การวิเคราะห์ประสิทธิภาพของขั้นตอนวิธีการแบ่งชิ้นส่วนไฟล์ของระบบสำรองข้อมูลแบบเพียร์ทูเพียร์

นางสาวจารินี ชยาภิรมย์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2552

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

PERFORMANCE ANALYSIS OF FILE-SPLITTING ALGORITHM IN PEER-TO-PEER NETWORK BACKUP SYSTEMS

Ms. Jarinee Chayapirom

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2009

Copyright of Chulalongkorn University

520389

หัวข้อวิทยานิพนธ์	การวิเคราะห์ประสิทธิภาพของขั้นตอนวิธีการแบ่งชิ้นส่วนไฟล์ของระบบสำรองข้อมูลแบบเพียร์ทูเพียร์
โดย	นางสาวจารินี ชยาภิรมย์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	อาจารย์ ดร. เกริก ภิรมย์โสภา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์ ดร. ณัฐวุฒิ หนูไพโรจน์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(อาจารย์ ดร. เกริก ภิรมย์โสภา)

..... กรรมการ
(อาจารย์ ดร.กุลธิดา โรจน์วิบูลย์ชัย)

..... กรรมการภายนอกมหาวิทยาลัย
(ดร.พงศ์ธวัช ชีพพิมลชัย)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

จารินี ชยาภิรมย์ : การวิเคราะห์ประสิทธิภาพของขั้นตอนวิธีการแบ่งชิ้นส่วนไฟล์ของระบบสำรองข้อมูลแบบเพียร์ทูเพียร์. (PERFORMANCE ANALYSIS OF FILE-SPLITTING ALGORITHM IN PEER-TO-PEER NETWORK BACKUP SYSTEMS) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : อ.ดร.เกริก ภิรมย์โสภา, 62 หน้า.

เวลาที่ใช้ในการสำรองข้อมูลและสภาพพร้อมใช้งานของข้อมูลเป็นปัจจัยหลักที่ผู้ใช้ระบบสำรองข้อมูลแบบเพียร์ทูเพียร์คำนึงถึง จุดประสงค์ของงานวิจัยนี้คือ นำเสนอแบบจำลองสำหรับการตัดสินใจประสิทธิภาพ (Performance Decisive Model, P) เพื่อประเมินประสิทธิภาพในด้านเวลาที่ใช้และสภาพพร้อมใช้งานของไฟล์ทั้งสองด้านพร้อมกัน โดยทำการทดลองเปรียบเทียบการสำรองข้อมูลบนระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ซึ่งใช้ขั้นตอนวิธีการแบ่งชิ้นส่วนย่อย 3 รูปแบบ ได้แก่ การแบ่งชิ้นส่วนย่อยตามจำนวนเครื่องในระบบ, การแบ่งชิ้นส่วนย่อยโดยกำหนดขนาดของชิ้นส่วนย่อยคงที่ และการแบ่งชิ้นส่วนย่อยโดยใช้อีเรเซอร์โค้ด (เลือกใช้ Reed-Solomon และ Luby Transform Code) ผลการทดลองแสดงให้เห็นว่าการเพิ่มสภาพพร้อมใช้งานของไฟล์ไม่ได้แปรผันตามจำนวนเครื่องภายในระบบเสมอ แบบจำลองดังกล่าวสามารถนำมาใช้เป็นแนวทางสำหรับการตัดสินใจเลือกขั้นตอนวิธีรวมถึงการเลือกพารามิเตอร์ที่เหมาะสมสำหรับแต่ละวิธี

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา .....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต : .....

สาขาวิชา .....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์หลัก : .....

ปีการศึกษา ..2552.....

5170246021 : MAJOR COMPUTER ENGINEERING

KEYWORDS: SYSTEM ARCHITECTURES / PEER-TO-PEER NETWORK BACKUP

JARINEE CHAYAPIROM : PERFORMANCE ANALYSIS OF FILE-SPLITTING ALGORITHM IN PEER-TO-PEER NETWORK BACKUP SYSTEMS. THESIS ADVISOR : KRERK PIROMSOPA, Ph.D., 62 pp.

Time and availability are two main factors that contribute to the adoption of peer-to-peer network backup systems. In this thesis, a Performance Decisive Model is proposed to provide a framework for evaluating backup time and file availability of the backup systems. In our study, files are splitting in three different modes: one chunk per node, fixed block size, and erasure code (Reed-Solomon and Luby Transform in particular). The results show that increasing the number of peers does not always increasing file availability. To get an optimal backup time (with best file availability), we discuss guidelines for suitable algorithm and parameter selections.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Department : ..Computer Engineering Student's Signature : .....

Field of Study : Computer Engineering Advisor's Signature : .....

Academic Year : ...2009.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความอนุเคราะห์อย่างยิ่งของอาจารย์ ดร. เกริก ภิรมย์โสภา อาจารย์ที่ปรึกษา ซึ่งท่านได้ให้ความรู้ แนะนำแนวทางการวิจัย ตรวจสอบให้คำแนะนำ และสนับสนุนเป็นอย่างดี จนทำให้การวิจัยในครั้งนี้สำเร็จออกมาด้วยดี

ขอขอบพระคุณ อาจารย์ ดร. ณัฐวุฒิ หนูไพโรจน์ อาจารย์ ดร. พงศ์ธวัช ชีพพิมลชัย และ อาจารย์ ดร.กุลธิดา โรจน์วิบูลย์ชัย กรรมการสอบวิทยานิพนธ์ ที่กรุณาเสียสละเวลา ให้คำแนะนำ ตรวจสอบ และแก้ไขวิทยานิพนธ์ฉบับนี้

ขอขอบพระคุณ ศูนย์คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ได้ เอื้อเฟื้อเครื่องคอมพิวเตอร์สำหรับการทดลองในงานวิจัยนี้

ท้ายที่สุด ผู้เสนอวิทยานิพนธ์ขอขอบคุณเพื่อน ๆ ทุก ๆ คน รวมทั้งครอบครัว ที่คอย ติดตาม ให้กำลังใจและสนับสนุน รวมถึงท่านอื่น ๆ ที่มิได้กล่าวชื่อไว้ ณ ที่นี้ที่มีส่วนช่วยให้ วิทยานิพนธ์สำเร็จได้ด้วยดี

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญภาพ ..... ฌ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของการวิจัย 2

1.3 ขอบเขตของการวิจัย 2

1.4 ขั้นตอนและวิธีดำเนินการวิจัย 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ 3

1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์ 4

1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์ 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5

2.1 ทฤษฎีที่เกี่ยวข้อง 5

2.2 งานวิจัยที่เกี่ยวข้อง 9

บทที่ 3 แนวทางการออกแบบการทดลองและแบบจำลองเบื้องต้น (Initial Models) 12

3.1 เครื่องมือที่ใช้ในการวิจัย 12

3.2 การออกแบบการทดลอง 12

3.3 แบบจำลองเบื้องต้น (Initial Models) 16

บทที่ 4 การทดลองในแต่ละขั้นตอนวิธีการแบ่งชิ้นส่วนย่อย 17

4.1 วิธีที่ 1 : แบ่งชิ้นส่วนย่อยตามจำนวนเครื่องในระบบ 17

4.2 วิธีที่ 2 : แบ่งชิ้นส่วนย่อยโดยกำหนดขนาดของชิ้นส่วนย่อยคงที่ 20

4.3 วิธีที่ 3 : แบ่งชิ้นส่วนย่อยโดยใช้อีเรเซอร์โค้ด 23

บทที่ 5 วิเคราะห์และสรุปผลการทดลอง 37

5.1 แบบจำลองของเวลาและสภาพพร้อมใช้งานของไฟล์ (Time-Availability Model) 37

5.2 การวิเคราะห์ประสิทธิภาพ 38

5.3 อภิปรายผลการวิจัย 50

บทที่ 6 บทสรุปของการวิจัย 52
6.1 สิ่งที่ได้จากการวิจัย (Contribution) 52
6.2 แนวทางการวิจัยต่อ 52
6.3 บทสรุป 52
รายการอ้างอิง 54
ภาคผนวก 56
ภาคผนวก ก. ระบบสำรองข้อมูลที่มีอยู่ในปัจจุบัน 57
ภาคผนวก ข. ตัวอย่างหน้าจอของส่วนต่อประสานผู้ใช้ของระบบสำรองข้อมูล 61
แบบเพียร์ทูเพียร์ที่พัฒนาโดยภาษาจาวา (JAVA) 61
ประวัติผู้เขียนวิทยานิพนธ์ 62

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 1 แผนภาพแสดงการแบ่งชิ้นส่วนย่อยเพื่อทำการเข้ารหัสและถอดรหัสข้อมูลในการทำ อีเรเซอร์โค้ด [3] 5
รูปที่ 2 ภาพแสดงการเปรียบเทียบระหว่าง Combinatorial Effect ($_{Sb}C_i$) และ Peer Availability Effect ($\mu^i (1- \mu)^{Sb-i}$) และผลจากการคูณกันของทั้งสองพจน์โดยเทียบกับจำนวนชิ้นส่วนย่อย 6
รูปที่ 3 ผลกระทบของสภาพพร้อมใช้งานของไฟล์เมื่อค่าสภาพพร้อมใช้งานของเพียร์ต่างกัน 7
รูปที่ 4 ผลกระทบของสภาพพร้อมใช้งานของไฟล์เมื่อค่าปัจจัยการยึดต่างกัน 8
รูปที่ 5 ภาพแสดงการแบ่งชิ้นส่วนย่อยตามจำนวนเครื่องในระบบ 13
รูปที่ 6 ภาพแสดงการแบ่งชิ้นส่วนย่อยโดยกำหนดขนาดของชิ้นส่วนย่อยคงที่ 13
รูปที่ 7 ภาพแสดงการแบ่งชิ้นส่วนย่อยโดยใช้อีเรเซอร์โค้ด 14
รูปที่ 8 กราฟแสดงเวลาที่ใช้ในการสำรองข้อมูลลงบนเครื่องตั้งแต่ 2 ถึง 30 เครื่องและเลือกใช้ขนาดไฟล์ตั้งแต่ 10 MB จนถึง 3 GB 18
รูปที่ 9 กราฟแสดงพื้นผิวของสมการที่ใช้ประมาณเวลาในการสำรองข้อมูล $T_1(sz,n)$ เทียบกับจำนวนเครื่องในระบบและขนาดไฟล์ 19
รูปที่ 11 กราฟแสดงเวลาในการสำรองข้อมูลเทียบกับจำนวนเครื่องในระบบและขนาดไฟล์เมื่อเลือกใช้ขนาดชิ้นส่วนย่อย 10 MB 22
รูปที่ 12 กราฟแสดงเวลาในการสำรองข้อมูลเทียบกับจำนวนชิ้นส่วนย่อยเมื่อเลือกใช้ขนาดไฟล์10, 50,100, 250 และ 500 MB บนจำนวน 2 เครื่อง โดยใช้ RS Code (S=2) 24
รูปที่ 13 กราฟแสดงเวลาในการสำรองข้อมูลเทียบกับจำนวนชิ้นส่วนย่อยเมื่อเลือกใช้ขนาดไฟล์10, 50 และ 100 MB บนจำนวน 3 เครื่อง โดยใช้ RS Code (S=2) 25
รูปที่ 14 กราฟแสดงเวลาในการสำรองข้อมูลเทียบกับจำนวนชิ้นส่วนย่อยเมื่อเลือกใช้ขนาดไฟล์10, 50 และ 100 MB บนจำนวน 5 เครื่อง โดยใช้ RS Code (S=2) 26
รูปที่ 15 กราฟแสดงเวลาในการสำรองข้อมูลในส่วนการเข้ารหัสเทียบกับจำนวนเครื่องในระบบเมื่อเลือกใช้ขนาดไฟล์ 10, 50 และ 100 MB โดยแบ่งชิ้นส่วนย่อย 3 ชิ้นด้วย RS Code (S=2) 27
รูปที่ 16 กราฟแสดงเวลาในการสำรองข้อมูลในส่วนการเข้ารหัสเทียบกับจำนวนเครื่องในระบบเมื่อเลือกใช้ขนาดไฟล์ 10, 50 และ 100 MB โดยแบ่งชิ้นส่วนย่อย 4 ชิ้นด้วย RS Code (S=2) 27

รูปที่ 17 กราฟแสดงเวลาในการสำรองข้อมูลในส่วนรับส่งข้อมูลเทียบกับจำนวนเครื่องในระบบ เมื่อเลือกใช้ขนาดไฟล์ 10, 50 และ 100 MB โดยแบ่งชิ้นส่วนย่อย 3 ชิ้นด้วย RS Code (S=2) 28
รูปที่ 18 กราฟแสดงเวลาในการสำรองข้อมูลในส่วนรับส่งข้อมูลเทียบกับจำนวนเครื่องในระบบเมื่อเลือกใช้ขนาดไฟล์ 10, 50 และ 100 MB โดยแบ่งชิ้นส่วนย่อย 4 ชิ้นด้วย RS Code (S=2) 29
รูปที่ 19 กราฟแสดงเวลาในการสำรองข้อมูลทั้งหมดโดยเปรียบเทียบระหว่าง RS Code และ LT Code ที่ค่า S=2 เมื่อเลือกใช้ขนาดไฟล์ 10 MB บนเครื่องจำนวน 2 เครื่อง 31
รูปที่ 20 กราฟแสดงเวลาในการสำรองข้อมูลทั้งหมดโดยเปรียบเทียบระหว่าง RS Code และ LT Code ที่ค่า S=2 เมื่อเลือกใช้ขนาดไฟล์ 50 MB บนเครื่องจำนวน 2 เครื่อง 32
รูปที่ 21 กราฟแสดงเวลาในการสำรองข้อมูลทั้งหมดโดยเปรียบเทียบระหว่าง RS Code และ LT Code ที่ค่า S=2 เมื่อเลือกใช้ขนาดไฟล์ 100 MB บนเครื่องจำนวน 2 เครื่อง 33
รูปที่ 22 กราฟแสดงเวลาในการสำรองข้อมูลในส่วนการเข้ารหัสข้อมูลและการรับส่งข้อมูลโดยเปรียบเทียบระหว่าง RS Code และ LT Code ที่ค่า S=2 เมื่อเลือกใช้ขนาดไฟล์ 10 MB บนเครื่องจำนวน 2 เครื่อง 34
รูปที่ 23 กราฟแสดงเวลาในการสำรองข้อมูลในส่วนการเข้ารหัสข้อมูลและการรับส่งข้อมูลโดยเปรียบเทียบระหว่าง RS Code และ LT Code ที่ค่า S=2 เมื่อเลือกใช้ขนาดไฟล์ 50 MB บนเครื่องจำนวน 2 เครื่อง 35
รูปที่ 24 กราฟแสดงเวลาในการสำรองข้อมูลในส่วนการเข้ารหัสข้อมูลและการรับส่งข้อมูลโดยเปรียบเทียบระหว่าง RS Code และ LT Code ที่ค่า S=2 เมื่อเลือกใช้ขนาดไฟล์ 100 MB บนเครื่องจำนวน 2 เครื่อง 36
รูปที่ 25 กราฟของปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) เทียบกับจำนวนชิ้นส่วนย่อย (b) และค่าปัจจัยการยึด (S) โดยใช้จำนวนเครื่องในระบบ 2 เครื่องและขนาดไฟล์ 10 MB เมื่อสภาพพร้อมใช้งานของเพียร์ (μ) เท่ากับ 0.3 40
รูปที่ 26 กราฟของปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) เทียบกับจำนวนชิ้นส่วนย่อย (b) และค่าปัจจัยการยึด (S) โดยใช้จำนวนเครื่องในระบบ 2 เครื่องและขนาดไฟล์ 10 MB เมื่อสภาพพร้อมใช้งานของเพียร์ (μ) เท่ากับ 0.5 41
รูปที่ 27 กราฟของปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) เทียบกับจำนวนชิ้นส่วนย่อย (b) และค่าปัจจัยการยึด (S) โดยใช้จำนวนเครื่องในระบบ 2 เครื่องและขนาดไฟล์ 10 MB เมื่อสภาพพร้อมใช้งานของเพียร์ (μ) เท่ากับ 0.7 42

รูปที่ 28 กราฟแสดงการเปรียบเทียบปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) ระหว่างวิธีแรก และวิธีที่สามซึ่งใช้ RS Code (S=2, μ=0.3, b=2) โดยเทียบกับจำนวนเครื่องในระบบ (n) และใช้ขนาดไฟล์ 750 MB 44
รูปที่ 29 กราฟแสดงการเปรียบเทียบปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) ระหว่างวิธีแรก และวิธีที่สามซึ่งใช้ RS Code (S=2, μ=0.7, b=30) โดยเทียบกับจำนวนเครื่องในระบบ (n) และใช้ขนาดไฟล์ 1500 MB 45
รูปที่ 30 กราฟแสดงการเปรียบเทียบปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) ของวิธีแรก ระหว่างค่าสภาพพร้อมใช้งานของเพียร์ μ=0.3 และ 0.7 โดยเทียบกับจำนวนเครื่องในระบบ (n) และใช้ขนาดไฟล์ 1500 MB 46
รูปที่ 31 กราฟแสดงการเปรียบเทียบปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) ของวิธีแรก เทียบกับจำนวนเครื่องในระบบ (n) ที่ค่าสภาพพร้อมใช้งานของเพียร์ μ= 0.7 โดยใช้ขนาดไฟล์ 750 MB, 1000 MB และ 1500 MB 47
รูปที่ 32 กราฟแสดงการเปรียบเทียบปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) ระหว่างวิธีที่ 2 และวิธีที่ 3 ซึ่งใช้ RS Code (S=2, μ=0.3) โดยเทียบกับจำนวนชิ้นส่วนย่อย (b) และใช้ขนาดไฟล์ 750 MB บนจำนวนเครื่อง 5 เครื่อง 49
รูปที่ 33 กราฟแสดงการเปรียบเทียบปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) ระหว่างวิธีที่ 2 และวิธีที่ 3 ซึ่งใช้ RS Code (S=2, μ=0.7) โดยเทียบกับจำนวนชิ้นส่วนย่อย (b) และใช้ขนาดไฟล์ 750 MB บนจำนวนเครื่อง 5 เครื่อง 50
รูปที่ 34 ภาพแสดงอันดับและเปรียบเทียบคุณสมบัติการใช้งานด้านต่างๆของโปรแกรมสำรองข้อมูลที่ได้รับความนิยมมากที่สุด 10 อันดับแรก 60
รูปที่ 35 ตัวอย่างหน้าจอแสดงส่วนต่อประสานกับผู้ใช้ (User Interface) ในส่วนของการสำรองข้อมูล (Backup) 61

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากปัจจุบันข้อมูลในด้านต่าง ๆ มีความสำคัญต่อมนุษย์และระบบการทำงานมากขึ้น การดำเนินธุรกรรมต่าง ๆ รวมถึงในด้านงานวิทยาศาสตร์หรืองานวิจัยต่างก็มีองค์ประกอบหลักคือข้อมูลซึ่งมีความสำคัญต่อผู้ใช้เป็นอย่างมาก ดังนั้นเพื่อให้สามารถกู้คืนข้อมูลกลับมาในกรณีที่ข้อมูลเหล่านั้นสูญหายหรือเปลี่ยนแปลงไป จึงจำเป็นที่จะต้องมีการสำรองข้อมูลอย่างสม่ำเสมอ ในการสำรองข้อมูลนั้นปัจจัยที่สำคัญซึ่งต้องคำนึงถึงมีหลายด้านด้วยกัน อาทิเช่น เวลาที่ใช้ในการสำรองข้อมูล (Backup Time), ความเชื่อถือได้ของข้อมูล (Data Reliability), ความปลอดภัยของข้อมูล (Security) เป็นต้น โดยปัจจัยหลักที่ผู้ใช้ให้ความสำคัญในการสำรองข้อมูลคือเวลาที่ใช้ในขณะที่ทำการสำรองข้อมูลและสภาพพร้อมใช้งานของไฟล์ (File Availability) ที่ได้ทำการสำรองลงในระบบเมื่อทำการกู้คืน

จากการสำรวจระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ (Peer-to-Peer Network Backup System) ที่มีอยู่ทั่วไป (รายละเอียดเพิ่มเติมอยู่ในภาคผนวก ก. หน้า 57) พบว่าส่วนใหญ่มีการทำงานในลักษณะเดียวกันคือ เมื่อผู้ใช้ได้เลือกไฟล์ที่จะทำการสำรองข้อมูล ระบบจะทำการแบ่งไฟล์เป็นชิ้นส่วนย่อย แล้วส่งชิ้นส่วนย่อยเหล่านั้นไปยังเครื่องภายในระบบเพื่อจัดเก็บ เมื่อผู้ใช้ระบุไฟล์ที่จะทำการกู้คืน ระบบจะเรียกคืนข้อมูลจากเครื่องในระบบที่ได้เก็บชิ้นส่วนย่อยของไฟล์นั้น หลังจากที่ได้รับชิ้นส่วนครบจึงทำการรวมเป็นไฟล์ต้นฉบับเดิม จะเห็นได้ว่าขั้นตอนในการแบ่งไฟล์เป็นชิ้นส่วนย่อยเพื่อทำการจัดเก็บในระบบนั้นสามารถทำได้หลายขั้นตอนวิธี (Algorithm) เช่น แบ่งโดยกำหนดขนาดของชิ้นส่วนย่อยคงที่, แบ่งตามจำนวนเครื่องในระบบ หรือแบ่งโดยเข้ารหัสด้วยอีเรเซอร์โค้ด (Erasure Code) สำหรับสร้างสำเนาข้อมูลเพื่อเพิ่มสภาพพร้อมใช้งานของไฟล์ในกรณีที่มีชิ้นส่วนย่อยสูญหาย การทำสำเนาข้อมูลโดยใช้อีเรเซอร์โค้ดนั้นในปัจจุบันมีรูปแบบการเข้ารหัสข้อมูลที่นิยมหลายรูปแบบ เช่น Reed-Solomon Code [1],Tornado Code [1], Luby Transform Code [1] และ Raptor Code [1] เป็นต้น แต่ละขั้นตอนวิธีที่ได้กล่าวข้างต้นมีจุดเด่นและจุดด้อยที่ต่างกัน รวมทั้งส่งผลต่อเวลาในการสำรองข้อมูลและสภาพพร้อมใช้งานของไฟล์แตกต่างกัน นอกจากนี้ งานวิจัยที่เกี่ยวข้องและระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ที่มีอยู่ในปัจจุบันค่อนข้างมุ่งเน้นการพัฒนาประสิทธิภาพด้านความเร็วหรือเพิ่มสภาพพร้อมใช้งานของไฟล์เพียงด้านใดด้านหนึ่งเท่านั้น งานวิจัยนี้จึงจัดทำขึ้นเพื่อศึกษาและเปรียบเทียบประสิทธิภาพของระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ทั้งในด้านความเร็วและสภาพพร้อมใช้งานของไฟล์เมื่อเลือก

ขั้นตอนวิธีในการแบ่งชิ้นส่วนย่อยที่แตกต่างกัน สำหรับเป็นแนวทางในการตัดสินใจเพื่อเลือกวิธีที่เหมาะสมและเป็นประโยชน์ในการเปรียบเทียบประสิทธิภาพการทำงานของระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ที่มีอยู่ในปัจจุบัน

1.2 วัตถุประสงค์ของการวิจัย

การวิจัยมีวัตถุประสงค์ ดังนี้

1. เพื่อศึกษารูปแบบและการทำงานของการแบ่งชิ้นส่วนย่อยในระบบสำรองข้อมูลแบบเพียร์ทูเพียร์
2. เพื่อทดลองและเปรียบเทียบประสิทธิภาพด้านความเร็วและสภาพพร้อมใช้งานของไฟล์ขณะทำการสำรองข้อมูลเมื่อเลือกใช้ขั้นตอนวิธีในการแบ่งชิ้นส่วนย่อยที่ต่างกัน
3. เพื่อวัดและวิเคราะห์ประสิทธิภาพทั้งสองด้านพร้อมกัน โดยสร้างแบบจำลองซึ่งเป็นประโยชน์ในการตัดสินใจเลือกขั้นตอนวิธีในการแบ่งชิ้นส่วนย่อยที่มีประสิทธิภาพ รวมถึงแนวทางการเลือกพารามิเตอร์ที่เหมาะสมกับระบบ

1.3 ขอบเขตของการวิจัย

ขอบเขตของการวิจัยถูกกำหนดไว้ ดังนี้

1. ทำการทดลองกับระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ซึ่งประกอบไปด้วยเครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการวินโดวส์ (Windows) เชื่อมต่อกันในเครือข่าย LAN ที่มีจำนวนเครื่องภายในระบบไม่เกิน 30 เครื่อง โดยการติดต่อกันภายในระบบอาศัยบริการ mDNS (Multicast DNS) เพื่ออ้างอิงเลขที่อยู่(IP Address) ในการรับส่งข้อมูลระหว่างเครื่อง และทดลองกับขนาดไฟล์ ตั้งแต่ 10 MB จนถึง 3 GB
2. มุ่งเน้นศึกษารูปแบบของการแบ่งชิ้นส่วนย่อยในระบบสำรองข้อมูลแบบเพียร์ทูเพียร์เพื่อพิจารณาประสิทธิภาพการทำงานของระบบในด้านความเร็วและสภาพพร้อมใช้งานของไฟล์ในขณะที่ทำการสำรองข้อมูลเท่านั้น ยังมิได้รวมถึงวิธีในการกระจายชิ้นส่วนย่อย (Data Distribution) ไปยังเครื่องภายในระบบและการกู้คืนข้อมูล
3. กำหนดขั้นตอนวิธีในการแบ่งชิ้นส่วนย่อย 3 รูปแบบ ได้แก่ การแบ่งตามจำนวนเครื่องในระบบ, การแบ่งโดยกำหนดขนาดของชิ้นส่วนย่อยคงที่ และการแบ่งโดยใช้อีเรเซอร์โค้ด
4. การแบ่งชิ้นส่วนย่อยโดยใช้อีเรเซอร์โค้ด ได้เลือกรูปแบบการเข้ารหัส 2 รูปแบบ คือ Reed-Solomon Code (RS Code) และ Luby Transform Code (LT Code) (สำหรับ Tornado Code และ Raptor Code นั้นได้รับการจดสิทธิบัตรและสามารถ

ใช้ได้เฉพาะภายในประเทศสหรัฐอเมริกาเท่านั้น จึงไม่สามารถนำมาปรับใช้ในการทดลองได้)

1.4 ขั้นตอนและวิธีดำเนินการวิจัย

วิธีดำเนินการวิจัย ถูกแบ่งเป็นห้าขั้นตอน ดังนี้

1. ศึกษางานวิจัยที่เกี่ยวข้องกับระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ที่มีอยู่ในปัจจุบัน, การทำสำเนาข้อมูลโดยใช้อีเรเซอร์โค้ด และการวิเคราะห์ประสิทธิภาพของระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ในด้านต่างๆ
2. ปรับปรุงระบบสำรองข้อมูลแบบเพียร์ทูเพียร์โดยใช้ขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยตามที่ได้กล่าวข้างต้น
3. ทดลองเพื่อเปรียบเทียบประสิทธิภาพด้านความเร็วและสภาพพร้อมใช้งานของไฟล์ในแต่ละขั้นตอนวิธี
4. วัดและวิเคราะห์ประสิทธิภาพด้านความเร็วในการสำรองข้อมูลและสภาพพร้อมใช้งานของไฟล์และสร้างแบบจำลองในการตัดสินใจเลือกขั้นตอนวิธีที่มีประสิทธิภาพและเหมาะสมกับระบบ
5. สรุปผลการวิจัยและข้อเสนอแนะ และจัดทำวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย ได้แก่

1. เข้าใจวิธีในการแบ่งชิ้นส่วนย่อยของข้อมูลในระบบสำรองข้อมูลแบบเพียร์ทูเพียร์รูปแบบต่างๆ
2. ได้รับความรู้เกี่ยวกับขั้นตอนวิธีในการแบ่งชิ้นส่วนย่อย รวมทั้งข้อดีและข้อเสีย และผลกระทบในด้านประสิทธิภาพของระบบ
3. ได้แบบจำลองในการตัดสินใจเพื่อสรุปและเลือกขั้นตอนวิธีในการแบ่งชิ้นส่วนย่อยที่มีประสิทธิภาพและเหมาะสมกับระบบ รวมถึงแนวทางการเลือกพารามิเตอร์ในแต่ละวิธี โดยสามารถนำไปปรับใช้กับระบบสำรองข้อมูลแบบเพียร์ทูเพียร์เพื่อให้มีความยืดหยุ่นและเหมาะสมกับสภาพแวดล้อมที่ต่างกันได้อย่างมีประสิทธิภาพ
4. สามารถนำความรู้จากผลการวิจัยนี้ไปประยุกต์ใช้จริงในระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ต่อไปในอนาคต

1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์นี้แบ่งเนื้อหาออกเป็น 6 บท ดังต่อไปนี้ บทที่ 1 เป็นบทนำซึ่งกล่าวถึง ความเป็นมาและความสำคัญของปัญหา รวมถึงวัตถุประสงค์ของการวิจัย บทที่ 2 กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการวิจัยนี้ บทที่ 3 กล่าวถึงแนวทางการออกแบบการทดลองและแบบจำลองเบื้องต้น บทที่ 4 กล่าวถึงการทดลองในแต่ละขั้นตอนวิธีการแบ่งชิ้นส่วนย่อย บทที่ 5 กล่าวถึงวิเคราะห์และสรุปผลการทดลอง และบทที่ 6 กล่าวถึงบทสรุปของการวิจัย

1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตอบรับให้ตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่อง "Performance Analysis of File-Splitting Algorithm in Peer-to-Peer Network Backup Systems" โดยนางสาวจารินี ชยาภิรมย์ และอาจารย์ ดร. เกริก ภิรมย์โสภา ในงานประชุมวิชาการ "International Conference on Education Technology and Computer (ICETC 2010)" ณ เมืองเซี่ยงไฮ้ ประเทศสาธารณรัฐประชาชนจีน วันที่ 22-24 มิถุนายน พ.ศ. 2553

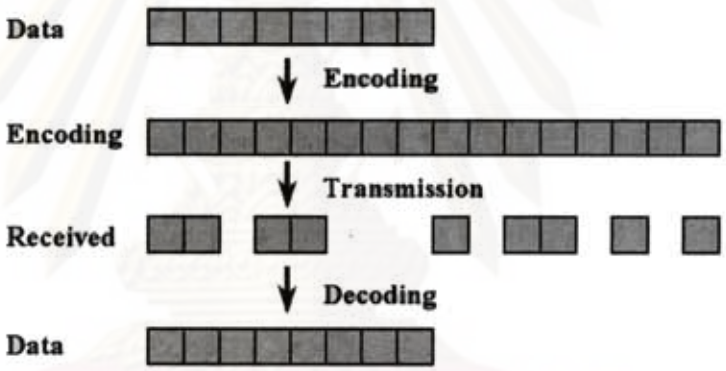
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 อีเรเซอร์โค้ด [2]

ในปัจจุบันมีผู้นำอีเรเซอร์โค้ดมาประยุกต์ใช้ในระบบสำรองข้อมูลแบบเพียร์ทูเพียร์เป็นจำนวนหลายระบบด้วยกัน แนวคิดของอีเรเซอร์โค้ดคือ การแบ่งไฟล์ขนาด f เป็นชิ้นส่วนย่อยจำนวน n ชิ้น โดยสามารถจำลองข้อมูลเดิมโดยใช้จำนวนชิ้นส่วนย่อย m ชิ้น (โดย m<n) อาทิเช่น ในกรณีที่สำรองข้อมูลลงบนจานบันทึกหลายจานที่เชื่อมต่อกัน จะยังคงสามารถกู้คืนข้อมูลเดิมได้อย่างสมบูรณ์แม้ว่าจานบันทึกจานใดจานหนึ่งไม่สามารถใช้งานได้ ดังรูปที่ 1



รูปที่ 1 แผนภาพแสดงการแบ่งชิ้นส่วนย่อยเพื่อทำการเข้ารหัสและถอดรหัสข้อมูลในการทำอีเรเซอร์โค้ด [3]

2.1.2 การวิเคราะห์สภาพพร้อมใช้งาน (Availability Analysis)

เป็นที่ทราบกันดีว่าการเพิ่มสภาพพร้อมใช้งานของไฟล์วิธีหนึ่งที่นิยมใช้ในระบบสำรองข้อมูลแบบเพียร์ทูเพียร์นั้นก็คือการทำอีเรเซอร์โค้ด ซึ่งจะทำการเข้ารหัสข้อมูลก่อนที่จะแบ่งไฟล์ ทำให้ได้จำนวนชิ้นส่วนย่อยหลังจากการแบ่งที่เพิ่มมากขึ้น ข้อดีของการใช้อีเรเซอร์โค้ดคือถึงแม้จะมีบางชิ้นส่วนหายไปจะยังคงสามารถกู้คืนข้อมูลเดิมกลับมาได้

เมื่อแบ่งไฟล์ออกเป็นชิ้นส่วนย่อยจำนวน b ชิ้นแล้วทำการเข้ารหัสข้อมูล จะทำให้ได้จำนวนชิ้นส่วนย่อยเพิ่มขึ้นเท่ากับ c ชิ้น โดยขนาดไฟล์หรือจำนวนชิ้นส่วนย่อยที่เพิ่มขึ้นกำหนดด้วยค่าปัจจัยการยืด (stretch factor, S) ซึ่งเป็นอัตราส่วนของจำนวนชิ้นส่วนย่อยหลังทำการเข้ารหัสแล้วต่อจำนวนชิ้นส่วนย่อยก่อนที่จะเข้ารหัส หรือมีค่าเท่ากับ c/b ค่าปัจจัยการยืดนี้เป็น

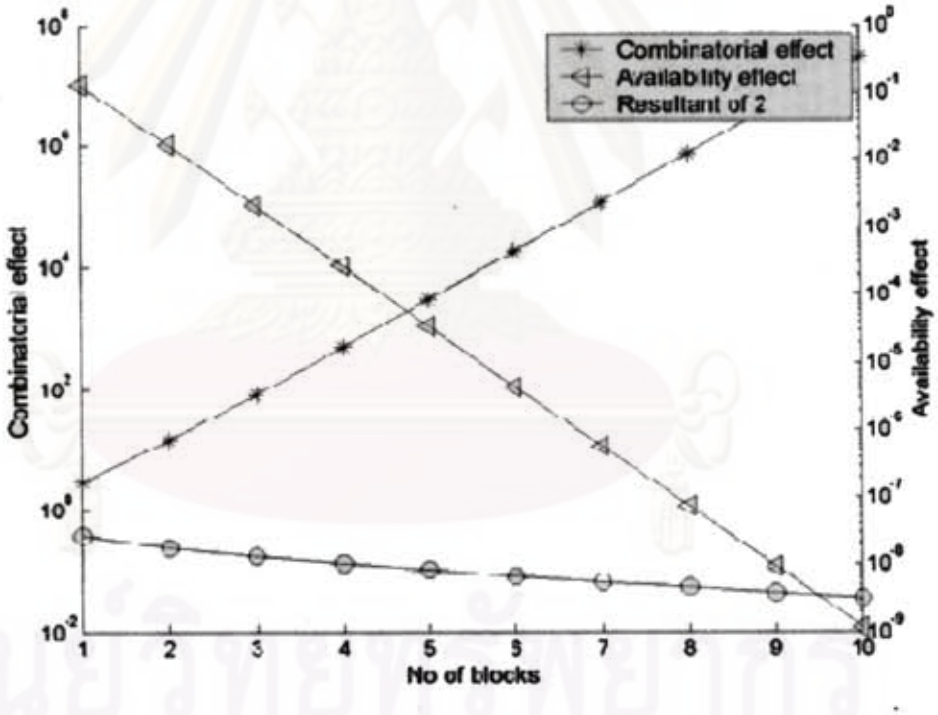
ปัจจัยหนึ่งที่ส่งผลกับสภาพพร้อมใช้งานของไฟล์ควบคู่กับอีกปัจจัยที่มีผลคือสภาพพร้อมใช้งานของเพียร์ (Peer Availability, μ) ซึ่งเป็นค่าที่แสดงความถี่หรือความบ่อยของเครื่องที่เข้า-ออกระบบ

2.1.2.1 การคำนวนค่าสภาพพร้อมใช้งานของไฟล์

การคำนวณค่าสภาพพร้อมใช้งานของไฟล์ (A) ของการทำอีเรเซอร์โค้ดขึ้นอยู่กับจำนวนชิ้นส่วนย่อย (b) โดยสามารถคำนวณได้โดยใช้ทฤษฎีพื้นฐานของความน่าจะเป็น (อ้างอิงจากงานวิจัย [4]) ดังสมการต่อไปนี้

$$A(S,b) = \sum_{i=b}^{Sb} \binom{Sb}{i} \mu^{i} (1-\mu)^{Sb-i} \quad (1)$$

จากสมการ (1) เมื่อจำนวนชิ้นส่วนย่อย (b) เพิ่มขึ้น สภาพพร้อมใช้งานของไฟล์ (A) ไม่ได้มีค่าสูงขึ้นด้วยเสมอไป เนื่องจากเมื่อชิ้นส่วนย่อย (b) เพิ่มขึ้นจะมีผลกับ Combinatorial Effect ($_{Sb}C_i$) และ Peer Availability Effect ($\mu^i (1-\mu)^{Sb-i}$) ในทางตรงกันข้ามกัน ดังรูปที่ 2



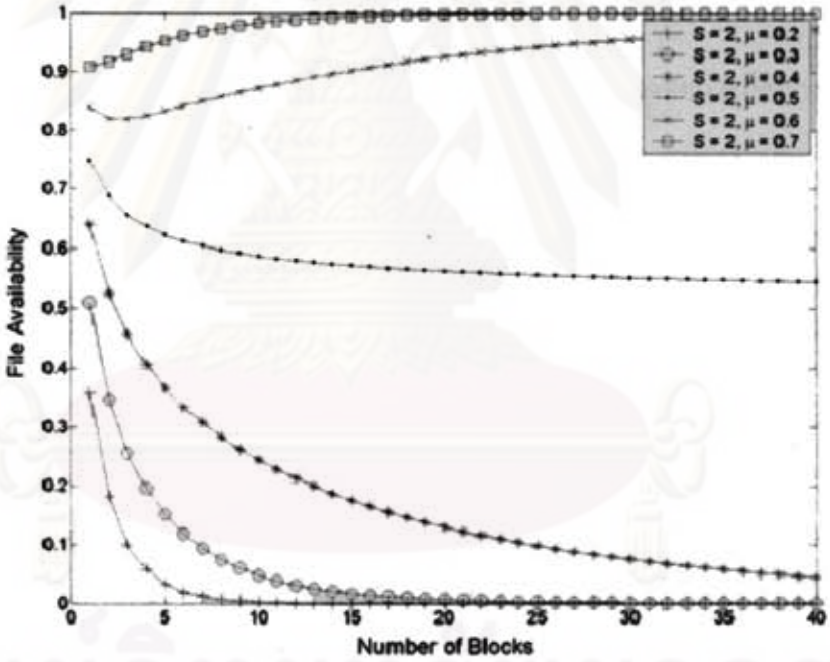
รูปที่ 2 ภาพแสดงการเปรียบเทียบระหว่าง Combinatorial Effect ($_{Sb}C_i$) และ Peer Availability Effect ($\mu^i (1-\mu)^{Sb-i}$) และผลจากการคูณกันของทั้งสองพจน์โดยเทียบกับจำนวนชิ้นส่วนย่อย

2.1.2.2 ผลกระทบของค่าปัจจัยการยึดและสภาพพร้อมใช้งานของเพียร์ต่อสภาพพร้อมใช้งานของไฟล์

การศึกษาสภาพพร้อมใช้งานของไฟล์สามารถคำนวนจากการใช้สมการ (1) โดยปรับค่าพารามิเตอร์ที่สำคัญคือ ค่าปัจจัยการยึด (S) และสภาพพร้อมใช้งานของเพียร์ (μ) โดยมีผลสรุปดังนี้

1. ผลกระทบของสภาพพร้อมใช้งานของเพียร์

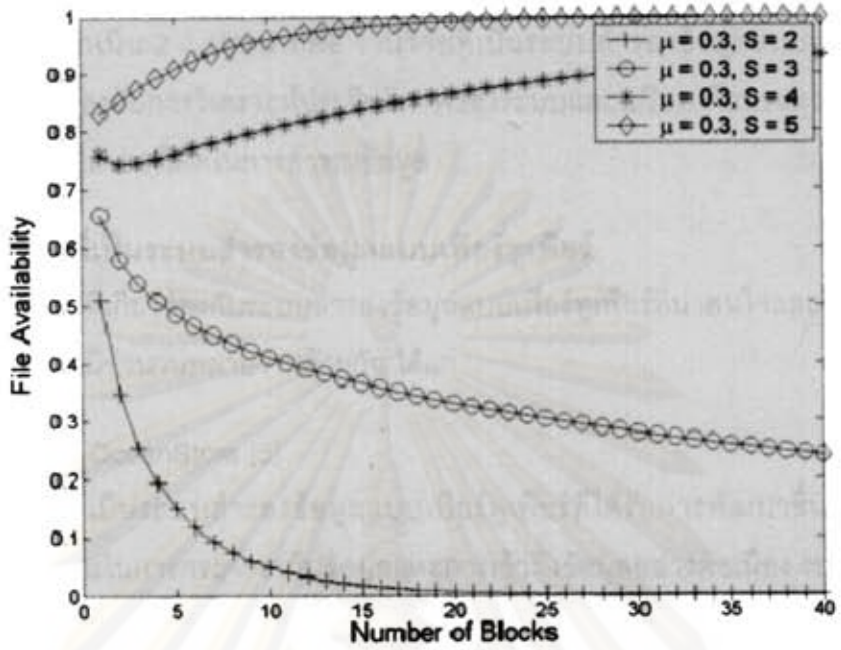
สำหรับค่าปัจจัยการยึดเดียวกัน การเพิ่มจำนวนชิ้นส่วนย่อยจะเพิ่มค่าสภาพพร้อมใช้งานของไฟล์เมื่อเพิ่มสภาพพร้อมใช้งานของเพียร์มากกว่าค่าหนึ่ง หากค่าสภาพพร้อมใช้งานของเพียร์น้อยกว่าค่านี้แล้วการเพิ่มจำนวนชิ้นส่วนย่อยจะให้ผลในทางตรงกันข้ามคือจะเป็นการลดค่าสภาพพร้อมใช้งานของไฟล์ ดังรูปที่ 3



รูปที่ 3 ผลกระทบของสภาพพร้อมใช้งานของไฟล์เมื่อค่าสภาพพร้อมใช้งานของเพียร์ต่างกัน

2. ผลกระทบของค่าปัจจัยการยึด

เมื่อค่าสภาพพร้อมใช้งานของเพียร์คงที่ การเพิ่มค่าปัจจัยการยึดมากขึ้นจะส่งผลให้การเพิ่มจำนวนชิ้นส่วนย่อยเป็นการเพิ่มค่าสภาพพร้อมใช้งานของไฟล์ ดังรูปที่ 4



รูปที่ 4 ผลกระทบของสภาพพร้อมใช้งานของไฟล์เมื่อค่าปัจจัยการยึดต่างกัน

จากผลสรุปข้างต้นแสดงให้เห็นว่าสภาพพร้อมใช้งานของไฟล์ (A) จะเพิ่มขึ้นแปรผันตามชิ้นส่วนย่อย (b) เมื่ออยู่ภายใต้สภาวะหรือปัจจัยที่เหมาะสม สำหรับค่าปัจจัยการยึด (S) คงที่ค่าหนึ่ง สภาพพร้อมใช้งานของเพียร์ (μ) ที่มากกว่าค่าขีดแบ่ง (Threshold) μ' จะให้ผลในทางที่ดีคือค่าสภาพพร้อมใช้งานของไฟล์ (A) จะมีค่าเพิ่มขึ้นเมื่อเพิ่มจำนวนชิ้นส่วนย่อย (b) ดังนั้นควรเลือกจำนวนชิ้นส่วนย่อยที่มากที่สุดเท่าที่เป็นไปได้เพื่อให้ได้สภาพพร้อมใช้งานของไฟล์สูง ในทางกลับกันหากค่าสภาพพร้อมใช้งานของเพียร์ (μ) น้อยกว่าค่าขีดแบ่ง μ' จะให้ผลในทางตรงกันข้ามคือเมื่อเพิ่มจำนวนชิ้นส่วนย่อย (b) จะทำให้สภาพพร้อมใช้งานของไฟล์ (A) ลดลง ดังนั้นจึงควรเลือกจำนวนชิ้นส่วนย่อย 1 ชิ้น หรือการทำสำเนาไฟล์ทั้งหมด (Whole-File Replication) จะให้สภาพพร้อมใช้งานของไฟล์สูงที่สุด

2.1.2.3 **จำนวนชิ้นส่วนย่อยที่เหมาะสม**

ปัจจัยในการเลือกจำนวนชิ้นส่วนย่อยที่เหมาะสมเมื่อสำรองข้อมูลโดยใช้อีเรเซอร์โค้ดจะพิจารณาจากค่า μ*S หากค่า μ*S > 1 การทำอีเรเซอร์โค้ดควรเพิ่มขนาดชิ้นส่วนย่อยให้มากที่สุดเท่าที่เป็นไปได้เพื่อเพิ่มสภาพพร้อมใช้งานของข้อมูลสูงที่สุด แต่หาก μ*S < 1 ไม่ควรเลือกการทำสำเนาด้วยอีเรเซอร์โค้ด และไม่ควรเพิ่มจำนวนชิ้นส่วนย่อยเพราะจะส่งผลให้ได้สภาพพร้อมใช้งานของไฟล์ลดลง ควรเลือกจำนวนชิ้นส่วนย่อยจำนวนน้อยหรือเลือกการทำสำเนาไฟล์ทั้งหมด (Whole-File Replication) จะให้ผลที่ดีกว่า

2.2 งานวิจัยที่เกี่ยวข้อง

แบ่งออกเป็น 2 ประเภทคือ งานวิจัยที่เป็นระบบสำรองข้อมูลแบบเพียร์ทูเพียร์และงานวิจัยที่เกี่ยวข้องกับการวิเคราะห์ประสิทธิภาพของระบบและเปรียบเทียบระหว่างการทำสำเนาข้อมูลกับการใช้อีเรเซอร์โค้ดในการสำรองข้อมูล

2.2.1 งานวิจัยที่เป็นระบบสำรองข้อมูลแบบเพียร์ทูเพียร์

งานวิจัยที่เกี่ยวข้องกับระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ที่น่าสนใจและได้รับการพัฒนาจนถึงปัจจุบันนั้นมีจำนวนหลายงานด้วยกัน ได้แก่

2.2.1.1 OceanStore [5]

เป็นระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ที่ได้รับการพัฒนาขึ้นในปี 2000 มีโครงสร้างพื้นฐานในการกระจายเก็บข้อมูลและการเข้าถึงข้อมูลอย่างต่อเนื่อง ระบบจะทำสำเนาข้อมูล (Data Replication) โดยใช้อีเรเซอร์โค้ด (Reed-Solomon Code และ Tornado Code) และทำการเข้ารหัสเพื่อป้องกันการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต ในการเพิ่มประสิทธิภาพของระบบ ข้อมูลจะถูกแคช (Cache) เพื่อสำรองข้อมูลในระบบชั่วคราว นอกจากนี้ระบบมีส่วนที่ทำหน้าที่เฝ้าสังเกต (Monitor) รูปแบบการใช้งานภายในระบบเพื่อป้องกันการโจมตี การเฝ้าสังเกตนี้ได้รวมถึงการกระตุ้นและตรวจสอบการเคลื่อนไหวของข้อมูลเพื่อเพิ่มประสิทธิภาพของระบบอีกทางหนึ่งด้วย

2.2.1.2 pStore [6]

เป็นโครงงานวิจัยที่ถูกพัฒนาในปี 2001 โดยสถาบัน MIT ระบบ pStore เป็นระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ที่ใช้โครงสร้างของบล็อกข้อมูล (File Block) และรายการบล็อกข้อมูล (File Block List) ในการเก็บข้อมูลที่เข้ารหัส (Encryption)แล้ว เนื่องจากยังขาดประสิทธิภาพด้านความเร็วและมีอัตราการถ่ายโอนข้อมูลค่อนข้างสูง จึงมีผู้นำมาปรับปรุงและใช้เป็นข้อมูลอ้างอิงในการพัฒนาต่อในหลายงานวิจัย

2.2.1.3 PeerStore [7]

เป็นระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ที่ถูกเสนอขึ้นในปี 2004 โดยได้รวมจุดเด่นและข้อดีของหลายระบบเข้าด้วยกัน PeerStore สามารถรองรับการทำงานกับเครือข่ายขนาดใหญ่ที่ไม่เสถียร (Unstable Nework) เช่น อินเทอร์เน็ตได้ ระบบมีพื้นฐานการทำงานและโครงสร้างในการเก็บข้อมูลใกล้เคียงกับ pStore แต่สิ่งที่แตกต่างและโดดเด่นจากระบบอื่นคือการทำงานที่แบ่งเป็น 2 ชั้น (Layer) คือ ชั้นสำหรับเก็บข้อมูลของข้อมูล (Metadata Layer) โดยใช้ตารางแฮชแบบกระจาย (Distributed Hash Table : DHT) สำหรับจัดการข้อมูลของข้อมูล

(Metadata) ซึ่งแยกออกจากชั้นหน่วยเก็บ (Storage Layer) ซึ่งใช้ Symmetric Trading Scheme สำหรับควบคุมการเก็บข้อมูลจริงในระบบ การเก็บข้อมูลใช้วิธีการเก็บข้อมูลอย่างปลอดภัย (Safekeeping) และการกระจายข้อมูลแบบเท่าเทียมกัน (Fair Contribution) เพื่อรับประกันความถูกต้องและความเชื่อถือได้ของข้อมูล (Data Reliability) ไปพร้อมกัน ส่งผลให้ปริมาณของข้อมูลที่เคลื่อนย้ายในระบบขณะทำการสำรองข้อมูลน้อยกว่าระบบ pStore มาก รวมทั้งมีค่าใช้จ่ายในการบำรุงรักษา (Maintenance Cost) ที่น้อยกว่า

2.2.1.4 iDIBS [8]

เป็นระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ที่ได้รับการพัฒนาในปี 2006 เป็นการพัฒนาและปรับปรุงระบบสำรองข้อมูลเดิม (ระบบ Distributed Internet Backup System : DIBS) ในหลายด้าน เช่น เพิ่มความเชื่อถือได้ของข้อมูลโดยเพิ่มประสิทธิภาพของกลไกการส่งผ่านกลุ่มข้อมูล (Packet Transmission Mechanism) ในการแบ่งข้อมูลเพื่อกระจายเก็บลงบนเครื่องภายในระบบอาศัยอีเรเซอร์โค้ดโดยเลือกใช้ Luby Transform Code แทน Reed-Solomon Code เพื่อเพิ่มความเร็วในการคำนวณและรองรับความสามารถในการปรับขนาดได้ (Scalability) ของข้อมูลขนาดใหญ่

2.2.1.5 vanDisk [9]

เป็นระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ซึ่งได้รับการพัฒนาถัดมาในปี 2007 ระบบทำงานโดยอาศัยไดร์ฟเสมือน (Virtual Drives) ในการสร้างสำเนาของข้อมูลซึ่งทำการเข้ารหัสด้วยอีเรเซอร์โค้ด (LDPC Code) ระบบใช้การเฝ้าสังเกตและตรวจสอบสภาพพร้อมใช้งานของเครื่องในระบบเพื่อเพิ่มความเชื่อถือได้ของข้อมูล (Data Reliability) โครงสร้างของระบบเป็นได้ทั้งแบบรวมศูนย์ (Centralized) หรือกระจาย (Decentralized) ที่มีผู้นำกลุ่ม (Leader) เป็นเครื่องควบคุมหรือจัดการเครื่องภายในระบบ ซึ่งมีข้อเสียในด้านความยุ่งยากและซับซ้อน และเพิ่มค่าใช้จ่ายอื่น (Overhead) ในการติดต่อกันภายในระบบ

2.2.2 งานวิจัยที่เกี่ยวข้องกับการวิเคราะห์ประสิทธิภาพของระบบและเปรียบเทียบระหว่างการทำสำเนาข้อมูลกับการใช้อีเรเซอร์โค้ดในการสำรองข้อมูล

2.2.2.1 Replication Strategies for Highly Available Peer-to-peer Storage Systems [10]

ศึกษากลยุทธ์ในการสำรองข้อมูลบนระบบสำรองข้อมูลแบบเพียร์ทูเพียร์เพื่อให้มีสภาพพร้อมใช้งานสูงโดยมีค่าใช้จ่ายอื่น (Overhead) ที่เหมาะสม งานวิจัยได้เสนอแบบจำลองเชิงวิเคราะห์ (Analytical Model) เพื่อวิเคราะห์ประสิทธิภาพโดยเปรียบเทียบระหว่างการทำสำเนา

ไฟล์ทั้งหมด (Whole-File Replication) กับอีเรเซอร์โค้ดในด้านสภาพพร้อมใช้งานของไฟล์ (File Availability), สภาพพร้อมใช้งานของระบบ (System-Level Availability) และความต้องการด้านหน่วยเก็บข้อมูล (Storage Requirements)

2.2.2.2 Erasure Code Replication Revisited [4]

ศึกษาระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ ทำการวิเคราะห์และเสนอแนวทางเลือกเพื่อตัดสินใจเลือกวิธีในการสำรองข้อมูลซึ่งใช้อีเรเซอร์โค้ดโดยเปรียบเทียบกับการทำสำเนาไฟล์ทั้งหมด (Whole-File Replication) เพื่อให้มีความเหมาะสมกับสถานการณ์และสภาพแวดล้อม โดยพิจารณาจากสภาพพร้อมใช้งานของข้อมูล

2.2.2.3 A Performance Evaluation and Examination of Open-Source Erasure Coding Libraries For Storage [11]

เปรียบเทียบความเร็วในการเข้ารหัสและถอดรหัส (Encoding/Decoding Speed) ระหว่าง Open-Source Erasure Coding Libraries 5 ประเภท คือ Classic Reed-Solomon Code, Cauchy Reed-Solomon Code, EVENODD, Row Diagonal Parity (RDP) และ Minimal Density RAID-6 Code พร้อมทั้งเสนอแนวทางการเลือกพารามิเตอร์ที่เหมาะสม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
แนวทางการออกแบบการทดลองและแบบจำลองเบื้องต้น (Initial Models)

ในบทนี้แบ่งออกเป็น 3 ส่วนคือ เครื่องมือที่ใช้ในการวิจัย, การออกแบบการทดลองและแบบจำลองเบื้องต้น ในส่วนแรกกล่าวถึงรายละเอียดของเครื่องมือที่ใช้งานวิจัยนี้ ส่วนที่สองจะกล่าวถึงรูปแบบและรายละเอียดของการทดลองเพื่อวัดและศึกษาเปรียบเทียบประสิทธิภาพของระบบสำหรับแต่ละขั้นตอนวิธีในการแบ่งชิ้นส่วนย่อย รวมถึงพิจารณาปัจจัยที่มีอิทธิพลและพารามิเตอร์ (Parameter) ที่เกี่ยวข้อง ในส่วนที่สามได้เสนอแบบจำลองเบื้องต้น (Initial Models) เพื่อแสดงแนวทางและภาพรวมของการทดลองในบทถัดไป

3.1 เครื่องมือที่ใช้ในการวิจัย

เครื่องมือที่ใช้ในการวิจัยนี้ประกอบด้วย

3.1.1 NetBeans 5.5-6.0

สำหรับพัฒนาโปรแกรมภาษาจาวา (JAVA)

3.1.2 Cygwin

ซอฟต์แวร์จำลองการทำงานบนระบบปฏิบัติการยูนิกซ์ (Unix Operating System) สำหรับบางส่วนของระบบที่ใช้คำสั่งในโปรแกรมซึ่งจำเป็นต้องใช้งานบนระบบปฏิบัติการยูนิกซ์

3.2 การออกแบบการทดลอง

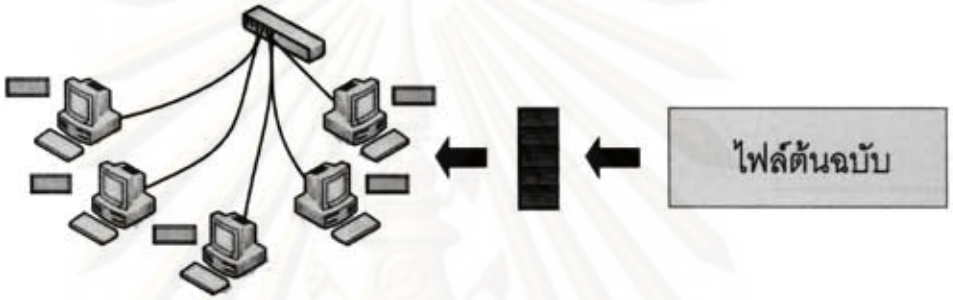
งานวิจัยนี้จะพิจารณาประสิทธิภาพของระบบในด้านความเร็วที่ใช้ในการสำรองข้อมูล (Backup Speed) และสภาพพร้อมใช้งานของไฟล์ (File Availability) เมื่อเลือกใช้ขั้นตอนวิธีในการแบ่งชิ้นส่วนย่อยแตกต่างกัน ความเร็วในการสำรองข้อมูลจะพิจารณาเวลาที่ใช้ในการสำรองข้อมูล (Backup Time) แต่ละครั้ง (ในหน่วยวินาที) โดยนับตั้งแต่ผู้ใช้เริ่มสำรองข้อมูล (หลังจากผู้ใช้เลือกไฟล์และสั่งงาน) จนกระทั่งชิ้นส่วนย่อยทุกชิ้นกระจายเก็บลงบนเครื่องในระบบเรียบร้อย การออกแบบการทดลองแบ่งออกเป็น 3 ขั้นตอน ดังนี้

3.2.1 การกำหนดขั้นตอนวิธีในการแบ่งชิ้นส่วนย่อย

งานวิจัยนี้ได้เลือกขั้นตอนวิธีในการแบ่งชิ้นส่วนย่อย 3 รูปแบบ ดังนี้

3.2.1.1 **การแบ่งชิ้นส่วนย่อยตามจำนวนเครื่องในระบบ**

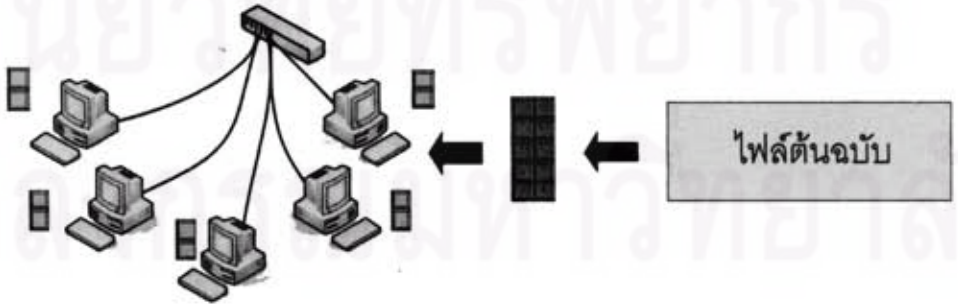
วิธีนี้ระบบจะทำการแบ่งไฟล์โดยพิจารณาจากจำนวนเครื่องภายในระบบที่สามารถใช้งานได้ในขณะนั้น โดยแบ่งข้อมูลเป็นชิ้นส่วนย่อยที่มีขนาดเท่ากัน แล้วทำการส่งชิ้นส่วนย่อยแต่ละชิ้นไปยังเครื่องภายในระบบแต่ละเครื่อง ดังรูปที่ 5 ซึ่งวิธีนี้มีข้อดีคือทุกเครื่องในระบบได้รับชิ้นส่วนย่อยที่มีขนาดเท่ากันในจำนวนที่ใกล้เคียงกัน อย่างไรก็ตามวิธีนี้ไม่เหมาะสมในกรณีที่ทำการแบ่งชิ้นส่วนย่อยของไฟล์ที่มีขนาดเล็ก เนื่องจากทำให้ชิ้นส่วนย่อยแต่ละชิ้นมีขนาดเล็กมาก



รูปที่ 5 ภาพแสดงการแบ่งชิ้นส่วนย่อยตามจำนวนเครื่องในระบบ

3.2.1.2 **การแบ่งชิ้นส่วนย่อยโดยกำหนดขนาดของชิ้นส่วนย่อยคงที่**

วิธีนี้ระบบจะกำหนดขนาดของชิ้นส่วนย่อยที่คงที่ตามระดับที่ได้แบ่งไว้โดยแปรผันตามขนาดของข้อมูล หลังจากนั้นระบบจะทำการแบ่งไฟล์ออกเป็นชิ้นส่วนย่อยที่มีขนาดเท่ากันและมีขนาดคงที่ แล้วทำการกระจายไปยังเครื่องภายในระบบ ดังรูปที่ 6 จะเห็นได้ว่าวิธีนี้มีความสะดวกในการแบ่งข้อมูลเนื่องจากขนาดของชิ้นส่วนย่อยที่คงที่ และสามารถกำหนดขนาดของชิ้นส่วนย่อยได้ตามต้องการ แต่วิธีนี้ไม่เหมาะสมในกรณีที่ไฟล์มีขนาดใหญ่ เนื่องจากจะทำให้มีจำนวนชิ้นส่วนย่อยจำนวนมาก จึงจำเป็นต้องทำการแบ่งระดับของขนาดชิ้นส่วนย่อยที่เหมาะสมกับขนาดของข้อมูล



รูปที่ 6 ภาพแสดงการแบ่งชิ้นส่วนย่อยโดยกำหนดขนาดของชิ้นส่วนย่อยคงที่

3.2.1.3 **การแบ่งชิ้นส่วนย่อยโดยใช้อีเรเซอร์โค้ด**

วิธีนี้มีข้อดีในด้านการเพิ่มสภาพพร้อมใช้งานของข้อมูลในระบบ ในกรณีที่ชิ้นส่วนย่อยที่แบ่งไว้สูญหาย ระบบยังคงสามารถกู้คืนข้อมูลเดิมได้อย่างสมบูรณ์และถูกต้อง เนื่องจากการเข้ารหัสด้วยอีเรเซอร์โค้ดสามารถกู้คืนไฟล์ต้นฉบับได้แม้จะได้รับจำนวนชิ้นส่วนย่อยไม่ครบถ้วนสมบูรณ์ ข้อเสียในการเลือกใช้อีเรเซอร์โค้ดคือการเข้ารหัสในขณะที่ทำการแบ่งข้อมูลส่งผลต่อเวลาในการทำงานของระบบที่เพิ่มมากขึ้น แผนภาพแสดงการทำงานด้วยวิธีอีเรเซอร์โค้ดแสดงดังรูปที่ 7 นอกจากนี้ สภาพพร้อมใช้งานของข้อมูลของระบบจะเพิ่มมากขึ้นหรือลดลงแปรผันตามจำนวนชิ้นส่วนย่อย นั่นคือจำนวนชิ้นส่วนย่อยที่แบ่งในขณะแบ่งไฟล์มีผลต่อสภาพพร้อมใช้งานของไฟล์ด้วย

รูปที่ 7 ภาพแสดงการแบ่งชิ้นส่วนย่อยโดยใช้อีเรเซอร์โค้ด

3.2.2 **การพิจารณาปัจจัยที่ส่งผลกับประสิทธิภาพของระบบ**

ในส่วนนี้จะกล่าวถึงปัจจัยที่มีส่วนสำคัญกับประสิทธิภาพของระบบในแต่ละด้าน ดังนี้

3.2.2.1 **เวลาที่ใช้ในการสำรองข้อมูล** (Backup Time)

ปัจจัยที่ส่งผลต่อเวลาในการสำรองข้อมูลในแต่ละขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยได้แก่

1. จำนวนเครื่องภายในระบบหรือเพียร์ (Peer)
2. ขนาดไฟล์ (File Size)
3. ความเร็วและโครงสร้างของระบบเครือข่าย

3.2.2.2 **สภาพพร้อมใช้งานของไฟล์** (File Availability)

ปัจจัยที่ส่งผลต่อสภาพพร้อมใช้งานของไฟล์มีดังนี้

1. จำนวนเครื่องภายในระบบหรือเพียร์ (Peer)
2. สภาพพร้อมใช้งานของเพียร์ (Peer Availability)
3. จำนวนชิ้นส่วนย่อย (Block/Chunk)
4. ค่าใช้จ่ายภายในระบบ (Stretch Factor, Storage Overhead)

3.2.3 แนวทางการออกแบบการทดลองเพื่อวัดและศึกษาเปรียบเทียบประสิทธิภาพของระบบ

จากสมการ (1) ในหัวข้อ 2.1.2.1 หน้า 6 แสดงการคำนวณสภาพพร้อมใช้งานใช้หลักการทางทฤษฎีความน่าจะเป็น แสดงให้เห็นว่าประสิทธิภาพของระบบในส่วนสภาพพร้อมใช้งานของไฟล์ (File Availability) สามารถประเมินได้โดยไม่ต้องอาศัยการทดลอง ดังนั้นการทดลองจะพิจารณาประสิทธิภาพเฉพาะในด้านความเร็วของแต่ละขั้นตอนวิธีโดยทำการศึกษาเปรียบเทียบและวัดเวลาที่ใช้ในการสำรองข้อมูลซึ่งปรับค่าพารามิเตอร์ตามปัจจัยที่กำหนดในหัวข้อ 3.2.2.1 อย่างไรก็ตามแต่ละขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยยังมีพารามิเตอร์อื่นเพิ่มเติมที่แตกต่างกันไปนอกเหนือจากที่กล่าวข้างต้น โดยจะพิจารณาและกล่าวถึงรายละเอียดในหัวข้อ 3.3 แบบจำลองเบื้องต้น (Initial Models) ต่อไป

การทดลองได้ทดสอบกับระบบสำรองข้อมูลแบบเพียร์ทูเพียร์บนระบบปฏิบัติการวินโดวส์ (Windows) เครื่องภายในระบบ (เพียร์) เชื่อมต่อกันด้วยเครือข่าย LAN และติดต่อกันโดยใช้ mDNS (Multicast DNS) เพื่อแพร่สัญญาณ (Broadcast) เลขที่อยู่ (IP Address) ของเพียร์ขณะเข้าสู่ระบบ

3.2.4 แนวทางการวิเคราะห์ประสิทธิภาพของระบบ

ขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยที่กล่าวข้างต้นใช้เวลาในการสำรองข้อมูลและมีสภาพพร้อมใช้งานของไฟล์แตกต่างกันขึ้นอยู่กับพารามิเตอร์แต่ละแบบ การสำรองข้อมูลโดยส่วนใหญ่หากต้องการสภาพพร้อมใช้งานของไฟล์สูงมักส่งผลกับเวลาที่ใช้ เช่น วิธีที่ใช้อีเรเซอร์โค้ดจะใช้เวลามากกว่าวิธีอื่นเพราะต้องเสียเวลาการเข้ารหัสก่อนแบ่งข้อมูล งานวิจัยนี้ทำการวิเคราะห์ประสิทธิภาพในการสำรองข้อมูลโดยพิจารณาทั้งสองด้านควบคู่กัน รวมถึงศึกษาปัจจัยที่ส่งผลกระทบและการปรับค่าพารามิเตอร์ในแต่ละขั้นตอนวิธีเพื่อให้มีความเหมาะสมในการสำรองข้อมูล

นอกจากนี้ แต่ละขั้นตอนวิธีมีค่าใช้จ่ายภายในระบบ (cost) แตกต่างกัน การวัดประสิทธิภาพในด้านสภาพพร้อมใช้งานของไฟล์จำเป็นต้องนำค่าใช้จ่ายเหล่านั้นมาพิจารณาด้วย หากพิจารณาวิธีแบ่งโดยใช้อีเรเซอร์โค้ดพบว่าในบางกรณีสภาพพร้อมใช้งานของไฟล์ในระบบเพิ่มมากขึ้นเมื่อแบ่งชิ้นส่วนย่อยจำนวนมากขึ้น อย่างไรก็ตามการแบ่งชิ้นส่วนย่อยจำนวนมากอาจส่งผลกระทบในด้านอื่น เช่น ความซับซ้อนในการแบ่งข้อมูล, เวลาในการแบ่งข้อมูลและอัตราการส่งผ่านข้อมูลเพิ่มขึ้น เป็นต้น ดังนั้นการวิเคราะห์ประสิทธิภาพของวิธีที่ใช้อีเรเซอร์โค้ดจะครอบคลุมถึงจำนวนชิ้นส่วนย่อยที่เหมาะสมและคุ้มค่ากับค่าใช้จ่ายที่ต้องสูญเสียภายในระบบด้วย

3.3 แบบจำลองเบื้องต้น (Initial Models)

ขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยแต่ละวิธีมีปัจจัยที่ส่งผลกับเวลาในการสำรองข้อมูลแตกต่างกัน ในหัวข้อนี้ได้เสนอแบบจำลองเบื้องต้นเพื่อคาดการณ์เวลาที่ใช้ในการสำรองข้อมูลซึ่งแปรผันตามพารามิเตอร์ที่ต่างกันในแต่ละวิธี จากหัวข้อ 3.2.2 พบว่าขนาดไฟล์ (File Size) เป็นพารามิเตอร์ที่สำคัญและส่งผลกับเวลาที่ใช้ในทุกขั้นตอนวิธี สำหรับการแบ่งชิ้นส่วนย่อยวิธีแรกระบบจะแบ่งชิ้นส่วนย่อยเป็นจำนวนเท่ากับจำนวนเครื่องในระบบ เวลาที่ใช้จึงแปรผันตามจำนวนเครื่องภายในระบบ (n) และขนาดไฟล์ (sz) สำหรับการแบ่งชิ้นส่วนย่อยวิธีที่สอง ระบบแบ่งข้อมูลเป็นชิ้นส่วนย่อยที่มีขนาดเท่ากันตามที่กำหนด ขนาดของชิ้นส่วนย่อย (bs) เป็นปัจจัยสำคัญที่มีผลกระทบในการสำรองข้อมูลเช่นเดียวกับจำนวนเครื่องในระบบ นั่นคือเวลาที่ใช้ในการสำรองข้อมูลเมื่อแบ่งชิ้นส่วนย่อยด้วยวิธีนี้แปรผันตามขนาดไฟล์, จำนวนเครื่องในระบบ และขนาดของชิ้นส่วนย่อย สำหรับการแบ่งชิ้นส่วนย่อยวิธีที่สามซึ่งใช้อีเรเซอร์โค้ด นอกจากขนาดไฟล์และจำนวนเครื่องในระบบจะมีผลกับเวลาที่ใช้ จำนวนชิ้นส่วนย่อย (b) และค่าปัจจัยการยืด (S) ส่งผลกับเวลาที่ใช้ในการสำรองข้อมูลเช่นกัน ดังนั้นเวลาที่ใช้การสำรองข้อมูลจึงเป็นฟังก์ชันที่ขึ้นอยู่กับพารามิเตอร์ดังกล่าว โดยสามารถร่างแบบจำลองเบื้องต้นของเวลาที่ใช้ในการสำรองข้อมูล (T) ในแต่ละวิธีได้ดังสมการดังนี้

$$T = \begin{cases} T_1(sz, n), & \text{วิธีที่ } 1 \\ T_2(sz, n, bs), & \text{วิธีที่ } 2 \\ T_3(sz, n, b, S), & \text{วิธีที่ } 3 \end{cases} \quad (2)$$

สำหรับการศึกษาทดลองเพื่อวัดเวลาที่ใช้ในการสำรองข้อมูลในแต่ละขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยโดยปรับค่าพารามิเตอร์ตามสมการของแบบจำลองข้างต้นจะกล่าวถึงในบทถัดไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

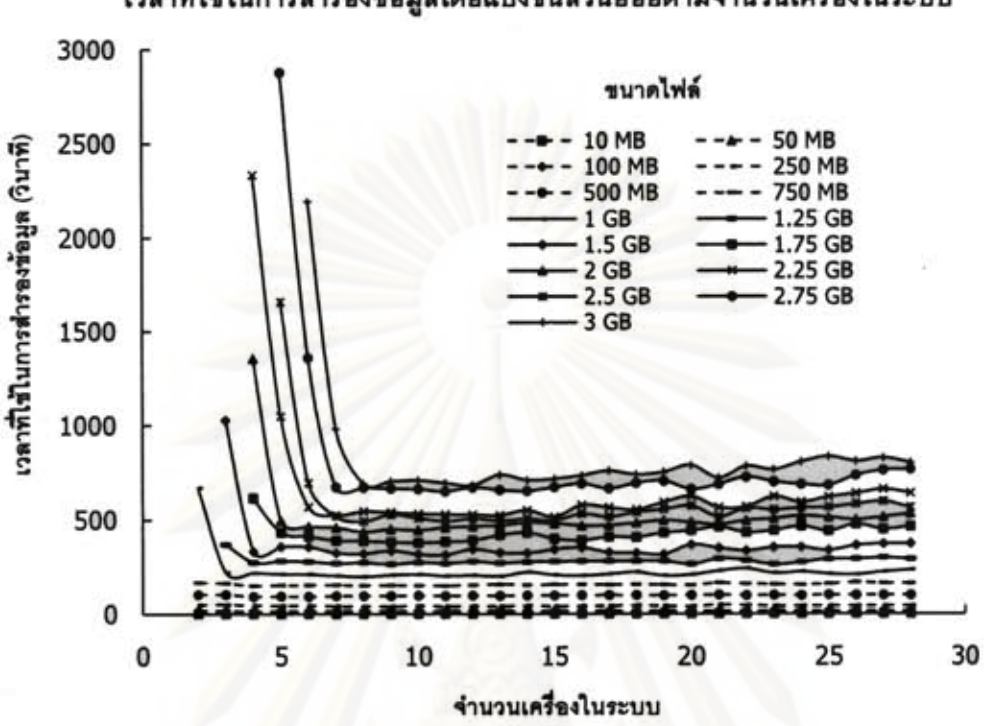
บทที่ 4
การทดลองในแต่ละขั้นตอนวิธีการแบ่งชิ้นส่วนย่อย

แต่ละขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยส่งผลกับเวลาในการสำรองข้อมูลที่ต่างกัน ในบทนี้จะกล่าวถึงการทดลองเพื่อวัดเวลาที่ใช้ในการสำรองข้อมูลโดยปรับค่าพารามิเตอร์ของแต่ละวิธีการแบ่งชิ้นส่วนย่อยทั้งสามวิธีตามแบบจำลองเบื้องต้นในหัวข้อ 3.3 สำหรับการแบ่งชิ้นส่วนย่อยโดยใช้อีเรเซอร์โค้ดเปรียบเทียบระหว่าง Reed-Solomon Code (RS Code) และ Luby Transform Code (LT Code) การทดลองทดสอบกับระบบสำรองข้อมูลแบบเพียร์ทูเพียร์ซึ่งเป็นซอฟต์แวร์ที่พัฒนาด้วยภาษาจาวา (JAVA) บนเครื่องคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการวินโดวส์ (Windows XP Professional SP2, Intel Pentium 4 CPU 3.60GHz, 1 GB of RAM) เชื่อมต่อกันในเครือข่าย LAN จำนวน 30 เครื่อง การทดลองเลือกใช้ขนาดไฟล์ตั้งแต่ 10 MB จนถึง 3 GB

4.1 วิธีที่ 1 : แบ่งชิ้นส่วนย่อยตามจำนวนเครื่องในระบบ

สำหรับขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยวิธีนี้ ระบบจะแบ่งไฟล์เป็นจำนวน n ชิ้น (เมื่อมีจำนวนเครื่องในระบบ n เครื่อง) แต่ละชิ้นส่วนที่มีขนาดเท่ากันจะถูกกระจายและเก็บลงบนเครื่องโดยแต่ละเครื่องจะได้รับชิ้นส่วนเพียงหนึ่งชิ้น จากแบบจำลองเบื้องต้นในหัวข้อ 3.3 แสดงให้เห็นถึงพารามิเตอร์ที่ส่งผลกับเวลาที่ใช้ในวิธีนี้อันได้แก่ ขนาดไฟล์ (sz) และจำนวนเครื่องในระบบ (n) การทดลองจึงทำการวัดเวลาที่ใช้ในการสำรองข้อมูลกับขนาดไฟล์ต่างกันตั้งแต่ 10 MB จนถึง 3 GB บนเครื่องตั้งแต่ 2 ถึง 30 เครื่อง ผลการทดลองที่ได้แสดงดังรูปที่ 8

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 8 กราฟแสดงเวลาที่ใช้ในการสำรองข้อมูลลงบนเครื่องตั้งแต่ 2 ถึง 30 เครื่องและเลือกใช้ขนาดไฟล์ตั้งแต่ 10 MB จนถึง 3 GB

ในอุดมคติแล้วเมื่อจำนวนเครื่องเพิ่มมากขึ้น เวลาที่ใช้ย่อมลดลง อย่างไรก็ตาม รูปที่ 8 แสดงให้เห็นว่าเวลาที่ใช้ในการสำรองข้อมูลไม่ได้ลดลงเสมอไปเมื่อใช้จำนวนเครื่องในระบบมากขึ้น เวลาที่ใช้ในการสำรองข้อมูลสำหรับแต่ละขนาดไฟล์ลดลงอย่างรวดเร็วในช่วงแรก (เมื่อใช้จำนวนเครื่องน้อย) แต่เมื่อเพิ่มจำนวนเครื่องไปจนถึงจุดหนึ่งจะพบว่าเวลาค่อนข้างคงที่ เห็นได้ชัดว่ากราฟของเวลาถูกแบ่งเป็นสองบริเวณคือ บริเวณที่เมื่อเพิ่มจำนวนเครื่องแล้วจะส่งผลให้เวลาที่ใช้ลดลงและบริเวณที่เมื่อเพิ่มจำนวนเครื่องแล้วเวลาที่ใช้ค่อนข้างคงที่ สังเกตได้ว่ากราฟของเวลาสำหรับแต่ละขนาดไฟล์จะมีจุดแบ่ง (Cut-Off Point) ซึ่งเมื่อจำนวนเครื่องในระบบมากกว่าจุดนี้จะทำให้การเพิ่มจำนวนเครื่องไม่เป็นประโยชน์ที่จะทำให้เวลาลดลง

พิจารณาบริเวณแรกพบว่าในแต่ละขนาดไฟล์เมื่อเพิ่มจำนวนเครื่องในระบบ เวลาที่ได้ลดลงอย่างรวดเร็วโดยมีแนวโน้มการลดลงแบบ Exponential สมการสำหรับประมาณเวลาที่ใช้ในการสำรองข้อมูล (T_1) ในรูปของฟังก์ชันที่แปรผันตามจำนวนเครื่องในระบบ (n) และขนาดไฟล์ (sz) สามารถแสดงได้ดังนี้

$$T_1(sz, n) = k_1 a^{(n+k_2 sz+k_3)} + k_4 sz + k_5 \quad (3)$$

ในพจน์แรกของสมการแสดงถึงเวลาที่ลดลงอย่างรวดเร็วแบบ Exponential โดยแปรผันตามจำนวนเครื่องที่เพิ่มขึ้น เมื่อขนาดไฟล์เพิ่มขึ้นส่งผลกับเวลาที่ใช้ในการสำรองข้อมูลที่มากขึ้น นั่นคือเวลาที่ใช้แปรผันตรงกับขนาดไฟล์ดังแสดงในพจน์ k_4sz สำหรับพจน์ k_2sz แสดงถึงการเลื่อนของกราฟไปทางด้านขวาเมื่อขนาดของข้อมูลเพิ่มขึ้น จากสมการ (3) สามารถประมาณค่าสัมประสิทธิ์ a, k_1, k_2, k_3, k_4 and k_5 จากระเบียบวิธีกำลังสองน้อยที่สุด (Least Square Method) ได้ดังนี้

$a = 0.298$

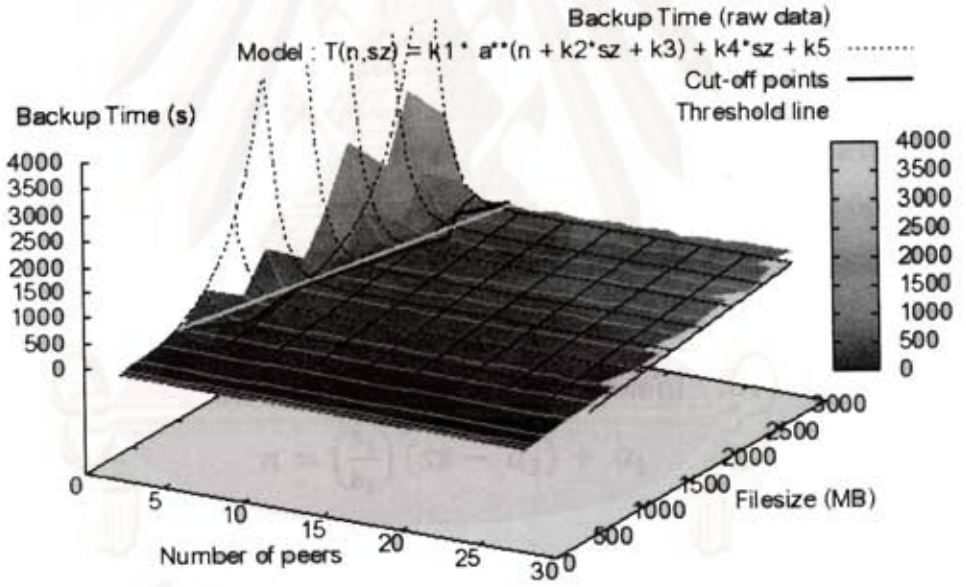
$k_1 = 0.881$

$k_2 = -0.002$

$k_3 = -4.980$

$k_4 = 0.263$

$k_5 = -192.488$



รูปที่ 9 กราฟแสดงพื้นผิวของสมการที่ใช้ประมาณเวลาในการสำรองข้อมูล $T_1(sz,n)$ เทียบกับจำนวนเครื่องในระบบและขนาดไฟล์

ผลการทดลองวัดเวลาที่ใช้ในการสำรองข้อมูลเทียบกับจำนวนเครื่องในระบบและขนาดไฟล์สามารถแสดงดังพื้นผิวในกราฟสามมิติในรูปที่ 9 เส้นประสีดำแสดงถึงแบบจำลองของสมการที่ใช้ประมาณเวลาในการสำรองข้อมูล $T_1(sz,n)$ ในบริเวณแรกซึ่งลดลงอย่างรวดเร็วเมื่อเพิ่มจำนวนเครื่องในระบบ เส้นสีแดงเป็นเส้นเชื่อมจุดแบ่ง (Cut-off point) ของกราฟเวลาที่ใช้ในแต่ละขนาดไฟล์ สมการของเส้นขีดแบ่ง (Threshold Line) ในรูปของสมการเวกเตอร์ (Vector Equation) ซึ่ง

เป็นสมการเส้นตรงซึ่งประมาณค่าจุดแบ่งของกราฟเวลาที่ใช้ในแต่ละขนาดไฟล์ แสดงด้วยเส้นสีฟ้าของรูปที่ 9 มีสมการดังนี้

$$\begin{pmatrix} n \\ sz \\ T_1 \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + t \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \quad (4)$$

โดยมีค่าสัมประสิทธิ์ดังนี้

a_1 = 1.988

a_2 = 746.149

a_3 = 167.321

b_1 = 7.991

b_2 = 2248.028

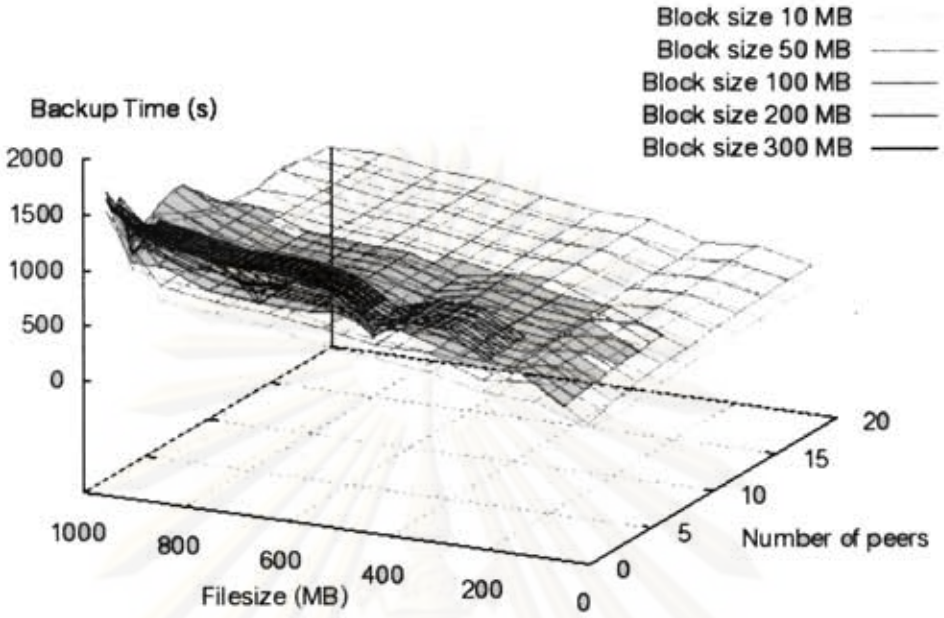
b_3 = 540.612

จากสมการ (4) สามารถจัดรูปสมการใหม่ได้ โดยเขียนความสัมพันธ์ของจำนวนเครื่องในระบบในรูปฟังก์ชันของขนาดไฟล์ ดังแสดงในสมการ (5) ซึ่งสามารถนำมาใช้ประมาณหรือทำนายจำนวนเครื่องที่เพียงพอและเหมาะสมสำหรับการสำรองข้อมูลโดยกำหนดขนาดไฟล์ซึ่งเมื่อเพิ่มจำนวนเครื่องมากกว่าค่านี้แล้วเวลาที่ใช้จะไม่ลดลงอีก ยกตัวอย่างเช่น เมื่อมีไฟล์ขนาด 1.5 GB (sz = 1500) เมื่อประมาณด้วยสมการ (5) จะให้ค่า n = 4.668093528 นั่นคือในการสำรองข้อมูลครั้งนี้จำนวนเครื่องที่มากเพียงพอเพื่อที่จะใช้เวลาน้อยที่สุดคือ 5 เครื่อง เนื่องจากเมื่อเพิ่มจำนวนเครื่องมากกว่านี้จะไม่เกิดประโยชน์ในการที่จะช่วยลดเวลาอีกต่อไป

$$n = \left(\frac{b_1}{b_2}\right)(sz - a_2) + a_1 \quad (5)$$

4.2 วิธีที่ 2 : แบ่งชิ้นส่วนย่อยโดยกำหนดขนาดของชิ้นส่วนย่อยคงที่

จากแบบจำลองเบื้องต้นในหัวข้อ 3.3 การคาดการณ์พารามิเตอร์ที่เกี่ยวข้องกับเวลาที่ใช้ในขั้นตอนวิธีนี้ประกอบด้วยขนาดไฟล์ (sz), จำนวนเครื่องในระบบ (n) และขนาดชิ้นส่วนย่อย (bs) การทดลองในส่วนนี้ได้ทำการวัดเวลาที่ใช้ในการสำรองข้อมูลโดยทดสอบกับขนาดไฟล์ตั้งแต่ 50 MB จนถึง 1 GB ลงบนเครื่องจำนวน 2 ถึง 20 เครื่อง โดยแบ่งเป็นชิ้นส่วนย่อยขนาด 10 MB, 50 MB, 100 MB, 200 MB และ 300 MB ผลการทดลองแสดงดังรูปที่ 10



รูปที่ 10 กราฟแสดงเวลาในการสำรองข้อมูลเทียบกับจำนวนเครื่องในระบบและขนาดไฟล์เมื่อเลือกใช้ขนาดชิ้นส่วนย่อยแตกต่างกัน

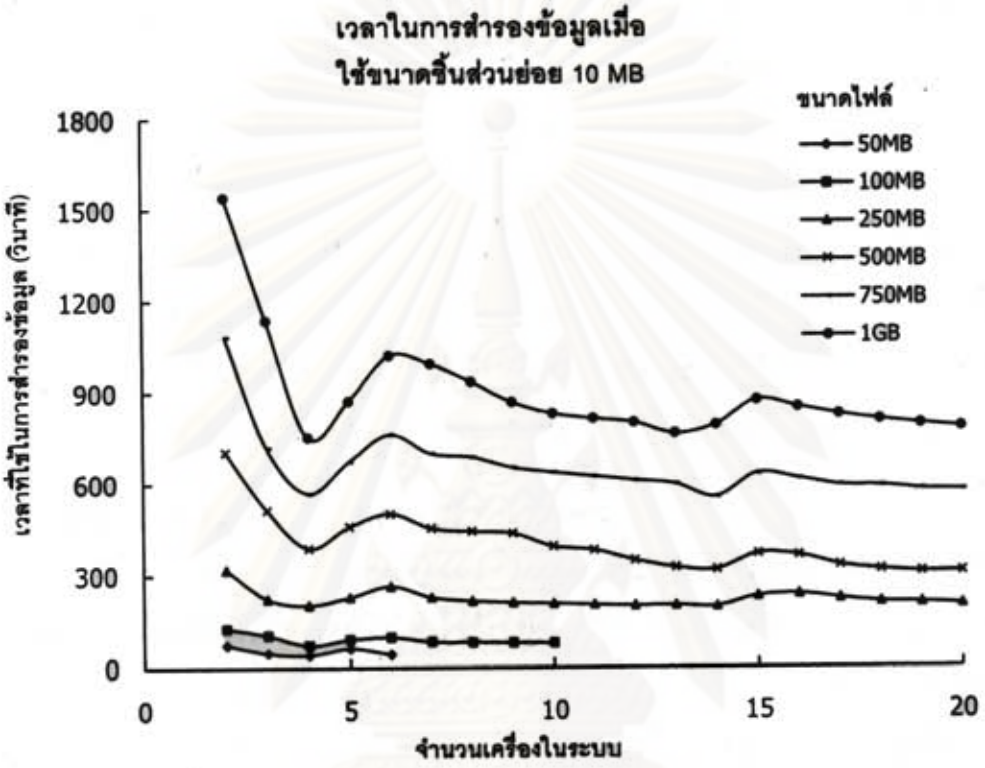
แต่ละแผ่นตาราง (Grid) ในรูปที่ 10 แสดงถึงเวลาที่ใช้ในการสำรองข้อมูลของแต่ละขนาดชิ้นส่วนย่อย จากกราฟแสดงให้เห็นว่าที่ขนาดไฟล์เท่ากันเมื่อเลือกใช้ชิ้นส่วนย่อยขนาดใหญ่ขึ้นจะใช้เวลาในการสำรองข้อมูลเพิ่มขึ้นด้วย นั่นคือเวลาที่ใช้แปรผันตรงกับขนาดของชิ้นส่วนย่อย

หากใช้ขนาดชิ้นส่วนย่อยเดียวกันพบว่าเวลาที่ใช้ลดลงเมื่อจำนวนเครื่องในระบบเพิ่มขึ้นในทำนองเดียวกับกราฟในวิธีแรก ตัวอย่างกราฟของเวลาที่ใช้ในการสำรองข้อมูลเมื่อเทียบกับจำนวนเครื่องในระบบและขนาดไฟล์ในกรณีที่เลือกใช้ขนาดชิ้นส่วนย่อย 10 MB แสดงดังรูปที่ 11 จากกราฟแสดงให้เห็นว่าเมื่อจำนวนเครื่องเพิ่มมากขึ้นเวลาที่ใช้ลดลงอย่างรวดเร็วตามที่คาดการณ์เบื้องต้น อย่างไรก็ตามสังเกตว่าเวลาที่ใช้เพิ่มขึ้นเมื่อใช้จำนวนเครื่องตั้งแต่ 4 เครื่องถึง 6 เครื่อง หลังจากนั้นเวลาที่ใช้ลดลงอย่างรวดเร็วอีกครั้ง จากแนวโน้มของกราฟที่อยู่ในลักษณะเส้นโค้งแบบไซน์ เราสามารถเสนอแบบจำลองของเวลาที่ใช้ในการสำรองข้อมูล (T_2) ซึ่งแปรผันตามขนาดไฟล์ (sz), จำนวนเครื่องในระบบ (n) และขนาดชิ้นส่วนย่อย (bs) ด้วยสมการดังนี้

$$T_2(sz,n,bs) = \begin{matrix} p_0 + p_1 n + (p_2 sz)Ae^{\alpha n}\cos\big(w(n-phase)\big) \\ + p_3 sz + p_4 bs \end{matrix} \quad (6)$$

จากรูปที่ 11 กราฟของเวลาอยู่ในลักษณะกราฟไซน์ที่มีแอมพลิจูด (Amplitude) ลดลง ทำให้ได้พจน์ที่เป็นการคูณกันของพจน์ exponential ($Ae^{\alpha n}$) และพจน์ไซน์ (cos (w(n-phase)) ในสมการ (6) ขนาดของแอมพลิจูดเพิ่มขึ้นแปรผันตามขนาดไฟล์ จึงเพิ่มพจน์ $p_2$sz ในพจน์ของแอม

พลิจูดด้วย ในพจน์แรก (p_0) แสดงถึงเวลาสำหรับการจัดเตรียม (setup time) ในการสำรองข้อมูลแต่ละครั้งซึ่งมีขนาดคงที่ เมื่อจำนวนเครื่องเพิ่มขึ้นกราฟไซน์มีแกนของกราฟเอนลงเป็นผลให้เกิดพจน์ p_1n นอกจากนี้เมื่อขนาดไฟล์เพิ่มขึ้นเป็นผลให้เวลาเพิ่มขึ้น เวลาที่ใช้แปรผันตามขนาดไฟล์และขนาดของชิ้นส่วนย่อยจึงเป็นที่มาของพจน์ p_3sz และ p_4bs ตามลำดับ



รูปที่ 11 กราฟแสดงเวลาในการสำรองข้อมูลเทียบกับจำนวนเครื่องในระบบและขนาดไฟล์เมื่อเลือกใช้ขนาดชิ้นส่วนย่อย 10 MB

สัมประสิทธิ์ในสมการ (6) สามารถประมาณโดยใช้วิธีกำลังสองน้อยที่สุด ได้ดังนี้

A	=	419.251
α	=	-0.541
w	=	1.525
phase	=	2.221
p_0	=	35.855
p_1	=	-6.486
p_2	=	0.004
p_3	=	0.906
p_4	=	0.638

กราฟของเวลาที่ใช้ในการสำรองข้อมูลด้วยวิธีนี้มีความแตกต่างจากวิธีแรก หากมองกราฟในแต่ละขนาดไฟล์พบว่ามีจุดที่เวลาสูงหรือต่ำผิดปกติ (Outliers) หากมองกราฟโดยรวมจุดที่ใช้เวลาลดลงผิดปกติคือที่จำนวนเครื่อง 4 กับ 14 เครื่องและจุดที่ใช้เวลาเพิ่มขึ้นสูงผิดปกติคือที่จำนวนเครื่อง 6 กับ 15 เครื่อง ผลการทดลองที่ได้นี้มีแนวโน้มในทำนองเดียวกันโดยไม่ขึ้นกับขนาดไฟล์และขนาดชิ้นส่วนย่อย เมื่อเปรียบเทียบกับสมการของเวลาที่ใช้ในการสำรองข้อมูลในวิธีแรกจะพบว่าพจน์ที่เป็น Exponential ได้ถูกพจน์ไซน์คูณเพิ่มเข้าไปเป็นผลให้เกิดการแกว่งของเวลาที่ทำให้ได้ค่าน้อยผิดปกติและมากผิดปกติ เวลาที่ผิดปกตินี้คาดว่าเป็นผลจากการกระจายชิ้นส่วนข้อมูลไปยังเครื่องในระบบ (Data Distribution) เนื่องจากการแบ่งชิ้นส่วนย่อยโดยกำหนดขนาดของชิ้นส่วนย่อยคงที่จะทำให้แต่ละเครื่องในระบบได้รับชิ้นส่วนย่อยจำนวนไม่เท่ากัน ดังนั้นเวลาที่ใช้ในการรับส่งข้อมูลไปยังแต่ละเครื่องจึงมีความแตกต่างกันด้วย อีกปัจจัยที่มีอิทธิพลในการส่งข้อมูลคือแบนด์วิดท์ของเครือข่าย (Network Bandwidth) ที่จำกัดรวมถึงความสามารถและอัตราในการรับส่งข้อมูลของแต่ละเครื่องในแต่ละช่วงเวลาไม่เท่าเทียมกัน

4.3 วิธีที่ 3 : แบ่งชิ้นส่วนย่อยโดยใช้อีเรเซอร์โค้ด

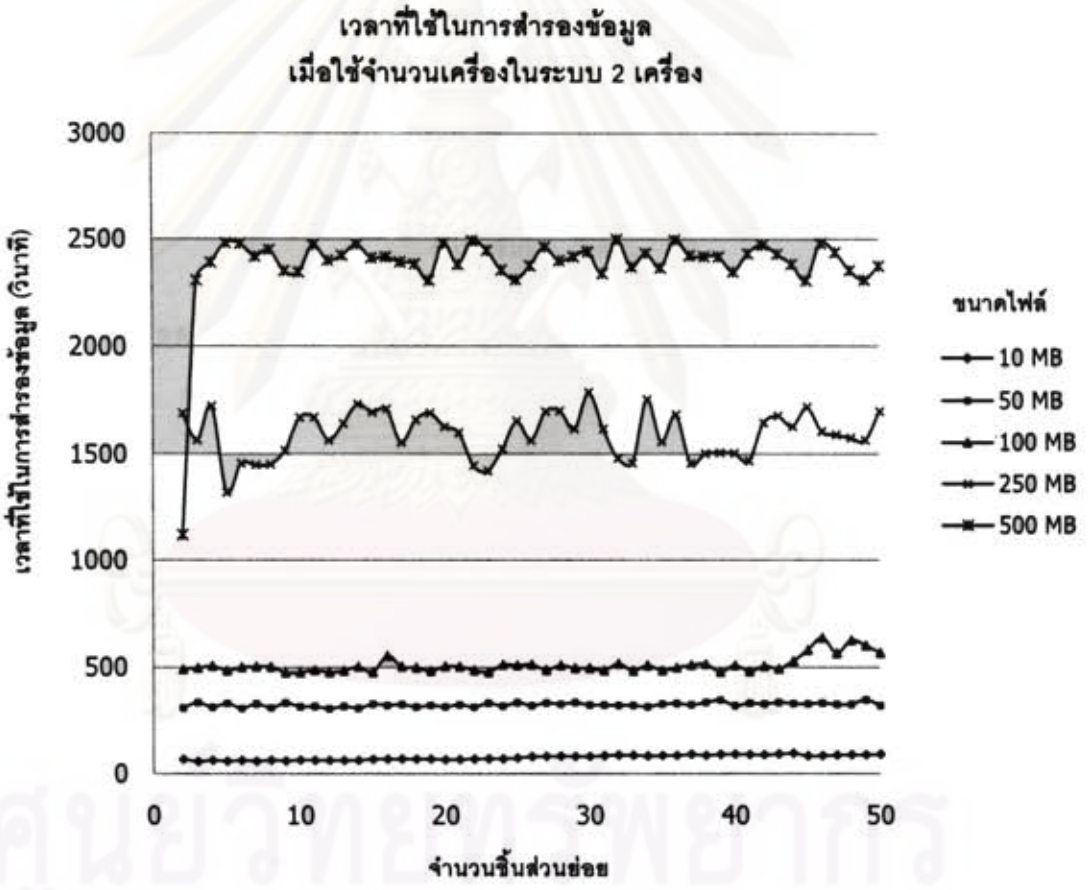
จากหัวข้อ 2.1.2 การเพิ่มจำนวนชิ้นส่วนย่อยในการแบ่งไฟล์ด้วยวิธีอีเรเซอร์โค้ดจะเป็นการเพิ่มสภาพพร้อมใช้งานของไฟล์ (A) เมื่ออยู่ภายใต้สภาพแวดล้อมที่เหมาะสม (แปรผันตามค่าสภาพพร้อมใช้งานของเพียร์ (μ) และค่าปัจจัยการยืด (S) การศึกษาผลกระทบของค่าปัจจัยการยืดแสดงให้เห็นว่าสำหรับค่าสภาพพร้อมใช้งานของเพียร์ที่เหมาะสม (มากกว่าค่าขีดแบ่ง) เมื่อเพิ่มค่าปัจจัยการยืดมากขึ้นจะช่วยเพิ่มสภาพพร้อมใช้งานของไฟล์ อย่างไรก็ตามการเพิ่มค่าปัจจัยการยืดส่งผลกับเวลาที่ใช้ในการสำรองข้อมูลที่มากขึ้นเช่นกัน

จากแบบจำลองเบื้องต้นในหัวข้อ 3.3 พารามิเตอร์ที่คาดว่าจะส่งผลกับเวลาที่ใช้ในการแบ่งชิ้นส่วนย่อยด้วยอีเรเซอร์โค้ดได้แก่ ขนาดไฟล์ (sz), จำนวนเครื่องในระบบ (n), จำนวนชิ้นส่วนย่อย (b) และค่าปัจจัยการยืด (S) ดังนั้นการทดลองในหัวข้อนี้ได้ทำการศึกษาและวัดเวลาที่ใช้ในการสำรองข้อมูลขนาด 10 MB, 50 MB, 100 MB, 250 MB และ 500 MB โดยแบ่งชิ้นส่วนย่อยตั้งแต่ 2 ถึง 50 ชิ้น การเข้ารหัสข้อมูลด้วยอีเรเซอร์โค้ดเลือกใช้ Reed-Solomon code (RS Code) และ Luby Transform code (LT Code) ที่ค่าปัจจัยการยืด 2 ค่าคือ S=1.5 และ 2 ผลการทดลองที่ได้สามารถพิจารณาได้ดังนี้

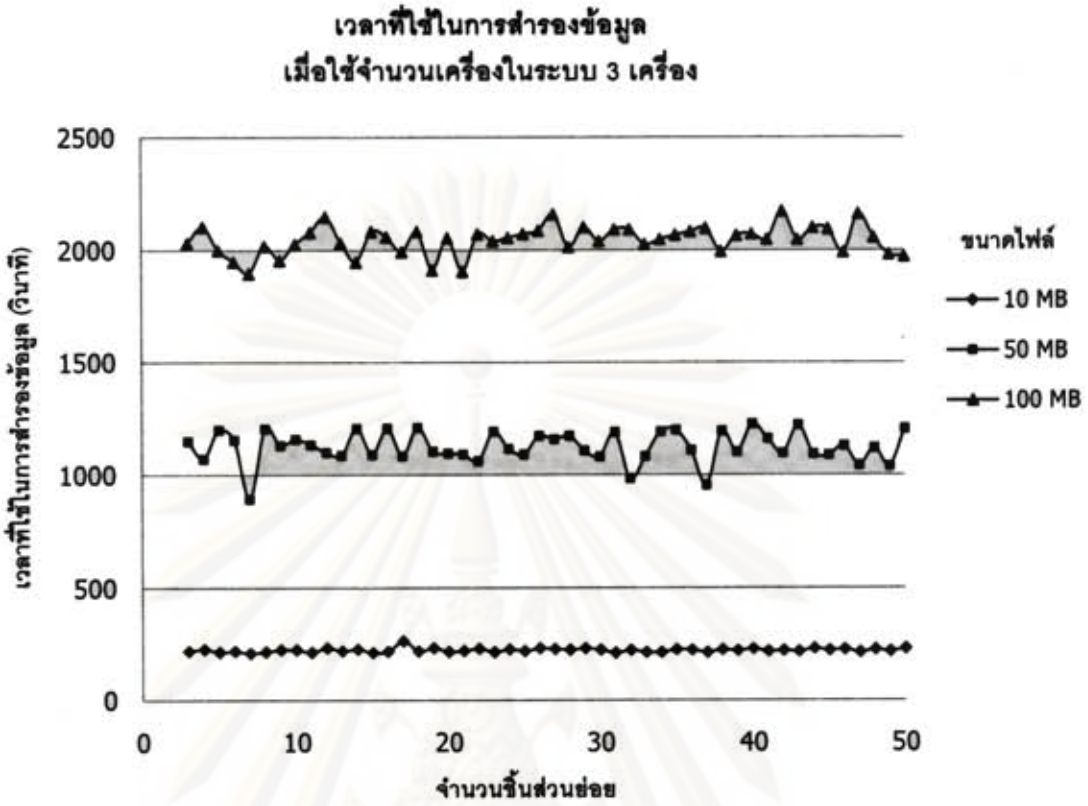
4.3.1 ผลการทดลองวัดเวลาที่ใช้ในการสำรองข้อมูลทั้งหมด

จากผลการทดลองวัดเวลาที่ใช้ในการสำรองข้อมูลทั้งหมดพบว่า เมื่อใช้จำนวนเครื่องในระบบเท่ากัน พิจารณาการสำรองไฟล์ที่ขนาดหนึ่ง การแบ่งชิ้นส่วนย่อยจำนวนต่างกันใช้เวลาใน

การสำรองข้อมูลไม่แตกต่างกันมากนักดังแสดงในรูปที่ 13, รูปที่ 13 และ รูปที่ 14 ซึ่งแสดงกราฟของเวลาที่ใช้ในการสำรองข้อมูลที่ขนาดไฟล์ต่างๆ เมื่อเลือกใช้ RS Code ที่ค่า S=2 โดยใช้เครื่องในระบบจำนวน 2, 3 และ 5 เครื่องตามลำดับ การทดลองสำหรับการเข้ารหัสด้วย LT Code ให้ผลในทำนองเดียวกันคือเมื่อเพิ่มจำนวนชิ้นส่วนย่อยเวลาที่ใช้ไม่แตกต่างกันมากในแต่ละขนาดไฟล์ นั่นคือเวลาที่ใช้ไม่ได้แปรตามจำนวนชิ้นส่วนย่อย (b) ดังที่คาดไว้ในแบบจำลองเบื้องต้น ดังนั้นแบบจำลองของเวลาที่ใช้ในการสำรองข้อมูลด้วยวิธีการแบ่งชิ้นส่วนย่อยโดยใช้อีเรเซอร์โค้ดสามารถเขียนใหม่ได้โดยอยู่ในรูปฟังก์ชันของขนาดไฟล์ (sz), จำนวนเครื่องในระบบ (n) และค่าปัจจัยการยืด (S) คือ $T_3(sz,n,S)$

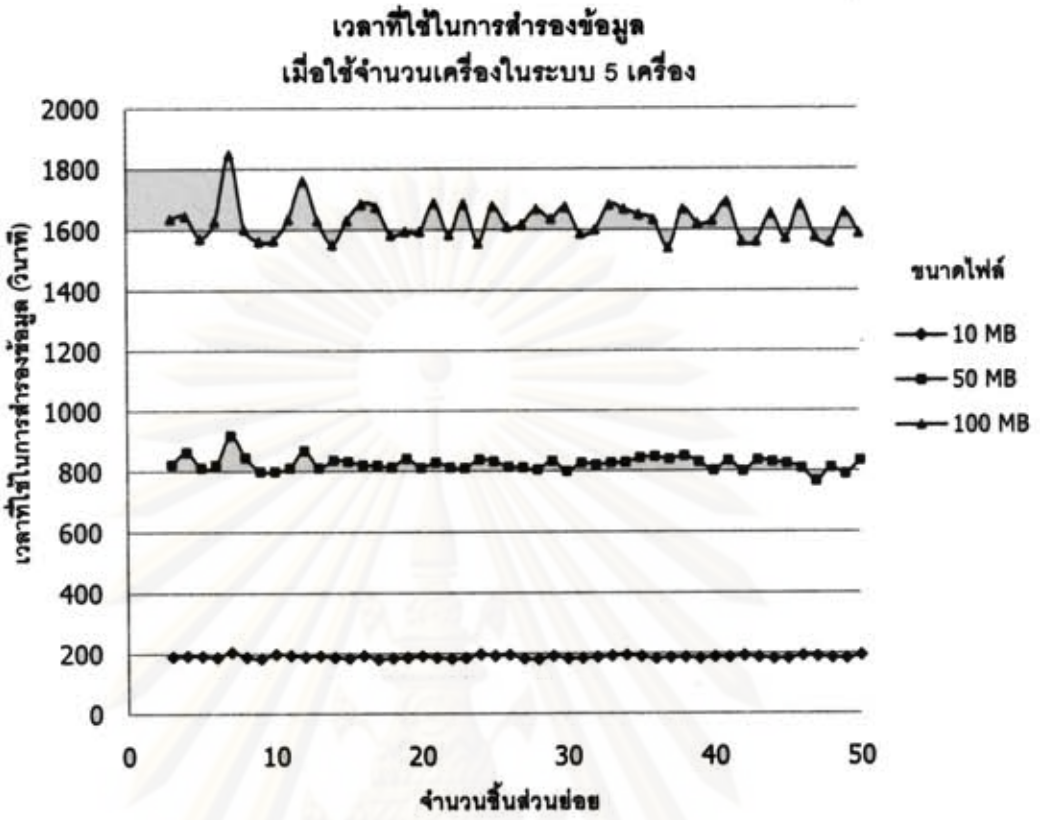


รูปที่ 12 กราฟแสดงเวลาในการสำรองข้อมูลเทียบกับจำนวนชิ้นส่วนย่อยเมื่อเลือกใช้ขนาดไฟล์10, 50,100, 250 และ 500 MB บนจำนวน 2 เครื่อง โดยใช้ RS Code (S=2)



รูปที่ 13 กราฟแสดงเวลาในการสำรองข้อมูลเทียบกับจำนวนชิ้นส่วนย่อยเมื่อเลือกใช้ขนาดไฟล์10, 50 และ 100 MB บนจำนวน 3 เครื่อง โดยใช้ RS Code (S=2)

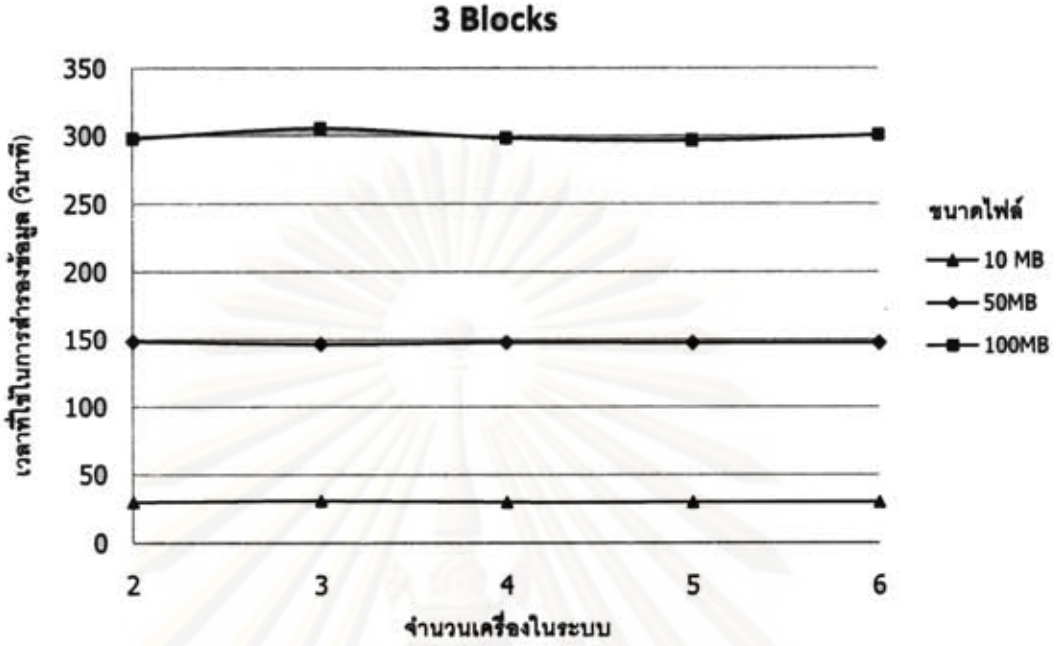
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



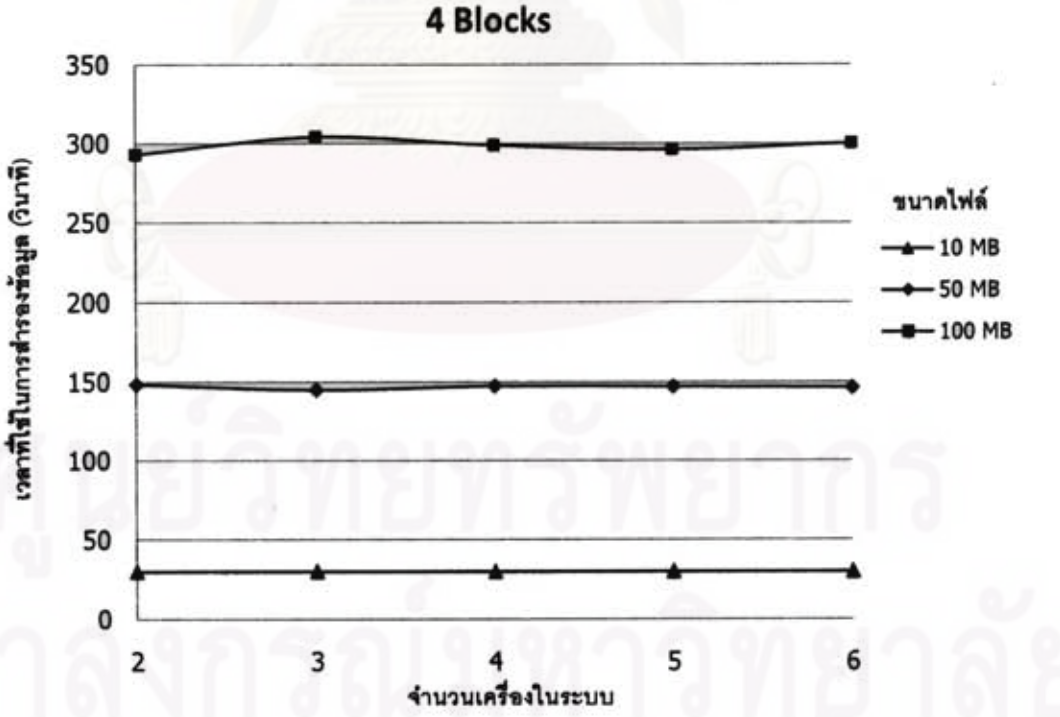
รูปที่ 14 กราฟแสดงเวลาในการสำรองข้อมูลเทียบกับจำนวนชิ้นส่วนย่อยเมื่อเลือกใช้ขนาดไฟล์10, 50 และ 100 MB บนจำนวน 5 เครื่อง โดยใช้ RS Code (S=2)

4.3.2 เวลาในการสำรองข้อมูลในแต่ละขั้นตอนของวิธีการแบ่งชิ้นส่วนย่อยด้วยอีเรเซอร์โค้ด

การสำรองข้อมูลโดยแบ่งชิ้นส่วนย่อยด้วยอีเรเซอร์โค้ดนั้นมีขั้นตอนหลัก 2 ขั้นตอนคือ การเข้ารหัสข้อมูลด้วยอีเรเซอร์โค้ดและการส่งชิ้นส่วนย่อยไปยังเครื่องภายในระบบ พิจารณาเวลาที่ใช้ในการเข้ารหัสด้วยอีเรเซอร์โค้ดที่ใช้ RS Code (S=2) โดยเทียบกับจำนวนเครื่องในระบบที่ขนาดไฟล์ต่างกันโดยแบ่งชิ้นส่วนย่อยจำนวน 3 และ 4 ชิ้นส่วนย่อย ดังแสดงในรูปที่ 15 และรูปที่ 16 ตามลำดับ แสดงให้เห็นว่าเวลาในส่วนนี้ไม่ขึ้นกับจำนวนเครื่องในระบบและจำนวนชิ้นส่วนย่อยแต่จะแปรผันตามขนาดไฟล์

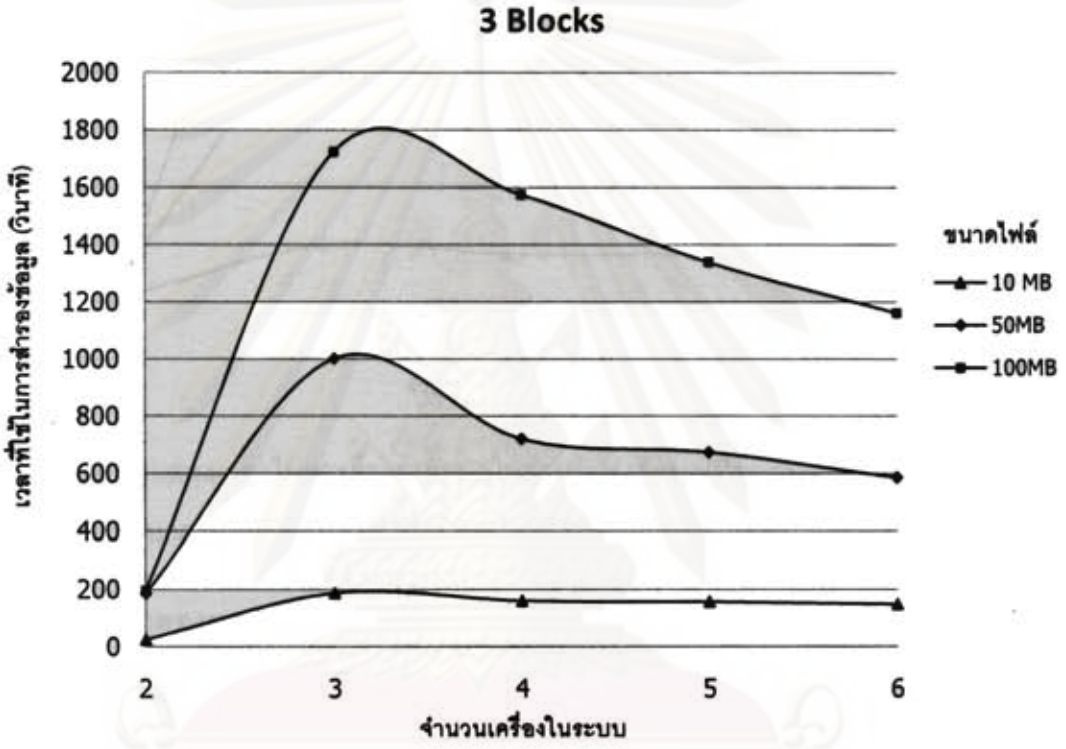


รูปที่ 15 กราฟแสดงเวลาในการสำรองข้อมูลในส่วนการเข้ารหัสเทียบกับจำนวนเครื่องในระบบเมื่อเลือกใช้ขนาดไฟล์ 10, 50 และ 100 MB โดยแบ่งชิ้นส่วนย่อย 3 ชิ้นด้วย RS Code (S=2)



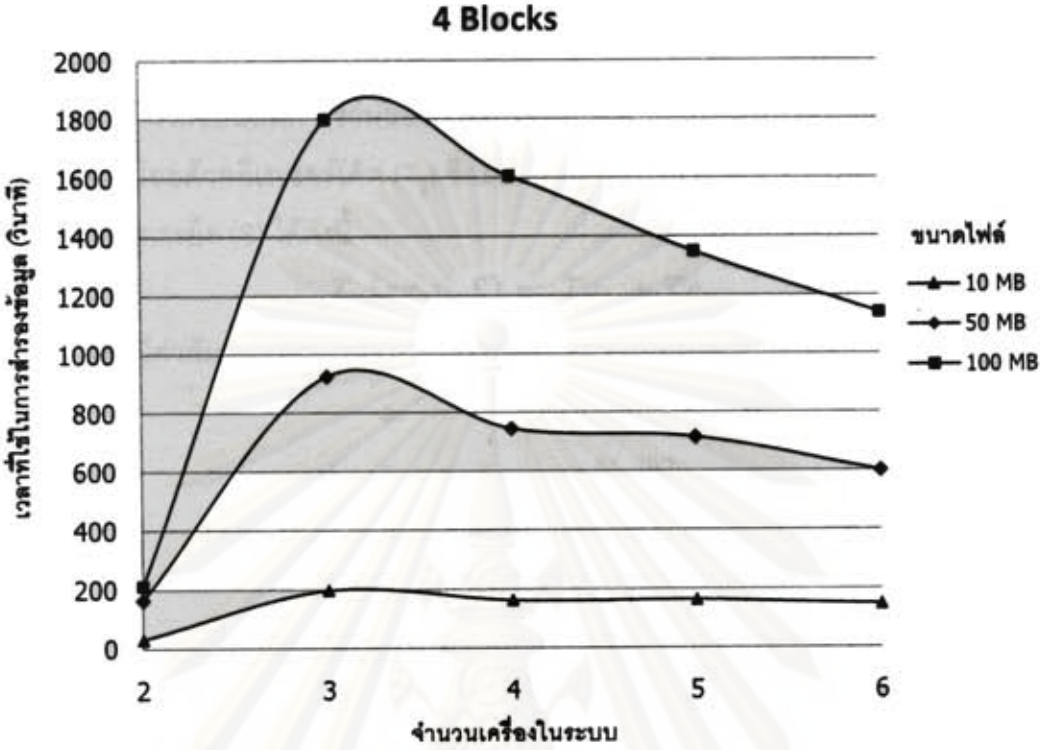
รูปที่ 16 กราฟแสดงเวลาในการสำรองข้อมูลในส่วนการเข้ารหัสเทียบกับจำนวนเครื่องในระบบเมื่อเลือกใช้ขนาดไฟล์ 10, 50 และ 100 MB โดยแบ่งชิ้นส่วนย่อย 4 ชิ้นด้วย RS Code (S=2)

พิจารณาเวลาที่ใช้ในการส่งชิ้นส่วนย่อยไปยังเครื่องในระบบเมื่อจำนวนเครื่องมากขึ้น กราฟของเวลาที่ใช้ลดลงในลักษณะเส้นโค้งแบบ Exponential และเวลาที่ใช้เพิ่มขึ้นแปรผันตามขนาดไฟล์ในทำนองเดียวกับการแบ่งชิ้นส่วนย่อยวิธีแรก ดังรูปที่ 17 และรูปที่ 18 ซึ่งแสดงกราฟของเวลาที่ใช้เทียบกับจำนวนเครื่องในระบบเมื่อใช้ขนาดไฟล์ต่างกันด้วย RS Code (S=2) โดยแบ่งชิ้นส่วนย่อย 3 และ 4 ชิ้นตามลำดับ นอกจากนี้เวลาที่ใช้ยังแปรผันตามค่าปัจจัยการยืด (S) ด้วยเนื่องจากจำนวนชิ้นส่วนย่อยที่แบ่งจะเพิ่มขึ้นตามค่า S



รูปที่ 17 กราฟแสดงเวลาในการสำรองข้อมูลในส่วนรับส่งข้อมูลเทียบกับจำนวนเครื่องในระบบเมื่อเลือกใช้ขนาดไฟล์ 10, 50 และ 100 MB โดยแบ่งชิ้นส่วนย่อย 3 ชิ้นด้วย RS Code (S=2)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 18 กราฟแสดงเวลาในการสำรองข้อมูลในส่วนรับส่งข้อมูลเทียบกับจำนวนเครื่องในระบบเมื่อเลือกใช้ขนาดไฟล์ 10, 50 และ 100 MB โดยแบ่งชิ้นส่วนย่อย 4 ชิ้นด้วย RS Code (S=2)

4.3.3 แบบจำลองของเวลาที่ใช้ในการสำรองข้อมูลด้วยขั้นตอนวิธีการแบ่งด้วยอีเรเซอร์โค้ด

จากผลการทดลองในหัวข้อ 4.3.2 สรุปได้ว่าเวลาที่ใช้ในการสำรองข้อมูลเมื่อแบ่งชิ้นส่วนย่อยโดยใช้อีเรเซอร์โค้ด สามารถแบ่งได้เป็น 2 ส่วนคือ

1. เวลาที่ใช้ในการเข้ารหัสข้อมูล เนื่องจากขนาดไฟล์มากขึ้นส่งผลให้ใช้เวลาในการเข้ารหัสมากขึ้น เวลาในส่วนนี้จึงแปรผันตรงกับขนาดไฟล์เริ่มต้น (sz) การประมาณเวลาในส่วนของการเข้ารหัสข้อมูล (Te_1) สามารถแสดงได้ดังสมการดังนี้

$$Te_1(sz) = c_1 sz + c_2 \quad (7)$$

2. เวลาที่ใช้ในการส่งข้อมูล เนื่องจากเวลาที่ใช้ลดลงเมื่อจำนวนเครื่องมากขึ้นและแปรผันตามขนาดไฟล์ โดยมีกราฟแบบ Exponential ในลักษณะเดียวกับวิธีแรก และเวลาที่ใช้มากขึ้นแปรผันตรงตามค่าปัจจัยการยืดที่เพิ่มขึ้นด้วย ดังนั้นสมการของเวลาที่ใช้ในส่วนนี้สามารถเขียนในรูปฟังก์ชันของขนาดไฟล์ (sz), จำนวนเครื่องในระบบ (n) และค่าปัจจัยการยืด (S) ได้ดังนี้

$$Te_2(sz, n, S) = d_1 b^{n+d_2 sz} + d_3 sz + d_4 S + d_5 \quad (8)$$

สามารถเขียนสมการรวมของเวลาที่ใช้ในการสำรองข้อมูลทั้งหมดโดยใช้ขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยด้วยอีเรเซอร์โค้ด (T_3) ซึ่งขึ้นกับขนาดไฟล์ (sz), จำนวนเครื่องภายในระบบ (n) และค่าปัจจัยการยืด (S) ได้ดังนี้

$$T_3(sz, n, S) = Te_1 + Te_2 \quad (9)$$

โดยมีค่าสัมประสิทธิ์ของสมการ (7), (8) และ (9) ดังนี้

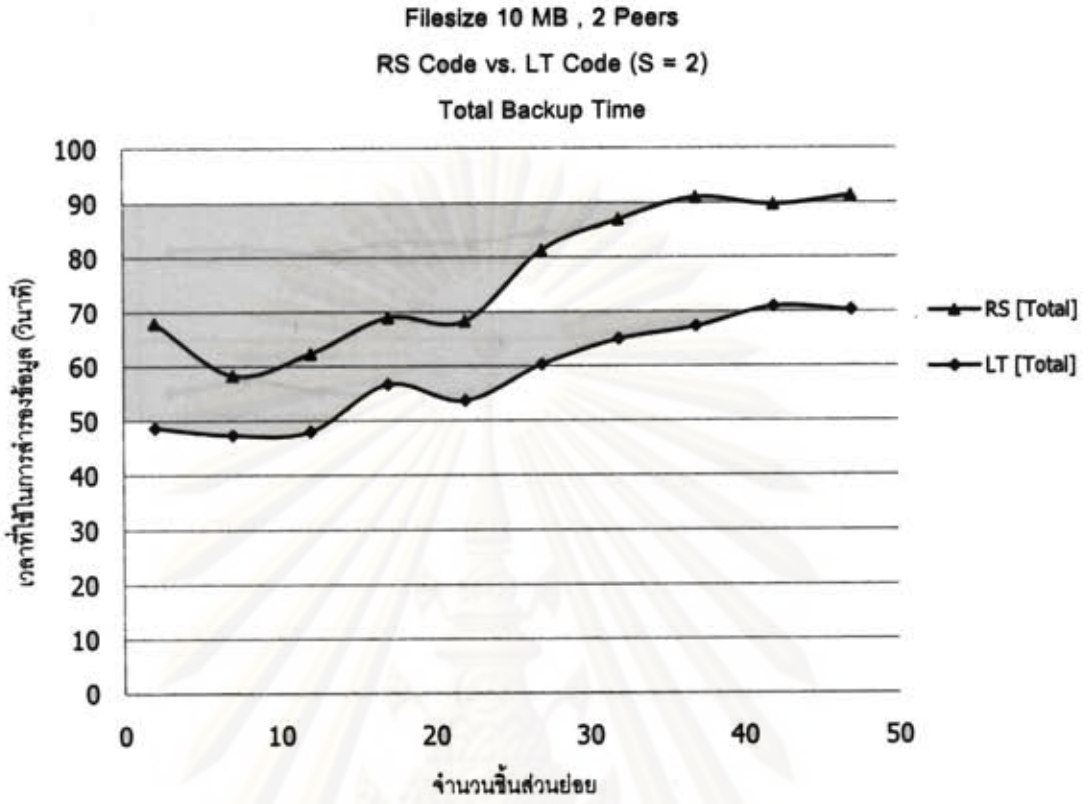
c_1	=	5.120
c_2	=	- 41.382
b	=	0.658
d_1	=	777.867
d_2	=	-0.029
d_3	=	10.970
d_4	=	1000.000
d_5	=	-2072.747

ค่าสัมประสิทธิ์ที่ได้คำนวณจากการทดลองที่ใช้ RS Code ในการเข้ารหัส สำหรับการเข้ารหัสด้วย LT Code สามารถคำนวณค่าสัมประสิทธิ์ได้ในทำนองเดียวกัน

4.3.4 การเปรียบเทียบเวลาที่ใช้ในการสำรองข้อมูลระหว่าง RS Code และ LT Code

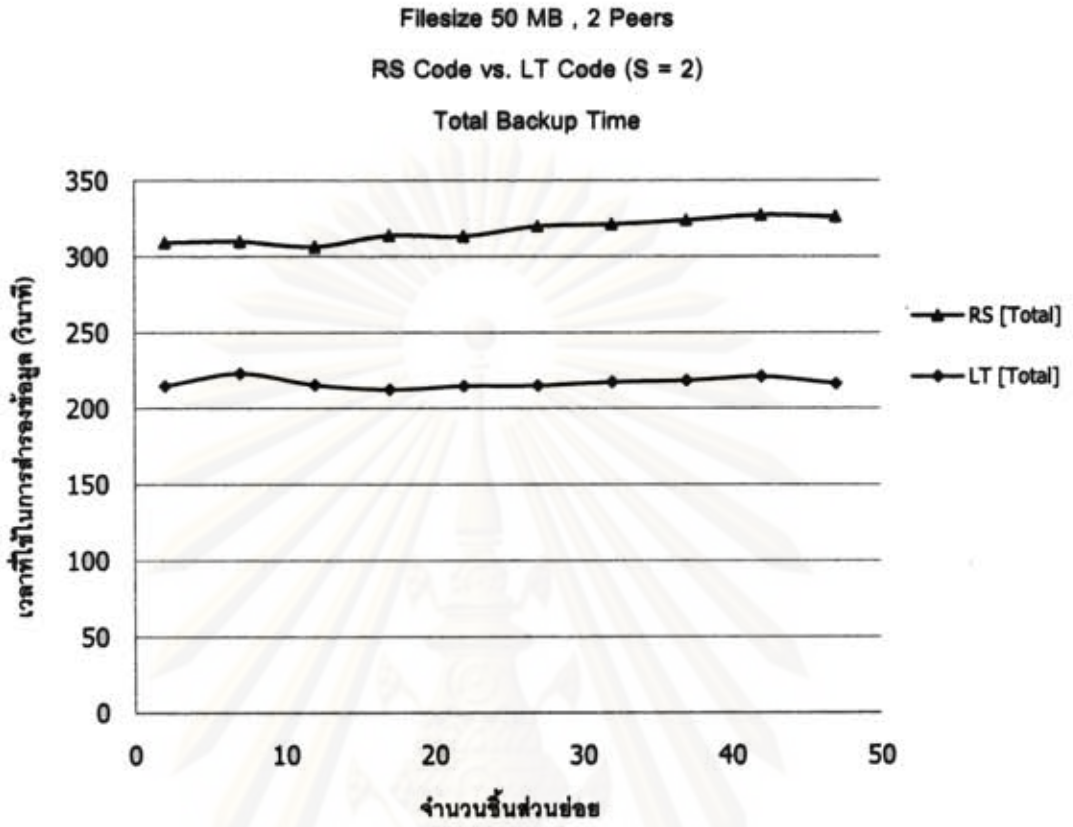
เมื่อเปรียบเทียบเวลาที่ใช้ในการสำรองข้อมูลโดยใช้การเข้ารหัสด้วย RS Code และ LT Code พบว่าเมื่อใช้จำนวนเครื่องและกำหนดค่าปัจจัยการยืดเท่ากัน เวลาที่ใช้ในการสำรองข้อมูลทั้งหมดของ LT Code น้อยกว่า RS code ในทุกขนาดไฟล์ ดังรูปที่ 19, รูปที่ 20 และรูปที่ 21 ซึ่งแสดงการเปรียบเทียบเวลาที่ใช้ในการสำรองข้อมูลทั้งหมดเมื่อเข้ารหัสด้วย RS code และ LT Code ที่ค่า S=2 โดยใช้จำนวนเครื่องในระบบ 2 เครื่องและขนาดไฟล์ 10 MB, 50 MB และ 100 MB ตามลำดับ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



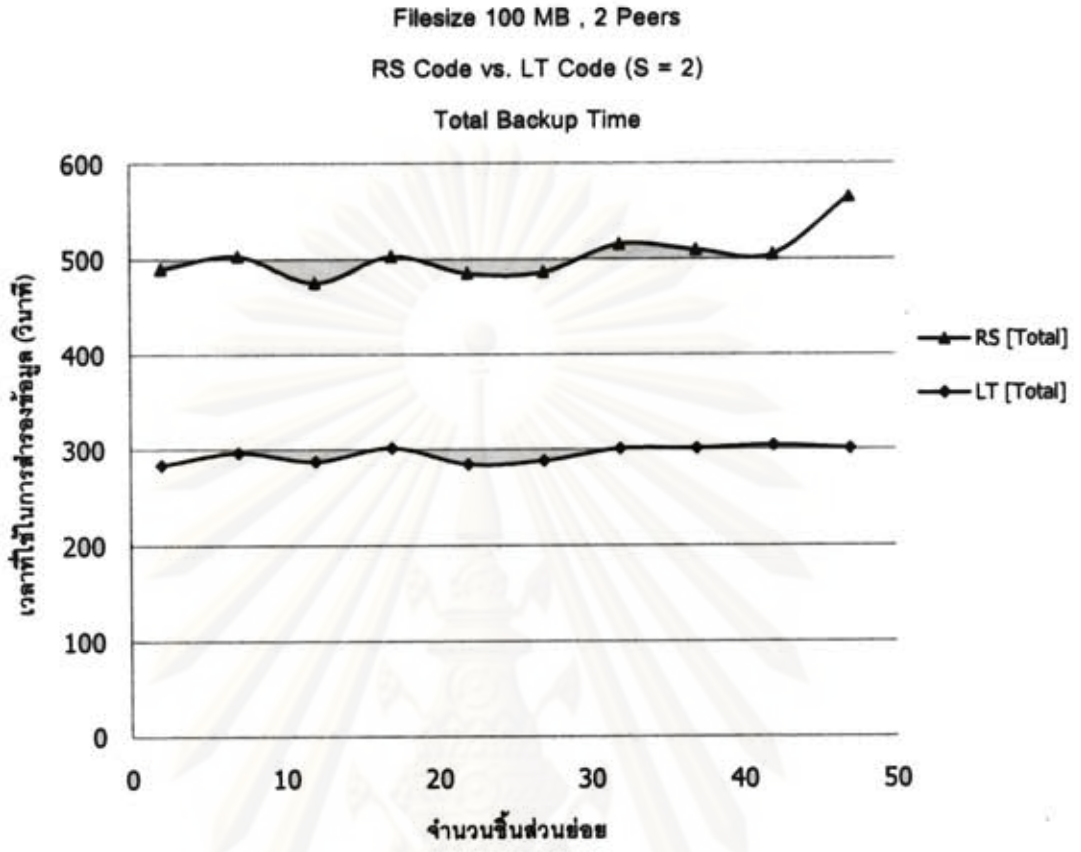
รูปที่ 19 กราฟแสดงเวลาในการสำรองข้อมูลทั้งหมดโดยเปรียบเทียบระหว่าง RS Code และ LT Code ที่ค่า S=2 เมื่อเลือกใช้ขนาดไฟล์ 10 MB บนเครื่องจำนวน 2 เครื่อง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



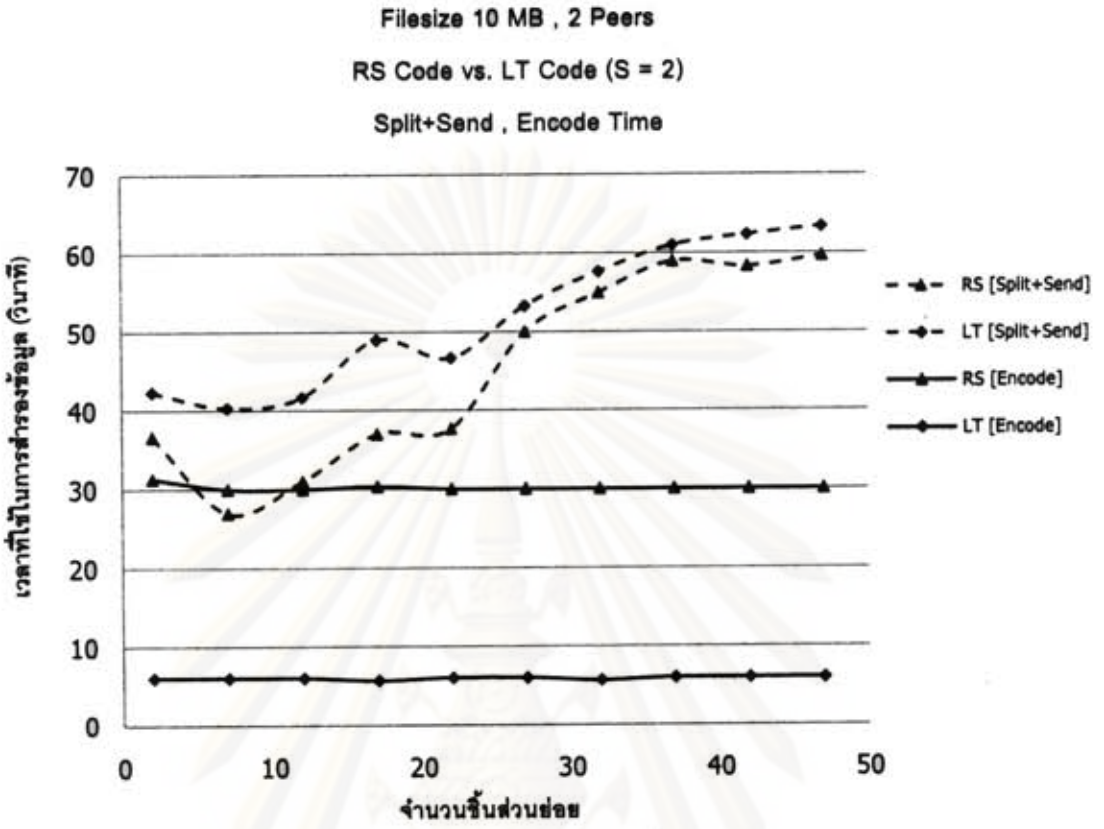
รูปที่ 20 กราฟแสดงเวลาในการสำรองข้อมูลทั้งหมดโดยเปรียบเทียบระหว่าง RS Code และ LT Code ที่ค่า S=2 เมื่อเลือกใช้ขนาดไฟล์ 50 MB บนเครื่องจำนวน 2 เครื่อง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



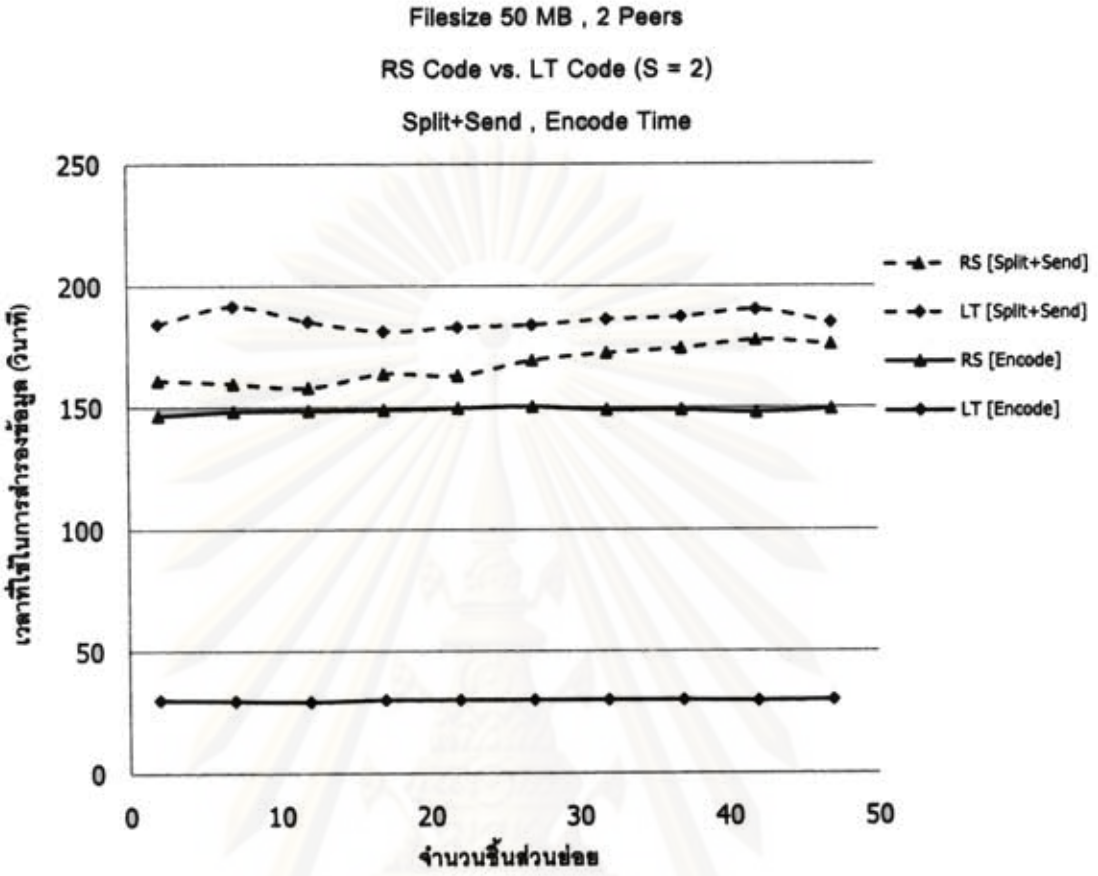
รูปที่ 21 กราฟแสดงเวลาในการสำรองข้อมูลทั้งหมดโดยเปรียบเทียบระหว่าง RS Code และ LT Code ที่ค่า S=2 เมื่อเลือกใช้ขนาดไฟล์ 100 MB บนเครื่องจำนวน 2 เครื่อง

พิจารณาเวลาที่ใช้ในการสำรองข้อมูลในแต่ละขั้นตอนของ RS Code และ LT Code พบว่า RS code ใช้เวลาในการเข้ารหัสมากกว่า LT Code เนื่องจากการเข้ารหัสด้วย RS code มีความซับซ้อนมากกว่าทำให้ใช้เวลาในส่วนนี้นานกว่า LT Code อย่างไรก็ตามการเข้ารหัสด้วย LT Code ต้องเสียค่าใช้จ่ายอื่น (Overhead) ในการทำดัชนี (Index) เพิ่มเติมทำให้ขนาดไฟล์หลังการเข้ารหัสเพิ่มขึ้นเล็กน้อย ส่งผลให้ใช้เวลาในการส่งข้อมูลมากกว่าการเข้ารหัสด้วย RS Code โดยจะเห็นได้ดังรูปที่ 22, รูปที่ 23 และรูปที่ 24 ซึ่งแสดงการเปรียบเทียบเวลาที่ใช้ในการสำรองข้อมูลในแต่ละส่วนเมื่อเข้ารหัสด้วย RS code และ LT Code ที่ค่า S=2 โดยใช้จำนวนเครื่องในระบบ 2 เครื่องและขนาดไฟล์ 10 MB, 50 MB และ 100 MB ตามลำดับ



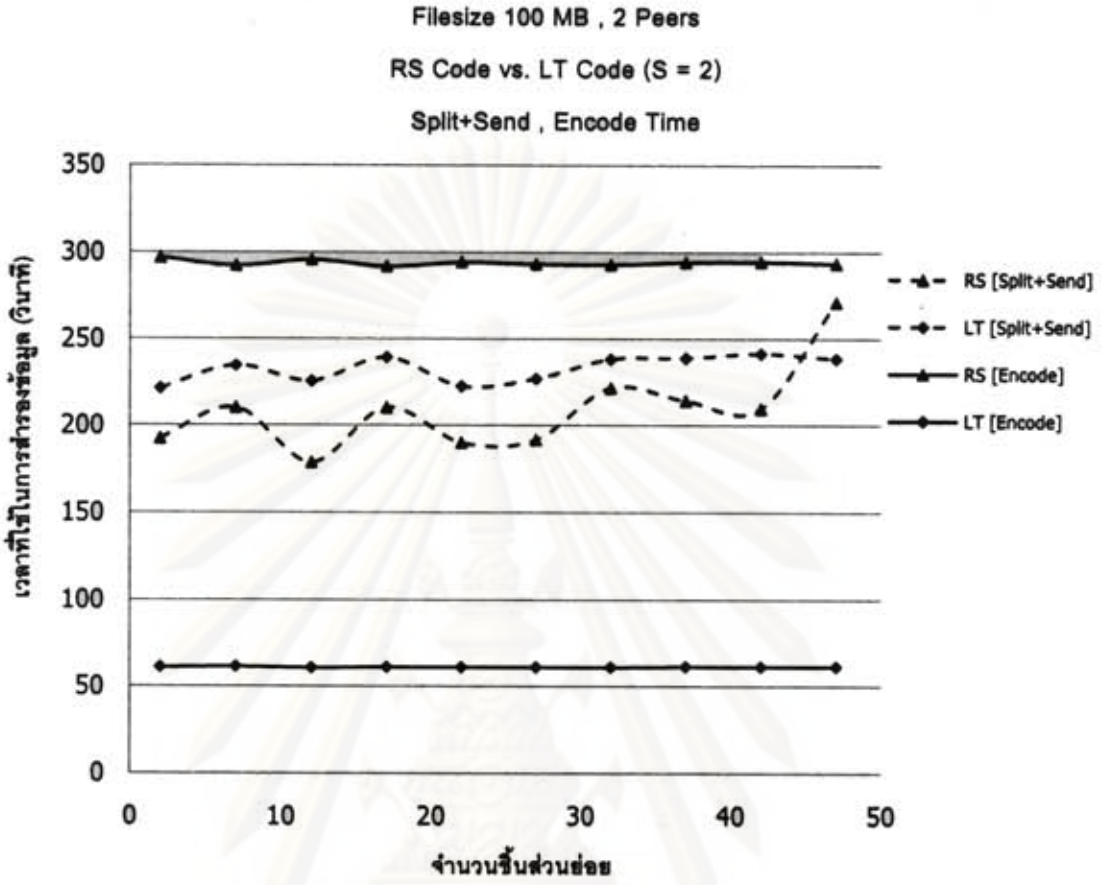
รูปที่ 22 กราฟแสดงเวลาในการสำรองข้อมูลในส่วนการเข้ารหัสข้อมูลและการรับส่งข้อมูลโดยเปรียบเทียบระหว่าง RS Code และ LT Code ที่ค่า S=2 เมื่อเลือกใช้ขนาดไฟล์ 10 MB บนเครื่องจำนวน 2 เครื่อง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 23 กราฟแสดงเวลาในการสำรองข้อมูลในส่วนการเข้ารหัสข้อมูลและการรับส่งข้อมูลโดยเปรียบเทียบระหว่าง RS Code และ LT Code ที่ค่า S=2 เมื่อเลือกใช้ขนาดไฟล์ 50 MB บนเครื่องจำนวน 2 เครื่อง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 24 กราฟแสดงเวลาในการสำรองข้อมูลในส่วนการเข้ารหัสข้อมูลและการรับส่งข้อมูลโดยเปรียบเทียบระหว่าง RS Code และ LT Code ที่ค่า S=2 เมื่อเลือกใช้ขนาดไฟล์ 100 MB บนเครื่องจำนวน 2 เครื่อง

เนื่องจากการเข้ารหัสด้วย LT Code ใช้เวลาในการส่งข้อมูลมากกว่า RS Code เพียงเล็กน้อยในขณะที่เวลาที่ใช้ในการเข้ารหัสด้วย RS Code ใช้เป็นจำนวนมากกว่า เวลาในส่วนนี้จึงส่งผลหลักกับเวลาในการสำรองข้อมูลรวมและทำให้เวลาที่ใช้ในการเข้ารหัสด้วย RS Code ยังคงมากกว่า LT Code

บทที่ 5
วิเคราะห์และสรุปผลการทดลอง

ในบทนี้ได้นำเสนอแบบจำลองของเวลาและสภาพพร้อมใช้งานของไฟล์ (Time-Availability Model) และกล่าวถึงการวิเคราะห์ประสิทธิภาพในด้านเวลาที่ใช้และสภาพพร้อมใช้งานของไฟล์จากแบบจำลองดังกล่าว

5.1 แบบจำลองของเวลาและสภาพพร้อมใช้งานของไฟล์ (Time-Availability Model)

ในบทที่ 4 ได้พิจารณาผลการทดลองวัดเวลาที่ใช้ในการแบ่งชิ้นส่วนย่อยแต่ละขั้นตอนวิธี และเสนอแบบจำลองของเวลาในการสำรองข้อมูลโดยแปรผันตามพารามิเตอร์ที่เกี่ยวข้อง สำหรับหัวข้อนี้ได้นำเสนอแบบจำลองซึ่งพิจารณาประสิทธิภาพของการสำรองข้อมูลในด้านเวลาและสภาพพร้อมใช้งานของข้อมูลทั้งสองด้านพร้อมกัน

จากสมการ (1) ในหัวข้อ 2.1.2 เนื่องจากการแบ่งชิ้นส่วนย่อยตามจำนวนเครื่องในระบบและการแบ่งชิ้นส่วนย่อยโดยกำหนดขนาดของชิ้นส่วนย่อยคงที่ไม่ได้ใช้การเข้ารหัสในขณะแบ่งข้อมูลจึงไม่มีจำนวนชิ้นส่วนย่อย (b) เพิ่มขึ้น สามารถคำนวนสภาพพร้อมใช้งานของไฟล์ (A) ได้โดยกำหนดค่าปัจจัยการยืด (S) เท่ากับ 1 ทำให้ได้ค่าสภาพพร้อมใช้งานของข้อมูล $A = \mu^b$

เนื่องจากค่าสภาพพร้อมใช้งานของเพียร์ (μ) มีค่าไม่เกิน 1 ดังนั้นสภาพพร้อมใช้งานของข้อมูล (A) ในการสำรองข้อมูลด้วยสองวิธีดังกล่าวจะมีค่าสูงเมื่อใช้จำนวนชิ้นส่วนย่อยน้อย และจะมีค่าลดลงอย่างรวดเร็วเมื่อเพิ่มจำนวนชิ้นส่วนย่อยมากขึ้น ในทางตรงกันข้ามสำหรับวิธีการแบ่งชิ้นส่วนย่อยโดยใช้อีเรเซอร์โค้ดเมื่อมีค่าสภาพพร้อมใช้งานของเพียร์ (μ) และกำหนดค่าปัจจัยการยืด (S) ที่เหมาะสมจะช่วยเพิ่มสภาพพร้อมใช้งานของข้อมูลเมื่อเพิ่มจำนวนชิ้นส่วนย่อย ถึงแม้ว่าการกำหนดค่าปัจจัยการยืด (S) ที่มีค่ามากจะทำให้ได้ค่าสภาพพร้อมใช้งานของไฟล์มากสำหรับค่าสภาพพร้อมใช้งานของเพียร์ (μ) ค่าหนึ่ง การสำรองข้อมูลโดยเพิ่มค่าปัจจัยการยืด (S) ส่งผลกับเวลาที่มากขึ้นเช่นกัน

จากที่กล่าวข้างต้นจะเห็นได้ว่าขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยแต่ละวิธีส่งผลกับเวลาที่ใช้และสภาพพร้อมใช้งานของไฟล์แตกต่างกันซึ่งขึ้นอยู่กับปัจจัยและสภาพแวดล้อมที่กำหนด การพิจารณาเลือกขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยที่เหมาะสมจึงใช้การเปรียบเทียบประสิทธิภาพในด้านเวลาที่ใช้ในการสำรองข้อมูลและสภาพพร้อมใช้งานของไฟล์โดยนำเสนอแบบจำลองสำหรับการตัดสินใจประสิทธิภาพ (Performance Decisive Model, P) ซึ่งเป็นอัตราส่วนระหว่างสภาพพร้อมใช้งานของไฟล์และเวลาที่ใช้ในการสำรองข้อมูลทั้งหมด ดังสมการดังนี้

$$P = A/T = \begin{cases} \frac{\mu^b}{T_1(sz,n)} & , \text{วิธีที่ 1} \\ \frac{\mu^b}{T_2(sz,n,bs)} & , \text{วิธีที่ 2} \\ \frac{A(S,b)}{T_3(sz,n,S)} & , \text{วิธีที่ 3} \end{cases} \quad (10)$$

ค่าปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) นี้สามารถนำมาใช้เป็นแนวทางการตัดสินใจเพื่อเลือกขั้นตอนวิธีในการสำรองข้อมูลที่เหมาะสม รวมถึงการเลือกพารามิเตอร์ในขั้นตอนวิธีนั้นด้วย โดยสมการข้างต้นแสดงให้เห็นว่าขั้นตอนวิธีที่ใช้เวลาในการสำรองข้อมูลน้อยและให้สภาพพร้อมใช้งานของไฟล์มากจะส่งผลให้ค่าปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) มาก

5.2 การวิเคราะห์ประสิทธิภาพ

ในหัวข้อนี้ได้ศึกษาพารามิเตอร์ในแต่ละขั้นตอนวิธีการแบ่งชิ้นส่วนย่อย รวมถึงผลกระทบของพารามิเตอร์กับประสิทธิภาพในด้านเวลาที่ใช้ในการสำรองข้อมูลและสภาพพร้อมใช้งานของไฟล์ โดยการวิเคราะห์ประสิทธิภาพโดยรวมจะพิจารณาค่าปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) จะอ้างอิงจากแบบจำลองในสมการ (10) ในหัวข้อ 5.1

การศึกษาผลกระทบของพารามิเตอร์กับประสิทธิภาพในด้านเวลาและสภาพพร้อมใช้งานของไฟล์จำเป็นต้องวิเคราะห์พารามิเตอร์ทีละพารามิเตอร์โดยกำหนดพารามิเตอร์อื่นๆคงที่ จากแบบจำลองข้างต้นพบว่าในแต่ละขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยมีพารามิเตอร์หลายพารามิเตอร์ด้วยกัน บางพารามิเตอร์มีผลกระทบกับประสิทธิภาพทั้งสองด้านพร้อมกัน บางพารามิเตอร์มีผลกระทบเฉพาะเวลาที่ใช้หรือสภาพพร้อมใช้งานของไฟล์เพียงด้านเดียว เนื่องจากปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) เป็นอัตราส่วนระหว่างประสิทธิภาพทั้งสองด้าน พารามิเตอร์ที่มีผลเพียงด้านใดด้านหนึ่งสามารถเปรียบเทียบค่าปัจจัยนี้ได้โดยการกำหนดประสิทธิภาพของอีกด้านให้คงที่ ดังนั้นในหัวข้อนี้จะวิเคราะห์ผลกระทบของพารามิเตอร์ที่เกี่ยวข้องกับปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) เฉพาะพารามิเตอร์ที่ส่งผลกับประสิทธิภาพทั้งสองด้าน (เวลาที่ใช้ในการสำรองข้อมูลและสภาพพร้อมใช้งานของไฟล์) พร้อมกัน

5.2.1 การวิเคราะห์ประสิทธิภาพในการสำรองข้อมูลของขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยโดยใช้อีเรเซอร์โค้ด

ขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยโดยใช้อีเรเซอร์โค้ดมีจำนวนพารามิเตอร์ที่ส่งผลกับปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) เป็นปริมาณมาก หากจำลองสถานการณ์จริงขณะทำการสำรองข้อมูลจะพบว่าขนาดไฟล์ (sz) เป็นพารามิเตอร์ที่ไม่สามารถเลือกกำหนดได้ในการสำรองข้อมูลแต่ละครั้ง พารามิเตอร์ที่สามารถเลือกปรับได้ ได้แก่ จำนวนเครื่องในระบบ (n), จำนวนชิ้น

ส่วนย่อย (b) และค่าปัจจัยการยืด (S) นอกจากนี้ในสมการ (10) ของหัวข้อ 5.1 แสดงให้เห็นว่า ขนาดไฟล์ (sz) และจำนวนเครื่องในระบบ (n) ส่งผลกับเวลาที่ใช้ในการสำรองข้อมูล (T_3) เท่านั้น หากมองภาพรวมของค่าปัจจัย P ของวิธีที่ 3 เมื่อเทียบกับค่าปัจจัยการยืด (S) พบว่าการเพิ่มค่าปัจจัยการยืด (S) จะทำให้ค่าสภาพพร้อมใช้งานของไฟล์สูงขึ้น แต่ในขณะเดียวกันเวลาที่ใช้ในการสำรองข้อมูลเพิ่มขึ้นเช่นกัน ดังนั้นพารามิเตอร์ที่สนใจในการแบ่งชิ้นส่วนย่อยด้วยวิธีนี้คือ จำนวนชิ้นส่วนย่อย (b) และค่าปัจจัยการยืด (S) โดยจะศึกษาวิเคราะห์และเปรียบเทียบระหว่างพารามิเตอร์เหล่านี้เพื่อให้มีความเหมาะสมมากที่สุด (มีค่าปัจจัย P มากที่สุด)

จากพารามิเตอร์ที่คาดว่าจะมีผลกระทบกับประสิทธิภาพทั้งสองด้านอันได้แก่ จำนวนชิ้นส่วนย่อย (b) และค่าปัจจัยการยืด (S) นำมาวาดกราฟของปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) เมื่อเทียบกับจำนวนชิ้นส่วนย่อย (b) โดยกำหนดค่าปัจจัยการยืด (S) ต่างๆกันบนจำนวนเครื่องในระบบ 2 เครื่องและขนาดไฟล์ 10 MB ซึ่งมีสภาพพร้อมใช้งานของเพียร์ (μ) แตกต่างกันดังนี้

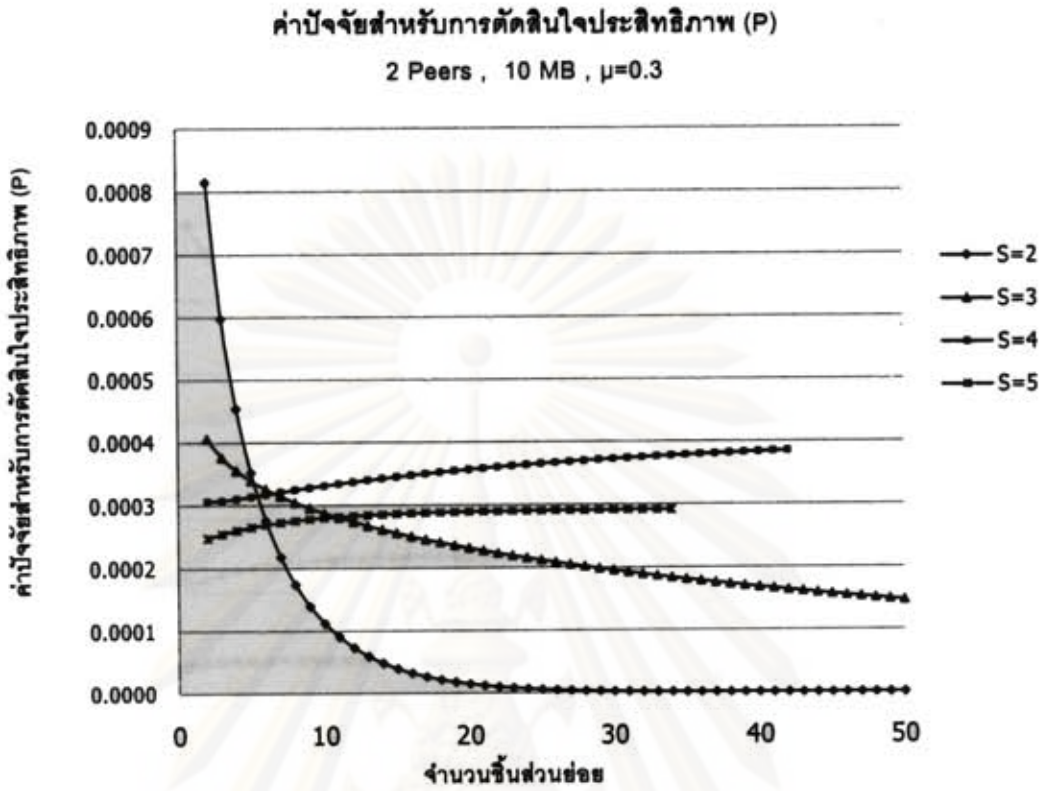
5.2.1.1 **สภาพพร้อมใช้งานของเพียร์ μ = 0.3**

พิจารณากราฟของค่าปัจจัย P ในรูปที่ 25 ที่มีสภาพพร้อมใช้งานของเพียร์ μ = 0.3 พบว่าเมื่อค่าปัจจัยการยืด (S) มีค่ามาก (ตั้งแต่ 4 เป็นต้นไป) การเพิ่มจำนวนชิ้นส่วนย่อย (b) จะเป็นผลให้ได้ค่าปัจจัย P สูง หากค่าปัจจัยการยืด (S) น้อยกว่า 4 แล้วการเพิ่มจำนวนชิ้นส่วนย่อยจะทำให้ค่าปัจจัย P ลดลง

จากการวิเคราะห์จำนวนชิ้นส่วนย่อยที่เหมาะสมในหัวข้อ 2.1.2.3 ในหน้า 8 และกราฟในรูปที่ 25 จะเห็นว่าเมื่อค่า μ*S > 1 (ในกรณีนี้คือเมื่อ S มีค่าตั้งแต่ 4 เป็นต้นไป) จะทำให้ได้ค่าปัจจัย P มากขึ้นเมื่อเพิ่มจำนวนชิ้นส่วนย่อยมากขึ้น ดังนั้นในการทำอีเรเซอร์โค้ดควรเลือกจำนวนชิ้นส่วนย่อยให้มากที่สุดที่เป็นไปได้ ในทางตรงกันข้าม หาก ค่า μ*S < 1 (ในกรณีนี้คือเมื่อ S มีค่าน้อยกว่า 4) ไม่ควรเพิ่มจำนวนชิ้นส่วนย่อยเพราะจะให้ได้ค่าปัจจัย P ลดลง

นอกจากนี้จากกราฟสังเกตว่าในช่วงแรกที่จำนวนชิ้นส่วนย่อยมีค่าน้อย (เมื่อจำนวนชิ้นส่วนย่อยไม่เกิน 6 ชิ้น) ค่าปัจจัยการยืด (S) ที่มีค่าน้อยจะให้ค่าปัจจัย P สูงกว่ามาก ดังนั้นควรเลือกค่าปัจจัยการยืด (S) ที่มีค่าน้อยที่สุด (ในกรณีนี้คือ S=2) จะให้ผลที่ดีกว่า

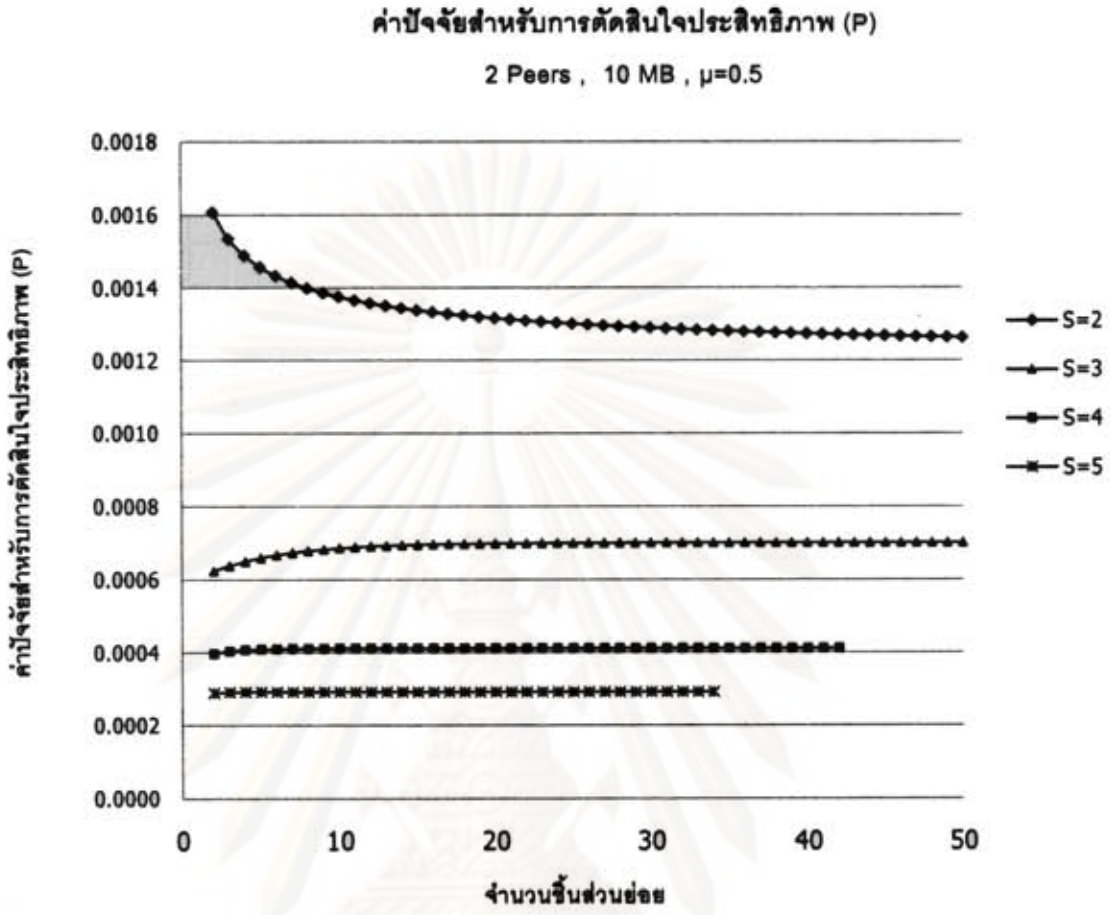
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 25 กราฟของปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) เทียบกับจำนวนชิ้นส่วนย่อย (b) และค่าปัจจัยการยืด (S) โดยใช้จำนวนเครื่องในระบบ 2 เครื่องและขนาดไฟล์ 10 MB เมื่อสภาพพร้อมใช้งานของเพียร์ (μ) เท่ากับ 0.3

5.2.1.2 **สภาพพร้อมใช้งานของเพียร์ μ = 0.5**

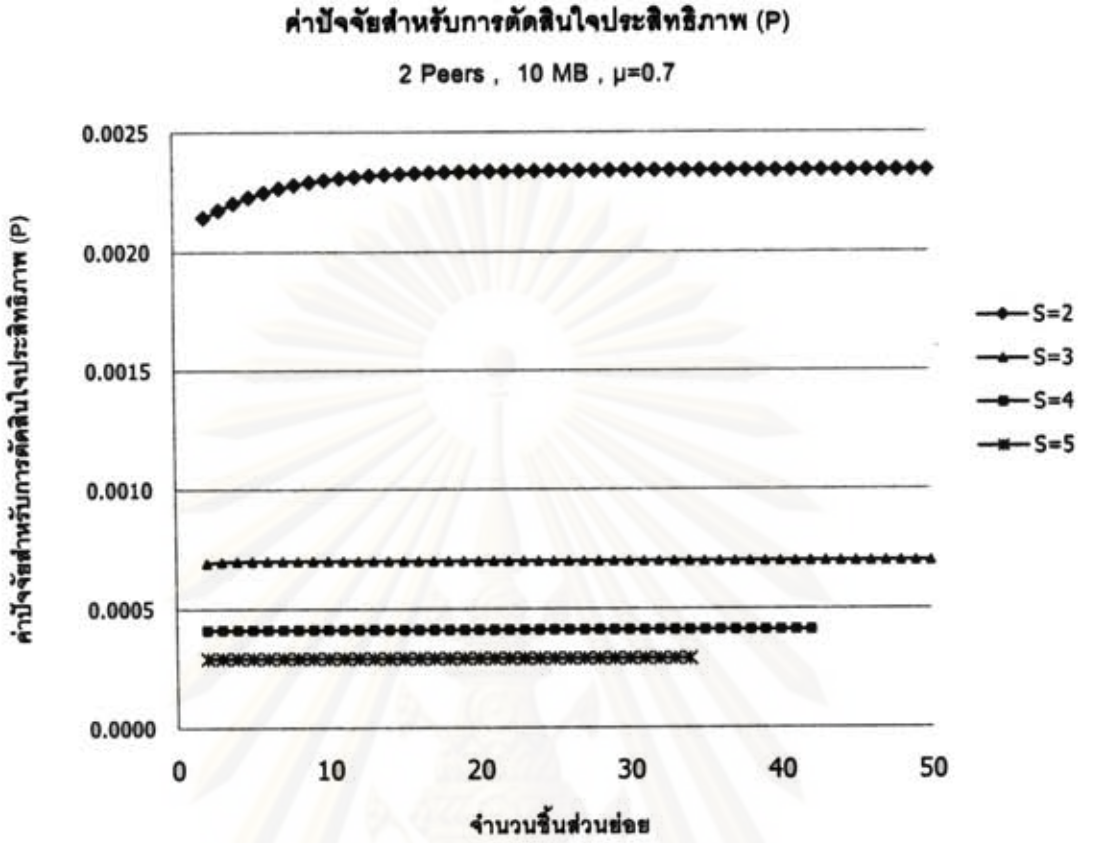
พิจารณากราฟของค่าปัจจัย P ในรูปที่ 26 ซึ่งมีสภาพพร้อมใช้งานของเพียร์ μ = 0.5 แสดงผลในทำนองเดียวกับหัวข้อ 5.2.1.1 คือในกรณีที่ค่า μ*S > 1 (ในกรณีนี้คือเมื่อ S มีค่าตั้งแต่ 2 เป็นต้นไป) ค่าปัจจัย P จะมากขึ้นเมื่อเพิ่มจำนวนชิ้นส่วนย่อยมากขึ้น อย่างไรก็ตามผลการวิเคราะห์มีความแตกต่างจากหัวข้อที่แล้วคือถึงแม้ว่าค่า μ*S < 1 (ในกรณีนี้คือเมื่อ S มีค่าน้อยกว่า 2) จะทำให้ได้ค่าปัจจัย P ลดลงเมื่อเพิ่มจำนวนชิ้นส่วนย่อย แต่ค่าปัจจัย P โดยรวมทั้งหมดยังคงมีค่ามากกว่าในกรณีที่ค่า μ*S > 1 ที่เป็นเช่นนี้เนื่องจากเมื่อเพิ่มค่าปัจจัยการยืด (S) มากส่งผลให้ใช้เวลาในการสำรองข้อมูลเพิ่มขึ้นมากเมื่อเทียบกับสภาพพร้อมใช้งานของไฟล์ ดังนั้นเมื่อเปรียบเทียบค่าปัจจัย P ซึ่งเป็นอัตราส่วนระหว่างสภาพพร้อมใช้งานของไฟล์และเวลาที่ใช้แล้วแสดงให้เห็นว่าการใช้ค่าปัจจัยการยืด (S) ที่มากไม่เหมาะสมและคุ้มค่า ดังนั้นในกรณีนี้ควรเลือกค่าปัจจัยการยืด (S) ที่น้อยที่สุด รวมถึงใช้จำนวนชิ้นส่วนย่อย (b) น้อยที่สุดด้วยจึงจะมีความเหมาะสม



รูปที่ 26 กราฟของปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) เทียบกับจำนวนชิ้นส่วนย่อย (b) และค่าปัจจัยการยืด (S) โดยใช้จำนวนเครื่องในระบบ 2 เครื่องและขนาดไฟล์ 10 MB เมื่อสภาพพร้อมใช้งานของเพียร์ (μ) เท่ากับ 0.5

5.2.1.3 **สภาพพร้อมใช้งานของเพียร์** μ = 0.7

พิจารณากราฟของค่าปัจจัย P ในรูปที่ 27 ซึ่งมีสภาพพร้อมใช้งานของเพียร์ μ = 0.7 โดยวิเคราะห์ผลในทำนองเดียวกับหัวข้อ 5.2.1.1 พบว่าในกรณีนี้ค่า μ*S > 1 สำหรับทุกค่า S ที่กำหนด จากกราฟแสดงให้เห็นว่าควรเลือกค่าปัจจัยการยืด (S) ที่มีค่าน้อยที่สุด และเลือกจำนวนชิ้นส่วนย่อย (b) ให้มากที่สุดที่เป็นไปได้ จะทำให้ได้ค่าปัจจัย P มากที่สุด โดยผลที่ได้นี้มีสาเหตุมาจากเมื่อ μ*S > 1 การเลือกค่า S ที่มากจะส่งผลกับพจน์ของเวลาที่ใช้ในการสำรองข้อมูลมากกว่าสภาพพร้อมใช้งานของไฟล์ ทำให้ค่าปัจจัย P ที่ได้มีค่าน้อยกว่า ดังนั้นการเลือกใช้ค่าปัจจัยการยืด (S) ที่มีค่ามากจึงไม่เหมาะสมสำหรับกรณีนี้



รูปที่ 27 กราฟของปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) เทียบกับจำนวนชิ้นส่วนย่อย (b) และค่าปัจจัยการยึด (S) โดยใช้จำนวนเครื่องในระบบ 2 เครื่องและขนาดไฟล์ 10 MB เมื่อสภาพพร้อมใช้งานของเพียร์ (μ) เท่ากับ 0.7

จากผลการวิเคราะห์ข้างต้นในแต่ละกรณี สามารถเสนอข้อสรุปและแนวทางในการเลือกพารามิเตอร์ที่เหมาะสมโดยรวมสำหรับทุกกรณีได้ดังนี้

1) ควรเลือกค่าปัจจัยการยึด (S) ที่มีค่าน้อยจะให้ผลที่ดีกว่า โดยเลือกจำนวนชิ้นส่วนย่อยดังนี้
   - หาก $\mu^*S < 1$ ไม่ควรเพิ่มจำนวนชิ้นส่วนย่อย ควรใช้จำนวนชิ้นส่วนย่อยที่น้อยที่สุดที่เป็นไปได้
   - หาก $\mu^*S > 1$ ควรเพิ่มจำนวนชิ้นส่วนย่อยที่มากที่สุดที่เป็นไปได้
2) ในกรณีที่ค่าสภาพพร้อมใช้งานของเพียร์ (μ) มีค่ามาก เมื่อเลือกปัจจัยการยึด (S) ที่มีค่าน้อย และเพิ่มจำนวนชิ้นส่วนย่อยมากขึ้น ยังคงมีความเหมาะสมมากกว่า(ให้ค่าปัจจัย P มากกว่า) การเลือกปัจจัยการยึด (S) มาก

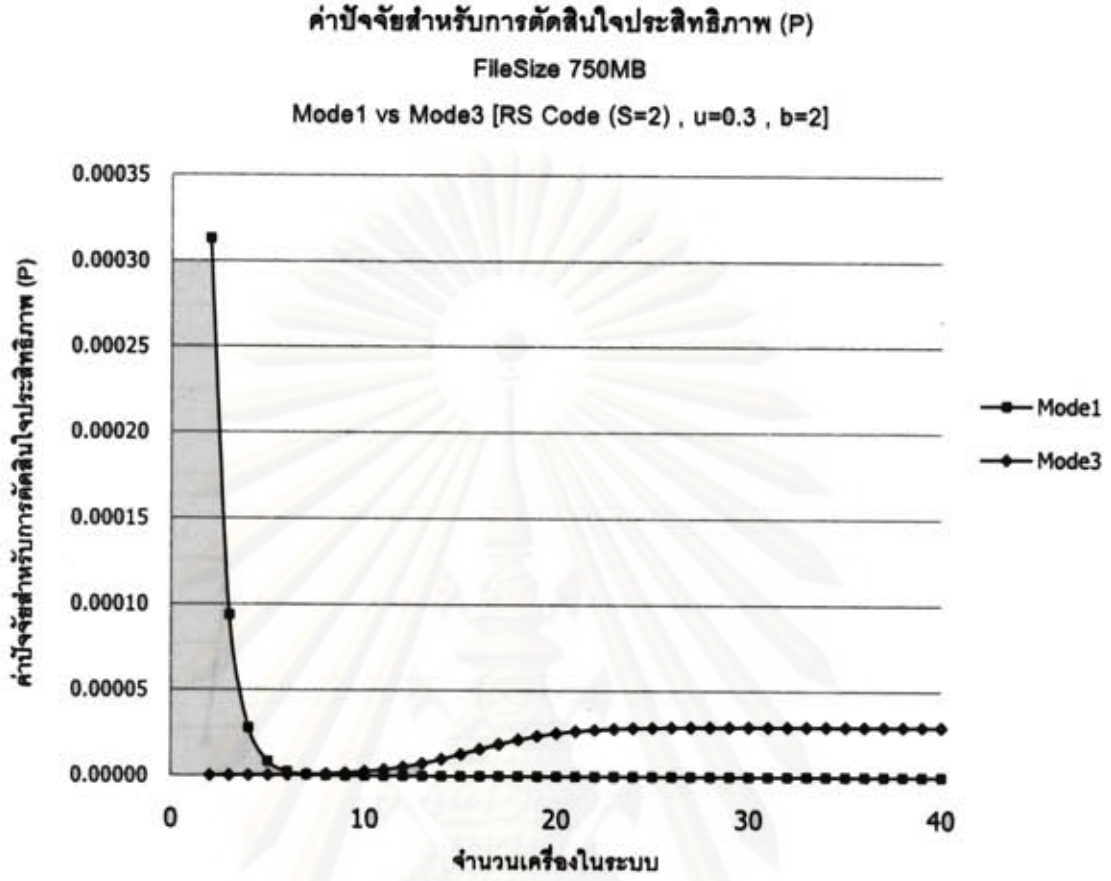
5.2.2 การวิเคราะห์และเปรียบเทียบประสิทธิภาพในการสำรองข้อมูลระหว่างขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยแต่ละวิธี

5.2.2.1 การเปรียบเทียบระหว่างขั้นตอนวิธีที่ 1 และ 3

จากแบบจำลองสำหรับการตัดสินใจประสิทธิภาพ (Performance Decisive Model, P) ในหัวข้อ 5.1 ขั้นตอนวิธีที่ไม่ได้ใช้อีเรเซอร์โค้ดจะมีค่าสภาพพร้อมใช้งานของไฟล์ A = μ^b การแบ่งชิ้นส่วนย่อยวิธีแรกซึ่งแบ่งตามจำนวนเครื่องในระบบจะมีจำนวนชิ้นส่วนย่อยเท่ากับจำนวนเครื่องในระบบ ดังนั้นค่าสภาพพร้อมใช้งานของไฟล์ A = μ^n หากพิจารณาค่าปัจจัย P ในขั้นตอนวิธีแรกอย่างคร่าวๆ จะพบว่าค่าปัจจัย P ของจำนวนเครื่องในระบบที่มากกว่าค่าจุดแบ่ง (Cut-Off Point) ย่อมน้อยกว่าบริเวณที่มากกว่าค่าจุดแบ่ง เนื่องจากเวลาที่ใช้ในการสำรองข้อมูลเมื่อเพิ่มจำนวนเครื่องมากกว่าจุดนี้จะไม่ลดลงอีกต่อไปในขณะที่จำนวนเครื่อง (n) มีค่าเพิ่มขึ้นทำให้ค่าสภาพพร้อมใช้งานของไฟล์ ($A = \mu^n$ ซึ่ง μ มีค่าน้อยกว่า 1) มีค่าลดลงอย่างรวดเร็ว

พิจารณาการสำรองข้อมูลในวิธีแรกที่จำนวนเครื่องน้อยกว่าค่าจุดแบ่งจะพบว่าเวลาที่ใช้ลดลงเมื่อจำนวนเครื่องมากขึ้น รวมถึงค่าสภาพใช้งานของไฟล์ลดลงเช่นกัน เมื่อเปรียบเทียบกับการสำรองข้อมูลด้วยขั้นตอนวิธีที่สามโดยใช้อีเรเซอร์โค้ดซึ่งใช้เวลาในการสำรองข้อมูลที่ขนาดไฟล์เดียวกันมากกว่า พบว่าค่าสภาพพร้อมใช้งานของไฟล์ไม่แปรผันตามจำนวนเครื่องในระบบแต่เวลาที่ใช้จะลดลงเมื่อจำนวนเครื่องเพิ่มขึ้น แนวโน้มของค่าปัจจัย P ของการแบ่งทั้งสองขั้นตอนวิธีแสดงดังกราฟในรูปที่ 28 ซึ่งแสดงการเปรียบเทียบค่าปัจจัย P ของการสำรองข้อมูลในวิธีแรกและวิธีที่สามซึ่งใช้ RS Code (S=2) สภาพพร้อมใช้งานของเพียร์ μ=0.3 และกำหนดจำนวนชิ้นส่วนย่อย b=2 ตามแนวทางการเลือกพารามิเตอร์ที่เหมาะสมในหัวข้อ 5.2.1 คือ ในกรณีที่ค่า $\mu^*S < 1$ ควรเลือกจำนวนชิ้นส่วนย่อยที่น้อยที่สุด ในที่นี้คือจำนวน 2 ชิ้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

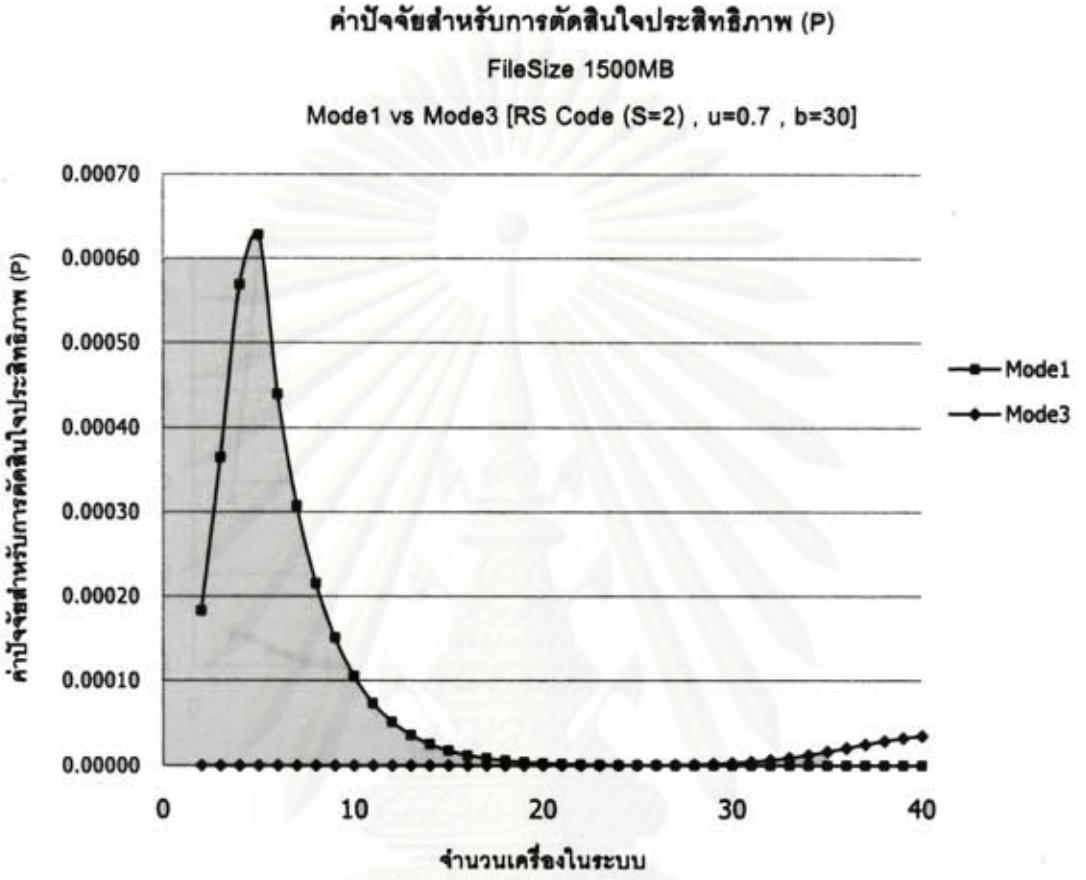


รูปที่ 28 กราฟแสดงการเปรียบเทียบปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) ระหว่างวิธีแรกและวิธีที่สามซึ่งใช้ RS Code (S=2, μ=0.3, b=2) โดยเทียบกับจำนวนเครื่องในระบบ (n) และใช้ขนาดไฟล์ 750 MB

จากกราฟจะเห็นว่าในช่วงแรกเมื่อใช้จำนวนเครื่องในระบบน้อย ค่าปัจจัย P ในวิธีแรกมีค่ามากกว่าวิธีที่สามมาก เมื่อเพิ่มจำนวนเครื่องในระบบมากขึ้นจะทำให้วิธีแรกมีค่าปัจจัย P ลดลงอย่างรวดเร็วและวิธีที่สามมีค่าปัจจัย P เพิ่มขึ้นมากกว่า โดยมีจุดที่เริ่มเปลี่ยนแนวโน้มของค่าปัจจัย P เมื่อเพิ่มจำนวนเครื่องในระบบประมาณ 6-7 เครื่อง

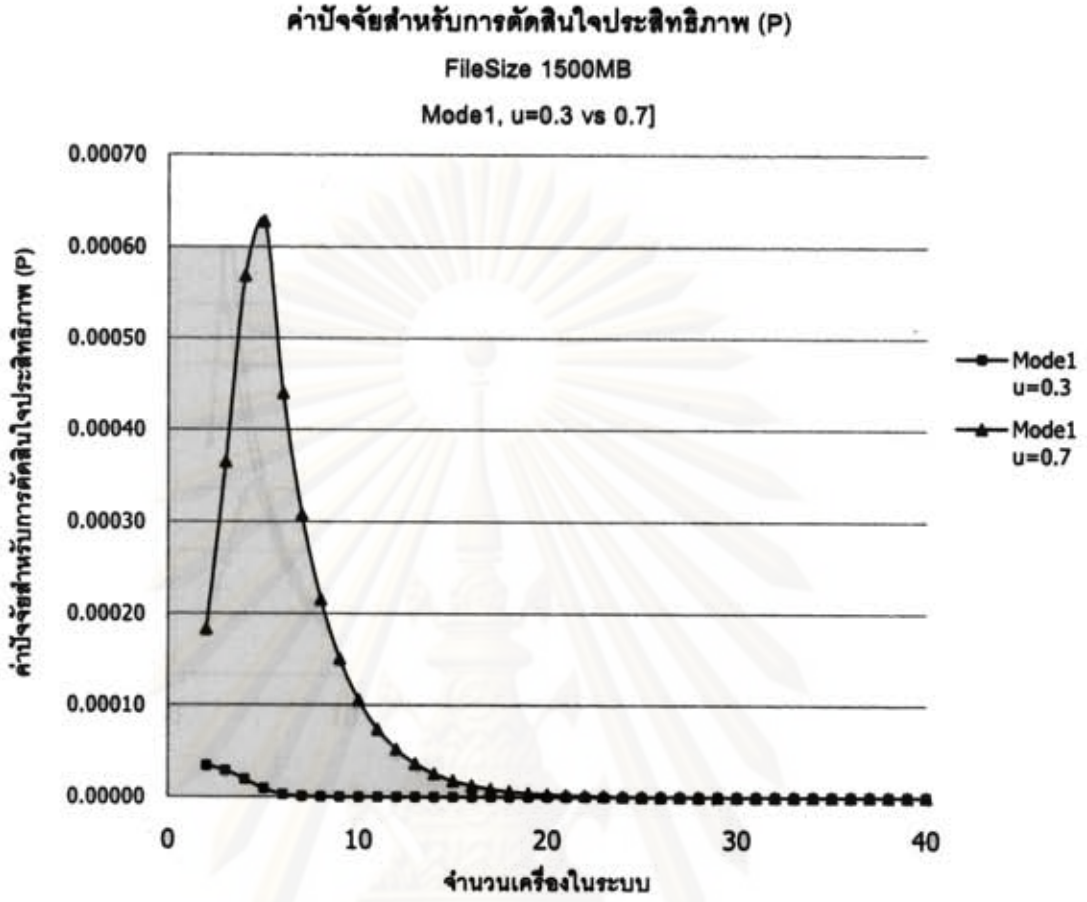
ในรูปที่ 29 แสดงการเปรียบเทียบค่าปัจจัย P ของการสำรองข้อมูลในวิธีแรกและวิธีที่สามซึ่งใช้ RS Code (S=2) สภาพพร้อมใช้งานของเพียร์ μ=0.7 และกำหนดจำนวนชิ้นส่วนย่อย b=30 เนื่องจากค่า μ*S > 1 จึงเลือกจำนวนชิ้นส่วนย่อยที่มีค่ามากที่สุดที่เป็นไปได้ ในที่นี้ได้เลือกจำนวนชิ้นส่วนย่อย 30 ชิ้น ผลการวิเคราะห์จากกราฟให้ผลในทำนองเดียวกับเมื่อเลือกใช้ RS Code ที่มี S=2, μ=0.3 และ b=2 คือวิธีแรกมีค่าปัจจัย P มากกว่าวิธีที่สามมากในช่วงแรกที่จำนวนเครื่องน้อย และเมื่อเพิ่มจำนวนเครื่องในระบบมากขึ้นวิธีที่สามจะให้ค่าปัจจัย P มากกว่า โดยจุดที่เริ่มเปลี่ยนแนวโน้มของค่าปัจจัย P เมื่อเพิ่มจำนวนเครื่องในระบบต่างกันคืออยู่ที่

ประมาณ 21-22 เครื่อง นอกจากนี้สังเกตจากกราฟจะพบว่าจำนวนเครื่องที่เหมาะสมที่จะให้ค่าปัจจัย P มากที่สุดยังแตกต่างจากในกรณีก่อนหน้าซึ่งเป็นจำนวนเครื่องที่น้อยที่สุด



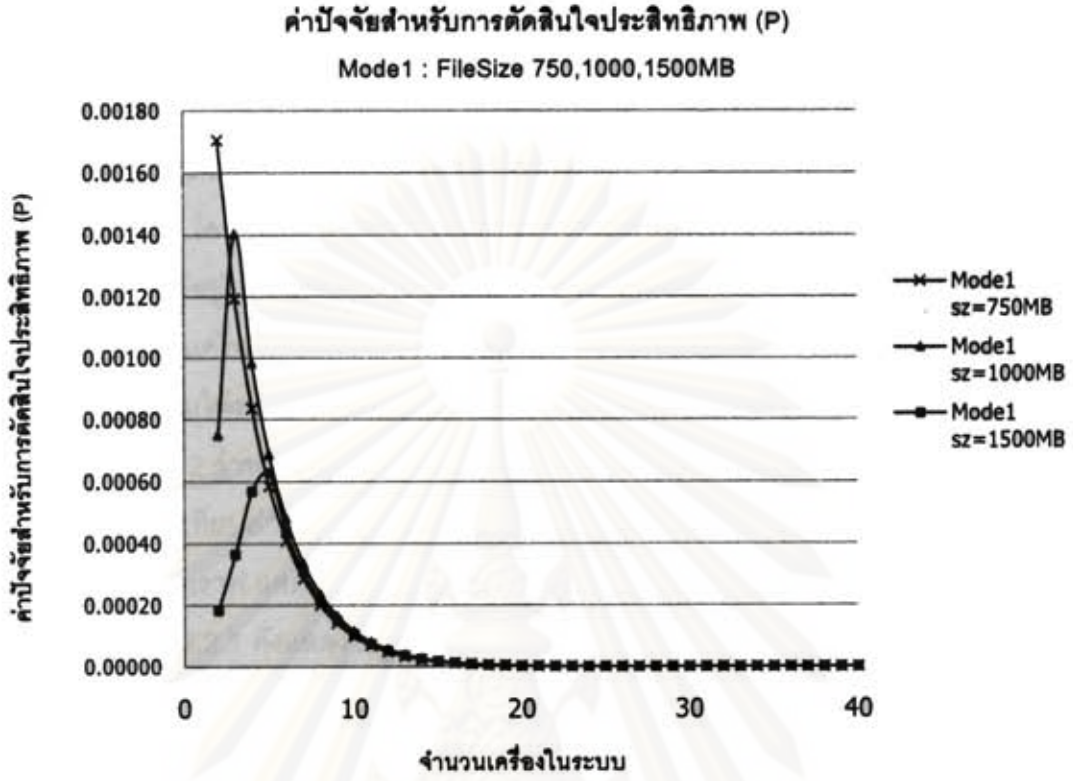
รูปที่ 29 กราฟแสดงการเปรียบเทียบปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) ระหว่างวิธีแรกและวิธีที่สามซึ่งใช้ RS Code (S=2, μ=0.7, b=30) โดยเทียบกับจำนวนเครื่องในระบบ (n) และใช้ขนาดไฟล์ 1500 MB

หากพิจารณาการสำรองข้อมูลด้วยขั้นตอนวิธีแรกเมื่อสภาพพร้อมใช้งานของเพียร์ (μ) แตกต่างกัน ดังกราฟในรูปที่ 30 ซึ่งเปรียบเทียบค่าปัจจัย P กับจำนวนเครื่องในระบบ (n) เมื่อมีค่าสภาพพร้อมใช้งานของเพียร์ μ=0.3 และ 0.7 โดยใช้ขนาดไฟล์ 1500 MB แสดงให้เห็นว่าจำนวนเครื่องในระบบที่เหมาะสมที่จะให้ค่าปัจจัย P มากที่สุดนั้นขึ้นอยู่กับค่าสภาพพร้อมใช้งานของเพียร์ (μ) และจะเห็นได้ชัดเจนเมื่อมีค่าสภาพพร้อมใช้งานของเพียร์สูง (μ=0.7 จากกราฟ) รูปที่ 31 แสดงจำนวนเครื่องในระบบที่เหมาะสมที่จะให้ค่าปัจจัย P มากที่สุด ซึ่งก็คือจำนวนเครื่องที่เท่ากับค่าจุดแบ่ง (Cut-Off Point) สำหรับแต่ละขนาดไฟล์ โดยมีค่าเท่ากับ 2, 3 และ 5 เครื่อง สำหรับขนาดไฟล์ 750 MB, 1000 MB และ 1500 MB ตามลำดับ (การคำนวนค่าจุดแบ่ง (Cut-Off Point) สำหรับแต่ละขนาดไฟล์สามารถคำนวนได้จากสมการ (5) ในหัวข้อ 4.1)



รูปที่ 30 กราฟแสดงการเปรียบเทียบปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) ของวิธีแรก ระหว่างค่าสภาพพร้อมใช้งานของเพียร์ μ=0.3 และ 0.7 โดยเทียบกับจำนวนเครื่องในระบบ (n) และใช้ขนาดไฟล์ 1500 MB

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 31 กราฟแสดงการเปรียบเทียบปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) ของวิธีแรก เทียบกับจำนวนเครื่องในระบบ (n) ที่ค่าสภาพพร้อมใช้งานของเพียร์ μ= 0.7 โดยใช้ขนาดไฟล์ 750 MB, 1000 MB และ 1500 MB

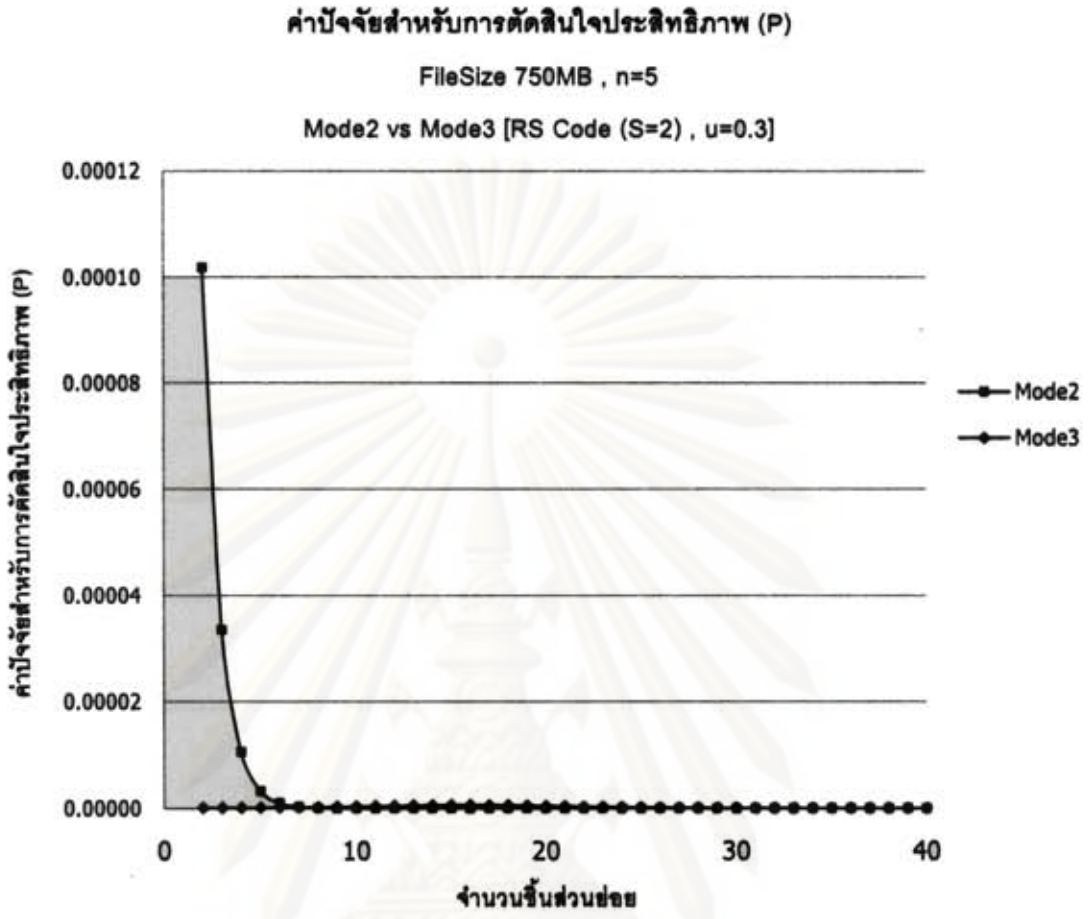
จากผลการวิเคราะห์สามารถเสนอแนวทางการเลือกขั้นตอนวิธีในการแบ่งชิ้นย่อยที่เหมาะสมได้ดังนี้

1) การสำรองข้อมูลควรเลือกขั้นตอนวิธีแรกในการสำรองข้อมูลโดยใช้จำนวนเครื่องในระบบจำนวนน้อยจะให้ค่าปัจจัย P มากกว่าหรือมีความเหมาะสมมากกว่า โดยใช้จำนวนเครื่องที่เหมาะสมเท่ากับค่าจุดแบ่ง (Cut-Off Point) สำหรับแต่ละขนาดไฟล์
2) หากต้องการสำรองข้อมูลโดยใช้จำนวนเครื่องในระบบ (n) จำนวนมาก ควรเลือกขั้นตอนวิธีที่สามคือการใช้อีเรเซอร์โค้ดจะมีความเหมาะสมมากกว่า โดยการกำหนดค่าพารามิเตอร์สำหรับขั้นตอนวิธีที่สามอาศัยแนวทางการเลือกพารามิเตอร์ที่เหมาะสมจากหัวข้อ 5.2.1

5.2.2.2 การเปรียบเทียบระหว่างขั้นตอนวิธีที่ 2 และ 3

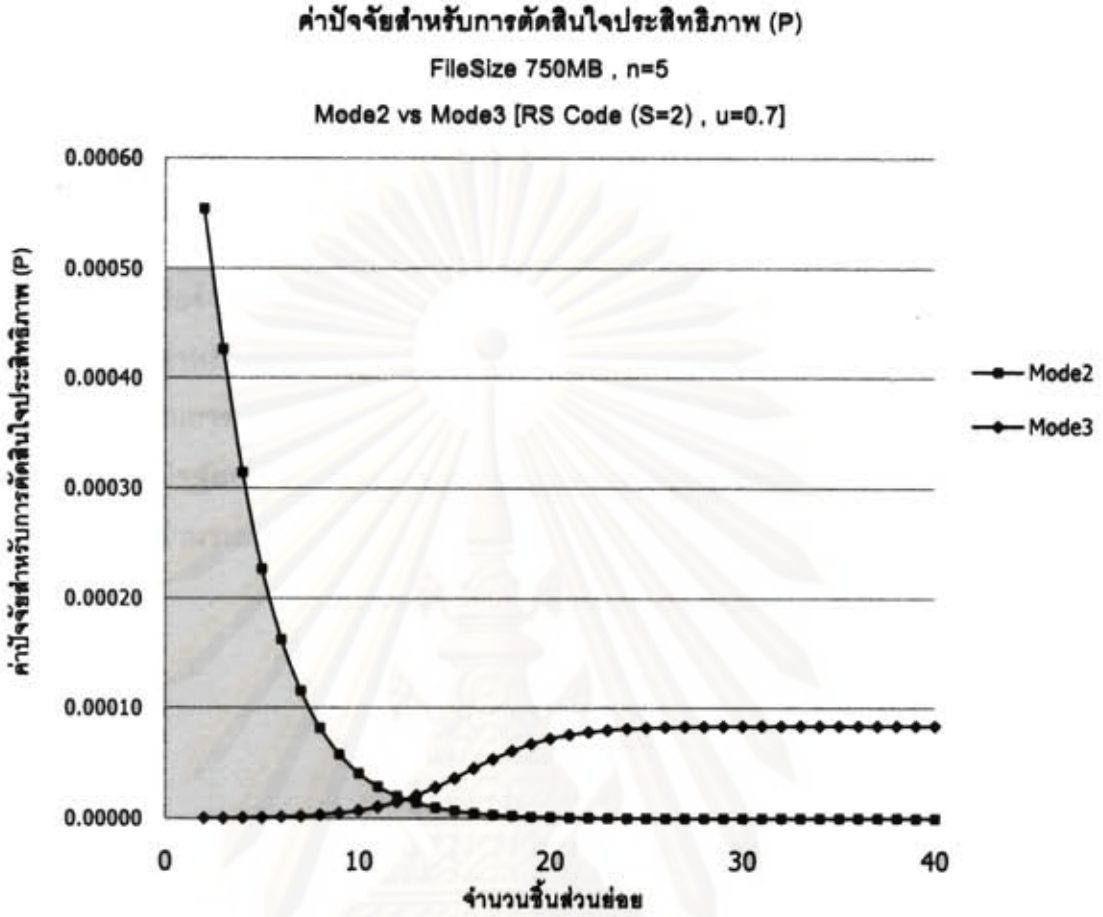
จากสมการของแบบจำลองสำหรับการตัดสินใจประสิทธิภาพ (Performance Decisive Model, P) ในหัวข้อ 5.1 ขั้นตอนวิธีที่ 2 มีพจน์ที่เป็นเวลาในการสำรองข้อมูล $T_2(sz,n,bs)$ เป็นฟังก์ชันที่แปรผันตามขนาดไฟล์ (sz), จำนวนเครื่องในระบบ (n) และขนาดชิ้นส่วนย่อย (bs) โดยขนาดของชิ้นส่วนย่อย (bs) สามารถคำนวณได้จากจำนวนชิ้นส่วนย่อย (b) ความสัมพันธ์ที่ได้อยู่ในรูป bs = sz/b ดังนั้นสามารถเขียนเวลาที่ใช้ได้ใหม่ในรูปฟังก์ชันของขนาดไฟล์ (sz), จำนวนเครื่องในระบบ (n) และจำนวนชิ้นส่วนย่อย (b) คือ $T_2(sz,n,sz/b)$ สำหรับวิธีที่ 2 นี้มีค่าสภาพพร้อมใช้งานของไฟล์ $A = \mu^b$ ซึ่งขึ้นอยู่กับจำนวนชิ้นส่วนย่อย (b) เหมือนวิธีแรก จะเห็นได้ว่าในวิธีที่ 2 จำนวนชิ้นส่วนย่อย (b) มีผลทั้งกับเวลาที่ใช้และสภาพพร้อมใช้งานของไฟล์ ดังนั้นการเปรียบเทียบค่าปัจจัย P ระหว่างวิธีที่ 2 และ 3 นี้ทำได้โดยการพิจารณาปรับค่าจำนวนชิ้นส่วนย่อย (b) กราฟ แสดงการเปรียบเทียบค่าปัจจัย P ของวิธีที่ 2 กับวิธีที่ 3 ให้ผลในทำนองเดียวกับหัวข้อ 5.2.2.1 ดังแสดงในรูปที่ 32 และรูปที่ 33 ซึ่งเลือกใช้จำนวนเครื่องในระบบ 5 เครื่อง ขนาดไฟล์ 750 MB และวิธีที่ 3 ใช้ RS Code ที่มีค่า S=2, μ= 0.3 และ 0.7 ตามลำดับ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 32 กราฟแสดงการเปรียบเทียบปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) ระหว่างวิธีที่ 2 และวิธีที่ 3 ซึ่งใช้ RS Code (S=2, μ=0.3) โดยเทียบกับจำนวนชิ้นส่วนย่อย (b) และใช้ขนาดไฟล์ 750 MB บนจำนวนเครื่อง 5 เครื่อง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 33 กราฟแสดงการเปรียบเทียบปัจจัยสำหรับการตัดสินใจประสิทธิภาพ (P) ระหว่างวิธีที่ 2 และวิธีที่ 3 ซึ่งใช้ RS Code (S=2, μ=0.7) โดยเทียบกับจำนวนชิ้นส่วนย่อย (b) และใช้ขนาดไฟล์ 750 MB บนจำนวนเครื่อง 5 เครื่อง

5.3 อภิปรายผลการวิจัย

ขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยแต่ละวิธีที่กล่าวมานั้นใช้เวลาในการสำรองข้อมูลและมีสภาพพร้อมใช้งานของไฟล์แตกต่างกัน แบบจำลองของเวลาในการสำรองข้อมูลสามารถนำมาประมาณเวลาที่ใช้ในแต่ละวิธีได้ นอกจากนี้แบบจำลองสำหรับการตัดสินใจประสิทธิภาพ (Performance Decisive Model, P) ที่ได้เสนอในงานวิจัยนี้ สามารถนำมาพิจารณาเพื่อเปรียบเทียบประสิทธิภาพของแต่ละขั้นตอนวิธีโดยมองประสิทธิภาพทั้งสองด้านพร้อมกัน ขั้นตอนวิธีที่ให้ค่าปัจจัยสำหรับการตัดสินใจประสิทธิภาพสูงย่อมมีความเหมาะสมในการสำรองข้อมูลมากที่สุด โดยผลการทดลองและวิเคราะห์สามารถสรุปได้ดังนี้

1. สำหรับวิธีการแบ่งชิ้นส่วนย่อยตามจำนวนเครื่องในระบบ สามารถทำนายจำนวนเครื่องในระบบที่มากเพียงพอที่จะใช้เวลาในการสำรองข้อมูลน้อยที่สุดด้วยสมการ

ของเส้นขีดแบ่ง (Threshold Line) ซึ่งเป็นสมการเวกเตอร์ในรูปฟังก์ชันของขนาดไฟล์

2. สำหรับรูปแบบการแบ่งชิ้นส่วนย่อยโดยกำหนดขนาดชิ้นส่วนย่อยคงที่มีกราฟของเวลาในการสำรองข้อมูลที่ต่างจากวิธีอื่นคือมีค่าเวลาที่สูงและต่ำผิดปกติ เนื่องมาจากการกระจายชิ้นส่วนข้อมูลไปยังเครื่องในระบบที่ไม่เท่าเทียมกันและข้อจำกัดของแบนด์วิดท์ของเครือข่าย (Network Bandwidth)
3. สำหรับการแบ่งชิ้นส่วนย่อยโดยใช้อีเรเซอร์โค้ด การเข้ารหัสด้วย LT Code ใช้เวลาในการสำรองข้อมูลน้อยกว่า RS Code ค่าสภาพพร้อมใช้งานของเพียร์ (μ) และค่าปัจจัยการยึด (S) เป็นพารามิเตอร์ที่สำคัญและส่งผลกับจำนวนชิ้นส่วนย่อยและจำนวนเครื่องในระบบที่เหมาะสม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6
บทสรุปของการวิจัย

ในบทนี้จะกล่าวถึงสิ่งที่ได้จากการวิจัย แนวทางการวิจัยต่อ และบทสรุป ดังนี้

6.1 สิ่งที่ได้จากการวิจัย (Contribution)

สิ่งที่ได้จากการวิจัยนี้ ได้แก่

1. อธิบายขั้นตอนวิธีการสำรองข้อมูลโดยใช้ขั้นตอนวิธีการแบ่งชิ้นส่วนย่อยแบบต่างๆ
2. นำเสนอแบบจำลองของเวลาที่ใช้ในการประมาณเวลาในการสำรองข้อมูลแต่ละขั้นตอนวิธีการแบ่งชิ้นส่วนย่อย
3. นำเสนอแบบจำลองสำหรับการตัดสินใจประสิทธิภาพ (Performance Decisive Model, P) เพื่อเป็นแนวทางในการตัดสินใจเลือกขั้นตอนวิธีในการสำรองข้อมูลที่เหมาะสมและมีประสิทธิภาพทั้งในด้านเวลาและสภาพพร้อมใช้งานของข้อมูล)
4. นำเสนอแนวทางการเลือกพารามิเตอร์ที่เหมาะสมในแต่ละขั้นตอนวิธีการแบ่งชิ้นส่วนย่อย

6.2 แนวทางการวิจัยต่อ

ในงานวิจัยนี้ได้ศึกษาและวิเคราะห์ในด้านเวลาในการสำรองข้อมูลและสภาพพร้อมใช้งานของไฟล์ซึ่งเป็นประสิทธิภาพที่ผู้ใช้ส่วนใหญ่ให้ความสำคัญ อย่างไรก็ตามงานวิจัยนี้สามารถนำไปศึกษาต่อในแง่ของประสิทธิภาพด้านอื่น เช่น ความปลอดภัยของข้อมูล, รูปแบบการเข้ารหัสข้อมูล, การกระจายเก็บชิ้นส่วนย่อยของข้อมูลภายในระบบ เป็นต้น

นอกจากนี้ยังมีประเด็นที่ต้องศึกษาต่ออีกหลายประเด็นด้วยกัน เช่น ในงานวิจัยนี้ได้วิเคราะห์สภาพพร้อมใช้งานของไฟล์โดยอยู่บนสมมติฐานที่ว่าสภาพพร้อมใช้งานของเพียร์ (μ) ของทุกเครื่องในระบบมีค่าเท่ากัน ซึ่งในสถานการณ์จริง เครื่องภายในระบบ (เพียร์) ย่อมมีการเข้า-ออกตลอดเวลาทำให้มีสภาพพร้อมใช้งานของเพียร์ไม่คงที่และไม่เท่ากันทุกเครื่อง โดยสามารถนำงานวิจัยนี้ไปศึกษาต่อในกรณีนี้ได้

6.3 บทสรุป

เราสามารถประมาณเวลาที่ใช้ในการสำรองข้อมูลได้จากแบบจำลองของเวลาในการสำรองข้อมูลในแต่ละขั้นตอนวิธีการแบ่งชิ้นส่วนย่อย รวมถึงเลือกขั้นตอนวิธีการแบ่งชิ้นส่วนย่อย

และพารามิเตอร์ที่เหมาะสมในแต่ละวิธีของการสำรองข้อมูลได้โดยใช้ประโยชน์จากแบบจำลองสำหรับการตัดสินใจประสิทธิภาพ (Performance Decisive Model, P)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] M. Luby. Erasure Coding Approaches [Online]. Available from: http://www.usenix.org/events/usits03/tech/luby/tsld015.htm [2009, Oct 20]

[2] M. Riley and I. Richardson. An introduction to Reed-Solomon codes: principles, architecture and implementation [Online]. Available from: http://www.cs.cmu.edu/afs/cs.cmu.edu/project/pscico-guyb/realworld/www/reedsolomon/reed_solomon_codes.html [2009, May 3]

[3] M. Luby. Erasure Codes [Online]. Available from: http://www.usenix.org/events/usits03/tech/luby/sld011.htm [2009, Oct 20]

[4] W.K. Lin, D.M. Chiu, and Y.B. Lee. Erasure Code Replication Revisited. P2P'04: Proceedings of the 4th International Conference on Peer-to-Peer Computing, pp. 90-97, Zürich, Switzerland: IEEE Computer Society, 2004.

[5] K. John, et al. OceanStore: An Architecture for Global-Scale Persistent Storage. ASPLOS'00: Proceedings of the 9th International Conference On Architectural Support for Programming Languages and Operating Systems, pp. 190-201, New York: ACM, 2000.

[6] B. Christopher, B. Kenneth, S. Arvind, and T. Stanley. pStore: A Secure Peer-to-Peer Backup System. Technical Memo MIT-LCS-TM-632, Massachusetts Institute of Technology Laboratory for Computer Science, 2002.

[7] L. Martin, Z. Han, and T. Kian-Lee. PeerStore: Better Performance by Relaxing in Peer-to-Peer Backup. P2P'04: Proceedings of the 4th International Conference on Peer-to-Peer Computing, pp. 72-79, Zürich, Switzerland: IEEE Computer Society, 2004.

[8] F. Morcos, T. Chantem, P. Little, T. Gasiba, and D. Thain. iDIBS: An Improved Distributed Backup System. ICPADS'06: Proceedings of the 12th International Conference on Parallel and Distributed Systems, pp. 10, Minneapolis, Minnesota: IEEE Computer Society, 2006.

[9] J. Amir, A. Tony, B. Armin, L. Guy, L. Roman, and R. Matei. vanDisk: An Exploration in Peer-To-Peer Collaborative Back-up Storage. CCECE'07: 20th Canadian

Conference on Electrical and Computer Engineering, pp. 219-222, Vancouver, BC, Canada: IEEE Computer Society, 2007.

[10] B. Ranjita, S. Stefan, and V. Geoffrey M. Replication Strategies for Highly Available Peer-to-peer Storage Systems. Technical Report CS2002-0726, UCSD, 2002.

[11] J. S. Plank, J. Luo, C. D. Schuman, L. Xu and Z. Wilcox-O'Hearn. A Performance Evaluation and Examination of Open-Source Erasure Coding Libraries For Storage. FAST'09: Proceedings of the 7th USENIX Conference on File and Storage Technologies, pp. 253-265, San Francisco, CA: USENIX Association, 2009.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก.
ระบบสำรองข้อมูลที่มีอยู่ในปัจจุบัน

เนื่องจากในปัจจุบันมีระบบสำรองข้อมูลที่ใช้ทั่วไปอยู่เป็นจำนวนมาก ในส่วนนี้จะกล่าวถึงระบบสำรองข้อมูลที่อยู่ในรูปของโปรแกรมและซอฟต์แวร์เชิงพาณิชย์ (Commercial Software) ดังนี้

<u>ประเภทของการสำรองข้อมูล</u>

โปรแกรมและซอฟต์แวร์เชิงพาณิชย์ที่พบในปัจจุบันส่วนใหญ่มีลักษณะของการสำรองข้อมูลในรูปแบบต่างๆ ดังนี้

1. Full Backup เป็นการสำรองข้อมูลทั้งหมดตามที่ผู้ใช้กำหนด อาจเป็นข้อมูลทั้งหมดในฮาร์ดไดรฟ์ (Hard Drive) ซึ่งรวมถึงไฟล์ (File),โฟลเดอร์ (Folder) และแอปพลิเคชัน (Application) หรืออาจเป็นข้อมูลบางส่วนที่เลือก
2. Incremental Backup เป็นการสำรองข้อมูลเฉพาะข้อมูลส่วนที่มีการเปลี่ยนแปลงเมื่อเทียบกับการสำรองข้อมูล (ทั้งรูปแบบ Full Backup และ Incremental Backup) ครั้งหลังสุดเท่านั้น ทำให้การสำรองข้อมูลในแต่ละครั้งมีความรวดเร็วและใช้พื้นที่ในหน่วยความจำน้อยกว่า แต่ในการกู้คืนข้อมูลอาจใช้เวลานาน เนื่องจากต้องเรียกคืนข้อมูลจากการสำรองข้อมูลครั้งก่อนหน้าทั้งในส่วนของ Full Backup และ Incremental Backup ทุกครั้ง การใช้งาน Incremental Backup มีความเหมาะสมเมื่อทำการสำรองข้อมูลอย่างสม่ำเสมอและมีพื้นที่ในหน่วยความจำน้อย
3. Differential Backup เป็นการสำรองข้อมูลเฉพาะข้อมูลส่วนที่มีการเปลี่ยนแปลงเมื่อเทียบกับการสำรองข้อมูลในรูปแบบ Full Backup ครั้งหลังสุด ซึ่งจะทำให้ใช้เวลานานในการสำรองข้อมูลและใช้พื้นที่ในหน่วยความจำมากกว่า แต่การกู้คืนข้อมูลสามารถทำได้รวดเร็วกว่า เนื่องจากการสำรองข้อมูลด้วยวิธีนี้จะทำการดึงข้อมูลเฉพาะที่ได้สำรองข้อมูลในรูปแบบ Full Backup และ Differential Backup ครั้งหลังสุดเท่านั้น โดยการสำรองข้อมูลแบบนี้เหมาะสำหรับผู้ใช้ที่ต้องการความรวดเร็วในการกู้คืนข้อมูล

<u>คุณสมบัติหลักที่สำคัญของโปรแกรมสำรองข้อมูลในปัจจุบัน</u>

1. Data Compression เป็นการบีบอัดข้อมูลที่จะทำการสำรองลงในระบบ เพื่อให้สามารถสำรองข้อมูลในหน่วยความจำได้มากขึ้น

2. File Filtering เป็นการกรองไฟล์ข้อมูลเพื่อจำกัดขอบเขตในการสำรองข้อมูล โดยสามารถกำหนดตัวกรองจากชื่อไฟล์ข้อมูลหรือชนิดของไฟล์ข้อมูลได้
3. Backup Verification เป็นการตรวจสอบข้อมูลในขณะที่ทำการกู้คืนข้อมูลเพื่อตรวจสอบความถูกต้องกับไฟล์ข้อมูลต้นฉบับที่ทำการสำรองข้อมูล
4. Drive Spanning เป็นการสำรองข้อมูลลงบนสื่อหลายชนิด ในกรณีที่ข้อมูลที่ต้องการสำรองมีขนาดใหญ่กว่าหน่วยความจำที่มีอยู่

<u>ตัวอย่างคุณสมบัติของโปรแกรม, ซอฟต์แวร์เชิงพาณิชย์ที่ได้รับความนิยม</u>

1. Genie Backup Manager
    a. มีความสามารถในการสำรองข้อมูลจำพวกไปรษณีย์อิเล็กทรอนิกส์ (E-Mail) และรีจิสทรี (Registry) รวมทั้งการตั้งค่าต่าง ๆ ของเครื่องคอมพิวเตอร์
    b. สามารถทำการสำรองข้อมูลได้หลายชนิด
    c. ในด้านความปลอดภัยใช้การเข้ารหัสชื่อผู้ใช้ (Username) และรหัสผ่าน (Password)
    d. ส่วนต่อประสาน (Interface) ของโปรแกรมมีรูปแบบที่สามารถใช้งานได้ง่าย
    e. มีการทำงานโดยไม่รบกวนการทำงานปกติของผู้ใช้
2. Acronis True Image
    a. ลักษณะเด่นของโปรแกรมคือสามารถสร้างข้อกำหนดในการสำรองข้อมูลได้ เช่น กำหนดจำนวนพื้นที่ของหน่วยความจำในการสำรองข้อมูล จำนวนสำเนาของการสำรองข้อมูล และระยะเวลาในการรักษาข้อมูลที่ทำการสำรองนั้นไว้ เป็นต้น
    b. ใช้การทำ Disk Imaging ซึ่งเป็นการทำสำเนาข้อมูลของเครื่องทั้งหมด รวมถึงระบบปฏิบัติการ, แอปพลิเคชัน และการตั้งค่าภายในระบบ
    c. ในกรณีที่ฮาร์ดไดรฟ์ ล้มเหลวหรือถูกโจมตีด้วยไวรัสคอมพิวเตอร์ โปรแกรมสามารถกู้คืนข้อมูลได้อย่างรวดเร็ว
    d. นอกจากนี้โปรแกรมยังสามารถสร้างพื้นที่ในฮาร์ดไดรฟ์ สำหรับเป็นเขตปลอดภัย (Secure Zone) ในการสำรองข้อมูลโดยเฉพาะ เพื่อกำหนดการเข้ามาใช้งานพื้นที่ในส่วนนั้น
3. NTI Backup NOW! Deluxe Suite
    a. สามารถตั้งตารางเวลาในการสำรองข้อมูลได้

b. ใช้รหัสผ่านในการรักษาความปลอดภัย รวมทั้งมีการใช้รูปแบบการสำรองข้อมูลทั้งแบบ Incremental และ Differential Backup

c. สามารถเปลี่ยนตำแหน่งในหน่วยความจำในการสำรองข้อมูลได้

4. Powerbackup

a. ใช้การบีบอัดข้อมูล (Data Compression) ในการลดขนาดของข้อมูลที่จะทำการสำรอง

b. ใช้การกรองข้อมูล (File Filtering) ในการเลือกและจำกัดชื่อหรือชนิดของไฟล์ข้อมูลที่จะทำการสำรองข้อมูล

c. มีการเปรียบเทียบเทียบไฟล์ข้อมูลที่ทำการสำรองกับไฟล์ข้อมูลที่มีอยู่ในขณะนั้นเพื่อเป็นทางเลือกในการสำรองข้อมูล

d. รองรับไฟล์สื่อประสม (Multimedia) ในช่วงที่กว้างมากยิ่งขึ้น

5. Norton Ghost

a. ใช้การสร้าง Backup Images โดยไม่ต้องอาศัยการ Restart Windows ใหม่

b. ใช้รูปแบบการสำรองข้อมูลแบบ Incremental Backup ทำให้ประหยัดพื้นที่และเวลาในการสำรองข้อมูล

c. สามารถตั้งเวลาในการทำการสำรองข้อมูล

d. สามารถทำการสำรองข้อมูลทุกประเภทในฮาร์ดไดรฟ์ และพาร์ทิชัน (Partition)

e. สามารถกู้คืนข้อมูลจาก Backup Image ได้ แม้ว่าเครื่องคอมพิวเตอร์ไม่สามารถเปิดวินโดวส์ (Windows) ได้ก็ตาม

<u>อันดับการใช้งานและการเปรียบเทียบคุณสมบัติของโปรแกรมที่ได้รับความนิยมสูงสุด</u>

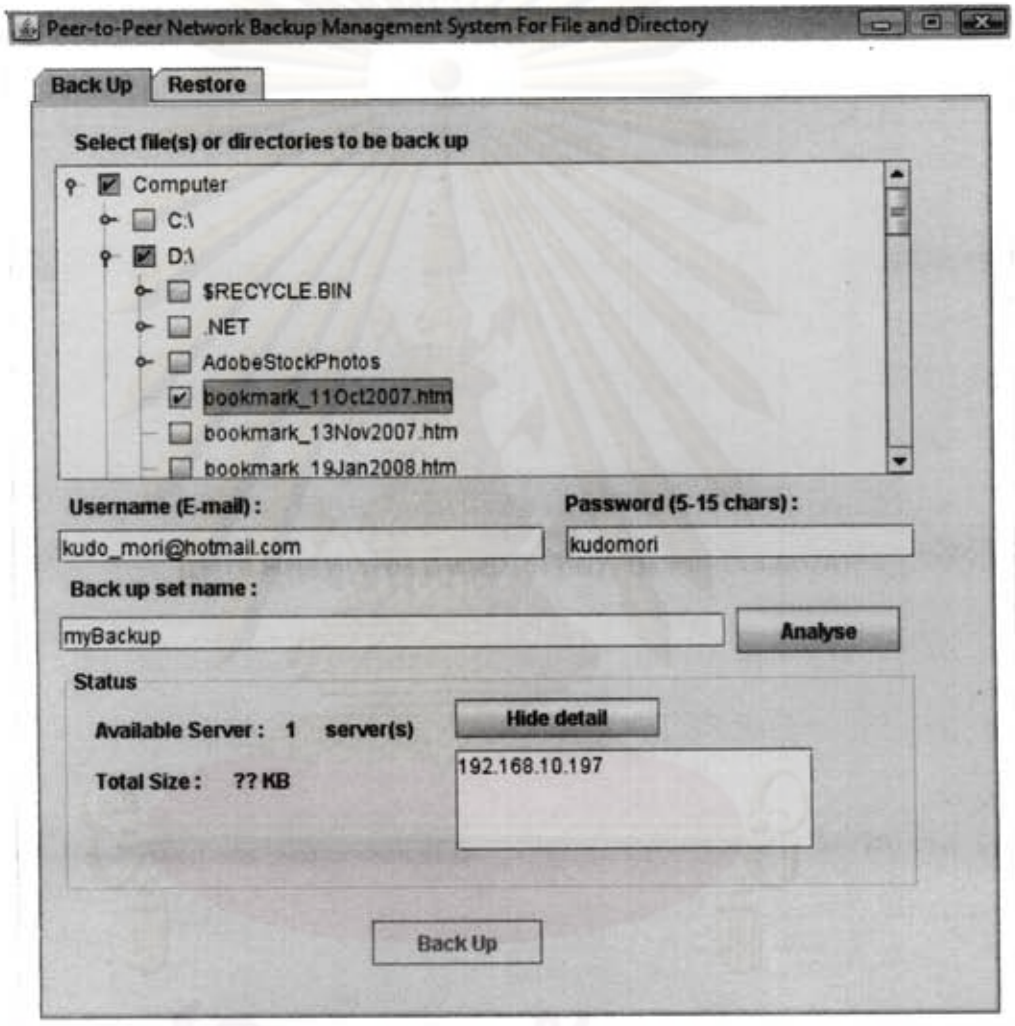
จากการสำรวจโปรแกรมสำรองข้อมูลในปัจจุบัน สามารถแสดงอันดับการใช้งานและการเปรียบเทียบคุณสมบัติของโปรแกรมที่ได้รับความนิยมสูงสุด 10 อันดับแรก ได้ดังรูปที่ 34

Excellent / Very Good / Good / Fair / Poor	Genie Backup Manager	Acronis True Image	NTI Backup NOW! Deluxe Suite	PowerBac	Norton Ghost	AIS Backu	TurboBac	Handy Backup	PC Backup	WinBacku
Rank	GOLD	SILVER	BRONZE	4	5	6	7	8	9	10
Overall Rating	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■
Ratings										
Feature Set	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■
Ease of Use	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■
Backup/Restore	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■
Help/Documentation	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■	■■■■■
Type of Backup										
Full Backup	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Incremental Backup	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Differential Backup	✓	✓	✓	✓			✓		✓	
Backup Capabilities										
Individual File Backup	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Files--In-Use Backup	✓	✓	✓	✓		✓	✓	✓	✓	✓
Individual Folder Backup	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Complete System Backup (Image Backup)		✓			✓	✓			✓	
One-Click Drive Backup		✓			✓				✓	
Address Book Backup	✓	✓	✓	✓	✓	✓	✓			
Registry Backup	✓	✓	✓		✓	✓	✓		✓	
Favorites Backup	✓	✓	✓	✓	✓	✓	✓			✓
Backup Scheduling	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Individual eMail Backup	✓	✓	✓			✓	✓			✓
Outlook Express	✓	✓	✓	✓		✓	✓			✓
Outlook	✓	✓	✓	✓		✓	✓			✓
MSN	✓									
Eudora	✓						✓			
Output Features										
Backup to Hard Drive	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Backup to Removable Media (CD,DVD,Zip,etc.)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Backup to FTP Locations	✓	✓	✓			✓	✓	✓		
Advanced Features										
Backup Password Protection	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Backup Password Encryption	✓	✓	✓			✓		✓		
Data Compression	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
File Filtering	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Backup Verification	✓	✓	✓	✓	✓	✓	✓		✓	✓
Drive Spanning	✓		✓	✓		✓	✓			
Event Logs/Reports/ Catalog Backups	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Burning Software Included	✓		✓	✓			✓	✓	✓	✓
CD/DVD Erase Capability	✓	✓	✓	✓		✓	✓			
Create Bootable Backup	✓	✓	✓		✓	✓	✓		✓	
Restore Features										
Restore File Paths	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Restore Individual Files/Folders	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Restores Hard Drive Images	✓	✓			✓	✓			✓	
Restores to a Separate Location	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Replaces Older Files and Duplicates	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Technical Support										
eMail Support	✓	✓	✓	✓		✓	✓	✓		✓
Automatic Updates	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Help Documentation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Supported Configurations										
Vista	✓	✓			✓	✓	✓	✓	✓	
Windows XP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Windows Server 2003						✓		✓		
Windows 2000	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Windows 98			✓			✓	✓	✓	✓	✓
Windows 95			✓			✓	✓	✓	✓	
Windows NT	✓		✓				✓	✓	✓	
Windows ME			✓	✓		✓	✓	✓	✓	✓

รูปที่ 34 ภาพแสดงอันดับและเปรียบเทียบคุณสมบัติการใช้งานด้านต่างๆของโปรแกรมสำรองข้อมูลที่ได้รับความนิยมมากที่สุด 10 อันดับแรก

ภาคผนวก ข.
ตัวอย่างหน้าจอของส่วนต่อประสานผู้ใช้ของระบบสำรองข้อมูล แบบเพียร์ทูเพียร์ที่พัฒนาโดยภาษาจาวา (JAVA)

ส่วนการสำรองข้อมูล



รูปที่ 35 ตัวอย่างหน้าจอแสดงส่วนต่อประสานกับผู้ใช้ (User Interface) ในส่วนของการสำรองข้อมูล (Backup)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวจารินี ชยาภิรมย์ เกิดเมื่อวันที่ 25 มีนาคม พ.ศ. 2528 ที่จังหวัดพัทลุง สำเร็จการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2550 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2551

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย