การกู้ข้อมูลจากความขัดข้องที่เกิดจากระบบคอมพิวเตอร์

ความขัดข้องของระบบมีผลกระทบโดยตรงต่อข้อมูล และค่าตัวแปรต่าง ๆ ในหน่วยความจำหลัก (main memory) โดยทั่วไปข้อมูลต่าง ๆ ที่ถูกอ่านหรือเขียน (read/write) จะไม่กระทำบนฐานข้อมูลโดยตรง แต่จะอ่านหรือเขียนข้อมูลไปพักในที่พักข้อมูลชั่วคราว (buffer) ซึ่งเป็นส่วนหนึ่งของหน่วยความจำหลัก ซึ่งจะช่วยลดการติดต่อกับส่วนรับเข้า/ส่งออก (input/output) ทำให้การทำงานเร็วขึ้น เมื่อเนื้อที่ของที่พักข้อมูลชั่วคราวเต็มหรือมีงานอื่นๆ ที่ต้องการใช้เนื้อที่ส่วนนี้ ข้อมูลในที่พักข้อมูลชั่วคราวก็จะถูกเคลื่อนย้ายไปเก็บไว้ในหน่วยความจำสำรอง คือ ฐานข้อมูลในดิสก์ การเกิดความขัดข้องจะทำให้ข้อมูลที่อยู่ในที่พักข้อมูลรับเข้า/ส่งออก (I/O buffer) สูญหาย แต่จะไม่มีผลกระทบกับข้อมูลในหน่วยความจำสำรอง คือข้อมูลในฐานข้อมูลจะไม่สูญหายไป แต่ข้อมูลอาจไม่ถูกต้องสมบูรณ์ เนื่องจากระหว่างที่ทรานแซกชันทำงานอยู่นั้น อาจมีข้อมูลบางส่วนถูกบันทึกในฐานข้อมูลแล้ว แต่บางส่วนยังคงอยู่ในที่พักข้อมูลชั่วคราวและเมื่อเกิดความขัดข้องขึ้นก็จะทำให้ข้อมูลที่อยู่ในที่พักข้อมูลชั่วคราวเกิดการสูยหายขึ้น ทำให้ข้อมูลในฐานข้อมูลอาจไม่ถูกต้องได้ วิธีการฟื้นสภาพที่เกิดจากความขัดข้องของระบบคอมพิวเตอร์ อาจใช้ ยกเลิก (UNDO) ทำซ้ำ (REDO) และ จุดตรวจสอบ (CHECKPOINT) ซึ่งจะได้กล่าวต่อไป

1.ตัวอย่างความขัดข้องของระบบ

พิจารณาตัวอย่าง *การโอนเงินจากบัญชี ก. ไปบัญชี ข.* สมมติ บัญชี ก. มียอดเงิน 1,000 บาท บัญชี ข. มียอดเงิน 800 บาท และต้องการโอนเงิน 300 บาท จากบัญชี ก. ไปยังบัญชี ข.

ขั้นตอนการทำงานเป็นดังนี้

1. อ่านยอดเงินบัญชี ก. จากฐานข้อมูลในดิสก์ไปเก็บไว้ในที่พักข้อมูลชั่วคราว (buffer) ในหน่วยความจำหลัก

2. คำนวณยอดเงินบัญชี ก. โดยหักเงินโอน 300 บาท ดังนั้นยอดเงินในบัญชี ก. ใหม่ คือ 700 บาท ซึ่งทำให้ค่าของข้อมูลในที่พักข้อมูลชั่วคราว (buffer) ในหน่วยความจำหลักเปลี่ยนแปลงไป

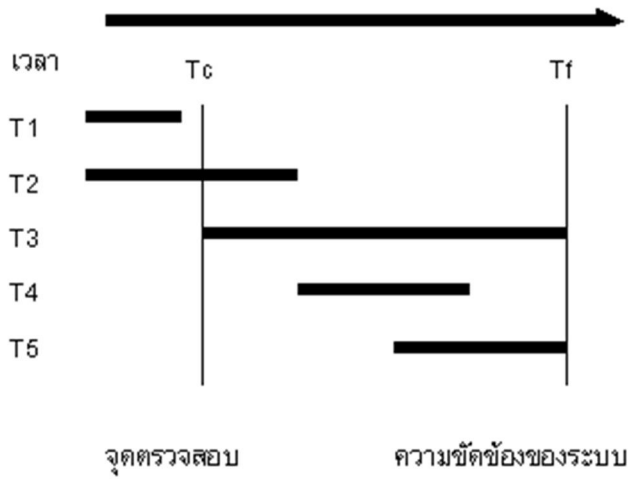
3. อ่านยอดเงินจากบัญชี ข. จากฐานข้อมูลในดิสก์ มาเก็บไว้ในที่พักข้อมูลชั่วคราวในหน่วยความจำหลัก

4. คำนวณยอดเงินบัญชี ข. ได้ยอดเงินใหม่ คือ 1100 บาท ซึ่งจะเก็บไว้ในที่พักข้อมูลชั่วคราวในหน่วยความจำหลัก

5. เมื่อที่พักข้อมูลชั่วคราวเต็ม จะมีการเคลื่อนย้ายข้อมูลจากที่พักข้อมูลชั่วคราวไปเก็บในฐานข้อมูลในดิสก์ ถ้ายอดเงินบัญชี ก. ถูกบันทึกลงฐานข้อมูลในดิสก์ก่อน แต่ยอดเงินบัญชี ข. ยังไม่ถูกบันทึกลงฐานข้อมูลในดิสก์ และเกิดปัญหาขัดข้องระหว่างนี้ก็จะทำให้ข้อมูลในที่พักข้อมูลชั่วคราวสูญหายไป และทำให้ยอดเงินบัญชี ข. ในฐานข้อมูลในดิสก์ไม่ถูกต้อง

2.ขั้นตอนการฟื้นสภาพ

จากตัวอย่างข้างต้นจะเห็นว่า แม้ว่าทรานแซกชันจะทำงานจนเสร็จสิ้นถึงคอมมิต (COMMIT) ก่อนเกิดปัญหาขัดข้องแล้วก็ตาม แต่ค่ายอดเงินของบัญชี ข. ยังอยู่ในที่พักข้อมูลชั่วคราวเป็น 1100 บาท ยังมิทันได้เก็บบันทึกลงฐานข้อมูลจริงเลย คือค่าในฐานข้อมูลยังเป็น 800 บาท ซึ่งไม่ถูกต้องเพราะค่ายอดเงินในบัญชี ก. ของฐานข้อมูลเปลี่ยนแปลงเป็นค่าใหม่แล้ว แต่ค่ายอดเงินในบัญชี ข. ยังคงเป็นค่าเดิม ทำให้เราต้องทำการฟื้นสภาพ แต่มีคำถามที่เกิดขึ้นคือเราต้องทำการฟื้นสภาพย้อนหลังถึงเมื่อไร ซึ่งโดยทั่วไปแล้วเราไม่มีโอกาสทราบได้เลยว่าเมื่อระบบเกิดข้อขัดข้องขึ้น แล้วข้อมูลส่วนไหนที่เขียนลงฐานข้อมูลไปบ้าง ดังนั้นโดยหลักการแล้วเพื่อความมั่นใจว่าข้อมูลในฐานข้อมูลถูกต้องทั้งหมด จึงต้องทำซ้ำ (REDO) ข้อมูลตั้งแต่ต้นของไฟล์ประวัติ (log file) ซึ่งกรณีนี้เป็นวิธีที่ไม่มีประสิทธิภาพ เพราะจะต้องทำซ้ำใหม่ตั้งแต่ต้น ดังนั้นจึงมี***จุดตรวจสอบ (CHECK POINT)*** ขึ้นเพื่อให้ทราบว่าจะทำการฟื้นสภาพ (recover) ย้อนหลังถึงจุดไหนนั่นเอง ซึ่งประโยชน์ของจุดตรวจสอบ คือทำให้การฟื้นสภาพทำได้เร็วขึ้น



Tc คือ จุดตรวจสอบครั้งสุดท้าย ณ เวลา Tc

Tf คือ ความขัดข้องของระบบ ณ เวลา Tf

ภาพที่ 11.1 แสดงลักษณะของทรานแซกชัน

จากภาพที่ 11.1 นั้น ลักษณะของทรานแซกชันต่าง ๆ ที่เกิดขึ้นจัดได้ 5 ลักษณะ เมื่อกำหนดระบบเกิดความขัดข้อง ณ เวลา Tf และจุดตรวจสอบครั้งสุดท้าย กระทำขึ้น ณ เวลา Tc

ทรานแซกชัน T1 กระทำเสร็จสมบูรณ์ก่อนเวลา Tc
ทรานแซกชัน T2 เกิดขึ้นก่อนเวลา Tc และเสร็จสิ้นก่อน Tf
ทรานแซกชัน T3 เกิดขึ้นก่อนเวลา Tc และณ เวลา Tf ยังทำงานไม่เสร็จ
ทรานแซกชัน T4 เกิดขึ้นหลังเวลา Tc และเสร็จสิ้นก่อน Tf
ทรานแซกชัน T5 เกิดขึ้นหลังเวลา Tc และ ณ เวลา Tf ยังทำงานไม่เสร็จ

จะเห็นได้ว่า เมื่อเกิดความขัดข้องขึ้น ณ เวลา Tf ใดๆนั้น

· ทรานแซกชัน T1 เท่านั้น ซึ่งเสร็จสิ้นสมบูรณ์และปลอดภัย เพราะคอมมิต (COMMIT) แล้วก่อนจุดตรวจสอบครั้งสุดท้าย ณ เวลา Tc และข้อมูลได้ถูกเก็บไว้ในฐานข้อมูลจริงแล้ว

· ทรานแซกชัน T3,T5 ยังไม่ได้คอมมิตจึงต้องยกเลิก (UNDO) การทำงานทั้งหมด

· ทรานแซกชัน T2,T4 คอมมิตแล้ว ถึงแม้ว่าทรานแซกชัน T2 และ T4 จะเสร็จสิ้นสมบูรณ์แล้ว แต่ยังไม่ทันที่ทรานแซกชันทำงานเสร็จสิ้นก่อนจุดตรวจสอบเพื่อเคลื่อนย้ายข้อมูลไปสู่ฐานข้อมูลจริง ก็เกิดปัญหาขัดข้องเสียก่อน ดังนั้นจึงต้องทำซ้ำ (REDO) ทรานแซกชันนั้นใหม่อีกครั้ง

ระบบจัดการฐานข้อมูลจะมีส่วนจัดการการฟื้นสภาพ (Recovery Manager; RM) เป็นตัวจัดการการฟื้นสภาพทั้งหมด โดยเริ่มต้นพิจารณาว่าทรานแซกชันใดต้องยกเลิก (UNDO) หรือ ทำซ้ำ (REDO) ซึ่งจะอาศัยข้อมูลจากไฟล์ประวัติ (log file) ดังขั้นตอนต่อไปนี้

ขั้นที่ 1 เริ่มต้นด้วยการสร้างลิสต์ของทรานแซกชัน 2 กลุ่ม คือ

· ลิสต์ทรานแซกชันของกลุ่มที่ต้องยกเลิก (UNDO-LIST)

· ลิสต์ทรานแซกชันของกลุ่มที่ต้องทำซ้ำ (REDO-LIST)

ณ จุดตรวจสอบครั้งสุดท้าย ณ เวลาที่ Tc พิจารณาว่ามีทรานแซกชันใดทำงานผ่านจุดตรวจสอบนั้น ให้นำทรานแซกชันทั้งหมดไปอยู่ในกลุ่ม UNDO-LIST ในขั้นตอนนี้จะได้ UNDO-LIST ประกอบด้วยทรานแซกชัน T2 และ T3 และในเบื้องต้นกำหนดให้ REDO-LIST เป็นลิสต์ว่าง

ขั้นที่ 2 พิจารณาข้อมูลในไฟล์ประวัติ (log file) โดยเริ่มตรวจสอบ*หลังจาก*จุดตรวจสอบครั้งสุดท้าย ณ เวลา Tc

· ถ้าพบทรานแซกชันใดเริ่มต้นทำงาน (start) ให้เพิ่มทรานแซกชันนั้นเข้าไปใน UNDO-LIST

(ในที่นี้จะได้ UNDO-LIST ประกอบด้วยทรานแซกชัน T2,T3,T4 และ T5)

· ถ้าพบทรานแซกชันใดคอมมิต (COMMIT) ให้ย้าย ทรานแซกชันนั้นจาก UNDO-LIST ไปยัง REDO-LIST

(ในที่นี้ REDO-LIST ประกอบด้วยทรานแซกชัน T2 และ T4 และ UNDO-LIST ประกอบด้วยทรานแซกชัน T3 และ T5

โดยสรุปจะได้ลิสต์ทรานแซกชันทั้ง 2 กลุ่มมาดำเนินการ UNDO หรือ REDO ต่อไป โดยส่วนจัดการการฟื้นสภาพจะพิจารณาข้อมูลในไฟล์ประวัติ กรณีที่ยกเลิก (UNDO) ก็จะดึงข้อมูลเดิม (old value) ก่อนการเปลี่ยนแปลงกลับมา และเริ่มทำงานตามทรานแซกชันนั้นอีกครั้ง สำหรับกรณีทำซ้ำ (REDO) ก็จะไปดึงข้อมูลหลังการเปลี่ยนแปลง (new value) มาแทนที่ จนกระทั่งทรานแซกชันนั้นสมบูรณ์

มีข้อสังเกตคือทรานแซกชัน T2 จะทำซ้ำ (REDO) เฉพาะการเปลี่ยนแปลงที่เกิดขึ้นหลังจุดตรวจสอบ ณ เวลา Tc เท่านั้น เนื่องจากการเปลี่ยนแปลงของทรานแซกชันก่อนจุดตรวจสอบ ณ เวลา Tc ได้ถูกบันทึกในฐานข้อมูลจริงแล้ว และในทำนองเดียวกันทรานแซกชัน T3 ก็จะยกเลิก (UNDO) เฉพาะการเปลี่ยนแปลงที่เกิดขึ้นก่อนจุดตรวจสอบ ณ เวลา Tc เท่านั้น เนื่องจากการเปลี่ยนแปลงที่เกิดขึ้นหลังจากจุดตรวจสอบ ณ เวลา Tc กระทำอยู่ในที่พักข้อมูลชั่วคราว (buffer) เท่านั้นยังไม่ได้บันทึกลงฐานข้อมูลจริง ส่วนการเปลี่ยนแปลงที่เกิดก่อนจุดตรวจสอบ ณ เวลา Tc ได้ถูกเก็บไว้ในฐานข้อมูลแล้ว ดังนั้นจึงจำเป็นต้องยกเลิกการเปลี่ยนแปลงต่างๆ ที่เกิดขึ้นก่อนจุดตรวจสอบ ณ เวลา Tc เพื่อให้ข้อมูลในฐานข้อมูลกลับสู่สภาพเดิม

ที่มา http://sot.swu.ac.th/Portals/156/sot/CP342/lesson11/ms1t2.htm