

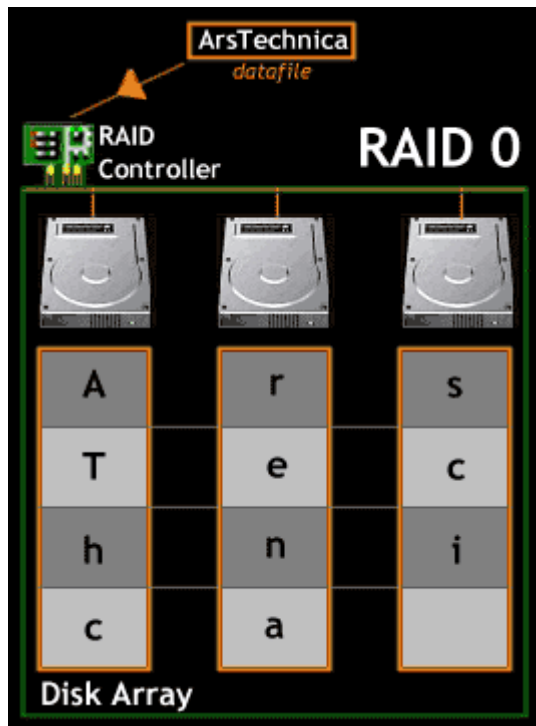
RAID เทคโนโลยีสูงสุดของการป้องกันความเสียหายของข้อมูลและความเร็วในฮาร์ดดิสก์

คำว่า RAID หรือ Redundant Array of Inexpensive Disks หรือการนำเอาฮาร์ดดิสก์หลายๆตัวมาเชื่อมต่อกันผ่านตัว Controller เพื่อจุดประสงค์ในการเพิ่มความจุ หรือความเร็วในการอ่าน/เขียนข้อมูล หลายคนทำงานเกี่ยวกับด้าน File Sever , Database Server หรือ Video, Image Editing คงจะรู้จักเป็นอย่างดี แต่สำหรับหลายๆคนที่ไม่ได้สัมผัสกับงานเหล่านี้โดยตรง อาจจะเป็นการยากเมื่อถูกถามให้อธิบายความแตกต่างระหว่างของ RAID แต่ละชนิด ไล่ตั้งแต่ RAID 0,1,3,5 และ 6 บทความนี้จะทำให้ท่านที่ไม่เคยรู้เรื่อง RAID เลยหรือรู้อะไรบ้างแต่ยังไม่ลึกซึ้งได้เข้าใจถึงการ ทำงาน, การนำไปใช้งาน และข้อดีข้อเสียของ RAID แต่ละชนิดกัน

ทำไมถึงต้องมี RAID ?

...อย่างที่เรทราบกันดีแล้วว่ายิ่งขนาดของฮาร์ดดิสก์มีขนาดมากเท่าไร ราคาของมันก็จะยิ่งมากขึ้นเท่านั้น ดังนั้นสำหรับงานที่จำเป็นต้องใช้เก็บข้อมูลจำนวนมากอย่าง File หรือ Database Server ถ้าเราเลือกใช้ฮาร์ดดิสก์ความจุมากๆเพียงตัวเดียวในการเก็บข้อมูลหรือที่เรียกกันว่าเป็นการใช้ฮาร์ดดิสก์แบบ SLED หรือ Single Large Expensive Disk ราคาที่เราเสียไปกับฮาร์ดดิสก์ตัวเดียวนั้น อาจจะไม่คุ้มค่าเท่ากับการใช้ฮาร์ดดิสก์ที่มีความจุต่ำกว่า (ซึ่งแน่นอนว่าราคาต้องถูกกว่าหลายเท่าด้วย)นำมาต่อเพื่อให้ทำงานร่วมกันหรือที่เรียกกันว่าเป็นการใช้ฮาร์ดดิสก์แบบ RAID ซึ่งนอกจากความคุ้มค่าในแง่ของราคาแล้ว ประสิทธิภาพในการทำงานแบบ RAID ยังมีมากกว่าแบบ SLED ด้วย ทั้งเรื่องของความเร็วในการเข้าถึงข้อมูล, ความน่าเชื่อถือของข้อมูล (Reliability), การบริโภคกำลังงาน และความยืดหยุ่นในการขยาย ความจุในอนาคต (Scalability) ซึ่งใน RAID แต่ละชนิดก็จะตอบสนองต่อคุณสมบัติเหล่านี้ได้ต่างกัน ในหัวข้อต่อไปจะเป็นการกล่าวถึง RAID ชนิดต่างๆ ว่ามีการทำงานอย่างไรและเหมาะกับการนำไปใช้งานด้านใดบ้าง

RAID 0 : Striped Disk Array without Fault Tolerance



รูปที่ 1

..RAID ชนิดแรกที่จะพูดถึงก็คือ RAID 0 หรือ Striped Disk Array คำว่า “Stripe” มีความหมายว่าลายขบวนพื้นผ้า ซึ่งใช้เปรียบเทียบการเก็บข้อมูลของ RAID 0 ลงจากรูปที่ 1 ประกอบข้อมูลที่เข้ามาจะถูกแตกออกที่ RAID Controller และถูกเขียนลงฮาร์ดดิสก์ที่นำมาต่อในลักษณะ “ขนาน” นี้ก็จะเกิดขึ้นเช่นเดียวกันในกรณีของการอ่านข้อมูลด้วย ดังนั้นจากรูปที่ 1 การอ่านและการเขียนข้อมูลของ RAID 0 ที่มีการนำเอาฮาร์ดดิสก์ 3 ตัวมาต่อเชื่อมกันก็จะมีความเร็วเป็น 3 เท่าของการใช้ฮาร์ดดิสก์เพียงตัวเดียว

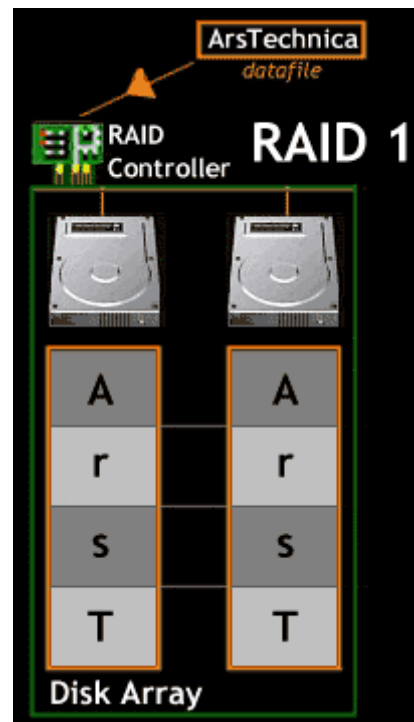
อย่างไรก็ตาม แม้ความเร็วในการอ่านและเขียนจะเพิ่มขึ้น แต่ RAID 0 ก็ยังมีข้อเสียอยู่ตรงที่ไม่มี Full Tolerance หรือกระบวนการตรวจสอบความผิดพลาดของข้อมูล ลองพิจารณา รูปที่ 1 ใหม่อีกครั้ง ถ้าฮาร์ดดิสก์ตัวใดตัวหนึ่งมีอาการเสียหายเกิดขึ้น นั่นหมายความว่าข้อมูลทั้งหมดก็จะใช้ไม่ได้ตามไปด้วย ดังนั้น RAID 0 จึงเหมาะสำหรับระบบที่ไม่ใหญ่มาก และให้ความสำคัญกับเรื่องความเร็วของการส่งผ่านข้อมูลจำนวนมาก และไม่ก่อให้เกิดความสำคัญของความผิดพลาดของข้อมูลมากเท่าไร เพราะถ้าข้อมูลเกิดผิดพลาดก็สามารถแก้ไขได้ทันที และสามารถทำการแบ็คอัพได้ทุกครั้งที่ทำงานเสร็จแล้ว

RAID 1 : Disk Mirroring

..RAID 1 มีลักษณะโครงสร้างภายในตามชื่อของมัน ก็คือจะมีฮาร์ดดิสก์ 2 ตัวที่เก็บข้อมูลเหมือนกันทุกประการ (100% Data Redundancy) เราจึงถือได้ว่าฮาร์ดดิสก์ตัวหนึ่งเสมือนเป็น “เงา” ของอีกตัวได้ และในยามที่ข้อมูลในฮาร์ดดิสก์ตัวใดตัวหนึ่งเกิดความผิดพลาดขึ้น ข้อมูลของฮาร์ดดิสก์อีกตัวก็จะถูก ก๊อปปี้ทับข้อมูลที่ผิดพลาดนั้นแทน ดังนั้น ระบบนี้จึงถือได้ว่าเป็นระบบที่มีประสิทธิภาพในการทำการตรวจจับและแก้ไขเหตุเสีย (Error Checking/Correction) สูงที่สุด เพราะข้อมูลจะเสมือนมีการแบ็คอัพไว้ตลอดเวลา

...และด้วยการที่มันต้องแบ็คอัพอยู่ตลอดเวลาเองที่ทำให้ RAID 1 มีประสิทธิภาพในการเขียนข้อมูลช้ากว่าฮาร์ดดิสก์ตัวเดียวใดๆเสียอีก อย่างไรก็ตามข้อเสียอันนี้ก็ถูกชดเชยด้วยประสิทธิภาพในการอ่านที่เพิ่มมากขึ้นกว่าฮาร์ดดิสก์ตัวเดียว 2 เท่า ทำให้เราสามารถนำ RAID 1 ไปใช้งานที่คำนึงถึงความเร็วในการอ่านมากกว่าความเร็วในการเขียนข้อมูล งานประเภทที่ว่าก็ได้แก่งาน Web Server หรือ FTP Server ระดับกลาง หรือจะนำไปใช้กับงานที่ต้องการความแน่นอนของข้อมูลสูงๆ เช่นงานด้านการเงิน การบัญชี งานจำพวกนี้ไม่ได้ต้องการส่งผ่านข้อมูลที่รวดเร็วเหมือนงานใน RAID 0 แต่ต้องการความแน่นอนของข้อมูลมากกว่า

มีข้อควรระวังข้อหนึ่งเกี่ยวกับการใช้ RAID 1 ก็คือถ้าเราใช้ซอฟต์แวร์อย่าง Windows NT/2000 เป็นตัวควบคุมการทำงาน RAID 1 หรือใช้ RAID Controller ที่ไม่รองรับคุณสมบัติ Mirroring แล้ว จะทำให้ไปหน่วยงานการทำงานของซีพียู และทำให้ประสิทธิภาพการทำงานลดลงอย่างมาก ดังนั้นควรจะใช้ฮาร์ดแวร์ RAID Controller เป็นตัวควบคุมการทำงานจะดีกว่า



รูปที่ 2

RAID 3 : Parallel Transfer with Parity

..สำหรับ RAID 3 นั้นตามโครงสร้างจะมีลักษณะการต่อฮาร์ดดิสก์เป็นแบบ Stripe เช่นเดียวกับ RAID 0 โดยข้อมูลที่เข้ามาแต่ละครั้งจะถูกแยกเก็บไว้ที่ฮาร์ดดิสก์แต่ละตัว ส่วนแต่ละตัวจะเก็บข้อมูลที่บิตหรือไบนารีที่ขึ้นอยู่ที่การกำหนด (ขอเรียกข้อมูลที่เก็บอยู่ใน A0, A1, A2, ตามรูปที่ 6 ว่าเป็น 1 word) จากรูปที่ 6 ข้อมูลที่เข้ามาก็จะถูกแบ่งย่อยเพื่อเก็บไว้ที่ A0, A1, A2, A3, B0, จนกว่าจะเก็บข้อมูลได้หมด ลักษณะการต่อฮาร์ดดิสก์ตามรูปที่ 6 เป็นการต่อที่เรียกว่า 4+1 คือมีฮาร์ดดิสก์ที่ไว้เก็บข้อมูล 4 ตัว และเก็บ Parity ที่ได้จากการคำนวณทางคณิตศาสตร์ของข้อมูลในแถวๆนั้น (ดูตามรูปที่ 6 A Parity จะเป็น Parity ของข้อมูล A0-A3) อีก 1 ตัว

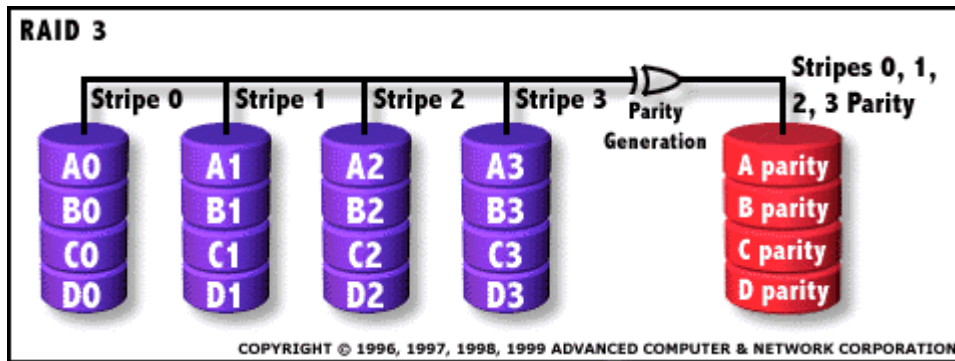
การคำนวณหา Parity นั้นจะอาศัยวิธีการทางคณิตศาสตร์ง่ายๆ คือการนำเอาลอจิก XOR เข้ามาช่วย โดยเงื่อนไขของการทำ XOR เป็นดังที่แสดงในตารางที่ 2

XOR Example		
A XOR B	Result	
0	0	0
1	0	1
0	1	1
1	1	0

ตารางที่ 2

สาเหตุที่นำลอจิก XOR มาใช้ก็เนื่องมาจากคุณสมบัติที่ว่า ถ้านำข้อมูล word A และ B มา XOR กัน ได้ผลลัพธ์ตัวหนึ่ง แล้วเกิดเหตุการณ์ที่ word A หรือ B อันใดอันหนึ่งเกิดสูญหายไป เราสามารถสร้าง word ที่หายไปอันนั้นให้กลับมาเหมือนเดิมได้ โดยการ XOR ผลลัพธ์กับ word ที่ยังเหลืออยู่ดังตัวอย่างต่อไปนี้ สมมติให้ข้อมูล word A คือ 0 1 1 0 และ word B คือ 1 1 0 0 ทั้งสอง word เมื่อนำมา XOR กันจะได้ผลลัพธ์ 1010 สมมติให้ word A เกิดการสูญหาย เราสามารถสร้าง Word A กลับมาอีกครั้งได้โดยนำผลลัพธ์มา XOR กับ word B จะได้ 0110 ซึ่งก็คือ word A นั้นเอง

จากเหตุผลข้างต้นจึงมีการนำลอจิก XOR มาสร้าง Parity ขึ้นมาโดย $A \text{ Parity} = A0 \text{ XOR } A1 \text{ XOR } A2 \text{ XOR } A3$ (ในส่วนของแถว B, C และ D ก็เป็นในทำนองเดียวกัน) และถ้าข้อมูลจากฮาร์ดดิสก์ตัวใดเกิดสูญหาย เราก็สามารถสร้าง word ข้อมูลนั้นขึ้นมาใหม่ได้ด้วยการนำ A Parity มา XOR กับ word ข้อมูลที่ยังเหลืออยู่ สมมติ word ข้อมูลเป็นดังนี้ $A0=1010, A1=0011, A2=0001$ และ $A3=1000$ ดังนั้น A Parity จะมีค่าเท่ากับ 0000 และสมมติให้ข้อมูล A2 เกิดสูญหายอันเนื่องมาจากฮาร์ดดิสก์ตัวที่ 3 เกิดปัญหาเราจะสร้างข้อมูล A2กลับคืนมาได้โดยนำ A Parity 0000 XOR กับ word ข้อมูลที่ยังเหลืออยู่ซึ่งก็คือ A0, A1 และ A3 จะได้ผลลัพธ์คือ 0001 ซึ่งก็คือ word A2 นั้นเอง

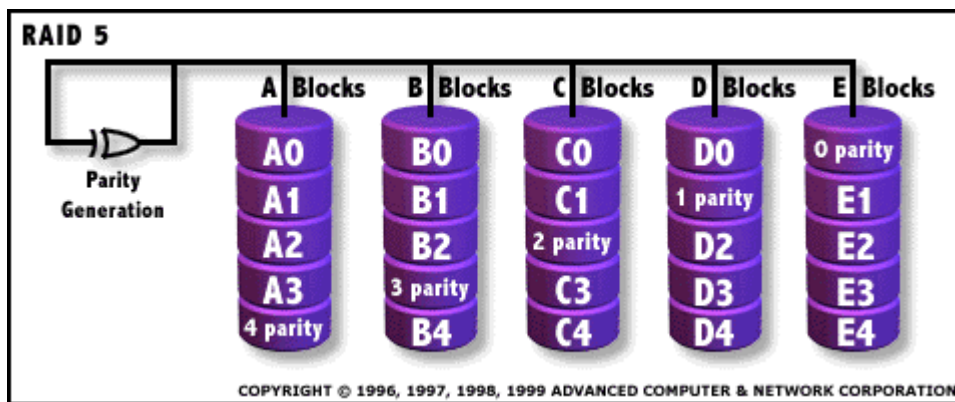


รูปที่ 6

แม้ว่า RAID 3 จะมีข้อดีในการอ่านและเขียนข้อมูลได้อย่างรวดเร็ว เพราะมีลักษณะการต่อฮาร์ดดิสก์เป็นแบบ Stripe และใช้ฮาร์ดดิสก์ในการเก็บ Parity เพียงแค่ตัวเดียวเท่านั้น แต่ถ้านำ RAID 3 ไปใช้กับงานที่ข้อมูลที่มีการผ่านเข้าออกเป็นจำนวนเล็กน้อยในแต่ละครั้ง แต่ลักษณะการเขียนข้อมูลมีการกระจายไปทั่วทั้งฮาร์ดดิสก์ จะส่งผลให้เกิดเวลาคอขวดขึ้นที่ฮาร์ดดิสก์ที่เก็บ Parity เพราะไม่ว่าข้อมูลจะไปปรากฏอยู่ในกลุ่มของ word ข้อมูลแถวไหน RAID 3 ก็จะมีการสร้าง Parity ขึ้นมาตลอด ซึ่งแม้ว่าข้อมูลเราจะมีขนาดเล็ก แต่ RAID 3 ก็จำเป็นต้องเสียเวลาไปสร้าง Parity ขึ้นมาตลอด ซึ่งแม้ว่าข้อมูลเราจะมีขนาดเล็ก แต่ RAID 3 ก็จำเป็นต้องเสียเวลาไปสร้าง Parity ขึ้นในทุกๆ แถวที่ข้อมูลนั้น ไปอยู่ ยิ่งข้อมูลกระจายไปอยู่แถวต่างๆ มากขึ้น จำนวน Parity ก็จะมากขึ้นตามและในแต่ละครั้งของการที่จะเขียนข้อมูลใหม่ลงไป RAID 3 จำเป็นจะต้องรอให้ Parity ถูกเขียนให้เสร็จก่อนที่ข้อมูล word ต่อไปจะถูกเขียน (เป็นลักษณะของการทำงานแบบ Synchronous) ยกตัวอย่าง ตามรูปที่ 6 ถ้าเราต้องการเขียนข้อมูลลงที่ word A0 ฮาร์ดดิสก์ที่ต้องใช้ในการเขียนจะมีอยู่ 2 ตัวคือฮาร์ดดิสก์ตัวที่ 1 และฮาร์ดดิสก์ตัวที่ 5 จะเห็นได้ว่าก่อนที่ word B2 จะถูกเขียนมันจะต้องรอจนกว่าฮาร์ดดิสก์ตัวที่ 5 จัดการ Parity ของ word A1 ให้เสร็จก่อน เวลาที่เกิดขึ้นนี้แหละที่เป็นเวลาคอขวด ยังมีการกระจายการเขียนข้อมูลไปหลายๆ แถวเมื่อใด ก็ยังต้องเสียเวลารอมากขึ้นเท่านั้น ฉะนั้นงานที่เหมาะสมจะเอา RAID 3 ไปใช้งานก็ควรจะเป็นงานที่ต้องการการอ่านข้อมูลจำนวนมากในเวลารวดเร็ว เพราะการอ่านจะไม่ไปยุ่งกับส่วนของ Parity ถ้าข้อมูลไม่สูญหาย เช่นงานด้านการผลิตหรือตัดต่อ Video

RAID 5 : Independent Data Disk with Distributed Parity Blocks

RAID 5 จะมีการเก็บข้อมูลเป็น Block. ข้อเสียที่สำคัญที่สุดของ RAID 3 ก็คือ เรื่องเวลาคอขวดที่เกิด เนื่องจากการเขียนส่วน Parity ดังที่ได้ยกตัวอย่างไปแล้ว สำหรับ RAID 5 จะมีการแก้ไขปัญหานี้โดยการ นำเอาส่วนของ Parity Block ไปกระจายอยู่ในฮาร์ดดิสก์แต่ละตัว ไม่แยกมาเก็บโดดๆเหมือนใน RAID 3 การกระทำเช่นนี้จะลดเวลาคอขวดได้อย่างไร จะขอเปรียบเทียบกับตัวอย่างที่ได้ยกไปในหัวข้อ RAID 3 นั่นคือการเขียนข้อมูลที่ Word A0 และ B1 ในกรณีของ RAID 5 การเขียนข้อมูลลง Word A0 จะใช้ฮาร์ดดิสก์ 2 ตัวคือ ตัวที่ 1 และ 5 ส่วน Word B1 จะใช้ตัวที่ 2 และ 4 จะเห็นได้ว่าเราไม่ต้องไปเสียเวลารอให้ RAID 5 ทำส่วนของ Parity ให้เสร็จก่อนเหมือนอย่าง RAID 3 แล้ว เพราะเราสามารถเขียนทั้ง Word A0 และ B1 ไปได้พร้อมๆกันเลย นี่จึงเป็นที่มาของการกระจายส่วนของ Parity Block ให้อยู่ในทุกๆฮาร์ดดิสก์



รูปที่ 8

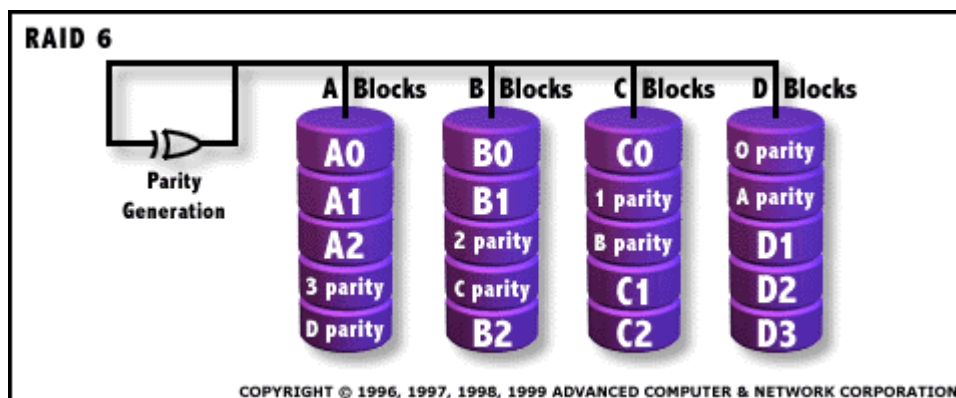
อย่างไรก็ตามการมี Parity Block อย่างใน RAID 3 หรือ 5 จะส่งผลกระทบต่อความเร็วในการเขียนข้อมูลค่อนข้างมากเช่นกัน เพราะถ้าเปรียบเทียบระหว่าง RAID 5 กับ RAID 0 จะทำได้ง่าย เพราะจะเป็นการเขียนลงฮาร์ดดิสก์เพียงตัวเดียว แต่ในกรณีของ RAID 5 จำนวน Block ที่ต่ำสุดที่จะถูกเขียนจะไม่ใช่ 1 Block แล้ว อย่างในรูปที่ 8 การเขียนข้อมูลลงใน Block B2 ข้อมูลเก่าที่อยู่ใน Block B 2 และ Parity Block 2 จะต้องถูกอ่านขึ้นมาก่อน แล้วนำมาเก็บไว้ในหน่วยความจำแคช แล้วนำข้อมูลใน Block B2 มาทำการ XOR กับ Parity ผลที่ได้ก็คือข้อมูลของ Block ที่เหลือในแถวนั้นที่ได้ถูกนำมา XOR กันแล้ว หลังจากอ่านข้อมูลเก่าเสร็จ ข้อมูลใหม่ก็จะถูกเขียนลงไปที่ Block B2 จากนั้นก็จะมีการคำนวณ Parity ใหม่อีกครั้ง โดยการนำเอาข้อมูลใหม่ไป XOR กับผลที่ได้จากการอ่านในครั้งแรกก่อน แล้วจึงจะมีการเขียน Parity ลงใน Parity Block จะเห็นได้ว่าต้องใช้กระบวนการอ่านและเขียนถึง 4 ครั้ง กับการเขียนข้อมูลเพียงบิตเดียว

ดังนั้นจึงมีการนำหน่วยความจำแคชมาใช้กับตัว Controller RAID 5 เพื่อช่วยเพิ่มความเร็วในการเขียนข้อมูลให้เร็วมากขึ้น(จาก 5-12 m sec ลดเหลือ 0.5 m sec) โดยมีขนาดประมาณ 64-256 MB การนำหน่วยความจำแคชมาใช้ส่งผลให้ประสิทธิภาพในการทำงานของ RAID 5 ดีกว่า RAID 0 เสียอีก

นอกจากนี้ RAID 5 ยังมาพร้อมกับคุณสมบัติ “Hot Swap” นั่นคือสามารถมีการสลับสับเปลี่ยนข้อมูลระหว่างฮาร์ดดิสก์ที่ดีกับฮาร์ดดิสก์ที่เสียได้ ขณะที่ระบบกำลังทำงานอยู่ สาเหตุก็เพราะ RAID 5 สามารถสร้างข้อมูลที่หายไปกลับมาใหม่ได้โดยอาศัย Parity Block ที่เก็บอยู่ในฮาร์ดดิสก์ตัวอื่นนั่นเอง สำหรับงานที่เหมาะสมกับการนำ RAID 5 ไปใช้งานก็ได้แก่งานที่ไม่ต้องการปริมาณการเขียนข้อมูลมากนัก แต่จะถูกอ่านข้อมูลมากกว่า และข้อมูลที่เก็บก็มีขนาดใหญ่มากกว่าที่จะลงทุนทำที่ฮาร์ดดิสก์เงาเพื่อแบ็คอัพข้อมูลอย่างใน RAID 1 เช่นงานจำพวก File Server, Database Server, WWW, E-mail หรือ News Server หรือ Intranet Servers

RAID 6 : Independent Data Disks with Two Independent Distributed Parity Schemes

RAID 6 นั้นจะมีพื้นฐานการทำงานเหมือนกับ RAID 5 แทบทุกประการ เพียงแต่ RAID 6 นั้นมีการเพิ่ม Fault Tolerance ให้มากขึ้นกว่า RAID 5 เท่านั้น วิธีการก็คือการเพิ่ม Parity Block เข้าไปอีก 1 ชุด ดังรูปที่ 9 เพื่อยอมให้เราสามารถทำการสลับเปลี่ยนฮาร์ดดิสก์ได้พร้อมกัน 2 ตัว ในยามที่เกิดเหตุเสียหายพร้อมกัน และด้วยการที่มี Parity Block เพิ่มขึ้นนี้จึงส่งผลให้การเขียนข้อมูลช้าขีงกว่าใน RAID 5 อีก นอกจากนั้นการเพิ่มจำนวน Parity Block ก็ยังส่งผลให้จำนวนฮาร์ดดิสก์ที่ใช้เพิ่มขึ้นอีกด้วย โดยสมมติว่าเราใช้ฮาร์ดดิสก์สำหรับเก็บข้อมูล n ตัว เราต้องจัดหาฮาร์ดดิสก์มาสำหรับการต่อแบบ RAID 6 นี้ $n+2$ ตัว ดังรูปที่ 9 ประกอบ ดังนั้นงานที่จะนำ RAID 6 ไปใช้ก็ควรจะเป็นงานประเภท Mission Critical หรืองานที่ต้องการเสถียรภาพของข้อมูลในระดับสุดยอดจริงๆถึงจะคุ้มค่าต่อการลงทุน



รูปที่ 9
