การพัฒนาระบบรวมแฟ้มข้อมูลส่วนบุคคลบนระบบปฏิบัติการวินโดว์ด้วยโดคาน

นายวราวุฒิ บัวเทศ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2554

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย
บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository(CUIR)
are the thesis authors' files submitted through the Graduate School.

DEVELOPMENT OF PERSONAL FILE SYSTEM AGGREGATION ON WINDOWS USING DOKAN

Mr. Warawut Buathet

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2011

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การพัฒนาระบบรวมแฟ้มข้อมูลส่วนบุคคลบนระบบปฏิบัติการวินโดว์ด้วยโดคาน
โดย	นายวราวุฒิ บัวเทศ
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์ ดร.ยรรยง เต็งอำนวย)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา)

..... กรรมการ
(อาจารย์ ดร.นัทที นิภานันท์)

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

วราวุฒิ บัวเทศ : การพัฒนาระบบรวมแฟ้มข้อมูลส่วนบุคคลบนระบบปฏิบัติการวินโดว์ด้วยโดคาน. (DEVELOPMENT OF PERSONAL FILE SYSTEM AGGREGATION ON WINDOWS USING DOKAN) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.เกริก ภิรมย์โสภา, 46 หน้า.

งานวิจัยนี้นำเสนอระบบแฟ้มข้อมูลแบบรวมกลุ่มบนระบบปฏิบัติการวินโดว์ เนื่องจากระบบแฟ้มแบบกระจายที่มีอยู่ในปัจจุบันส่วนใหญ่มักทำงานบนลินุกซ์ และไม่สามารถนำมาประยุกต์ใช้บนวินโดว์ได้โดยตรง ดังนั้นผู้วิจัยจึงนำหลักการของระบบแฟ้มแบบกระจาย ระบบแฟ้มแบบร่วม และโดคานมาประยุกต์ใช้ร่วมกัน เพื่อสร้างระบบแฟ้มข้อมูลที่มีความสามารถในการรวบรวมแฟ้มจากหลายแหล่งข้อมูล (พร้อมการแก้ปัญหาชื่อซ้ำซ้อน) มาแสดงเสมือนเป็นแฟ้มข้อมูลในเครื่องผู้ใช้เอง นอกจากนี้ระบบยังสนับสนุนการทำงานแบบนอกสายโดยการสนับสนุนของแคชแบบแอลอาร์ยู ระบบต้นแบบแสดงให้เห็นว่า ระบบแฟ้มที่พัฒนาขึ้นสามารถรวมแฟ้มข้อมูลจากหลายแหล่งมาแสดงเป็นไดว์ในวินโดว์ได้ โดยมีค่าใช้จ่ายและประสิทธิภาพในระดับที่ยอมรับได้

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อ อ. ที่ปรึกษาวิทยานิพนธ์หลัก .....
ปีการศึกษา 2554 .....

5270485821 : MAJOR COMPUTER SCIENCE
KEYWORDS : DISTRIBUTED FILE SYSTEMS / UNIFICATION FILE SYSTEMS / DOKAN / LEAST RECENTLY USED

WARAWUT BUATHET : DEVELOPMENT OF PERSONAL FILE SYSTEM AGGREGATION ON WINDOWS USING DOKAN. ASST. PROF. KRERK PIROMSOPA, Ph.D., 46 pp.

In this work, we proposed file system aggregation to manage files on Windows. Existing distributed file systems usually run on Linux and are difficult to implement on Windows. Therefore, we apply the concept of distributed file systems, union file systems and Dokan to combine files from several data nodes (with name-conflict resolution) and make them available as a local drive. In addition, our system also supports offline mode with help from LRU based Cache. Our prototype shows that the system can combine files from several data nodes as a single drive on Windows with acceptable overhead.

Department: Computer Engineering Student's Signature .....
Field of Study: Computer Science Advisor's Signature.....
Academic Year: 2011

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้มิอาจเสร็จสมบูรณ์ได้หากปราศจากความช่วยเหลือจากอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา ที่ช่วยให้คำแนะนำและข้อคิดเห็นที่เป็นประโยชน์ต่องานวิจัยชิ้นนี้เสมอมา ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

ขอขอบคุณคณะกรรมการสอบวิทยานิพนธ์ อาจารย์ ดร.ยรรยง เต็งอำนวย อาจารย์ ดร.นัทที นิภานันท์ และ อาจารย์ ดร.พงศ์ธวัช ชีพพิมลชัย ที่สละเวลามาตรวจ ให้ข้อเสนอแนะและข้อคิดเห็นที่เป็นประโยชน์ต่อการพัฒนาวิทยานิพนธ์ฉบับนี้

ขอขอบคุณจุฬาลงกรณ์มหาวิทยาลัยที่ได้ประสิทธิ์ประสาทความรู้ในด้านวิชาการ เป็นสถานที่เก็บเกี่ยวประสบการณ์การใช้ชีวิตในมหาวิทยาลัย ทำให้ได้พบกับอาจารย์ เพื่อน พี่ น้อง ที่คอยช่วยเหลือและให้คำปรึกษามาโดยตลอด

สุดท้ายนี้ ขอขอบคุณทุกคนในครอบครัวที่สนับสนุน และเป็นกำลังใจตลอดมา จนสามารถทำวิทยานิพนธ์นี้ได้เสร็จสมบูรณ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฌ

สารบัญภาพ ..... ฎ

บทที่

1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของการวิจัย 3

1.3 ขอบเขตของการวิจัย 3

1.4 ประโยชน์ที่คาดว่าจะได้รับ 3

1.5 ขั้นตอนและวิธีดำเนินการวิจัย 4

1.6 ผลงานตีพิมพ์ 4

2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5

2.1 ทฤษฎีที่เกี่ยวข้อง 5

2.1.1 ระบบแฟ้มแบบกระจาย 5

2.1.2 ระบบแฟ้มยูนิฟิเคชั่น 8

2.1.3 อัลกอริทึมการแทนที่แบบแอลอาร์ยู 9

2.1.4 โดคาน 10

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง 11

2.2.1 Network Distributed File System in Userspace 11

2.2.2 Disconnected Operation in the Coda File System 12

2.2.3 Personal Cloud File System (SPAFS2) 13

บทที่ หน้า

3 การออกแบบระบบรวมแฟ้มข้อมูลส่วนบุคคล 15
3.1 จุดมุ่งหมาย 15
3.2 ภาพรวมของระบบ 15
3.3 ส่วนประกอบของระบบ 16
3.4 ลักษณะการทำงานของระบบ 21
4 การพัฒนาระบบรวมแฟ้มข้อมูลส่วนบุคคล 28
4.1 ระบบรวมแฟ้มข้อมูลส่วนบุคคลบนระบบปฏิบัติการวินโดว์ด้วยโดคาน 28
4.2 การแก้ไขความขัดแย้งของข้อมูล 29
4.3 การเก็บและดูข้อมูลการใช้งาน 31
4.4 ระบบแคช 33
4.5 รายการการทำงานของแฟ้มข้อมูล WinSPAFS 34
4.6 การตรวจสอบสถานะและอัพเดทสถานะของแฟ้มข้อมูล 37
5 การทดสอบประสิทธิภาพ 38
5.1 อุปกรณ์ที่ใช้ในการทดสอบ 38
5.2 การทดสอบ 39
5.3 ผลการทดสอบ 40
5.4 วิเคราะห์ผลการทดสอบ 41
6 บทสรุป 43
6.1 สรุปผล 43
6.2 ข้อจำกัด 43
6.3 แนวทางการวิจัยในอนาคต 43
รายการอ้างอิง 44
ประวัติผู้เขียนวิทยานิพนธ์ 46

สารบัญตาราง

ตารางที่ หน้า

4.1 รายการการทำงานของแฟ้มข้อมูล WinSPAFS 37
5.1 อุปกรณ์ที่ใช้ในการทดสอบ 38
5.2 ผลการทดสอบประสิทธิภาพ.....40

สารบัญภาพ

ภาพที่ หน้า

2.1 ระบบแฟ้มแบบกระจาย....5
2.2 ลักษณะการทำงานของระบบแฟ้มยูนิฟิเคชั่น....8
2.3 การสับเปลี่ยนข้อมูลแบบแอลอาร์ยู....10
2.4 สถาปัตยกรรมของโดคาน....11
2.5 สถาปัตยกรรมของ FUSE....11
2.6 ลักษณะการทำงานของ Coda....12
2.7 ส่วนประกอบของ SPAFS2....13
3.1 ลักษณะของระบบรวมแฟ้มข้อมูล WinSPAFS....15
3.2 ส่วนประกอบของระบบรวมแฟ้มข้อมูล WinSPAFS....16
3.3 ส่วนต่อประสาน....17
3.4 Cache Management....17
3.5 Replacement Algorithm....18
3.6 Sync Management....18
3.7 Database....19
3.8 จัดการข้อมูลเกี่ยวกับการใช้งานแฟ้มข้อมูล....19
3.9 Storage Device....20
3.10 Path File sharing....20
3.11 Data Node....20
3.12 ลักษณะการเชื่อมต่อกับเครื่องปลายทาง....21
3.13 การรวมแฟ้มข้อมูล....21
3.14 ผลที่ได้จากแก้ปัญหาด้วยแนวคิดพื้นฐานของระบบแฟ้มยูนิฟิเคชั่น....22
3.15 ผลที่ได้จากการใช้วิธีระบุพาทของแฟ้มข้อมูลต่อท้ายชื่อแฟ้มข้อมูล....23
3.16 เงื่อนไขในการเก็บและจัดการแคช....24
3.17 ตัวเลือกการจัดเก็บแคช....25
3.18 พื้นที่จัดเก็บข้อมูลบนฮาร์ดดิสก์....25
3.19 ผลที่ได้จากการเก็บแคช....25
3.20 การใช้งานแฟ้มข้อมูลจาก Data Cache โดยตรง....26

ภาพที่ หน้า

3.21 การรักษาสภาวะของแฟ้มข้อมูล....27
4.1 การพัฒนาส่วนต่อประสานงานกับผู้ใช้....28
4.2 การจัดลำดับความสำคัญของแฟ้มข้อมูลในระบบ....29
4.3 การจัดลำดับความสำคัญของแฟ้มข้อมูลที่มาจากเครื่องปลายทาง....30
4.4 การกำหนดสัญลักษณ์ที่ใช้คั่นระหว่างชื่อแฟ้มข้อมูลกับพาทของแฟ้มข้อมูล....31
4.5 ส่วนของการจัดเก็บการใช้งานแฟ้มข้อมูลในเมาส์พอยท์....32
4.6 ส่วนของการแสดงข้อมูลการใช้งานแฟ้มข้อมูลในเมาส์พอยท์....32
4.7 ส่วนของการจัดเก็บแคชหรือแฟ้มข้อมูลในเมาส์พอยท์....33
4.8 ส่วนของการคัดเลือกแคชหรือแฟ้มข้อมูลในเมาส์พอยท์....34
4.9 ส่วนของการตรวจสอบและอัพเดทแฟ้มข้อมูล....37
5.1 รูปแบบการเชื่อมต่อที่ใช้ในการทดสอบ....39
5.2 ตัวอย่างผลทดสอบที่ได้จากการทดสอบประสิทธิภาพด้วยโปรแกรมไอโอโซน....40
5.3 เปรียบเทียบประสิทธิภาพการทำงานของระบบแฟ้มข้อมูล....41

บทที่ 1

บทนำ

ในบทนำนี้จะแบ่งเป็นเจ็ดหัวข้อย่อย กล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ ขอบเขตของการวิจัย ประโยชน์ที่คาดว่าจะได้รับ ขั้นตอนและวิธีดำเนินการวิจัย ลำดับขั้นตอนในการเสนอผลการวิจัย และผลงานที่ตีพิมพ์จากวิทยานิพนธ์ ตามลำดับดังนี้

1.1 ความเป็นมาและความสำคัญของปัญหา

คอมพิวเตอร์มีบทบาทที่สำคัญยิ่งต่อสังคมของมนุษย์เราในปัจจุบัน มีการนำมาใช้ในงานเกือบทุกด้าน กล่าวได้ว่าคอมพิวเตอร์เป็นปัจจัยที่สำคัญอย่างยิ่งต่อการดำเนินชีวิตประจำวัน การใช้งานคอมพิวเตอร์แบบตั้งโต๊ะ (Desktop) ซึ่งการปรับแต่งที่ทำได้ง่าย ราคาไม่สูง แต่พกพาได้ลำบาก การใช้คอมพิวเตอร์แล็ปท๊อป (Laptop) หรือคอมพิวเตอร์เน็ตบุ๊ค (Netbook) ทำให้พกพาได้สะดวก แต่การปรับแต่งทำได้ยาก มีราคาสูง และการใช้คอมพิวเตอร์ประเภทให้เซิร์ฟเวอร์ (Server) ในกรณีต้องการเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูงหรือทำงานเฉพาะด้าน ซึ่งมีราคาสูง ยากต่อการพกพา ซึ่งเป็นไปได้ว่าจะมีผู้ใช้งานและจัดเก็บแฟ้มข้อมูลลงบนคอมพิวเตอร์มากกว่าหนึ่งเครื่องขึ้นไป

การใช้งานคอมพิวเตอร์หลายเครื่องทำให้ผู้ใช้งานเกิดปัญหาและความยุ่งยากในการใช้งานหรือจัดการแฟ้มข้อมูล มีความเป็นไปได้ที่จะมีแฟ้มข้อมูลกระจายอยู่ต่างที่หรือต่างเครื่องกัน หลีกเลี่ยงไม่ได้ที่ผู้ใช้งานจะต้องทำการถ่ายโอนแฟ้มข้อมูลหรือแฟ้มข้อมูลไปมาระหว่างเครื่อง จึงต้องมีวิธีการถ่ายโอนแฟ้มข้อมูลจากเครื่องหนึ่งไปยังอีกเครื่องหนึ่ง แม้ว่าการจัดเก็บแฟ้มข้อมูลเดียวกันไว้หลายๆ ที่ หรือหลายๆ เครื่องจะสามารถช่วยแก้ปัญหานี้ได้ แต่อาจไม่ใช่วิธีที่เหมาะสม เพราะทำให้สิ้นเปลืองพื้นที่ในการจัดเก็บแฟ้มข้อมูล ในกรณีที่มีการเปลี่ยนแปลงเนื้อหาภายในแฟ้มข้อมูลในเครื่องใดเครื่องหนึ่ง ต้องมีการอัพเดทแฟ้มข้อมูลในทุกๆ เครื่องให้ตรงกันเสมอ และอาจทำให้มีความซ้ำซ้อนกันของแฟ้มข้อมูลได้อีกด้วย (ยกเว้นแต่ว่าเป็นแฟ้มข้อมูลที่ทำการสำรองข้อมูลไว้แล้วสำหรับกรณีฉุกเฉิน) ซึ่งการถ่ายโอนแฟ้มข้อมูลทั่วไปในปัจจุบันสามารถทำได้หลายวิธีดังนี้

การถ่ายโอนแฟ้มข้อมูลโดยใช้อุปกรณ์จัดเก็บข้อมูล เช่น แฟลตไดร์ (Flash drive) แผ่นซีดี (CD) แผ่นดีวีดี (DVD) แผ่นบลูเรย์ (Blu-ray Disc) และฮาร์ดดิสก์แบบภายนอก (External Hard disk) เป็นต้น ข้อดีคือ สามารถหาซื้อได้ทั่วไป เลือกขนาดพื้นที่จัดเก็บแฟ้มข้อมูลและชนิดของพอร์ตในการเชื่อมต่อได้ตามที่ผู้ใช้ต้องการ การใช้งานง่าย ไม่มีความซับซ้อน ไม่จำเป็นต้องอาศัย

การเชื่อมต่อผ่านระบบเครือข่าย แต่ข้อเสียหรือปัญหาที่ตามมาคือ ถ้าหากข้อมูลที่ต้องการอยู่อีกสถานที่หรือมีความห่างไกลกันมาก อาจทำให้เกิดความไม่สะดวกในการใช้งาน เพราะผู้ใช้ต้องเสียเวลาเดินทางในการไปนำแฟ้มข้อมูลจากเครื่องที่ต้องการมายังเครื่องที่ผู้ใช้ใช้งานอยู่และการนำแฟ้มข้อมูลจากเครื่องหนึ่งไปคัดลอกลงอีกเครื่องหนึ่งต้องทำการคัดลอกแฟ้มข้อมูลมาใส่อุปกรณ์จัดเก็บข้อมูลก่อน แล้วจึงนำอุปกรณ์จัดเก็บข้อมูลไปต่อเข้ากับอีกเครื่องหนึ่ง เพื่อคัดลอกแฟ้มข้อมูลลงเครื่องนั้น จึงทำให้เสียเวลามาก และบางครั้งยังมีข้อจำกัดเรื่องพื้นที่ในการจัดเก็บแฟ้มข้อมูลของอุปกรณ์จัดเก็บแฟ้มข้อมูลอีกด้วย เช่น ถ้าต้องการพื้นที่ในการจัดเก็บมาก ราคาก็จะสูงขึ้นตามไปด้วย

การถ่ายโอนแฟ้มข้อมูลโดยผ่านทางระบบเครือข่าย (Network) หรือการใช้โปรแกรมถ่ายโอนแฟ้มข้อมูลต่างๆ เช่น เอฟทีพี ดรอปบ็อกซ์และการแชร์แฟ้มข้อมูลผ่านทางเครือข่ายเริ่มเข้ามามีบทบาทมาก เนื่องจากความเจริญก้าวหน้าของระบบเครือข่ายและอินเทอร์เน็ต ซึ่งการถ่ายโอนแฟ้มข้อมูลผ่านทางเครือข่ายเป็นวิธีที่สะดวกกว่าการใช้อุปกรณ์จัดเก็บแฟ้มข้อมูลแบบพกพา แต่ก็ยังมีข้อจำกัดบางประการ เช่น การใช้เอฟทีพี (FTP : File Transfer Protocol) [1] ต้องทำการคัดลอกแฟ้มข้อมูลนั้นมายังเครื่องของผู้ใช้งานก่อน จึงจะทำการใช้งานแฟ้มข้อมูลนั้นได้ การใช้ดรอปบ็อกซ์ (Drop Box) [2] จะเสียค่าบริการถ้าต้องการพื้นที่ในการเก็บแฟ้มข้อมูลเพิ่มขึ้น ส่วนการใช้วิธีแชร์แฟ้มข้อมูลผ่านทางเครือข่าย (File Sharing) [3] ผู้ใช้จะรู้สึกว่าแฟ้มข้อมูลเหล่านั้นไม่ได้อยู่ในเครื่องของผู้ใช้เอง และวิธีต่างๆ ข้างต้น ไม่สามารถใช้งานได้เลย ในกรณีที่ผู้ใช้ไม่สามารถเข้าถึงเครือข่ายได้ (Offline)

ระบบแฟ้มแบบกระจายนั้นทำให้ผู้ใช้สามารถทำการเข้าถึงแฟ้มข้อมูลที่อยู่ในเครื่องปลายทางได้เสมือนกับว่าแฟ้มข้อมูลเหล่านั้นอยู่ในเครื่องของผู้ใช้เอง ปัจจุบันมีการพัฒนาระบบแฟ้มแบบกระจายขึ้นมาหลายระบบ เช่น AFS [4], Coda [5], NFS [6] และ Personal Cloud File System [7] แต่ละระบบก็มีข้อเสียบางประการ เช่น บางระบบทำงานได้บนระบบปฏิบัติการลินุกซ์เท่านั้น ไม่สามารถนำมาประยุกต์ใช้บนระบบปฏิบัติการวินโดว์ได้โดยตรง บางระบบมีความยุ่งยากในการติดตั้ง ใช้งาน บางระบบต้องมี Dedicated Server ในการทำงานและบางระบบนำมาพัฒนาต่อได้ยาก

ระบบปฏิบัติการวินโดว์เป็นระบบปฏิบัติการที่มีผู้นิยมใช้งานมากที่สุด เนื่องจากมีส่วนต่อประสานงานที่น่าใช้งาน มีสิ่งเอื้ออำนวยให้ใช้งาน เรียนรู้ได้ง่ายและมีการพัฒนาอย่างต่อเนื่อง ทำให้ระบบปฏิบัติการวินโดว์นั้นได้รับความนิยมอย่างกว้างขวาง และมีการนำมาใช้งานกับเครื่องคอมพิวเตอร์ประเภทต่างๆ อย่างมากมาย

งานวิจัยนี้ได้ทำการออกแบบและพัฒนาระบบรวมแฟ้มข้อมูลที่มีลักษณะการทำงานแบบส่วนบุคคลบนระบบปฏิบัติการวินโดว์ขึ้นมา เพื่อช่วยในการจัดการการเข้าถึงและถ่ายโอนแฟ้มข้อมูลมีประสิทธิภาพยิ่งขึ้น ซึ่งผู้ใช้งานจะมองเห็นแฟ้มข้อมูลจากเครื่องปลายทางเสมือนว่าแฟ้มข้อมูลต่างๆ เหล่านี้อยู่บนเครื่องของผู้ใช้เองและยังสามารถใช้งานแฟ้มข้อมูลต่างๆ เหล่านี้ได้เสมือนว่าแฟ้มข้อมูลต่างๆ เหล่านี้อยู่บนเครื่องผู้ใช้ โดยใช้วิธีการรวมแฟ้มข้อมูลต่างๆ จากเครื่องปลายทางซึ่งเชื่อมต่อผ่านทางระบบเครือข่าย ไม่ต้องมี Dedicated Server ในการทำงาน มีระบบแคชที่ช่วยให้การทำงานเร็วขึ้นและสนับสนุนการใช้งานแบบนอกสายในกรณีที่ผู้ใช้ไม่สามารถเข้าถึงเครือข่ายได้ โดยเน้นให้ใช้งานง่าย สะดวก และสามารถนำไปพัฒนาต่อได้ง่ายในอนาคตเมื่อเทียบกับระบบแฟ้มกระจายที่มีอยู่ในปัจจุบัน

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 ศึกษาข้อดีข้อเสียของระบบแฟ้มแบบกระจายที่มีอยู่ในปัจจุบัน

