บทที่ 2

การประมวลผลแบบกริด

2.1. แนวความคิดของการประมวลผลแบบกริด

การประมวลผลแบบกริดเป็นการประมวลผลแบบกระจายที่มีจุดประสงค์เพื่อที่จะสร้างคอมพิวเตอร์เสมือนที่มีขนาดใหญ่ และมีกำลังในการประมวลผลมากโดยใช้การรวมกลุ่มของระบบที่มีความแตกต่างกันซึ่งเชื่อมต่อกันผ่านทางเครือข่ายคอมพิวเตอร์เพื่อแชร์ทรัพยากรที่มีความหลากหลาย เพื่อพัฒนาไปเรื่อยๆ จนเข้าใกล้การคำนวณแบบกระจายที่รวบรวมทรัพยากรต่างๆเพื่อให้ผู้ใช้เข้าถึงได้อย่างไม่จำกัด ในปัจจุบันการใช้การประมวลผลแบบกริดมีจุดประสงค์หลัก 2 ประการ ประการแรกคือ เพื่อการวิจัย และทดลอง ประการที่สอง คือ เพื่อพัฒนาไว้ใช้งานในองค์กร

การประมวลผลแบบกริด ทำให้สามารถใช้งานทรัพยากรของเครื่องคอมพิวเตอร์ที่มีอยู่ได้อย่างเต็มประสิทธิภาพและคุ้มค่า โดยการประมวลผลแบบกริดกำลังดำเนินรอยตาม SETI@home, Beowulf, Condor และ Unicore ในการสร้างทรัพยากรขนาดใหญ่เพื่อใช้ในการประมวลผล คือทำให้เครือข่ายกลายเป็นเครื่องคอมพิวเตอร์ขนาดยักษ์เพียงเครื่องเดียวที่ทุกคนสามารถเข้ามาใช้งานได้

การประมวลผลแบบกริดมีความคล้ายคลึงการใช้พลังงานไฟฟ้าคือเมื่อคุณเสียบปลั๊กเครื่องใช้ไฟฟ้าสิ่งที่คุณคาดหวังจะได้รับคือพลังงานที่มีค่าโวลท์ถูกต้อง แต่ไม่มีความจำเป็นต้องทราบแหล่งที่มาของพลังงานนั้น โดยบริษัทสาธารณูปโภคท้องถิ่นของคุณจะทำการเชื่อมต่อไปยังเครือข่ายของเครื่องกำเนิดไฟฟ้าและแหล่งที่มาของพลังงานตลอดจนจัดการคุณภาพการบริการเพื่อตอบสนองความต้องการพลังงานของคุณ มุมมองของการประมวลผลแบบกริดก็เหมือนกัน คือมีการจัดโครงสร้างที่เหมาะสมและผู้ใช้จะเข้าถึงคอมพิวเตอร์เสมือนที่เชื่อถือได้และเหมาะกับความต้องการของผู้ใช้ คอมพิวเตอร์เสมือนจะประกอบด้วยทรัพยากรในการประมวลผลที่แตกต่างกันจำนวนมาก แต่ผู้ใช้จะไม่สามารถมองเห็นทรัพยากรแต่ละอันเหล่านี้ เหมือนกับที่ผู้อุปโภคพลังงานไฟฟ้าจะไม่ทราบว่าพลังงานที่ใช้อยู่นั้นถูกผลิตได้อย่างไรซึ่งในการประมวลผลแบบกริดย่อมต้องมีมาตรฐานเพื่อให้มีโครงสร้างที่ปลอดภัยและทนทาน ตัวอย่างของมาตรฐานที่กล่าวถึงนี้ เช่น Open Grid Service Architecture (OGSA) และเครื่องมือที่เป็นที่นิยม เช่น Globus Toolkit ซึ่งให้ เฟรมเวิร์ค ที่จำเป็น

2.2. ทฤษฎีที่เกี่ยวข้อง

2.2.1. ซุปเปอร์คอมพิวเตอร์ (Supercomputer)

เมื่อคอมพิวเตอร์เข้ามามีบทบาทในการทำงานมากขึ้นจึงเกิดความต้องการคอมพิวเตอร์ที่มีกำลังในการประมวลผลมากๆ ซึ่งเป็นที่มาของซุปเปอร์คอมพิวเตอร์

ซุปเปอร์คอมพิวเตอร์เป็นเครื่องคอมพิวเตอร์ที่เหมาะกับงานคำนวณที่ต้องมีการคำนวณตัวเลขจำนวนหลายล้านตัวภายในเวลาอันรวดเร็ว เช่น งานพยากรณ์อากาศที่ต้องนำข้อมูลต่างๆ เกี่ยวกับอากาศทั้งระดับภาคพื้นดินและระดับชั้นบรรยากาศเพื่อดูการเคลื่อนไหวและการเปลี่ยนแปลงอากาศซึ่งจำเป็น ต้องใช้เครื่องคอมพิวเตอร์ที่มีสมรรถนะสูงมาก นอกจากนี้ยังมีงานอื่นๆอีกเป็นจำนวน มากที่ต้องใช้ซุปเปอร์คอมพิวเตอร์ที่มีความเร็วสูง เช่น งานควบคุมขีปนาวุธ, งานควบคุมทางอวกาศ, งานประมวลผลภาพทางการแพทย์, งานด้านวิทยาศาสตร์ โดยเฉพาะทางด้านเคมี เภสัชวิทยา และงานด้านวิศวกรรมการออกแบบ ซุปเปอร์คอมพิวเตอร์สามารถทำงานได้เร็วเพราะมีโครงสร้างการคำนวณพิเศษ

2.2.1.1. Multiprocessor

คือคอมพิวเตอร์หนึ่งเครื่องมีซีพียูหลายตัว ซีพียูแต่ละตัวช่วยกันทำงานซึ่งได้แก่ SMP (Symmetric Multiprocessing)คือโปรเซสเซอร์(processor)หลายตัวใช้ทรัพยากรของระบบเช่น บัส(bus), หน่วย ความจำ, อินพุท เอาท์พุท(I/O) ร่วมกันโดยไม่สามารถแบ่งแยกเป็นส่วนย่อยๆ ได้ โดย SMP มีคุณสมบัติดังนี้

มีความสามารถเทียบได้กับโปรเซสเซอร์ 2 ตัวหรือมากกว่านั้น

มีการแบ่งหน่วยความจำหลักออกเป็นส่วนๆและติดต่อกับ อินพุท เอาท์พุท ได้ง่ายด้วยการเชื่อมต่อโดยใช้ระบบบัสหรือการเชื่อมต่อปกติ ดังนั้นจึงใช้เวลาในการเข้าถึงข้อมูลเท่ากับ โปรเซสเซอร์เพียงตัวเดียว

คอมพิวเตอร์ทุกเครื่องสามารถทำงานได้ด้วยตัวเอง

ระบบทุกระบบจะทำงานร่วมกันโดยมีระบบควบคุมระบบหนึ่งที่ควบคุมการทำงานระหว่างโปรเซสเซอร์แต่ละตัว

