บทที่ 3

ปัจจัยที่ส่งผลกระทบต่อสมรรถนะระบบคอมพิวเตอร์

เมื่อได้เรียนรู้แนวความคิดและสถาปัตยกรรมของระบบเครื่องคอมพิวเตอร์หลายซีพียูแล้ว เพื่อให้สามารถวิเคราะห์สมรรถนะได้อย่างดี จำเป็นที่จะต้องทราบถึงปัจจัยที่ส่งผลต่อสมรรถนะ เพื่อจะได้หาทางหลีกเลี่ยงอุปสรรคและดำเนินการตามปัจจัยที่ส่งผลกระทบในทางบวกต่อไปในบทที่ 4

จากบทนี้ท่านผู้ศึกษาจะได้ทราบถึงปัจจัยต่างๆ ที่ส่งผลกระทบต่อสมรรถนะ[1] อันได้แก่

1. ปัจจัยจากโครงแบบของระบบเครื่องคอมพิวเตอร์ การกำหนดโครงแบบ ซึ่งประกอบด้วย จานแม่เหล็ก แถบ ทางรับเข้า-ส่งออก และอื่นๆ เพื่อก่อให้เกิดสมรรถนะสูงสุด
2. บุริมภาพ (Priority)
3. พารามิเตอร์ในการสร้างแฟ้มข้อมูล
4. ขนาดบล็อกของแฟ้มข้อมูลจานแม่เหล็ก
5. การมีบัฟเฟอร์และแคช (Buffer and cache)
6. การพาร์ทิชั่น (Partition)
7. โครงแบบจากตัวควบคุมจานแม่เหล็ก
8. ปัจจัยจากอุปกรณ์ (Device factor)
9. ปัจจัยจากงานประยุกต์และโปรแกรมอรรถประโยชน์

3.1 โครงแบบของระบบ[2]

โครงแบบนับเป็นสิ่งสำคัญที่สุดของระบบคอมพิวเตอร์ขนาดใหญ่ เป็นตัวระบุการจัดสรรอุปกรณ์ต่างๆ ซึ่งใช้ในการทำงานของคอมพิวเตอร์

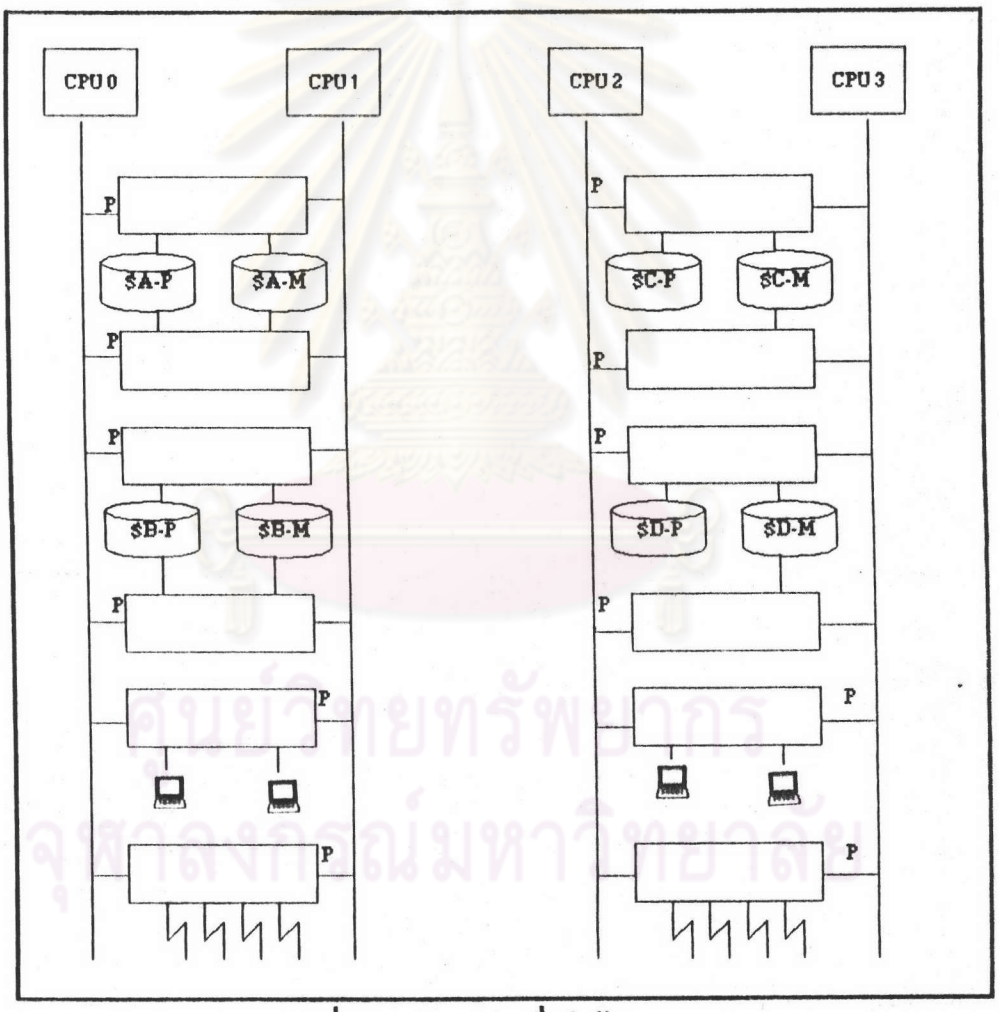
---

[1] Tandem Computer, Introduction to Performance Analysis. (1989)

[2] Tandem Computer, Performance Analysis and Tuning. (1989)

โครงแบบระบบ (System Configuration) ซึ่งเกิดจากการก่อกำเนิดระบบใหม่ (System Generation) ระบบใดก็ตามหากมีการจัดโครงแบบอย่างดี ก็ส่งผลให้มีการทำงานได้อย่างมีประสิทธิภาพ ปัญหาคือ รูปแบบใดที่จัดว่าเป็นโครงแบบที่ดี อัตราส่วนของตัวควบคุมกับอุปกรณ์ควรเป็นเช่นไร หรือ ระหว่างซีพียูกับตัวควบคุม และหน่วยความจำต่อซีพียู สิ่งเหล่านี้เป็นปัจจัยที่กำหนดการช่วงชิง (Contention) การใช้งานของปริมาณงานที่เข้ามาในระบบว่าจะสามารถให้บริการได้อย่างทันท่วงทีหรือไม่

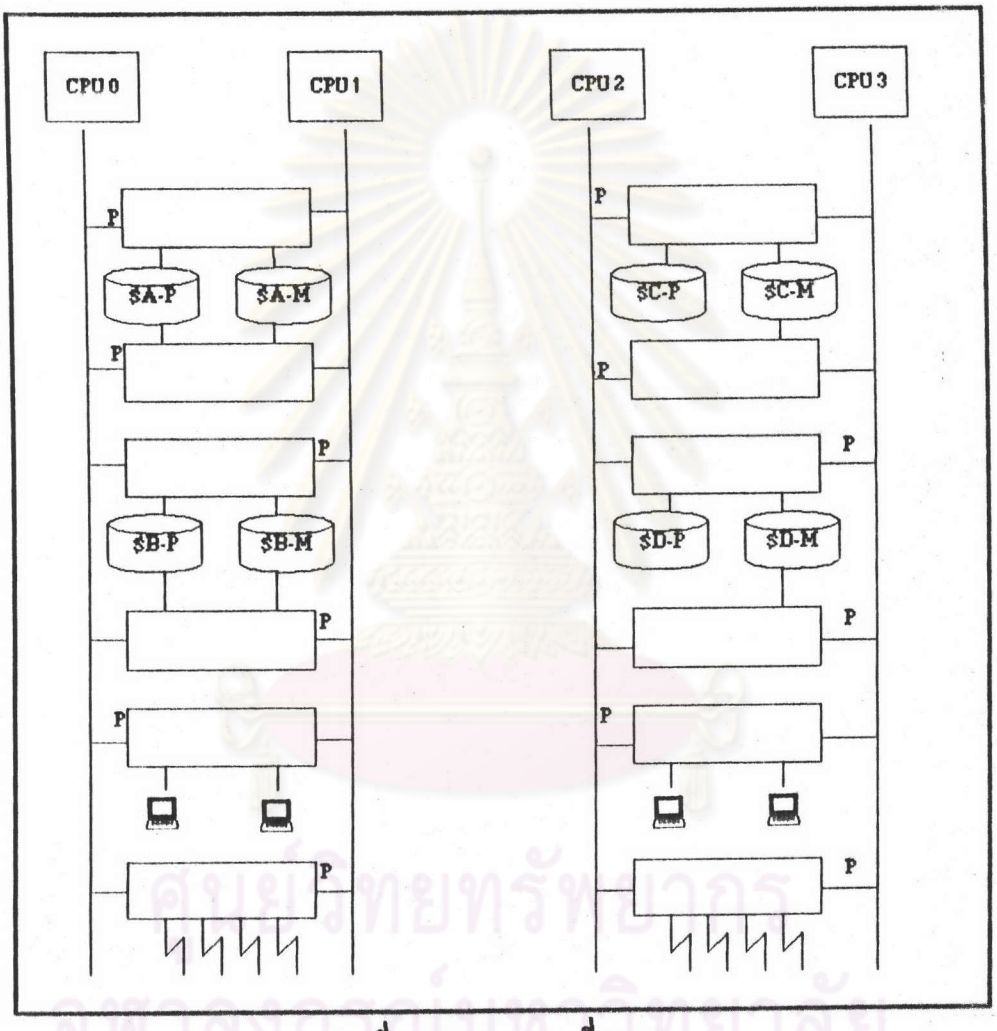
ในการกำหนดรูปแบบของโครงแบบของระบบนั้น สามารถกำหนดได้หลายรูปแบบซึ่งมีผลต่อสมรรถนะต่างกันดังรูป 3.1 รูป 3.2 และรูป 3.3



รูปที่ 3.1 โครงแบบที่ไม่ได้สมดุล

จากรูปที่ 3.1 แสดงโครงแบบระบบ ซึ่งส่งผลให้ระบบเครื่องคอมพิวเตอร์อาจมีปัญหาเรื่องสมรรถนะ เหตุเพราะการจัดการจานแม่เหล็กต่างๆไม่สมดุล ขอให้สังเกตตัวอักษรพี (P)

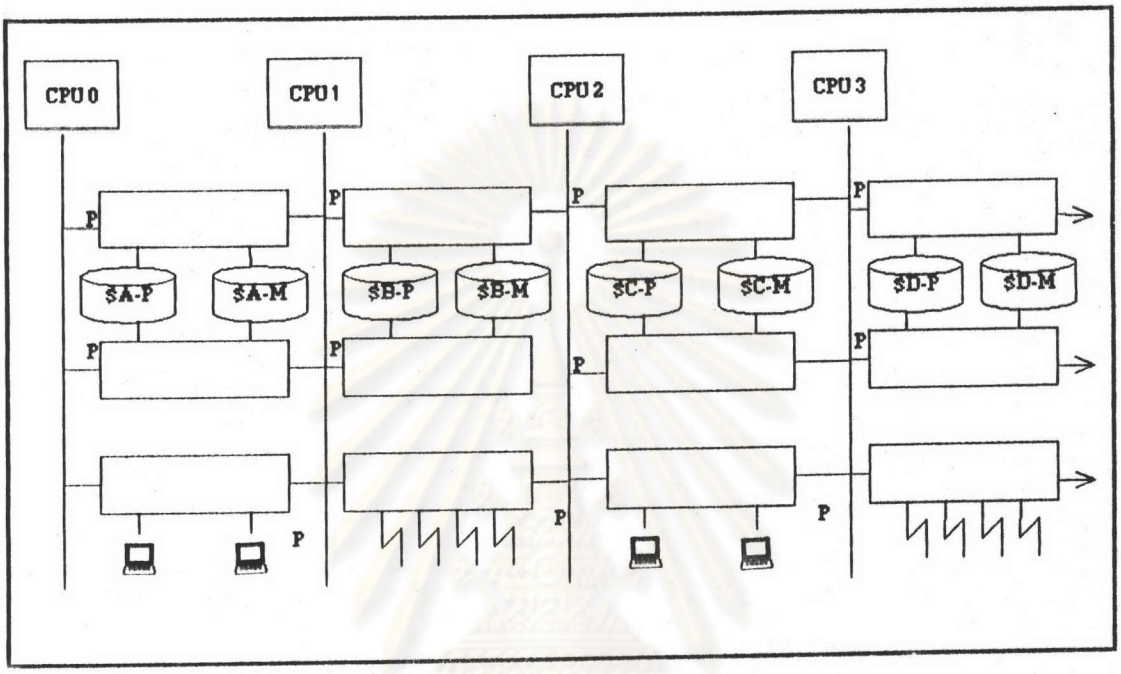
ในรูป แสดงถึงเส้นทางหลักของข้อมูล กล่าวคือ จานแม่เหล็กชื่อ $A-P จะผูกอยู่กับซีพียู 0 และซีพียู 1 แต่เมื่อมีการเข้าถึงข้อมูลในจานแม่เหล็ก เส้นทางเดินของข้อมูลจะไปทางซีพียู 0 ก่อนเช่นเดียวกัน ถ้าต้องการเข้าถึงจานแม่เหล็กที่ชื่อ $B-P ก็ต้องผ่านซีพียู 0 จากภาพโดยรวมทั้งหมด จะเห็นว่าการจัดสรรจานแม่เหล็กและอุปกรณ์เชื่อมโยงไม่ได้สมดุล สำหรับการเข้าถึงจานแม่เหล็ก ซีพียู 0 จะรับภาระหนักกว่าซีพียู 1 และซีพียู 2 รับภาระหนักกว่าซีพียู 3



รูปที่ 3.2 โครงแบบที่สมดุล

ในทางตรงข้ามรูปที่ 3.2 แสดงโครงแบบที่ส่งผลต่อสมรรถนะในทางบวก โดยจะสังเกตเห็นว่า เมื่อมีการเข้าถึงจานแม่เหล็กใดๆ ก็ตาม แต่ละซีพียูจะรับภาระไม่หนักไปกว่าซีพียูอื่นใดเป็นการแสดงถึงการกระจายภาระได้อย่างดีโดยไม่ต้องเปลี่ยนแปลงฮาร์ดแวร์ เพียงแต่กำหนดซีพียูในการก่อกำเนิดระบบใหม่ หรือการเปลี่ยนเส้นทางของข้อมูลใหม่ ซึ่งสำหรับเครื่องเทนเด็มใช้โปรแกรมอรรถประโยชน์ที่เกี่ยวกับอุปกรณ์รอบข้าง หรือ พัพ (PUP or Peripheral

Utility Program) และใช้คำสั่งย่อย คือ PRIMARY



รูปที่ 3.3 โครงแบบที่สมดุลและมีสมรรถนะสูง

จากรูปทั้งสองยังคงมีข้อจำกัดบางประการในการทำงานแบบคู่ขนาน (Parallelism) ซึ่งหากต้องการให้ทุกอุปกรณ์มีการกระจายภาระได้ดีมาก ควรจะเป็นดังรูปที่ 3.3 ซึ่งใช้ทุกซีพียู ในการติดต่อกับจานแม่เหล็ก และจะมีผลในการทนต่อความผิดพร่องสูงมาก เป็นการเพิ่มการทำงานแบบคู่ขนานมากขึ้น แต่รูปแบบดังกล่าวก็มีข้อเสียในการจัดการโครงแบบ คือมีความซับซ้อนสูงมาก

ได้มีการวิจัยเพื่อแสดงถึงการจัดสรรอุปกรณ์ภายใต้โครงแบบที่จะทำให้เกิดการช่วงชิงต่ำ คือ

โครงแบบที่ทำให้เกิดการช่วงชิงต่ำ
ตัวควบคุมจานแม่เหล็ก 2-4 ชุดต่อ ซีพียู มิรเรอร์ดิสก์ 2-4 ชุดต่อ ซีพียู หน่วยความจำที่มากกว่าหรือเท่ากับ 8 เมกะไบต์ต่อซีพียู

ตาราง 3.1 โครงแบบที่ทำให้เกิดการช่วงชิงต่ำ

จากตารางดังกล่าวเป็นเพียงแนวทางในการสร้างโครงแบบเท่านั้น ในการทำงานที่เป็นจริง แล้วจะต้องคำนึงถึงปัจจัยอื่นๆ ด้วย เช่น โครงแบบจานแม่เหล็ก แคช หน่วยความจำ และความต้องการอื่นๆ ประกอบ ทั้งนี้ต้องอาศัยประสบการณ์มากพอสมควรในการวินิจฉัย ข้อมูลเบื้องต้นที่ใช้ในการวิเคราะห์สมรรถนะ คือ รายงานจากการใช้คำสั่ง PUP LISTDEV และ LISTCACHE เปรียบเทียบกับแฟ้มข้อมูลเริ่มต้นในการก่อกำเนิด ซึ่งถ้ามีความแตกต่างกัน ก็อาจมีปัญหาสมรรถนะ ถัดมาก็คือ การตรวจสอบค่าต่างๆ เช่น พีซีบี แอลซีบี และดีซีที หากค่าต่างๆ เหล่านี้ไม่เพียงพอ จะทำให้เกิดปัญหาสมรรถนะ โปรแกรมอรรถประโยชน์ พีค ช่วยในการตรวจสอบการช่วงชิงการใช้งานตารางระบบ โปรแกรมเมเซอร์ ก็จะดักเก็บข้อมูลที่เกี่ยวกับการจองแอลซีบี และข้อมูลเกี่ยวกับซีพียู ภายใต้เมเซอร์ จะแสดงขนาดของตารางระบบให้ทราบ

ในบางกรณีการจัดโครงแบบโดยมีจานแม่เหล็กแฟนท่อม (Phantom disk) คือ การจัดโครงแบบโดยระบุจานแม่เหล็ก แต่ยังไม่ได้ใช้งาน เป็นแต่เป็นการเผื่อไว้ในอนาคตเท่านั้น อาจก่อให้เกิดปัญหาสมรรถนะ เนื่องจากระบบต้องกันเนื้อที่บางส่วนของหน่วยความจำเพิ่มขึ้นสำหรับจานแม่เหล็ก หากจานแฟนท่อมเป็นคู่จะมีผลต่อซีพียูอย่างน้อยก็คู่หนึ่ง โปรแกรมอรรถประโยชน์ พีค จะแสดงหน่วยความจำที่ใช้ในแต่ละซีพียูให้ทราบ หน่วยความจำที่แฟนท่อมใช้กระบวนการรับเข้า-ส่งออกจะคำนวณโดยใช้ ค่าพรี-เพจ-แมก(Pres-Pages-Max Process) จากเมเซอร์ ด้วยคำสั่ง LIST PROCESS ตามด้วยชื่อกระบวนการรับเข้า-ส่งออก โดยปกติกระบวนการที่จัดการจานแฟนท่อมจะใช้เนื้อที่หน่วยความจำ 10 หน้า ถ้ามีการกำหนดกระบวนการจานแม่เหล็ก 3 กระบวนการสำหรับจานแฟนท่อม 2 ชุด ดังนั้นจะใช้เนื้อที่หน่วยความจำเท่ากับ 6 กระบวนการหลัก กับ 6 กระบวนการสำรองคูณกับ 10 หน้ารวมทั้งสิ้นเป็น 120 หน้า หรือประมาณ .25 เมกะไบต์ ดังนั้นเพื่อเป็นการประหยัดเนื้อที่หน่วยความจำก็ควรจะนำจานแฟนท่อมออกจากโครงแบบของระบบ

