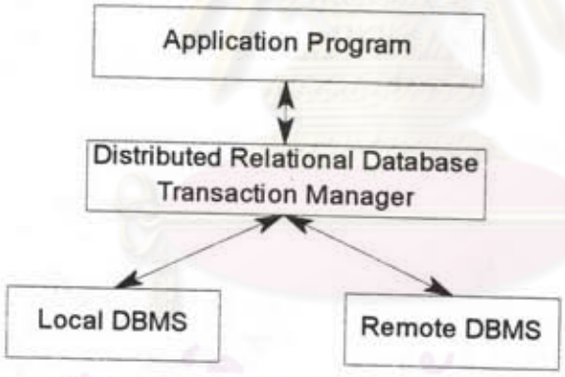
บทที่ 7

สรุปผลการวิจัยและข้อเสนอแนะ

สรุปผลการวิจัย

1. ลักษณะโดยรวมของ DRDBTM

การทำงานของชุดโปรแกรมที่พัฒนาขึ้นมานี้จะทำงานอยู่ในระหว่างโปรแกรมประยุกต์กับระบบจัดการฐานข้อมูลดังรูปที่7.1 ซึ่งจะรับหน้าที่ในการติดต่อกับฐานข้อมูลในระบบต่าง ๆ ตามกฎเกณฑ์ที่ระบุไว้ใน Control Table ทั้งหมด โดยที่ตารางควบคุมจะมีทั้งหมดดังนี้



รูปที่ 7.1 ระดับการให้บริการของ DRDBTM

1.1 ตารางควบคุมเครือข่าย (Network Directory Table : NDT) จะเก็บรายชื่อของระบบต่าง ๆ ในเครือข่ายที่เก็บฐานข้อมูลไว้ พร้อมทั้งกำหนดลำดับในการเข้าถึงของการเรียกใช้

1.2 ตารางควบคุมการแตกกระจาย (Fragment Control Table : FCT) ใช้สำหรับเก็บรายชื่อของแฟ้มฐานข้อมูลที่ต้องการใช้งานแบบฐานข้อมูลแบบกระจาย พร้อมทั้งต้องระบุให้ชัดเจนว่าจะถูกจัดเก็บด้วยวิธีหนึ่งวิธีใดจากสามวิธีนี้ เก็บซ้ำ เก็บแยกส่วน หรือเก็บรวมศูนย์

1.3 ตารางควบคุมสถานที่ตั้ง (Database Mapping Table : DBM) ใช้สำหรับบ่งชี้ว่าแฟ้มฐานข้อมูลจะถูกเก็บอยู่ที่ระบบใดบ้าง

1.4 ตารางควบคุมการแบ่งแยกข้อมูล (Partition Control Table : PCT) ใช้เป็นเงื่อนไขสำหรับการชี้ชัดว่ารายการข้อมูลแต่ละรายการจะถูกเก็บไว้ที่ระบบใด

2. การใช้ DRDBTM สำหรับพัฒนาระบบธุรกิจ

ในการทำวิจัยนี้ใช้เวลามากกว่า 50% สำหรับใช้ในการพัฒนาโปรแกรมผู้จัดการฐานข้อมูลแบบกระจาย (Distributed Relational Database Transaction Manager) ซึ่งจะเป็นโปรแกรมหลักที่จะให้บริการกับโปรแกรมประยุกต์ที่พัฒนาขึ้นด้วยภาษา RPG/400 โดยที่โปรแกรมประยุกต์ที่จะมาใช้บริการจะทำการเรียกใช้ได้โดยใช้คำสั่ง /COPY ซึ่งเป็น Compiler Directive ที่ใช้สำหรับโปรแกรมภาษา RPG/400 ที่จะทำการสำเนาส่วนประกอบที่อ้างถึงมาแทรกในโปรแกรมในการงานของ Compiler ขั้นตอนแรกก่อนที่จะทำการแปลให้เป็นภาษาเครื่องต่อไป ซึ่งทำให้ผู้พัฒนาโปรแกรมไม่จำเป็นต้องเขียนโปรแกรมเพิ่มเพื่อให้สามารถควบคุมการทำงานฐานข้อมูลให้เป็นแบบกระจาย ดังนั้นการพัฒนาระบบธุรกิจต่าง ๆ ที่ต้องการความเป็นฐานข้อมูลแบบกระจายจะใช้เวลาสั้นลง