2.2.1.2. MPP (Massively Parallel Processor)

คือ คอมพิวเตอร์ที่มีโปรเซสเซอร์หลายตัวโดยจะทำการแบ่งแยกงานให้กับตัวประมวลผลแต่ละตัว ซึ่งมีทรัพยากรอันได้แก่ อินพุท เอาท์พุท, หน่วยความจำ เป็นของตนเองและจะทำงานขนานกันไป

2.2.1.3. HPC (High Performance Computer)

คือคอมพิวเตอร์มีซีพียูและอุปกรณ์อื่นๆจำนวนมากเพื่อให้สามารถทำงานที่ต้องทำการคำนวณตัวเลขจำนวนมากๆได้เร็วขึ้น เช่น งานพยากรณ์อากาศเป็นต้น

2.2.2. เพียร์ทูเพียร์ (Peer-to-Peer ,P2P)

เพียร์ทูเพียร์ คือการประมวลผลแบบหนึ่งที่คอมพิวเตอร์แต่ละเครื่องในระบบเครือข่ายมีหน้าที่และความรับผิดชอบเท่ากัน ซึ่งต่างจากสถาปัตยกรรมแบบไคลเอนท์เซิร์ฟเวอร์(client/server) ที่จะมีเครื่องคอมพิวเตอร์เครื่องหนึ่งเรียกว่าเซิร์ฟเวอร์ ทำหน้าที่รับการร้องขอจากเครื่องคอมพิวเตอร์เครื่องอื่นที่เรียกว่าไคลเอนท์เพื่อให้บริการต่างๆ แต่ในเพียร์ทูเพียร์นั้นทุกเครื่องจะให้บริการกันเองจึงมีรูปแบบของเครือข่ายที่ง่ายกว่า ซึ่งเพียร์ทูเพียร์มีข้อจำกัดคือไม่เหมาะกับงานที่หนักมากๆ

2.2.3. คลัสเตอร์(Cluster)

คลัสเตอร์เกิดจากแนวความคิดที่จะนำกลุ่มของคอมพิวเตอร์หลายๆเครื่องที่เชื่อมต่อกันด้วยเครือข่ายมาช่วยกันทำงานที่มีขนาดใหญ่งานหนึ่ง โดยคลัสเตอร์ประกอบด้วยเครื่องคอมพิวเตอร์หลายเครื่องซึ่งแต่ละเครื่องมีซีพียูและหน่วยความจำของตัวเองเชื่อมต่อกันโดยเครือข่ายในการสื่อสาร (LAN)

เพื่อใช้ทรัพยากรร่วมกันในการทำงานที่มีขนาดใหญ่โดยเครื่องแต่ละเครื่องจะทำการประมวล ผลแยกจากกันอย่างเป็นอิสระและหน่วยความจำจะถูกแบ่งให้กับตัวประมวลผลแต่ละตัว และเครื่องแต่ละเครื่องจะติดต่อกันโดยการรับและส่งแมสเซส คลัสเตอร์ทำให้เราสามารถใช้ประโยชน์จากเครื่องที่มีอยู่ได้อย่างเต็มที่ และทำให้เราสามารถทำการประมวลผลแบบขนานโดยไม่ต้องใช้ซุปเปอร์คอมพิวเตอร์ที่มีราคาแพง และเนื่องจากในองค์กรส่วนมากมักจะมี LAN ที่มีความเร็วสูงเชื่อมต่อกับเครื่องจำนวนมากจึงอาจจะทำให้เกิดกำลังในการประมวลผลมากกว่าซุปเปอร์คอมพิวเตอร์เลยก็ได้

โครงสร้างของระบบคลัสเตอร์นั้นแบ่งออกเป็น 3 ส่วนหลักๆ คือ ฮาร์ดแวร์,ซอฟแวร์ และมิดเดิ้ลแวร์

2.2.3.1. Beowulf Cluster

Beowulf cluster คือกลุ่มของคอมพิวเตอร์ที่เชื่อมต่อกันผ่านเครือข่ายและทำการประมวลผลแบบขนานโดยBeowulf เป็นคอมพิวเตอร์แบบขนานที่มีประสิทธิภาพมากซึ่งไม่ได้เกิดจากการฮาร์ดแวร์ที่มีประสิทธิภาพสูง จะทำงานบนระบบปฏิบัติการอย่างเช่น Linux หรือ FreeBSD เชื่อมต่อโดยเครือข่ายที่มีความเร็วสูง โดยจะประกอบด้วยคลัสเตอร์ของworkstation เพื่อทำงานในการประมวลผลที่มีขนาดใหญ่

ในการใช้งาน Beowulf นั้นแอพพลิเคชั่นที่มีอยู่จะต้องถูกแตกออกเป็นงานที่ทำงานแบบขนานที่ติดต่อกันโดยใช้ MPI หรือ PVM โดยระบบจะประกอบด้วยฮาร์ดแวร์ที่เป็นเซิร์ฟเวอร์ 1 เครื่อง และ ไคลเอนท์อย่างน้อย 1 เครื่อง เชื่อมต่อกันด้วยระบบเครือข่ายภายใน เช่น Ethernet

รูปที่ 2-1 ตัวอย่างการสร้าง Beowulf Cluster ภายในบ้าน

2.2.3.2. ข้อเปรียบเทียบระหว่างกริดกับคลัสเตอร์

หัวข้อ	คลัสเตอร์	กริด
จำนวนเครื่องคอมพิวเตอร์	มากกว่า 1 เครื่อง	มากกว่า 1 เครื่อง
การทำคอมพิวเตอร์เสมือน	ทำ	ทำ
การประมวลผล	ขนาน	ขนาน
การใช้มิดเดิ้ลแวร์	ใช้	ใช้
เครือข่าย	LAN (จำกัด)	internet
จำนวนadmin	จำกัดเพียง1admin	ไม่จำกัด
จุดมุ่งหมาย	คำนวณงานขนาดใหญ่ และ สมรรถนะในการคำนวณ	กระจายงานไปยังเครื่องอื่นๆ และวิธีการแชร์ทรัพยากร
ชนิดของโปรเซสเซอร์	ชนิดเดียวกัน	แตกต่างกันได้
ชนิดของระบบปฏิบัติการ	เหมือนกัน	แตกต่างกันได้
ลักษณะของทรัพยากร	static	dynamic

ตารางที่ 2-1 ข้อเปรียบเทียบระหว่างคลัสเตอร์และกริด

2.2.4. มาตรฐานที่ใช้ในการส่งแมสเซส

เมื่อการประมวลผลแบบขนานได้รับความนิยมมากและความแตกต่างทาง สถาปัตยกรรมของเครื่องคอมพิวเตอร์จึงทำให้เกิดมาตรฐานขึ้นเพื่อให้การส่งแมสเซสมีประสิทธิภาพมากขึ้น คือ

2.2.4.1. PVM (Parallel Virtual Machine)

PVM เป็นซอฟแวร์แพคเกจ ที่ทำให้กลุ่มของเครื่องที่ต่างกันซึ่งเชื่อมต่อกันด้วยเครือข่ายที่ประกอบด้วยสถาปัตยกรรมที่ต่างกัน PVM มีจุดประสงค์เพื่อให้สภาพแวดล้อมที่ต่างกันสามารถติดต่อกันได้ สถาปัตยกรรมแบบหนึ่งสามารถถูกคัดลอกไปยังเครื่องที่มีสถาปัตยกรรมอีกแบบแล้วทำการคอมไพล์(compile) และเอกซีคิวท์(execute) ได้โดยไม่ต้องทำการดัดแปลงเพื่อให้เครื่องหลายๆเครื่องติดต่อกันเหมือนเป็นคอมพิวเตอร์แบบขนาน PVM ได้กำหนดระบบปฏิบัติการแบบกระจาย โดยจะมี เดมอน (daemon) โพรเซสทำงานบนเครื่องคอมพิวเตอร์ทั้งหมดเพื่อให้เกิด virtual machine และ PVM นี้ master จะถูกสร้างใน local machine และจะสร้างเดมอน (daemon) ในเครื่องอื่นๆโดยอัตโนมัติ แต่ใน MPI นั้น master กับ slave จะถูกสร้างขึ้นพร้อมกัน

2.2.4.2. MPI (Message Passing Interface)

MPI ถูกสร้างขึ้นเพื่อพัฒนามาตรฐานในการเขียนโปรแกรมส่งแมสเซสโดย MPI Forum MPI มีคุณสมบัติดังนี้

- MPI เป็น library เป็นมาตรฐานในการส่งแมสเซส
- ไม่ต้องทำการดัดแปลงซอสโค้ด(source code) เมื่อจะนำแอพพลิเคชั่นไปใช้กับแพลตฟอร์ม (platform) อื่นที่สนับสนุน MPI
- การอิมพลีเมนต์(implementation)ของผู้ผลิต(vendor)จะสามารถใช้ลักษณะเด่นของฮาร์ดแวร์ที่มีอยู่เพื่อให้เกิดประสิทธิภาพได้อย่างเต็มที่
- มีจำนวนฟังก์ชั่นมากกว่า 115 รูทีน(routine) และจะเพิ่มขึ้นได้โดยขึ้นอยู่กับ ประสบการณ์จากระบบการส่งข้อมูลต่างๆที่ใช้งาน MPI

2.2.5. แนวความคิดในการจัดการทรัพยากร

เมื่อมีการใช้ทรัพยากรร่วมกันจึงเกิดแนวความคิดในการจัดการในการใช้งานทรัพยากรได้แก่

2.2.5.1. Mutex (mutual exclusion object)

Mutex เป็นโปรแกรมที่ยอมให้ thread หลายๆ thread แชร์ทรัพยากรเดียวกัน thread ที่ต้องการทรัพยากรต้อง lock mutex จาก thread อื่นขณะที่มันกำลังใช้ทรัพยากร และ mutex จะ unlock เมื่อมันไม่ได้ใช้งานเป็นเวลานานหรือ thread สิ้นสุดลง ความแตกต่างระหว่าง mutex กับ semaphore คือ mutex ถูกเป็นเจ้าของโดย thread ซึ่ง lock มัน และมีเพียงโพรเซสที่ทำการ lock ที่จะสามารถ unlock ได้เท่านั้นขณะที่ semaphore สามารถถูกเปลี่ยนสถานะโดย thread หรือโพรเซสอื่นได้

2.2.5.2. Semaphore

Semaphore เป็นวิธีที่จะควบคุมการสื่อสารระหว่าง 2 โพรเซสที่ใช้ทรัพยากรร่วมกันเพื่อให้มี เพียงโพรเซสเดียวที่สามารถเข้าถึงทรัพยากรได้ในเวลาหนึ่งๆการจองทรัพยากรทำได้โดยกำหนดค่าของ

semaphore ค่านี้จะทำหน้าที่กำหนดจำนวนโพรเซสที่สามารถใช้ทรัพยากรได้ในช่วงเวลาหนึ่งๆ โดย โพรเซสจะต้องได้รับค่า semaphore ก่อนจึงจะใช้ทรัพยากรได้โดยจะทำการลดค่าของ semaphore และมันจะปล่อย semaphore เมื่อใช้ทรัพยากรเสร็จแล้วโดยจะทำการเพิ่มค่า semaphore และเพื่อที่จะได้รับทรัพยากรที่ถูกแชร์ โพรเซสจะต้องทำการทดสอบ semaphore ที่ควบคุมทรัพยากรโดย

- ถ้าค่าของ semaphore เป็นบวกแล้วโพรเซสสามารถใช้ทรัพยากรได้ และโพรเซสจะลด semaphore ลงหนึ่งเพื่อบ่งบอกว่าใช้ทรัพยากรไปแล้วหนึ่งหน่วย
- ถ้าค่าของทรัพยากรเป็นศูนย์แล้วโพรเซสจะทำการ sleep จนกว่าค่า semaphore จะเพิ่มมาก กว่าศูนย์ และเมื่อโพรเซส wake up จะกลับไปทำการตรวจสอบค่า semaphore ใหม่เมื่อโพรเซสใช้ทรัพยากรที่ถูกแชร์เสร็จแล้วค่าของ semaphore จะเพิ่มขึ้น 1 และโพรเซสที่ sleep อยู่จะถูก awake เพื่อให้ทำงานต่อได้

2.2.6. เทคโนโลยีอื่นๆในปัจจุบัน

ปัจจุบันเทคโนโลยีของซีพียูได้มีการพัฒนาไปมากจนเกิดเทคโนโลยีใหม่ๆ ได้แก่

2.2.6.1. Dual Core

สถาปัตยกรรมแบบ Dual Core คือการติดตั้ง 2 Core บนซีพียูตัวเดียวกัน เพื่อเพิ่มประสิทธิภาพการทำงานคล้ายกับ Dual CPU ซึ่งมี 2 ซีพียูในเครื่องๆเดียว แต่ Dual Core จะมีความเร็วมากกว่าเพราะสามารถส่งผลลัพธ์จาก core หนึ่งไปยังอีก core หนึ่งโดยไม่ต้องใช้system bus

2.2.6.2. Hyper-threading

เทคโนโลยี hyper-threading นี้จะใช้ hyperthread CPU ซึ่งสามารถจัดการกับชุดคำสั่งในหลายๆ thread ได้โดยไม่ต้องอาศัย system bus โดยจะมีหน่วยที่ทำหน้าที่จัดเรียงคำสั่งก่อนการประมวลผลที่ดีซึ่งทำให้สามารถใช้ execution resource ชุดเดียวใน การประมวลผลชุดคำสั่งในหลายๆ thread ไปพร้อมๆกันได้ โดยแทรกคำสั่งของแต่ละ thread ลงใน execution resource อย่างเหมาะสมโดยที่ชุดคำสั่งแต่ละชุดยังเป็นอิสระจากกัน

2.3. ประเภทของกริด

2.3.1. Computational Grid

Computational Grid จะเน้นไปที่กำลังในการประมวลผล เครื่องที่ใช้มักจะเป็นเครื่องเซิร์ฟเวอร์ที่มีประสิทธิภาพสูง

ตัวอย่างของ Computational Grid ที่รู้จักกันดีคือ SETI@home กริดชนิดนี้ประกอบด้วยคอมพิวเตอร์ที่มีกำลังต่ำ และความจุข้อมูลน้อย โดย CPU cycle ที่ว่างของเครื่องคอมพิวเตอร์ส่วนบุคคลใน

SETI@home ถูกรวมเข้าด้วยกันเพื่อสร้าง Computational Grid ที่ใช้วิเคราะห์คลื่นวิทยุที่ได้รับจากนอกโลกเพื่อค้นหาเชาว์ปัญญาจากโลกอื่น

องค์กรส่วนใหญ่ที่ใช้ Computational Grid เกิดจากมีการริเริ่มทางธุรกิจในการขยายความสามารถและเพิ่มการใช้ทรัพยากรที่มีอยู่ของคอมพิวเตอร์โดยการรวมและแชร์ทรัพยากร ในธุรกิจอาจจะต้องการความจุของคอมพิวเตอร์มากกว่าจำนวนที่มีอยู่ จึงมีธุรกิจที่สนใจในการดัดแปลงแอพพลิเคชั่นเพื่อใช้การประมวลผลแบบขนาน

การใช้ Computational Grid รวมถึงการคิดสมการทางคณิตศาสตร์, การคิดราคา, การประเมินค่า portfolio และการจำลองแบบ (โดยเฉพาะการวัดความเสี่ยง) โดยไม่ใช่ทุกอัลกอริทึมที่จะสามารถใช้การประมวลผลแบบขนานได้

2.3.2. Scavenging Grid

Scavenging Grid เกิดจากการรวมกันของเครื่อง Desktop จำนวนมาก โดยเป็นการรวม CPU cycle และทรัพยากรอื่นๆ ซึ่งจะมีเครื่องคอมพิวเตอร์ 1 เครื่องที่ทำหน้าที่ควบคุมทรัพยากรที่มีอยู่ในกริดให้กับเครื่องคอมพิวเตอร์เครื่องอื่นๆที่ทำงานร่วมกัน

2.3.3. Data Grid

Data Grid ดูแลและควบคุมการเข้าถึงข้อมูลในหลายองค์กร โดยที่ผู้ใช้ไม่จำเป็นต้องรู้ตำแหน่งที่อยู่ของข้อมูลตลอดระยะเวลาที่ใช้ข้อมูล ตัวอย่างเช่น มหาวิทยาลัยสองแห่งทำการวิจัยวิทยาศาสตร์ชีวภาพโดยใช้ข้อมูลเดียวกัน Data Grid จะช่วยในการแชร์ข้อมูล, จัดการข้อมูล, รักษาความปลอดภัยอย่างเช่น กำหนดว่าผู้ใช้แต่ละคนสามารถเข้าไปใช้ข้อมูลส่วนไหนได้บ้าง, และกำหนดการเข้าถึงข้อมูลที่น่าเชื่อถือและรวดเร็วธุรกิจที่ใช้ Data Grid มักจะเกิดจากการริเริ่มต้นทางสารสนเทศเพื่อขยายความ สามารถของเหมืองแร่ข้อมูล และเพิ่มประโยชน์ในการลงทุนโครงสร้างพื้นฐานในการเก็บข้อมูลที่มีอยู่และลดความซับซ้อนในการจัดการข้อมูล

2.4. การทำงานของกริด

2.4.1. การใช้ประโยชน์จากทรัพยากรที่ว่าง

ในองค์กรส่วนมากมักจะมีทรัพยากรส่วนหนึ่งที่ถูกใช้งานอย่างเต็มที่ ขณะที่ทรัพยากรอีกส่วนหนึ่งไม่ได้ใช้งานกริดจึงนำทรัพยากรที่ไม่ได้ใช้งานเหล่านั้นมาใช้งานเพื่อให้การทำงานมีประสิทธิภาพเพิ่มขึ้นและยังเป็นการใช้ทรัพยากรที่มีอยู่ให้เกิดประโยชน์สูงสุด

2.4.2. การทำงานแบบขนาน

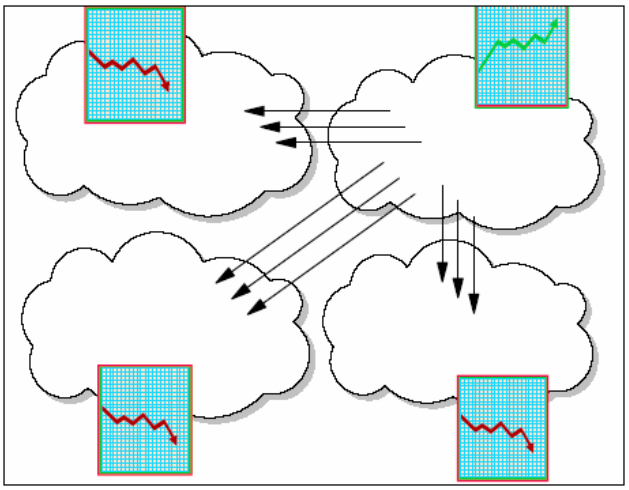
กริดจะทำการแตกงานออกเป็นหลายๆส่วนโดยแต่ละส่วนจะทำงานแยกจากกันอย่างเป็นอิสระและแยกกันไปทำงานในเครื่องที่อยู่ในที่ต่างๆกันภายในกริด แต่ก็มีงานบางประเภทที่ไม่สามารถแตกออกจากกันอย่างเป็นอิสระได้ อย่างเช่น งานที่ต้องรอผลลัพธ์จากอีกส่วนหนึ่งก่อนจึงจะสามารถทำงานต่อได้

2.4.3. การสร้างองค์กรเสมือน(Virtual Organization)

กริดได้สร้างมาตรฐานที่ทำให้ระบบที่ต่างกันสามารถทำงานด้วยกันได้เพื่อสร้างภาพของระบบการประมวลผลเสมือนขนาดใหญ่ที่มีทรัพยากรที่หลากหลาย โดยผู้ใช้กริดจะถูกจัดเข้าไปอยู่ในองค์กรเสมือนจำนวนหนึ่งซึ่งแต่ละองค์กรมีนโยบายที่ต่างกัน องค์กรเสมือนเหล่านี้สามารถแชร์ทรัพยากรร่วม กันเพื่อเป็นกริดที่มีขนาดใหญ่ขึ้นผู้มีส่วนร่วมและผู้ใช้กริดอาจจะเป็นสมาชิกขององค์กรจริงและองค์กร เสมือนที่ต่างกันได้หลายองค์กร กริดจะช่วยในการบังคับใช้กฎความปลอดภัยระหว่างองค์กรและสร้างกฎที่ช่วยจัดลำดับการใช้งานของทรัพยากรและผู้ใช้

2.4.4. การสร้างสมดุลให้กับทรัพยากร

กริดจะสร้างความสมดุลให้กับทรัพยากรโดยการจัดลำดับงานของกริดให้กับเครื่องที่ว่างโดยในช่วงที่มีงานจำนวนมาก งานบางส่วนจะถูกส่งไปยังเครื่องที่ว่างเพื่อช่วยในการประมวลผลและถ้ากริด ถูกใช้งานอย่างเต็มที่แล้ว งานที่มีลำดับความสำคัญต่ำสุดจะถูกพักไว้ชั่วคราวหรือยกเลิกไปก่อนเพื่อจะได้ทำงานที่มีลำดับความสำคัญสูงกว่าก่อน โดยการสร้างความสมดุลให้กับทรัพยากรจะช่วยลดความคับคั่งหรือระยะเวลาในการสื่อสาร



รูปที่ 2-2 งานถูกย้ายไปยังส่วนที่มีงานน้อยกว่าเพื่อสร้างความสมดุลให้กับทรัพยากร

2.4.5. การจัดการข้อมูลภายในกริด

เมื่อแอพพลิเคชั่นกริดถูกแตกออกเป็นงานย่อยแล้ว ข้อมูลอินพุทเป็นชุดข้อมูลที่มีขนาดจำกัดจึงทำให้เกิดการแชร์ข้อมูล โดยจะกำหนดที่ตั้งของข้อมูลที่แชร์ให้ใกล้กับงานที่ต้องการข้อมูลนั้นๆถ้าข้อ มูล ถูกใช้มากกว่าหนึ่งครั้งมันจะถูกทำสำเนาเก็บไว้หลายที่ในจำนวนที่มีพื้นที่เพียงพอที่จะเก็บได้

ข้อมูลที่แชร์อาจจะมีค่าคงที่หรือเปลี่ยนแปลง ตัวอย่างเช่นฐานข้อมูลอาจจะประกอบด้วยลำดับยีนที่เพิ่งพบไม่นานมานี้และกำลังเพิ่มขึ้นอยู่เสมอๆ อย่างไรก็ตามแอพพลิเคชั่นที่ใช้ข้อมูลนี้อาจจะยังไม่ต้องการข้อมูลลำดับยีนล่าสุดนี้โดยทันที ดังนั้นทำให้เป็นการง่ายขึ้นและมีประสิทธิภาพมากขึ้นในการแชร์ฐานข้อมูลในการปรับปรุงข้อมูลจะถูกรวมกลุ่มไว้และทำการประมวลผลในเวลาที่ไม่ได้ใช้งานสูง สุด นอกจากนั้นถ้ามีสำเนาข้อมูลมากกว่า 1 ชุดแล้วสำเนาไม่ต้องถูกปรับปรุงพร้อมๆกันทำให้ประสิทธิ ภาพดีขึ้นเพราะแอพพลิเคชั่นทั้งหมดที่ใช้ข้อมูลจะไม่ต้องถูกหยุดขณะที่กำลังปรับปรุงข้อมูล มีเพียงแค่การเข้าใช้สำเนาบางอันเท่านั้นที่ต้องถูกหยุดหรือพักไว้ชั่วคราว

เมื่อไฟล์หรือฐานข้อมูลถูกปรับปรุง งานจะไม่สามารถอ่านข้อมูลของไฟล์ได้ในนั้นโดยงานอื่นจะถูก lock หรือการทำไปพร้อมๆกันเพื่อป้องกันมันโดยอัตโนมัติเพื่อไม่ให้แอพพลิเคชั่นอ่านข้อมูลซึ่งจะทำให้ได้รับทั้งข้อมูลเก่าและใหม่ ในสถานการณ์ที่ข้อมูลถูกแชร์การปรับปรุงต้องไปถูกเลื่อนออกไป ตัวอย่างเช่นถ้างานย่อยกำลังประมวลผลการดำเนินงานทางการเงิน จะต้องถูกปรับปรุงโดยทันทีในฐานข้อมูลหลัก นอกจากนั้นถ้ามีสำเนาของฐานข้อมูลที่ใด จะต้องถูกปรับปรุงด้วยข้อมูลใหม่โดยทันทีจะ ต้องทำให้ข้อมูลจำนวนมากตรงกันในการสื่อสารระหว่างงานกับฐานข้อมูล ซึ่งจะทำให้เกิดปัญหา bottleneck ซึ่งส่งผลต่อประสิทธิภาพของกริดทั้งหมด จึงต้องพิจารณาถึงวิธีที่จะแบ่งกิจกรรมของฐาน ข้อมูลเพื่อให้มีการแทรกแซงน้อยในส่วนต่างๆและมีการจัดการเพื่อทำให้ข้อมูลตรงกัน

2.4.6. การจัดการเรื่องความปลอดภัยภายในกริด

เนื่องจากมีการติดต่อรวบรวมและกระจายทรัพยากรออกไป ทำให้เป้าหมายที่สำคัญอย่างหนึ่งที่กริดต้องมีคือกระบวนการจัดการด้านความปลอดภัยในระบบซึ่งบริการเหล่านั้น ซึ่งจะกล่าว ในรายละเอียดในบทที่ 4

2.5. ทรัพยากรที่ใช้ในกริด

- หน่วยประมวลผล
- หน่วยเก็บข้อมูล

  เครื่องแต่ละเครื่องในกริดมักจะกำหนดปริมาณที่สามารถใช้เก็บข้อมูลในกริดได้โดยไฟล์หรือฐานข้อมูลของเครื่องๆหนึ่งสามารถกระจายไปยังอุปกรณ์เก็บข้อมูลหรือเครื่องหลายเครื่อง

ได้และผู้ใช้สามารถอ้างถึงข้อมูลในกริดได้อย่างง่ายโดยไม่ต้องรู้ตำแหน่งที่แท้จริงของข้อมูลนั้น

- แบนด์วิธในการสื่อสารข้อมูล

  แบนด์วิธที่ใช้ในการสื่อสารเป็นทรัพยากรที่สำคัญซึ่งมีอยู่จำกัดในการใช้งานกริด เส้นทางในการสื่อสารที่มีจำนวนมากจะช่วยควบคุมความผิดพลาดของเครือข่ายและความคับคั่งของข้อมูล และจะแก้ไขปัญหา bottleneck ในการสื่อสาร

- ซอฟท์แวร์

  ซอฟท์แวร์บางชนิดมีลิขสิทธิ์และมีราคาแพงเกินกว่าที่จะติดตั้งบนเครื่องทุกเครื่องในกริดโดยในกริดนั้นงานที่ต้องใช้ซอฟท์แวร์นี้ในการทำงานจะถูกส่งไปยังเครื่องที่มีซอฟท์ แวร์นี้ติดตั้งอยู่แล้ว เพื่อลดค่าใช้จ่ายขององค์กรและนอกจากนี้ลิขสิทธิ์ซอฟท์แวร์บางตัวยอมให้ติดตั้งได้บนทุกเครื่อง ของกริดแต่จะจำกัดจำนวนเครื่องที่สามารถใช้งานพร้อมกันได้ในเวลาขณะหนึ่ง

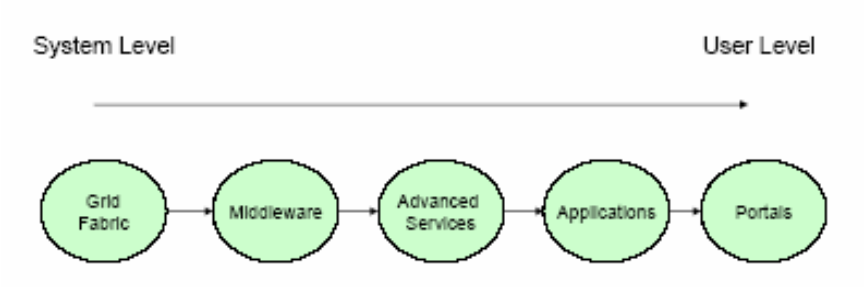
- อุปกรณ์ต่างๆ เช่น พรินเตอร์
- และอื่นๆ

2.6. สถาปัตยกรรมของกริด

กริดแสดงให้เห็นลักษณะที่แตกต่างของทางสถาปัตยกรรม ที่แสดงให้เห็นทิศทางที่ควรจะเป็น ไปของกริด และพบว่าสถาปัตยกรรมของกริดเป็นลักษณะของ Multi-faceted แบ่งได้เป็น 3 ประเภทคือ

2.6.1. สถาปัตยกรรมกริดแบบลำดับชั้น(N-Tier Grid Architecture)

สถาปัตยกรรมกริดแบบลำดับชั้น เป็นลักษณะของแบบจำลองเพื่อให้การพัฒนากริดได้รับความยืดหยุ่นและสามารถ นำแอพพลิเคชั่นที่เคยสร้างไว้กลับมาพัฒนา และใช้งานซ้ำได้การแบ่งแยกกริดออกเป็นชั้นๆ ทำให้ผู้พัฒนาสามารถทำการเปลี่ยนแปลงและเพิ่มเลเยอร์ที่ต้องการได้ จึงทำให้นักพัฒนาเลือกเพิ่มเติมหรือแก้ไขได้อย่างจำเพาะเจาะจงแทนที่จะต้องทำการสร้างแอพพลิเคชั่นขึ้นมาใหม่ ซึ่งเปรียบได้กับชั้นที่ 7 ของ OSI model จึงเป็นที่นิยมใช้กันโดยทั่วไป

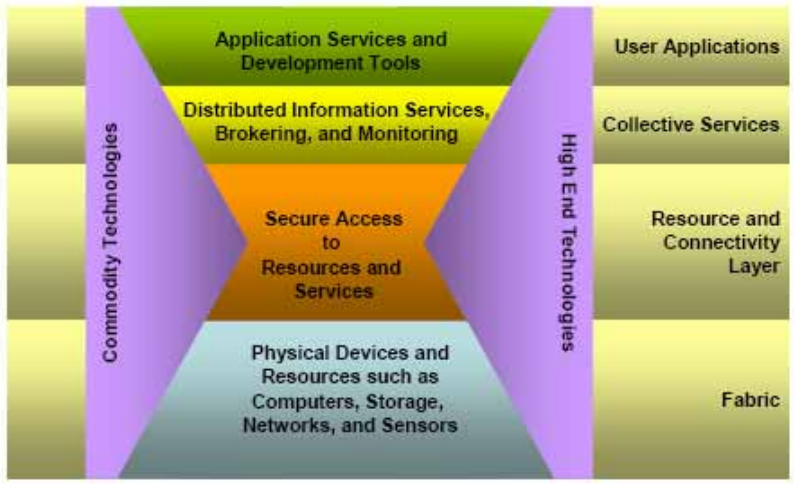


รูปที่ 2-3 รูปสถาปัตยกรรมกริดแบบลำดับชั้น

2.6.2. Role-Based Grid Architecture (สถาปัตยกรรมกริดที่อ้างอิงกับหน้าที่)

การเข้าถึงกลุ่มของทรัพยากรทางกายภาพได้อย่างปลอดภัย จะต้องใช้สถาปัตยกรรมแบบที่อ้าง อิงหน้าที่ ซึ่งสถาปัตยกรรมแบบนี้ง่ายที่จะบ่งบอกส่วนประกอบต่างๆของระบบที่ทำการติดต่อกับภาย นอกซึ่งเปรียบเสมือนเป็นโปรโตคอลที่แยกออกเป็นประเภทๆตามหน้าที่โดยร่วมใช้ข้อมูลชุดเดียวกัน

สถาปัตยกรรมแบบนี้แบ่งออกได้เป็น 5 ชั้น คือ fabric, connectivity, resource, collective, and application layer และใช้มาตรฐานเหล่านี้เพื่อช่วยให้การแลกเปลี่ยนข้อมูลเป็นไปด้วยความปลอดภัย และเสมือนเป็นทรัพยากรจากแหล่งเดียวกันทรัพยากรเหล่านี้ถูกควบคุมและดูแลโดย collective services ซึ่งทำให้มองเสมือนทรัพยากรเหล่านั้นคือทรัพยากรจากแหล่งเดียวกันเพื่อให้ง่ายต่อการออกแบบแอพพลิเคชั่นและง่ายสำหรับการเรียกใช้ของผู้ใช้งาน



รูปภาพที่ 2-4 รูปสถาปัตยกรรมกริดที่อ้างอิงกับหน้าที่

ในสถาปัตยกรรมแบบที่ขึ้นกับหน้าที่แบ่งเป็นเลเยอร์ต่างๆดังนี้

- Fabric layer

ประกอบไปด้วยโปรโตคอลต่างๆที่ทำการติดต่อกับ แอพพลิเคชั่นและเครื่องมือที่ทำให้เกิดการพัฒนาการบริการต่างๆ รวมไปถึงควบคุมส่วนต่างๆที่ทำการเข้าถึงทรัพยากร เช่น เครื่องคอมพิวเตอร์ , แหล่งเก็บทรัพยากร , ระบบเครือข่าย และ เซ็นเซอร์ เป็นต้น โดยควรจะมีการกำหนดกลไก enquiry ซึ่งจะหาโครงสร้างและสถานะ รวมถึงกลไกการจัดการทรัพยากรซึ่งควบคุมคุณภาพของการให้บริการ(quality of service)

- Connectivity layer

  เป็นส่วนที่รับผิดชอบหน้าที่ในการติดต่อ และ รับรองการเชื่อมต่อของเครือข่าย เพื่อทำให้เกิดความปลอดภัยในระหว่างที่ทำการติดต่อกับทรัพยากรของกริด เป็นโปรโตคอลและบริการที่มีหน้าที่แลก เปลี่ยนข้อความที่มีความปลอดภัย , การรับรองสิทธิ ,และการให้อำนาจในการติดต่อ

- Resource layer

  ประกอบไปด้วยโปรโตคอลที่ทำหน้าที่ดูแลและสร้างความปลอดภัยให้กับทรัพยากร โปรโตคอล Resource layer แบ่งออกเป็น

  - Information protocols (โปรโตคอลข้อมูล)

    ถูกใช้เพื่อรับข้อมูลเกี่ยวกับโครงสร้างและสถานะของทรัพยากร เช่นการ configuration, load ปัจจุบัน และกฎการใช้งาน

  - Management protocols (โปรโตคอลในการจัดการ)

    ถูกใช้เพื่อควบคุมการเข้าถึงทรัพยากรที่ถูกแชร์ เช่น การร้องขอทรัพยากร (รวมถึงการจองล่วงหน้าและการควบคุมคุณภาพของการให้บริการ) และการทำ operation อย่างเช่น การสร้างโพรเซสหรือการเข้าถึงข้อมูล

- Collective layer

  รับผิดชอบในส่วนของการทำงานร่วมกันของทรัพยากรจากหลายแหล่ง และบ่งบอกว่าอยู่ส่วนไหนในองค์กรเสมือน และสามารถกำหนดพฤติกรรมในการแชร์ได้

- Application layer

  ทำหน้าที่ติดต่อกับผู้ใช้โดยตรงโดยผ่านแอพพลิเคชั่นซึ่งจะติดต่อผ่านทางองค์กรเสมือนอีกทีหนึ่ง

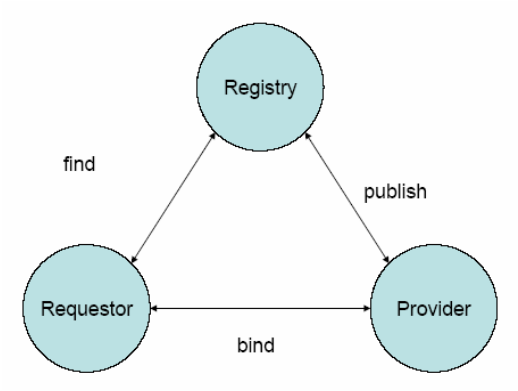
2.6.3. Service-Based Grid Architecture (สถาปัตยกรรมกริดที่อ้างอิงกับเซอร์วิส)

ปัจจุบันเราพบว่าแนวทางของเทคโนโลยีจะเน้นไปทางแนวคิดที่แต่ละส่วนของบริการใดๆเป็นอิสระต่อกัน และภาพรวมของกริดพบว่าลักษณะซอฟท์แวร์ที่เราต้องการคือไม่ขึ้นอยู่กับแพลตฟอร์มใดๆ

บริการที่เราต้องการจะสามารถเรียกถึงได้ง่ายซึ่งกระบวนการนี้เรียกว่า การค้นหารีซอร์ส (resource discovery)

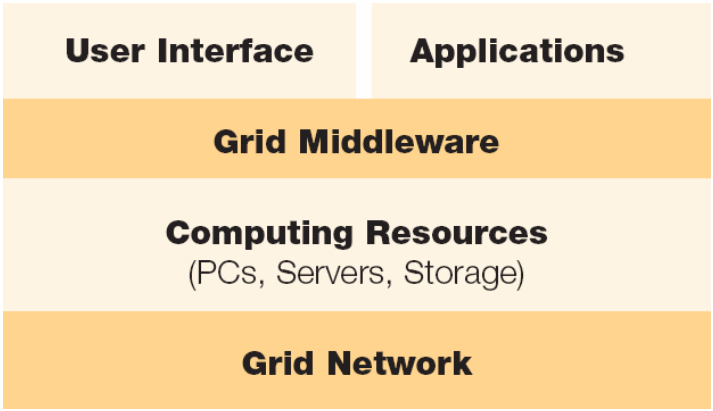
ประโยชน์ของสถาปัตยกรรมกริดที่ขึ้นกับเซอร์วิส เปรียบได้กับกลุ่มของคอมพิวเตอร์ที่ทำงานตามรูปแบบตารางเวลา เริ่มจาก หากลุ่มรีซอร์สที่สามารถใช้งานได้ ต่อมาก็เลือกรีซอร์สที่ใช้คำนวณจากกลุ่มที่ได้ในขั้นแรกเพื่อจัดตารางงาน โดยหลักการเลือกรีซอร์สนั้น อาจเลือกจากอัตราการโหลดของแต่ละรีซอร์ส หรือ จากค่าต่างๆ ตามแต่จะพิจารณาว่าสมควร

โดยสรุปแล้วโครงสร้างนี้สามารถพัฒนาเป็นการทำงานที่เชื่อมต่อกันระหว่างเซอร์วิสได้ เพราะรูปแบบที่ขึ้นกับเซอร์วิสนี้ เป็นเซอร์วิสที่ไม่ได้เกิดขึ้นในเวลาเดียวกันตามรูปแบบ asynchronous service นั่นเอง



รูปที่ 2-5 รูปสถาปัตยกรรมกริดที่อ้างอิงกับเซอร์วิส

2.7. โครงสร้างของระบบการประมวลผลแบบกริด



รูปที่ 2-6 โครงสร้างของระบบการประมวลผลแบบกริด

โครงสร้างของกริดนั้น จะประกอบไปด้วย

2.7.1. User Interface & Application

เป็นแอพพลิเคชั่นที่ติดต่อกับผู้ใช้งานระบบกริด เช่น ส่วนที่ใช้จัดการควบคุมดูแลกริด ส่วนที่ใช้จัดหาเซอร์วิส เป็นต้น

2.7.2. Grid Middleware

เป็นเครื่องมือที่ช่วยในการสร้างระบบกริด เปรียบเสมือนส่วนที่เชื่อมต่อระหว่างผู้ใช้งานกับทรัพยากรที่ใช้งานในกริด มีหลายเครื่องมือที่ใช้เป็นมิดเดิ้ลแวร์ เช่น Globus Toolkit, Rocks Cluster & Roll Grid เป็นต้น

2.7.3. Computing Resources

เป็นทรัพยากรที่ใช้ในการคำนวณ ประกอบไปด้วย คอมพิวเตอร์ แหล่งเก็บข้อมูล แหล่งประมวลผลต่างๆ เป็นต้น

2.7.4. Grid Network

ระบบเครือข่ายที่ใช้ในกริด ซึ่งรวมถึงระบบเครือข่ายภายใน (Local Area Network) และรวมถึงระบบเครือข่ายภายนอก (Wide Area Network)

ส่วนที่สำคัญที่สุดที่จะต้องกล่าวถึงเป็นพิเศษก็คือ มิดเดิ้ลแวร์ เนื่องจากเป็นเสมือนส่วนที่ทำให้ระบบกริด สามารถใช้งานได้จริง ซึ่งมิดเดิ้ลแวร์ที่นิยมใช้ คือ Globus Toolkit ซึ่งมีข้อดีก็คือ ได้นำเทคโนโลยีเว็บเซอร์วิสมาผนวกรวมเข้ากับมาตรฐานที่ใช้กำหนดระบบกริด (Open Grid Service Architecture หรือ OGSA) และยังเป็นเครื่องมือที่ผู้คิดค้นกริดที่รวมตัวเป็นกลุ่มๆหนึ่ง ที่เรียกว่า Global Grid Forum หรือ GFF พัฒนาขึ้นมาโดยตรงด้วย

ข้อดีของการนำกริดมาอ้างอิงบนพื้นฐานของ เว็บเซอร์วิส คือจะทำให้ได้เฟรมเวิร์ค ที่สามารถขยายได้เพื่อการเข้าถึง ทรัพยากรของกริด โดยใช้มาตรฐานต่างๆที่มีอยู่แล้ว เช่น SOAP, XML และ WS-Security, TCP/IP และ มาตรฐาน IETF และอื่นๆ ทำให้สามารถติดต่อสื่อสารแบบ interoperable ระหว่างระบบที่แตกต่างกันได้ในอินเตอร์เน็ตและเครือข่ายที่ใช้

2.8. ลักษณะของแอพพลิเคชั่นที่เหมาะแก่การใช้งานกริด

ถึงแม้ว่าการใช้งานกริดจะเกิดประโยชน์อย่างมาก แต่ก็มีแอพพลิเคชั่นบางชนิดที่ไม่เหมาะแก่การใช้งานกริดเนื่องจากมันใช้ทรัพยากรในการประมวลผลจำนวนไม่มาก ซึ่งถ้าใช้กริดในการทำงานอาจจะเป็นการลดประสิทธิภาพในการทำงานของมันเนื่องจากต้องเสียเวลาในการส่งข้อมูลผ่านทางเครือ ข่ายและควรพิจารณาว่างานนั้นเป็นการคำนวณที่สามารถทำงานในแบบขนานได้หรือไม่ อย่างเช่น ในการประมวลผลที่เกี่ยวกับการคำนวณบางอย่างอาจจะไม่ต้องทำตามลำดับก่อนหลัง จึงสามารถทำงานแบบขนานได้แต่ก็มีงาน

บางประเภทที่ต้องรอผลลัพธ์จากอีกส่วนหนึ่งก่อนจึงจะสามารถทำการประมวล ผลต่อได้ซึ่งไม่เหมาะกับการใช้งานกริด

2.9. ตัวอย่างของงานที่ใช้การประมวลผลแบบกริด

2.9.1. SETI@Home หรือ The Search for Extraterrestrial Intelligence

SETI@Home เป็นโครงการของ University of California at Berkeley โดยที่ UC. Berkeley มีโครงการ SETI ที่ค้นหาอารยธรรมนอกโลกมานานแล้วโดยใช้กล้องโทรทรรศน์วิทยุเฝ้าฟังสัญญาณเพื่อนำมาวิเคราะห์สัญญาณซึ่งจำเป็นต้องใช้เครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูงมากๆระดับซุปเปอร์คอมพิวเตอร์ แต่เนื่องจากซุปเปอร์คอมพิวเตอร์มีราคาแพงมาก ทีม SETI ของ UC. Berkeley จึงเกิดแนวความคิดที่จะนำเครื่อง PC ทั่วไปซึ่งมักจะไม่ได้ใช้ CPU time เต็ม 100% มาร่วมกันประมวลผลข้อมูล จึงเกิดโครงการ SETI@Home ขึ้น โดย SETI@Home ใช้หลักการของการประมวลผลแบบกระจาย (distributed computing) แบบไคลแอนท์เซิร์ฟเวอร์ (client/server) โดยจะมีโปรแกรมที่เป็นไคลเอนท์ ของ SETI@Home ติดตั้งไว้ที่เครื่อง PC และตัวโปรแกรมนี้จะติดต่อกับเซิร์ฟเวอร์ ที่ UC. Berkeley เพื่อรับงานไปประมวลผลและส่งผลลัพธ์กลับมา SETI@Home ให้อาสาสมัครจากทั่วโลกที่มีคอมพิวเตอร์ต่อกับอินเตอร์เน็ตสามารถดาว์นโหลดโปรแกรมไคลแอนท์ของ SETI@Home ไปติดตั้งได้ สำหรับ Windows และ Mac โปรแกรมไคลเอนท์ จะทำหน้าที่แทน screensaver คือจะประมวลผลเมื่อผู้ใช้ไม่ได้ใช้เครื่องอยู่ ส่วนบน platform อื่นไคลแอนท์จะทำงานเป็น background mode หรือ ควาสำคัญในระดับต่ำ (low priority) เพื่อไม่ให้รบกวนงานอื่นๆที่กำลังทำอยู่ คือจะไม่รบกวนงานปกติของผู้ใช้ปัจจุบัน SETI@Home มีกำลังในการประมวลผลอยู่ที่ประมาณ 180 TFLOPs/sec โดยมีการเขียนซอฟท์แวร์สำหรับทำงานแบบ ออฟไลน์ (offline) ได้ด้วย จึงสามารถทำงานได้แม้จะไม่ได้ต่อกับอินเตอร์เน็ตโดยเครื่องจะทำการส่งผลลัพธ์ที่เสร็จแล้วกลับไปเมื่อเราทำการเชื่อมต่ออินเตอร์เน็ตในภายหลัง

รูปที่ 2-7 Arecibo Radio Scope ที่ประเทศ Puerto Rico

ใช้สำหรับการรับฟังสัญญาณของ SETI@Home

2.9.2. Teragrid

Teragrid เป็นโครงการของมูลนิธิวิทยาศาสตร์แห่งชาติ (National Science Foundation หรือ NSF) ของสหรัฐอเมริกา เพื่อใช้สำหรับการใช้งานเพื่อการค้นคว้าในหลายๆสาขา โปรแกรมนี้มีกำลังในการประมวลผล 40 TeraFLOPs/s และมีอุปกรณ์จัดเก็บที่มีขนาดประมาณ 2 Petabyte ครอบคลุมทั่วทั้งศูนย์คอมพิวเตอร์ของมหาวิทยาลัยอิลลินอยส์, มหาวิทยาลัยซานดิเอโก, ห้องปฏิบัติการแห่งชาติอาร์กอนและแคลเทค (Argonne National Labs and Cal Tech) เชื่อมต่อกันโดยผ่านท่อ fiber optic ของ Qwest ที่มีความเร็วในการเชื่อมต่ออยู่ที่ 40 Gbps