1.2.2 ออกแบบระบบรวมแฟ้มข้อมูลส่วนบุคคลที่ทำงานอยู่บนระบบปฏิบัติการวินโดว์

1.2.3 พัฒนาระบบโดยการนำหลักการของระบบแฟ้มแบบกระจาย ระบบแฟ้มแบบร่วม อัลกอริทึมการแทนที่ และโคดานมาประยุกต์ใช้ร่วมกัน

1.2.4 สร้างระบบแฟ้มข้อมูลที่มีความสามารถในการรวบรวมแฟ้มข้อมูลจากหลายแหล่งข้อมูล (พร้อมการแก้ปัญหาชื่อซ้ำซ้อน)

1.2.5 ศึกษาปัจจัยที่มีต่อประสิทธิภาพของระบบรวมแฟ้มข้อมูลส่วนบุคคล

1.2.6 ทดสอบและวัดประสิทธิภาพของระบบรวมแฟ้มข้อมูลส่วนบุคคล

1.3 ขอบเขตของการวิจัย

1.3.1 ทำการพัฒนาและทดสอบระบบบนระบบปฏิบัติการวินโดว์

1.3.2 ทำการพัฒนาโดยใช้ Dokan และภาษา Visual C# .NET

1.3.3 ทำการสร้างและทดสอบกับอัลกอริทึมการแทนที่แอลอาร์ยู (LRU)

1.3.4 ทำการสร้างและทดสอบกับเครือข่ายด้วย TCP

1.3.5 ทำการเชื่อมต่อระบบด้วยพื้นฐานการเชื่อมต่อผ่านทางระบบเครือข่าย

1.3.6 ทำการทดสอบการทำงานแบบอินทราเน็ตและอินเทอร์เน็ต

1.3.7 ทำการทดสอบการทำงานโดยวัดประสิทธิภาพจาก throughput

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ได้ระบบต้นแบบสำหรับการรวมแฟ้มข้อมูลแบบกลุ่มบนระบบปฏิบัติการวินโดว์

1.4.2 สามารถนำไปประยุกต์ใช้กับปัญหาต่างๆ ได้

1.4.3 เป็นพื้นฐานของงานวิจัยอื่นๆ ในอนาคต

1.5 ขั้นตอนและวิธีดำเนินการวิจัย

1.5.1 ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง

1.5.2 กำหนดหัวข้อปัญหาของงานวิจัย

1.5.3 ศึกษาเครื่องมือที่ใช้ในการพัฒนางานวิจัย

1.5.4 ออกแบบและพัฒนาระบบ

1.5.5 ทดสอบประสิทธิภาพของระบบ

1.5.6 วิเคราะห์และสรุปผลการทดสอบ

1.5.7 เรียบเรียงวิทยานิพนธ์

1.6 ผลงานตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้ตีพิมพ์และนำเสนอในการประชุมวิชาการดังนี้

1.6.1 บทความชื่อ *"Development of Personal File System Aggregation on Windows Using Dokan"*

1.6.1.1 ชื่อผู้แต่ง วราวุฒิ บัวเทศ และ เกริก ภิรมย์โสภา

1.6.1.2 ตีพิมพ์และนำเสนอในงานประชุมวิชาการ "2012 ninth International Joint Conference on Computer Science and Software Engineering (JCSSE 2012)" ณ มหาวิทยาลัยหอการค้าไทย จังหวัดกรุงเทพมหานคร ระหว่างวันที่ 30 พฤษภาคม – 1 มิถุนายน 2555

บทที่ 2

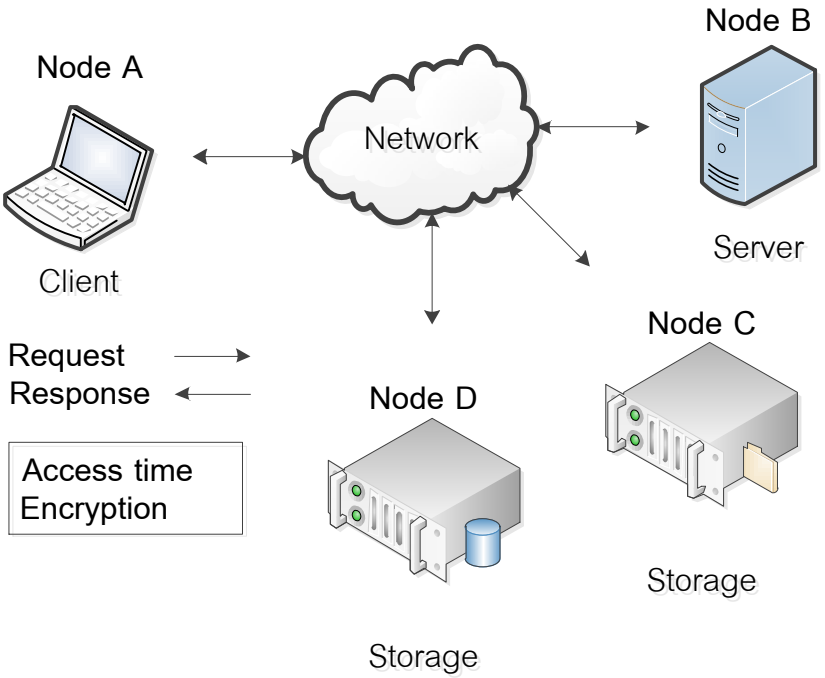
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

แนวคิดและทฤษฎีที่จะอธิบายในการวิจัยนี้ แบ่งเป็น 4 ส่วน ได้แก่ ระบบแฟ้มแบบกระจาย ระบบแฟ้มยูนิฟิเคชั่น อัลกอริทึมการแทนที่แอลอาร์ยู และโดคาน

2.1.1 ระบบแฟ้มแบบกระจาย

ระบบแฟ้มแบบกระจาย (Distributed File System) [8] ดังภาพที่ 2-1 มีคุณสมบัติใกล้เคียงกับระบบรวมแฟ้มข้อมูลส่วนบุคคลที่งานวิจัยนี้นำเสนอ คือ ระบบแฟ้มแบบกระจายนั้นทำให้ผู้ใช้สามารถทำการเข้าถึงแฟ้มข้อมูลและโฟลเดอร์ที่อยู่ในเครื่องปลายทางได้เสมือนกับว่าแฟ้มข้อมูลและโฟลเดอร์เหล่านั้นอยู่ในเครื่องของผู้ใช้ เพื่อการใช้ทรัพยากรร่วมกัน (sharing resources) อย่างสะดวกและมีประสิทธิภาพและสร้างเสถียรภาพให้กับระบบ



ภาพที่ 2-1 ระบบแฟ้มแบบกระจาย

ระบบแฟ้มแบบกระจายช่วยผู้ใช้หลายคนสามารถใช้ทรัพยากรร่วมกัน ซึ่งหมายถึงพื้นที่จัดเก็บแฟ้มข้อมูลและแฟ้มข้อมูล ช่วยให้การจัดการอุปกรณ์จัดเก็บข้อมูลบนเครือข่ายหรือสตอเรจ

(Storage) มีประสิทธิภาพมากขึ้น โดยมีลักษณะเป็นการเชื่อมโยงระบบแฟ้มข้อมูลที่มาจากแหล่งจัดเก็บข้อมูลหลายแหล่งเข้าด้วยกัน มีกลไกเป็นแบบเมาส์ (mount mechanism) เพื่ออยู่ในลักษณะของไดเรกทอรี โดยจะเรียกว่าเมาส์พอยท์ ซึ่งปกติแล้วแฟ้มข้อมูลจะถูกเก็บในลักษณะของไบนารี (Binary) โดยมีโครงสร้างเป็นแบบลำดับชั้นเช่นเดียวกับโครงสร้างของไดเรกทอรี (directory) ที่อยู่บนฮาร์ดดิสก์ตัวเดียว (single hard disk) จะใช้การเชื่อมต่อผ่านทางระบบเครือข่ายเพื่อถ่ายโอนข้อมูลจากเครื่องปลายทาง (remote machine) มายังเครื่องผู้ใช้ (local machine) โดยที่ผู้ใช้ไม่จำเป็นต้องรู้ถึงขั้นตอนและกระบวนการต่างๆ ในการเข้าถึง (access) แฟ้มข้อมูลบนเครือข่าย ผู้ใช้ไม่จำเป็นต้องรู้ว่าแฟ้มข้อมูลใดจัดเก็บอยู่ที่จุดใดหรือแม้แต่สถานที่เก็บแฟ้มข้อมูลทางกายภาพถูกเปลี่ยน เรียกได้ว่า Location transparency โดยผู้ใช้สามารถเข้าถึงแฟ้มข้อมูลในเครื่องปลายทางได้เสมือนกับการเข้าถึงแฟ้มข้อมูลในเครื่องผู้ใช้เอง และสามารถกระทำการใดๆ ได้เสมือนกับว่าแฟ้มข้อมูลนั้นถูกเก็บไว้ ณ ที่จุดที่ผู้ใช้กำลังใช้งาน

การเพิ่มประสิทธิภาพของระบบแฟ้มแบบกระจาย สามารถทำได้โดยการลดปริมาณการจราจรในเครือข่าย (network traffic) ซึ่งทำได้โดยการเก็บแคช (Cache) เช่น ถ้ามีความต้องการเข้าถึงแฟ้มข้อมูลซ้ำอีกก็สามารถจัดการได้บนเครื่องตัวเอง เพราะแฟ้มข้อมูลนั้นอยู่ในแคชของเครื่องผู้ใช้เอง หรือถ้าแฟ้มข้อมูลที่ต้องการยังไม่ได้อยู่ในแคชแฟ้มข้อมูลนั้นจะถูกส่งสำเนาจากเซิร์ฟเวอร์มายังผู้ใช้ (มีการติดต่อกับเซิร์ฟเวอร์เพียงบางครั้ง แทนที่จะต้องติดต่อทุกๆ ครั้งที่ต้องการเข้าถึงแฟ้มข้อมูล การเข้าถึงแฟ้มข้อมูลใบแบบ remote access จึงช่วยลดภาระของเซิร์ฟเวอร์และปริมาณการจราจรในเครือข่าย) การเข้าถึงแฟ้มข้อมูลในแคชแฟ้มข้อมูลที่ส่งมาจะเหมือนกับต้นฉบับที่อยู่บนเครื่องเซิร์ฟเวอร์ ซึ่งสำเนาของแฟ้มข้อมูลจะกระจายไปยังแคชที่ต่างๆ ซึ่งเมื่อแฟ้มข้อมูลมีการเปลี่ยนแปลง ก็จะทำให้แฟ้มข้อมูลในแต่ละแคชเกิดความแตกต่างกัน เกิดปัญหา Cache-consistency คือ ปัญหาแฟ้มข้อมูลที่อยู่ในแคชไม่ตรงกับแฟ้มข้อมูลต้นฉบับบนเซิร์ฟเวอร์ เช่น เป็นแฟ้มข้อมูลเก่าล้าสมัย ก็ต้องไปนำแฟ้มข้อมูลที่ได้อัพเดทแล้วมาไว้ที่แคชใหม่ ซึ่ง Caching ดีกว่า remote access ในกรณีที่การเขียนมีไม่มาก ถึงแม้จะต้องเจอปัญหาเรื่อง Cache-consistency แต่ก็ยังดีกว่าในแง่ของประสิทธิภาพ ปริมาณการจราจรในเครือข่าย และภาระของเซิร์ฟเวอร์ Caching จึงเหมาะกับการประมวลผลบนเครื่องที่มีหน่วยความจำขนาดใหญ่ ส่วนการใช้วิธี remote access ก็จะเหมาะกับเครื่องที่ไม่มีฮาร์ดดิสก์หรือพื้นที่ในการจัดเก็บของหน่วยความจำมีขนาดน้อย นอกจากนี้แล้วระบบแฟ้มแบบกระจายยังสนับสนุนการทำ Replication, Load balancing โดยการวัดประสิทธิภาพของระบบแฟ้มแบบกระจายจะวัดจากจำนวนเวลาที่ใช้ในการตอบสนองต่อการร้องของบริการ (request/response) โอเวอร์เฮด

(Overhead) ที่เกิดจากการเข้าถึงระยะไกล โดยรวมถึงเวลาที่ส่งคำขอไปยังเซิร์ฟเวอร์ เวลาที่ได้รับการตอบสนองผ่านเครือข่ายกลับไปยังเครื่องผู้ใช้งาน

เนื้อหาต่อไปนี้ จะกล่าวถึงระบบแฟ้มแบบกระจายที่มีความโดดเด่นและน่าสนใจ

2.1.1.1 Network File System (NFS)

Network File System หรือ NFS ถูกพัฒนาขึ้นโดย Sun Microsystem ในปี ค.ศ.1984 เป็นระบบแฟ้มแบบกระจายที่มีการใช้งานอย่างแพร่หลายตัวหนึ่ง เป็นระบบที่สามารถให้เครื่องคอมพิวเตอร์สามารถเข้าใช้งานฮาร์ดดิสก์บนเครื่องอื่นๆ ได้เหมือนกับใช้ฮาร์ดดิสก์บนเครื่องของตัวเอง โดยสามารถใช้งานได้อย่างสะดวกผ่านทางระบบเครือข่าย โดย NFS มีการใช้งานในหลายระบบปฏิบัติการ แต่จะใช้บนระบบ Unix เป็นส่วนใหญ่ ข้อเสียบางอย่างของ NFS คือ ไม่สามารถทำ Single view of Network File, Client Disk Cache และ Replication ได้ ทำงานได้ไม่ดีกับเครือข่ายที่มีแบนธ์วิธต่ำและช้า

2.1.1.2 Andrew File System (AFS)

Andrew File System หรือ AFS เป็นเทคโนโลยีที่พัฒนามาจากโปรเจ็ค Andrew ของมหาวิทยาลัย Carnegie Mellon (CMU) ซึ่ง AFS เป็นระบบแฟ้มแบบกระจายสำหรับระบบเครือข่าย โดยเป็นการใช้งานแฟ้มข้อมูลร่วมกันระหว่างไคลเอนต์-เซิร์ฟเวอร์ ซึ่งจะมีการสร้างแฟ้มเก็บข้อมูลชั่วคราว (Cache) หรือการทำสำเนาบนเครื่องไคลเอนต์สำหรับแฟ้มที่มีการเข้าถึงและใช้งานบ่อย เพื่อให้สามารถใช้งานได้รวดเร็วขึ้น มีประสิทธิภาพการแจกจ่ายแฟ้มข้อมูลสูง คุณสมบัติที่ยอดเยี่ยมของ AFS คือ สามารถขยายขนาดได้ไม่จำกัด AFS มีความสามารถสูงในการรักษาความปลอดภัย และมีความรวดเร็วในการแชร์แฟ้มข้อมูลร่วมกัน ซึ่งเหมาะสำหรับองค์กรขนาดกลางและขนาดใหญ่ ผู้ดูแลระบบสามารถเข้าควบคุมผ่านแพลตฟอร์มต่างๆ ที่รองรับ เช่น Windows NT, Unix, Solaris นอกจากนี้ยังสามารถเชื่อมต่อเพื่อแชร์แฟ้มข้อมูลหรือเชื่อมต่อกับระบบ AFS อื่นๆ ได้ ซึ่ง AFS ยังนำโพรโตคอล Kerberos มาใช้ในการพิสูจน์ตัวตน นอกจากนี้ AFS ยังสนับสนุนการทำ Replication เพื่อเพิ่มความคงทนต่อความล้มเหลวของระบบ ข้อเสียคือ Disconnect Operation เป็นแบบอ่านอย่างเดียว และ License เป็นแบบทางธุรกิจ (commercial)

2.1.1.3 Coda

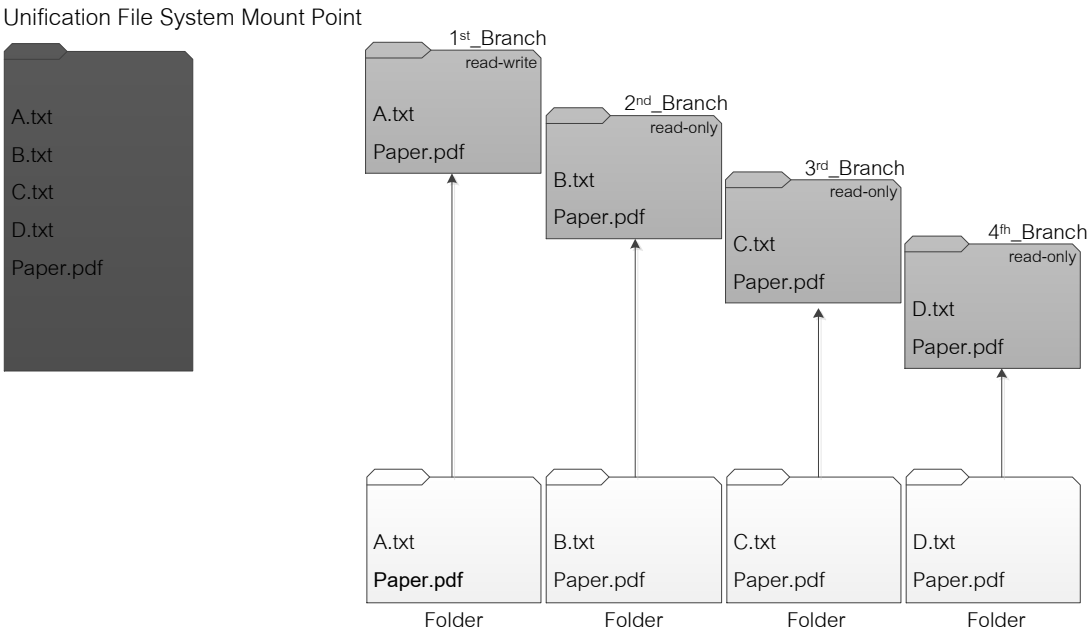
Coda พัฒนามาจาก Andrew File System (AFS) ความสามารถที่โดดเด่นของ Coda เรียกว่า disconnected operation คือ Coda สามารถทำงานได้แม้ไม่ได้เชื่อมต่อกับเครือข่าย โดยใช้การ Cache แฟ้มข้อมูลเอาไว้ในเครื่องและจะทำการ Update แฟ้มข้อมูลทันทีเมื่อเกิดการ

เชื่อมต่ออีกครั้ง Coda เองถูกพัฒนามาโดยมีเป้าหมายอยู่ที่ Mobile Networking ทั้งหมด ใช้งานได้กับระบบเครือข่ายต่างๆ ที่ช้าหรือติดๆ หลุดๆ โดยสนับสนุนระบบปฏิบัติการ Linux, NetBSD, FreeBSD, Windows 2000, XP โดยคุณสมบัติมีคุณสมบัติดังนี้

- สามารถนำ Source Code ไปใช้ได้อย่างเสรี
- สนับสนุน Disconnected Operation สำหรับใช้งานบนอุปกรณ์เคลื่อนย้ายได้
- สนับสนุนการ replication โดยอัตโนมัติ และใช้ Kerberos ในการพิสูจน์ตัวตน

2.1.2 ระบบแฟ้มยูนิฟิเคชั่น

ระบบแฟ้มยูนิฟิเคชั่น (Unification File System) [9] ทำการรวมเนมสเปซ (Namespace) ของโฟลเดอร์ตั้งแต่สองโฟลเดอร์ขึ้นไปเข้าด้วยกันเป็นหนึ่งเนมสเปซ ทำให้โฟลเดอร์ที่ได้มีเนื้อหาของสองโฟลเดอร์หรือมากกว่าสองโฟลเดอร์ขึ้นไปรวมกันอยู่ ดังภาพที่ 2-1 จะเห็นได้ว่า โฟลเดอร์ที่ได้มีเนื้อหาของสองโฟลเดอร์หรือมากกว่าสองโฟลเดอร์ขึ้นไปรวมกันอยู่ โดยมีแนวคิดพื้นฐาน คือ Branch, Write out and Opaque directories, Copy up



ภาพที่ 2-2 ลักษณะการทำงานของระบบแฟ้มยูนิฟิเคชั่น

2.1.2.1 บรานซ์

บรานซ์ (Branch) ซึ่งบรานซ์เป็นชื่อเรียกของโฟลเดอร์ต่างๆ ที่ได้นำมารวมเข้าด้วยในเมาท์พอยท์ (Mount Point) ของระบบแฟ้มยูนิฟิเคชั่น โดยแต่ละบรานซ์จะมีนโยบายการเข้าถึง เช่น Read-Only หรือ Writable ซึ่งบรานซ์จะมีลำดับก่อนหลัง โดยทั่วไปแล้วบรานซ์ที่อยู่สูงสุดจะมี

นโยบายการเข้าถึงเป็น Writable บรานซ์ที่อยู่สูงกว่าจะมีความสำคัญมากกว่า โดยสามารถนำลำดับความสำคัญไปใช้แก้ไขความขัดแย้งในกรณีที่แต่ละบรานซ์มีแฟ้มข้อมูลที่มีชื่อเหมือนกันที่อาจเกิดจากใช้งานและจัดเก็บแฟ้มข้อมูลจากคอมพิวเตอร์ตั้งแต่สองเครื่องขึ้นไปได้อีกด้วย ข้อดี คือ สามารถรวมเนื้อหาของสองโฟลเดอร์ขึ้นไปรวมกันอยู่เป็นโฟลเดอร์เดียวได้ ข้อเสียคือ เมื่อมีแฟ้มข้อมูลใดๆ ที่มีชื่อซ้ำกันตั้งแต่สองแฟ้มข้อมูลขึ้นไป ระบบแฟ้มยูนิฟิเคชั่นจะเลือกแฟ้มข้อมูลที่มีชื่อซ้ำกันเหล่านั้นให้ผู้ใช้เห็นเพียงแค่แฟ้มข้อมูลเดียว

2.1.2.2 Write out and Opaque directories

Write out and Opaque directories คือ กลไกในการแก้ปัญหาในกรณีที่ผู้ใช้ต้องการลบแฟ้มข้อมูลที่อยู่ในบรานซ์ที่มีนโยบายการเข้าถึงเป็น read-only โดยเมื่อผู้ใช้ทำการลบแฟ้มข้อมูลที่อยู่ในระบบแฟ้มยูนิฟิเคชั่นจะทำการสร้างแฟ้มข้อมูลที่เรียกว่า Write out ในบรานซ์ที่อยู่สูงที่สุดขึ้นแทนการลบแฟ้มข้อมูลดังกล่าว โดยหลังจากที่มีการสร้าง Write out ของระบบแฟ้มข้อมูลแล้ว เมื่อผู้ใช้ทำการเรียกดูแฟ้มข้อมูลและโฟลเดอร์ที่อยู่ในระบบแฟ้มยูนิฟิเคชั่นก็จะไม่เห็นแฟ้มข้อมูลที่ถูกเรียกว่า Write out ในรายชื่อของแฟ้มข้อมูลและโฟลเดอร์ถึงแม้ว่าแฟ้มข้อมูลนั้นจะยังอยู่ในบรานซ์ของตัวเองก็ตาม

2.1.2.3 Copy up

ในกรณีที่ผู้ใช้ต้องการแก้ไขแฟ้มข้อมูลที่อยู่ในบรานซ์ที่มีนโยบายการเข้าถึงเป็นแบบอ่านอย่างเดียว (read-only) นั้น เมื่อผู้ใช้แก้ไขหรือทำการเปลี่ยนแปลงคุณลักษณะต่างๆ ของแฟ้มข้อมูลที่อยู่ใน read-only branch ระบบแฟ้มยูนิฟิเคชั่นจะทำสำเนา (Copy) แฟ้มข้อมูลที่ผู้ใช้ต้องการทำการแก้ไขขึ้นไปไว้ในบรานซ์ที่มีนโยบายการเข้าถึงเป็น Writable แล้วจึงให้ผู้ใช้ทำการแก้ไขสำเนาของระบบแฟ้มข้อมูลที่อยู่ใน Writable branch

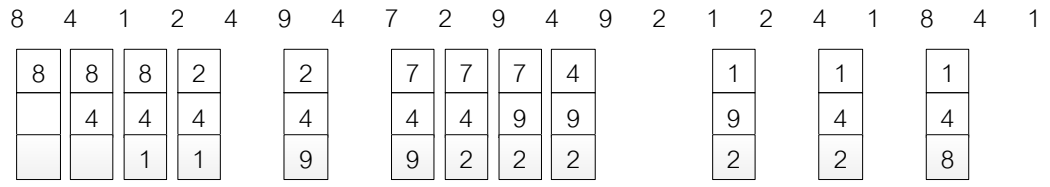
2.1.3 **อัลกอริทึมการแทนที่แอลอาร์ยู**

อัลกอริทึมการแทนที่แอลอาร์ยู (Least recently used: LRU) [10] หรือ ใช้มานานสุด-ออกก่อน คือ อัลกอริทึมการแทนที่ที่ทำการสับเปลี่ยนแฟ้มข้อมูล โดยนำแฟ้มข้อมูลใหม่มาแทนแฟ้มข้อมูลที่ไม่ได้มีการใช้งานเป็นเวลานานหรือมีอายุในแคชมากที่สุด ซึ่งแอลอาร์ยูนั้น ผลที่ได้มีความใกล้เคียงกับแบบที่ดีที่สุด (Optimal Algorithm) และสามารถใช้งานได้จริง

ความแตกต่างระหว่างแบบมาก่อน-ออกก่อน (FIFO) กับแบบที่ดีที่สุด (OPT) คือ แบบ FIFO จะดูเวลาที่สับเปลี่ยนแฟ้มข้อมูลเข้ามาในหน่วยความจำ ส่วนแบบ OPT จะดูเวลาที่แฟ้มข้อมูลถูกอ้างอิง ถ้าเราใช้ข้อมูลในอดีตที่ผ่านมาประมาณการณ์อนาคตอันใกล้ เราอาจจะ

สับเปลี่ยนแฟ้มข้อมูลที่ไม่ได้ถูกอ้างอิงมาเป็นเวลานานที่สุด คือ แบบใช้มานานสุด-ออกก่อนหรือแอลอาร์ยูนั่นเอง

ขั้นตอนวิธีแอลอาร์ยู จะบันทึกเวลาที่แต่ละข้อมูลถูกอ้างอิงครั้งล่าสุดไว้ เมื่อต้องการเลือกข้อมูลเพื่อสับเปลี่ยนออก ก็จะเลือกข้อมูลที่ไม่ได้ใช้มาเป็นเวลานานที่สุด (ข้อมูลที่มีตัวเลขเวลาน้อยที่สุด) จะเห็นได้ว่าวิธีนี้มองย้อนเวลาไปในอดีต แทนที่จะมองไปในอนาคต แสดงให้เห็นดังภาพที่ 2-3 การสับเปลี่ยนข้อมูลแบบแอลอาร์ยู

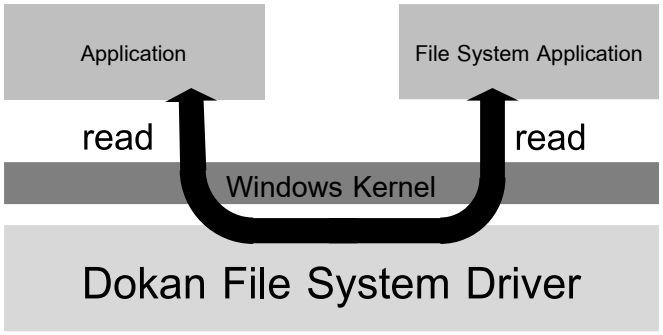


ภาพที่ 2-3 การสับเปลี่ยนข้อมูลแบบแอลอาร์ยู

2.1.4 โดคาน

โดคาน (Dokan) [11] ใช้สำหรับการสร้างหรือพัฒนาระบบแฟ้มข้อมูลบนระบบปฏิบัติการวินโดว์ เช่น การปรับปรุง NTFS, FAT จำเป็นต้องทำการเขียนหรือพัฒนาในส่วนของไดร์เวอร์จัดการของระบบแฟ้ม ซึ่งการพัฒนาไดร์เวอร์ในส่วนของเคอร์เนลบนระบบปฏิบัติการวินโดว์นั้นทำได้ยาก เนื่องมาจากมีความซับซ้อนมาก จึงได้มาทำในส่วนที่เป็นยูเซอร์สเปซของระบบปฏิบัติการวินโดว์แทน โดยการใช้ Dokan library ช่วยให้สามารถสร้างระบบแฟ้มข้อมูลได้โดยไม่ต้องเขียนหรือพัฒนาไดร์เวอร์เอง

Dokan คล้ายกับ FUSE (Linux user mode file system) [12] แต่ทำงานบนระบบปฏิบัติการวินโดว์ โดยมีลักษณะสถาปัตยกรรมดังภาพที่ 2-4 (Dokan คิดขึ้นโดย Hiroki Asakawa ซึ่งเป็นชาวญี่ปุ่น Dokan เป็นภาษาญี่ปุ่น หมายถึง ท่อดินเหนียว) โดยมีโดคานดอทเน็ตไบน์ดิง (Dokan .NET Binding) ทำให้สามารถทำการสร้างระบบแฟ้มข้อมูลโดยใช้ภาษาดอตเน็ต (.NET Language) ซึ่งเป็นภาษาที่ได้รับความนิยมและเรียนรู้ได้ง่าย เช่น ซีชาร์ฟ (C#) วิชวลเบสิก (Visual Basic) ซีพลัสพลัส (C++) ได้นอกจากนี้แล้วยังมีโดคานรูบี้ไบน์ดิง (Dokan Ruby Binding) ทำให้สามารถทำการสร้างระบบแฟ้มข้อมูลโดยใช้ภาษารูบี้ (Ruby Language) ได้อีกด้วย

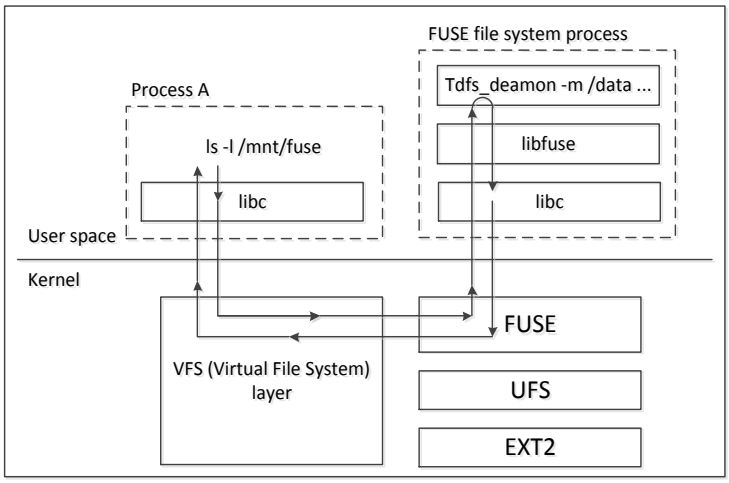


ภาพที่ 2-4 สถาปัตยกรรมของโดคาน [11]

2.1 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.2.1 Network Distributed File System in Userspace [13]

เป็นโครงงานวิจัยที่ถูกพัฒนาขึ้นในปี ค.ศ. 2006 ได้เสนอวิธีการแก้ปัญหาการกระจายของระบบแฟ้มข้อมูลที่กระจายอยู่บนเครือข่ายคอมพิวเตอร์ ซึ่งจะช่วยทำให้ผู้ใช้มองเห็นแฟ้มข้อมูลที่กระจายอยู่นั้นรวมกันอยู่ ซึ่งช่วยให้ผู้ใช้สามารถที่จะจัดเก็บและจัดการแฟ้มข้อมูลของระบบได้สะดวก โดยสิ่งที่น่าสนใจ คือ การพัฒนาขึ้นมาบนโอเพนซอร์สในยูเซอร์สเปซโดยการใช้ File System in Userspace (FUSE) ดังภาพที่ 2-5 แสดงให้เห็นถึงสถาปัตยกรรมของ FUSE



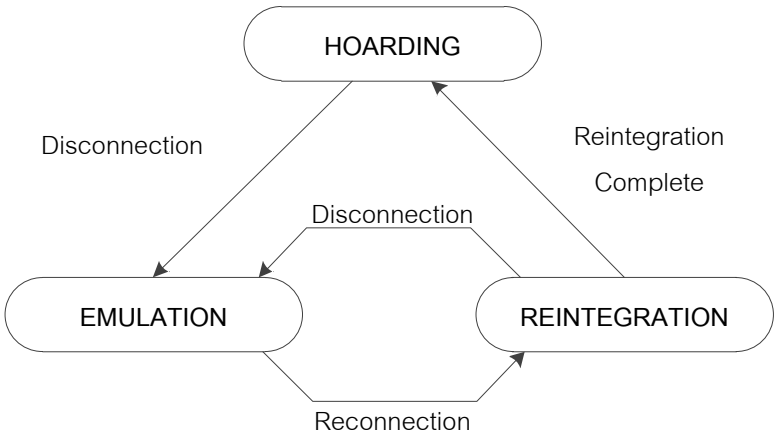
ภาพที่ 2-5 สถาปัตยกรรมของ FUSE [13]

ซึ่งการใช้ FUSE จะช่วยให้สามารถเขียนระบบแฟ้มข้อมูลในยูเซอร์สเปซได้ ซึ่งมีความซับซ้อนน้อยกว่าการเขียนระบบแฟ้มในเคอร์เนล มีการออกแบบที่ไม่ซับซ้อน ได้ประสิทธิภาพเป็นที่น่าพอใจและสามารถนำไปพัฒนาต่อในอนาคตได้ง่าย

2.2.2 Disconnected Operation in the Coda File System [6]

เป็นงานวิจัยที่ถูกพัฒนาขึ้นในปี ค.ศ. 1992 โดยได้เสนอเกี่ยวกับภาพรวมของ Coda และการดำเนินการของ Disconnected Operation ซึ่งเป็นโหมดการทำงานที่ช่วยให้ไคลเอ็นต์สามารถดำเนินการเข้าถึงแฟ้มข้อมูลที่สำคัญระหว่างที่ระบบเกิดความผิดพลาดขณะที่ทำการแชร์แฟ้มข้อมูลที่ใช้ร่วมกันบนพื้นที่ที่เก็บแฟ้มข้อมูลได้ ซึ่งการประยุกต์ใช้ Disconnected Operation ในการสนับสนุนเครื่องคอมพิวเตอร์แบบพกพา โดยส่วนมาก Coda จะใช้งานบน UNIX สิ่งที่น่าสนใจ คือ File hoarding, Callback, Weakly Connect Operation, open-to-close consistency โดยไคลเอ็นต์แต่ละตัวจะมี Cache Manage (VICE) มีการทำ High availability ด้วย Server replication โดยวิธีการหลักๆ ของ coda คือ การทำ replication และ File hoarding ซึ่งทางฝั่งไคลเอ็นต์ (Venus) จะมีการทำงานอยู่ 3 โหมด คือ Hoard, Emulation และ Reintegrating แสดงการทำงานให้เห็นดังภาพที่ 2-6

- Hoard สภาวะปกติ จะมีการสะสมหรือกักตุ่นแฟ้มข้อมูลไว้
- Emulation จะใช้งานเมื่อขาดการเชื่อมต่อ จะใช้งานในส่วนแคชของเซิร์ฟเวอร์แทน โดยเก็บแฟ้มข้อมูลต่างๆ ที่จำเป็นไว้ใน log เพื่อที่จะใช้เมื่อระบบกลับสู่สภาวะเดิม
- Reintegrating เมื่อระบบกลับสู่สภาวะเดิมไคลเอ็นต์จะทำการอัพเดทการเปลี่ยนแปลงต่างๆ จาก emulation แล้วส่งไปยังเซิร์ฟเวอร์

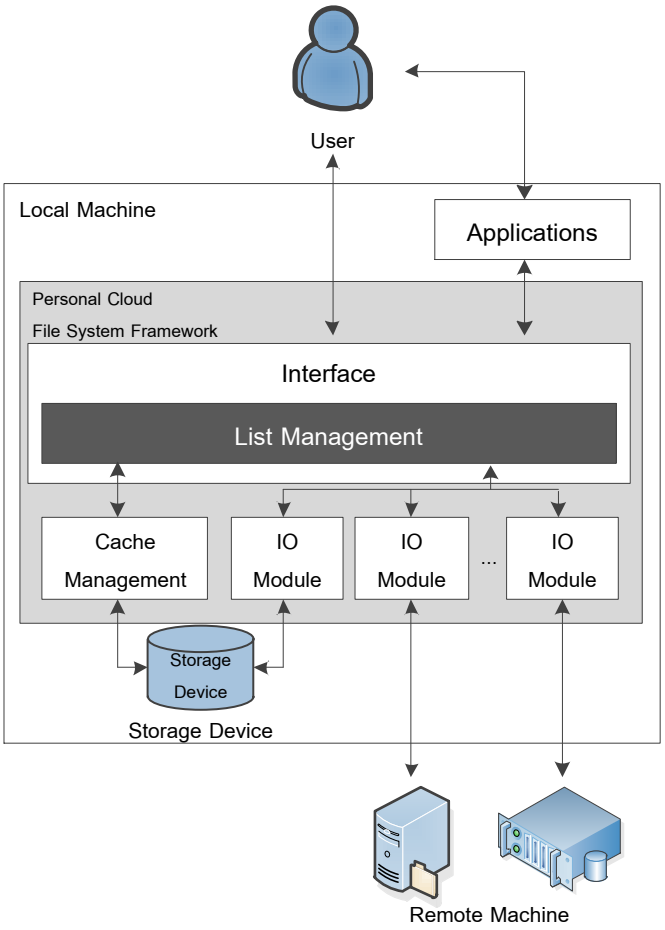


ภาพที่ 2-6 ลักษณะการทำงานของ Coda [6]

Disconnected Operation ใน Coda เป็นรูปแบบหนึ่งของระบบแบบกระจาย ช่วยลดการจราจรบนเครือข่าย เพิ่มความโปร่งใสและความทนทานต่อความผิดพลาดของระบบ ทำให้ระบบมีความน่าเชื่อถือและมีประสิทธิภาพ ซึ่งถือว่าเป็นสิ่งที่จำเป็นสำหรับระบบที่มีลักษณะเป็น high availability

2.2.3 Personal Cloud File System (SPAFS2) [7]

เป็นงานวิจัยที่ถูกพัฒนาขึ้นในปี ค.ศ. 2011 เสนอเฟรมเวิร์คระบบแฟ้มคลาวด์ส่วนบุคคลบนระบบปฏิบัติการลินุกซ์และยูนิกซ์ ที่อำนวยความสะดวกในการถ่ายโอนจัดการแฟ้มข้อมูล โดยระบบแฟ้มข้อมูลของงานวิจัยนี้ มีส่วนประกอบดังภาพที่ 2-7



ภาพที่ 2-7 ส่วนประกอบของ SPAFS2 [7]

โดยจะทำหน้าที่เปรียบเสมือนมิดเดิลแวร์ (middleware) คอยจัดการการเข้าถึงและถ่ายโอนแฟ้มข้อมูลของผู้ใช้งานไปมาระหว่างเครื่องคอมพิวเตอร์ อุปกรณ์เคลื่อนย้ายได้ รวมไปถึงบริการจัดเก็บข้อมูลบนอินเทอร์เน็ตต่างๆ โดยอัตโนมัติ ซึ่งในสายตาของผู้ใช้งานนั้นจะมองเห็นแฟ้มข้อมูลจากเครื่องปลายทางและบริการจัดเก็บข้อมูลบนอินเทอร์เน็ตต่างๆ เสมือนว่าแฟ้มข้อมูลเหล่านั้นรวมอยู่ในเครื่องของผู้ใช้เอง โดยใช้เทคนิคของระบบแฟ้มข้อมูลรวม (unification file system) เพื่อที่จะทำการรวมแฟ้มข้อมูลเข้าด้วยกัน และยังสามารถใช้งานแฟ้มข้อมูลต่างๆ ในระบบแฟ้มคลาวด์ส่วนบุคคลนี้ได้ไม่ต่างจากแฟ้มข้อมูลที่อยู่ในเครื่องของผู้ใช้เอง โดยเน้นให้ใช้บนอุปกรณ์เคลื่อนย้ายได้เป็นหลัก โดยมีความสามารถการทำงานแบบนอกสาย สำหรับกรณีที่ผู้ใช้ไม่

สามารถเข้าถึงเครือข่ายได้ มีการเก็บแคชแฟ้มข้อมูลที่ผู้ใช้มีการใช้งานเป็นประจำมาเก็บไว้บนเครื่องของผู้ใช้โดยอัตโนมัติ สามารถปรับแต่งการทำงานของระบบได้ง่าย สนับสนุนโพรโตคอลเครือข่ายหลายชนิด มีลักษณะเป็นโมดูลเฟรมเวิร์คหรือเป็นแบบมอดูลาร์ (modular) สามารถที่จะปรับเปลี่ยนการทำงานและเพิ่มความสามารถให้กับระบบได้ผ่านทาง IO Module จึงทำให้ง่ายต่อการพัฒนา

นอกจากนี้ยังมีกระบวนการที่ทำให้ผู้ใช้งานสามารถระบุแฟ้มข้อมูลที่ต้องการใช้งานได้โดยตรง คือ Branch Tag หรือการเพิ่มชื่อ branch ต่อท้ายแฟ้มข้อมูล เช่น แฟ้มข้อมูลชื่อว่า name มาจาก branch1 จะได้ชื่อแฟ้มข้อมูลใหม่คือ name+branch1 ดังนั้นเมื่อมีแฟ้มข้อมูลที่ชื่อว่า name ใน branch1 และ branch2 ก็จะได้แฟ้มข้อมูลที่มีชื่อว่า name+branch1 และ name+branch2 ตามลำดับ ซึ่งจะให้ผู้ใช้สามารถมองเห็นแฟ้มข้อมูลที่มีชื่อซ้ำซ้อนกันได้ โดย SPAFS2 มีประสิทธิภาพในการทำงานเทียบเท่าหรือสูงกว่าระบบแฟ้มข้อมูลอื่นๆ ที่มีคุณสมบัติใกล้เคียงกัน ซึ่งระบบแฟ้มของงานวิจัยนี้ไม่สามารถนำมาประยุกต์ใช้งานบนวินโดว์ได้โดยตรง การใช้งานยังไม่สะดวกเท่าที่ควรเนื่องจากเป็น command line และต้องมีการสั่งและตั้งค่าการทำงานทุกครั้งที่เริ่มระบบใหม่

บทที่ 3

การออกแบบระบบรวมแฟ้มข้อมูล

ในบทนี้จะกล่าวถึงการนำเอาหลักการ แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้องต่างๆ มาประยุกต์ใช้ในการออกแบบระบบรวมแฟ้มข้อมูลในลักษณะแบบส่วนบุคคลที่ทำงานบนระบบปฏิบัติการวินโดว์

3.1 จุดมุ่งหมาย

1) ระบบต้นแบบที่มีความสามารถในการรวบรวมแฟ้มจากหลายแหล่งข้อมูลบนระบบปฏิบัติการวินโดว์ พร้อมทั้งแก้ปัญหาความซ้ำซ้อนของชื่อของแฟ้มข้อมูลอย่างอัตโนมัติ

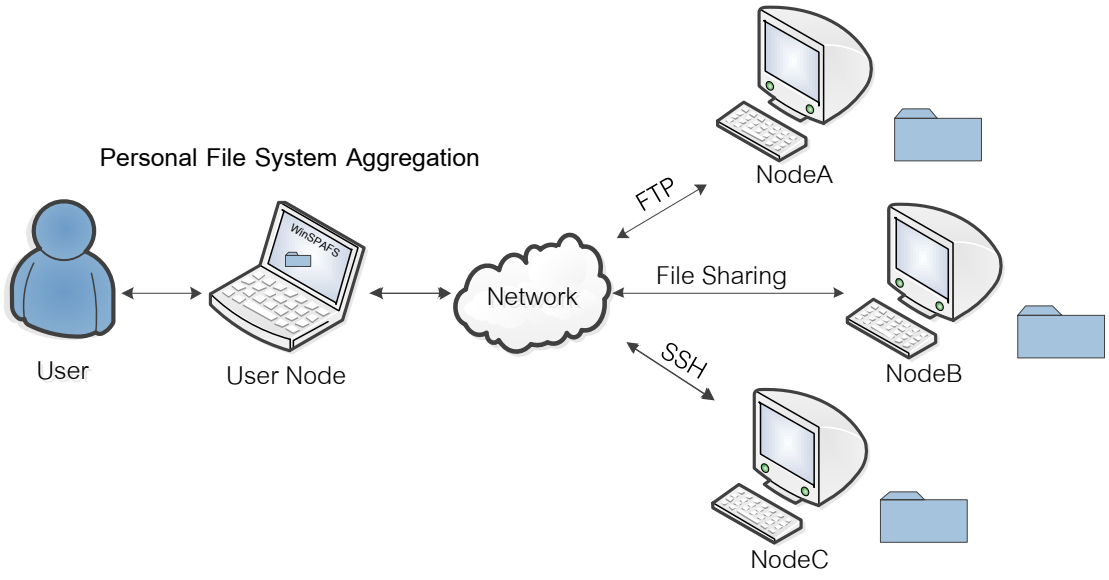
2) ระบบต้นแบบที่ช่วยให้การจัดการและถ่ายโอนแฟ้มข้อมูลทำได้ง่ายขึ้น ลดขั้นตอนและความยุ่งยากในการใช้งาน

3) ระบบต้นแบบที่มีระบบแคชเพื่อช่วยให้ระบบทำงานได้เร็วขึ้นและสนับสนุนการทำงานแบบนอกสายได้อย่างอัตโนมัติ

4) ระบบต้นแบบให้สามารถพัฒนาต่อได้ง่าย

3.2 ภาพรวมของระบบ

ลักษณะของระบบรวมแฟ้มข้อมูลส่วนบุคคลมีสถาปัตยกรรมคล้ายกับระบบแฟ้มแบบกระจาย ดังภาพที่ 3-1

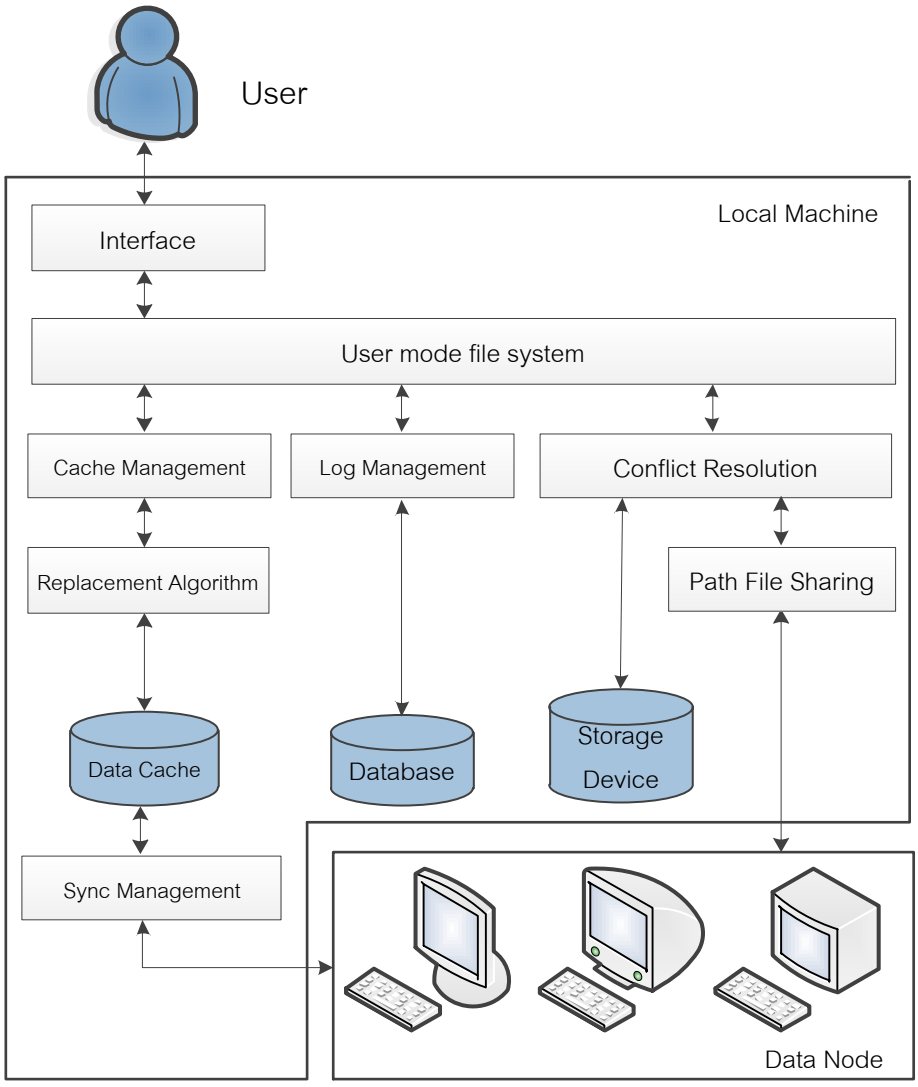


ภาพที่ 3-1 ลักษณะของระบบรวมแฟ้มข้อมูล

ระบบรวมแฟ้มข้อมูลส่วนบุคคลไม่มี Dedicated Server ของตนเอง ไม่ต้องติดตั้งระบบรวมแฟ้มข้อมูลส่วนบุคคลที่เครื่องปลายทาง แต่จะทำการเชื่อมต่อกับเครื่องปลายทางผ่านโพรโตคอลเครือข่ายต่างๆ และสามารถใช้กลไกพื้นฐานการเชื่อมต่อผ่านเครือข่ายที่มีอยู่ เช่น การแชร์ทรัพยากรบนเครือข่าย (File Sharing) เพื่อลดขั้นตอนและความยุ่งยากในการติดตั้ง ซึ่งการพัฒนาให้มีการใช้งานแบบส่วนบุคคลนั้นเพื่อลดความต้องการด้านทรัพยากรของระบบ ลดโอเวอร์เฮดที่เกิดขึ้น ลดปัญหาด้านความปลอยภัย ลดปัญหาการกำหนดสิทธิ์ในการเข้าถึงและลดปัญหาการใช้แฟ้มข้อมูลพร้อมกันได้

3.3 ส่วนประกอบของระบบ

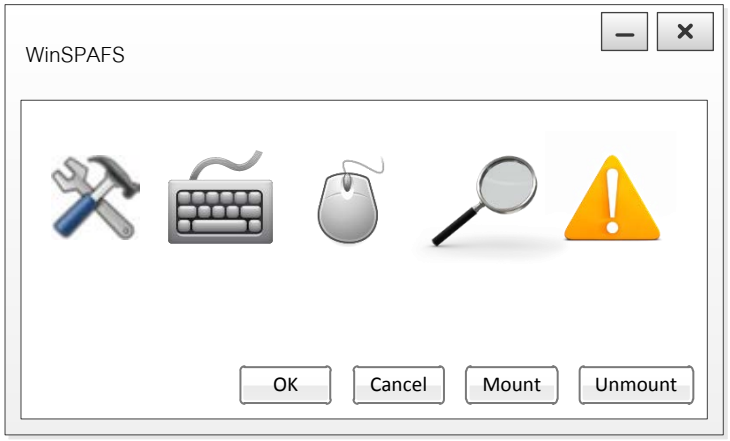
ส่วนประกอบของระบบประกอบไปด้วยหลายส่วนต่างๆ ดังภาพที่ 3-2 อธิบายได้ดังนี้



ภาพที่ 3-2 ส่วนประกอบของระบบรวมแฟ้มข้อมูล

3.3.1 **ส่วนต่อประสาน** (Interface)

เป็นส่วนต่อประสานหรือติดต่อกับผู้ใช้งาน โดยมีลักษณะเป็น Graphic User Interface เพื่อให้ใช้งานได้ง่ายและสะดวก โดยในส่วนนี้จะรวมไปถึงการตั้งค่าการใช้งานต่างๆ เช่น การตั้งค่า Drive letter, Volume label, Path File sharing การเลือกเก็บข้อมูลการใช้งานแฟ้มข้อมูล การกำหนดพาท พื้นที่ในการจัดเก็บแฟ้มข้อมูลและเงื่อนไขในการเก็บแคชอีกด้วย



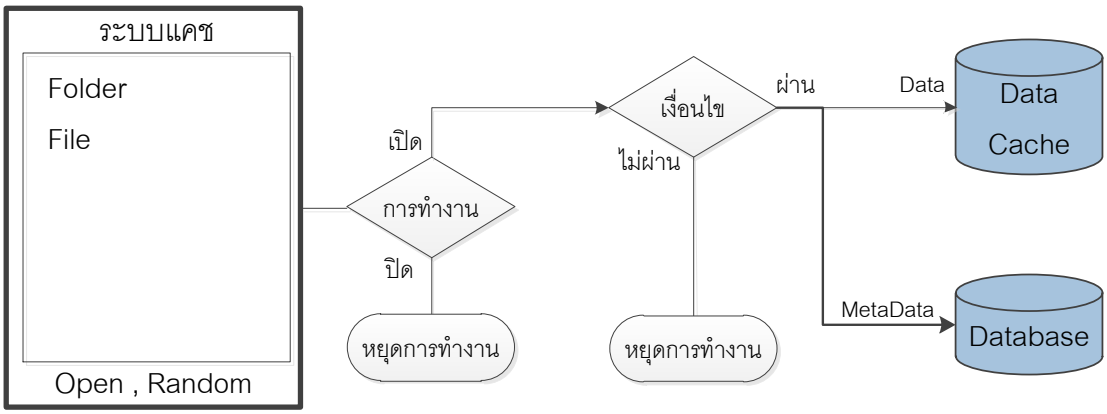
ภาพที่ 3-3 Interface

3.3.2 **โดคานยูเซอร์โหมด** (User Mode File System for Windows)

ใช้ในการพัฒนาส่วนต่างๆ ของระบบในยูเซอร์สเปซบนระบบปฏิบัติการวินโดว์โดยใช้โดคาน และยังเป็นส่วนที่เชื่อมส่วนประกอบต่างๆ ในระบบเข้าด้วยกัน

3.3.3 Cache Management

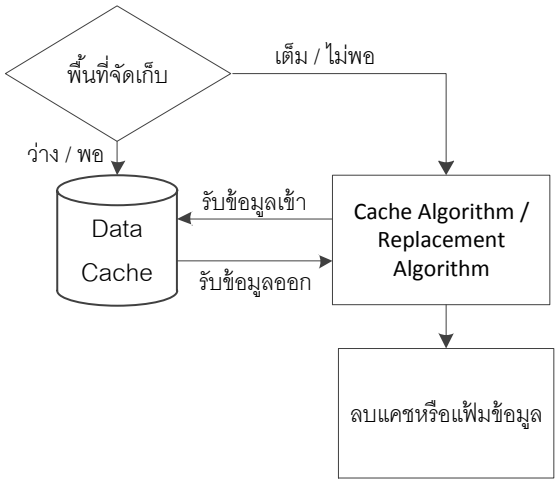
ใช้จัดการและจัดเก็บแคช รวมไปถึงเงื่อนไขต่างๆ ในการคัดเลือกแฟ้มข้อมูลเข้าไปเก็บในสตอเรจดีไวส์ (Storage device)



ภาพที่ 3-4 Cache Management

3.3.4 Replacement Algorithm

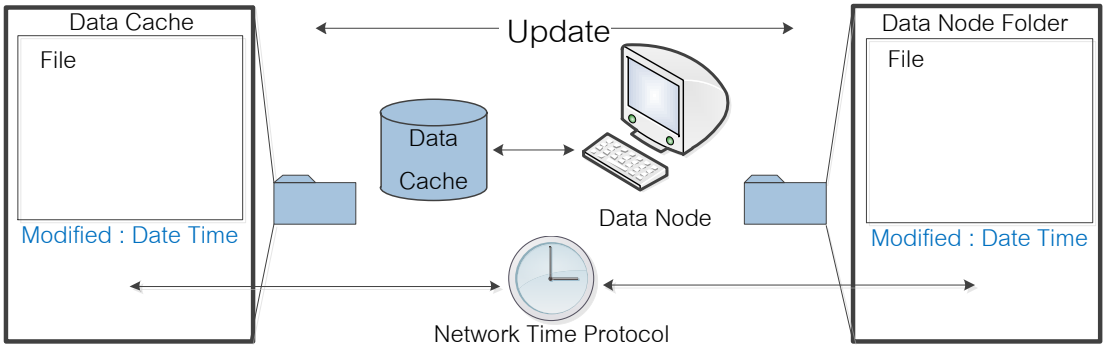
ใช้จัดการแฟ้มข้อมูลที่ถูกจัดเก็บอยู่ใน Data Cache โดยใช้อัลกอริทึมการแทนที่แบบแอลอาร์ยูในการสับเปลี่ยนแฟ้มข้อมูลเมื่อมีแฟ้มข้อมูลใหม่เข้ามา



ภาพที่ 3-5 Replacement Algorithm

3.3.5 Sync Management

ใช้ในการจัดการเปรียบเทียบและตรวจสอบสถานะของแฟ้มข้อมูลที่อยู่ใน Data Cache กับเครื่องปลายทางว่าแฟ้มข้อมูลใดเป็นแฟ้มข้อมูลล่าสุด นอกจากนี้แล้วยังเป็นส่วนที่จะทำการอัพเดทหรือลบแฟ้มข้อมูลใน Data Cache โดยใช้ NTP ของ time.windows.com ซึ่งสามารถเลือกใช้ NTP ได้เองตามความต้องการ



ภาพที่ 3-6 Sync Management

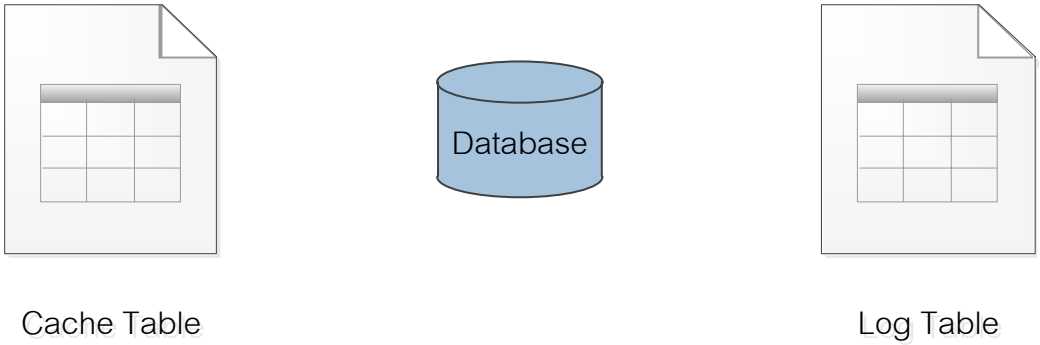
3.3.6 Data Cache

ใช้ในการเก็บ Cache หรือแฟ้มข้อมูลไว้ใช้ในกรณีที่เครือข่ายถูกตัดขาดและในกรณีที่ผู้ใช้ใช้งานแฟ้มข้อมูลบางแฟ้มข้อมูลที่ได้เก็บไว้ใน Data Cache ทำให้สามารถเรียกใช้งานแฟ้มข้อมูล

นั้นได้โดยตรง โดยไม่ต้องทำการดึงแฟ้มข้อมูลนั้นมาจากเครื่องปลายทาง จึงช่วยให้การเข้าถึงแฟ้มข้อมูลทำได้อย่างรวดเร็ว

3.3.7 Database

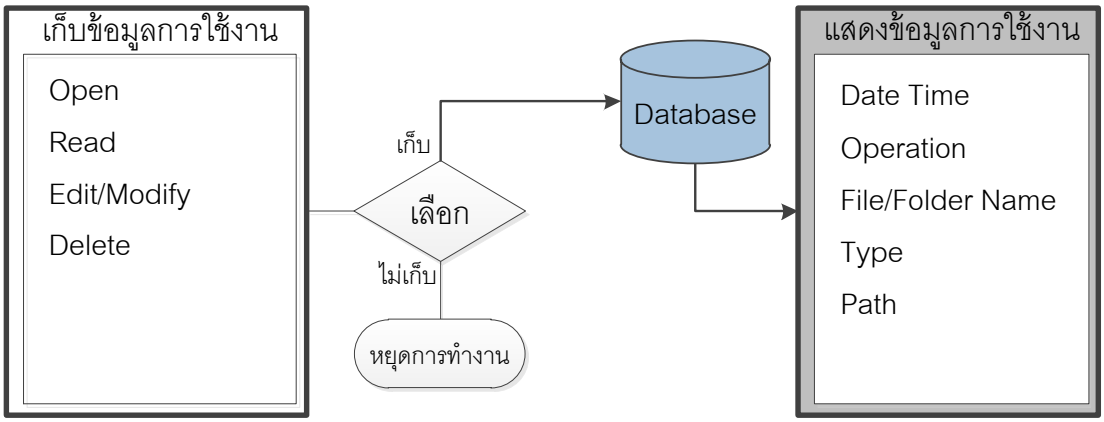
ใช้จัดเก็บข้อมูลการใช้งานแฟ้มข้อมูลต่างๆ ของระบบและเก็บ Metadata ของระบบแคช โดยจะแบ่งออกเป็น 2 ตาราง เก็บด้วยฐานข้อมูล MySQL ซึ่งเป็นฐานข้อมูลที่ใช้งานง่ายและเป็นที่แพร่หลาย



ภาพที่ 3-7 Database

3.3.8 Log Management

ใช้จัดการข้อมูลเกี่ยวกับการใช้งานแฟ้มข้อมูลต่างๆ เช่น การเปิด อ่าน แก้ไข ลบ แฟ้มข้อมูลต่างๆ ที่อยู่ในเมาท์พอยท์ (Mount Point)



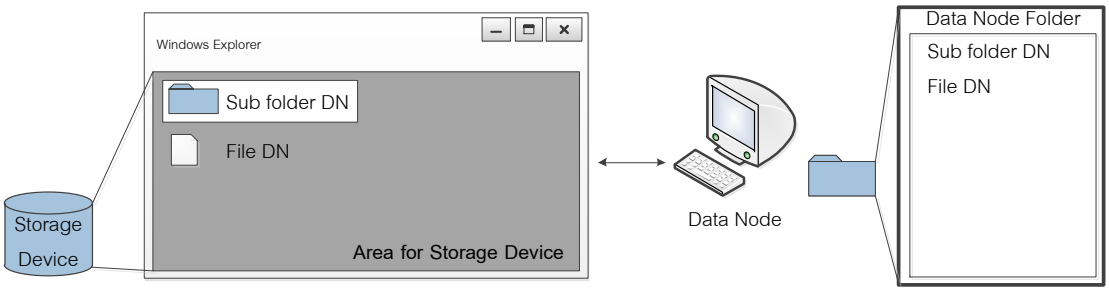
ภาพที่ 3-8 จัดการข้อมูลเกี่ยวกับการใช้งานแฟ้มข้อมูล

3.3.9 Conflict Resolution

ใช้ในการแก้ไขความขัดแย้งที่เกิดขึ้นของแฟ้มข้อมูล เช่น กรณีที่ชื่อแฟ้มข้อมูลซ้ำกัน

3.3.10 Storage Device

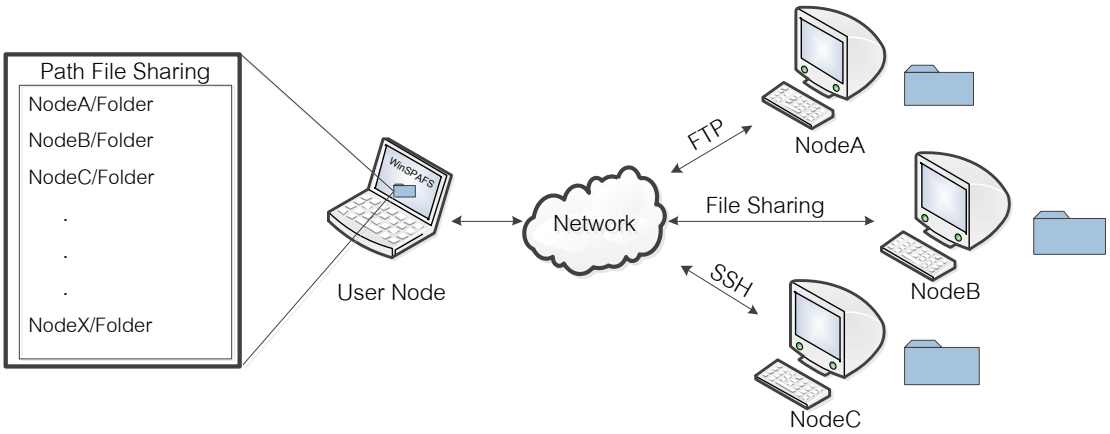
ใช้ในการเก็บแฟ้มข้อมูลที่ถูกสร้างขึ้นมาในเมาท์พอยท์ในกรณีที่ไม่ได้สร้างแฟ้มข้อมูลหรือโฟลเดอร์นั้นในโฟลเดอร์ที่มาจากเครื่องปลายทาง



ภาพที่ 3-9 Storage Device

3.3.11 Path File sharing

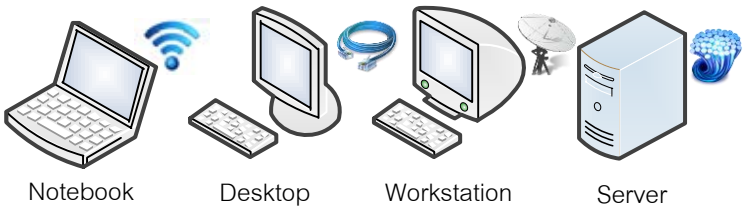
ใช้จัดการและจัดเก็บพาทของเครื่องปลายทางต่างๆ ที่ต้องการเชื่อมต่อกับระบบ



ภาพที่ 3-10 Path File sharing

3.3.12 Data Node

เครื่องปลายทางที่ต้องการเชื่อมต่อกับระบบ โดยเป็นเครื่องที่สามารถเชื่อมต่อกับระบบเครือข่ายได้

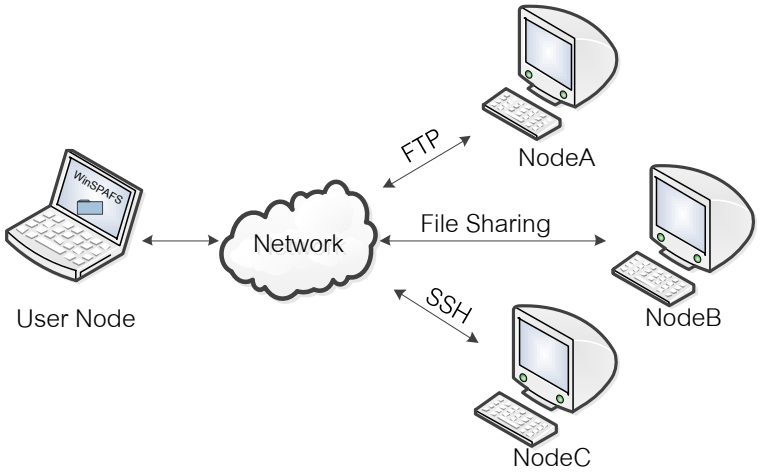


ภาพที่ 3-11 Data Node

3.4 ลักษณะการทำงานของระบบ

3.4.1 การเชื่อมต่อกับเครื่องปลายทาง

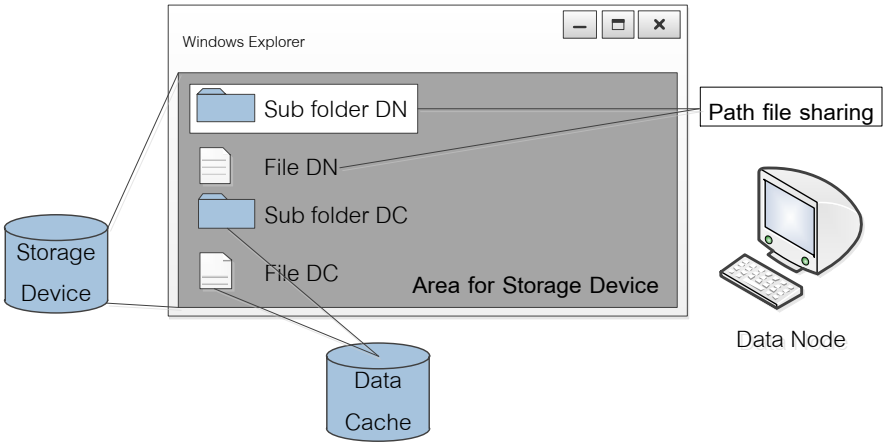
การเชื่อมต่อกับเครื่องปลายทางจะทำการเชื่อมต่อผ่านทางโปรโตคอล TCP/IP โดยใช้พื้นฐานความสามารถของการแชร์แฟ้มข้อมูล เนื่องจากการแชร์แฟ้มข้อมูลสามารถทำได้ง่าย ช่วยลดความยุ่งยากและขั้นตอนในการเชื่อมต่อ นอกจากนี้ยังใช้ร่วมกับระบบแชร์แฟ้มข้อมูลบนระบบปฏิบัติการอื่นๆ และนำไปพัฒนาให้ใช้กับโพรโตคอลการเชื่อมต่อแบบต่างๆ ได้อีกด้วย



ภาพที่ 3-12 ลักษณะการเชื่อมต่อกับเครื่องปลายทาง

3.4.2 การรวมแฟ้มข้อมูล

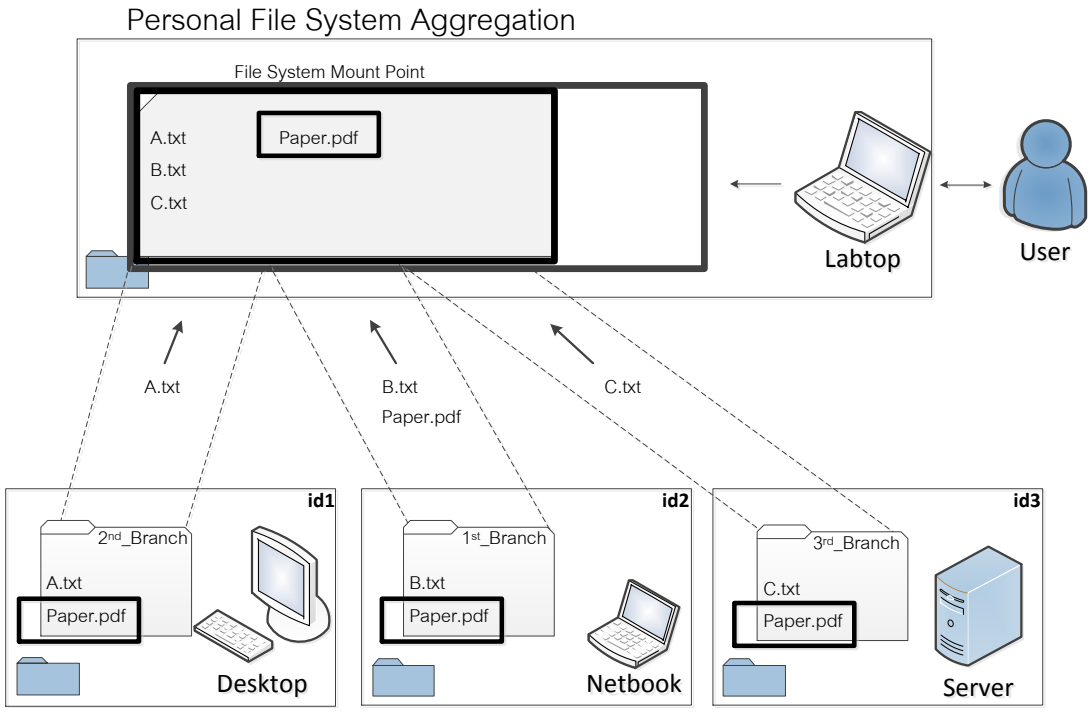
การรวมแฟ้มข้อมูลจะทำการรวมแฟ้มข้อมูลที่อยู่ในเครื่องปลายทาง ที่มีการระบุพาทไว้ใน Path Files sharing ที่มาจากการเชื่อมต่อกับเครื่องปลายทาง รวมถึงแฟ้มข้อมูลต่างๆ ที่เก็บไว้ใน Data Cache และ Storage Device ด้วย ตัวอย่างเช่น ระบบ Union File System [7]



ภาพที่ 3-13 การรวมแฟ้มข้อมูล

3.4.3 การแก้ไขปัญหาการขัดแย้งของข้อมูล

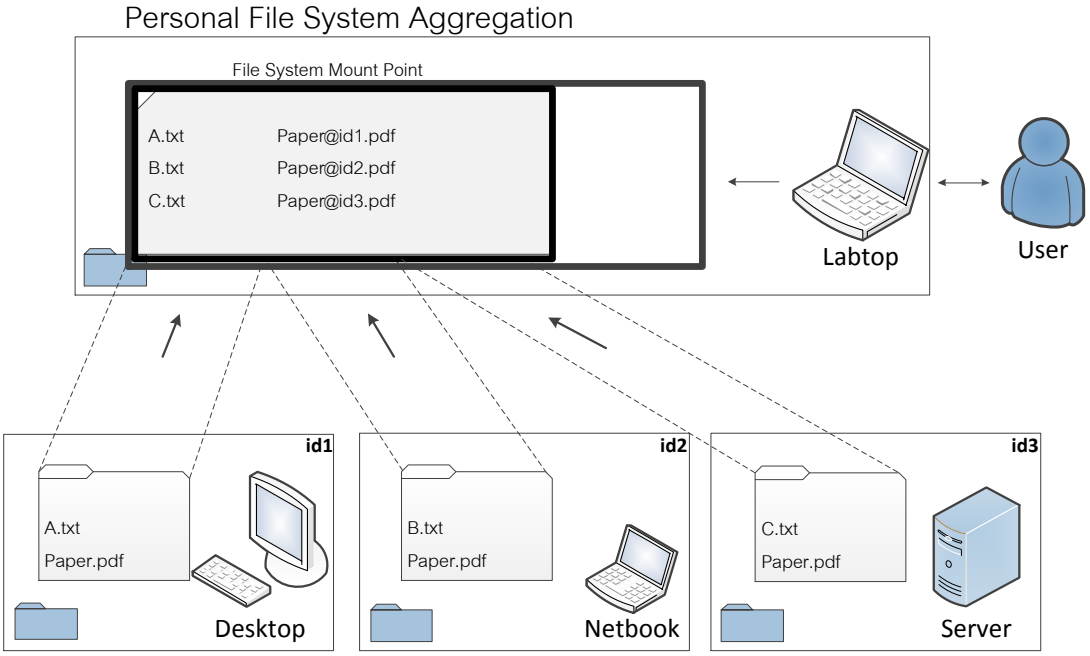
การแก้ไขปัญหาความขัดแย้งของข้อมูลจะใช้แนวคิดพื้นฐานของระบบแฟ้มยูนิฟิเคชั่นบางส่วน คือ บรานช์ ในกรณีที่ทำการรวมแฟ้มข้อมูลที่มีชื่อซ้ำกันในระบบ ระบบจะเลือกแฟ้มข้อมูลที่อยู่ในบรานช์ที่อยู่สูงกว่าแสดงให้เห็นเพียงแฟ้มข้อมูลเดียว จึงช่วยแก้ไขความขัดแย้งในกรณีที่แต่ละบรานช์มีแฟ้มข้อมูลที่มีชื่อซ้ำหรือเหมือนกันได้ แสดงให้เห็นดังภาพที่ 3-14



ภาพที่ 3-14 ผลที่ได้จากแก้ปัญหาด้วยแนวคิดพื้นฐานของระบบแฟ้มยูนิฟิเคชั่น

เนื่องจากระบบแฟ้มยูนิฟิเคชั่นจะเลือกแฟ้มข้อมูลที่มีชื่อซ้ำกันเหล่านั้นให้ผู้ใช้แสดงให้เห็นเพียงแค่แฟ้มข้อมูลเดียว ในกรณีที่ผู้ใช้ต้องการใช้งานแฟ้มข้อมูลอื่นๆ ที่มีชื่อเดียวกันนี้ ไม่สามารถทำได้ จึงได้นำวิธีระบุพาทของแฟ้มข้อมูลต่อท้ายชื่อแฟ้มข้อมูลนั้นๆ มาใช้ด้วย ซึ่งจะช่วยให้มองเห็นแฟ้มข้อมูลที่ชื่อเหมือนกันทั้งหมดได้ ดังภาพที่ 3-15

ประเด็นคือ จะทำให้ระบบขาดความโปร่งใส (Transparency) เนื่องจากผู้ใช้จะรู้ว่าแฟ้มข้อมูลต่างๆ ที่มีพาทต่อท้ายนั้น เก็บอยู่ที่เครื่องปลายทางใด ซึ่งไม่ใช่หลักการของระบบแฟ้มแบบกระจาย แต่เนื่องจากงานวิจัยนี้ นำหลักการของระบบแฟ้มแบบกระจายมาประยุกต์ใช้ในบางส่วนเท่านั้น ทั้งยังใช้งานในแบบส่วนบุคคลหรือผู้ใช้งานเป็นผู้จัดการระบบเองทั้งหมด และในบางครั้งการรับรู้ว่าแฟ้มข้อมูลเหล่านั้นเก็บอยู่ที่เครื่องปลายทางใด ไม่ได้เป็นผลเสียต่อระบบแต่อย่างใด



ภาพที่ 3-15 ผลที่ได้จากการใช้วิธีระบุพาทของแฟ้มข้อมูลต่อท้ายชื่อแฟ้มข้อมูล

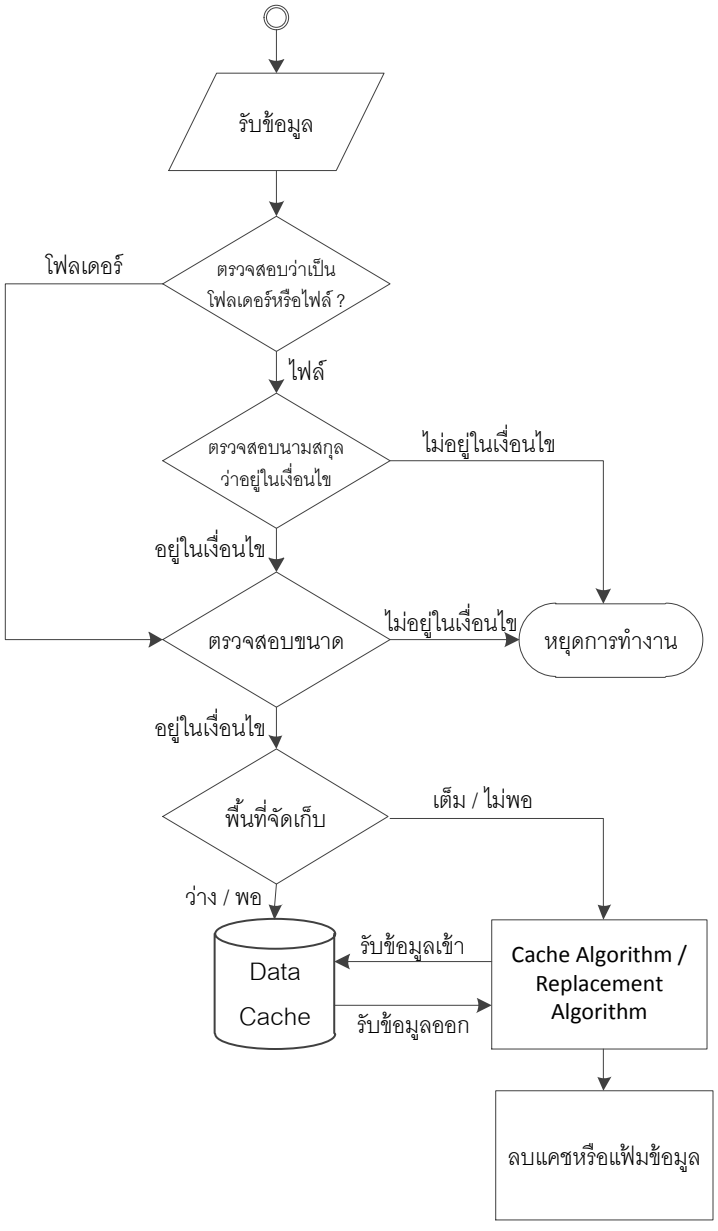
3.4.4 ระบบแคช

ระบบแคชแบ่งเป็นสองส่วน คือ การเก็บแคชและการจัดการพื้นที่เก็บแคช ในส่วนแรกคือส่วนของการเก็บแคชหรือแฟ้มข้อมูลลง Data Cache โดยก่อนที่จะจัดเก็บแคชหรือแฟ้มข้อมูลลง Data Cache ต้องผ่านขั้นตอนการคัดเลือกแคชหรือแฟ้มข้อมูลก่อน

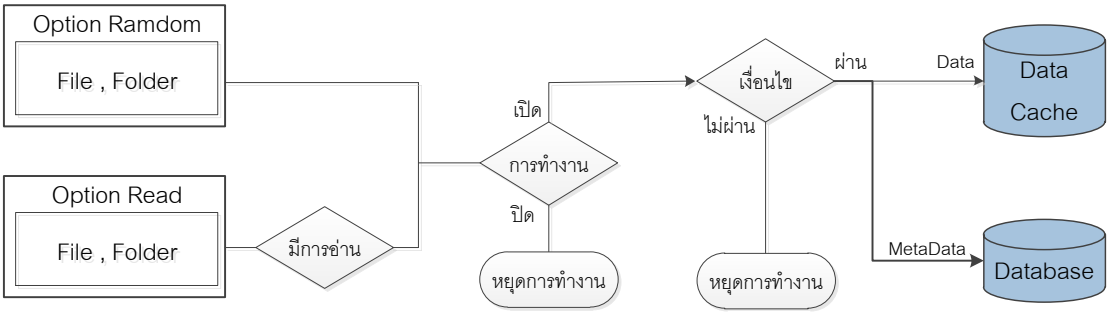
เมื่อทำการอ่านแฟ้มข้อมูลใดๆ จะเป็นการรับข้อมูลเพื่อตรวจสอบชนิดและขนาดว่าอยู่ในเงื่อนไขในการเก็บแคชหรือแฟ้มข้อมูลหรือไม่ เช่น ไม่ต้องการเก็บแฟ้มข้อมูลชนิด MP3 และแฟ้มข้อมูลมีขนาดใหญ่เกิน 1 กิกะไบต์ เป็นต้น ถ้าอยู่ในเงื่อนไขจะมีการจัดเก็บแฟ้มข้อมูลลง Data Cache อย่างอัตโนมัติ

นอกจากนี้ยังสามารถเลือกจัดเก็บแฟ้มข้อมูลโดยการสุ่ม (Random) ได้อีกด้วย โดยเมื่อระบบเริ่มทำงานจะมีการสุ่มแฟ้มข้อมูล โดยจะเป็นการรับข้อมูลเพื่อตรวจสอบชนิดและขนาดว่าอยู่ในเงื่อนไขในการเก็บแคชหรือแฟ้มข้อมูลหรือไม่ ถ้าอยู่ในเงื่อนไขจะมีการจัดเก็บแฟ้มข้อมูลลง Data Cache อย่างอัตโนมัติด้วยเช่นเดียวกัน ดังภาพที่ 3-17 ซึ่งจะดีกว่าการอ่านแฟ้มข้อมูลนั้นก่อนจึงจะมีการจัดเก็บ เนื่องจากในกรณีที่ผู้ใช้เริ่มใช้งานระบบแต่ยังไม่ได้ทำการอ่านแฟ้มข้อมูลใดๆ และระบบเกิดการขาดการติดต่อกับระบบเครือข่าย ผู้ใช้จะไม่สามารถใช้แฟ้มข้อมูลที่อยู่ใน Data Cache ได้เลย เนื่องจากแฟ้มข้อมูลต่างๆ นั้นยังไม่ถูกจัดเก็บลง Data Cache แต่ข้อเสียก็คือระบบต้องทำการดาวน์โหลดแฟ้มข้อมูลจากเครื่องปลายทางมายังเครื่องผู้ใช้งาน และในบางกรณี

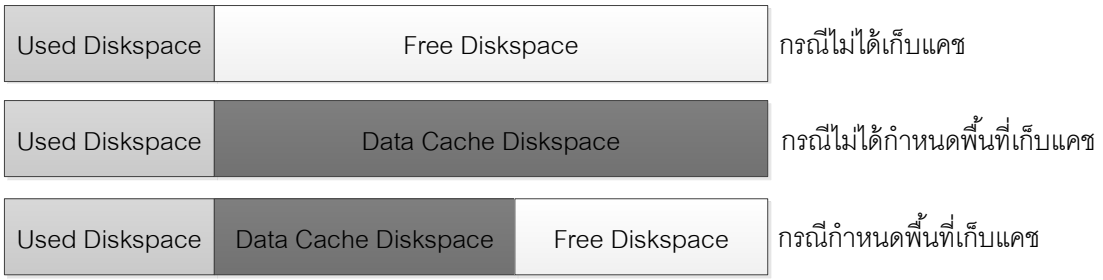
แฟ้มข้อมูลที่ถูกดาวน์โหลดมายังเครื่องผู้ใช้ก็ไม่ใช่แฟ้มข้อมูลที่ผู้ใช้ต้องการงาน ทำให้เกิดการสิ้นเปลืองทรัพยากรในระบบอย่างเปล่าประโยชน์ แต่ด้วยการกำหนดขนาดและเงื่อนไขในการเก็บแคชทำให้เกิดการคัดเลือกแฟ้มข้อมูลที่อาจจะตรงกับความต้องการและได้ประโยชน์มากที่สุด โดยถ้าระบบไม่ได้มีการกำหนดขนาดพื้นที่ในการเก็บแคชจะทำให้เกิดปัญหาการใช้งานพื้นที่จัดเก็บข้อมูลบนฮาร์ดดิสก์ได้ ดังภาพที่ 3-18



ภาพที่ 3-16 เงื่อนไขในการเก็บและจัดการแคช

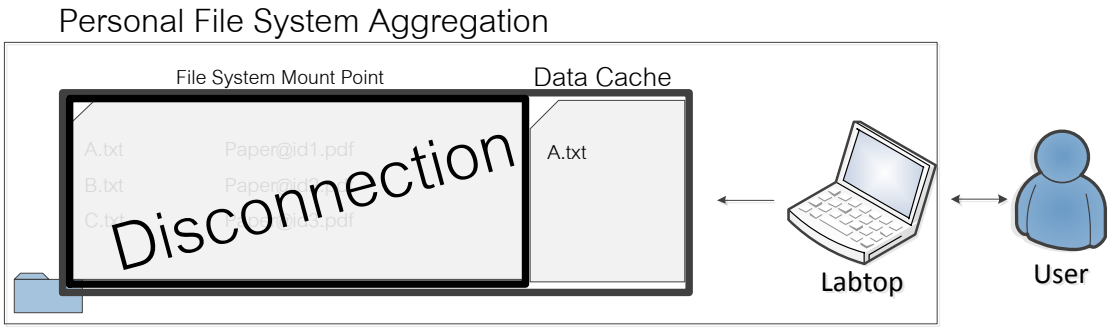


ภาพที่ 3-17 ตัวเลือกการจัดเก็บแคช

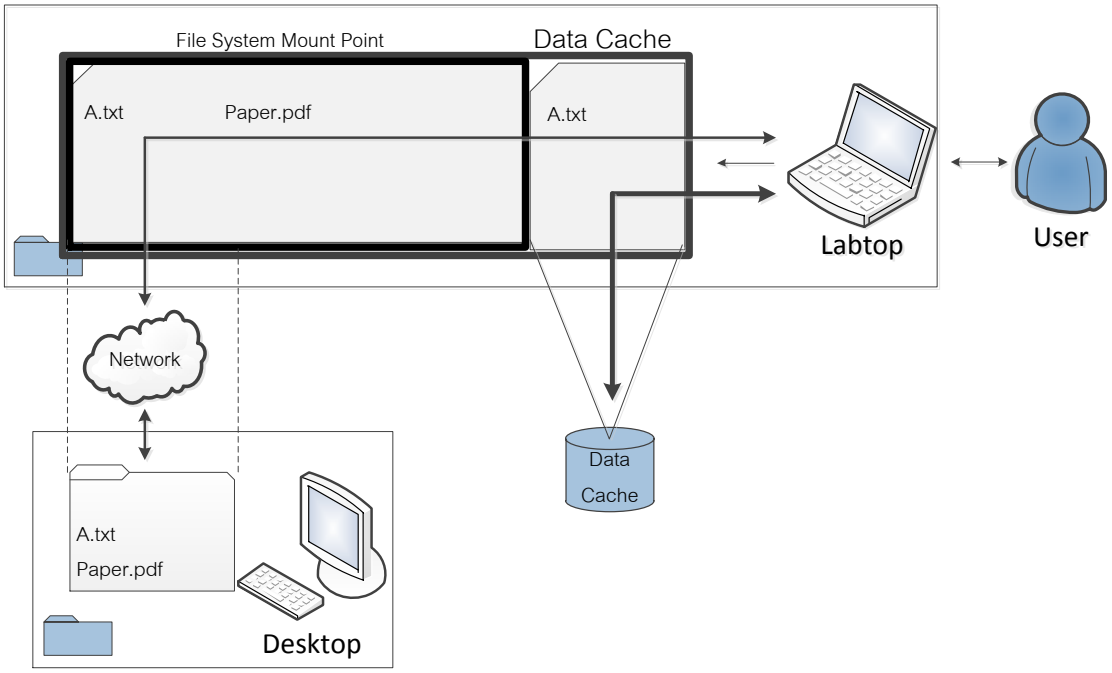


ภาพที่ 3-18 พื้นที่จัดเก็บข้อมูลบนฮาร์ดดิสก์

ส่วนที่สองคือ ส่วนของการจัดการพื้นที่เก็บแคชหรือแฟ้มข้อมูล ผู้ใช้สามารถกำหนดพื้นที่ของการจัดเก็บแคชหรือแฟ้มข้อมูลใน Data Cache ได้เอง เพื่อให้เกิดผลกระทบกับพื้นที่การใช้งานฮาร์ดดิสก์ของคอมพิวเตอร์ที่ผู้ใช้ใช้งานเท่าที่ต้องการเท่านั้น และเมื่อพื้นที่ใน Data Cache เต็ม ระบบจะใช้อัลกอริทึมการแทนที่แอลอาร์ยูในการสับเปลี่ยนแฟ้มข้อมูล แสดงให้เห็นดังภาพที่ 3-19 ทำให้ผู้ใช้สามารถใช้งานแฟ้มข้อมูลของระบบบางส่วนที่ถูกจัดเก็บอยู่ใน Data Cache ได้แม้ว่าระบบจะขาดการเชื่อมต่อ แสดงให้เห็นดังภาพที่ 3-20 ซึ่งระบบแคชอาจทำให้ระบบมีความเร็วในการทำงานเพิ่มขึ้นในกรณีที่มีการเรียกใช้งานแฟ้มข้อมูลที่ถูกเก็บภายใน Data Cache ได้โดยตรงแทนที่จะเรียกใช้งานแฟ้มข้อมูลจากเครื่องปลายทาง



ภาพที่ 3-19 ผลที่ได้จากการเก็บแคช



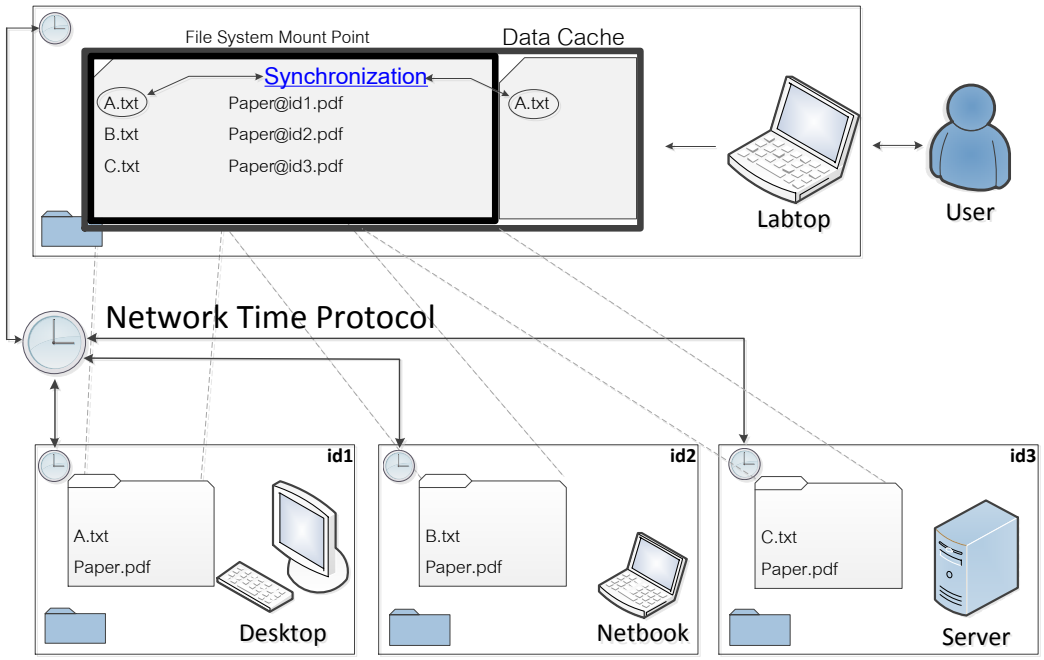
ภาพที่ 3-20 การใช้งานแฟ้มข้อมูลจาก Data Cache โดยตรง

3.4.5 การจัดเก็บข้อมูลการใช้งาน

การเก็บข้อมูลการใช้งานแฟ้มข้อมูลช่วยให้ผู้ใช้สามารถบันทึกและเรียกดูรายละเอียดของการสร้าง อ่าน แก้ไข หรือลบแฟ้มข้อมูลที่อยู่ในเมาท์พอยท์ (Mount Point) ต่างๆ ย้อนหลังได้ สามารถค้นหาโดยใช้คำค้น (Keyword) ได้ โดยจะเก็บวันเวลา ชื่อ ชนิด พาท ขนาด ของแฟ้มข้อมูล ซึ่งจะเก็บไว้ในฐานข้อมูล MySQL โดยสามารถเลือกเก็บการกระทำต่างของแฟ้มข้อมูลและปิดการทำงานในส่วนนี้ได้ถ้าไม่ต้องการ เพื่อช่วยลดโอเวอร์เฮดที่เกิดขึ้นได้

3.4.6 การรักษาสภาวะแฟ้มข้อมูล

เมื่อการเชื่อมต่อผ่านทางระบบเครือข่ายถูกตัดขาด ผู้ใช้งานจะยังสามารถใช้งานและแก้ไขแฟ้มข้อมูลบางส่วนที่ถูกจัดเก็บใน Data Cache แทนได้ และเมื่อมีการเชื่อมต่อเกิดขึ้นภายหลัง ผู้ใช้สามารถเปรียบเทียบและตรวจสอบสถานะของแฟ้มข้อมูลที่อยู่ใน Data Cache กับเครื่องปลายทางได้ว่า แฟ้มข้อมูลใดเป็นแฟ้มข้อมูลล่าสุด ถ้าพบว่าแฟ้มข้อมูลที่ถูกเก็บอยู่ใน Data Cache เป็นแฟ้มข้อมูลที่มีการเปลี่ยนแปลงเนื้อหาภายในล่าสุด ผู้ใช้สามารถทำการอัพเดทแฟ้มข้อมูลนี้ไปยังเครื่องปลายทางได้ แต่ถ้าพบว่าแฟ้มข้อมูลที่ถูกเก็บอยู่ใน Data Cache ไม่ได้เป็นแฟ้มข้อมูลที่มีการเปลี่ยนแปลงเนื้อหาภายในล่าสุด ผู้ใช้สามารถทำการลบแฟ้มข้อมูลนี้ได้ โดยมีลักษณะการทำงานดังภาพที่ 3-21



ภาพที่ 3-21 การรักษาสภาวะของแฟ้มข้อมูล

บทที่ 4

การพัฒนาระบบรวมแฟ้มข้อมูลส่วนบุคคล

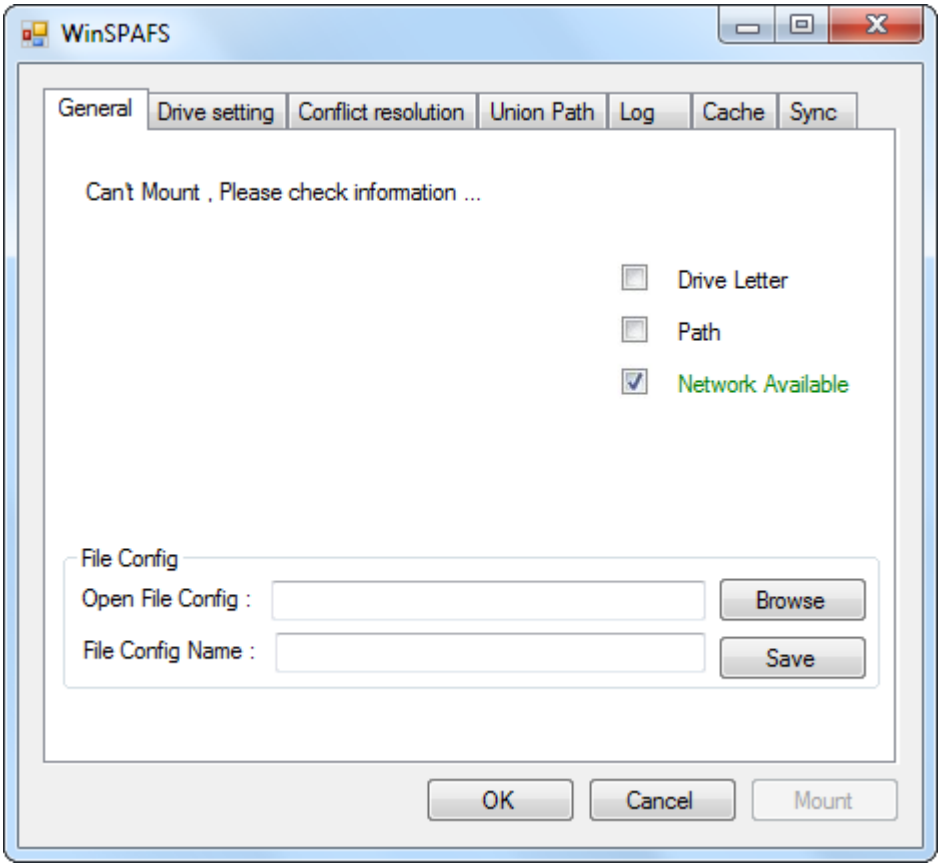
ในบทนี้จะกล่าวถึงรายละเอียดการพัฒนาระบบรวมแฟ้มข้อมูลส่วนบุคคลบนระบบปฏิบัติการวินโดว์ด้วยโดคาน ดังนี้

4.1 ระบบรวมแฟ้มข้อมูลส่วนบุคคลบนระบบปฏิบัติการวินโดว์ด้วยโดคาน (WinSPAFS)

การพัฒนาระบบรวมแฟ้มส่วนบุคคลบนระบบปฏิบัติการวินโดว์นั้น จะทำการพัฒนาด้วยโดคานซึ่งเป็นโอเพนซอร์ส ใช้โดคานดอทเน็ตไบน์ดิงและภาษาดอตเน็ต คือ ซีชาร์ฟ (C#)

4.1.1 การพัฒนาส่วนต่อประสานงานกับผู้ใช้

งานวิจัยนี้ได้ทำการพัฒนาส่วนต่อประสานกับผู้ใช้งานให้มีลักษณะเป็น Graphic User Interface (GUI) ดังภาพที่ 4-1 เพื่อให้ระบบมีลักษณะที่น่าใช้และการใช้งานสามารถทำได้ง่ายสะดวกและรวดเร็ว



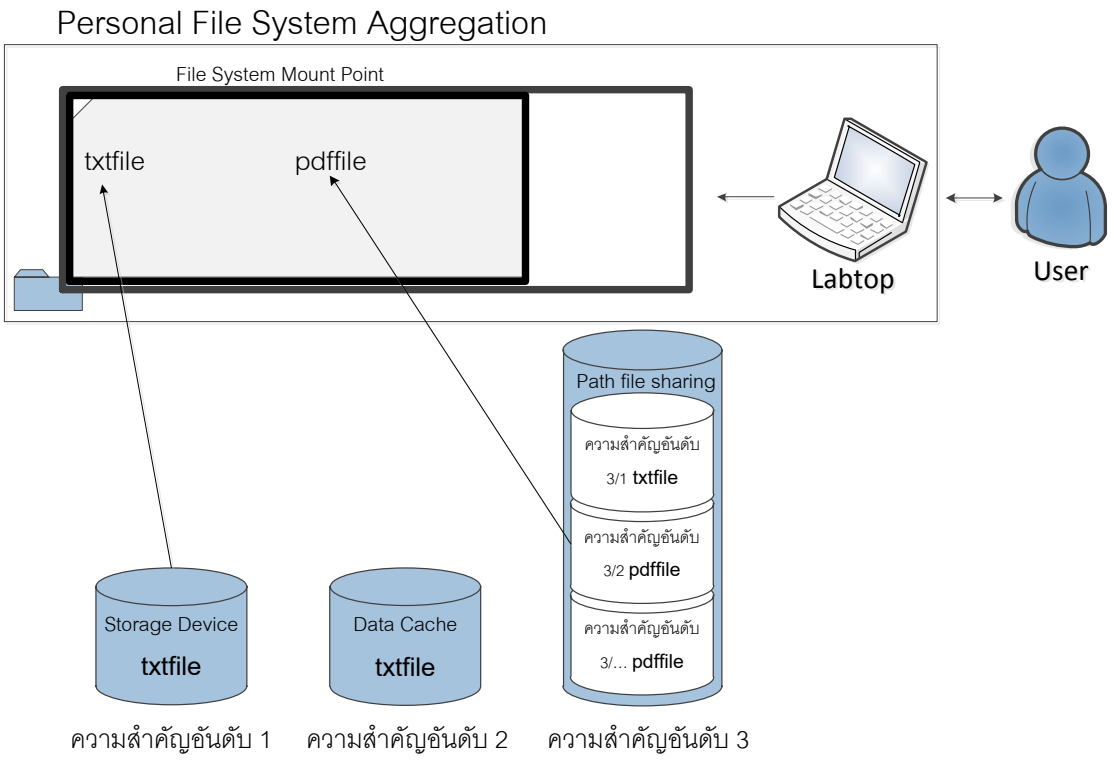
ภาพที่ 4-1 ส่วนต่อประสานงานกับผู้ใช้

4.2 การแก้ไขความขัดแย้งของข้อมูล

การแก้ไขความขัดแย้งของข้อมูลที่เกิดจากแฟ้มข้อมูลที่ต้องการนำมารวมกันในเมาส์พอยท์มีชื่อซ้ำกัน โดยจะใช้วิธีการจัดลำดับความสำคัญและการระบุชื่อพาทต่อท้ายแฟ้มข้อมูลเพื่อให้ระบบมีประสิทธิภาพยิ่งขึ้น

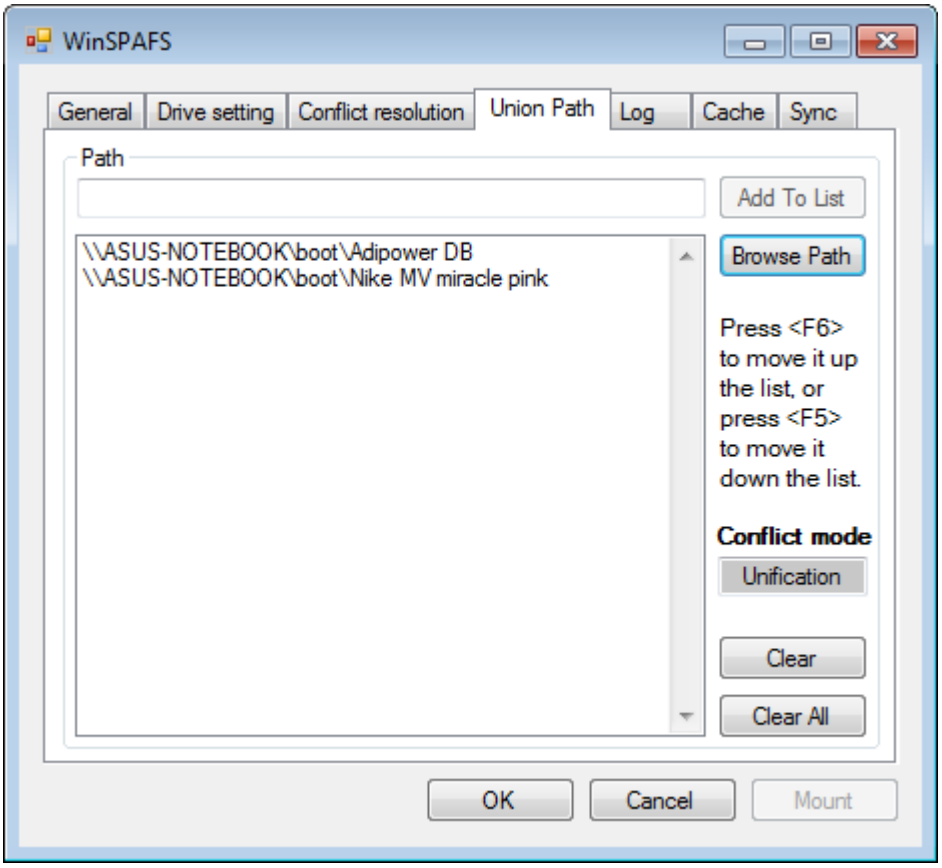
4.2.1 การจัดลำดับความสำคัญของแฟ้มข้อมูล

การจัดลำดับความสำคัญของแฟ้มข้อมูลมีความสำคัญต่อระบบรวมแฟ้มข้อมูลอย่างมาก เนื่องจากเป็นวิธีที่ใช้ในการแก้ไขปัญหาความขัดแย้งของแฟ้มข้อมูลในกรณีที่มีแฟ้มข้อมูลที่มีชื่อซ้ำกันในเมาส์พอยท์ ซึ่งการจัดลำดับความสำคัญของแฟ้มข้อมูลจะขึ้นอยู่กับการจัดลำดับบรานช์จะเริ่มจากการเลือกพาทที่อยู่ในระบบทั้งหมด โดยให้ความสำคัญกับพาทที่อยู่ใน Storage Device มากที่สุด รองลงมาคือ Data Cache และพาทที่มาจากเครื่องปลายทาง (Path file sharing) ตามลำดับ เนื่องจากพาทที่มาจากเครื่องปลายทางนั้นอาจจะมีหลายพาท ระบบจะใช้วิธีในการให้ความสำคัญย่อยลงไปอีก คือ ดูลำดับในรายการของพาทที่มาจากเครื่องปลายทาง (Path file sharing) พาทใดอยู่บนสุดจะมีความสำคัญมากกว่าพาทที่อยู่ด้านล่าง ดังภาพที่ 4-2 แสดงการจัดลำดับความสำคัญของแฟ้มข้อมูลในระบบ จะเห็นได้ว่าแฟ้มข้อมูลที่ถูกเลือกมา คือ txtfile และ pdffile ซึ่งซ้ำกันอยู่ในระบบจะถูกคัดเลือกโดยดูจากอันดับความสำคัญที่มากกว่า



ภาพที่ 4-2 การจัดลำดับความสำคัญของแฟ้มข้อมูลในระบบ

พาทที่มากจากเครื่องปลายทางนั้นโดยปกติแล้วพาทที่ถูกเลือกก่อนจะอยู่ลำดับบนสุดในรายการของพาทที่มาจากเครื่องปลายทาง แต่ผู้ใช้สามารถสลับลำดับได้ภายหลัง หมายความว่าสามารถเลือกพาทโดยไม่คำนึงถึงลำดับที่ แต่สามารถจะจัดลำดับความสำคัญภายหลังได้ ดังภาพที่ 4-3 นอกจากการเลือกโดยการ Browse Path และ ยังสามารถเพิ่มพาทด้วยการพิมพ์พาทเองได้อีกด้วย



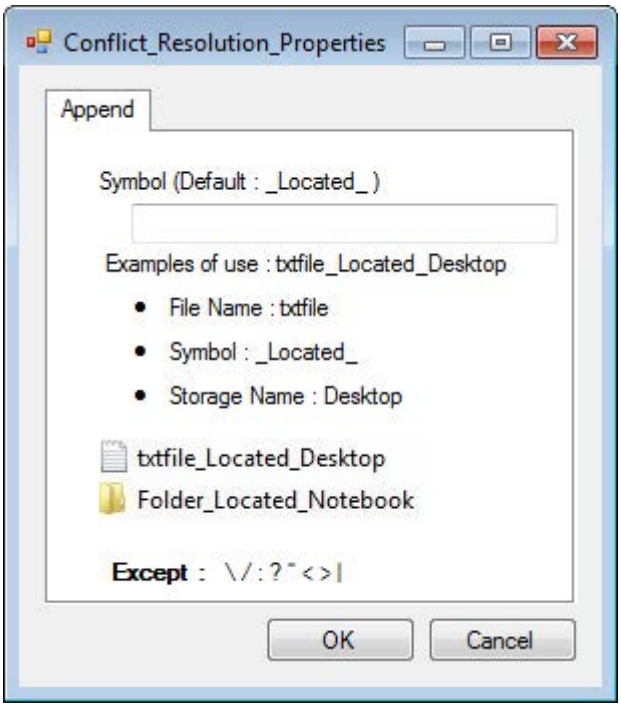
ภาพที่ 4-3 การจัดลำดับความสำคัญของแฟ้มข้อมูลที่มาจากเครื่องปลายทาง

4.2.2 การระบุชื่อพาทต่อท้ายชื่อแฟ้มข้อมูล

การระบุชื่อพาทต่อท้ายแฟ้มข้อมูลนั้น เราจะใช้วิธีระบุชื่อเครื่องปลายทางต่อท้ายแฟ้มข้อมูลแทนการใช้ชื่อพาท เพื่อไม่ให้ผิดกฎการตั้งชื่อแฟ้มข้อมูลและลดความยาวของชื่อลง โดยระบบจะสร้างสัญลักษณ์ที่ใช้คั่นระหว่างชื่อแฟ้มข้อมูลกับเครื่องปลายทางของแฟ้มข้อมูลนั้นเพื่อให้ง่ายและป้องกันความผิดพลาดที่อาจจะเกิดขึ้นจากการตั้งชื่อ เช่น ในระบบมีเครื่องปลายทางที่ชื่อว่า A และในระบบรวมแฟ้มข้อมูลมีแฟ้มข้อมูลที่ชื่อว่า fileA ซึ่งไม่แน่ว่าแฟ้มข้อมูลที่ชื่อว่า file มาจากเครื่อง A แต่ถ้าใช้สัญลักษณ์คั่นระหว่างชื่อแฟ้มข้อมูลกับเครื่องปลายทาง เช่น ใช้สัญลักษณ์ว่า !@#$% (แต่ต้องมั่นใจว่า สัญลักษณ์ที่กำหนดขึ้นมานี้ไม่มีอยู่ในชื่อแฟ้มข้อมูลทั้ง

ระบบ) ผลที่ได้คือ สามารถแน่ใจได้ว่าแฟ้มข้อมูลที่มีสัญลักษณ์ว่า !@#$% เป็นแฟ้มข้อมูลที่มาจากเครื่องปลายทางแน่นอน

โดยผู้ใช้งานสามารถกำหนดสัญลักษณ์ที่ใช้คั่นระหว่างชื่อแฟ้มข้อมูลกับพาทของแฟ้มข้อมูลนั้นได้เอง (โดยสัญลักษณ์ที่กำหนดขึ้นมานั้นจะต้องไม่ผิดกฎการตั้งชื่อ ซึ่งระบบจะมีการตรวจสอบด้วย) จะเห็นได้ว่ามีการตั้งค่าของสัญลักษณ์ไว้ว่า _Located_ ดังนั้น เมื่อมีแฟ้มข้อมูลชื่อซ้ำกันนั้นมีชื่อว่า txtfile โดยมาจากเครื่องปลายทางที่ชื่อว่า Desktop ผลที่ได้ คือ ได้แฟ้มข้อมูลที่มีชื่อว่า txtfile_Located_Desktop ดังภาพที่ 4-4

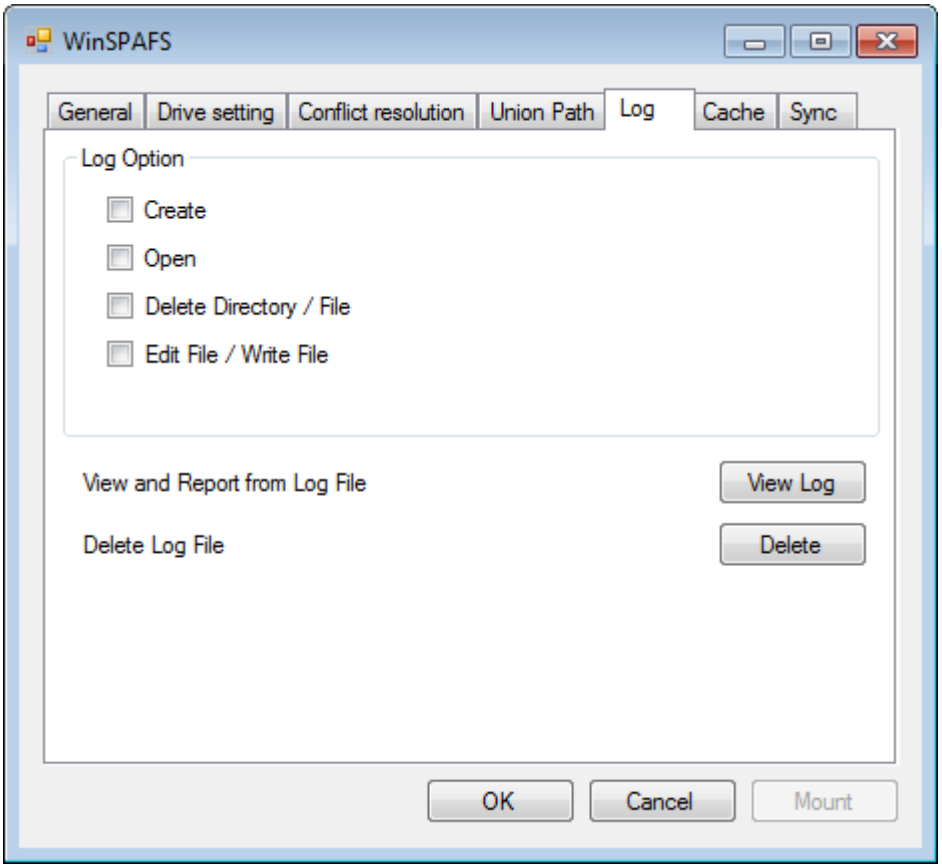


ภาพที่ 4-4 การกำหนดสัญลักษณ์ที่ใช้คั่นระหว่างชื่อแฟ้มข้อมูลกับพาทของแฟ้มข้อมูล

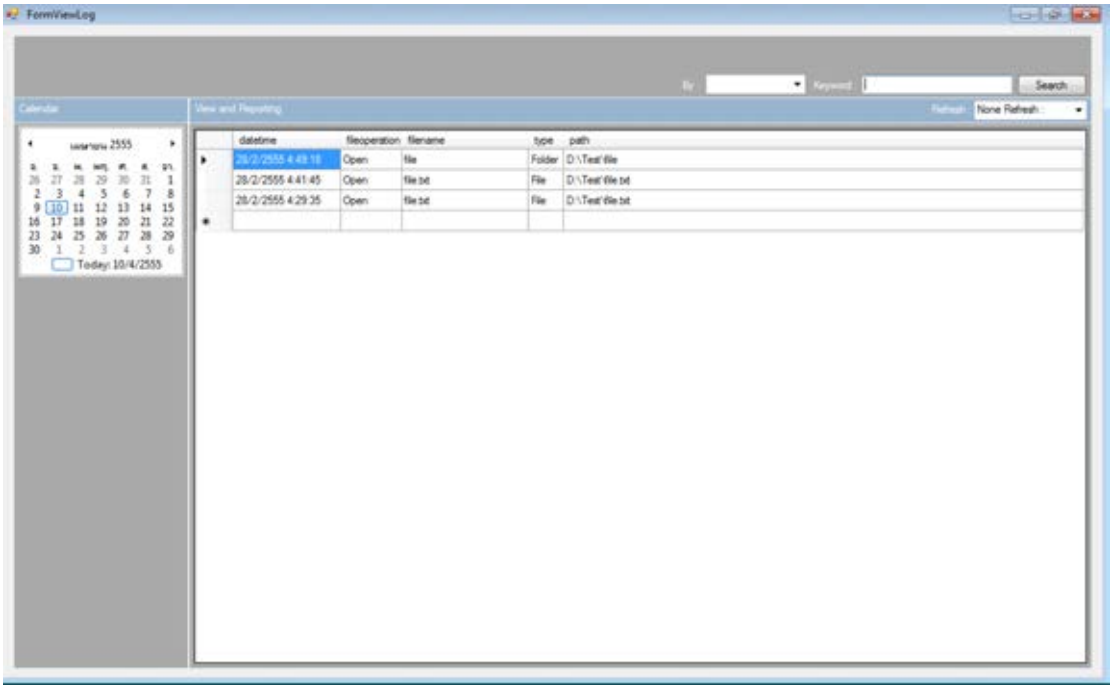
4.3 การเก็บและการดูข้อมูลการใช้งาน

การเก็บข้อมูลการใช้งานสามารถเก็บข้อมูลการอ่าน เปิด แก้ไข ลบแฟ้มข้อมูลที่อยู่ในเมาส์พอยท์ได้ ดังภาพที่ 4-5 ผู้ใช้สามารถเลือกเก็บข้อมูลการใช้งานแฟ้มข้อมูลได้ในกรณีที่ต้องการเก็บการใช้งานบางส่วน แต่ถ้าไม่ต้องการเก็บข้อมูลใดๆ ก็สามารถทำได้เพื่อเป็นการประหยัดพื้นที่ฮาร์ดดิสก์และลดโอเวอร์เฮด นอกจากนี้แล้วยังสามารถลบข้อมูลการใช้งานที่ถูกเก็บไว้ได้อีกด้วย

การดูข้อมูลการใช้งาน ผู้ใช้สามารถดูข้อมูลการใช้งานแฟ้มข้อมูลได้อย่างสะดวก และยังสามารถค้นหาการใช้งานแฟ้มข้อมูลได้อีกด้วย ดังภาพที่ 4-6 โดยการเก็บแฟ้มข้อมูลใดๆ นั้น จะมีการเก็บรายละเอียดของแฟ้มข้อมูล ได้แก่ วันเวลา การกระทำ ชื่อแฟ้มข้อมูล ชนิด พาท



ภาพที่ 4-5 ส่วนของการจัดเก็บการใช้งานแฟ้มข้อมูลในเมาส์พอยท์



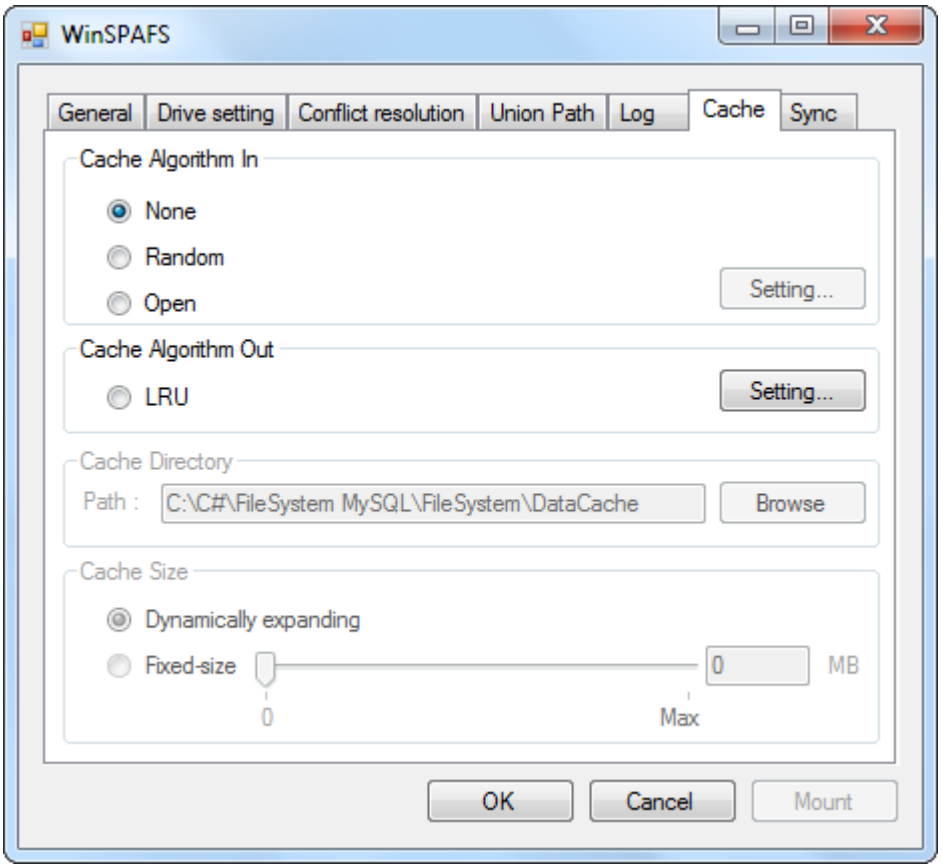
ภาพที่ 4-6 ส่วนของการแสดงข้อมูลการใช้งานแฟ้มข้อมูลในเมาส์พอยท์

ในส่วนของซอร์สโค้ดนั้น จะมีคลาส (Class) ของการเชื่อมต่อ (MySQL_Connection) และการเก็บข้อมูล (DatabaseManager) ลงฐานข้อมูล MySQL โดยในส่วนนี้สามารถที่จะพัฒนาให้ใช้กับฐานข้อมูลอื่นๆ ได้ในอนาคต

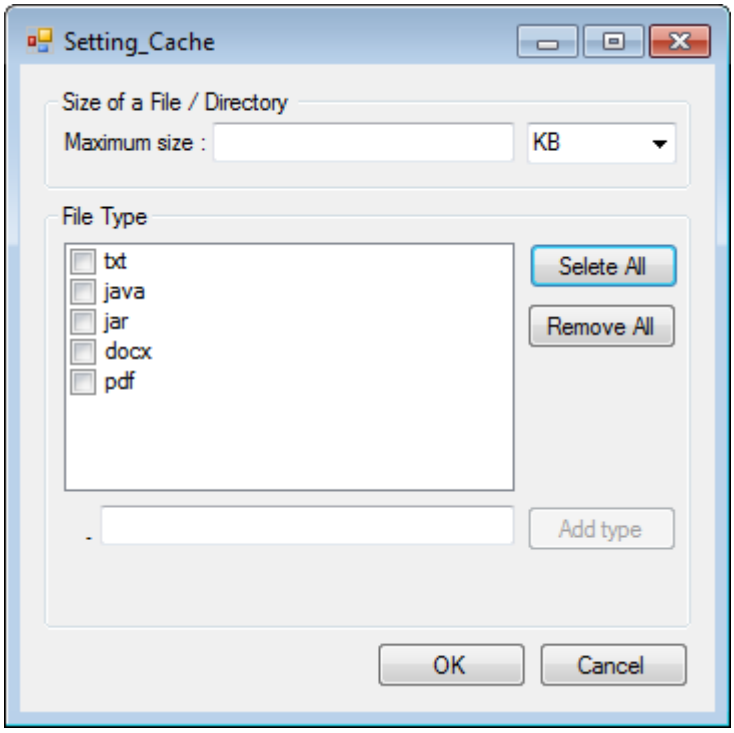
4.4 ระบบแคช

ผู้ใช้สามารถระบุโฟลเดอร์ที่ใช้ในการจัดเก็บได้เอง และยังสามารถกำหนดขนาดพื้นที่ในการจัดเก็บแคชหรือแฟ้มข้อมูล (Data Cache) ได้อีกด้วย เพื่อที่ระบบจะส่งผลกับพื้นที่ฮาร์ดดิสก์ตามที่ผู้ใช้ต้องการเท่านั้น ดังภาพที่ 4-7

โดยแคชหรือแฟ้มข้อมูลที่ถูกเก็บจะต้องผ่านขั้นตอนการคัดเลือกก่อน โดยเมื่อทำการอ่านหรือสุ่มแฟ้มข้อมูลใดๆ จะมีการตรวจสอบชนิดหรือขนาดว่าอยู่ในเงื่อนไขในการเก็บแคชหรือแฟ้มข้อมูลหรือไม่ ดังภาพที่ 4-8 เมื่อแคชหรือแฟ้มข้อมูลที่เข้ามาใหม่ ระบบจะทำการตรวจสอบพื้นที่ว่างที่เหลือใน Data Cache ว่าเหลือพอสำหรับแฟ้มข้อมูลที่เข้ามาใหม่หรือไม่ ถ้าไม่พอระบบจะใช้อัลกอริทึมการแทนที่แอลอาร์ยูในการสับเปลี่ยนแฟ้มข้อมูล แต่ในเบื้องต้นผู้วิจัยได้ทำการพัฒนาให้มีการใช้เก็บแคชหรือแฟ้มข้อมูลเมื่อมีการอ่านแฟ้มข้อมูลนั้น



ภาพที่ 4-7 ส่วนของการจัดเก็บแคชหรือแฟ้มข้อมูลในเมาส์พอยท์



ภาพที่ 4-8 ส่วนของการคัดเลือกแคชหรือแฟ้มข้อมูลในเมาส์พอยท์

4.5 รายการการทำงานของแฟ้มข้อมูล

รายการการทำงานของแฟ้มข้อมูลหรือไฟล์โอเปอร์เรชั่น (File Operation) ของ WinSPAFS จะประกอบด้วยการทำงานพื้นฐานของระบบแฟ้มข้อมูล แสดงดังตารางที่ 4-1 แสดงรายการของการทำงานของแฟ้มข้อมูลของ WinSPAFS

ตารางที่ 4-1 รายการ การทำงานของแฟ้มข้อมูลของ WinSPAFS

รายการของการทำงานของแฟ้มข้อมูล (File Operation)	
CreateFile	DeleteFile
WriteFile	DeleteDirectory
CreateDirectory	MoveFile
OpenDirectory	MoveDirectory
ReadFile	GetFileInformation
EditFile	GetDirectoryInformation
RenameFile	CopyDirectory
RenameDirectory	CopyFile

4.5.1 ลักษณะการทำงานในกรณีของ name conflict (Append)

```
//f.fullName คือ path
 //ตัวอย่าง C:\path1\Example
 //v = ตำแหน่งของ \ อันสุดท้าย ของ C:\path1\Example
 int v = f.FullName.LastIndexOf("\\");
 //fto คือ path ที่ไม่มีชื่อ folder = C:\path1
 string fto = f.FullName.Substring(0, v);
 //v = ตำแหน่งของ \ อันสุดท้าย ของ C:\path1
 int vv = fto.LastIndexOf("\\");
 //fto2 คือ ชื่อ path = path1
 string fto2 = fto.Substring(vv + 1);
 //check ว่าหา @ เจอหรือไม่
 if (f.Name.IndexOf("@") == -1)
 {
 //Copy C:\path1\Example to C:\path1\path1@Example
 Directory.Move(f.FullName, fto + "\\" + fto2 + "@" + f.Name);
 //fi.FileName เป็นชื่อที่จะใช้แสดงใน windows
 //fi.FileName = path1@Example
 fi.FileName = fto2 + "@" + f.Name;
 }
else
{
 //ถ้ามี @ แล้ว ไม่ต้องทำการเพิ่ม @
fi.FileName = f.Name;
 }
 //files เป็นตัวที่ใช้สำหรับแสดงใน windows
 files.Add(fi);
 filen[i, p[i]] = f.Name;
 p[i]++;
```

4.5.2 ลักษณะการทำงานในกรณีของ OpenDirectory

```
1. begin
2.      int OpenDirectory(String filename, DokanFileInfo info)
3.      info.Context _openFilesNextKey++
4.              for (int i 0; i < roots.Count; i++)
5.                      if (Directory.Exists(GetPathAt(filename, i)))
6.                      return 0;
7.                      end if
8.              end for
9.      return -DokanNet.ERROR_PATH_NOT_FOUND
10. end
```

4.5.3 ลักษณะการทำงานในกรณีของ readFile

```
1. begin
2.      ReadFile(String filename, Byte[] buffer, ref uint readBytes, long offset,
DokanFileInfo info)
3.              try
4.                      ContextType contextKey = (ContextType)info.Context;
5.                      Stream file = _openFiles[contextKey].fileStream;
6.                      file.Seek(offset, SeekOrigin.Begin);
7.                      readBytes = (uint)file.Read(buffer, 0, buffer.Length);
8.              catch { return -1; }
9.      return 0;
10. end
```

4.5.4 ลักษณะการทำงานในกรณีของ Cache

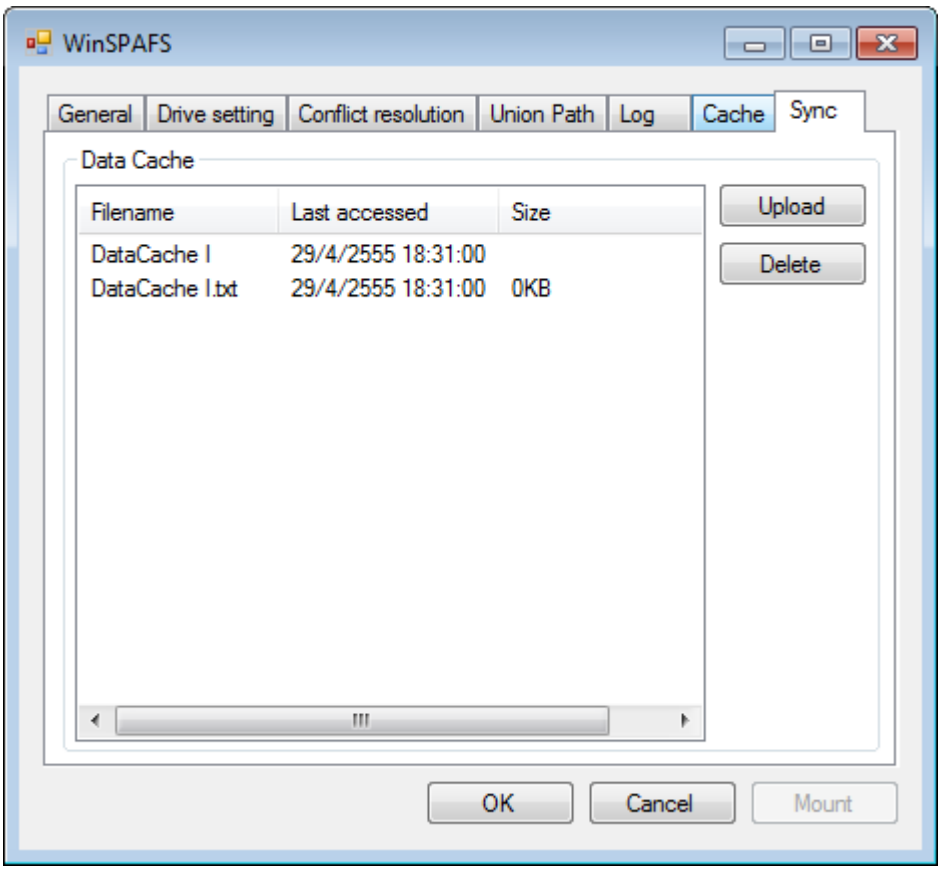
1. ถ้าเลือกใช้แคช ระบบแคชเปิดการทำงาน
2. รับข้อมูลเมื่อมีการอ่านแฟ้มข้อมูลหรือโฟลเดอร์ให้ทำการรับเข้าสู่เงื่อนไข
    1. ถ้าเป็นแฟ้มข้อมูลให้ตรวจสอบนามสกุล
3. ทำการตรวจสอบขนาดของแฟ้มข้อมูลหรือโฟลเดอร์
4. ถ้าผ่านเงื่อนไขให้จัดเก็บลงแคช

4.5.5 ลักษณะการทำงานในกรณีของ Disconnect

1. ระบบขาดการเชื่อมต่อ
2. ตรวจสอบว่าแฟ้มข้อมูลที่ใช้งานอยู่ มาจากเครื่องใด
    1. ถ้ามาจากเครื่องปลายทาง
        1. บันทึกลง Data Cache แทน
    2. ถ้ามาจากเครื่องผู้ใช้
        1. ทำการอ่าน แก้ไข และบันทึกบนเครื่องผู้ใช้เองตามปกติ

4.6 การเก็บและการดูข้อมูลการใช้งาน

การตรวจสอบสถานะและอัพเดทสถานะของแฟ้มข้อมูลจะใช้ในกรณีที่ระบบมีการเชื่อมต่อใหม่หลังจากที่ระบบขาดการเชื่อมต่อ สามารถตรวจสอบแฟ้มข้อมูลที่อยู่ใน Data Cache สามารถอัพเดทแฟ้มข้อมูลไปยังเครื่องปลายทาง และยังสามารถลบแฟ้มข้อมูลได้อีกด้วย



ภาพที่ 4-9 ส่วนของการตรวจสอบและอัพเดทแฟ้มข้อมูล

บทที่ 5

การทดสอบประสิทธิภาพ

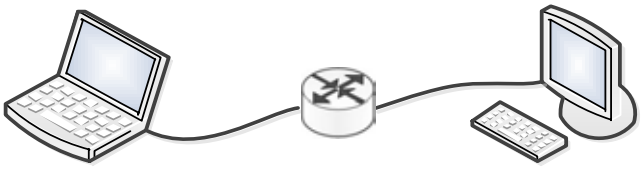
ในบทนี้จะกล่าวถึงการทดสอบประสิทธิภาพของระบบรวมแฟ้มข้อมูลส่วนบุคคลที่ได้พัฒนาขึ้น โดยเริ่มอธิบายจากอุปกรณ์ที่ใช้ในการทดสอบ การทดสอบ ผลการทดสอบ และสรุปผลการทดสอบ

5.1 อุปกรณ์ที่ใช้ในการทดสอบ

ผู้วิจัยได้ทำการได้ทำการติดตั้งระบบรวมแฟ้มข้อมูลส่วนบุคคลลงบนเครื่องคอมพิวเตอร์จำนวน 2 เครื่อง ด้วยอุปกรณ์ดังตารางที่ 5-1 ในลักษณะของ Local Area Network ดังภาพที่ 5-1 ทำการเชื่อมต่อผ่านทางระบบเครือข่ายด้วยความเร็ว 1 กิกะบิตต่อวินาที (Gbps)

ตารางที่ 5-1 อุปกรณ์ที่ใช้ในการทดสอบ

Client	
Processor	Intel U7300 1.3GHz
Memory	4096 MB
HDD	Seagate ST9500325AS
NIC	1GbE
OS	Windows 7 Professional 64 bit
Data Node	
Processor	AMD Phenom ii X2 555 3.2GHz
Memory	6144 MB
HDD	Western Digital WD5001AALS
NIC	1GbE
OS	Windows 7 Ultimate 64 bit
Switch/Router	
Model	Billion BiPAC 5200G R4 (10/100)



ภาพที่ 5-1 รูปแบบการเชื่อมต่อที่ใช้ในการทดสอบ

5.2 การทดสอบ

ผู้วิจัยได้ทำการทดสอบประสิทธิภาพ การอ่าน (read) การอ่านซ้ำ (re-read) การเขียน (write) การเขียนซ้ำ (re-write) แฟ้มข้อมูลด้วยโปรแกรมไอโอโซน (IOzone File Benchmark) [14] โดยจะทำการทดสอบใน 5 ลักษณะดังนี้

กรณีที่ 1) ทดสอบระบบแฟ้ม NTFS (การใช้งานแฟ้มข้อมูลทั่วไปบนระบบปฏิบัติการวินโดว์) บน Local Disk

กรณีที่ 2) ทดสอบระบบแฟ้ม NTFS ผ่านทางระบบเครือข่าย

กรณีที่ 3) ทดสอบระบบแฟ้ม WinSPAFS บน Local Disk

กรณีที่ 4) ทดสอบระบบแฟ้ม WinSPAFS ผ่านทางระบบเครือข่าย

กรณีที่ 5) ทดสอบระบบแฟ้ม WinSPAFS ผ่านทางระบบเครือข่ายและมีการใช้งานฐานข้อมูล

ซึ่งการทดสอบแฟ้ม WinSPAFS บน Local Disk (กรณีที่ 3) จะเทียบได้กับการใช้งานแคชหรือแฟ้มข้อมูลที่ถูกเก็บไว้บนฮาร์ดดิสก์ของเครื่องผู้ใช้เอง แต่ระบบแคชจะทำงานได้นั้นต้องมีการทำงานผ่านทางระบบเครือข่ายและมีการใช้งานฐานข้อมูล (กรณีที่ 5) ดังนั้นผลการทดสอบที่ได้ของการทำงานของระบบแฟ้ม WinSPAFS ร่วมกับแคชผ่านทางระบบเครือข่ายที่มีการใช้งานฐานข้อมูลจะมีค่าอยู่ระหว่าง (กรณีที่ 3) และ (กรณีที่ 5) โดยจะมีค่ามากหรือน้อยขึ้นอยู่กับว่า ผู้ใช้มีการใช้งานแฟ้มข้อมูลใน Data Cache และแฟ้มข้อมูลที่มาจากเครื่องปลายทางมากน้อยแค่ไหน

ผู้วิจัยได้ทำการทดสอบกับแฟ้มข้อมูลสูงสุดขนาด 1 เมกะไบต์ (maximum file size) ด้วยโปรแกรมไอโอโซน โดยใช้คำสั่งผ่านทาง Command Line ดังนี้

```
iozone -Rab output.wks -f[path][filename] -g 1M -i0 -i1
```

โดยได้ทำการทดสอบเป็นจำนวน 10 ครั้งและนำผลที่ได้มาหาค่าเฉลี่ย

5.3 ผลการทดสอบ

ผลการทดสอบด้วยโปรแกรมไอโอโซน ใช้ค่า file size ขนาด 1 เมกะไบต์และ Record size ขนาด 1 เมกะไบต์ ดังภาพที่ 5-2 มาหาค่าเฉลี่ย โดยแสดงดังตารางที่ 5-2 และภาพที่ 5-3 ตามลำดับ

iozone -Rab E:\WinSPAFS+Database.wks -f i:\file -g 1M -i 0 -i 1

The top row is records sizes, the left column is file sizes

Writer Report

	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	5191	10737	20454	36413	66624								
128	4928	10058	20952	36860	67170	124422							
256	4851	9811	3473	35858	70223	128491	206138						
512	2984	7697	18712	30425	71450	127225	119205	247977					
1024	4785	9313	13678	32173	70686	126237	218092	216472	331406				

Re-writer Report

	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	5297	10569	19083	35485	116247								
128	1360	10524	20580	35786	66313	184507							
256	4922	9532	18227	34957	65700	125061	312722						
512	4817	7733	19470	35650	72307	73819	221620	308983					
1024	3648	9493	18853	36263	66667	100827	200504	168204	358090				

Reader Report

	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	24133	35580	39525	70736	350079								
128	29813	42322	45765	69299	93775	338410							
256	32170	49772	23638	50527	75149	121325	434655						
512	39890	62556	53844	63324	74362	94367	131695	418036					
1024	43853	63273	62789	58351	79864	89421	91071	132575	250341				

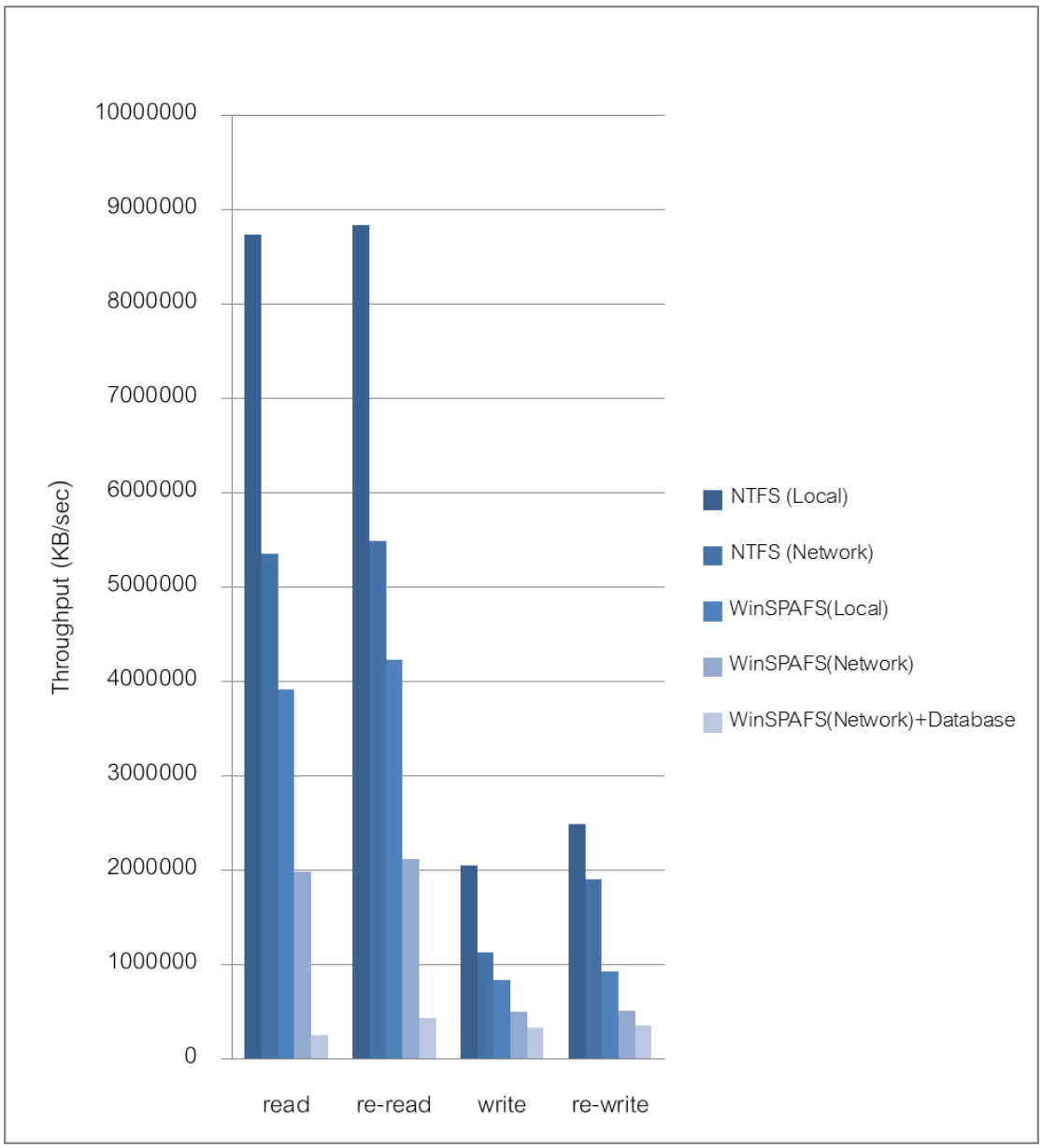
Re-reader Report

	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
64	23882	37039	41389	73163	319638								
128	32724	41832	52994	32403	77050	363820							
256	35626	49423	53433	63141	85151	112550	418446						
512	40618	55275	60304	56288	81377	90405	132896	427466					
1024	41443	65991	65121	62403	74928	89857	109192	120481	438735				

ภาพที่ 5-2 ตัวอย่างผลทดสอบที่ได้จากการทดสอบประสิทธิภาพด้วยโปรแกรมไอโอโซน

ตารางที่ 5-2 ผลการทดสอบประสิทธิภาพ

File System / Throughput (KB/sec)	NTFS (Local)	NTFS (Network)	WinSPAFS (Local)	WinSPAFS (Network)	WinSPAFS (Network) +Database
Read	8733074	5355052	3914963	1984035	250341
Re-Read	8831076	5484194	4230751	2119560	438735
Write	2048756	1134032	840303	497111	331406
Re-Write	2493552	1909949	932143	514196	358090



ภาพที่ 5-3 เปรียบเทียบประสิทธิภาพการทำงานของระบบแฟ้มข้อมูล

5.4 วิเคราะห์ผลการทดสอบ

จากผลการทดลองพบว่าความเร็วในการอ่านการเขียนแฟ้มข้อมูลของระบบแฟ้มข้อมูล WinSPAFS (Network)+Database มีค่าน้อยกว่า WinSPAFS (Local) พอสมควรเนื่องจากโอเวอร์เฮดที่เกิดจากการใช้งานผ่านระบบเครือข่ายร่วมกับฐานข้อมูล จึงทำให้ความเร็วลดลง แต่เมื่อมีการนำระบบแคชมาใช้งานร่วมด้วย (WinSPAFS (Network)+Database+Cache) ทำให้ความเร็วเพิ่มขึ้น เนื่องจากการอ่านการเขียนแฟ้มข้อมูลบางส่วนจะเรียกใช้งานจากแคชที่อยู่ใน Data Cache (Local) ได้โดยตรง แต่ก็ยังมีความเร็วน้อยกว่าการใช้งาน NTFS (Network) ที่ทำงานผ่าน

ระบบเครือข่าย และ NTFS (Local) ที่ทำงานบนเครื่องผู้ใช้เอง โดยนอกจากโอเวอร์เฮดที่เกิดจากการใช้งานฐานข้อมูลแล้ว ยังมีโอเวอร์เฮดที่เกิดจากขั้นตอนการทำงานภายในหลายขั้นตอน และการทำงานในส่วนของยูเซอร์สเปซอีกด้วย

ถึงแม้ว่าความเร็วในการอ่านการเขียนข้อมูลมีค่าไม่สูง แต่ระบบแฟ้มของงานวิจัยนี้สามารถรวบรวมแฟ้มข้อมูลจากหลายแหล่งข้อมูล (พร้อมการแก้ปัญหาชื่อซ้ำซ้อน) มาแสดงเสมือนเป็นแฟ้มข้อมูลในเครื่องผู้ใช้เอง และระบบยังสนับสนุนการทำงานแบบนอกสายโดยการสนับสนุนของแคชแบบแอลอาร์ยู ซึ่งความสามารถที่ได้กล่าวมาเมื่อเทียบกับความเร็วในการอ่านการเขียนข้อมูล ถือว่าอยู่ในระดับที่ยอมรับได้

บทที่ 6
บทสรุป

6.1 บทสรุป

ในงานวิจัยนี้ได้เสนอระบบรวมแฟ้มข้อมูลส่วนบุคคลที่ทำงานอยู่บนระบบปฏิบัติการวินโดว์ โดยใช้หลักการของระบบแฟ้มแบบกระจาย ระบบแฟ้มแบบร่วม อัลกอริทึมการแทนที่แบบแอลอาร์ยู และโดคานมาประยุกต์ใช้ร่วมกัน ทำการพัฒนาในส่วนของยูเซอร์สเปซโดยใช้โดคานดอทเน็ตไบน์ดิง ซึ่งระบบจะมีลักษณะคล้ายกับ SPAFS [7] แต่จะทำงานบนระบบปฏิบัติการวินโดว์ โดยมีความสามารถในการรวบรวมแฟ้มจากหลายแหล่งข้อมูล มาแสดงเสมือนเป็นแฟ้มข้อมูลในเครื่องผู้ใช้เอง พร้อมกับแก้ปัญหาชื่อซ้ำซ้อนด้วยวิธีของระบบแฟ้มยูนิฟิเคชั่นและยังเลือกใช้วิธีการระบุชื่อแฟ้มข้อมูลต่อท้ายแฟ้มข้อมูลได้ มีการเก็บแคชข้อมูลเพื่อเพิ่มประสิทธิภาพในการทำงานและสามารถทำงานแบบนอกสายได้ มีอัลกอริทึมที่ใช้ในการจัดเก็บแคชและจัดการแฟ้มข้อมูลที่เก็บไว้ใน Data Cache อย่างอัตโนมัติ โดยมีการสับเปลี่ยนแฟ้มข้อมูลตามเงื่อนไขเพื่อให้ตรงตามความต้องการของผู้ใช้งานมากที่สุด มีการเก็บข้อมูลการใช้งานต่างๆ เกี่ยวกับแฟ้มข้อมูลอีกด้วย

โดย WinSPAFS เป็นระบบต้นแบบของระบบรวมแฟ้มข้อมูลส่วนบุคคลที่ทำงานอยู่บนระบบปฏิบัติการวินโดว์ ซึ่งแสดงให้เห็นว่า WinSPAFS มีประสิทธิภาพอยู่ในระดับที่ยอมรับได้ เมื่อเทียบกับคุณสมบัติและความสามารถในการทำงาน นอกจากนี้แล้วยังสามารถนำไปพัฒนาต่อได้ง่ายในอนาคต

6.2 ข้อจำกัด

ระบบรวมแฟ้มข้อมูลส่วนบุคคลที่งานวิจัยนี้นำเสนอ เป็นเพียงระบบต้นแบบ นำไปติดตั้งบนระบบปฏิบัติการวินโดว์เท่านั้น

6.3 แนวทางการวิจัยในอนาคต

วิจัยเพื่อหาแนวทางที่จะพัฒนาระบบให้มีอัตราการถ่ายโอนข้อมูลที่ดีขึ้น ลดโอเวอร์เฮดที่เกิดขึ้นจากการใช้งานฐานข้อมูล ทดสอบกับฐานข้อมูลที่มีขนาดเล็กและใช้ทรัพยากรระบบให้น้อยลง สร้างระบบให้มีความยืดหยุ่น สามารถปรับแต่งได้มากขึ้น สามารถใช้ร่วมกับแพลตฟอร์มอื่นๆ ที่ไม่ใช้ระบบปฏิบัติการวินโดว์ได้

รายการอ้างอิง

[1] Linigtion P.F. File Transfer Protocol. IEEE journal on selected areas in Communications. (7 September 1989) : 1052 – 1059.

[2] Houston D. and A. Ferdowsi. Drop Box. [Online]. Available: http://www.dropbox.com. [2012, April 12]

[3] Burstein P. Efficient search in file-sharing network. ICPADS '07 Proceedings of the 13th International Conference on Parallel and Distributed Systems. (7 December 2007): 1 – 9.

[4] Sun Microsystem. NFS : Network File System Protocol Specification. [Online]. Availiable : http://www.ietf.org/rfc/rfc1094.txt [2012, May 1]

[5] Howard J.H., Kazar M.L., Nichols, S.G., Nichols D.A., Satyanarayanan M., Sidebotham R.N. and West M.J. Scale and Performance in a Distributed File System. ACM Transactions on Computer Systems. (February 1988) : 51–81.

[6] Kistler J. and Satyanarayanan M. Disconnected Operation in the Coda File System. ACM Transactions on Computer Systems, (February 1992) : 3-25.

[7] Dhumbumroong S. and Piromsopa K. Personal Cloud Filesystem: A distributed unification filesystem for personal computer and portable device. 2011 Eighth International Joint Conference on Computer Science and Software Engineering (JCSSE). (May 2011) : 58-62

[8] Levy D.E. and Silberschatz A. Distributed file systems : concepts and examples. Journal ACM Computing Surveys (1990). (Dec 1990) : 321-374

[9] Pendry J.S. and McKusick M.K. Union mounts in 4.4bsd-lite. Proceedings of the USENIX 1995 Technical Conference Proceedings. TCON'95. Berkeley. CA. USA : USENIX Association. 1995. [Online]. Available from: http://portal.acm.org/citation.cfm?id= 1267411.1267414. [2012,May 1]

[10] Gelenbe E. A Unified Approach to the Evaluation of a Class of Replacement Algorithms. IEEE Transactions on computers. (June 1973) : 611 – 618.

[11] Asakawa, H. Dokan. [Online]. 2008. Available from: http://dokan-dev.net/en. [2012,May 1]

[12] Szeredi M. FUSE : Filesystem in Userspace. [Online]. Available from : http://fuse.sourceforge.net. [2012,May 1]

[13] Voras I. and Zagar M. Network distributed file system in user space. ITI 2006 Proceedings of the 28th International Conference on Information Technology Interface. (2006) : 669 – 674.

[14] Norcott W.D. IOzone filesystem benchmark tool. [Online]. [2008] Available from : http://www.iozone.org. [2012,May 13]

ประวัติผู้เขียนวิทยานิพนธ์

นายวราวุฒิ บัวเทศ เกิดเมื่อวันที่ 20 พฤศจิกายน พ.ศ. 2528 ที่จังหวัดนครสวรรค์ สำเร็จการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาคณิตศาสตร์ จากภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร ในปีการศึกษา 2550 และเข้าศึกษาในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2552

Test

ORIGINALITY REPORT

8 %	6 %	2 %	6 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Chulalongkorn University *Student Paper*	5%
2	mobility.eecs.umich.edu *Internet Source*	< 1%
3	Submitted to King Mongkut's University of Technology Thonburi *Student Paper*	< 1%
4	www.ms.kku.ac.th *Internet Source*	< 1%
5	dataverse.dvn.utcc.ac.th *Internet Source*	< 1%
6	bmegias.wordpress.com *Internet Source*	< 1%
7	Submitted to Auckland University of Technology *Student Paper*	< 1%
8	www.corpus4u.org *Internet Source*	< 1%
9	John Eberhard. "Efficient Object Caching for Distributed Java RMI Applicatio... *Publication*	< 1%
10	Jian Liang. "Minimizing Metadata Access Latency in Wide Area Networked ... *Publication*	< 1%
11	storageconference.org *Internet Source*	< 1%
12	Submitted to Asia Pacific University College of Technology and Innovation (... *Student Paper*	< 1%
13	Pierre K.Y. Lai. "The Rules of Time on NTFS File System", Second Interna... *Publication*	< 1%
14	bib.irb.hr *Internet Source*	< 1%
15	"Table of contents", 28th International Conference on Information Technolog... *Publication*	< 1%
16	users.iit.demokritos.gr *Internet Source*	< 1%

17	www.arstudy.com *Internet Source*	< 1%
18	www.phd.mbu.ac.th *Internet Source*	< 1%
19	thaigamestudio.com *Internet Source*	< 1%
20	www.krirk.ac.th *Internet Source*	< 1%
21	ling.arts.chula.ac.th *Internet Source*	< 1%
22	webshop.powertech.se *Internet Source*	< 1%
23	www.cpc.ku.ac.th *Internet Source*	< 1%
24	www.riverpub.com *Internet Source*	< 1%
25	www.charm.au.edu *Internet Source*	< 1%
26	wanglum.org *Internet Source*	< 1%

EXCLUDE QUOTES OFF
EXCLUDE BIBLIOGRAPHY OFF
EXCLUDE MATCHES OFF