สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รายงานการวิจัยฉบับสมบูรณ์

เรื่อง

โครงการคอมพิวเตอร์ประสิทธิภาพสูง

(High Performance Computer)

จัดทำโดย

นายนพรัตน์ พันธุ์เสนา

รตท. สมชาติ เลิกบางพลัด

นายกิตติ อนุศิษฏ์วิวัฒน์

RCH
OA
76.58
เลขหมู่ 61845
เลขทะเบียน 79733
วัน,เดือน,ปี 11 มิ.ย. 2551

ฝ่ายควบคุมเครื่อง

สำนักวิจัยและบริการคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

b 11903657x
i

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ 4

Abstract 4

บทที่ 1 บทนำ 5

1.1 บทนำ 5

บทที่ 2 แนะนำระบบและการออกแบบระบบ 6

2.1 บทนำ 6

2.2 ระบบการประมวลผลแบบพีซีคลัสเตอร์ (PC Cluster) 6

2.3 ระบบกริดคอมพิวเตอร์ 8

2.4 กริดมิดเดิ้ลแวร์ (Grid Middleware) 10

2.5 การรักษาความปลอดภัยภายในกริด 12

2.6 ซอฟแวร์จัดการลำดับงาน 18

2.7 การออกแบบระบบ 22

2.8 บทสรุป 22

บทที่ 3 การติดตั้งระบบ 23

3.1 บทนำ 23

3.2 การติดตั้ง Globus Toolkit 23

3.3 การติดตั้ง Simple CA 25

3.4 การขอและการติดตั้ง User Certificate 25

3.5 การขอและการติดตั้ง Host Certificate 28

3.6 การมอบหมายสิทธิ (Authorization) 30

3.7 บทสรุป 31

บทที่ 4 การใช้งานระบบกริดคอมพิวเตอร์ 32

4.1 บทนำ 32

4.2 การสั่งงานผ่านทางซอฟแวร์จัดลำดับงาน 32

4.2.1 งานแบบลำดับ (Sequential Job) 32

4.2.2 งานแบบขนาน (Parallel Job) 33

4.3 การสั่งงานกริดผ่านทางเครื่องมือของ Globus 35

4.3.1 การสั่งงานกริดแบบ Interactive และ Batch 35

4.3.2 การสั่งงาน Grid ผ่านทางตัวจัดการลำดับงาน 38

4.3.3 การรับ-ส่งไฟล์ผ่านทาง GridFTP และ Remote Login ผ่าน GSI-SSH 41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 การสั่งงานผ่านทาง Web Service 42
4.3.5 การสั่งงานผ่านทาง Grid Portal 42
4.4 บทสรุป 43
บทที่ 5 บทสรุป 44
5.1 บทนำ 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง โครงการคอมพิวเตอร์ประสิทธิภาพสูง
(High Performance Computer)

โดย นายนพรัตน์ พันธุ์เสนา หัวหน้าโครงการวิจัย
รตท. สมชาติ เลิกบางพลัด
นายกิตติ อนุศิษฎ์วิวัฒน์
สำนักวิจัยและบริการคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทคัดย่อ

โครงการวิจัยนี้ได้ทำการศึกษาและพัฒนาระบบการให้บริการเครื่องแม่ข่ายเพื่อการคำนวณประสิทธิภาพสูง ในรูปแบบของเทคโนโลยีกริดคอมพิวเตอร์ โดยจะต้องทำการติดตั้งระบบปฏิบัติการ Rocks Cluster Linux และกำหนดค่าต่าง ๆ ให้เหมาะสม ทำการทดสอบการทำงานเบื้องต้น ทำการเชื่อมต่อกับมหาวิทยาลัยต่าง ๆ และทำการทดสอบการประมวลผลงานบนระบบที่ได้สร้างขึ้นมา

ผลสำเร็จของโครงการวิจัยนี้ทำให้เกิดการร่วมมือกับมหาวิทยาลัยต่าง ๆ ในประเทศไทย เพื่อทำการแบ่งปันทรัพยากรการประมวลผลงานในรูปแบบของกริดคอมพิวเตอร์ อีกทั้งเป็นการเตรียมการเครื่องแม่ข่ายเพื่อการคำนวณสำหรับอาจารย์ นักวิจัย และนักศึกษาในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังอีกด้วย

Abstract

This research project is attended to build the computational resources for KMITL's researcher in term of Grid computer technology. Rocks Cluster, Linux distribution for PC Cluster and Grid Computer, are selected to install on our systems. The most important part of system is grid middleware. Globus Toolkit is selected for this task.

The outcome of this project results in collaboration among the universities in Thailand to share the computational resources for academic research. In addition, this project is provided the high performance computer to lecturers, researchers and students in King Mongkut's Institute of Technology Ladkrabang.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1 บทนำ

1.1 บทนำ

ในรายงานการวิจัยเล่มนี้จะได้แสดงให้เห็นถึง รูปแบบของการประมวลผลบนลีนุกส์คลัสเตอร์และกริดคอมพิวเตอร์ ซึ่งเป็นการขยายขอบเขตประสิทธิภาพการคำนวณให้มากขึ้นได้อย่างไม่จำกัด โดยปัจจัยสำคัญที่ทำให้เกิดความร่วมมือจากหลาย ๆ องค์กรในการที่จะแบ่งปัน และร่วมกันใช้ทรัพยากรในรูปแบบของกริดคอมพิวเตอร์นั้น คือความต้องการระบบคอมพิวเตอร์ที่สามารถประมวลผลงานต่าง ๆ ให้แล้วเสร็จภายในเวลาอันรวดเร็ว ทันกับความต้องการและความเจริญก้าวหน้าของโลกเทคโนโลยีในปัจจุบัน และการที่จะทำให้เกิดความร่วมมือต่าง ๆ ได้นั้น จะต้องมีตัวเชื่อมประสาน ระบบต่าง ๆ ที่หลากหลายรูปแบบ และหลากหลายหน่วยงานเข้าด้วยกันได้ ดังนั้นจึงต้องมีซอฟแวร์ที่ทำหน้าที่เป็นตัวกลางในการทำงานนี้ หรือที่เรียกว่า กริดมิดเดิ้ลแวร์ (Grid Middleware) ซึ่งในปัจจุบันกริดมิดเดิ้ลแวร์ที่ได้ความนิยมเป็นอย่างมาก คือ โกลบัส ทูลคิท (Globus Toolkit) ซึ่งก็จะถูกนำมาใช้ในโครงการนี้ด้วย

องค์ประกอบของกริดคอมพิวเตอร์ ประกอบด้วยหลายส่วนด้วยกัน ไม่ว่าจะเป็น เครื่องคอมพิวเตอร์ ระบบคลัสเตอร์ กริดมิดเดิ้ลแวร์ โปรแกรมประยุกต์ โปรแกรมของผู้ใช้งาน และผู้ใช้งาน ดังนั้นจึงต้องมีการกำหนดมาตรฐานต่าง ๆ ไว้มากพอสมควร เช่น การเข้าสู่ระบบ การรักษาความปลอดภัย การเข้าถึงบริการต่าง ๆ ภายในกริด ซึ่งสิ่งเหล่านี้จำเป็นต้องเป็นไปในทิศทางเดียวกัน และภายในกริดมิดเดิ้ลแวร์เองก็จะต้องสามารถรองรับมาตรฐานต่าง ๆ ที่กล่าวมาทั้งหมด และในรายงานเล่มนี้ก็ได้มีการกล่าวถึงรูปแบบการรักษาความปลอดภัยของระบบ และข้อมูล การพิสูจน์สิทธิ และการพิสูจน์ตัวตนต่าง ๆ ไว้เป็นการเบื้องต้น

หลังจากนั้นภายในรายงานจะได้กล่าวถึง ขั้นตอนการติดตั้งระบบ และการปรับค่า Configuration ต่าง ๆ เพื่อให้ระบบสามารถทำงานได้อย่างเหมาะสม และทำงานร่วมกับคลัสเตอร์จากองค์กรอื่น ๆ ที่อยู่ภายใต้กริดเดียวกันได้ โดยประเด็นสำคัญคือ ใบรับรองดิจิตอล (Digital Certificate) ซึ่งเป็นสิ่งสำคัญมากในการพิสูจน์ตัวตน และการรักษาความปลอดภัยภายกริด

วิธีการสั่งงานกริดคอมพิวเตอร์ หรือการส่งงานเข้าไปประมวลผลบนกริดคอมพิวเตอร์ เป็นอีกประเด็นหนึ่งที่มีความสำคัญมาก และได้แสดงขั้นตอนโดยละเอียดไว้ในรายงานเล่มนี้ด้วย ในการสั่งงานบนกริดนั้น สามารถที่จะสั่งประมวลผลงานนั้น ๆ ได้โดยตรง หรืออีกทางหนึ่ง สามารถที่จะสั่งประมวลผลผ่านตัวจัดการลำดับงาน (Job Scheduler) ได้ด้วย ซึ่งวิธีการหลังจะส่งผลให้ระบบกริดโดยรวมสามารถทำงานได้ดีกว่า เกิดการกระจายการทำงานไปยังเครื่องต่าง ๆ ที่เหมาะสมอย่างแท้จริง

จากรายละเอียดที่ได้กล่าวมาทั้งหมด น่าจะทำให้รายงานการวิจัยฉบับนี้มีความสมบูรณ์ในตัวเอง และทำให้เกิดความรู้ ความเข้าใจ อีกทั้งความชำนาญในการจัดการ และการสั่งงานกริดคอมพิวเตอร์มากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 แนะนำระบบและการออกแบบระบบ

2.1 บทนำ

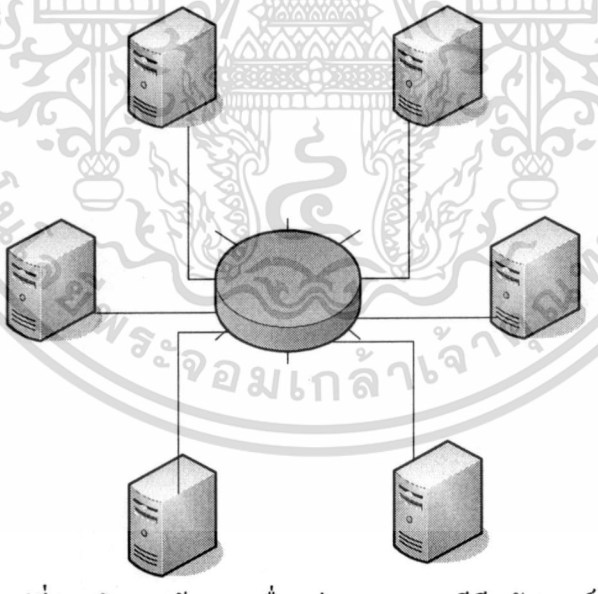
พีซีคลัสเตอร์และกริดคอมพิวเตอร์เป็นเทคโนโลยี่ที่ได้ความสนใจเป็นอย่างมาก และมีการพัฒนาอย่างต่อเนื่องในปัจจุบัน งานทางด้านวิทยาศาสตร์การคำนวณหลาย ๆ ต้องการความสามารถในการประมวลผลที่สูงขึ้นทำให้พีซีคลัสเตอร์และกริดคอมพิวเตอร์มีบทบาทในงานวิจัยมากขึ้นเช่นกัน

ในบทนี้จะได้ทำการกล่าวแนะนำถึงเครื่องคอมพิวเตอร์ประสิทธิภาพสูงแบบพีซีคลัสเตอร์ กริดคอมพิวเตอร์ กริดมิดเดิ้ลแวร์ ซอฟแวร์จัดลำดับงาน และการออกแบบระบบ ซึ่งภายในบทนี้จะเป็นการปูพื้นฐานเพื่อให้เกิดความเข้าใจในระบบต่าง ๆ ที่กล่าวมา

2.2 ระบบการประมวลผลแบบพีซีคลัสเตอร์ (PC Cluster)

ระบบการประมวลผลแบบพีซีคลัสเตอร์นั้นเป็นการนำเอาเครื่องคอมพิวเตอร์หลายเครื่องมาเชื่อมต่อกันผ่านทางเครือข่าย และมีซอฟแวร์ที่ทำหน้าที่เป็นตัวกลาง (Cluster Middleware) ในการทำให้กลุ่มคอมพิวเตอร์เหล่านี้สามารถสื่อสารกัน ช่วยกันทำงาน (Sequential Application) หรือทำงานเดียวกันร่วมกันได้ (Parallel Application) ทำให้การประมวลผลงานต่าง ๆ สามารถเสร็จภายในเวลาอันรวดเร็วมากยิ่งขึ้น

โครงสร้างการเชื่อมต่อของระบบพีซีคลัสเตอร์สามารถแสดงได้ดังนี้



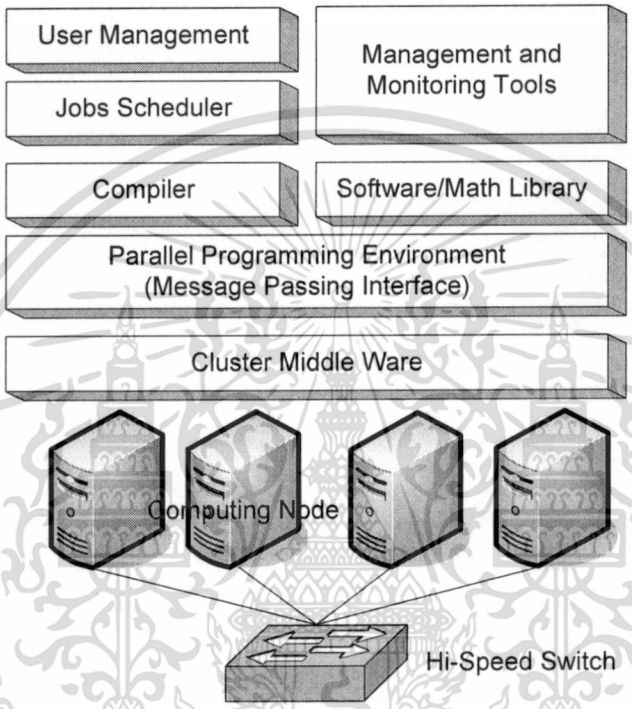
รูปที่ 2.1 โครงสร้างการเชื่อมต่อของระบบพีซีคลัสเตอร์

สำหรับการทำงานของระบบนี้เมื่อทำการประมวลผลแบบขนาน เราอาจจะเรียกระบบแบบนี้ได้อีกอย่างหนึ่งว่าระบบแบบหน่วยความจำแบบกระจาย (Distributed Memory) เนื่องจาก เครื่องแต่ละเครื่องที่อยู่ในระบบจะมีหน่วยประมวลผล และหน่วยความจำแยกกัน หน่วยประมวลผลของ Node A ไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถเข้าถึงหน่วยความจำของ Node B โดยตรงได้ แต่จะต้องทำการสื่อสารกันผ่านทางช่องทางหรือเครื่องมือในการสื่อสาร (Message Passing Interface) แทน ซึ่งจะแตกต่างจากระบบแบ่งปันหน่วยความจำ (Share Memory) เช่นเครื่องคอมพิวเตอร์ที่มีสถาปัตยกรรมเป็นแบบหลายหน่วยประมวลผล แต่ใช้หน่วยความจำร่วมกัน (SMP: Symmetric Multi Processor) แต่ละหน่วยประมวลผลสามารถที่จะเข้าถึง (ทั้งการอ่านและเขียน) หน่วยความจำส่วนกลางได้โดยตรง

องค์ประกอบของระบบพีซีคลัสเตอร์ประกอบด้วยส่วนประกอบต่าง ๆ ดังนี้



รูปที่ 2.2 โครงสร้างส่วนประกอบของระบบคลัสเตอร์

จากรูปสามารถที่จะแบ่งการทำงานออกได้ทั้งหมดสามส่วนด้วยกันคือ

1. โครงสร้างพื้นฐานของระบบคลัสเตอร์ ซึ่งจะประกอบไปด้วยส่วนประกอบทางฮาร์ดแวร์ การเชื่อมต่อเครือข่าย ระบบปฏิบัติการ และคลัสเตอร์มิดเดิลแวร์
2. เครื่องมือและโปรแกรมอัตถะประโยชน์ ประกอบด้วย โปรแกรมสื่อสารระหว่างโหนดโดยการส่งข้อความ คอมไพเลอร์ และชุดคำสั่งทางคณิตศาสตร์
3. การจัดการระบบคลัสเตอร์ ประกอบด้วยตัวจัดการลำดับงาน การจัดการสิทธิของผู้ใช้งานในระบบ ระบบตรวจสอบและจัดการคลัสเตอร์

โดยเครื่องคอมพิวเตอร์ที่สามารถนำมาสร้างระบบแบบนี้ได้นั้น สามารถเป็นเครื่องพีซีธรรมดาหรือเป็นเครื่องขนาดใหญ่ที่มีหลายหน่วยประมวลผลก็ได้ ข้อดีส่วนนี้ทำให้การขยายระบบของคลัสเตอร์สามารถทำได้อย่างง่ายดาย เราจึงสามารถที่จะเพิ่มประสิทธิภายในการคำนวณได้อย่างไม่จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ระบบกริดคอมพิวเตอร์

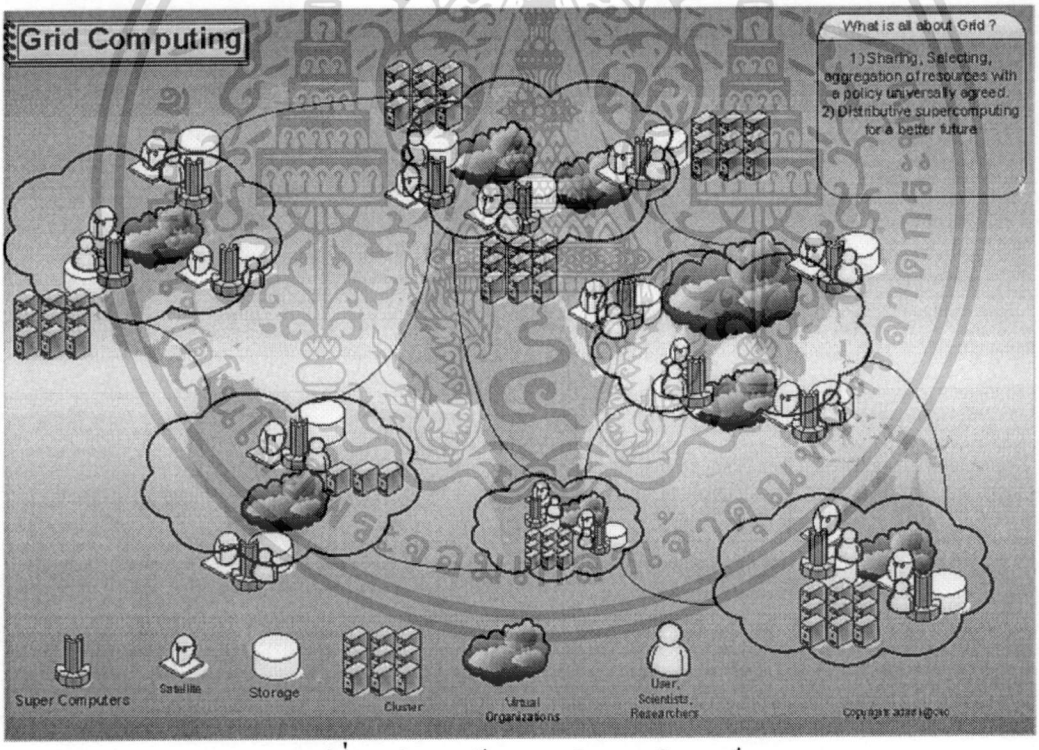
ระบบกริดคอมพิวเตอร์เป็นระบบคอมพิวเตอร์ขนาดใหญ่ที่ถูกนำมาเชื่อมต่อกันผ่านทางเครือข่าย เช่นเครือข่ายอินเทอร์เน็ต แล้วสามารถช่วยกันทำงานในด้านต่าง ๆ หรือแบ่งปันทรัพยากรต่าง ๆ ร่วมกันได้ โดยสมาชิกภายในกริดสามารถเป็นได้ทั้งเครื่องคอมพิวเตอร์ส่วนบุคคล เครื่องคอมพิวเตอร์ขนาดใหญ่ รวมถึงเครื่องคอมพิวเตอร์แบบคลัสเตอร์ที่ได้กล่าวถึงในหัวข้อที่ผ่านมาอีกด้วย

Ian Foster ได้ทำการนิยามว่า กลุ่มของคอมพิวเตอร์ใด ๆ ว่าเป็นกริดคอมพิวเตอร์หรือไม่ ให้ทำการพิจารณาด้วยเงื่อนไขต่อไปนี้

A grid system is that:

1. Coordinates resources that are not subject to centralized control…
2. …using standard, open, general-purpose protocols and interfaces…
3. …to deliver nontrivial qualities of service

จากนิยามดังกล่าว สามารถแสดงรูปแบบของระบบกริดคอมพิวเตอร์ได้ดังภาพ



รูปที่ 2.3 โครงสร้างของกริดคอมพิวเตอร์

กริดคอมพิวเตอร์สามารถจัดแบ่งประเภทออกเป็นได้ 3 รูปแบบคือ

1. กริดเพื่อการคำนวณ (Computational Grid) โดยกริดคอมพิวเตอร์ประเภทนี้จะถูกออกแบบและสร้างขึ้นมาเพื่อการประมวลผลงานที่ต้องการการประมวลผลมาก ๆ เป็นหลัก

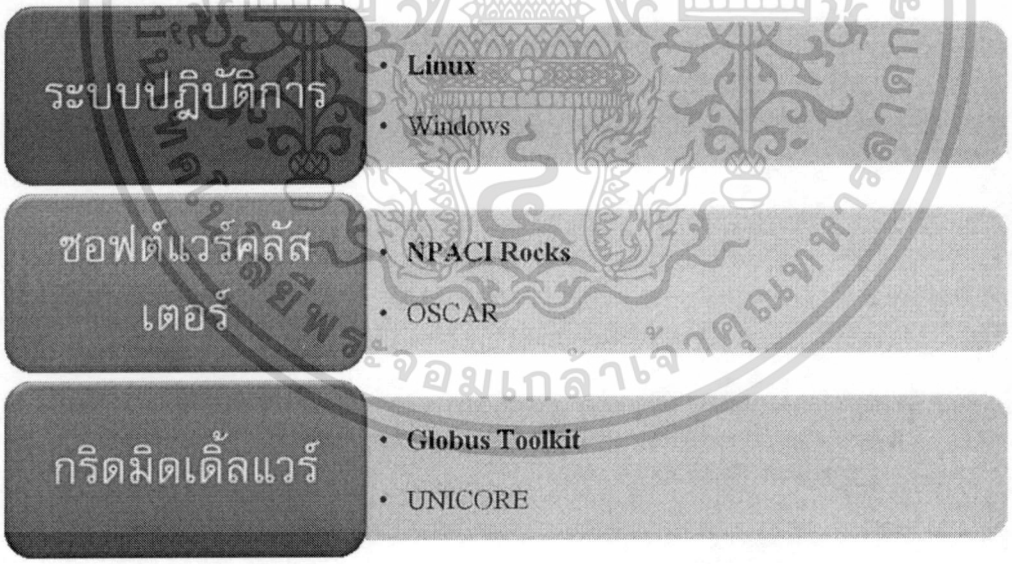
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กริดเพื่อการเก็บข้อมูล (Data Grid) จะถูกสร้างขึ้นมาเพื่อให้มีการจัดเก็บข้อมูลแบบกระจายได้ในที่ต่าง ๆ เป็นการเพิ่มพื้นที่และความสามารถในการเก็บข้อมูล การแบ่งปันข้อมูลของหน่วยงานที่อยู่ภายในกริดเดียวกัน
3. กริดเพื่อการเข้าถึงบริการ (Access Grid) เป็นบริการสำหรับติดต่อสื่อสารระยะไกล ที่ถูกสร้างขึ้นมาเพื่อรองรับการบริการเช่น ประชุมทางไกลบนเครือข่ายกริด เป็นต้น

เหตุผลหลักในการสร้างกริดคอมพิวเตอร์ก็คือการแบ่งปันทรัพยากรต่าง ๆ ที่อยู่ภายในกริด ให้เกิดการใช้งานอย่างคุ้มค่ามากที่สุด โดยทรัพยากรที่อยู่ในกริดนั้นจะหมายถึงสิ่งต่าง ๆ ดังนี้

- กลุ่มคน: ผู้ใช้, นักวิจัย, ผู้ควบคุมระบบ
- คอมพิวเตอร์: คอมพิวเตอร์, คลัสเตอร์, ไฟล์เซิร์ฟเวอร์
- แอพพลิเคชั่น: กริดมิลเดิ้ลแวร์, โปรแกรมของผู้ใช้
- ข้อมูล: ฐานข้อมูล, ผลลัพธ์การประมวลผล
- เครือข่าย: ช่องทางการส่งข้อมูล

เมื่อทำการพิจาณาลึกลงไปในรายละเอียดของส่วนโครงสร้าง จะได้องค์ประกอบของกริดคอมพิวเตอร์ดังแผนภาพ



รูปที่ 2.4 องค์ประกอบภายในของกริดคอมพิวเตอร์

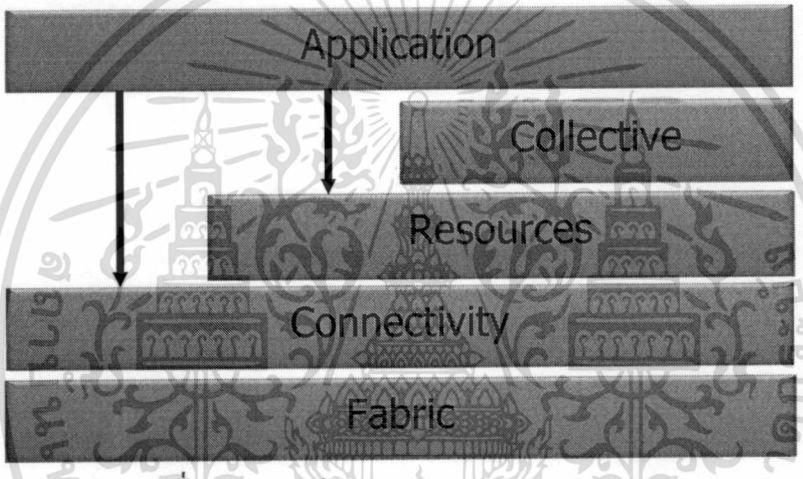
โดยจะประกอบด้วยชั้นของระบบปฏิบัติการ ชั้นของคลัสเตอร์มิดเดิ้ลแวร์ และชั้นของกริดมิดเดิ้ลแวร์ โดยซอฟแวร์ที่ทำงานในชั้นของ Grid Middleware ที่ได้ความนิยมมากในปัจจุบันคือ Globus Toolkit ซึ่งจะได้แนะนำเกี่ยวกับโครงสร้างการทำงานของซอฟแวร์นี้ในหัวข้อต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 กริดมิดเดิ้ลแวร์ (Grid Middleware)

กริดมิดเดิ้ลแวร์เป็นซอฟแวร์ที่ถูกสร้างขึ้นมาเพื่อเป็นตัวเชื่อมประสารการทำงานต่าง ๆ ของคอมพิวเตอร์ คลัสเตอร์ ที่อยู่ภายใต้กริดเดียวกัน ให้สามารถแบ่งปันทรัพยากรต่าง ๆ ร่วมกันได้ โดยมีการรักษาความปลอดภัย และการตรวจสอบสิทธิอย่างรัดกุม ทำให้ผู้ใช้สามารถติดต่อและเรียกใช้ทรัพยากรที่มีอยู่อย่างกระจัดกระจายได้ โดยที่ทรัพยากรเหล่านั้นมีความหลากหลาย สามารถเพิ่มและลดได้ตลอดเวลา

กริดมิดเดิ้ลแวร์ที่มีการใช้งานอย่างแพร่หลายคือ Globus Toolkits ซึ่งถูกพัฒนาขึ้นมาโดย Globus Alliance (http://www.globus.org/alliance/) ในปัจจุบันได้มีการพัฒนาอย่างต่อเนื่องจนถึงเวอร์ชั่น 4 และมีการทำงานอยู่บนพื้นฐาน Web Service เป็นหลัก สำหรับโครงสร้างกริดตามความหมายของ Globus Toolkit สามารถแสดงได้ดังรูป



รูปที่ 2.5 โครงสร้างของกริดตาม Globus Toolkit

โดยแต่ละชั้นของโครงสร้างมีรายละเอียดคือ

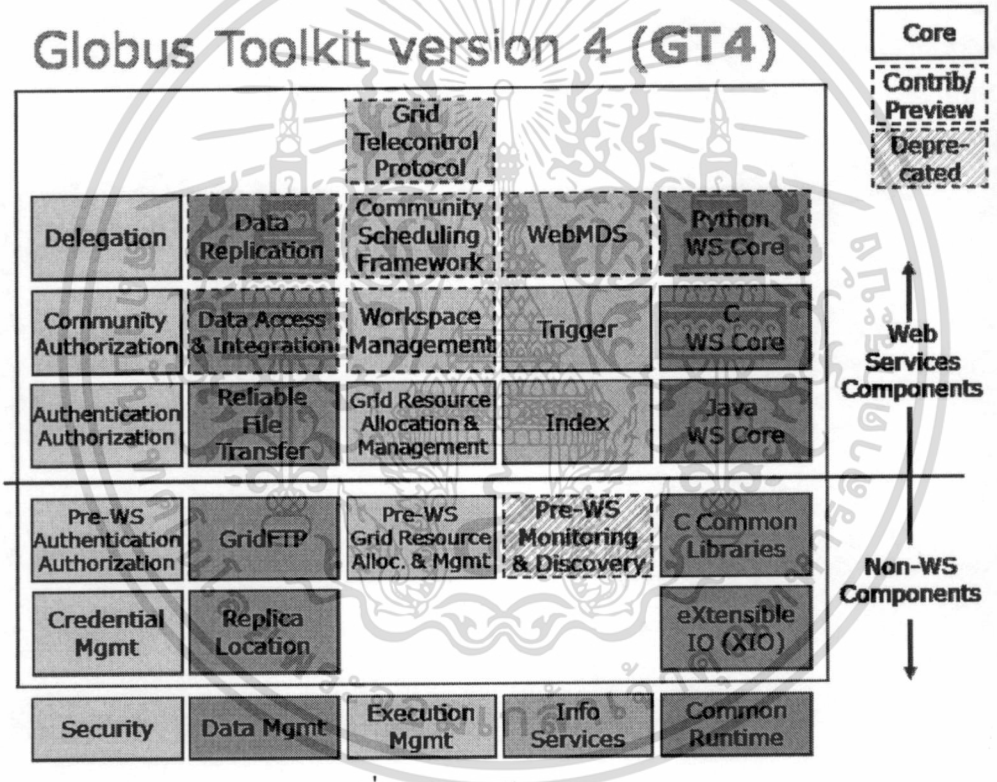
- **Fabric** เป็นทรัพยากรที่ถูกเรียกใช้โดยโปรโตคอลของกริด เช่น เครื่องคอมพิวเตอร์ คลัสเตอร์ หน่วยประมวลผล ที่เก็บข้อมูล
- **Connectivity** เป็นโปรโตคอลสำหรับการเชื่อมต่อเครือข่ายเข้าด้วยกันให้เป็นกริด บนความปลอดภัยแบบ **GSI** (Grid Security Infrastructure) ซึ่งเป็นความปลอดภัยระดับสูง
- **Resources** เป็นโปรโตคอลและเอพีไอสำหรับการเข้าใช้ทรัพยากรหนึ่งๆ โดยมีส่วนประกอบย่อยอื่น ๆ อีก เช่น
    - **GRAM** (Grid Resource Allocation Management) ทำหน้าที่จองทรัพยากรสำหรับประมวลผล และทำหน้าที่แสดงสถานะการทำงาน และสามารถควบคุมงานที่ประมวลผลบนทรัพยากรนั้น
    - **GridFTP** ทำหน้าที่ในการโอนถ่ายข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **GRIS** (Grid Resource Information Service) ทำหน้าที่แสดงข้อมูลของทรัพยากรของหนึ่งทรัพยากรว่าประกอบด้วยหน่วยประมวลผล, หน่วยจัดเก็บข้อมูลใดบ้าง

- **Collective** เป็นบริการสำหรับการจัดการกลุ่มของทรัพยากร เช่น การกระจายงาน การตรวจสอบสถานะของกลุ่มทรัพยากร เป็นต้น
- **Application** คือโปรแกรมที่ทำงานในระบบกริดและเรียกใช้บริการต่างๆ ในชั้นที่อยู่ต่ำกว่า

Globus Toolkit เวอร์ชั่น 4 (GT4) นั้นได้มีการพัฒนาขึ้นมาอย่างต่อเนื่อง ทำงานอยู่บนพื้นฐานของ Web Service แต่ก็ยังรองรับการทำงานกับส่วนประกอบเดิมที่เคยมีในซอฟแวร์รุ่นก่อนหน้า ดังนั้นใน GT4 จึงสามารถแยกส่วนประกอบออกเป็นสองกลุ่มคือ กลุ่ม Web Service Component และ กลุ่ม Non-Web Service Component สามารถแสดงโครงสร้างได้ดังรูป



รูปที่ 2.6 โครงสร้างของ GT4

จากรูปจะเห็นได้ว่า GT4 ประกอบด้วย ส่วนประกอบย่อย ๆ ดังนี้

- ระบบรักษาความปลอดภัย (Security)
- การจัดการข้อมูล (Data management)
- การจัดการประมวลผล (Execution management)
- การแสดงสถานะการทำงาน (Info Service)
- โครงสร้างพื้นฐานสำหรับการสร้างบริการ (Core runtime)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

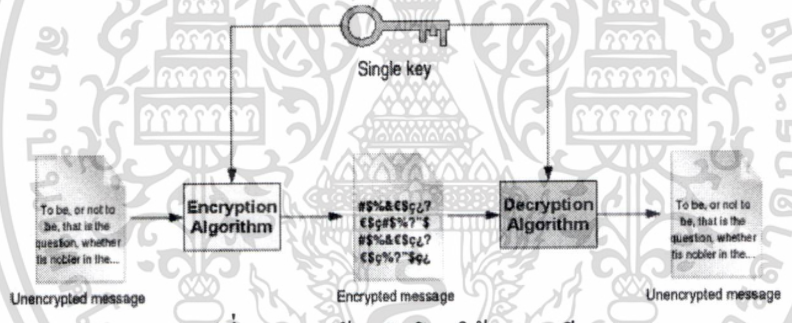
2.5 การรักษาความปลอดภัยภายในกริด

ระบบการรักษาความปลอดภัยภายในกริด เป็นองค์ประกอบที่สำคัญเป็นอย่างยิ่งภายในกริด เนื่องจากทรัพยากรมีอยู่อย่างกระจัดกระจาย ไม่ว่าจะเป็นคอมพิวเตอร์ คลัสเอตร์ หรือหน่วยเก็บข้อมูล ซึ่งการจัดการส่วนประกอบต่าง ๆ เหล่านี้ไม่ได้มีการจัดการจากศูนย์กลาง แต่จะขึ้นอยู่กับหน่วยงานหรือองค์กรที่รับผิดชอบ ดังนั้นการเข้าถึงทรัพยากรต่าง ๆ ผู้ใช้งานที่อยู่กริดจึงเป็นเรื่องที่ต้องมีมาตรฐานร่วมกัน และมีการรักษาความปลอดภัย และการพิสูจน์สิทธิอย่างเข้มงวด

สำหรับการการสื่อสารแบบปลอดภัย (Secure Communication) นี้ จะมีองค์ประกอบสำคัญอยู่สามส่วน คือ

- **Privacy** จะหมายถึงส่วนที่ทำการเข้ารหัสข้อมูลเพื่อให้ผู้ส่งสารและผู้รับสารเท่านั้นที่มีสิทธิ์รับทราบเนื้อหาของการสื่อสาร ผู้อื่นนอกเหนือจากนี้อาจได้รับข้อมูล แต่ไม่สามารถเข้าใจเนื้อสารได้ ซึ่งวิธีการในการเข้ารหัสนี้มีสองแบบคือ แบบ Symmetric Key Encryption และ Asymmetric Key Encryption

  การเข้ารหัสโดยใช้กุญแจเดียวสามารถแสดงได้ดังนี้



รูปที่ 2.7 การเข้ารหัสโดยใช้กุญแจเดียว

ตัวอย่างการเข้ารหัส

1 2 3 4 5 5 4 3 2 1	สาร
+1 2 2 3 3 3 4 4 4 4	กุญแจ
2 4 5 7 8 8 8 7 6 5	สารที่เข้ารหัส

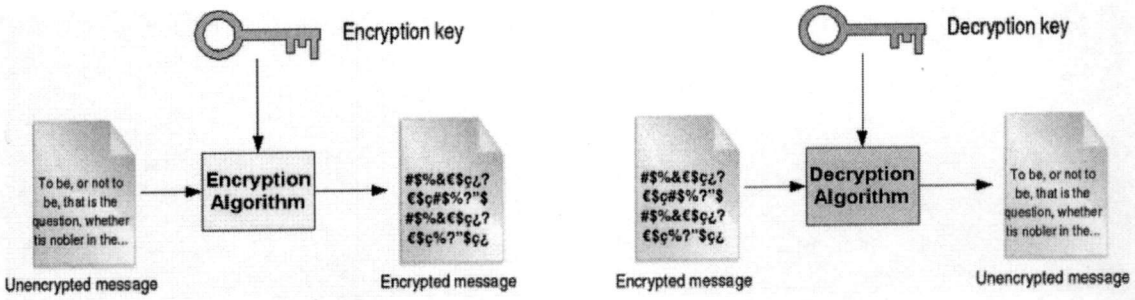
ตัวอย่างการถอดรหัส

2 4 5 7 8 8 8 7 6 5	สารที่เข้ารหัส
-1 2 2 3 3 3 4 4 4 4	กุญแจ
1 2 3 4 5 5 4 3 2 1	สาร

ส่วนการเข้ารหัสแบบ Asymmetric Key Encryption นั้นสามารถแสดงได้ดังนี้

79733

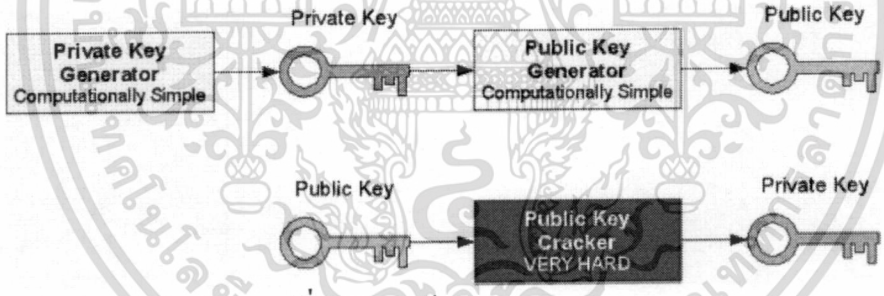
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 การเข้ารหัสแบบ Asymmetric Key Encryption

ในขั้นตอนการเข้ารหัส ผู้ส่งจะใช้กุญแจดอกแรกในการเข้ารหัส ส่วนในฝั่งผู้รับเมื่อได้รับสารแล้วจะทำการถอดรหัสโดยใช้กุญแจอีกดอกหนึ่ง ซึ่งจะเป็นกุญแจคนละดอกกัน

ซึ่งการเข้ารหัสแบบนี้เรียกอีกอย่างหนึ่งว่าการเข้ารหัสแบบกุญแจสาธารณะ (PKI: Public Key Infrastructure) โดยจะมีกุญแจอยู่สองอย่างคือ Private Key และ Pubic Key โดยที่ Public Key จะเป็นกุญแจที่แจกจ่ายเป็นบุคคลที่ต้องการทำการสื่อสารด้วย ส่วน Private Key นั้น จะต้องเก็บเป็นความลับเฉพาะเจ้าของกุญแจเท่านั้น ลักษณะของการทำงานคือ เราสามารถที่จะสร้าง Public Key ขึ้นมาจาก Private Key ได้ง่าย ๆ แต่ในทางกลับกัน เป็นไปได้ยากมากที่จะสร้าง Private Key กลับคืนมาจาก Public Key แสดงได้ดังรูป



รูปที่ 2.8 การสร้างกุญแจของ PKI

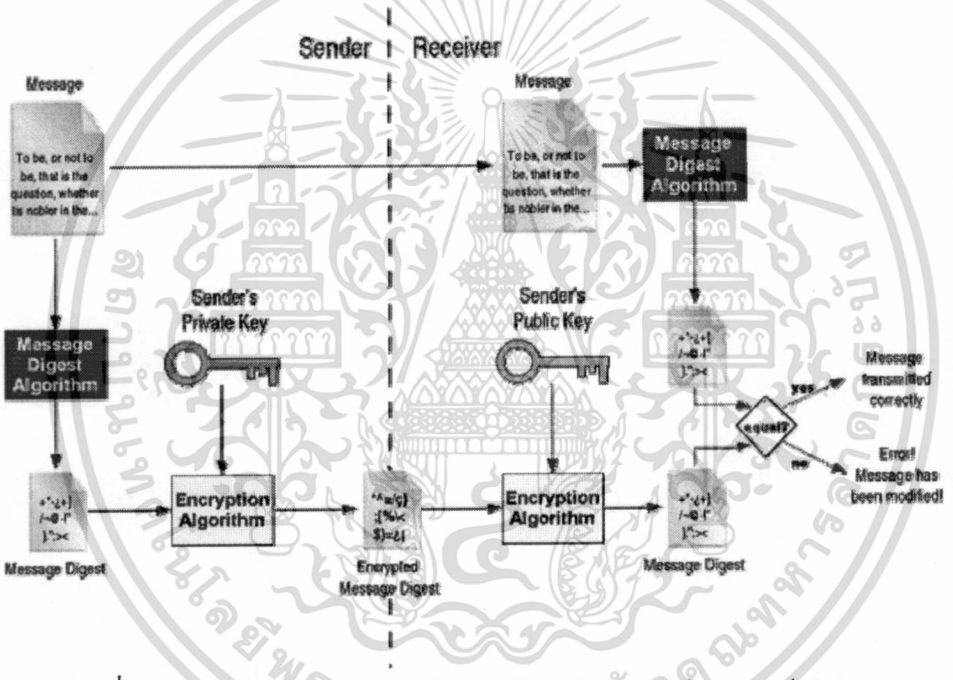
สำหรับข้อแตกต่าง ของการเข้ารหัสแบบ Symmetric Key Encryption และ Asymmetric Key Encryption คือ การเข้ารหัสแบบกุญแจเดี่ยวนั้น สะดวกและง่ายในการทำงานมาก แต่ก็ไม่ปลอดภัย ส่วนการเข้ารหัสแบบกุญแจคู่นั้น อาจยุ่งยากซับซ้อนกว่าแต่ก็ปลอดภัยกว่าด้วยเช่นกัน ซึ่งภายใน GSI นี้จะใช้การเข้ารหัสแบบกุญแจคู่เป็นหลัก

แต่แค่ขั้นตอนการเข้ารหัสดังที่กล่าวมายังไม่ได้พิสูจน์ว่า ข้อมูลที่ส่งมานั้นมีความถูกต้อง มีการบิดเบือน หรือการแก้ไขข้อมูลจากบุคคลอื่นในระหว่างการส่งหรือไม่ ดังนั้นจะต้องมีการตรวจสอบความถูกต้องของข้อมูลตามขั้นตอนถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Integrity** เป็นส่วนที่ทำหน้าที่ในการพิสูจน์ว่าข้อมูลที่ได้รับมามีความถูกต้อง มีการบิดเบือนหรือแก้ไขโดยบุคคลอื่นในระหว่างการส่งหรือไม่ โดยภายใน GSI นั้นจะใช้หลักการของลายเซ็นต์ดิจิตอล (Digital Signature) ในการทำงานดังกล่าว โดยมีขั้นตอน คือ
    - สรุปข้อความของสารที่จะส่ง
    - เข้ารหัสสารที่ย่อแล้วด้วย Private Key
    - ส่งข้อมูลดังกล่าวไปพร้อมกับสารต้นฉบับ
    - ผู้รับถอดรหัสสารที่ถูกย่อด้วย Public Key
    - นำสารต้นฉบับที่ได้รับมาทำการย่อสรุปแล้วเปรียบเทียบกับสารที่ถูกเข้ารหัส
    - ถ้าตรงกันก็จะแสดงว่าสารที่ได้รับไม่ถูกบิดเบือน

สามารถอธิบายขั้นตอนได้ดังรูป



รูปที่ 2.9 การตรวจสอบความถูกต้องของข้อมูลโดยการใช้ลายเซนต์ดิจิตอล

จากรูปจะเห็นได้ว่าเราสามารถที่จะตรวจสอบความถูกต้องของข้อมูลโดยการใช้ลายเซนต์ดิจิตอลได้ แต่เนื่องจากข้อมูลต้นฉบับไม่ได้ทำการเข้ารหัส ดังนั้นในการใช้งานจริง จะต้องมีการเข้ารหัสข้อมูลที่เป็นต้นฉบับด้วย และสำหรับ Message Digest Algorithm ที่นิยมใช้งาน เช่น MD5 ซึ่งการเปลี่ยนแปลงเพียงเล็กน้อยของข้อมูล ผลลัพธ์ที่ได้ของลายเซ็นต์ดิจิตอลก็จะต่างกันด้วย ดังตัวอย่าง

```
[aue@blue65 ~]$ echo "Computer Research and Service Center" | md5sum
f9092ddd4b73544d8e94c399f63d7382  -
[aue@blue65 ~]$ echo "Computer Research and Services Center" | md5sum
26434047f76f65f7b763ecfac7ba332f  -
[aue@blue65 ~]$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ในขั้นตอนนี้ก็ยังไม่สามารถพิสูจน์ได้ว่าผู้ส่งสารหรือผู้รับสารเป็นบุคคลตัวจริงดังที่กล่าวอ้างหรือไม่ จะต้องมีการพิสูจน์ตัวตนก่อน โดยจะมีขั้นตอนในหัวข้อถัดไป

- **Authentication** เป็นส่วนที่ทำหน้าที่ในการพิสูจน์ว่าผู้ส่งและผู้รับสาร เป็นตัวจริงตามที่ได้มีการแสดงตนจริงหรือไม่ โดยในส่วนนี้จะใช้เทคโนโลยีที่เรียกว่าใบรับรองดิจิตอลหรือ Digital Certificate ในการพิสูจน์ตัวตน ใบรับรองดิจิตอลเป็นไปตามมาตรฐาน X.509 และมีไว้เพื่อทำการยืนยันตัวตนของคนที่เป็นเจ้าของ Public key ว่าเป็นเจ้าของ Public Key นั้น ๆ จริงหรือไม่ โดยในการทำงานแบบนี้จะมีผู้เกี่ยวข้องอยู่สองส่วนคือ ผู้ถือใบรับรอง และผู้ออกใบรับรองหรือเรียกอีกชื่อหนึ่งว่า Certificate Authority (CA) นั่นเอง โดยรูปเนื้อหาของ Digital Certificate มีตัวอย่างดังนี้

```
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 501 (0x1f5)
    Signature Algorithm: md5WithRSAEncryption
    Issuer: O=Grid, OU=ThaiGrid, CN=ThaiGrid CA
    Validity
      Not Before: Nov  6 15:23:49 2006 GMT
      Not After : Nov  5 15:23:49 2007 GMT
    Subject: O=Grid, OU=ThaiGrid, OU=thaigrid.or.th, CN=Nopparat Pantsaena
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (1024 bit)
        Modulus (1024 bit):
          00:dc:28:2a:3b:cd:3d:59:91:c5:08:b1:54:55:8b:
          be:84:7d:73:00:95:06:09:7c:1e:7d:9d:35:a9:85:
          f3:1e:23:a9:9a:87:5c:be:b5:6c:55:4e:16:b0:5b:
          e4:d1:71:85:d1:3b:f0:49:b6:a0:c9:8f:2b:02:f5:
          fe:33:8d:4b:86:96:4d:b1:94:e3:37:71:d4:3c:a9:
          00:e1:64:33:5b:0d:1a:09:25:ff:e0:f3:23:f0:06:
          52:4b:c7:4e:7f:4e:ff:d6:9c:a3:bd:74:39:f9:20:
          24:92:2b:b4:e7:8b:18:cb:ef:e6:14:98:1a:26:5b:
          53:60:d4:1a:f7:6f:a1:55:e9
        Exponent: 65537 (0x10001)
    X509v3 extensions:
      Netscape Cert Type:
        SSL Client, SSL Server, S/MIME, Object Signing
  Signature Algorithm: md5WithRSAEncryption
    07:5e:2e:47:29:6a:3a:dc:ca:f3:93:f3:66:ca:12:7a:c1:09:
    21:46:9e:52:37:36:b2:04:b1:f5:d6:b1:97:1e:05:44:05:42:
    39:5b:82:f7:82:64:d0:84:b0:4f:a7:ef:21:96:cb:d2:33:35:
    0a:8e:fa:6f:15:c8:9f:5e:9c:af:c0:9e:d6:e0:01:38:db:32:
    33:16:68:2c:80:d0:ed:e7:9e:ce:88:d1:de:4e:e3:f6:c8:98:
    c4:c7:2d:39:a4:20:69:83:7f:16:d5:4f:83:24:7e:8d:5f:18:
    f7:32:b3:2f:f5:39:f3:4c:d9:c9:50:fc:8d:22:91:52:0c:37:
    7e:bb
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในใบรับรองดิจิตอลนี้จะประกอบด้วยส่วนสำคัญ ดังนี้

Subject	คือ รายละเอียดของผู้ถือใบรับรอง
Subject's Public Key	คือ ข้อมูล Public Key ของผู้ถือใบรับรอง และจะบอกด้วยว่า Key นี้ถูกสร้างขึ้นด้วยวิธีการ (Algorithm) แบบใด
Issuer's Subject	คือ ผู้ออกใบรับรอง (CA)
Digital Signature	คือ ข้อมูลลายเซ็นต์ดิจิตอลของข้อมูลทั้งหมดภายในใบรับรอง ซึ่งข้อมูลส่วนนี้ถูกเข้ารหัสไว้ด้วย Private Key ของ CA ดังนั้นเราจึงต้องใช้ Public Key ของ CA ในการถอดรหัสของข้อมูลส่วนนี้

สำหรับ Subject และ Issuer's Subject จะถูกเก็บอยู่ในรูปแบบของ Distinguish Name (DN) โดยแต่ละส่วนหมายถึง

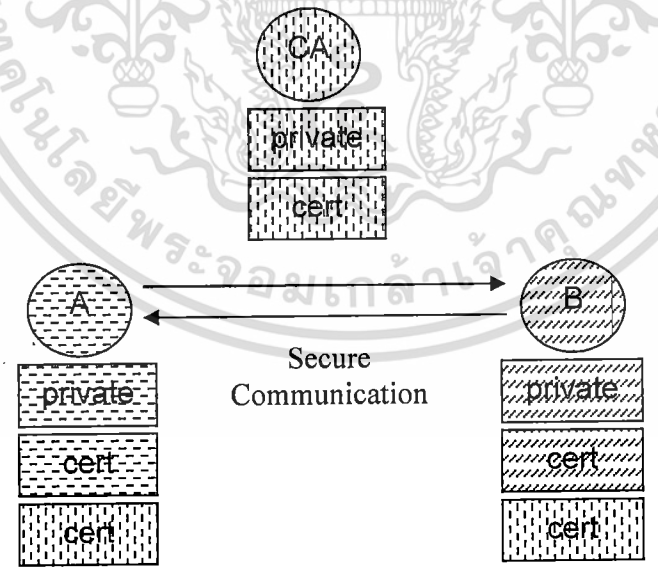
O (Organization) หมายถึง หน่วยงาน

OU (Organization Unit) หมายถึง หน่วยงานย่อย

CN (Common Name) หมายถึง ชื่อของผู้ถือใบรับรอง หรือผู้ออกใบรับรอง

C (Country) หมายถึง ชื่อของประเทศ

เราสามารถแสดงความสัมพันธ์ของการสื่อสารที่มี CA ได้ดังรูป



รูปที่ 2.10 ภาพแสดงความการติดต่อสื่อสารโดยใช้ใบรับรองดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Grid Security Infrastructure (GSI) เป็นประกอบภายใน Globus Toolkit ที่ถูกพัฒนาขึ้นมาบนพื้นฐานของการเข้ารหัสแบบ PKI ลายเซ็นต์ดิจิตอล และใบรับรองดิจิตอล เพื่อใช้เป็นมาตรฐานเพื่อการรักษาความปลอดภัยภายในกริด ทำให้ GSI ครอบคลุมการสื่อสารแบบปลอดภัยดังที่กล่าวมาทั้งหมด อีกทั้ง GSI ยังมีหน้าที่สำคัญ ๆ ดังต่อไปนี้

- การทำซิงเกิ้ลไซน์ออน (Single sign-on) ทำให้ผู้ใช้ทำการ Login เพียงครั้งเดียวก็สามารถเข้าถึงทรัพยากรต่าง ๆ ที่อยู่ภายในกริดได้ทั้งหมด ไม่ต้อง Login ใหม่เมื่อมีการเรียกใช้งานทรัพยากรจากคลัสเตอร์ซึ่งอยู่ต่างสถานที่กัน
- การมอบหมายอำนาจ (Delegation) GSI ได้มีการพัฒนาระบบการมอบหมายอำนาจเพื่อให้ผู้อื่นกระทำแทน เสมือนเจ้าของใบรับรองนั้นเป็นคนสั่งงานต่าง ๆ เอง
- การพิสูจน์ตัวจริง (Authentication) ในขั้นตอนนี้จะใช้หลักการพิสูจน์ตัวจริงตามวิธีการของ Digital Certification
- การอนุญาต (Authorization) ในส่วนนี้แม้จะไม่นับว่าเป็นหัวใจของการสื่อสารแบบปลอดภัย แต่ถือว่าเป็นส่วนสำคัญของการเข้าใช้งานกริด เนื่องจากเมื่อมีการพิสูจน์ตัวตนได้แล้วจะต้องมีการอนุญาตและมอบหมายสิทธิให้แก่ผู้ใช้ที่ทำการติดต่อเข้ามานั้น โดยผู้ใช้แต่ละคนก็จะมีสิทธิต่างกัน ตามหน้าที่และบทบาทที่รับผิดชอบ เช่น ผู้ดูแลระบบและผู้ใช้งานทั่วไป เป็นต้น

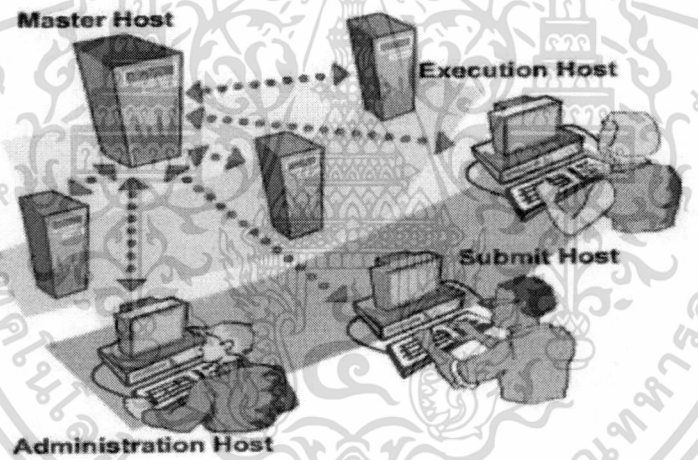
ในหัวข้อถัดไปจะเป็นเรื่องของซอฟแวร์ที่ทำหน้าที่ในการจัดลำดับงานที่จะประมวลผลในคลัสเตอร์และกริดคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ซอฟแวร์จัดการลำดับงาน

ในระบบคลัสเตอร์ หรือกริดคอมพิวเตอร์ที่มีผู้ใช้จำนวนหลายคน จะต้องมีการจัดการระบบงาน (Job Management) ที่ดี มีการแบ่งปันการใช้ทรัพยากรอย่างเป็นระบบและเสมอภาค ดังนั้นในการส่งงานเข้าไปประมวลผลบนระบบคลัสเตอร์หรือระบบกริดคอมพิวเตอร์นั้น จะต้องมีซอฟแวร์ที่คอยช่วยเหลือในการจัดลำดับงาน (Job Scheduler) เพื่อให้เกิดการใช้งานทรัพยากรระบบอย่างเต็มที่และเหมาะสม โดยเมื่อผู้ใช้ทำการส่งงานเข้ามาในระบบ และซอฟแวร์จัดลำดับงานได้รับแล้ว ผู้ใช้สามารถออกจากระบบได้เลย ตัวซอฟแวร์จัดลำดับงานจะช่วยส่งงานไปประมวลผลยังเครื่องต่าง ๆ ในระบบต่อไป และจะเก็บรายละเอียดของผลลัพธ์ รายละเอียดเกี่ยวกับข้อผิดพลาดที่เกิดขึ้นในการประมวลผลงานนั้น ๆ ไว้ให้อีกด้วย ตัวอย่างของซอฟแวร์จัดลำดับงานที่นิยมใช้กันอย่างแพร่หลายเช่น Sun Grid Engine (SGE), Condor และ OpenPBS เป็นต้น ซึ่งปกติ SGE จะถูกติดตั้งมาพร้อมกับระบบปฏิบัติการ Rock Cluster

Sun Grid Engine เป็นซอฟแวร์เปิดที่ได้รับการสนับสนุนจาก Sun Microsystem และ Sun เองก็มีซอฟแวร์ทางการค้าชื่อว่า Sun N1 Grid Engine ซึ่งพัฒนาต่อจาก Sun Grid Engine อีกที เราสามารถแสดงโครงสร้างการทำงานของ Sun Grid Engine ได้ดังรูป



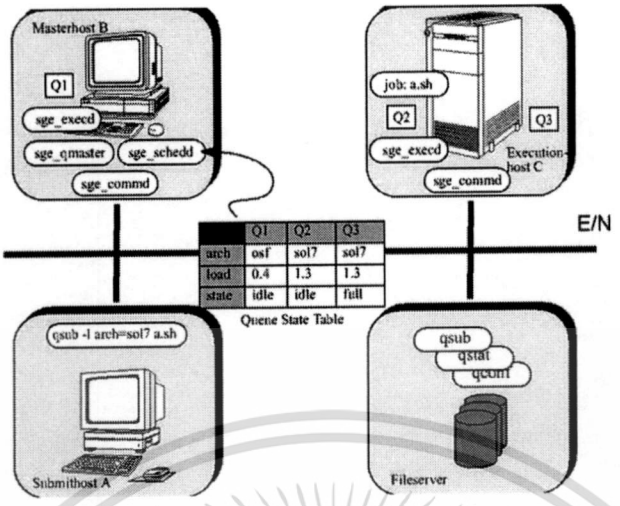
รูปที่ 2.11 ตัวอย่างการทำงานของ Sun Grid Engine

จากรูปจะเห็นได้ว่า Sun Grid Engine ได้แบ่งประเภทของเครื่องที่อยู่ในระบบออกเป็น 4 ประเภททดังนี้

1. เครื่องมาสเตอร์ (Master Host) เป็นเครื่องที่ควบคุมการจัดลำดับของงานที่ถูกส่งเข้ามาในระบบ โดยปกติแล้วจะทำงานอยู่บนเครื่องแม่ข่าย (Front-end node)
2. เครื่องประมวลผล (Execution Host) เป็นเครื่องที่ทำหน้าที่ประมวลผลงานต่าง ๆ ที่ถูกส่งมาจากเครื่อง Master
3. เครื่องที่ส่งงาน (Submit Host) เป็นเครื่องที่ผู้ใช้เข้ามาในระบบและเป็นที่ส่งงานไปยังระบบจัดลำดับงาน SGE ซึ่งในคลัสเตอร์แบบ Rocks นั้นจะอยู่บนเครื่องแม่ข่าย
4. เครื่องผู้ดูแลระบบ (Administrative Host) เป็นเครื่องที่ผู้ดูแลระบบเข้ามาในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากมองลึกเข้าไปในรายละเอียดของ SGE จะประกอบด้วยส่วนต่าง ๆ ดังรูป



รูปที่ 2.12 โครงสร้างภายในของ SGE

จากรูปจะเห็นได้ว่าโครงสร้างการทำงานของ SGE ประกอบด้วยหลายส่วน โดยจะมีเดมอน (Daemon) ที่เกี่ยวข้องกับการทำงานดังนี้

- sge_commd เป็นส่วนที่ทำหน้าที่ในการติดต่อสื่อสารระหว่างกันของ SGE โดย Daemon ตัวนี้จะถูกเรียกขึ้นมาให้ทำงานบนทุกเครื่องในระบบ
- sge_qmaster เป็นส่วนที่ทำหน้าที่ในการจัดลำดับงาน SGE และจะทำงานอยู่บนเฉพาะเครื่อง Front-end เท่านั้น
- sge_execd เป็นส่วนที่รอรับคำสั่งการประมวลผล จาก Master โดยปกติจะทำงานอยู่เฉพาะเครื่องประมวลผล (Execution Host) เท่านั้น

คิวประมวลผลงาน (Queue)

ส่วนประกอบอีกอย่างที่สำคัญของ SGE คือ คิวประมวลผลงาน โดยจะเป็นส่วนที่ทำหน้าที่เก็บรายการของงานที่ถูกส่งเข้ามาในระบบ เพื่อรอประมวลผล ในระบบของ SGE สามารถสร้างสร้างคิวขึ้นมาได้หลายคิวตามนโยบาย (policy) ที่ได้กำหนดไว้ เพื่อให้เกิดประโยชน์สูงสุดในการประมวลผล โดยนโยบายต่าง ๆ จะพิจารณาจากคุณลักษณะงาน ดังนี้

- ชนิดของหน่วยประมวลผลกลาง (CPU) เช่น หน่วยประมวลผลแบบ 32 Bit และ 64 Bit ผู้ดูแลระบบสามารถ สร้างคิวขึ้นมาโดยให้ชื่อว่า x32.q และ x64.q เป็นต้น
- การใช้งานหน่วยความจำ (Memory) เช่น himem.q เป็นต้น
- เวลาที่ใช้ประมวลผล (Time) เช่น short.q, long.q, weekly.q, monthly.q เป็นต้น
- ใบอนุญาตการใช้งานซอฟต์แวร์ (Licences) ผู้ดูแลระบบสามารถสร้างคิวขึ้นมาโดยเฉพาะสำหรับซอฟแวร์ที่ต้องการไลเซนต์ เช่น matlab.q, mathematica.q, fluent.q เป็นต้น

การส่งงานไปประมวลผล ผู้ใช้ไม่จำเป็นต้องระบุคิวประมวลผลงาน ระบบจัดลำดับงาน SGE จะเป็นผู้จัดการให้โดยพิจารณาตามคุณสมบัติของงานแล้วจัดลำดับในคิวที่เหมาะสมให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องมือการใช้งานระบบจัดลำดับงาน SGE

ในการจัดการงานที่ถูกส่งเข้าไปประมวลผลใน SGE นั้น จะมีเครื่องมือพื้นฐานสำหรับจัดการลำดับงานต่าง ๆ ให้ โดยประกอบด้วยชุดคำสั่งต่าง ๆ ดังนี้

คำสั่ง	คำอธิบาย
qsub	ส่งงานไปประมวลผล
qstat	แสดงสถานะการทำงาน
qdel	ยกเลิกงานที่ประมวลผล
qhost	แสดงเครื่องประมวลผลที่สามารถติดต่อได้ในขณะนี้
qalter	แก้ไขค่าคุณลักษณะของงาน

สำหรับการส่งงานเข้าไปประมวลผลจะใช้คำสั่ง qsub โดยมีรูปแบบคำสั่งดังนี้

งานแบบ Serial Application

```
qsub -cwd \
     -S /bin/bash \
     -o /home/USER_NAME/OUTPUT_LOG \
     -e /home/USER_NAME/ERROR_LOG \
     -N ProgramName_EstimatedHoursToFinish_MiscComment \
     script ARGUMENTS
```

งานแบบ Parallel Application

```
qsub -cwd \
     -S /bin/bash \
     -pe mpi 10 \
     -o /home/USER_NAME/OUTPUT_LOG \
     -e /home/USER_NAME/ERROR_LOG \
     -N ProgramName_EstimatedHoursToFinish_MiscComment
     script ARGUMENTS
```

โดยที่

-cwd กำหนดให้ sge ใช้ Directory ปัจจุบันในการทำงานงาน (Current Working Directory)

-S เป็นการระบุให้ /bin/bash เป็น Shell มาตรฐานในการทำงาน

-pe ย่อมาจาก Parallel Environment โดยจะระบุ Option นี้ในการประมวลผลแบบขนาน สำหรับตัวเลขที่ใส่ไปด้านหลังจะหมายถึงจำนวนของ Processing Element ที่งานนี้สามารถใช้ได้ เราสามารถดูรายการ Parallel Environment อื่น ๆ ได้ โดยใช้คำสั่ง **qconf –spl**

-o เป็นการระบุไฟล์ที่จะจัดเก็บผลลัพธ์ที่ถูกส่งออกมาทางหน้อจอขณะทำงาน

-e เป็นการระบุไฟล์ที่จะจัดเก็บข้อผิดพลาดที่ถูกแสดงออกมาทางหน้าจอขณะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-N เป็นการระบุชื่อของงานที่ทำการประมวลผล

Script คือ Shell Script ที่ถูกเขียนขึ้นมาเพื่อสั่งให้โปรแกรมที่ต้องการทำการประมวลผล

ARGUMENTS คือ ค่าต่าง ๆ ที่ต้องการส่งให้แก่โปรแกรม

การเขียนสคริปต์งาน

จากตัวอย่างการใช้งาน qsub จะพบว่าการส่งงานเข้าไปใน SGE นั้น จะต้องเขียนคำสั่งต่าง ๆ ให้เป็น Shell Script ก่อน โดยภายใน Script สามารถเรียกโปรแกรมต่าง ๆ ขึ้นมาทำงานได้มากกว่า 1 โปรแกรม แสดงได้ดังตัวอย่าง

1. เขียน Script งาน

```
#!/bin/bash
date
echo "Hello world"
```

2. ส่งงานไปยังระบบของ SGE ด้วยคำสั่ง qsub

```
[aue@blue65 ~]$ qsub simple.sh
Your job 357 ("simple.sh") has been submitted
[aue@blue65 ~]$
```

สิ่งที่สำคัญที่ต้องให้ความสนใจคือ หมายเลขงาน (Job ID) ซึ่งต้องเอาไว้อ้างอิงถึงงานนี้ในภายหลัง เช่น ถ้าหากต้องการยกเลิกการประมวลผลงาน จะต้องใช้คำสั่ง qdel พร้อม Job ID เป็นต้น

3. ทำการตรวจสอบสถานะของงานภายใน Queue

```
[aue@blue65 ~]$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
    357 0.00000 simple.sh  aue          qw    01/04/2006 13:08:52                                    1
[aue@blue65 ~]$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
    357 0.55500 simple.sh  aue          r     01/04/2006 13:09:04 all.q@compute-0-1.local        1
[aue@blue65 ~]$
```

สถานะ "qw" หมายถึง งานกำลังรอการประมวลผลอยู่ในคิว ส่วน "r" หมายถึง งานนั้นถูกสั่งให้ทำงานแล้วโดย SGE

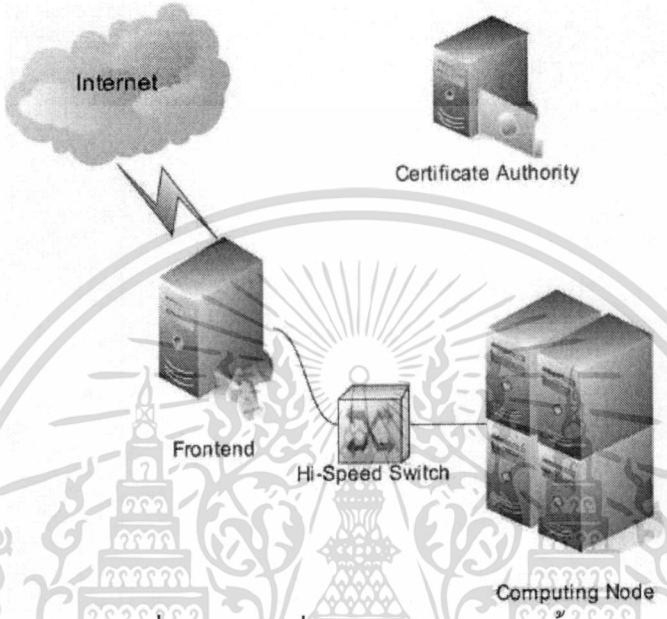
4. ตรวจสอบผลลัพธ์จากการทำงานดังนี้

```
 [aue@blue65 ~]$ ls -dl simple.sh*
-rwxrwxr-x  1 aue aue  36 Jan  4 13:08 simple.sh
-rw-r--r--  1 aue aue   0 Jan  4 13:09 simple.sh.e357
-rw-r--r--  1 aue aue 125 Jan  4 13:09 simple.sh.o357
[aue@blue65 ~]$ cat simple.sh.o357
Warning: no access to tty (Bad file descriptor).
Thus no job control in this shell.
Fri Jan  4 13:09:05 ICT 2006
Hello world
[aue@blue65 ~]$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การออกแบบระบบ

ในงานวิจัยนี้จะได้จัดทำระบบในรูปแบบของลีนุกซ์คลัสเตอร์ หลังจากนั้นจึงจะทำการเชื่อมต่อระบบเข้ากับอีกหลาย ๆ คลัสเตอร์ของสถาบันสมาชิก ให้กลายเป็นระบบกริดคอมพิวเตอร์ในที่สุด โดยทาง ThaiGrid จะทำหน้าที่เป็น Certificate Authority (CA) และทำการแจกจ่าย Certificate มายัง Cluster ต่าง ๆ ที่เชื่อมต่ออยู่ด้วยกัน ระบบที่ออกแบบไว้จะมีรายละเอียด ดังรูป



รูปที่ 2.12 ระบบที่จะถูกสร้างในงานวิจัยนี้

ในระบบที่จะจัดทำขึ้น จะมีหนึ่งเครื่องที่ทำหน้าที่เป็น Front-end ส่วนเครื่องอื่น ๆ ทำหน้าที่เป็น Computing Node (หรือ Execution Host ตามความหมายของ SGE) ที่ Front-end จะทำหน้าที่เป็น Storage ให้แก่เครื่องอื่น ๆ ด้วย

โครงสร้างทางฮาร์ดแวร์ของโครงการวิจัยนี้นั้นไม่ได้ซับซ้อนหรือมีส่วนประกอบมากมาย แต่ส่วนที่ถือว่าสำคัญคือ กริดมิดเดิ้ลแวร์หรือซอฟต์แวร์ที่เป็นตัวกลางที่ทำให้เกิดความร่วมมือในการประมวลผล และในปัจจุบัน กริดมิดเดิ้ลแวร์ที่ได้รับความนิยมมาก คือ Globus Toolkit ซึ่งภายในโครงการวิจัยนี้ก็จะได้ทำการติดตั้งซอฟต์แวร์ตัวนี้ลงไปบนระบบ และจะได้กล่าวโดยละเอียดในบทถัดไป

2.8 บทสรุป

ในบทนี้ได้กล่าวถึงระบบลีนุกซ์คลัสเตอร์ กริดคอมพิวเตอร์ และกริดมิดเดิ้ลแวร์ การรักษาความปลอดภัยบนกริดโดยใช้มาตรฐานของ GSI: Grid Security Infrastructure ทำให้เห็นภาพและเกิดความเข้าใจเกี่ยวกับระบบการเข้ารหัส ลายเซนต์ดิจิตอล และใบรับรองดิจิตอล ซึ่งถือเป็นหัวใจสำคัญของ GSI และสุดท้ายได้กล่าวถึงระบบที่จะได้จัดทำขึ้นในโครงการนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การติดตั้งระบบ

3.1 บทนำ

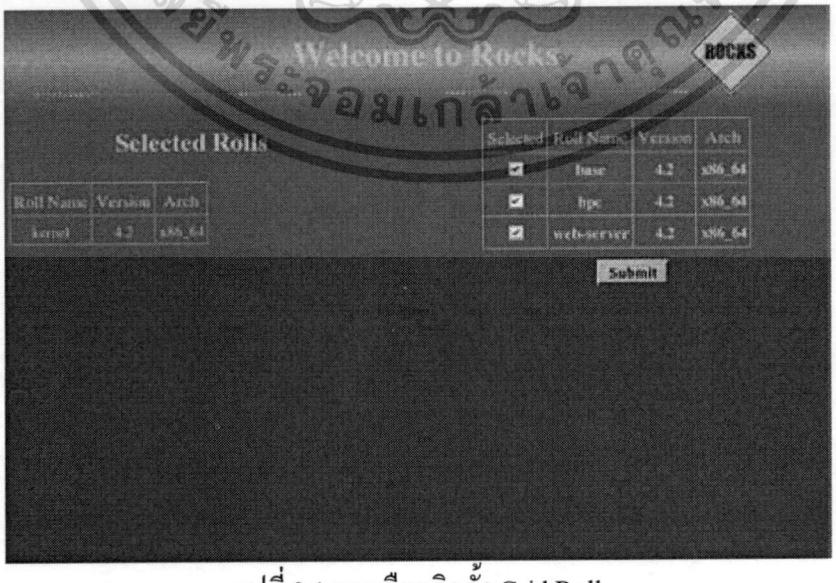
ในบทนี้จะได้กล่าวถึงขั้นตอนวิธีการต่าง ๆ ในการติดตั้งซอฟแวร์ การจัดการกี่ยวกับใบรับรองดิจิตอล ซึ่งเป็นขั้นตอนสำคัญมากของ Grid Security Infrastructure (GSI) ที่ได้กล่าวไว้ในบทที่แล้ว และในบทนี้เองจะได้มีการกล่าวถึงการจัดการคิวงาน และการให้บริการแก่ผู้ใช้งานทั่วไปอีกด้วย

3.2 การติดตั้ง Globus Toolkit

การติดตั้ง Globus Toolkit สามารถติดตั้งทั้งแบบวิธีการคอมไพล์ (Compile) จาก Source Code และติดตั้งโดยใช้ Binary file ที่มีการคอมไพล์มาแล้ว โดยในขั้นตอนการคอมไพล์ Globus Toolkit นั้นจะต้องทำการ Download ไฟล์ต้นฉบับจาก http://www.globus.org/toolkit แล้วทำการคอมไพล์ด้วย C Compiler และต้องการซอฟแวร์อื่น ๆ ด้วยเช่น Java Runtime, Java Compiler, Apache Ant และเครื่องมืออื่นๆ โดยต้องตรวจสอบจากคู่มือการคอมไพล์ก่อน เมื่อทำการคอมไพล์เสร็จแล้ว ก็ต้องทำการปรับค่าต่าง ๆ ให้เป็นค่าที่เหมาะสมต่อไป ซึ่งจะเห็นได้ว่าขั้นตอนต่าง ๆ ค่อนข้างจะต้องใช้เวลาในการทำงานค่อนข้างนาน ดังนั้นเราจึงจะไม่ใช้วิธีนี้ในการติดตั้ง

ข้อดีของ Rocks Clusters เกี่ยวกับ Grid Computer ก็คือ Rocks ได้ทำการคอมไพล์ซอฟแวร์ต่าง ๆ ที่เกี่ยวข้องรวมทั้ง Grid Middleware แล้วทำการรวบรวมไว้เป็นชุดของซอฟแวร์เรียกว่า Grid Roll ดังนั้นการติดตั้งและการปรับค่า Configuration ต่าง ๆ ของ Grid Middleware หลังจากติดตั้งไปแล้วจึงสามารถทำได้ง่ายมาก เพราะ Rocks เองได้กำหนดค่าเบื้องต้นหลาย ๆ อย่างไว้ให้แล้ว รวมทั้งมี Initialized Script ในการกำหนดค่าให้อีกด้วย

ในขั้นตอนการติดตั้ง หากเป็นการติดตั้งลงบน Front-end ใหม่ที่ยังไม่เคยมีการติดตั้งเลย จะต้องทำการใส่แผ่น Grid Roll เข้าไปในระหว่างทำการติดตั้ง และทำการเลือกดังภาพ



รูปที่ 3.1 การเลือกติดตั้ง Grid Roll

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้น Globus จะถูกติดตั้งลงไปบนเครื่องโดยอัตโนมัติ แต่ถ้าหากจะทำการติดตั้ง Grid Roll ในเครื่อง Front-end ที่ได้ติดตั้งซอฟแวร์ต่าง ๆ ไปแล้วให้ทำการ Download ไฟล์ ISO ของ Grid Roll มา และให้ทำตามขั้นตอนดังนี้

```
$ su - root
# mount -o loop grid.iso /mnt/cdrom
# cd /home/install
# rocks-dist --install copyroll
# umount /mnt/cdrom
# rocks-dist dist
# kroll grid | bash
# restart
```

ข้อควรระวังของการติดตั้ง Grid Roll มีดังต่อไปนี้

- เครื่องที่ติดตั้ง Globus จะต้องมีชื่อเครื่องที่ถูกต้องใน DNS (Domain Name Service) และเป็น FQDN (Fully Qualified Domain Name) เสมอ นั่นหมายถึงว่า จะต้องสามารถทำการแปลงกลับไปกลับมาระหว่างชื่อกับ IP ได้อย่างสมบูรณ์ และชื่อที่ใช้ก็ต้องไปเป็น Alias Name ด้วย ตัวอย่างเช่น

```
$ nslookup blue65.crsc.kmitl.ac.th
Server:         161.246.52.21
Address:        161.246.52.21#53

Name:   blue65.crsc.kmitl.ac.th
Address: 161.246.34.100

$ nslookup 161.246.34.100
Server:         161.246.52.21
Address:        161.246.52.21#53

100.34.246.161.in-addr.arpa     name = blue65.crsc.kmitl.ac.th.
$
```

- การระบุชื่อเครื่องในตอนติดตั้ง Rocks จะต้องระบุชื่อเครื่อง และหมายเลข IP ให้ถูกต้องตรงกับค่าที่ควรจะเป็น เนื่องจากข้อมูลเหล่านี้ไม่สามารถแก้ไขได้โดยง่าย หลังจากการติดตั้งเสร็จสิ้นเรียบร้อยแล้ว เช่น

  ชื่อเครื่อง = blue65.crsc.kmitl.ac.th

  ประเทศ = Thailand

  Organization = KMITL

  Locality = Ladkrabang

  State = Bangkok

- ข้อมูลดังกล่าว จะถูกนำไปใช้ในการทำงานของซอฟต์แวร์อื่นๆในภายหลัง ถ้าหากมีการระบุผิดพลาด ไม่ตรงกับความเป็นจริงจะทำให้การทำงานไม่สมบูรณ์ เช่น การพิสูจน์ตัวตน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การติดตั้ง Simple CA

เมื่อ Grid Roll ถูกติดตั้งลงไปในระบบแล้ว Globus Simple Certificate Authority (CA) หรือ Simple CA ก็จะถูกติดตั้งลงไปบนระบบด้วยเช่นกัน โดย Simple CA นี้จะทำให้เราสามารถจัดการ สร้าง หรือให้การรับรองใบรับรองดิจิตอลได้ โดยขั้นตอนที่สำคัญจะเริ่มหลังจากติดตั้ง Grid Roll ไปแล้ว และทำการ Boot เครื่องขึ้นมา ระบบจะถาม Passphrase ของ CA ดังนี้

```
Enter CA passphrase:
Enter CA passphrase (again):
Installing Globus CA (takes a few minutes)
Done.
Stopping xinetd:                                    [ OK ]
Starting xinetd:                                    [ OK ]
Stopping MDS                                        [ OK ]
Starting MDS                                        [ OK ]
```

ในขั้นตอนนี้จะต้องทำการใส่ค่า Passphrase ลงไป โดยจะต้องเก็บไว้อย่างปลอดภัย และควรเป็นคนละค่ากับ Root Password ของระบบ

Simple CA นี้จะถูกติดตั้งพร้อมกับ Grid Roll เสมอ เนื่องจาก Simple CA มีประโยชน์ทำให้สามารถสร้าง หรือจัดการใบรับรอง เพื่อทำการทดสอบระบบกริดที่เราได้สร้างขึ้นมาได้อย่างง่าย ๆ แต่หากเป็นระบบกริดคอมพิวเตอร์จริงที่ใช้ในงานวิจัย ต่าง ๆ จะต้องใช้ Certificate จาก CA เดียวกัน ตามที่ได้ตกลงกันไว้ ซึ่งในโครงการนี้ศูนย์เทคโนโลยีกริดแห่งชาติ (Thaigrid) จะทำหน้าที่เป็น CA และแจกจ่ายในรับรองให้แก่หน่วยงานและมหาวิทยาลัยอื่น ๆ ซึ่งแต่ละหน่วยงานจะต้องทำตามขั้นตอนการขอใบรับรองและการติดตั้งใบรับรองในหัวข้อถัดไป

3.4 การขอและการติดตั้ง User Certificate

ในระบบกริดนั้นจะมีใบรับรองอยู่สองรูปแบบคือ Host Certificate และ User Certificate โดยจะมีความแตกต่างกันคือ Host Certificate จะใช้ในการยืนยันตัวตนของ Host นั้น ๆ ว่าเป็นเครื่องที่มีอยู่จริงและเป็นเครื่องที่เชื่อถือได้ ส่วน User Certificate จะใช้เพื่อการยืนยันตัวตนของผู้ใช้งาน โดยในขั้นตอนการขอใบรับรอง และการติดตั้งจะมีขั้นตอนแตกต่างกันเล็กน้อย และ Host Certificate จะถูกขอและติดตั้งโดยผู้ดูแลระบบของ Cluster นั้น ๆ

เมื่อทำการติดตั้ง Rocks Cluster ลงบนเครื่องใหม่นั้นจะยังไม่มีใบรับรอง User Certificate มาให้จะต้องทำการสร้างคำขอใบรับรองขึ้นมาก่อน โดยมีขั้นตอนในการขอและติดตั้ง User Certificate ดังนี้

1. ที่เครื่อง Front-end ต้องทำการติดตั้ง Certificate ของ CA ลงไปก่อน โดยในตัวอย่างนี้จะให้ blue65.crsc.kmitl.ac.th ทำหน้าที่เป็น CA (Certificate Authority) ให้กับ Grid ที่จะทำการสร้างขึ้นมา มีขั้นตอนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# wget http://blue65.crsc.kmitl.ac.th/simple-ca/globus_simple_ca_4de7aca7_setup-0.19.tar.gz
# gpt-build globus_simple_ca_4de7aca7_setup-0.19.tar.gz
# gpt-postinstall
# $GLOBUS_LOCATION/setup/globus_simple_ca_4de7aca7_setup /setup-gsi
# tg-sync-gridsecurity                    #ทำการ synchronize certificate ไปยัง computing node
```

2. ที่เครื่อง Front-end ให้ทำการเพิ่มรายชื่อผู้ใช้งานเข้าไปในระบบ เช่น user01 เป็นต้น

```
[root@blue65 ~]# useradd user01
[root@blue65 ~]# echo "user01" | passwd user01 --stdin     #ทำการ set password ให้กับ user01
[root@blue65 ~]# rocks-user-sync                #ทำการ synchronize user ไปยัง computing node ทุกเครื่อง
```

3. ทำการ Login เข้าเครื่อง Front-end โดยใช้รายชื่อผู้ใช้งานเป็น user01
4. ทำการสร้างคำร้องเพื่อขอใบรับรองสำหรับ user นั้น ๆ

```
[user01@blue65 ~]$ grid-cert-request -ca 4de7aca7

Using CA: 4de7aca7 - /O=Grid/OU=CRSC/OU=blue65.crsc.kmitl.ac.th/CN=Globus
Simple CA

Enter your name, e.g., John Smith: Nopparat Pantsaena          #ป้อนชื่อที่ต้องการ
A certificate request and private key is ..
..
..
obtain a new certificate.

Generating a 1024 bit RSA private key
.....++++++
writing new private key to '/home/user01/.globus/userkey.pem'
Enter PEM pass phrase:                                         #ป้อน pass phrase ที่ต้องการ
Verifying - Enter PEM pass phrase:
```

5. ทำการตรวจสอบไฟล์ที่ถูกสร้างขึ้น

```
[user01@blue65 ~]$ ls -dl ~/.globus/*
-rw-r--r--  1 user01 user01    0 Nov 12 23:20 /home/user01/.globus/usercert.pem
-rw-r--r--  1 user01 user01 1411 Nov 12 23:20 /home/user01/.globus/usercert_request.pem
-r--------  1 user01 user01  963 Nov 12 23:20 /home/user01/.globus/userkey.pem
```

6. ทำการส่งไฟล์ Certificate Request ไปยัง CA ทางอีเมล์
7. เมื่อ CA ได้รับข้อมูลการร้องขอใบรับรองจากผู้ใช้งาน จะทำการสร้างใบรับรองขึ้นมาด้วยวิธีการดังนี้

```
# grid-ca-sign -in usercert_request.pem -out usercert.pem
```

หลังจากนั้นจะทำการส่งใบรับรองกลับมาให้แก่ผู้ใช้งาน

8. ผู้ใช้งานทำการติดตั้งใบรับรองที่ได้รับ โดยการ Copy ไปวางไว้ใน .globus ซึ่งเป็นโฟล์เดอร์อยู่ในโฟล์เดอร์บ้าน (Home Folder) หลังจากนั้นให้ทำการตรวจสอบค่าภายใน Certificate ที่ได้มาโดยใช้คำสั่ง grid-cert-info

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[user01@blue65 ~]$ grid-cert-info -file ~/.globus/usercert.pem
Certificate:
   Data:
      Version: 3 (0x2)
      Serial Number: 3 (0x3)
      Signature Algorithm: md5WithRSAEncryption
      Issuer: O=Grid, OU=CRSC, OU=blue65.crsc.kmitl.ac.th, CN=Globus Simple CA
      Validity
         Not Before: Nov 12 16:40:41 2006 GMT
         Not After : Nov 11 16:40:41 2007 GMT
      Subject: O=Grid, OU=CRSC, OU=blue65.crsc.kmitl.ac.th, OU=crsc.kmitl.ac.th,
CN=Nopparat Pantsaena
      Subject Public Key Info:
         Public Key Algorithm: rsaEncryption
         RSA Public Key: (1024 bit)
            Modulus (1024 bit):
               00:bc:12:3d:ad:6a:88:82:7a:c5:3f:cc:73:7a:8f:
               b0:15:65:cd:b2:08:0d:51:e8:9e:c3:05:61:cc:84:
               4a:26:66:38:20:21:7e:39:ff:c6:3c:1c:10:85:43:
               85:e7:f8:17:15:be:a9:c8:5c:88:86:f8:0e:8b:c0:
               3f:53:37:dd:91:57:1b:f7:4b:ac:82:ae:bd:62:70:
               2c:51:e3:4b:7a:56:ed:0b:67:da:1a:a0:87:ca:36:
               73:1a:ad:c5:60:c1:2a:e3:de:21:c6:86:ad:a8:87:
               38:ae:0f:98:d3:f8:1d:48:16:7f:f6:9e:03:a3:ba:
               34:9c:78:c6:8c:29:9e:8b:53
            Exponent: 65537 (0x10001)
      X509v3 extensions:
         Netscape Cert Type:
            SSL Client, SSL Server, S/MIME, Object Signing
   Signature Algorithm: md5WithRSAEncryption
      8f:4b:ee:fa:10:6d:16:6d:9d:0c:67:4c:8f:ae:fc:89:cd:8c:
      c0:1f:36:aa:f9:f1:0c:35:c3:3c:45:99:f6:0a:8a:db:81:ec:
      a5:8e:07:94:69:b4:88:03:d8:8b:62:29:44:38:e3:72:f7:5d:
      1a:09:12:c4:89:9c:e8:cd:cd:e6:d0:a3:6b:0f:fb:f6:a8:e2:
      2e:a3:d1:ed:2a:3d:0b:43:93:2a:5d:f0:1a:76:3b:97:5b:7a:
      08:90:e7:0d:d1:2a:d3:93:fa:55:76:1c:58:3e:86:c4:a0:f2:
      c1:fd:9d:da:57:d0:ac:a7:b1:cf:e3:3a:32:58:38:99:a5:c7:
      30:69
```

สำหรับค่าที่สำคัญคือ

Issuer จะหมายถึง ผู้ออกใบรับรองนั้น ซึ่งก็คือ CA นั่นเอง

Subject หมายถึงผู้ถือใบรับรอง

9. ทำการทดสอบการทำงานของ Globus โดยใช้ Certificate ที่ได้รับมานั้น

```
[user01@blue65 ~]$ grid-proxy-init
Your identity: /O=Grid/OU=CRSC/OU=blue65.crsc.kmitl.ac.th/OU=crsc.kmitl.ac.th/CN=Nopparat
Pantsaena
Enter GRID pass phrase for this identity:
Creating proxy ..... Done
Your proxy is valid until: Tue Nov 13 11:45:23 2006
[user01@blue65 ~]$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง grid-proxy-init นั้น จะทำการสร้าง Proxy Credential หรือ**ใบรับรองตัวแทน**ขึ้นมาในระบบ เพื่อใช้ส่งงานต่าง ๆ เข้าไปในระบบกริดอีกที โดยปกติ ใบรับรองตัวจริง เมื่อจะใช้งานจะต้องทำการป้อน Passphase ก่อนทุกครั้ง แต่เมื่อเราทำการสร้างใบรับรองตัวแทนขึ้นมาในระบบ ใบรับรองตัวแทนจะถูกนำไปใช้งานได้โดยไม่ต้องป้อน Passphase อีกรอบ ทำให้การระบุตัวตน การเข้ารหัสต่าง ๆ เป็นไปโดยอัตโนมัติ แต่ใบรับรองตัวแทนนี้จะมีอายุจำกัด เช่นในตัวอย่างก็จะหมดอายุในวันที่ Tue Nov 13 11:45:23 2006 เป็นต้น

เราสามารถตรวจสอบค่าภายในของใบรับรองตัวแทนเพิ่มเติมได้โดยใช้คำสั่งต่อไปนี้

```
[user01@blue65 ~]$ grid-proxy-info
subject  : /O=Grid/OU=CRSC/OU=blue65.crsc.kmitl.ac.th/OU=crsc.kmitl.ac.th/CN=Nopparat
Pantsaena/CN=1046371927
issuer   : /O=Grid/OU=CRSC/OU=blue65.crsc.kmitl.ac.th/OU=crsc.kmitl.ac.th/CN=Nopparat
Pantsaena
identity : /O=Grid/OU=CRSC/OU=blue65.crsc.kmitl.ac.th/OU=crsc.kmitl.ac.th/CN=Nopparat
Pantsaena
type     : Proxy draft (pre-RFC) compliant impersonation proxy
strength : 512 bits
path     : /tmp/x509up_u502
timeleft : 11:59:43
```

คำสั่ง **grid-proxy-info** จะทำการแสดงรายละเอียดต่าง ๆ ของใบรับรองตัวแทน โดยบอกที่อยู่ของใบรับรองตัวแทน (PATH) และอายุการใช้งานที่เหลืออยู่ (Time left) เป็นต้น

หลังจากนั้นให้ทำการทดสอบความถูกต้องของใบรับรองนั้น โดยการส่งงานไปประมวลผลบนกริด ดังนี้

```
[user01@blue65 ~]$ globusrun -a -r blue65.crsc.kmitl.ac.th

GRAM Authentication test successful      --แสดงว่าเราสามารถติดต่อและสั่งงานผ่าน Grid Service ได้แล้ว
[user01@blue65 ~]$ globus-job-run blue65.crsc.kmitl.ac.th/jobmanager-sge /bin/uname
compute-0-0                              --สั่งงานผ่าน Grid Service โดยใช้ SGE เป็น Job Scheduler
[user01@blue65 ~]$
```

3.5 การขอและการติดตั้ง Host Certificate

การขอ Host Certificate นั้นจะต้องกระทำโดยผู้ดูแลระบบของ Cluster นั้น ๆ เท่านั้น โดยจะต้องทำตามขั้นตอนดังนี้

1. ที่เครื่อง Front-end ให้ทำการสร้างใบคำร้องขอ Certificate ดังนี้

```
# grid-cert-request -host `hostname` -ca 4de7aca7
```

โดยที่หมายเลข 4de7aca7 หมายถึง หมายเลข Simple CA ของผู้ออกใบรับรอง (CA) ที่ได้ติดตั้งไปก่อนหน้านี้ หลังจากนั้นทำการส่ง email ไฟล์ /etc/grid-security/hostcert_request.pem ซึ่งเป็นคำร้องขอใบรับรองไปยัง CA เมื่อ CA ได้รับจะทำการสร้างใบรับรองดังนี้

```
# grid-ca-sign -in hostcert_request.pem -out hostcert.pem
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมื่อได้รับไฟล์ใบรับรองจาก CA ให้ทำการติดตั้งที่เครื่อง Front-end โดยการเก็บไฟล์ไว้ที่ /etc/grid-security/hostcert.pem และทำการ Synchronize Certificate ไปยังเครื่องอื่น ๆ ในระบบโดยการใช้คำสั่ง

```
# tg-sync-gridsecurity
```

3. และให้ทำการดูรายละเอียดของ Host Certificate ที่ได้รับมานั้น

```
# grid-cert-info -file /etc/grid-security/hostcert.pem
Certificate:
   Data:
      Version: 3 (0x2)
      Serial Number: 1 (0x1)
      Signature Algorithm: md5WithRSAEncryption
      Issuer: O=Grid, OU=CRSC, OU=blue65.crsc.kmitl.ac.th, CN=Globus Simple CA
      Validity
         Not Before: Nov 12 00:50:26 2006 GMT
         Not After : Nov 11 00:50:26 2007 GMT
      Subject: O=Grid, OU=CRSC, OU=blue65.crsc.kmitl.ac.th,
CN=host/blue65.crsc.kmitl.ac.th
      Subject Public Key Info:
         Public Key Algorithm: rsaEncryption
         RSA Public Key: (1024 bit)
            Modulus (1024 bit):
               00:c3:1d:91:dd:05:fd:23:09:b0:46:f0:52:b8:17:
               17:42:c7:89:79:5b:93:d1:5b:39:39:bc:eb:cd:bb:
               ab:e2:4b:f4:0b:87:03:0b:02:f1:99:96:02:3b:a7:
               6e:42:0a:13:8f:54:b6:e7:14:82:0d:4c:58:c8:76:
               a1:90:5e:5e:fc:e1:23:b4:b1:05:ad:f6:57:75:a6:
               ca:8e:9c:57:50:9e:91:10:cd:a3:d7:d4:01:64:ee:
               a2:6b:91:cc:8e:27:1d:56:19:67:a9:86:67:e9:33:
               7f:9b:0d:3f:70:e6:5f:18:e3:59:5e:a5:82:70:aa:
               7c:3f:e5:c2:ea:9f:56:ca:51
            Exponent: 65537 (0x10001)
      X509v3 extensions:
         Netscape Cert Type:
            SSL Client, SSL Server, S/MIME, Object Signing
   Signature Algorithm: md5WithRSAEncryption
      44:94:cc:31:52:76:56:e1:ab:4f:1f:9c:68:a1:8b:2e:8b:1d:
      be:5c:14:b1:1d:bb:e1:aa:d9:58:bb:fc:58:97:98:d2:26:47:
      5b:92:c4:28:72:a6:db:0f:85:3c:b0:45:ab:21:2e:b2:b0:59:
      88:f9:71:a1:72:2b:cf:0c:80:52:d3:dc:57:6e:c5:62:5c:10:
      a2:77:74:a2:43:2b:a0:93:a9:6a:80:c5:33:11:55:96:95:46:
      42:fe:eb:aa:4b:80:fc:cc:a0:3b:b8:dc:7e:d2:54:3f:63:53:
      32:73:cc:05:f1:20:e0:5e:47:87:96:4f:08:1b:26:09:70:a9:
      e1:06
#
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การมอบหมายสิทธิ (Authorization)

ในการสั่งงานบนกริดคอมพิวเตอร์นั้น องค์ประกอบหนึ่งที่สำคัญคือการมอบหมายสิทธิ และการให้สิทธิ โดยที่ทุก Cluster จะต้องทำการกำหนดรายชื่อผู้ใช้งานกลางขึ้นมา เพื่อใช้เป็นตัวแทนในการสั่งงานต่าง ๆ ภายในกริด และจะต้องทำการกดหนดค่าให้แก่ Globus ด้วย โดยไฟล์ที่ใช้ในการมอบหมายสิทธินี้คือ /etc/grid-security/grid-mapfile ซึ่งเนื้อหาภายในไฟล์จะประกอบด้วยรายละเอียดดังตัวอย่าง

```
"/O=Grid/OU=KMITL/OU=blue65.crsc.kmitl.ac.th/OU=mclab.crsc.kmitl.ac.th/CN=Ann" griduser
"/O=Grid/OU=KMITL/OU=blue65.crsc.kmitl.ac.th/OU=mclab.crsc.kmitl.ac.th/CN=Trin" griduser
"/O=Grid/OU=KMITL/OU=blue65.crsc.kmitl.ac.th/OU=mclab.crsc.kmitl.ac.th/CN=kaew" griduser
```

โดยผู้ใช้งานทุกคน เมื่อทำการสั่งงานผ่านทางกริดจะได้รับสิทธิเป็นผู้ใช้งานระบบชื่อว่า griduser นั่นเอง (ขึ้นอยู่กับการตกลงร่วมกันภายในกริด) ดังนั้นทุก Cluster จะต้องทำการสร้างผู้ใช้งานกลางดังกล่าวขึ้นมา ตามขั้นตอนนี้

```
# useradd griduser                                   -- สร้าง user สำหรับการ Run คำสั่งใน grid ที่ front-end
# mkpasswd -l 20 | passwd griduser --stdin
# rocks-user-sync

# su - griduser
It doesn't appear that you have set up your ssh key.
This process will make the files:
    /home/griduser/.ssh/id_rsa.pub
    /home/griduser/.ssh/id_rsa
    /home/griduser/.ssh/authorized_keys

Generating public/private rsa key pair.
Enter file in which to save the key (/home/griduser/.ssh/id_rsa):                    --
Enter ผ่านเลย
Created directory '/home/griduser/.ssh'.
Enter passphrase (empty for no passphrase):                          -- Enter ผ่านเลย
Enter same passphrase again:                                         -- Enter ผ่านเลย
Your identification has been saved in /home/griduser/.ssh/id_rsa.
Your public key has been saved in /home/griduser/.ssh/id_rsa.pub.
The key fingerprint is:
9c:33:fe:d8:f9:65:83:d1:3f:34:51:5e:33:6f:c5:ac griduser@blue65.crsc.kmitl.ac.th
[griduser@blue65 ~]$ pwd
/home/griduser
[griduser@blue65 ~]$ exit
```

หลังจากนั้นให้ทำการ update grid-mapfile ตามแต่จะได้ตกลงกันไว้ในแต่ละกริด

4. สามารถทำการทดสอบการให้สิทธิบนบริการของกริดได้ดังนี้

```
[aue@blue65 ~]$ globusrun -a -r blue65.crsc.kmitl.ac.th --สามารถเปลี่ยนจาก blue65 เป็น
hostname ของ Cluster อื่น ๆ ภายในกริดเดียวกัน
GRAM Authentication test successful
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[aue@blue65 ~]$ globus-job-run blue65.crsc.kmitl.ac.th /opt/rocks/bin/cluster-fork
uname
compute-0-0: Linux
compute-0-1: Linux
compute-0-2: Linux
compute-0-3: Linux
[aue@blue65 ~]$ globus-job-run blue65.crsc.kmitl.ac.th /usr/bin/whoami
Griduser                                        --สิทธิของผู้ใช้งานที่ได้รับ
[aue@blue65 ~]$ globus-job-run blue65.crsc.kmitl.ac.th /bin/pwd
/home/griduser
[aue@blue65 ~]$ grid-proxy-destroy
```

ซึ่งจากตัวอย่างจะเห็นได้ว่าถึงแม้ผู้ใช้งานต้นทางจะเป็นรายชื่ออื่น แต่เมื่อสั่งประมวลผลงานบนกริดแล้ว สิทธิของผู้ใช้งานและกลายเป็น griduser ตามรายละเอียดใน grid-mapfile นั่นเอง

3.7 บทสรุป

ในบทนี้ได้กล่าวถึงการติดตั้ง Globus Toolkit แบบ Binary File ตามวิธีการของ Rocks Clusters ซึ่งถือว่าเป็นวิธีการที่ง่ายมากในปัจจุบัน รวมทั้งได้กล่าวถึงการติดตั้ง SimpleCA เพื่อทำการออกใบรับรองให้แก่ผู้ใช้งานต่าง ๆ (ในกรณีที่ทำหน้าที่เป็น Certificate Authority ด้วย) การติดตั้งใบรับรองสำหรับผู้ใช้งาน ใบรับรองสำหรับเครื่องคลัสเตอร์ และการมอบหมายสิทธิบนระบบกริด ซึ่งหัวข้อต่าง ๆ ในบทนี้ทำให้เกิดความกระจ่างเกี่ยวกับการติดตั้งค่าต่าง ๆ ของ Globus และระบบกริดคอมพิวเตอร์มากยิ่งขึ้นนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การใช้งานระบบกริดคอมพิวเตอร์

4.1 บทนำ

ในบทนี้จะได้กล่าวถึงการใช้งานกริดเป็นการเบื้องต้น โดยจะมีหัวข้อคือ การสั่งงานผ่านซอฟแวร์จัดลำดับงาน (Job Scheduler) ทั้งที่เป็นงานแบบลำดับ และงานแบบขนาน รวมทั้งจะกล่าวถึงการส่งงาน กริดผ่านทางเครื่องมือของ Globus อีกด้วย

4.2 การสั่งงานผ่านทางซอฟแวร์จัดลำดับงาน

4.2.1 งานแบบลำดับ (Sequential Job)

ในหัวข้อนี้จะได้แสดงตัวอย่างการส่ง Serial Application เพื่อไปประมวลผลบน Local Cluster โดยใช้ Job Scheduler ซึ่งในที่นี้จะใช้ SGE ทำหน้าที่เป็น Job Scheduler

1. ให้เขียนโปรแกรมต่อไปนี้ โดยกำหนดให้ไฟล์ชื่อ hello.c

```
#include <stdio.h>
int main(){
      system("/bin/hostname");
      printf("Hello world.\n");
}
```

2. ทำการ Compile โดยใช้คำสั่ง

```
gcc -o hello hello.c
```

3. สร้าง SGE script (Sun Grid Engine) เพื่อสั่งงาน Local Cluster โดยตั้งชื่อไฟล์เป็น hello-ser.job

```
#!/bin/bash

#$ -N sge_hello
#$ -cwd

./hello
```

โดยที่ตัวกำนดค่าที่อยู่ด้านหลังของเครื่องหมาย #$ จะเป็นการกำหนดค่าให้แก่ SGE เพื่อใช้ในการทำงาน โดยในตัวอย่างนี้

–N หมายถึง การกำหนดชื่อให้แก่งาน (Job) นั้น ๆ

-cwd หมายถึง ให้ SGE ใช้โฟล์เดอร์ปัจจุบันในการทำงาน (Current Working Directory)

4. สั่งงาน Local Cluster ผ่านทาง SGE

```
[aue@blue65 ~]$ qsub hello-ser.job
Your job 140 ("hello") has been submitted
[aue@blue65 ~]$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots ja-
task-ID
-----------------------------------------------------------------------------------------------------------
    140 0.00000 hello      aue          qw    11/27/2006 12:51:24                                    1
[aue@blue65 ~]$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots ja-
task-ID
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
-----------------------------------------------------------------------------------------------------------
   140 0.55500 hello     aue          r     11/27/2006 12:51:33 all.q@compute-0-1.local
1
[aue@blue65 ~]$ qstat
[aue@blue65 ~]$
```

qsub จะเป็นคำสั่งที่ถูกใช้ในการส่งงานเข้าไปในคิว (Queue) ของ SGE เพื่อรอการประมวลผลต่อไป ส่วน qstat จะเป็นคำสั่งเพื่อใช้ดูสถานะของงานนั้น ๆ ว่าอยู่ในสถานะไหน โดย "qw" หมายถึง กำลังรอดำเนินการอยู่ ส่วน "r" หมายถึง งานนั้นถูกสั่งให้ทำงานเรียบร้อยแล้ว

5. ตรวจสอบผลลัพธ์จากการทำงาน

```
[aue@blue65 ~]$ ll *140
-rw-r--r--  1 aue aue  0 Nov 27 12:51 sge_hello.e140
-rw-r--r--  1 aue aue 31 Nov 27 12:51 sge_hello.o140
[aue@blue65 ~]$ more sge_hello.o140
compute-0-1.local
Hello world.
[aue@blue65 ~]$
```

4.2.2 งานแบบขนาน (Parallel Job)

ในการส่งงานแบบขนานไปประมวลผลบนคลัสเตอร์นั้นจะต้องมีการกำหนดรายละเอียดเพิ่มเติม เพื่อให้ SGE สามารถเข้าใจว่างานนั้น ๆ เป็นงานแบบขนานและจะต้องสั่งให้ทำงานแบบขนาน ขั้นตอนต่าง ๆ สามารถแสดงได้ดังตัวอย่าง

1. ให้เขียนโปรแกรมต่อไปนี้ โดยกำหนดให้ไฟล์ชื่อ mpi-hello.c

```
#include <stdio.h>
#include <mpi.h>
int main(int argc, char **argv) {
      int nprocs, rank;
      MPI_Init(&argc, &argv);
      MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
      MPI_Comm_rank(MPI_COMM_WORLD, &rank);
      printf("hello world from %d of %d\n", rank, nprocs);
      MPI_Finalize();
}
```

2. ทำการ Compile โดยใช้คำสั่ง

```
/opt/mpich/gnu/bin/mpicc -o mpi-hello mpi-hello.c
```

3. ทำการสั่งงานบน Local Cluster โดยใช้คำสั่ง

```
[aue@blue65 ~]$ /opt/mpich/gnu/bin/mpirun -np 4 mpi-hello
hello world from 0 of 4
hello world from 3 of 4
hello world from 1 of 4
hello world from 2 of 4
[aue@blue65 ~]$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการสร้าง SGE script (Sun Grid Engine) เพื่อสั่งงาน Local Cluster

```
#!/bin/bash

### Change to the current working directory:
#$ -cwd

### Job name:
#$ -N sge_mpi_hello

### Specify SHELL
#$ -S /bin/bash

### Use Parallel Environment mpi, 4 processes:
#$ -pe mpi 4

### Export some environment varaibles:
#$ -v MPICH_HOME=/opt/mpich/gnu

#Start MPI Program
$MPICH_HOME/bin/mpirun -machinefile $TMPDIR/machines -np $NSLOTS mpi-hello
```

5. ส่งงานไป Run ใน Local Cluster ผ่านทาง SGE Job Scheduler

```
[aue@blue65 ~]$ qsub mpi-hello.job
Your job 142 ("sge_mpi_hello") has been submitted
[aue@blue65 ~]$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
    142 0.00000 sge_mpi_he aue          qw    11/27/2006 12:58:57                                    4
[aue@blue65 ~]$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
    142 0.55500 sge_mpi_he aue          r     11/27/2006 12:59:03 all.q@compute-0-3.local        4
[aue@blue65 ~]$ qstat
[aue@blue65 ~]$
```

ผลลัพธ์จากคำสั่ง "qstat" ทำให้ทราบสถานะของงานที่อยู่ในคิวรวมทั้งจำนวน SLOT ทั้งหมดที่ถูกสร้างขึ้นมา เพื่อใช้ในการทำงานนี้อีกด้วย ซึ่งก็จะเห็นได้ว่าเป็นค่าเดียวกับที่กำหนดไว้ในไฟล์ SGE Script นั่นเอง

6. ตรวจสอบผลลัพธ์การทำงาน

```
[aue@blue65 ~]$ ll *142
-rw-r--r--  1 aue aue   0 Nov 27 12:59 sge_mpi_hello.e142
-rw-r--r--  1 aue aue  96 Nov 27 12:59 sge_mpi_hello.o142
-rw-r--r--  1 aue aue  68 Nov 27 12:59 sge_mpi_hello.pe142
-rw-r--r--  1 aue aue 120 Nov 27 12:59 sge_mpi_hello.po142
[aue@blue65 ~]$ more sge_mpi_hello.o142
hello world from 3 of 4
hello world from 1 of 4
hello world from 2 of 4
hello world from 0 of 4
[aue@blue65 ~]$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การสั่งงานกริดผ่านทางเครื่องมือของ Globus

4.3.1 การสั่งงานกริดแบบ Interactive และ Batch

ในการส่งงานไปประมวลผลในระบบกริดนั้นจะมีรูปแบบการสั่งงานอยู่สองรูปแบบคือ แบบที่ตอบสนองการสั่งงานในทันที (Interactive) และแบบทำการประมวลผลภายหลัง (Batch) ซึ่งเป็นงานที่ต้องใช้เวลาในการประมวลผลมากไม่ต้องการรอการตอบสนองจากระบบ และสามารถมาเก็บผลลัพธ์การทำงานภายหลังได้ ซึ่งคำสั่งที่ใช้ในการสั่งงานทั้งหมดมีดังนี้

Interactive Command

คำสั่ง	คำอธิบาย
globus-job-run	ส่งงานไปประมวลผลในระบบกริด

Batch Command

คำสั่ง	คำอธิบาย
globus-job-submit	ส่งงานไปประมวลผลในระบบกริด
globus-job-status	แสดงสถานะการทำงาน
globus-job-get-output	แสดงค่าผลลัพธ์ที่เกิดขณะทำงานทาง Standard Output
globus-job-cancel	ยกเลิกงานที่ประมวลผล
globus-job-clean	ทำการลบงานออกจากระบบกริด

การสั่งงานกริดแบบ Interactive

ในการสั่งงานกริดแบบ Interactive นั้นสามารถแสดงตัวอย่างขั้นตอนการทำงานได้ดังนี้

1. ก่อนที่จะทำการส่งงานไปประมวลผลในระบบกริดทุกครั้ง จะต้องทำการสร้างสร้างใบรับรองตัวแทน (Credential) ขึ้นมาก่อนเสมอ โดยใช้คำสั่งต่อไปนี้

```
[std01@gridmd ~]$ grid-proxy-init
Your identity: /O=Grid/OU=ThaiGrid/OU=cpe.ku.ac.th/CN=mike-training
Enter GRID pass phrase for this identity:                    --ให้ใส่ Passphase
Creating proxy ..................................... Done
Your proxy is valid until: Wed Nov  7 10:06:38 2006
[std01@gridmd ~]$
```

2. ทำการตรวจสอบค่าของใบรับรองตัวแทน

```
[std01@gridmd ~]$ grid-proxy-info
subject  : /O=Grid/OU=ThaiGrid/OU=cpe.ku.ac.th/CN=mike-training/CN=1935716498
issuer   : /O=Grid/OU=ThaiGrid/OU=cpe.ku.ac.th/CN=mike-training
identity : /O=Grid/OU=ThaiGrid/OU=cpe.ku.ac.th/CN=mike-training
type     : Proxy draft (pre-RFC) compliant impersonation proxy
strength : 512 bits
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
path    : /tmp/x509up_u511
timeleft : 11:57:21
[std01@gridmd ~]$
```

3. ทำการสั่งงาน GridMD ซึ่งเป็น Local Cluster โดยใช้คำสั่งต่อไปนี้

```
[std01@gridmd ~]$ globus-job-run gridmd.kmitl.ac.th /opt/rocks/bin/cluster-fork
uname
compute-0-0: Linux
compute-0-1: Linux
compute-0-2: Linux
compute-0-3: Linux
[std01@gridmd ~]$ globus-job-run gridmd.kmitl.ac.th /usr/bin/whoami
thaigrid
[std01@gridmd ~]$ globus-job-run gridmd.kmitl.ac.th /bin/pwd
/home/thaigrid
[std01@gridmd ~]$
```

4. ทำการสั่งงานไปยัง Cluster อื่นที่อยู่ภายใน Thaigrid เช่น sunyata.thaigrid.or.th

```
[std01@gridmd ~]$ globus-job-run sunyata.thaigrid.or.th /opt/rocks/bin/cluster-fork
uname
compute-0-0: Linux
compute-0-1: Linux
compute-0-2: Linux
compute-0-3: Linux
[std01@gridmd ~]$ globus-job-run sunyata.thaigrid.or.th /usr/bin/whoami
thaigrid
[std01@gridmd ~]$ globus-job-run sunyata.thaigrid.or.th /bin/pwd
/home/thaigrid
[std01@gridmd ~]$ grid-proxy-destroy
```

ซึ่งจะเห็นได้ว่าเมื่อทำการส่งงานไปประมวลผลโดยคำสั่ง "globus-job-run" แล้ว หลังจากนั้น งานจะถูกสั่งประมวลผลทันที และจะได้ผลลัพธ์ตอบสนองกลับมาภายในเวลาไม่นานนัก ซึ่งปกติการสั่งงานแบบนี้ใช้เพื่อเป็นการทดสอบระบบ หรือการสั่งงานเล็ก ๆ น้อย ๆ เท่านั้น แต่หากเป็นงานที่ต้องใช้เวลาในการประมวลผลนานจะต้องใช้การสั่งงานแบบ Batch แทน

การสั่งงานกริดแบบ Batch

สามารถแสดงตัวอย่างได้ดังต่อไปนี้

1. ทำการสั่งงานไปยัง GridMD ซึ่งเป็น Local Cluster แบบ Batch ผ่านทางคำสั่งต่อไปนี้

```
[std01@gridmd ~]$ grid-proxy-init
[std01@gridmd ~]$ globus-job-submit gridmd.kmitl.ac.th /bin/hostname
https://gridmd.kmitl.ac.th:10004/26992/1194362274/
[std01@gridmd ~]$ globus-job-status
https://gridmd.kmitl.ac.th:10004/26992/1194362274/  -ได้มาจากผลลัพธ์ของคำสั่ง globus-job-submit

DONE
[std01@gridmd ~]$ globus-job-get-output
https://gridmd.kmitl.ac.th:10004/26992/1194362274/
gridmd.kmitl.ac.th
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[std01@gridmd ~]$ globus-job-clean
https://gridmd.kmitl.ac.th:10004/26992/1194362274/
   WARNING: Cleaning a job means:
      - Kill the job if it still running, and
      - Remove the cached output on the remote resource
   Are you sure you want to cleanup the job now (Y/N) ?
y
Cleanup successful.
[std01@gridmd ~]$
```

2. ทำการสร้าง Shell Script ขึ้นมาดังนี้

```
[std01@gridmd ~]$ cat > lazy.sh
#!/bin/bash

echo "Hi.. I'm a lazy boy."
sleep 2000
ctrl+d                    ---หมายถึง กดปุ่ม Ctrl พร้อมกับ "d" บนแป้นพิมพ์
[std01@gridmd ~]$ chmod +x lazy.sh
[std01@gridmd ~]$ ls -dl lazy.sh
-rwxrwxr-x  1 std01 std01 104 Nov  6 22:25 lazy.sh
[std01@gridmd ~]$
```

3. ทำการสั่งให้ lazy.sh ทำงานบน Sunyata Cluster

```
[std01@gridmd ~]$ globus-job-submit sunyata.thaigrid.or.th –s lazy.sh
https://sunyata.thaigrid.or.th:10152/6327/1194364079/
[std01@gridmd ~]$ globus-job-status
https://sunyata.thaigrid.or.th:10152/6327/1194364079/
ACTIVE
[std01@gridmd ~]$ globus-job-get-output
https://sunyata.thaigrid.or.th:10152/6327/1194364079/
Hi.. I'm a lazy boy.
[std01@gridmd ~]$ globus-job-cancel
https://sunyata.thaigrid.or.th:10152/6327/1194364079/
   Are you sure you want to cancel the job now (Y/N) ?
y
Job canceled.
NOTE: You still need to clean files associated with the
job by running globus-job-clean <jobID>
[std01@gridmd ~]$ globus-job-clean
https://sunyata.thaigrid.or.th:10152/6327/1194364079/
   WARNING: Cleaning a job means:
      - Kill the job if it still running, and
      - Remove the cached output on the remote resource
   Are you sure you want to cleanup the job now (Y/N) ?
y
Cleanup successful.
[std01@gridmd ~]$ grid-proxy-destroy
```

จากตัวอย่างที่แสดงด้านบนทำให้เห็นภาพเกี่ยวกับการสั่งงานกริดทั้งสองรูปแบบคือ แบบ Interactive และ Batch ซึ่งในหัวข้อต่อไปจะกล่าวถึงการสั่งงานกริดร่วมกับการเรียกใช้ Local Job Scheduler บน Cluster นั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 การสั่งงาน Grid ผ่านทางตัวจัดการลำดับงาน

ในหัวข้อที่ผ่านมาเป็นการสั่งงานโดยตรงไปยังเครื่อง Front-end ของคลัสเตอร์นั้น ๆ และ Front-end จะเป็นเครื่องที่ทำหน้าที่ในการประมวลผลงานนั้น ๆ ให้ แต่เพื่อให้เกิดการทำงานอย่างเหมาะสม เราควรจะต้องสั่งงานไปยังคลัสเตอร์ใด ๆ ภายในกริดผ่านทางตัวจัดการลำดับงาน (Job Scheduler) เสมอ เพื่อให้ตัวจัดการลำดับงานบนคลัสเตอร์นั้น ๆ เก็บงานไว้ในคิวก่อนการสั่งประมวลผล (Execute) และหาเครื่องที่เหมาะสมที่สุดในการประมวลผลงานนั้น ๆ ต่อไป

การส่ง Serial Application เพื่อประมวลผลบน Grid

1. ทำการส่งงานไปประมวลผลบน Grid ผ่านทางเครื่องมือของ Globus

```
[aue@gridmd ~]$ grid-proxy-init
[aue@gridmd ~]$ globus-job-run araya.thaigrid.or.th /bin/hostname
araya.thaigrid.or.th
[aue@gridmd ~]$ globus-job-run araya.thaigrid.or.th/jobmanager-sge /bin/hostname
-catch_rsh /opt/gridengine/default/spool/compute-0-0/active_jobs/3900.1/pe_hostfile
compute-0-0
/opt/gridengine/bin/lx26-amd64/qrsh -V -inherit compute-0-0 /bin/sh
/home/thaigrid/.globus/job/araya.thaigrid.or.th/25179.1196139140/scheduler_sge_cmd_sc
ript
compute-0-0.local
[aue@gridmd ~]$ globus-job-run araya.thaigrid.or.th/jobmanager-sge /bin/hostname
-catch_rsh /opt/gridengine/default/spool/compute-0-1/active_jobs/3901.1/pe_hostfile
compute-0-1
/opt/gridengine/bin/lx26-amd64/qrsh -V -inherit compute-0-1 /bin/sh
/home/thaigrid/.globus/job/araya.thaigrid.or.th/25211.1196139168/scheduler_sge_cmd_sc
ript
compute-0-1.local
[aue@gridmd ~]$ grid-proxy-destroy
```

จากตัวอย่างนี้จะเห็นได้ว่า เมื่อทำการสั่งงานกริดผ่านทาง Job Scheduler แม้จะเป็นการสั่งงานคำสั่งเดียวกัน แต่สั่งคนละครั้ง เครื่องที่ถูกเรียกให้ทำงานอาจจะไม่ใช่เครื่องเดียวกัน เพราะการตัดสินใจให้ทำงานที่เครื่องใด ๆ จะเป็นการตัดสินใจจาก Job Scheduler ซึ่งในที่นี้คือ SGE นั่นเอง

2. ทำการสร้าง Shell Script ขึ้นมาดังนี้

```
#!/bin/bash
        uname
        echo "Hello world."
```

3. ทำการส่งงานไปประมวลผลบนกริดโดยใช้คำสั่งต่อไปนี้

```
[std02@gridmd ~]$ grid-proxy-init
[std02@gridmd ~]$ globus-job-run araya.thaigrid.or.th -s hello
araya.thaigrid.or.th
Hello world.
[std02@gridmd ~]$ globus-job-run araya.thaigrid.or.th/jobmanager-sge -s hello
-catch_rsh /opt/gridengine/default/spool/compute-0-2/active_jobs/3902.1/pe_hostfile
compute-0-2
/opt/gridengine/bin/lx26-amd64/qrsh -V -inherit compute-0-2 /bin/sh
/home/thaigrid/.globus/job/araya.thaigrid.or.th/26099.1196140388/scheduler_sge_cmd_sc
ript
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
compute-0-2.local
Hello world.
[std02@gridmd ~]$ globus-job-run araya.thaigrid.or.th/jobmanager-fork -s hello
araya.thaigrid.or.th
Hello world.
[std02@gridmd ~]$ grid-proxy-destroy
```

จากตัวอย่าง เราได้ทำการส่งงานไปประมวลผลบนกริด โดยเรียกใช้ Job Scheduler ที่แตกต่างกันสองตัวคือ jobmanager-fork และ jobmanager-sge ซึ่งการสั่งงานผ่าน jobmanager-fork นั้น เครื่อง Front-end จะเป็นเครื่องหลักในการทำงาน และอาจมีการแตก Process เพิ่มเติม (Fork) ส่วน jobmanager-sge นั้นจะไปทำการเรียก SGE บนคลัสเตอร์นั้น ๆ ขึ้นมาทำงานและตัดสินใจในการเรียกสมาชิกใด ๆ ภายในคลัสเตอร์ขึ้นมาทำงาน ดังนั้นโดยปกติแล้วควรจะสั่งงานกริดผ่านทาง jobmanager-sge มากกว่า เพื่อประโยชน์ในการตัดสินใจเลือกเครื่องที่เหมาะสมที่สุดในการประมวลผลงานต่อไป

การส่ง Parallel Application เพื่อประมวลผลบน Grid โดยใช้ Job Scheduler

1. เขียนโปรแกรมต่อไปนี้ โดยกำหนดให้ไฟล์ชื่อ getprocname.c

```
#include <stdio.h>
#include <stdlib.h>
#include "mpi.h"

int main(int argc, char **argv){

int id, size;
int namelen;
char processor_name[MPI_MAX_PROCESSOR_NAME];

if (MPI_Init (&argc, &argv) != MPI_SUCCESS) {
      fprintf(stdout, "Error to start MPI!!\n");
      exit(-1);
}

MPI_Comm_size(MPI_COMM_WORLD,&size);
MPI_Comm_rank (MPI_COMM_WORLD,&id);

MPI_Get_processor_name (processor_name,&namelen);

fprintf (stdout,"Process %d of %d on %s\n",  id, size, processor_name);
fflush(stdout);

MPI_Finalize();
return(0);
}
```

2. ทำการ Compile โดยใช้คำสั่ง

```
/opt/mpich/gnu/bin/mpicc -o getprocname getprocname.c
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ส่งงานไปประมวลผลบน Grid โดยใช้คำสั่ง

```
[aue@blue65 ~]$ grid-proxy-init
[aue@blue65 ~]$ globus-job-run araya.thaigrid.or.th/jobmanager-sge -x
'(jobtype=mpi)' -np 4 -s /home/aue/getprocname
Process 0 of 4 on compute-0-1.local
Process 1 of 4 on compute-0-1.local
Process 3 of 4 on compute-0-2.local
Process 2 of 4 on compute-0-2.local
[aue@blue65 ~]$
```

4. ส่งงานไปประมาลผลบนกริดโดยใช้ Batch Mode

```
[aue@blue65 ~]$ globus-job-submit araya.thaigrid.or.th/jobmanager-sge -x
'(jobtype=mpi)' -np 4 -s /home/aue/getprocname
https://araya.thaigrid.or.th:35921/28910/1196145607/
[aue@blue65 ~]$ globus-job-status
https://araya.thaigrid.or.th:35921/28910/1196145607/
DONE
[aue@blue65 ~]$ globus-job-get-output
https://araya.thaigrid.or.th:35921/28910/1196145607/
Process 0 of 4 on compute-0-0.local
Process 1 of 4 on compute-0-0.local
Process 2 of 4 on compute-0-3.local
Process 3 of 4 on compute-0-3.local
[aue@blue65 ~]$ grid-proxy-destroy
```

การส่ง Parallel Application เพื่อประมวลผลบน Grid โดยใช้

RSL (Resource Specification Language)

การสั่งงานกริดด้วย RSL ก็เป็นอีกวิธีการหนึ่งที่สามารถทำได้ง่าย เพราะจะเป็นการกำหนดรายละเอียดต่าง ๆ ไว้ในไฟล์เหมือน SGE Script แล้วส่งงานเข้าไปในกริด ทำให้สะดวกต่อการสั่งงานใหม่ หรือการแก้ไขในภายหลัง ขั้นตอนต่าง ๆ สามารถทำได้ตามตัวอย่างดังนี้

1. ทำการ Download ไฟล์ MPI_Exercises.zip จาก http://lcms.kmitl.ac.th/bounce.php?course=190 และแตกไฟล์โดยใช้คำสั่ง

```
unzip MPI_Exercises.zip
```

2. ทำการ Compile โปรแกรม 2D FFT ดังนี้

```
cd MPI_Exercises/2dfft/
/opt/mpich/gnu/bin/mpicc -o mpi_2dfft mpi_2dfft.c -lm
```

3. สร้างไฟล์ RSL ให้มีรายละเอียดดังนี้

```
&    (resourceManagerContact="blue65.crsc.kmitl.ac.th")
     (label="mpi_2dfft")
     (count="4")
     (jobtype="mpi")
     (executable=$(GLOBUSRUN_GASS_URL) # "/home/aue/MPI_Exercises/2dfft/mpi_2dfft")
```

โดยบรรทัดสุดท้ายให้เปลี่ยนเป็น Home Folder หรือ Folder ที่ใช้ในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สั่งงานไปประมวลผลบน Grid โดยใช้คำสั่งต่อไปนี้

```
[aue@blue65 2dfft]$ grid-proxy-init
[aue@blue65 2dfft]$ globusrun -r araya.thaigrid.or.th/jobmanager-sge -f
/home/aue/MPI_Exercises/2dfft/mpi_2dfft.rsl -s
compute-0-2
compute-0-2
compute-0-1
compute-0-1

MPI task ID = 1
MPI task ID = 3
MPI task ID = 2
MPI task ID = 0
512 x 512 2D FFT
                    Timings(secs)
      scatter  1D-FFT-row  transpose 1D-FFT-col  gather      total
cell 1:  0.013925  0.014051  0.013784  0.015193  0.001582    0.058535
cell 3:  0.015249  0.014004  0.010946  0.014992  0.003639    0.058830
cell 2:  0.015419  0.014057  0.012274  0.014868  0.014136    0.070754
cell 0:  0.013918  0.014054  0.012142  0.015466  0.028848    0.084428
Total Mflops= 279.4447
[aue@blue65 2dfft]$ grid-proxy-destroy
```

4.3.3 การรับ-ส่งไฟล์ผ่านทาง GridFTP และ Remote Login ผ่าน GSI-SSH

GridFTP เป็นบริการที่ใช้เพื่อการรับ-ส่งไฟล์ผ่านทางบริการของกริด ส่วน GSI-SSH นั้นเป็นบริการที่ใช้เพื่อทำการ Remote Login ไปยังเครื่องปลายทางผ่านบริการของกริด ทำให้ไม่ต้องมีการพิสูจน์ตัวตนใหม่อีกครั้ง ขั้นตอนการใช้งานสามารถแสดงได้ดังตัวอย่าง

```
[std01@gridmd ~]$ grid-proxy-init
[std01@gridmd ~]$ globus-url-copy file:///home/std01/lazy.sh
gsiftp://sunyata.thaigrid.or.th/home/thaigrid/
[std01@gridmd ~]$ gsissh -p 12222 sunyata.thaigrid.or.th   --เป็นการ Login ไปยังเครื่องปลาย
ทางผ่านทาง Service ของ Grid
Last login: Tue Nov  6 21:48:12 2006 from gridmd.kmitl.ac.th
Rocks Frontend Node - sunyata Cluster
Rocks 4.2.1 (Cydonia)
Profile built 11:30 19-Oct-2006
Kickstarted 18:53 19-Oct-2006
[thaigrid@sunyata ~]$ pwd   --ตอนนี้อยู่ที่เครื่อง sunyata และ user std01 ถูก map เป็น user ชื่อว่า thaigrid
/home/thaigrid
[thaigrid@sunyata ~]$ ls -dl lazy.sh
-rw-r--r--  1 thaigrid thaigrid 104 Nov  6 22:29 lazy.sh
[thaigrid@sunyata ~]$ chmod +x lazy.sh
[thaigrid@sunyata ~]$ ls -dl lazy.sh
-rwxr-xr-x  1 thaigrid thaigrid 104 Nov  6 22:29 lazy.sh*
[thaigrid@sunyata ~]$ exit
logout
Connection to sunyata.thaigrid.or.th closed.
[std01@gridmd ~]$                    --ตอนนี้กลับมาที่เครื่อง gridmd
[std01@gridmd ~]$ grid-proxy-destroy
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 การสั่งงานผ่านทาง Web Service

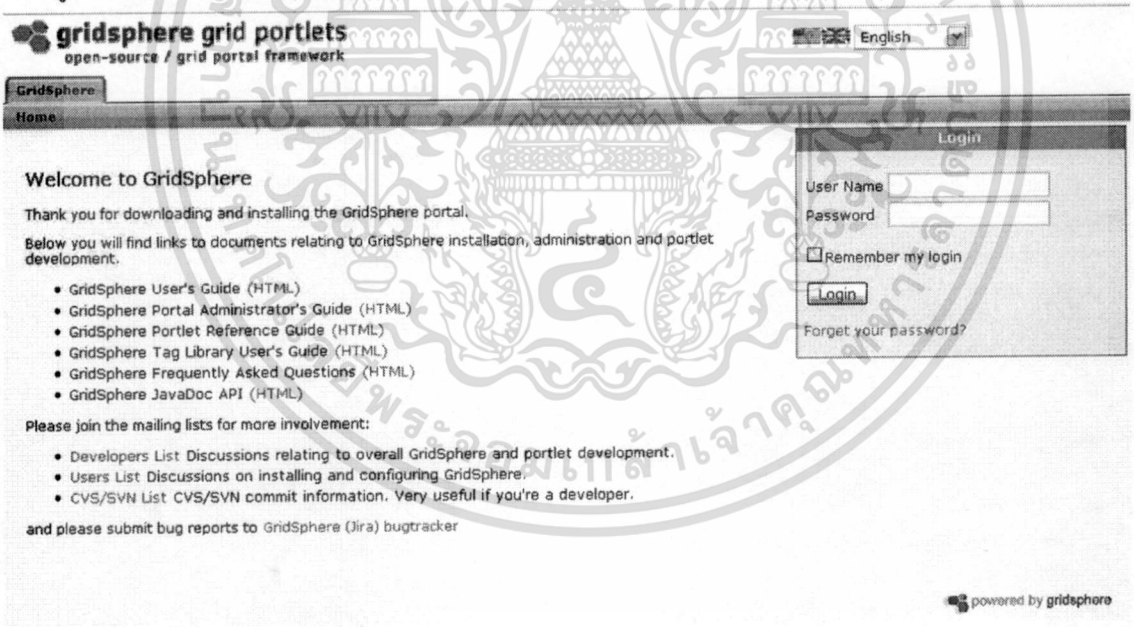
ในการสั่งงานกริดคอมพิวเตอร์นั้น ถ้าหากเป็นกริดที่ถูกสร้างขึ้นมาโดยใช้ Globus Toolkit เวอร์ชั่น 2 จะต้องทำการสั่งงานผ่านทาง GRAM ซึ่งเป็นช่องทางปกติ ของการสั่งงานกริด แต่ถ้าเป็น Globus Toolkit เวอร์ชั่น 4 เป็นต้นไป สามารถสั่งงานต่าง ๆ ผ่านทาง Web Service ได้โดยคำสั่งต่อไปนี้

```
# globusrun-ws -submit -factory  blue65.crsc.kmitl.ac.th -Ft SGE -J -S -s -c /bin/hostname
```

ซึ่งคำสั่ง globusrun-ws นี้จะให้ผลลัพธ์ในการทำงานค่อนข้างละเอียด และมีเสถียรภาพในการทำงานค่อนข้างมาก สามารถใช้เพื่อการค้นหาปัญหา หรือการแก้ไขปัญหาต่าง ๆ ได้ดี

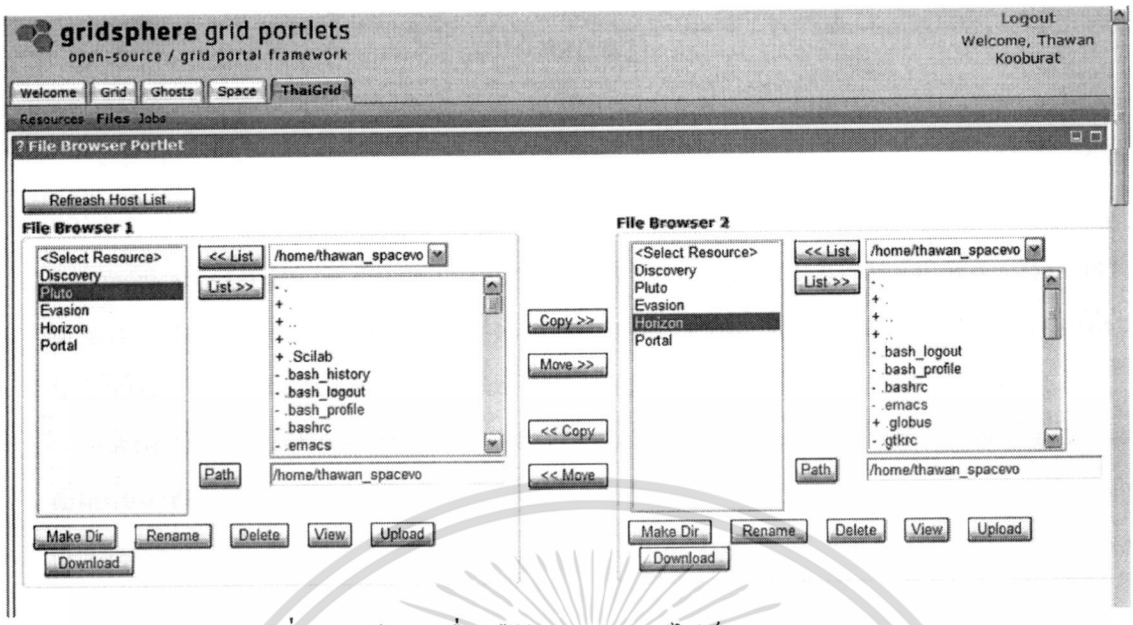
4.3.5 การสั่งงานผ่านทาง Grid Portal

เราสามารถสั่งงานกริดได้อีกช่องทางหนึ่งคือ ผ่านทาง Grid Portal โดยรูปแบบการใช้งานจะเหมือนกับการใช้งาน Web Application ทั่วไปไม่ได้มีความยุ่งยากซับซ้อนมาก ทำให้ผู้ใช้งานสามารถสั่งงานต่าง ๆ ได้ง่ายขึ้น เป็นการอำนวยความสะดวกให้มีการใช้งานกริดคอมพิวเตอร์เพิ่มอีกช่องทางหนึ่งด้วย รูปแบบของ Grid Portal สามารถแสดงได้ดังรูป

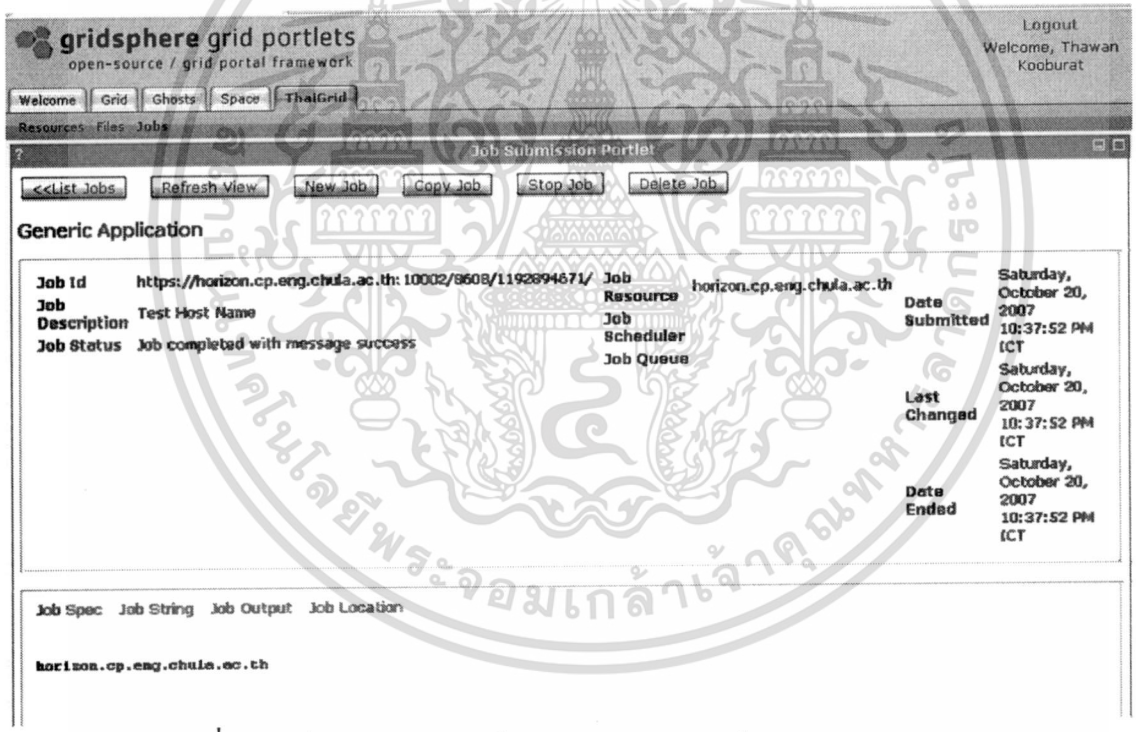


รูปที่ 4.1 หน้าจอในการ Login ของ Grid Portal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 หน้าจอเครื่องมือในการจัดการไฟล์ของ Grid Portal



รูปที่ 4.3 หน้าจอแสดงสถานะในการทำงานของงานต่าง ๆ บน Grid Portal

4.4 บทสรุป

ในบทนี้ได้แสดงให้เห็นเกี่ยวกับการสั่งงานกริดในรูปแบบและวิธีการต่าง ๆ เพื่อให้สามารถนำไปประยุกต์ใช้ในงานอื่น ๆ ต่อไปได้ ไม่ว่าจะเป็นการสั่งงานผ่านทาง Job Scheduler แบบต่าง ๆ ทั้งงานที่เป็นแบบ Serial และงานแบบ Parallel การสั่งงานผ่านทาง Web Service และรูปแบบการสั่งงานอีกอย่างหนึ่งที่สามารถทำได้ง่าย ๆ คือ การสั่งงานผ่านทางกริด Portal นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้