3. ประสิทธิภาพของ DRDBTM

โปรแกรมผู้จัดการฐานข้อมูลแบบกระจายนี้จะให้บริการสำหรับการเรียกใช้ฐานข้อมูลขั้นพื้นฐานต่างๆคือ เพิ่มรายการข้อมูลใหม่ แก้ไขรายการข้อมูลเก่า ลบรายการข้อมูลเก่า ค้นหาข้อมูลด้วยดัชนี และอ่านรายการข้อมูลอย่างเรียงลำดับด้วยดัชนี ซึ่งเวลาที่ตอบสนอง (Response time) อยู่ระหว่าง 3-5 วินาทีสำหรับการทำงานของฟังก์ชั่นพื้นฐานในแบบโต้ตอบ ซึ่งถือว่าเป็นที่ยอมรับได้ ยกเว้นการอ่านรายการข้อมูลอย่างเรียงลำดับด้วยดัชนี(Index Sequencial) ของการเก็บแบบแยกส่วนจะให้เวลาตอบสนองที่ช้ามาก สาเหตุเนื่องมาจากการอ่านรายการข้อมูลอย่างเรียงลำดับด้วยดัชนีของการเก็บแบบแยกส่วนนี้ต้องทำการเลือกรายการข้อมูลที่สอดคล้องกับดัชนีจากระบบต่าง ๆ เอง(Dynamic Sorting)

4. ข้อจำกัดของการใช้งาน DRDBTM

4.1 ใช้ได้กับโปรแกรมภาษา RPG/400 เท่านั้น

4.2 ห้ามใช้คำสั่ง OVRDBF เพราะจะทำให้ Database Manager ทำงานผิดพลาด

4.3 การใช้ Utility ของระบบฐานข้อมูลจะได้ผลสมบูรณ์เฉพาะการอ่านข้อมูลกรณีเป็นแฟ้มที่จัดเก็บแบบซ้ำเท่านั้น เพราะตัว Utility จะมองเห็นเฉพาะฐานข้อมูลท้องถิ่นเท่านั้น

4.4 ใช้ได้กับระบบเครือข่ายที่เป็น APPN เท่านั้น

5. ข้อควรระวังสำหรับการ ROLLBACK

การทำ ROLLBACK ของชุดโปรแกรมนี้ จะทำการคืนสภาพได้สมบูรณ์เฉพาะกรณีที่ในขณะทำการคืนสภาพจะต้องไม่พบภาวะล้มเหลวอีกต่อไป แต่ถ้าเกิดความล้มเหลวขึ้น โปรแกรมจะทำการคืนสภาพให้กับระบบที่สามารถจะติดต่อได้เท่านั้น ส่วนระบบที่ไม่สามารถติดต่อได้ผู้ดูแลฐานข้อมูลจะต้องเป็นผู้ทำการฟื้นสภาพให้เอง(Manual)

ข้อเสนอแนะ

1. ประสิทธิภาพของระบบ (Performance)

เนื่องจากการทำงานในระบบเครือข่าย จำเป็นต้องคำนึงถึงความเร็วในการตอบสนองของระบบต่อผู้ใช้ ดังนั้นเพื่อที่จะทำให้ระบบตอบสนองได้ดี จึงต้องพิจารณาในสิ่งต่าง ๆ ต่อไปนี้

1.1 ข้อควรพิจารณาเกี่ยวกับการสื่อสาร

1.1.1 ในกรณีที่ระบบเครือข่ายใช้เป็นแบบ APPN และการเรียกใช้ข้อมูลในระบบปลายทางจะต้องผ่านระบบอื่นๆในเครือข่ายก่อนนั้น จะทำให้เกิดการสูญเสียเวลามาก ดังนั้นแฟ้มฐานข้อมูลจึงไม่ควรกำหนดค่าพารามิเตอร์ WAITFILE ให้เป็น *IMMED แต่ควรกำหนดให้เป็นค่าหนึ่ง ที่จะทำให้มีเวลาพอที่จะรอให้ระบบต้นทางติดต่อมาถึง โดยต้องพิจารณาถึงขนาดและความซับซ้อนของระบบเครือข่ายประกอบกัน

1.1.2 ขนาดของชุดข้อมูลที่ส่งในแต่ละครั้ง ซึ่งจะอยู่ในพารามิเตอร์ Maxframe ของส่วนควบคุมการสื่อสาร (Controller description) โดยที่ขนาดของชุดข้อมูลที่ส่งนี้จะสัมพันธ์กับลักษณะงานที่เราต้องการให้มีความสำคัญต่างกัน ระหว่างงานแบบตอบโต้กับระบบ(Interactive) กับงานแบบประมวลผลข้อมูลคราวละมาก ๆ (Batch) ถ้าต้องการให้งานแบบตอบโต้มีการตอบสนองที่ดีก็ควรจะให้ขนาดของข้อมูลที่จะส่งมีขนาดเล็ก แต่ถ้าต้องการให้งานประมูลผลแบบยาวนานเสร็จสิ้นเร็วก็ควรให้ขนาดของข้อมูลที่ส่งมีขนาดใหญ่

1.1.3 ขนาดของบัพเฟอร์ที่ใช้ในการเก็บข้อมูลก่อนที่จะส่งจริง ซึ่งจะถูกกำหนดไว้ในพารามิเตอร์ Maxlenru ของโหมดที่ใช้สำหรับส่งข้อมูล(Mode description) โดยจะใช้หลักในการพิจารณาคล้ายกับขนาดของชุดข้อมูลที่จะส่ง คือถ้าต้องการให้การทำงานแบบโต้ตอบกับระบบมีการตอบสนองที่ดีควรจะให้บัพเฟอร์มีขนาดเล็ก แต่ถ้าต้องการให้การประมวลผลแบบยาวนานเสร็จสิ้นเร็วก็ควรจะต้องให้มีขนาดใหญ่

1.1.4 ปริมาณข้อมูลที่ต้องการจากระบบปลายทาง ควรจะให้มีปริมาณน้อยที่สุดเท่าที่จะน้อยได้ ทั้งนี้ขึ้นอยู่กับการออกแบบโปรแกรมประยุกต์ และการออกแบบฐานข้อมูล โดยต้องคำนึงถึงการให้มีการส่งข้อมูลผ่านระบบสื่อสารให้น้อยที่สุดนั่นเอง

1.1.5 ความเร็วของสื่อกลางที่ใช้ในการสื่อสารระหว่างระบบ ซึ่งจะมีผลโดยตรงกับประสิทธิภาพของระบบ กล่าวคือต้องพยายามใช้สื่อกลางที่สามารถให้ความเร็วในการส่งข้อมูลให้มากที่สุด

1.2 ข้อควรพิจารณาเกี่ยวกับการประมวลผลแบบยาวนาน (Batch)

1.2.1 ในกรณีที่โปรแกรมประยุกต์ จะทำการอ่านหรือเพิ่มข้อมูล แบบเรียงลำดับ (Sequential) ให้ใช้คำสั่ง read หรือ read next โดยระบบจะใช้หลักการของการกำหนด ขนาดของชุดข้อมูลที่เหมาะสม (Blocking technic) เพื่อให้เกิดประสิทธิภาพสูงสุด ซึ่งสามารถทำได้โดยการใช้คำสั่ง OVRDBF SEQONLY(*YES) แต่ควรแน่ใจด้วยว่าไม่มีการใช้คำสั่งที่เรียกใช้ข้อมูลแบบสุ่ม

1.2.2 ในกรณีที่ต้อการส่งแฟ้มข้อมูลโดยไม่ต้องผ่านการประมวลผลใด ๆ ควรจะใช้คำสั่งคัดลอกข้อมูล CPYF ผ่าน DDM หรือใช้โปรแกรมประยุกต์ที่กำหนดให้อ่านข้อมูลแบบเรียงลำดับ SEQONLY(*YES) ซึ่งจะเร็วกว่าการใช้โปรแกรมประยุกต์ที่ใช้ ICF เพราะว่าเราไม่เสียเวลาสำหรับการโต้ตอบ(Overhead) ระหว่างปลายทางกับต้นทางมากนัก

1.2.3 ในการส่งวัตถุอื่นๆที่ไม่ใช่แฟ้มข้อมูล เราสามารถใช้คำสั่ง savobj หรือ savlib โดยกำหนดพารามิเตอร์ DTACPR(*YES) เพื่อการอัดแน่นของอักขระ(data compress) ที่จะส่ง เพื่อลดปริมาณข้อมูลที่ซ้ำกันลง

1.3 ข้อควรพิจารณาเกี่ยวกับการประมวลผลแบบโต้ตอบ (Interactive)

เนื่องจากต้องเสียเวลาส่วนหนึ่ง ไประหว่างการส่งข้อมูลผ่านระบบสื่อสาร ดังนั้นการจะกำหนดว่าแฟ้มใดจะเก็บไว้ในระบบใดจึงเป็นสิ่งที่จำเป็น โดยเฉพาะอย่างยิ่งแฟ้มข้อมูลที่มีการใช้งานบ่อย ๆ

2. การนำไปใช้กับโปรแกรมภาษาอื่น

ในการนำไปใช้กับโปรแกรมภาษาอื่นเช่น ภาษา C/400 ก็สามารถกระทำได้เช่นเดียวกับภาษา RPG/400 แต่ต้องมีการเพิ่มส่วนของ Compiler Directive ที่จะใช้กับภาษา C คือการกำหนดโครงสร้างข้อมูล และการกำหนดฟังก์ชั่นใหม่ของการเรียกใช้ฐานข้อมูลแทนการใช้คำสั่งมาตรฐานเดิมที่มีอยู่เพื่อเรียกใช้ DRDBTM ส่วนอื่น ๆ ที่เหลือก็ยังคงสามารถใช้ของเดิมร่วมกับกับภาษา RPG ได้

3. การนำไปใช้กับระบบฐานข้อมูลอื่น

เนื่องจากได้มีการออกแบบการใช้งานเฉพาะสำหรับฐานข้อมูลของเครื่องไอบีเอ็มเอเอส400เท่านั้น ดังนั้นจึงไม่สามารถนำไปใช้กับเครื่องอื่น ๆ ได้ แต่อย่างไรก็ตามในแง่ของหลักการแล้ว ผู้ทำการวิจัยเชื่อว่าสามารถนำหลักการต่าง ๆ ที่ใช้ในการพัฒนาครั้งนี้ไปใช้สำหรับพัฒนาใช้กับเครื่องที่มีระบบจัดการฐานข้อมูลอื่น ๆ ได้ไม่มากก็น้อย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย