือข่ายเพียร์ทูเพียร์ส่วนใหญ่จะถูกใช้งานในลักษณะการแบ่งปันทรัพยากรกันระหว่างผู้ใช้ ซึ่งทรัพยากรดังกล่าวอาจหมายถึงแฟ้มข้อมูล ช่องสัญญาณระบบเครือข่าย การทำงานของหน่วยประมวลผลกลาง หรือแม้กระทั่งเนื้อที่ในการจัดเก็บข้อมูล

ดังนั้นงานวิจัยนี้จึงประยุกต์นำระบบเพียร์ทูเพียร์เข้ามาใช้ในการแบ่งปันพื้นที่ในการเก็บข้อมูลของเครื่องคอมพิวเตอร์ส่วนบุคคลที่ไม่ได้ถูกใช้งานอย่างเต็มที่ในการสำรองข้อมูล โดยตัวระบบจะเน้นไปที่การเพิ่มประสิทธิภาพในรับและส่งแฟ้มข้อมูลในการสำรองข้อมูลของผู้ใช้ โดยข้อมูลจะถูกแบ่งเป็นส่วนย่อยๆและถูกจัดเก็บกระจายไปยังเครื่องต่างๆที่เข้าร่วมในเครือข่ายเพียร์ทูเพียร์ ซึ่งจะสามารถขจัดปัญหาคอขวดที่เกิดกับเครื่องแม่ข่ายที่ใช้ในการสำรองข้อมูล

2. วรรณกรรมที่เกี่ยวข้อง

การใช้งานระบบแฟ้มข้อมูลแบบกระจายในปัจจุบันเริ่มได้รับการนิยมอย่างแพร่หลายโดยเฉพาะอย่างยิ่ง Hadoop[3] ซึ่งเป็นซอฟต์แวร์เปิด (open source) และไม่เสียค่าใช้จ่าย ที่ได้แนวคิดมาจาก Google File System[4] โดย Hadoop จะเหมาะสำหรับใช้ในการเก็บข้อมูลที่มีขนาดใหญ่ในหน่วยของกิกะไบต์ หรือเทราไบต์ ซึ่ง Hadoop จะแบ่งแฟ้มข้อมูลที่ต้องการจัดเก็บออกเป็นส่วนย่อยๆ แล้วกระจายไปเก็บตามเครื่องคอมพิวเตอร์ต่างๆที่อยู่ในระบบ อีกทั้ง Hadoop ยังรองรับการทำงานของโปรแกรมที่ใช้ในการจัดการกับข้อมูลขนาดใหญ่ที่เรียกว่า Map/Reduce[5] ทำให้ในปัจจุบันไม่ว่าจะเป็น facebook[6], twitter[7] หรือ yahoo[8] ได้นำ Hadoop มาใช้งานเพื่อเก็บข้อมูล แต่อย่างไรก็ตาม Hadoop มีลักษณะการทำงานแบบไคลเอ็นต์-มาสเตอร์ ทำให้ระบบไม่สามารถให้บริการได้ ถ้ามีความล้มเหลวเกิดขึ้นกับเครื่องที่ทำหน้าที่เป็นมาสเตอร์ อีกทั้ง Hadoop เหมาะสำหรับแฟ้มข้อมูลที่มีขนาดใหญ่ ถ้าใช้ Hadoop เพื่อเก็บแฟ้มข้อมูลที่มีขนาดเล็กประสิทธิภาพโดยรวมของระบบจะลดลง

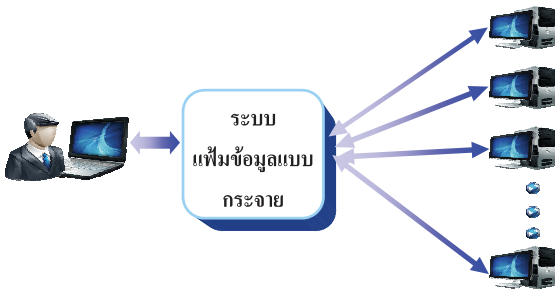
อีกทางด้านหนึ่งในปัจจุบันได้มีโปรแกรมประเภทแบ่งปันแฟ้มข้อมูลบนระบบเครือข่ายเพียร์ทูเพียร์ที่ได้รับความนิยมมากคือ Bittorrent โดยการใช้งานหลักๆ นั้นผู้ใช้จำเป็นต้องมีแฟ้มข้อมูลพิเศษนามสกุล torrent ซึ่งในแฟ้มข้อมูลพิเศษนี้จะเก็บรายละเอียดเกี่ยวกับแฟ้มข้อมูลที่ต้องการรับส่ง และช่องทางการเชื่อมต่อไปยัง tracker ที่ทำหน้าที่เก็บหมายเลขไอพีและพอร์ทของเครื่องคอมพิวเตอร์อื่นๆที่กำลังรับส่งแฟ้มข้อมูลนั้นๆ อยู่ เมื่อผู้ใช้ต้องการรับส่งแฟ้มข้อมูลนั้น โปรแกรมไคลเอ็นต์ bittorrent จะเชื่อมต่อไปยัง tracker เพื่อขอหมายเลขไอพีและพอร์ทของเครื่องคอมพิวเตอร์อื่นๆที่กำลังรับส่งแฟ้มข้อมูลนั้น จากนั้นเครื่องของผู้ใช้จะติดต่อไปยังเครื่องคอมพิวเตอร์อื่นๆตามหมายเลขไอพีและพอร์ทที่ได้รับมาจาก tracker เพื่อทำการดึงข้อมูล จะเห็นได้ว่า bittorrent มีการกระจายภาระของเครื่องคอมพิวเตอร์ในระบบได้เป็นอย่างดี และตัวระบบสามารถทนทานต่อความล้มเหลวได้

ดังนั้นจะเห็นได้ว่าถ้าสามารถนำข้อดีของ Bittorrent มาประยุกต์กับการทำงานของระบบแฟ้มข้อมูลแบบกระจาย

จะทำให้ได้ระบบแฟ้มข้อมูลแบบกระจายที่มีประสิทธิภาพมากขึ้น บทความวิจัยนี้จึงต้องการนำเสนอการพัฒนาระบบแฟ้มข้อมูลแบบกระจายบนเครือข่ายเพียร์ทูเพียร์ ซึ่งพัฒนาขึ้นด้วยภาษาจาวา และได้ใช้ ULAP[9] เป็นโครงสร้างของเครือข่ายเพียร์ทูเพียร์พื้นฐาน ซึ่ง ULAP จะมีลักษณะเป็นระบบเพียร์ทูเพียร์แบบไร้โครงสร้างที่มีความสามารถในการเรียนรู้ตำแหน่งของเพียร์จากความเร็วในการเชื่อมต่อระหว่างเพียร์ การค้นหาเพียร์ใน ULAP จะใช้การฟลัดเหมือนที่ใช้ใน Gnutella[10]

3. การทำงานของ P2P-DFS

ระบบแฟ้มข้อมูลแบบกระจายที่พัฒนาขึ้นมานั้นมีชื่อว่า P2P-DFS ซึ่งเครื่องคอมพิวเตอร์ที่ติดตั้งและเรียกใช้งานโปรแกรม P2P-DFS จะทำการเชื่อมต่อเข้ากับระบบเครือข่ายเพียร์ทูเพียร์ โดย P2P-DFS ได้ใช้ ULAP มาทำงานในส่วนของโครงสร้างระบบเครือข่ายเพียร์ทูเพียร์พื้นฐาน และเครื่องคอมพิวเตอร์ที่เชื่อมต่อกันภายในระบบจะมีความสามารถในการแบ่งปันเนื้อที่ในการเก็บข้อมูลภายในเครื่องให้กับเครื่องคอมพิวเตอร์อื่นๆที่อยู่ในระบบ ซึ่งผู้ใช้สามารถสั่งงาน P2P-DFS ผ่านอินเทอร์เฟสที่โปรแกรมให้บริการไม่ว่าจะอยู่ในรูปแบบของคอนโซลหรือกราฟิก ดังภาพที่ 1



ภาพที่ 1 แผนผังการทำงานของระบบแฟ้มข้อมูลแบบกระจาย

การทำงานที่ P2P-DFS รองรับในปัจจุบันสามารถแบ่งออกเป็นสองส่วนใหญ่ๆ คือ การส่งแฟ้มข้อมูลเข้าระบบ และการดึงแฟ้มข้อมูลออกจากระบบ โดยแฟ้มข้อมูลที่ต้องการจัดเก็บใน P2P-DFS จะถูกแบ่งออกเป็นชิ้นส่วนเล็กๆ ตามจำนวนที่ผู้ใช้กำหนดผ่านทางอินเทอร์เฟสของโปรแกรม จากนั้นโปรแกรมจะกระจายชิ้นส่วนของแฟ้มข้อมูลไปตามเครื่องคอมพิวเตอร์อื่นๆที่อยู่ในระบบเพื่อจัดเก็บ รวมทั้งผู้ใช้สามารถกำหนดจำนวนสำเนาของแฟ้มข้อมูลที่จะจัดเก็บได้ เพื่อเพิ่มความเชื่อมั่นให้กับผู้ใช้ว่าถึงแม้จะมีเครื่องคอมพิวเตอร์ที่จัดเก็บชิ้นส่วนของแฟ้มข้อมูลบางเครื่องไม่สามารถให้บริการได้ แต่ก็ยังมีเครื่องอื่นที่เก็บสำเนาของชิ้นส่วนนั้นสามารถให้บริการได้

P2P-DFS มีการเพิ่มความสามารถในการทนทานต่อความผิดพลาดของระบบได้ กล่าวคือถ้าในขณะที่กำลังส่งข้อมูลเข้าไปในระบบนั้นเครื่องที่ถูกใช้ให้เก็บชิ้นส่วนของข้อมูลล้มเหลว P2P-DFS จะทำการหาเครื่องใหม่แทนทันทีเพื่อส่งชิ้นส่วนนั้นไปเก็บ หรือแม้กระทั่งในขณะที่กำลังดึงแฟ้มข้อมูลออกจากระบบ ถ้าเครื่องที่ใช้เก็บข้อมูลมีปัญหา P2P-DFS ก็จะติดต่อไปยังเครื่องอื่นที่เก็บสำเนาชิ้นนี้เองโดยอัตโนมัติ

3.1 แฟ้มข้อมูลพิเศษ arp

ในการทำงานของ P2P-DFS ไม่ว่าจะเป็นในส่วนของการดึงข้อมูลออกจากระบบหรือนำข้อมูลเข้าไปในระบบจะมีการใช้แฟ้มข้อมูลพิเศษร่วมกันโดยกำหนดให้มีนามสกุล arp (Atiwat Rungroj Puwarun) ตามชื่อผู้พัฒนาโปรแกรม โครงสร้างของแฟ้มข้อมูลนี้แสดงในภาพที่ 2

ภาพที่ 2 โครงสร้างแฟ้มข้อมูลพิเศษ arp

แฟ้มข้อมูลนี้จะเก็บข้อมูลออกเป็นสองส่วน ส่วนแรกจะเก็บข้อมูลของแฟ้มข้อมูลซึ่งประกอบไปด้วย ชื่อแฟ้มข้อมูลที่จัดเก็บอยู่ในระบบ ค่า MD5 รวมของแฟ้มข้อมูล และจำนวนสำเนาของแฟ้มข้อมูล ในส่วนที่สองจะจัดเก็บข้อมูลของแต่ละชิ้นส่วนย่อยของแฟ้มข้อมูลซึ่งจะประกอบไปด้วยหมายเลขไอพีและพอร์ทของเครื่องที่จัดเก็บชิ้นส่วนข้อมูลนั้นไว้ ค่า MD5 ของชิ้นส่วนข้อมูลนั้น และ ตำแหน่งเริ่มต้นของชิ้นส่วนข้อมูลในแฟ้มข้อมูล (Offset)

ซึ่งลักษณะการทำงานของแฟ้มข้อมูลพิเศษ arp นี้จะมีลักษณะเหมือนกับแฟ้มข้อมูล torrent คือผู้ใช้งานสามารถเรียกใช้แฟ้มข้อมูลนี้ผ่านเครื่องคอมพิวเตอร์เครื่องใดก็ได้ที่มีการเรียกใช้งานโปรแกรม P2P-DFS

3.2 การส่งแฟ้มข้อมูลเข้าไปจัดเก็บใน P2P-DFS

เมื่อผู้ใช้ต้องการส่งแฟ้มข้อมูลเข้าไปจัดเก็บใน P2P-DFS โปรแกรมจะทำการคำนวณหาค่าขนาดชิ้นส่วนย่อยของแฟ้มข้อมูลเพื่อกระจายส่งไปยังเครื่องอื่นๆ ซึ่งจะคำนวณจากการนำขนาดของแฟ้มข้อมูลมาหารด้วยจำนวนของชิ้นส่วนที่ต้องการแบ่ง โดยผู้ใช้เป็นคนกำหนดเอง ในการแบ่งแฟ้มข้อมูลนั้นปกติทุกชิ้นส่วนย่อยจะมีขนาดเท่ากัน ยกเว้นชิ้นส่วนแรกจะมีขนาดใหญ่กว่าชิ้นส่วนอื่น เนื่องจากหากเมื่อมีการคำนวณแล้วผลที่ได้ไม่ลงตัว โปรแกรมจะนำค่านั้นไปเพิ่มให้กับชิ้นส่วนย่อยชิ้นแรก

จากนั้น P2P-DFS จะติดต่อกับ ULAP เพื่อขอหมายเลขไอพีและพอร์ทของเครื่องที่อยู่ในระบบเพื่อใช้สำหรับเก็บชิ้นส่วนของแฟ้มข้อมูล ซึ่งการส่งชิ้นส่วนแฟ้มข้อมูลเข้าไปจัดเก็บในระบบนั้น ชิ้นส่วนข้อมูลจะถูกส่งไปพร้อมกันกับค่า MD5 ของชิ้นส่วนนั้น โดยเครื่องที่รับข้อมูลจะบันทึกข้อมูลลงในแฟ้มข้อมูลที่มีชื่อตามค่าของ MD5 ที่ส่งมาด้วย จากนั้นจะทำการตรวจสอบข้อมูลที่รับมาโดยเปรียบเทียบค่า MD5 ของข้อมูลที่ได้จากการคำนวณกับค่า MD5 ที่รับมาว่าข้อมูลที่รับเข้ามามีความถูกต้องหรือไม่ ถ้าไม่มีปัญหาใดๆเกิดขึ้นเครื่องที่รับชิ้นส่วนของข้อมูลนั้นจะแจ้งให้เครื่องผู้ส่งรับทราบว่าการส่งข้อมูลเรียบร้อย ซึ่งระหว่างการส่งข้อมูลนี้เองโปรแกรมจะสร้างแฟ้มข้อมูลพิเศษ arp ขึ้นมาเพื่อเก็บรายละเอียดข้อมูลเกี่ยวกับแฟ้มข้อมูลที่ถูกจัดเก็บในระบบขึ้นที่ฝั่งผู้ส่งข้อมูล

3.3 การดึงแฟ้มข้อมูลออกจาก P2P-DFS

ในการดึงแฟ้มข้อมูลออกจาก P2P-DFS ผู้ใช้จำเป็นต้องมีแฟ้มข้อมูลพิเศษ arp ที่ใช้เก็บข้อมูลรายละเอียดเกี่ยวกับแฟ้มข้อมูลนั้น ซึ่งเมื่อผู้ใช้ป้อนแฟ้มข้อมูล arp ให้กับโปรแกรม โปรแกรมจะทำการอ่านข้อมูลหมายเลขไอพีและพอร์ทของเครื่องที่เก็บชิ้นส่วนแฟ้มข้อมูลนั้นไว้ และจะติดต่อไปยังเครื่องเหล่านั้นเพื่อทำการขอข้อมูลกลับมา การทำงานในส่วนของการดึงข้อมูลนี้จะทำการดึงข้อมูลจากเครื่องที่เก็บชิ้นส่วนย่อยทุกเครื่องพร้อมๆกัน ดังนั้นการรวมแฟ้มข้อมูลจะต้องคำนึงถึงตำแหน่งของชิ้นส่วนย่อยซึ่งจะต้องมีการเรียงลำดับก่อนหลังของข้อมูลอย่างถูกต้อง ปัญหาในส่วนนี้สามารถขจัดได้ด้วยการอาศัยข้อมูลที่ได้บันทึกไว้ในแฟ้มข้อมูลพิเศษ arp ขั้นตอนการตรวจสอบความถูกต้องของแฟ้มข้อมูลที่ดึงออกจาก P2P-DFS จะแบ่งออกเป็นสองขั้นตอน ขั้นตอนแรกเมื่อดึงข้อมูล แต่ละชิ้นส่วนย่อยเสร็จสิ้น โปรแกรมจะนำค่า MD5 ที่บันทึกไว้ในแฟ้มข้อมูล arp ของชิ้นส่วนนั้นมาเปรียบเทียบกับค่า MD5 ที่คำนวณได้ของชิ้นส่วนที่รับเข้ามา ถ้าค่าที่ได้ไม่เท่ากันโปรแกรมจะทำการร้องขอไปยังเครื่องอื่นที่เก็บสำเนาของชิ้นส่วนนี้ เนื่องจากอาจเป็นไปได้ว่าข้อมูลชิ้นส่วนนั้นถูกผู้ใช้เครื่องคอมพิวเตอร์เครื่องนั้นเปลี่ยนแปลงข้อมูล หรืออุปกรณ์จัดเก็บข้อมูลของเครื่องคอมพิวเตอร์นั้นมีปัญหา ถ้าการตรวจสอบชิ้นส่วนของแฟ้มข้อมูลทุกชิ้นส่วนถูกต้องขั้นตอนที่สองโปรแกรมจะทำการตรวจสอบค่าความถูกต้องของทั้งแฟ้มข้อมูลอีกครั้ง โดยการคำนวณค่า MD5 ของทั้งแฟ้มข้อมูลและเปรียบเทียบกับค่า MD5 รวมของแฟ้มข้อมูลที่ถูกเก็บในแฟ้มข้อมูลพิเศษ arp ถ้าการตรวจสอบถูกต้องเรียบร้อยทุกอย่าง โปรแกรมจะแจ้งกับผู้ใช้ให้ทราบถึงการเสร็จสิ้นการทำงาน

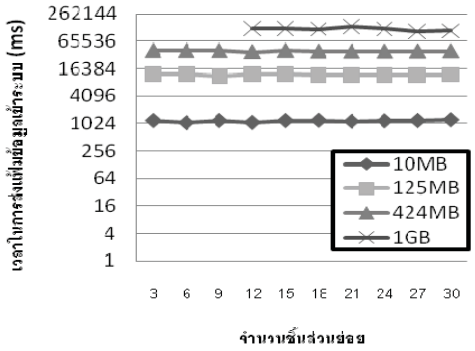
3.4 การค้นหาทรัพยากรใหม่เมื่อมีการล้มเหลวในระบบ

เมื่อมีการติดต่อกันระหว่างเครื่องคอมพิวเตอร์ในระบบเครือข่ายสิ่งที่หลีกเลี่ยงไม่ได้ในการพัฒนาโปรแกรมคือต้องมีการจัดการกับเหตุขัดข้องที่อาจเกิดขึ้นมาจากปัจจัยต่างๆได้ เช่น ระบบเครือข่ายล้มเหลว เครื่องคอมพิวเตอร์ปลายทางล้มเหลว ใน P2P-DFS การล้มเหลวสามารถเกิดขึ้นได้ทั้งขณะที่ส่งข้อมูลไปเก็บยังเครื่องปลายทาง และการดึงข้อมูลจากเครื่องปลายทาง แต่อย่างไรก็ตาม P2P-DFS สามารถแก้ไขปัญหาเหล่านี้ได้ ถ้าปัญหาเกิดขึ้นระหว่างการส่งข้อมูลโปรแกรม P2P-DFS จะติดต่อกับ ULAP เพื่อขอหมายเลขไอพีและพอร์ทของเครื่องคอมพิวเตอร์เครื่องอื่นในระบบมาแทนเครื่องที่มีปัญหา ส่วนปัญหาที่เกิดขึ้นระหว่างการดึงข้อมูล P2P-DFS ได้นำเสนอการทำสำเนาของแฟ้มข้อมูลและบันทึกไว้ในแฟ้มข้อมูล arp ดังนั้นโปรแกรมจะไปติดต่อกับเครื่องคอมพิวเตอร์เครื่องอื่นที่เก็บสำเนาเดียวกันกับเครื่องที่มีปัญหา

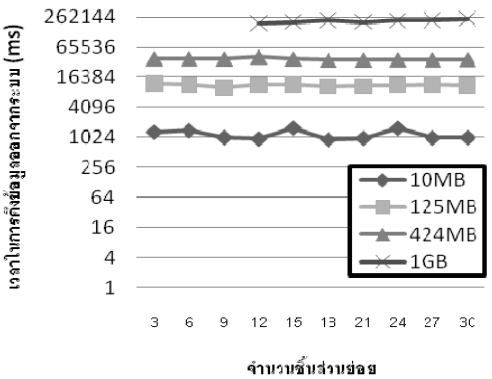
แทน ทุกขั้นตอนในการแก้ปัญหานี้จะทำโดยอัตโนมัติ ซึ่งผู้ใช้ไม่จำเป็นต้องเข้ามาจัดการใดๆทั้งสิ้น

4. ผลการทดลอง

การทดลองได้จัดทำภายในห้องประลองคอมพิวเตอร์หนึ่งห้อง ที่วิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ โดยมีเครื่องคอมพิวเตอร์ที่ทำการทดลองทั้งหมดจำนวน 30 เครื่อง (Core 2 duo E7500 2.9 Ghz, Ram 2 GB, HDD 250 GB) เชื่อมต่อกันภายในระบบเครือข่ายท้องถิ่นความเร็ว 100 Mbps การทดลองจะแบ่งออกเป็น 2 ส่วนคือ วัดความเร็วในการส่งแฟ้มข้อมูลเข้าไปจัดเก็บในระบบ และวัดความเร็วในการดึงแฟ้มข้อมูลออกจากระบบ ในการทดลองจะใช้แฟ้มข้อมูลทั้งหมด 4 ขนาดคือ 10MB, 125MB, 424MB และ 1 GB และมีการปรับเปลี่ยนจำนวนชิ้นส่วนย่อยที่ต้องการแบ่งให้กับแฟ้มข้อมูลออกเป็น 3, 6, 9, 12, 15, 18, 21, 24, 27, และ 30 ชิ้น



ภาพที่ 3 ความเร็วในการส่งแฟ้มข้อมูลเข้าระบบตามจำนวนชิ้นส่วนย่อยต่างๆ



ภาพที่ 4 ความเร็วในการดึงข้อมูลออกจากระบบตามจำนวนชิ้นส่วนย่อยต่างๆ

ภาพที่ 3 แสดงเวลาที่ใช้ในการส่งแฟ้มข้อมูลเข้าระบบ จากการทดลองพบว่าเวลาที่ใช้นั้นค่อนข้างคงที่ไม่ว่าจะแบ่งแฟ้มข้อมูลออกเป็น 3 ส่วนย่อย หรือ 30 ส่วนย่อย ซึ่งค่าที่แตกต่างกันนั้นอยู่ในหน่วยวินาทีที่สามารถละเลยได้ ความเร็วในการส่งข้อมูลของทุกขนาดแฟ้มข้อมูลที่ทดลองอยู่ที่ 8 – 10 MB/s ซึ่งถือว่ามีประสิทธิภาพในส่งข้อมูลดี เนื่องจากปกติโปรแกรมจะมีการคำนวณหาค่า MD5 ไปด้วยขณะที่ทำการส่งข้อมูลไปยังเครื่องคอมพิวเตอร์เครื่องอื่นๆ

ภาพที่ 4 แสดงเวลาที่ใช้ในการดึงแฟ้มข้อมูลออกจากระบบ ซึ่งเวลาที่ใช้ในการดึงข้อมูลสำหรับแฟ้มข้อมูลขนาด 10MB, 125 MB และ 424 MB ค่อนข้างจะคงที่ไม่ว่าจะแบ่งออกเป็นส่วนใหญ่กี่ส่วนก็ตาม และมีประสิทธิภาพในการทำงานใกล้เคียงกับประสิทธิภาพของการส่งแฟ้มข้อมูลเข้าสู่ระบบซึ่งจะมีความเร็วอยู่ที่ 8 – 10 MB/s แต่จะสังเกตเห็นว่าเมื่อแฟ้มข้อมูลมีขนาด 1 GB ประสิทธิภาพของระบบจะต่ำลง เนื่องจากการดึงแฟ้มข้อมูลนั้นปัญหาคอขวดจะอยู่ที่การเขียนข้อมูลลงในเครื่องของผู้ใช้ที่ต้องการแฟ้มข้อมูล เนื่องจากด้วยข้อมูลที่รับมาพร้อมกันหลายทางทำให้ตัวเขียนข้อมูลจะต้องเปลี่ยนตำแหน่งในการเขียนข้อมูลบ่อยครั้งโดยเฉพาะอย่างยิ่งกับแฟ้มข้อมูลที่มีขนาดใหญ่

5. บทสรุป

บทความวิจัยนี้ได้นำเสนอระบบแฟ้มข้อมูลแบบกระจายบนเครือข่ายเพียร์ทูเพียร์ที่ถูกพัฒนาขึ้นด้วยภาษา จาวา ซึ่งผู้ใช้สามารถทำการส่งแฟ้มข้อมูลที่ต้องการจัดเก็บเข้าไปยังระบบ โดยแฟ้มข้อมูลนั้นจะถูกแบ่งออกเป็นชิ้นส่วนย่อยๆ ตามที่ผู้ใช้กำหนดและจะถูกส่งไปเก็บยังเครื่องคอมพิวเตอร์อื่นๆที่เข้าร่วมอยู่ในระบบ อีกทั้งผู้ใช้สามารถเลือกจำนวนสำเนาของแฟ้มข้อมูลที่ต้องการจัดเก็บได้ เมื่อการส่งแฟ้มข้อมูลเข้าไปในระบบเสร็จสิ้น โปรแกรมจะสร้างแฟ้มข้อมูลพิเศษนามสกุล arp ขึ้นมาโดยผู้ใช้สามารถจะนำแฟ้มข้อมูลนี้ไปใช้ในการดึงแฟ้มข้อมูลจากระบบกลับมาโดยใช้เครื่องคอมพิวเตอร์เครื่องใดก็ได้ที่ลงโปรแกรม P2P-DFS เอาไว้ โปรแกรมมีการตรวจสอบความถูกต้องของข้อมูลที่รับส่งโดยใช้ค่า MD5 ประโยชน์ของระบบแฟ้มข้อมูลแบบกระจายที่ได้นี้ ทำให้มีการ

ใช้เนื้อที่ในการเก็บข้อมูลของเครื่องคอมพิวเตอร์อย่างมีประสิทธิภาพมากขึ้น มีการทำสำเนาเพื่อป้องกันข้อมูลสำคัญสูญหายเมื่อเครื่องคอมพิวเตอร์มีปัญหา มีการกระจายภาระของเครื่องคอมพิวเตอร์ในระบบเครือข่ายให้เท่าๆกันเพื่อป้องกันปัญหาคอขวดของเครื่องแม่ข่าย ผลจากการทดลองชี้ให้เห็นว่า P2P-DFS มีประสิทธิภาพการทำงานที่ดี แต่อย่างไรก็ตาม P2P-DFS ยังขาดคุณสมบัติหลายอย่างซึ่งจะมีการพัฒนาในอนาคต เช่น การอ่านและเขียนข้อมูลแบบสตรีมลงในระบบโดยตรง และ การทำให้รองรับการเขียนโปรแกรมแบบ Map/Reduce

6. กิตติกรรมประกาศ

ขอขอบคุณวิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือที่ให้ทุนในการวิจัยและพื้นที่ในการทดลอง

7. เอกสารอ้างอิง

[1] C. Shirky, “Peer-to-Peer: Harnessing the Power of Disruptive Technologies”, chapter 2, page 19-28, O’Reilly, 2001.
[2] B. Cohen, “Incentives build robustness in bittorrent” . [Online]. Available : http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.1911
[3] D. Borthakur, “The Hadoop Distributed File System: Architecture and Design”, The Apache Software Foundation, 2007.
[4] S. Ghemawat, H. Gobioff, and S. T. Leung, "The google file system," *SIGOPS Oper. Syst. Rev.*, vol. 37, no. 5, pp. 29-43, 2003. [Online]. Available: http://dx.doi.org/10.1145/945445.945450
[5] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," pp. 137-150. [Online]. Available: http://www.usenix.org/events/osdi04/tech/dean.html
[6] Facebook. http://www.facebook.com/.
[7] Twitter. http://www.twitter.com/.
[8] Yahoo!. http://www.yahoo.com/.
[9] C. Rattanapoka, “ULAP: An Unstructured Locality-Awareness Peer-toPeer Network”. EENET 2010, Chiang-mai
[10] A. Oram, “Peer-to-Peer: Harnessing the Power of Ditruptive Technologies, chapter Gnutella, page 94-122. O’Reilly, May 2001.