ในการกำหนดโครงแบบที่เกี่ยวพันกับแคชพบว่า หากมีจำนวนซีพียูมากการใช้บริการ

จากแคชจะมีมากขึ้น และมีลักษณะบริการคู่ขนานด้วย การกำหนดแคชบัฟเฟอร์ไม่ดีจะก่อให้เกิดการเข้าถึงอุปกรณ์เพิ่มขึ้น ซึ่งจะใช้หน่วยความจำมากกว่าใช้แคช การจะทราบว่าการกำหนดขนาดแคชเพียงพอหรือไม่ จะต้องดูจากรายงานของคำสั่ง PUP LISTCACHE เขตข้อมูลที่ชื่อแคชฟอล์ท (CACHE FAULTS) ซึ่งควรจะมีค่าน้อยมากใกล้เคียงศูนย์ หรืออาจดูจาก เมเซอร์ในการวิเคราะห์การใช้แคช บางครั้งการพิจารณาว่า ควรใช้ขนาดเท่าใดจะพบว่าจะต้องเกี่ยวพันกับพารามิเตอร์ในการสร้างแฟ้มข้อมูล เช่น ขนาดบล็อก การกำหนดขนาดแคชสามารถทำได้โดยคำสั่ง PUP SETCACHE vol-name,4096=1000 เป็นต้น

3.2 บุริมภาพของกระบวนการ

ระบบปฏิบัติการการ์เดียนที่กำลังศึกษานี้ไม่ใช่ระบบปฏิบัติการแบบแบ่งกันใช้เวลา (time sharing) แต่เป็นระบบการทำงานโดยบุริมภาพ การ์เดียน ก็คือ ชุดของกระบวนการที่ร่วมมือกันทำงาน ซึ่งเรียกว่า กระบวนการระบบ กระบวนการระบบควรจะมีบุริมภาพสูงสุดในการจัดสรรทรัพยากร ส่วนกระบวนการที่เหลือไม่จำเป็นต้องมีบุริมภาพเท่ากัน แต่ควรจะจัดสรรให้ร่วมมือกันในการใช้งานได้ดีที่สุด

ค่าบุริมภาพที่กำหนดจะมีความสำคัญน้อย ถ้ากำหนดให้กับกระบวนการระบบ สิ่งสำคัญกว่า คือ ช่วงของบุริมภาพซึ่งสัมพันธ์กัน กระบวนการซึ่งมีบุริมภาพสูงกว่าจะสามารถเข้าไปประมวลผลในซีพียูก่อนกระบวนการที่มีบุริมภาพต่ำ เช่น หากมี 2 กระบวนการซึ่งมีบุริมภาพ 100 กับ 99(ตัวเลขมาก คือ กระบวนการที่มีบุริมภาพสูง)ซึ่งทั้งคู่ต่างช่วงชิงในการประมวลผลในซีพียู กระบวนการที่มีบุริมภาพ 100 จะได้เข้าถึงซีพียูก่อน สำหรับกระบวนการผู้ใช้ บางโอกาสอาจจำเป็นต้องที่ต้องใช้การผสมผสานบุริมภาพ เมื่อไรก็ตามที่ความต้องการใช้ซีพียูมีมากเกินความจุที่จะให้บริการได้ การจัดการบุริมภาพจะไม่ช่วยแก้ไขปัญหาเลย ตรงกันข้ามกลับจะเป็นเหตุให้เกิดปัญหามากขึ้นไปอีก

กรรมวิธีในการกำหนดบุริมภาพ คือ กำหนดเป็นชุดของกระบวนการต่างๆ บนพื้นฐานของหน้าที่ โดยปกติกระบวนการผู้ใช้สามารถถูกกำหนดเป็นกลุ่มดังนี้ เฝ้าคุม (monitor) ตัวให้บริการ (server) ตัวร้องขอ (requester) โปรแกรมแปลคำสั่ง (command interpreter) และ งานแบบกลุ่ม (batch)

เฝ้าคุม (monitor) ควรเป็นกระบวนการที่มีช่วงบุริมภาพสูงสุด เนื่องจากกระบวนการเหล่านี้ทำหน้าที่เสมือนเป็นกระบวนการแม่ในการคลอดกระบวนการลูก และคอยดูแลการฟื้นฟูระบบการทนต่อความผิดพร่อง ตัวอย่างเช่น กระบวนการ

พาทมอน(pathmon) ในระบบพาทเวย์ (pathway system)กระบวนการตัวกำกับดูแล (supervisor)ของระบบตัวเก็บพัก (spooler)

ตัวบริการ (server) ควรเป็นกระบวนการที่มีบุริมภาพถัดมาจากกระบวนการเฝ้าคุม เป็นกระบวนการที่ให้บริการ ตัวอย่างเช่นตัวบริการของระบบ พาทเวย์ และกระบวนการตัวเก็บข้อมูลในระบบตัวเก็บพัก

ตัวร้องขอ (requester) ควรเป็นกระบวนการซึ่งมีบุริมภาพถัดมาจากกระบวนการตัวบริการ เป็นกระบวนการที่ร้องขอบริการจากตัวบริการ ตัวอย่างเช่น ทีซีพี (TCP)

คำสั่งงานในการแปลคำสั่ง (Command Interpreter) ควรเป็นกระบวนการซึ่งมีบุริมภาพถัดมาจากกระบวนการตัวร้องขอ ตัวอย่างเช่น แทคเคิล (TACL or Tandem Advanced Command Language) ซึ่งเหมือนกับ เชลสคริปต์ในยูนิกซ์ พาทคอมม์ (Pathcom) สปูลคอมม์ หรือ กระบวนการอื่นที่กำหนดให้ทำหน้าที่ติดต่อกับระบบปฏิบัติการหรือกระบวนการเฝ้าคุม เพื่อป้องกันการทำงานวนซ้ำ กระบวนการนี้ควรมีบุริมภาพสูงกว่ากระบวนการผู้ใช้ทั้งหมด

งานแบบกลุ่ม(Batch) ควรเป็นระดับที่ต่ำที่สุด คือ ต่ำกว่ากระบวนการแปลคำสั่ง ตัวอย่างเช่น การใช้โปรแกรมการสร้างรายงาน การคอมไพล์โปรแกรม (กรณีที่มีการพัฒนาระบบงาน) กระบวนการเหล่านี้จะเปลืองการใช้ซีพียูและไอโอ

จากคำอธิบายข้างต้นสามารถสร้างตัวอย่างในการกำหนดบุริมภาพดังนี้

กลุ่มกระบวนการ	ช่วงบุริมภาพ
เฝ้าคุม (Monitor)	170-179
ตัวบริการ (server)	160-169
ตัวร้องขอ (requester)	150-159
คำสั่งงานแปลคำสั่ง	140-149
กลุ่ม (Batch)	130-139

ตารางที่ 3.2 แสดงความสัมพันธ์ของบุริมภาพของกระบวนการในระบบ

สามารถใช้กระบวนการ *CMON เพื่อช่วยควบคุมบุริมภาพของกระบวนการต่างๆ ในระบบโดยมีการกำหนดบุริมภาพอยู่ในโครงแบบอีกชั้นหนึ่ง

ในบางโอกาสอาจมีการเปลี่ยนแปลงบุริมภาพในขณะที่กระบวนการนั้นกำลังทำงานอยู่ ลักษณะดังกล่าวอาจก่อผลเสียมากกว่าผลดี เนื่องจากการเปลี่ยนให้กระบวนการหนึ่งมีค่าบุริมภาพสูงกว่าอีกกระบวนการเท่ากับเป็นการเปลี่ยนช่วงเวลาการใช้ทรัพยากร ทำให้ภาระดังกล่าวไม่สามารถควบคุมหรือปรับให้สมดุลได้อย่างดี แต่บางครั้งการปรับเปลี่ยนบุริมภาพก็มีประโยชน์ในด้านการปกป้องไม่ให้กระบวนการที่ตั้งใจให้กระทำการ ทำงานเร็วเกินไปหรือช้าเกินไป

สภาวการณ์บางขณะกระบวนการที่มีบุริมภาพต่ำสามารถรบกวน แล้วส่งผลกระทบต่อกระบวนการที่มีบุริมภาพสูง เช่น งานทางด้านประมวลผลรายการเปลี่ยนแปลงแบบเชื่อมตรงหรืองานกระบวนการรับเข้า-ส่งออก สาเหตุพื้นฐาน คือ กระบวนการที่ร้องขอ ซึ่งมีบุริมภาพต่ำกำลังทำงานในซีพียูซึ่งมีภาระไม่มาก (สมมติให้เป็นซีพียู 0) และส่งคำร้องขอไปยังกระบวนการจานแม่เหล็กในอีกซีพียู (สมมติให้เป็นซีพียู 3) กระบวนการซึ่งมีบุริมภาพสูงกว่าก็ทำงานที่ซีพียู 3 เนื่องจากซีพียู 0 มีภาระไม่มาก ดังนั้นกระบวนการที่มีบุริมภาพต่ำสามารถสร้างคำร้องขอไปยังกระบวนการจานแม่เหล็กด้วยความเร็วสูงได้ จำนวนคำร้องขอจากกระบวนการบุริมภาพต่ำถูกส่งไปยังกระบวนการจานแม่เหล็กในซีพียู 3 และทำให้ต้องใช้เวลาของซีพียูมากในซีพียู 3 สิ่งนี้ทำให้กระบวนการที่มีบุริมภาพสูงไม่สามารถทำงานในซีพียูได้ตลอดเวลา จะเห็นได้ว่ากระบวนการบุริมภาพต่ำ (ในซีพียู 0) สามารถส่งผลต่อกระบวนการบุริมภาพสูงในทางลบที่ซีพียู 3

จากปัญหาข้างต้นสามารถหาทางแก้ไขได้ 3 ทางดังนี้

ทางแก้ที่ 1 ก. ให้กระบวนการบุริมภาพต่ำอยู่ในซีพียูเดียวกับกระบวนการจานแม่เหล็ก

ข. กระบวนการบุริมภาพสูงกว่าและต่ำกว่าอยู่ซีพียูเดียวกัน

ค. กระบวนการบุริมภาพสูงกว่าอยู่ในซีพียูอื่น ซึ่งไม่ใช่ซีพียูเดียวกับกระบวนการจานแม่เหล็กทั้งกระบวนการสำรองด้วย

ทางเลือกนี้จะช่วยลดผลกระทบของกระบวนการบุริมภาพต่ำในการรอกระบวนการบุริมภาพสูงกว่า หรือเป็นการให้กระบวนการบุริมภาพสูงควบคุมกระบวนการบุริมภาพต่ำในข้อ ก และ ข หรือการเคลื่อนย้ายภาวะการช่วงชิงการใช้ซีพียูของกระบวนการบุริมภาพสูงกว่าในข้อ ค

ทางแก้ที่ 2 ย้ายแฟ้มข้อมูลซึ่งกระบวนการบุริมภาพต่ำต้องการเข้าถึงไปยังอุปกรณ์ ซึ่งโยงกับซีพียูอื่น ที่ไม่ใช้ซีพียูที่กระบวนการบุริมภาพสูงกว่าทำงานอยู่ จุดนี้ทำให้เกิดการทำงานควบคู่ไม่ต้องรอกัน

ทางแก้ที่ 3 แบ่งแฟ้มข้อมูลที่กระบวนการบุริมภาพต่ำต้องการเข้าถึง โดยให้จานแม่เหล็กที่บรรจุแฟ้มนั้นผูกกับซีพียูอื่น ซึ่งไม่ใช้ซีพียูเดียวกับกระบวนการบุริมภาพสูง จุดนี้เป็นการลดการรบกวนของกระบวนการบุริมภาพต่ำในการใช้จานแม่เหล็กผ่านกระบวนการ เป็นการกระจาย

ความต้องการการให้บริการ

3.3 พารามิเตอร์ในการสร้างแฟ้มข้อมูล

การเลือกพารามิเตอร์ในการสร้างแฟ้มข้อมูลส่งผลต่อสมรรถนะทั้งในทางบวกและทางลบ สามารถแบ่งเป็น 5 ส่วนย่อยๆคือ

1. พารามิเตอร์ซึ่งเพิ่มสมรรถนะ เช่นการเลือกโครงสร้างแฟ้มข้อมูลที่เหมาะกับชนิดการเข้าถึงข้อมูล การเลือกขนาดบล็อก การกำหนดบัฟเฟอร์และการตัดแบ่งโครงสร้างแฟ้ม

2. พารามิเตอร์ซึ่งลดสมรรถนะ เช่น กำหนดให้บล็อกดัชนีและ/หรือบล็อกข้อมูลที่บีบอัด (compress) กำหนดให้มีการบันทึกข้อมูลทุกครั้งที่เกิดระเบียน

3. ชนิดของการจัดการฐานข้อมูล แบบเอสคิวแอล(SQL)หรือแบบใช้เอ็นช์สไคร์บ (ENSCRIBE)

4. การลงรหัสโปรแกรม กลไกต่างๆที่ส่งผลต่อสมรรถนะ

5. การจัดการระบบ เช่น ชนิดของตัวควบคุมจานแม่เหล็กและอุปกรณ์อื่น การอ่านและบันทึกแบบขนาน และการเกิดการแบ่งบล็อก (block splits)

ก่อนจะกล่าวรายละเอียดต่อไปคงต้องทบทวนชนิดของโครงสร้างแฟ้มข้อมูลก่อน บนเครื่องเทนเด็ม ซึ่งมีโปรแกรมในการจัดการแฟ้มข้อมูลที่ชื่อ เอ็นช์สไคร์บ (ENSCRIBE) มีชนิดแฟ้มข้อมูล 4 ชนิด คือ แบบไร้โครงสร้าง (unstructued) และ กลุ่มแฟ้มข้อมูลมีโครงสร้างมีแฟ้มข้อมูลดัชนี (key sequential) เรียงลำดับ (entry sequential) และแบบสัมพัทธ์ (relative) เฉพาะแฟ้มข้อมูลแบบดัชนีจะสามารถเข้าถึงแฟ้ม แบบใช้กุญแจรอง (alternate key)ได้อีก โดยกุญแจรองนี้จะอยู่ในแฟ้มข้อมูลกุญแจรอง (alternate key file)

สรุปคุณลักษณะของโครงสร้างแฟ้มข้อมูลชนิดต่างๆ

แฟ้มข้อมูลไร้โครงสร้าง (Unstructued file) ประกอบด้วยสายธารของไบต์ที่ต่อเนื่องกันโดยคีย์หลัก คือไบต์เลขที่อยู่สัมพัทธ์ (relative byte address) ซึ่งจะไม่มีโอเวอร์เฮดจากโครงสร้างแบบบล็อก แฟ้มข้อมูลชนิดนี้เหมาะกับการเข้าถึงแบบง่ายๆ เช่น ระเบียนที่ความยาวคงที่ การแฮช (hashing)และการเพิ่มเติมตอนท้ายแฟ้ม

แฟ้มข้อมูลดัชนี (Key sequenced file) โครงสร้างที่มีกุญแจหลัก (primary key) ที่ต่อเนื่องกัน ส่งผลให้สมรรถนะสูงขึ้น การเข้าถึงโดยกุญแจหลัก ซึ่งควบคุมโดยบล็อกดัชนี โดยที่แต่ละระดับของดัชนีแทนการเข้าถึงทางกายภาพ (ผ่านแคชหรืออุปกรณ์) ก่อนจึงจะเข้าถึงบล็อกข้อมูลต่อไป สมรรถนะของการเข้าถึงแฟ้มข้อมูลขึ้นกับจำนวนของระดับของดัชนีและความถี่ของการแบ่งบล็อก (ซึ่งถ้ามีการแบ่งบล็อกน้อย จะทำให้สมรรถนะสูง) ทางแก้เพื่อลดระดับของดัชนี คือ การสร้างแฟ้มข้อมูลให้มีขนาดบล็อกใหญ่และการพาร์ทิชั่น (partitioning) การลดการแบ่งบล็อก (block split) ก็โดยการบรรจุข้อมูลบ่อยๆ กรณีเป็นแฟ้มข้อมูลแบบกุญแจรอง ซึ่งเป็นแฟ้มข้อมูลแบบดัชนีเช่นกัน ดังนั้นสมรรถนะของแฟ้มดังกล่าวจะมีลักษณะเช่นเดียวกันด้วย กรณีแฟ้มข้อมูลดังกล่าวมีหลายกุญแจ ก็ควรจะสร้างหลายแฟ้มและอยู่ต่างจานแม่เหล็กกันเพื่อให้สมรรถนะสูง และการพาร์ทิชั่นแฟ้มข้อมูลดังกล่าวจะช่วยให้มีประสิทธิภาพสูง

แฟ้มข้อมูลสัมพัทธ์ (Relative file) แฟ้มข้อมูลชนิดนี้เหมาะกับการเข้าถึงแบบสุ่ม ซึ่งมีกุญแจ คือ เลขที่ระเบียนสัมพัทธ์ (relative record number) ซึ่งเป็นตัวเลขแทนตำแหน่งระเบียนทางกายภาพโดยนับจากต้นแฟ้มข้อมูล

แฟ้มข้อมูลแบบเรียงลำดับ (Sequential file) แฟ้มข้อมูลซึ่งมีกุญแจหลัก คือ ไบต์เลขที่สัมพัทธ์ จะเพิ่มข้อมูลโดยการเพิ่มเติมในตอนท้ายของแฟ้มข้อมูล

กลไกการทำงานของการเข้าถึงแฟ้มข้อมูล คงวนเวียนอยู่กับ 3 ขั้นตอนใหญ่ คือ เปิด (open) --> เข้าถึง (access) --> ปิด (close) ในแต่ละขั้นตอนเกี่ยวข้องกับสมรรถนะในแง่มุมของตารางระบบและความถี่ของการเข้าถึงมากเพียงใด กล่าวคือ ในการเปิดแฟ้มข้อมูลจะต้องมีการจัดสรรบล็อกควบคุมการเปิดแฟ้ม (Open control block) และตารางการเข้าถึงบล็อก (Access control block) ในกรณีถ้าเกี่ยวข้องกับแฟ้มข้อมูลจานแม่เหล็กก็จะทำให้เกิดแฟ้มควบคุมบล็อก(File Control Blobk) และเกี่ยวพันกับจำนวนพาร์ทิชั่นของแฟ้มข้อมูล หากความถี่ของการเปิด --> เข้าถึง --> ปิด มีมากก็ย่อมส่งผลต่อสมรรถนะในการประมวลผลแฟ้มข้อมูลเป็นอันมาก ซึ่งสามารถตรวจสอบพบได้โดยใช้ เมเซอร์ โดยหากมีจำนวนของแฟ้มและการเปิดใช้จานแม่เหล็กมากในช่วงเวลาจับข้อมูลสั้นๆ นั่นเป็นเครื่องหมายแสดงว่ามีปัญหาแล้ว

จุดหนึ่งที่น่าสนใจ คือ การมีกระบวนการชั่วครู่ (Transient process)จำนวนมาก

ก็ก่อให้เกิดปัญหาสมรรถนะได้ เนื่องจากเกิดภาวะการเริ่มต้นทำงานของรหัสของโปรแกรม และความผิดพร่องของข้อมูลในหน่วยความจำ ในการสร้างกระบวนการ

พารามิเตอร์ที่น่าสนใจมีดังนี้

DCOMPRESS
ICOMPRESS

คือการกำหนดให้แฟ้มข้อมูลมีการบีบอัดข้อมูลทั้งในระดับของข้อมูลและดัชนี สำหรับแฟ้มข้อมูลดัชนี โดยจะทำการบีบอัดข้อมูลตั้งแต่เริ่มต้นบล็อกในแต่ละครั้งของการทำงานกับระเบียน และทำการถอดการบีบอัดข้อมูลเมื่อมีการเรียกใช้ระเบียนนั้นๆ จะเห็นว่าพารามิเตอร์นี้ จะใช้ทรัพยากรของซีพียูมากเกินไปจะเห็นได้จากตารางข้างล่างนี้ สมมติว่าข้อมูล 1 บล็อกมีระเบียน 6 ระเบียน จะต้องการขั้นตอนการถอดการบีบอัดข้อมูล 21 ครั้ง เพราะสำหรับข้อมูล 1 ระเบียนจะใช้ 1 ครั้ง แต่ถ้าเป็นข้อมูล 2 ระเบียน จะทำการเข้าถึงครั้งที่ 1 และครั้งที่ 2

ระเบียนที่ต้องการเข้าถึง	การเข้าถึงระเบียนของบล็อกที่บีบอัด
1 -->	1
2 -->	1 2
3 -->	1 2 3
4 -->	1 2 3 4
5 -->	1 2 3 4 5
6 -->	1 2 3 4 5 6

ตารางที่ 3.3 แสดงผลกระทบของการบีบอัดข้อมูลและดัชนี

หากจำนวนระเบียนในบล็อกเป็น 12 จะต้องใช้กรรมวิธีในการถอดการบีบอัด 78 ครั้ง ถ้าเป็นการเรียงลำดับ แต่จะใช้เพียง 12 ครั้งในการที่จะนำระเบียนสุดท้ายในบล็อกสำหรับการเข้าถึงแบบสุ่ม ความต้องการประหยัดเนื้อที่จะทำให้มีการใช้ซีพียูสูง

REFRESH

สารบบจานแม่เหล็ก (disk directory) คือ โครงสร้างข้อมูลที่มีการเรียงลำดับตามกุญแจบนจานแม่เหล็กเชิงกายภาพ ป้าย (Label)ของแฟ้มข้อมูล คือ

ระเบียนต่างๆ ซึ่งอยู่ในสารบบ หากแฟ้มข้อมูลเป็นลักษณะพาร์ทิชั่น (partition) ซึ่งเป็นแฟ้มข้อมูลที่อยู่บนจานแม่เหล็กเชิงกายภาพหลายตัว แต่ถือเป็นแฟ้มเดียวกันในเชิงตรรกะ ก็จะมีป้ายหลายระเบียน เมื่อมีการบันทึกป้ายของแฟ้มข้อมูลไปยังจานแม่เหล็กจะเรียกว่า รีเฟรช (refresh) ซึ่งพารามิเตอร์นี้จะสามารถทำได้โดย

1. เรียกใช้กระบวนการคำสั่งจากบริการของระบบปฏิบัติการ
2. ใช้คำสั่ง PUP REFRESH
3. กำหนดพารามิเตอร์รีเฟรชในช่วงการสร้างแฟ้มข้อมูล

ทุกครั้งที่มีการเพิ่มระเบียนในแฟ้มข้อมูลจะทำการปรับปรุงป้ายนี้ จึงส่งผลกระทบต่อสมรรถนะต่อซีพียูและอุปกรณ์ที่ใช้มากเกินไป เนื่องจากต้องใช้การเข้าถึง 2 ครั้งสำหรับการบันทึก 1 ครั้งไปยังแฟ้มข้อมูล ที่กล่าวว่าต้องใช้ 2 ครั้ง เนื่องจากต้องเข้าถึงแฟ้มข้อมูลกับสารบบ และยังมีผลต่อสมรรถนะสูงมาก ถ้ามีความถี่ในการบันทึกข้อมูล เช่น แฟ้มข้อมูลพวกแบบล็อก (log) หรือ งานประเภทมีการตรวจสอบแฟ้มข้อมูลอยู่ตลอดเพื่องานกู้ระบบ (recovery)

3.4 ขนาดของบล็อกข้อมูลบนแฟ้มจาน

เพื่อให้เข้าใจปัจจัยที่ส่งผลกระทบต่อสมรรถนะของระบบคอมพิวเตอร์จากขนาดบล็อกได้ชัดเจนขึ้น จึงขอแสดงรายละเอียดดังตารางข้างล่างนี้ โดยแจกแจงขนาดของบล็อกต่างๆ คือ 512 ไบต์ 1024 ไบต์ 2048 ไบต์ และ 4096 ไบต์ตามลำดับทั้งนี้ได้กำหนดตัวแปรดังที่ดังนี้คือ

1. แฟ้มข้อมูลเป็นชนิดมีดัชนีซึ่งมีขนาด 1 ล้านระเบียน ขนาดระเบียน 128 ไบต์
2. สมมให้มีสแลค 20 เปอร์เซ็นต์ (สแลค คือ เปอร์เซ็นต์ในการจองเนื้อที่เผื่อการเพิ่มเติมระเบียนในอนาคต)
3. ข้อมูลในระเบียนให้เป็นแบบความยาวคงที่ (fixed length)
4. ระเบียนดัชนีให้มีการบีบอัดดัชนี 50 เปอร์เซ็นต์

จากตาราง ขนาดบล็อก 512 ไบต์ สามารถบรรจุได้ 3 ระเบียน ขณะที่บล็อกขนาด 4096 ไบต์ สามารถบรรจุ 31 ระเบียน จะเห็นว่าขนาดบล็อก 4096 ไบต์ จะมีการเข้าถึงอุปกรณ์เพียง 1 ครั้งก็จะได้ข้อมูล 31 ระเบียน แต่ถ้าขนาด 512 ไบต์จะเข้าถึงอุปกรณ์ 1 ครั้งต่อระเบียน 3 ระเบียน ซึ่งหากต้องการข้อมูล 31 ระเบียน จะต้องเข้าถึงอุปกรณ์ถึง 11 ครั้ง ทำให้เสียเวลาในการเข้าถึงมากกว่า และเปลืองทรัพยากรของเครื่องไม่ว่าจะเป็นซีพียู กระบวนการจานแม่เหล็ก ในการเลือกจ่ายงานของตัวเลือกจ่ายงานมากกว่า

ขนาดบล็อก	512	1024	2048	4096	หน่วย
ระเบียนข้อมูล					
ขนาดระเบียน	128	128	128	128	ไบต์
จำนวนระเบียน	3	7	15	31	ระเบียน
จำนวนไบต์ทั้งหมด	384	896	1,920	3,968	ไบต์
หัวระเบียน	38	46	62	94	ไบต์
จำนวนไบต์ที่ไม่ได้ใช้	90	82	66	34	ไบต์
จำนวนบล็อกที่บรรจุระเบียน	333,333	142,857	66,667	32,258	บล็อก
ขนาดแฟ้มข้อมูล	170.67	146.29	136.53	132.13	เมกะไบต์
ระเบียนดัชนี					
ขนาดกุญแจหลัก	12	12	12	12	ไบต์
ขนาดระเบียน	61	125	253	509	ไบต์
(อัตราบีบอัด 50 %)					
จำนวนไบต์ของดัชนี	366	750	1,518	3,054	คูณด้วย6
หัวระเบียน	146	274	530	1,042	ไบต์
จำนวนบล็อก	5,464	1,143	264	63	บล็อก
ขนาดดัชนีทั้งหมด	2.8	1.17	.54	.26	เมกะไบต์
เนื้อที่ข้อมูล	173.95	147.46	137.07	132.39	เมกะไบต์
เนื้อที่สแลค	34.69	29.49	27.41	26.48	เมกะไบต์
เนื้อที่ทั้งหมด	208.16	176.95	164.49	158.87	เมกะไบต์

ตารางที่ 3.4 ผลกระทบของขนาดบล็อกต่อสมรรถนะของระบบคอมพิวเตอร์

และหัวอ่านจานแม่เหล็ก เพื่อเข้าถึงข้อมูลที่ต้องการ ในด้านของเนื้อที่ขนาดบล็อก 512 ไบต์จะต้องใช้เนื้อที่ 333,333 บล็อกเพื่อบรรจุข้อมูล 1 ล้านระเบียน ในขณะที่ขนาดบล็อก 4096 ไบต์ ใช้เนื้อที่ 32,258 บล็อก หรือ เป็นอัตราส่วน 1 ใน 10 ของจำนวนการเข้าถึงอุปกรณ์จำนวน

บล็อกข้อมูลจะเป็นตัวกำหนดจำนวนระเบียนในบล็อกดัชนี ซึ่งแต่ละระเบียนในบล็อกดัชนีจะชี้ไปยังบล็อกข้อมูล จากตารางจะพบว่ากุญแจหลักขนาด 12 ไบต์ และ มีอัตราการบีบอัดข้อมูล 50 เปอร์เซ็นต์ส่งผลให้บล็อกขนาด 512 ไบต์ ใช้เนื้อที่ 5,464 บล็อกดัชนี โดยที่แต่ละบล็อกบรรจุ 61 ระเบียน ขณะที่เนื้อที่บล็อกขนาด 4096 ไบต์บรรจุ 61 บล็อกดัชนีโดยแต่ละบล็อกบรรจุ 509 ระเบียน ทำให้เปลืองเนื้อที่เพียง .26 เมกะไบต์ สำหรับบล็อกขนาด 4096 ไบต์ ซึ่งน้อยกว่าบล็อกขนาด 512 ไบต์ ทำให้ข้อมูลส่วนที่ต้องการมีโอกาสอยู่ในแคชมากกว่า และการประมวลผลดังกล่าวเป็นไปอย่างรวดเร็วไม่เสียเวลาเข้าถึงอุปกรณ์หลายครั้ง

โดยทั่วไปขนาดบล็อกดัชนีมักจะกำหนดให้มีขนาดใหญ่ที่สุดเท่าที่เป็นได้ เพราะจะส่งผลให้การเข้าถึงอุปกรณ์ทำได้รวดเร็วขึ้น เนื่องจากดัชนีเหล่านี้จะอยู่ในแคช ในแต่ละระดับของดัชนีทางหนึ่งที่จะทำให้ระดับดัชนีลดลง คือ การทำพาร์ทิชั่น เนื่องจากแฟ้มข้อมูลที่ถูกแบ่งออกมาจะสนใจเฉพาะบล็อกของแฟ้มเท่านั้น ไม่ได้สนใจทั้งแฟ้มใหญ่ทั้งแฟ้ม ตัวอย่างเช่น แฟ้มข้อมูลหนึ่งถูกแบ่งเป็นพาร์ทิชั่น 10 แฟ้มเท่าๆ กัน แต่ละพาร์ทิชั่นจะมีบล็อกดัชนี 7 บล็อกภายในแต่ละพาร์ทิชั่นเท่านั้น

การกำหนดขนาดบล็อกอยู่บนพื้นฐานชนิดของการเข้าถึงข้อมูลด้วยในบางครั้ง การกำหนดขนาดบล็อกใหญ่ไม่จำเป็นที่จะส่งผลดีเสมอไป กล่าวคือ

1. ถ้าแฟ้มข้อมูลมีการอ่านแบบเรียงลำดับบ่อยมาก การใช้ขนาดบล็อกใหญ่จะช่วยเพิ่มโอกาสพบข้อมูล (cache hit) และเป็นการประหยัดเนื้อที่จานแม่เหล็ก

2. แต่ถ้าแฟ้มข้อมูลเป็นการอ่านแบบสุ่มและมีขนาดใหญ่ การใช้บล็อกขนาดเล็กจะทำให้การใช้แคชน้อยลง และจำนวนส่วนวง (sector) ที่นำมาจากอุปกรณ์น้อยลง จุดนี้เป็นการเพิ่มเนื้อที่จานแม่เหล็กในการประมวลผล

3. แต่ถ้าแฟ้มข้อมูลมีขนาดเล็กและมีการเข้าถึงถี่มาก โอกาสที่จะทำให้มีการใช้แคชมากโดยมีจำนวนบล็อกน้อย และจำนวนการเข้าถึงอุปกรณ์ลดลงคงเป็นไปไม่ได้ การพาร์ทิชั่นจะช่วยให้สมรรถนะดีขึ้น จะทำให้มีการใช้แคชมากขึ้น

4. แต่ถ้าแฟ้มข้อมูลมีการปรับปรุงแบบสุ่มใช้ขนาดบล็อกเล็ก แต่ละระเบียนที่บันทึกจะต้องมีการคำนวณเลขตรวจสอบ และบันทึกกลับไปยังจานแม่เหล็ก จุดนี้ต้องการเนื้อที่จานแม่เหล็กมากขึ้นและเพิ่มระดับดัชนีสำหรับโครงสร้างแฟ้มข้อมูลแบบดัชนี

5. แต่ถ้าแฟ้มข้อมูลมีการบันทึกแบบเรียงลำดับ แล้วใช้บล็อกขนาดใหญ่และบันทึกจากบัฟเฟอร์ บล็อกขนาดใหญ่จะช่วยให้การทำงานกับบัฟเฟอร์ดีขึ้นโดยการลดการเข้าถึงอุปกรณ์

จะเห็นว่าไม่มีคำตอบตายตัวสำหรับการกำหนดขนาดบล็อก แต่ต้องอาศัยการวิเคราะห์และประสบการณ์ช่วยอย่างมาก

3.5 การใช้บัฟเฟอร์

การมีบัฟเฟอร์นับเป็นจุดเด่นมากในการเพิ่มสมรรถนะของคอมพิวเตอร์ สำหรับระบบปฏิบัติการการ์เดียนนั้นมี 2 แบบคือ

1. การบันทึกผ่านบัฟเฟอร์ (Buffered writes) จะใช้ในกรณีที่มีการบันทึกพร้อมๆ กันไปยังบล็อก เหมาะกับการเพิ่มเติมท้ายแฟ้มข้อมูล สามารถใช้กับแฟ้มข้อมูลทุกชนิด เป็นการลดจำนวนการเข้าถึงอุปกรณ์

2. การใช้บล็อกบัฟเฟอร์แบบเรียงลำดับ (Sequential block buffering) ใช้ในกรณีมีการอ่านแบบเรียงลำดับกับแฟ้มข้อมูลที่มีโครงสร้าง วิธีนี้จะไม่มีกลไกในการปิดกั้น(lock) ระเบียนแต่จะทำการประมวลผลไปตลอด เป็นการลดจำนวนการเลือกจ่ายงานของกระบวนการจานแม่เหล็ก

กรรมวิธีแรก การบันทึกผ่านบัฟเฟอร์

โดยปกติกระบวนการจานแม่เหล็กจะยอมให้มีการบันทึกระเบียนทันทีโดยผ่านแคชไปยังสื่อหรืออยู่ในบัฟเฟอร์ เพื่อจะนำไปบันทึกในแคชบล็อกที่ต้องการต่อไป

การรับเข้า-ส่งออกบนจานแม่เหล็กเชิงกายภาพเป็นหน้าที่ของกระบวนการจาน-แม่เหล็ก โดยการเพิ่มทีละส่วนวงซึ่งมีขนาด 512 ไบต์ ในกรณีที่เป็นแฟ้มข้อมูลไร้โครงสร้าง และเป็นหน่วยบล็อก ถ้าเป็นแฟ้มมีโครงสร้าง โดยทั่วไปขนาดบล็อกจะเป็นทวีคูณของขนาดเซกเมนต์ซึ่งมีจำนวนระเบียนโดยขึ้นกับชนิดของโครงสร้างแฟ้ม (เรียงลำดับ หรือ สัมพัทธ์ หรือ ดัชนี) ทั้งขนาดบล็อกและขนาดระเบียน กำหนดขณะสร้างแฟ้มข้อมูล ในกรณีการบันทึกผ่านบัฟเฟอร์ไปยังแฟ้มข้อมูลมีโครงสร้าง มีขนาดเท่ากับบล็อกของแฟ้มข้อมูล และมีขนาดเท่ากับแคชบล็อกซึ่งกระบวนการจานแม่เหล็กใช้งานอยู่ กรณีแฟ้มไร้โครงสร้าง ขนาดบัฟเฟอร์จะมีขนาดเท่ากับแคชบล็อกซึ่งกำหนดช่วงในการสร้างแฟ้มข้อมูลออปชั่น BUFFERSIZE

การบันทึกข้อมูลผ่านบัฟเฟอร์จะช่วยลดจำนวนการบันทึกเชิงกายภาพ (Physical write) และทำให้เกิดการบันทึกเชิงกายภาพช้าลง จะเห็นประสิทธิภาพนี้เมื่อมีการบันทึกระเบียนเชิงตรรกะซึ่งความยาวระเบียนน้อยกว่าความยาวระเบียนเชิงกายภาพ(เช่นขนาดบล็อกหรือทวีคูณของส่วนวง)

กรรมวิธีที่สอง การใช้บล็อกบัฟเฟอร์แบบเรียงลำดับ

คือ การนำบล็อกทั้งหมดไปไว้ยังพื้นที่ที่สัมพัทธ์กับกระบวนการที่ใช้งานอยู่ วิธีนี้จะช่วยกำจัดจำนวนการเลือกจ่ายงานของกระบวนการจานแม่เหล็กและการขัดจังหวะ ซึ่งพื้นที่ดังกล่าวอาจเป็นกองซ้อนข้อมูลของกระบวนการผู้ใช้หรือเซกเมนต์แฟ้มกระบวนการ หรือที่เรียกย่อว่า พีเอฟเอส (process file segment or PFS) ขึ้นกับชนิดของเครื่องคอมพิวเตอร์

ขั้นตอนการทำงาน คือ เมื่อมีการอ่านแฟ้มก็จะทำให้บล็อกข้อมูลทั้งหมดถูกนำไปยังกองซ้อนข้อมูล หรือ พีเอฟเอส จากนั้นรหัสของคลังโปรแกรมจะทำการอ่านข้อมูลโดยเคลื่อนย้ายระเบียนจากบล็อกบัฟเฟอร์ไปยังเนื้อที่ที่กำหนดในการเรียกรูทีนการอ่าน จุดนี้จะป้องกันการเลือกจ่ายงานของกระบวนการจานแม่เหล็กและกระบวนการผู้ใช้มากเกินไป ระบบแฟ้มข้อมูลจะทำการถอดการบีบอัดข้อมูล หากมีการกำหนดในช่วงสร้างแฟ้มช่วงนี้เป็นหน้าที่ของระบบแฟ้มข้อมูลไม่ใช่กระบวนการจานแม่เหล็ก โดยทำต่อเนื่องไปจนกระทั่งพบคำสั่ง POSITION หรือKEYPOSITION ถ้ามีการใช้คำสั่ง READUPDATE ระบบแฟ้มข้อมูลจะเลือกกระบวนการจานแม่เหล็กให้ทำการนำบล็อกไปยังแคชและส่งผ่านระเบียนที่ต้องการไปยังกระบวนการผู้ใช้ จุดนี้จะไม่มีการปิดกั้นระเบียน

3.6 พาร์ทิชั่นนิ่ง (Partitioning)

แฟ้มข้อมูลต่าง ๆ สามารถถูกตัดแบ่งในเชิงกายภาพเป็นหลายแฟ้มข้อมูลไปตามจานแม่เหล็กต่างๆในระบบคอมพิวเตอร์ เรียกลักษณะนี้ว่า หลายจานแม่เหล็ก(multiple disk volume) หรือ พาร์ทิชั่นนิ่ง (partitioning) ซึ่งสามารถแบ่งได้มากสุด 16 ส่วน ถ้าเป็นแฟ้มข้อมูลประเภทเรียงลำดับและแบบสัมพัทธ์ แต่สำหรับแฟ้มข้อมูลดัชนี ตัวเลขจำนวนแฟ้มข้อมูลย่อยมีค่ามากกว่านี้ พาร์ทิชั่นสามารถทำได้ทั้งบัส (node)เดียวกันหรือต่างบัสได้ การพาร์ทิชั่นแฟ้มข้อมูลถือเป็นปัจจัยที่สำคัญมากในการเพิ่มสมรรถนะของระบบเครื่องคอมพิวเตอร์และสนับสนุนการทำงานคู่ขนานอีกด้วย

จุดหนึ่งที่ควรหลีกเลี่ยงการใช้พาร์ทิชั่น คือ การมีแฟ้มข้อมูลซึ่งมีการเข้าถึงในลักษณะต่อเนื่องเข้าท้ายแฟ้มข้อมูล หรือการเข้าถึงแฟ้มข้อมูลผ่านกุญแจที่เรียงลำดับตามเวลาไม่ว่าจะเป็นแฟ้มแบบดัชนีหรือไม่ก็ตาม เหตุเพราะการทำพาร์ทิชั่นกับการเข้าถึงแฟ้มข้อมูลลักษณะนี้จะไม่ช่วยให้เกิดการกระจายการทำงานของพาร์ทิชั่นแต่ละแฟ้มที่แยกกัน แต่ควรจะเคลื่อนย้ายกิจกรรมนี้ไปยังจานแม่เหล็กอื่น

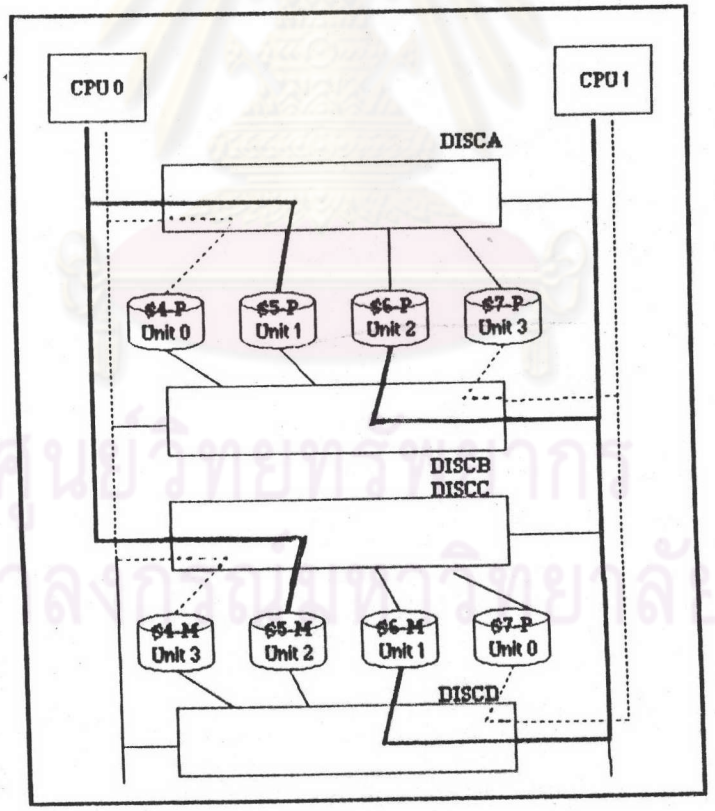
ดังได้เคยกล่าวแล้วว่าการทำพาร์ทิชั่นจะทำให้ซีพียูหลายซีพียู ช่องรับเข้า-ส่งออก ตัวควบคุม กระบวนการจานแม่เหล็กและแคช ทำงานในลักษณะภาวะพร้อมกัน (concurrency) เพื่อสนองตอบความต้องการของกระบวนการต่างๆที่เข้ามาใช้บริการ ซึ่งเป็นแนวความคิดของการทำงานแบบขนาน เพื่อให้บรรลุถึงสมรรถนะสูงสุดควรให้มีการทำพาร์ทิชั่นต่างตัวควบคุมซึ่งผูกอยู่กับซีพียูต่างกัน ยิ่งเพิ่มพาร์ทิชั่น การทำงานภาวะพร้อมกันก็ยิ่งเพิ่มขึ้นเท่านั้น การมีแคชหลายซีพียูเป็นการเพิ่มโอกาสการไม่ใช้การเข้าถึงอุปกรณ์เชิงกายภาพมากเกินไป

ผลกระทบอีกด้านหนึ่งต่อการทำพาร์ทิชั่น คือ การทำให้ระดับของดัชนีของแฟ้มข้อมูลลดลง โดยการลดจำนวนบล็อกข้อมูลซึ่งบรรจุในจานแม่เหล็กนั้น หรือแม้แต่แฟ้มข้อมูลชนิดใช้กุญแจอื่น

ก็เช่นกัน นอกจากนี้ยังมีผลต่อความจุแม่เหล็กเชิงกายภาพด้วย กล่าวคือ เนื่องจากจานแม่เหล็กมีขนาดจำกัดตามความจุที่ติดตั้งมา หากแฟ้มข้อมูลมีขนาดใหญ่เกินความจุจานแม่เหล็กก็จะมีปัญหาในการจัดการการเข้าถึงแฟ้มข้อมูล การทำพาร์ทิชั่นจะทำให้ความจุจานแม่เหล็กไม่เป็นอุปสรรค โดยการตัดแบ่งแฟ้มข้อมูลออกเป็นพาร์ทิชั่นที่สอง สาม และต่อๆไป

3.7 โครงแบบตัวควบคุมจานแม่เหล็ก (Disk Controller Configuration)

เพื่อให้การประมวลผลโดยใช้จานแม่เหล็กมีลักษณะแบบขนานทั้งการบันทึกและการอ่าน โครงแบบของตัวควบคุมจานแม่เหล็ก จึงควรจัดวางในลักษณะที่เอื้อต่อวัตถุประสงค์ดังกล่าว การบันทึกแบบขนานจะทำให้เวลาล่วงผ่าน (elapsed time) มีค่าน้อยกว่าการบันทึกแบบลำดับ ทั้งนี้ โครงแบบที่จัดว่าก่อให้เกิดสมรรถนะสูง คือ การมีตัวควบคุมจานแม่เหล็ก 4 ตัว เพราะอย่างน้อยต้องมีตัวควบคุมจานแม่เหล็ก 2 ตัวผูกกับจานแม่เหล็กที่เป็นตัวหลักกับมิรเรอร์ดิสก์ ซึ่งแต่ละตัวจะผูกกับตัวควบคุมที่แยกกันดังรูปที่ 3.4



รูปที่ 3.4 โครงแบบ 4 ตัวควบคุม

จากรูปจะทำให้ระบบมีความเชื่อถือได้ (reliability) และมีสมรรถนะสูงในขณะเดียวกัน กล่าวคือ เมื่อจานแม่เหล็กตัวใดตัวหนึ่งไม่สามารถทำงานได้ อีกตัวจะทำงานแทนได้เสมอ หรือเมื่อตัวควบคุมจานแม่เหล็กตัวใดมีปัญหา คู่ของมันที่ผูกอยู่กับจานแม่เหล็กที่มีปัญหาก็สามารถทำงานได้ตลอด ลักษณะเช่นนี้ถือว่ามีความทนต่อความผิดพร่องสองเท่า (double fault tolerant) และมีลักษณะการบันทึก (write) และการอ่าน (read) ที่เป็นแบบขนานทุกๆตัวทำให้สมรรถนะสูง มีการเปรียบเทียบการใช้โครงแบบ 2 ตัวควบคุมแบบลำดับ (serial)และแบบขนานกับ 4 ตัวควบคุม เพื่อให้เห็นคุณลักษณะที่แตกต่างกันดังตารางที่ 3.5 นี้

PERFORMANCE TUNING & RELIABILITY CONSIDERATION	2 CTLR SERIAL	2 CTLR PARAL.	4 CTLR	PERF	TUNE	RELIABILITY
Primary/Mirror on Different Controllers	No	Yes	Yes	/	/	/
Primary volumes in Different CPU'S	Yes	No	Yes	/	/	
Parallel Reads Parallel Writes	No No	Yes Yes	Yes Yes	/ /		

ตารางที่ 3.5 เปรียบเทียบโครงแบบ 2 และ 4 ตัวควบคุม

3.8 ปัจจัยจากอุปกรณ์ (Device factor)

เป็นปัจจัยที่ส่งผลต่อสมรรถนะของคอมพิวเตอร์ต่างเครื่อง เนื่องจากสมรรถนะของอุปกรณ์ต่างๆ ที่ประกอบเป็นระบบคอมพิวเตอร์มีขนาดต่างๆ ทางกายภาพในแต่ละรุ่น ใช้ข้อมูล ณ จุดนี้เพื่อประกอบการวิเคราะห์ได้อย่างดีเพราะเป็นเสมือนบรรทัดฐานในการเปรียบเทียบ และประเมินผล

คุณลักษณะของแต่ละซีพียู และอัตราส่วนสมรรถนะของซีพียูรุ่นต่างๆ ดังตารางที่ 3.6

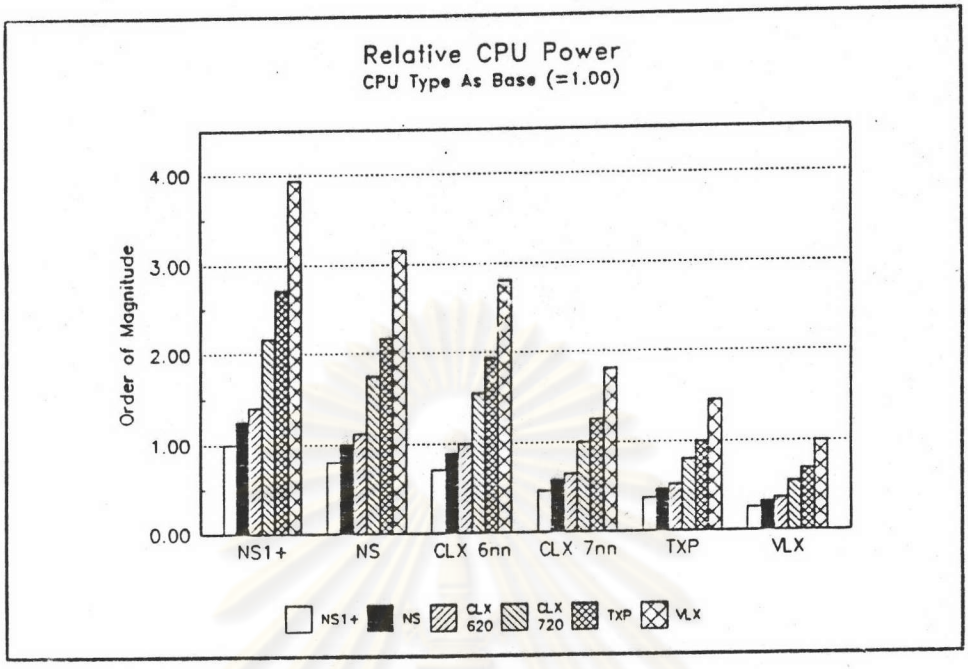
และตารางที่ 3.7 นอกจากนี้ยังสามารถเห็นภาพได้ชัดเจนขึ้นจากแผนภูมิในรูปที่ 3.6 สำหรับงานวิจัยนี้ได้ใช้ซีพียูชนิดทีเล็กซ์ที ทีเอ็นเอสทู และซีแอลเอกซ์เป็นตัวแบบ

CPU	VLX	TXP	EXT 25	CLX 720	CLX 620	NS	EXT 10	NS 1+	unit
Introduced	4/86	10/83	8/86	4/89	4/87	4/81	8/86	1976	
FOX connection	yes	yes	no	no	no	yes	no	no	
Maximum Cpu's	16	16	4	8	6	16	4	16	ea
Minimum Cpu's	2	2	2	2	1	2	2	2	ea
I/O channel	5	5	5	3.7	4	5	5	4	mb
IPB: 1 bus	20	13.3	10	20	20	13.3	10	13.3	mb
SQL ET1: 90% 2 sec. rt	14.4	9.0	9.0	7.4	5.0	3.6	3.6	na	tps
Estimated Mips	2.9	2.0	2.0	1.6	1.03	.92	.92	0.9	mips
Cycle time	83.3	83.3	83.3	na	100	100	100	100	ns
memory fetch bits	64	64	64	na	64	16	16	16	
Data path bits	32	32	32	32	32	16	16	16	
Memory cache	64	64	64	128	64	na	na	na	kb
Addr. translation memory	4	2	2	na	na	na	na	na	kb
Scratch pad memory	8	na	na	na	na	na	na	na	kb
Cache cycle	100	116	116	na	na	na	na	na	ns
DMA cycle	416	666	666	na	na	400	400	500	ns
Read 64 bits	416	666	666	na	na	na	na	na	ns
Write 64 bits	167	na	na	na	na	na	na	na	ns
Maximum Memory	96	16	16	16	12	8	8	2	mb

ตารางที่ 3.6 แสดงคุณลักษณะของซีพียู

PERFORMANCE RATIO	NS1+	NS EXT10	CLX 620	CLX 720	TXP EXT25	VLX
SCALE	1.00	1.25	1.12	1.55	1.25	1.45
Performance / NS1+	1.00	1.25	1.40	2.17	2.71	3.93
Performance / NS	0.80	1.00	1.12	1.74	2.17	3.15
Performance / CLX 620	0.71	0.89	1.00	1.55	1.94	2.81
Performance / CLX 720	0.46	0.58	0.65	1.00	1.25	1.81
Performance / TXP	0.37	0.46	0.52	0.80	1.00	1.45
Performance / VLX	0.25	0.32	0.36	0.55	0.69	1.00

ตารางที่ 3.7 แสดงอัตราส่วนสมรรถนะของซีพียูรุ่นต่างๆ



รูปที่ 3.5 แสดงแผนภูมิสมรรถนะของซีพียูรุ่นต่างๆ

อุปกรณ์อีกชิ้นหนึ่งที่สำคัญมากในการประมวลผลซึ่งจะขาดเสียไม่ได้ คือ จานแม่เหล็ก ได้แสดงคุณลักษณะของจานแม่เหล็กรุ่นต่างๆ ในตารางที่ 3.6 สำหรับงานวิจัยนี้ใช้รุ่นเอ็กซ์แอลเอทช์ (XL8)

นอกจากปัจจัยที่กล่าวแล้วข้างต้นยังมีปัจจัยด้านอื่นอีก เช่น ปัจจัยด้านการออกแบบงานประยุกต์ในลักษณะที่ไม่ก่อให้เกิดการทำงานแบบขนาน การออกแบบโครงแบบการทำงานประยุกต์ เช่น การจัดภาษาควบคุมงานหรือเจซีแอล (Job Control Language or JCL) ที่ไม่เอื้ออำนวยในการกระจายการทำงานที่มีการกระจุกตัวบนซีพียูใดซีพียูหนึ่งมากเกินไป การใช้โปรแกรมอรรถประโยชน์ที่ไม่สอดคล้องกับงาน เช่น ถ้าต้องการบรรจุข้อมูลใหม่ เพื่อให้ระดับดัชนีมีค่าน้อยกว่า 2 และทราบว่าแฟ้มข้อมูลนำเข้าเป็นแฟ้มมีดัชนีก็ควรใช้เงื่อนไขในการบรรจุข้อมูล เพื่อระบุว่าแฟ้มนั้นมีการเรียงลำดับอยู่แล้วไม่ต้องทำการเรียงข้อมูลใหม่อีกครั้งก่อนบรรจุลงแฟ้ม ตัวอย่าง คือ FUP LOAD FILEA,FILEB,SORTED โดยที่เงื่อนไข SORTED คือ สิ่งที่อธิบายแล้วข้างต้น นอกจากนี้ในการออกแบบโปรแกรมประยุกต์ควรหลีกเลี่ยงการเปิดปิดแฟ้มข้อมูลบ่อยๆ จนเกินไป เนื่องจากในการเปิดแฟ้มจะต้องใช้ตารางระบบมากมายและใช้ทรัพยากรมากด้วย ทั้งนี้จะต้องทราบคุณลักษณะของโปรแกรมอรรถประโยชน์ที่ช่วยเหลือการทำงานอย่างดี เพื่อเพิ่มประสิทธิภาพสูงสุดให้กับงาน

CAPACITY	Unit	5200 OSF	4170 XL80	4160 V80	4130 XL8	4120 V8
Formatted	mb	83,900	895	267	415	123
Unformatted	mb	na	1,130	337	520	168
Platters/Cartridges	#	32	8	6	6	6
Usable Surfaces [1]	#	64	14	10	10	10
Read and Write Heads	#	na	27	10	19	10
Bytes per Track	kb	na	40.96	40.96	36.30	20.48
Tracks per Cylinder	#	na	27	10	19	10
Cylinders	#	na	1,024	823	760	823
Density Per Inch	kbit	19.5	21.4	19.7	18.6	9.87
PERFORMANCE						
Transfer Rate [2]	mb	.3	2.4	2.4	1.86	1.20
Seek - Full	ms	300	27	40	25	40
Seek - Average	ms	200	15	20	15	20
Seek - 1 Track	ms	1	5	5	4	5
Rotations per Minute	rpm	600	3,600	3,600	3,070	3,600
Rotations per Second [3]	rps	10	60	60	51	60
Average Latency	ms	na	8.33	8.33	9.77	8.33
Start Time Maximum	sec	na	30	40	30	40
Stop Time Maximum	sec	na	15	40	20	40

ตารางที่ 3.8 คุณลักษณะของจานแม่เหล็กรุ่นต่างๆ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย