การประยุกต์ใช้เทคโนโลยีกริด เพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ

โดย

นายธรรมรงค์ กำมะณี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

ภาควิชาคอมพิวเตอร์

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ปีการศึกษา 2552

ลิขสิทธิ์ของบัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

การประยุกต์ใช้เทคโนโลยีกริด เพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ

โดย

นายธรรมรงค์ กำมะณี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

ภาควิชาคอมพิวเตอร์

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ปีการศึกษา 2552

ลิขสิทธิ์ของบัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

APPLIED GRID TECHNOLOGY FOR ANIMATION RENDERING IN THREE DIMENSION GAME

By

Thammarong Kummanee

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

Department of Computing

Graduate School

SILPAKORN UNIVERSITY

2009

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร อนุมัติให้วิทยานิพนธ์เรื่อง “ การประยุกต์ใช้เทคโนโลยีกริด เพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ ” เสนอโดย นายธรรมรงค์ กำมะณี เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

……..

(รองศาสตราจารย์ ดร.ศิริชัย ชินะตังกูร)

คณบดีบัณฑิตวิทยาลัย

วันที่.....เดือน..... พ.ศ.....

อาจารย์ที่ปรึกษาวิทยานิพนธ์

อาจารย์ ดร.สุนีย์ พงษ์พินิจภิญโญ

คณะกรรมการตรวจสอบวิทยานิพนธ์

..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ปานใจ ธารทัศนวงศ์)

...../...../.....

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ภุชงค์ อุทโยภาศ)

...../...../.....

..... กรรมการ

(อาจารย์ ดร.สุนีย์ พงษ์พินิจภิญโญ)

...../...../.....

47307306 : สาขาวิชาวิทยาการคอมพิวเตอร์
คำสำคัญ : การประมวลผลแบบกริด/การประมวลผลภาพ 3 มิติ

ธรรมรงค์ กำมะณี : การประยุกต์ใช้เทคโนโลยีกริด เพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ. อาจารย์ที่ปรึกษาวิทยานิพนธ์ : อ.ดร.สุนีย์ พงษ์พินิจภิญโญ. 63 หน้า.

วิทยานิพนธ์เรื่อง “การประยุกต์ใช้เทคโนโลยีกริดเพื่อสร้างภาพเคลื่อนไหวในเกม3 มิติ” มีวัตถุประสงค์เพื่อออกแบบและพัฒนาโปรแกรมสร้างภาพเคลื่อนไหวในเกม 3 มิติ โดยใช้เทคโนโลยีกริด

การพัฒนาระบบนี้นำเสนอเกี่ยวกับการนำเอาเทคโนโลยี Grid Computing ที่เป็นเทคโนโลยีในการประมวลผลแบบขนานเข้ามาประยุกต์ใช้ในการพัฒนาสร้างภาพเคลื่อนไหวในเกม 3 มิติ เป็นการวิจัยเชิงทดลองเพื่อเป็นการพิสูจน์หาผลลัพธ์ในการประมวลผลภาพ 3 มิติแบบขนาน โดยได้มีการเตรียมสภาวะแวดล้อมของการทดลองให้เหมือนกัน ทั้งประสิทธิภาพของเครื่องคอมพิวเตอร์ที่นำมาใช้ในการทดลอง ระบบปฏิบัติการ ซอฟต์แวร์เกม และซอฟต์แวร์ประมวลผลภาพ 3 มิติ ที่ใช้ในเครื่องคอมพิวเตอร์ จะแตกต่างกันที่จำนวนเครื่องในการทดลองซึ่งเป็นตัวแปรสำคัญในการทดลองในครั้งนี้ ซึ่งผู้วิจัยได้เลือกจำนวนเครื่องคอมพิวเตอร์ที่มาใช้ในการทดลองดังนี้ คือ 1 , 4 , 6 , 24 และ 108 เครื่อง โดยที่แต่ละเครื่องจะมีชิ้นงานที่ต้องเข้าไปประมวลผลเหมือนกัน แต่จะทำการประมวลผลกันคนละส่วน แล้วจึงนำข้อมูลที่ได้ส่งกลับไปยังเครื่องที่รอรับข้อมูลอยู่ซึ่งก็คือเครื่องที่ใช้ในการเล่นเกม 3 มิติ จากนั้นเครื่องที่ใช้เล่นเกม 3 มิติจะทำการรวมข้อมูลในส่วนต่าง ๆ แล้วทำการแสดงผลออกมาทางจอภาพ จากการทดลองทั้งหมด 10 ครั้ง พบว่าจำนวนที่เหมาะสมที่สุดในการแบ่งเพื่อส่งไปประมวลผลนั้นอยู่ที่ขนาด 128 pixel หรือแบ่งเป็น 24 ชิ้นงาน มีเวลาเฉลี่ยรวมดีที่สุด โดยใช้เวลาเฉลี่ยที่ 8.60 วินาที/ภาพ

ภาควิชาคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร ปีการศึกษา 2551

ลายมือชื่อนักศึกษา.....

ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์ .....

47307306 : MAJOR : COMPUTER SCIENCE
KEY WORD : GRID COMPUTING / THREE DIMENSION RENDERING
THAMMARONG KUMMANEE : APPLIED GRID TECHNOLOGY FOR ANIMATION RENDERING IN THREE DIMENSION GAME. THESIS ADVISOR : SUNEE PONGPINIGPINYO,Ph.D. 63 pp.

Thesis. "Applied Grid Technology for Animation Rendering in Three Dimension Game" is intended to design and develop programs to create animations using 3D game grid.

This system proposes a take on bringing technology to the Grid Computing technology in parallel processing applications to create animations in the development of 3D games to prove the results for image processing. three dimensional parallel with the preparation of the testing environment to the same performance of the computer are used in the experiments. The Operating system software and game 3-dimensional image processing software used in computers. Will vary in the number of trials is a key variable in this trial. The researcher has selected a number of computers used in the experiments is as follows: 1, 4, 6, 24 and 108 machines, each machine will have to work to process the same but will be processed กัน each section Then the data is sent back to the machine is waiting to receive information that is used in a game from the 3-dimensional computer game that uses Three dimensional information will be included in the display are then made. by monitor From all 10 trials found that the number of times in the most appropriate division for processing is sent to the 128 pixel size or divided into 24 pieces of work have included the best average time. Using the average time 8.60 sec / image.

Department of Computing Graduate School, Silpakorn University Academic Year 2008
Student's signature .....
Thesis Advisor's signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจสำเร็จลุล่วงได้หากไม่ได้รับความกรุณา และ อนุเคราะห์อย่างดียิ่งจากอาจารย์ดร.สุนีย์ พงษ์พินิจภิญโญ ที่กรุณาสละเวลาอันมีค่ารับเป็นอาจารย์ที่ปรึกษา และ ผศ. ดร.ปานใจ ธารทัศนวงศ์ ให้ข้อมูล ดำเนินการช่วยเหลือต่าง ๆ พร้อมทั้งให้คำแนะนำและตรวจแก้วิทยานิพนธ์จนวิทยานิพนธ์เสร็จสิ้นเรียบร้อย ผู้เขียนจึงขอกราบขอบพระคุณท่านอาจารย์ไว้ ณ ที่นี้ด้วย

ผู้เขียนขอกราบขอบพระคุณ ผศ. ดร.ปานใจ ธารทัศนวงศ์ และ ผศ. ดร.ภุชงค์ อุทโยภาศ ที่ได้สละเวลาอันมีค่ายิ่งรับเป็นกรรมการในการสอบวิทยานิพนธ์และให้คำแนะนำต่างๆ จนทำให้การ ศึกษานี้สำเร็จลุล่วงไปด้วยดี

สุดท้ายนี้ ผู้เขียนขอน้อมรำลึกถึงพระคุณบิดา มารดา ที่ได้เป็นกำลังใจที่สำคัญจนเป็นผลให้การศึกษาในระดับปริญญาวิทยาศาสตร์มหาบัณฑิตของผู้เขียนสำเร็จลุล่วงไปได้ในที่สุด

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญภาพ ..... ฌ

บทที่

1 บทนำ 1

ความเป็นมาและความสำคัญของปัญหา 1

ปัญหาในการวิจัย 3

วัตถุประสงค์ของการวิจัย 3

ขั้นตอนการวิจัย 4

ประโยชน์ที่คาดว่าจะได้รับ 4

2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5

อะไรคือกริดเทคโนโลยี 5

สถาปัตกรรมพื้นฐานของกริดเทคโนโลยี 6

สถาปัตยกรรมระบบกริดยุคที่หนึ่ง 6

สถาปัตยกรรมของระบบกริดยุคที่สอง 7

เทคโนโลยีที่สำคัญบนกริด 9

เร็พพลิก้าโลเคชั่นเซอร์วิส 9

อ็อกซ่าดีเอไอ 13

กริดเอฟทีพี 19

ค็อกคิท 21

แนวคิดทฤษฎีการสร้างภาพ 3 มิติ 22

แนวคิดทฤษฎีการคำนวณเส้นทางการเคลื่อนที่ของภาพเคลื่อนไหวโดยใช้

การประมวลผลแบบกระจาย 22

งานวิจัยที่เกี่ยวข้อง 22

3 วิธีการดำเนินการวิจัย 24

ศึกษาข้อมูลเกี่ยวกับการพัฒนาเกมแบบ 3 มิติแบบต่าง ๆ 24

บทที่ หน้า
ศึกษาข้อมูลเกี่ยวกับระบบกริด 24
การเตรียมตัวอย่างสำหรับการทดลอง 26
เครื่องมือและอุปกรณ์ 26
ซอฟต์แวร์ที่ใช้ในการวิจัย 26
ขั้นตอนการทำงานของโปรแกรมประมวลผล 27
4 ผลการดำเนินการวิจัย 31
การประมวลผลภาพ 31
การส่งผ่านข้อมูล 33
การรวมข้อมูลและแสดงผล 34
5 สรุป อภิปรายผลและข้อเสนอแนะ 39
การบรรลุวัตถุประสงค์การวิจัย 39
ปัญหาและอุปสรรค 40
ข้อเสนอแนะ 40
ข้อเสนอแนะสำหรับการวิจัย 40
ข้อเสนอแนะในการวิจัยครั้งต่อไป 40

บรรณานุกรม 41

ภาคผนวก 43
ภาคผนวก ก การติดตั้งและเรียกใช้งาน Blender v2.48 44
ภาคผนวก ข โครงสร้างภายในของ Blender v2.48
และตำแหน่งที่เก็บ Source Code 47
ภาคผนวก ค Blender v2.48 API with Python Script 50
ภาคผนวก ง การติดตั้ง vsftpd สำหรับ FTP 56
ภาคผนวก จ ส่วนของการรวมภาพและแสดงผลภาพ 62

ประวัติผู้วิจัย 63

สารบัญภาพ

ภาพที่		หน้า
1	โครงสร้างทางซอฟต์แวร์ของระบบกริด	6
2	แนวคิดพื้นฐานของสถาปัตยกรรม SOA	8
3	แนวความคิดของการใช้ เร็พพลิก้าแคทตาล็อก	11
4	การคิวรี่ข้อมูลแบบไม่ใช่อ็อกซ่าดีเอไอ	14
5	การคิวรี่ข้อมูลผ่านอ็อกซ่าดีเอไอ	15
6	การทำงานทั่วไปในแบบ SOA	15
7	การทำงานของอ็อกซ่าดีเอไอ	16
8	แสดงสถาปัตยกรรมของอ็อกซ่าดีเอไอ	16
9	แสดงการทำงานของ อ็อกซ่าดีเอไอ	17
10	โครงสร้างการทำงานของโครงการไบโอกริดข้อมูลภาพจากมหาวิทยาลัยโอซาก้า ประเทศญี่ปุ่น	18
11	แสดงสถาปัตยกรรมของกริดเอฟทีพี	21
12	ระบบกริดที่จัดเตรียมไว้โดยผ่านโปรโตคอล TCP/IP ไปยังแต่ละ Node	25
13	ขั้นตอนการสื่อสารในระบบกริดระหว่าง Cluster1 และ Cluster2	26
14	สรุปขั้นตอนการทำงานของโปรแกรมประมวลผลภาพ 3 มิติ จาก Game Player และ Render Node	27
15	การแบ่งชิ้นส่วนสำหรับการประมวลผลในแต่ละ Render Node	28
16	แสดงการส่งข้อมูลจากเครื่อง Render Node ไปยังเครื่อง Game Player โดยใช้วิธี FTP	29
17	แสดงการประกอบชิ้นส่วนของรูปภาพให้ตรงตำแหน่งเพื่อใช้แสดงผล	30
18	โปรแกรม Blender v2.48	46
19	แผนภูมิแสดงโครงสร้างของ Blender v2.48	48
20	ตำแหน่งที่เก็บ Source Code ที่จำเป็นต่อการเรียกใช้งาน	49

บทที่ 1
บทนำ

ความเป็นมาและความสำคัญของปัญหา

อินเตอร์เน็ต (Internet) เป็นเทคโนโลยีที่เริ่มต้นจากการที่นักวิทยาศาสตร์ที่กระจายอยู่ตามมหาวิทยาลัยและสถาบันวิจัยทั่วโลกมีความจำเป็นต้องการแลกเปลี่ยนข้อมูล ข้อคิดเห็นและติดตามผลงานใหม่ๆให้ทันสมัย ทำให้ World Wide Web ถูกประดิษฐ์ขึ้น Sir Timothy J. Berners-Lee (1997) มาเพื่อสนองความต้องการดังกล่าว และต่อมา สิ่งนี้ก็ถูกพัฒนาต่อเนื่องไปเป็นอินเตอร์เน็ต (Internet) ที่ถูกนำไปประยุกต์ใช้กับงานหลายๆด้าน ทั้งทางการศึกษา การพาณิชย์บันเทิง อุตสาหกรรมและงานราชการ จนกลายมาเป็นสิ่งที่อำนวยความสะดวกแก่ชีวิตประจำวันอย่างที่ไม่มีใครคาดคิดมาก่อน อันเป็นตัวอย่างของ Technology Transfer จากงานวิจัยด้านวิทยาศาสตร์บริสุทธิ์ไปสู่การพัฒนาคุณภาพชีวิตของมนุษย์ที่เด่นชัดที่สุดชิ้นหนึ่ง (ฮิกส์แมน 2548)

สืบเนื่องมาจากความซับซ้อนของปัญหาที่นักวิทยาศาสตร์กำลังทำการศึกษาในปัจจุบัน ไม่ว่าจะเป็น งานวิจัยด้านชีวเคมี ที่ต้องการศึกษาโครงสร้างของโมเลกุลขนาดใหญ่ที่ซับซ้อนมหาศาลจำนวนมาก เพื่อหาลักษณะที่เหมาะสมของสารที่จะนำมาทำตัวยารักษาโรค หรือไม่ว่าจะเป็นกรณีของงานวิจัยที่ โครงการ LHC (Large Hadron Collider) ที่ศูนย์วิจัยฟิสิกส์อนุภาคแห่งชาติยุโรป CERN ที่จำเป็นต้องมีการถ่ายโอนข้อมูลที่ได้จากผลการทดลอง ที่คาดว่าจะมีขนาดกว่า 10^{15} ไบต์ต่อปี ไปให้นักฟิสิกส์ทั่วโลกทำการวิเคราะห์ หรือไม่ว่าจะเป็น งานวิจัยอื่นๆ เช่น งานวิจัยทางด้านถอดรหัสพันธุกรรมที่ทุกคนคุ้นหูกัน หรืองานด้านตรวจจับ คลื่นแรงโน้มถ่วง (gravitational wave) เพื่อพิสูจน์ทฤษฎี General Relativity (Einstein 1915) หรือการวิเคราะห์หาสัญญาณของสิ่งมีชีวิตนอกโลก (E.T.) ล้วนแล้วแต่ต้องการสมรรถนะคอมพิวเตอร์และระบบเน็ตเวิร์คที่ว่าล้ำหน้าสุดๆ ที่แม้กระทั่งความล้ำสมัยของเทคโนโลยี IT ในปัจจุบันที่ถึงแม้กำลังเพิ่มประสิทธิภาพแบบเลขยกกำลัง ก็ยังดูล้าหลังและไม่เพียงพอต่องานวิจัยต่างๆที่กล่าวมานี้ และนี่เป็นสาเหตุให้เทคโนโลยี Grid จำเป็นต้องมีการพัฒนาขึ้นเพื่อมารองรับ

คำว่า Grid หมายถึง “เครือข่ายที่เชื่อมโยงกันและกระจายทรัพยากรให้กัน” ซึ่งในที่นี้ Grid Computing คือเครือข่ายของทรัพยากรคอมพิวเตอร์ที่เชื่อมโยงกันหมดและจะกระจายทรัพยากรด้านคอมพิวเตอร์ให้กัน ไม่ว่าจะเป็นสมรรถนะในด้านการประมวลผล ความจุ หรือ

สมรรถนะในการถ่ายโอนข้อมูล โดยการจ่ายทรัพยากรนี้จะมีลักษณะเช่นเดียวกับ Grid ของพลังงานไฟฟ้าที่พวกเราใช้ตามบ้าน สิ่งที่เราต้องทำก็เพียงเสียบปลั๊กและเปิดสวิทซ์ เราก็จะได้พลังงานไฟฟ้ามาบริโภคมากตามที่เราต้องการ และเทคโนโลยี Grid Computing ก็ถูกออกแบบให้ผู้ที่อยู่ในเครือข่าย เพียงเสียบปลั๊กและใส่รหัสก็จะเข้าถึงทรัพยากรคอมพิวเตอร์ได้อย่างสะดวกและเรียกใช้ทรัพยากรทาง Computer ได้มากเท่าที่เราต้องการเช่นเดียวกัน

ในกรณีของ Electricity Grid หรือเครือข่ายของพลังงานไฟฟ้าที่จ่ายไฟตามบ้านนั้น แหล่งจ่ายไฟจะส่งมาจาก เขื่อน หรือโรงงานผลิตไฟฟ้าถ่านหิน หรือ นั่นก็คือ มีแหล่งจ่ายทรัพยากรอยู่แหล่งเดียว แต่ในกรณีของ Grid Computing จะใช้ทรัพยากรที่แบ่งปันมาจากเครือข่ายที่เชื่อมโยงกันของคอมพิวเตอร์ทั่วโลกที่เป็นสมาชิกของเครือข่าย ไม่ว่าจะเป็นคอมพิวเตอร์นั้นจะเป็นคอมพิวเตอร์ส่วนบุคคล หรือ Supercomputer ตามศูนย์วิจัย หรือ Cluster ตามมหาวิทยาลัย โดยไม่จำเป็นที่คอมพิวเตอร์เหล่านี้ต้องมีแพลตฟอร์มเหมือนกัน(ข้อแตกต่างระหว่าง Cluster กับ Grid คือ Cluster เป็นการเชื่อมต่อเพื่อเพิ่มสมรรถนะของการประมวลผลด้วยเครื่องคอมพิวเตอร์ที่มีแพลตฟอร์มเดียวกันอยู่ในพื้นที่จำกัด ส่วน Grid นั้นจะเชื่อมต่อได้ทุกแพลตฟอร์ม ไม่ว่าแต่ละแพลตฟอร์มจะห่างไกลกันเท่าไร)

Grid นี้กำลังอยู่ในขั้นตอนการพัฒนาอยู่ ยังไม่เสร็จสมบูรณ์ ส่วนประกอบหลักมีอยู่ 4 ส่วน โดยแต่ละส่วนแบ่งเป็นชั้นๆ (layer) ได้แก่

1. Network layer
2. Resources layer
3. Middleware layer
4. Application & Serviceware layer

โดยตัวชั้นสุดท้ายนี้เป็นส่วนที่ผู้ใช้จะมองเห็นและสัมผัส จริงๆแล้วเทคโนโลยี Grid นี้ไม่ได้กำลังจะเกิด แต่มันมีมาเกือบสิบปีแล้ว จากโครงการ SETI@home(Search for Extraterrestrial Intelligence) ที่เป็นการยืมความสามารถในการประมวลผลของคอมพิวเตอร์ส่วนบุคคลที่อยู่ในช่วงไม่ได้ใช้งานมาวิเคราะห์ข้อมูลที่ได้จากยานอวกาศเพื่อหาสัญญาณของสิ่งมีชีวิตนอกโลก ผู้ที่สมัครใจให้คอมพิวเตอร์ตัวเองช่วยประมวลผลจะได้ของแถมเป็น screen saver โดยสามารถสมัครไปได้ที่ http://boinc.berkeley.edu/ ซึ่งปัจจุบัน ไม่ว่าจะเป็นงานทางด้านการวิจัยรักษามะเร็ง การพิสูจน์หา

คลื่นแรงโน้มถ่วงที่ไอน์สไตน์ได้ทำนายไว้ หรือไม่ว่าจะเป็นงานด้านการทำนายสภาพอากาศล่วงหน้า ได้เข้าร่วมเป็นส่วนหนึ่งของโครงการเดียวกับ SETI แล้ว โดยสามารถเข้าไปดูและดาวน์โหลดได้ที่เว็บของมหาวิทยาลัยเบร์กเลย์ http://boinc.berkeley.edu/

ทุกวันนี้ เครื่องคอมพิวเตอร์ทั้งแบบตั้งโต๊ะและแบบพกพา ด้วยสมรรถนะที่เท่ากับ supercomputer ที่เป็นสุดยอดคอมพิวเตอร์ในสมัยเมื่อ 10 ปีที่แล้ว อีกทั้งมีเครือข่ายอินเตอร์เน็ตที่สามารถส่งข้อมูลระหว่างกันได้ด้วยอัตราเร็วที่ 155 Mbps (megabits per second) ซึ่งเป็นหลายร้อยเท่าของความเร็วในการถ่ายโอนข้อมูลของเครือข่ายที่เชื่อมต่อ supercomputer ในสหรัฐฯ ทั้งหมดเมื่อสมัยปี ค.ศ. 1985 ความจุของฮาร์ดดิสค์ทุกวันนี้ เครื่องใหม่ๆ คงได้ความจุราวๆ 100 GB ซึ่งเท่ากับความจุของเครื่อง supercomputer ทั้งหมดที่มีอยู่รวมกันในสมัยเมื่อ 10 ปีที่แล้วอีกเช่นกัน อีกไม่นาน เทคโนโลยี Grid Computing จะเข้าถึงทรัพยากรคอมพิวเตอร์ที่มากที่สุดอย่างที่ไม่เคยคิดมาก่อน และวิทยาศาสตร์ในอนาคตก็จะเป็นยุคของ E-Science อันเนื่องมาจากการใช้สมรรถนะในการประมวลผลของคอมพิวเตอร์จากเทคโนโลยี Grid Computing

นอกจากประโยชน์ของ Grid ต่องานทางด้านวิทยาศาสตร์แล้ว ทางภาคธุรกิจคาดว่าเทคโนโลยี Grid นี้สามารถนำมาช่วยขจัดปัญหาค่าใช้จ่ายด้านทรัพยากรคอมพิวเตอร์ หรือช่วยในการเก็บและวิเคราะห์ข้อมูลลูกค้าที่มากมายมหาศาล ส่วนทางด้านอื่นๆ Grid คาดว่าอาจถูกนำมาใช้สำหรับทางด้านบันเทิงอย่าง real-time calculation หรือ อินเตอร์เน็ต (Internet) on-line game

ปัญหาในการวิจัย

เกม 3 มิติจะใช้การประมวลผลทางภาพ (Graphic) สูงมาก ซึ่งในบางครั้งจะมีการประมวลผลที่ซับซ้อนเครื่องคอมพิวเตอร์เพียงเครื่องเดียวไม่สามารถประมวลผลได้ทัน จึงมีแนวความคิดนำเอาเทคโนโลยีของกริดเข้ามาช่วยในการประมวลผล

วัตถุประสงค์ของการวิจัย

ออกแบบและพัฒนาโปรแกรมเพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ โดยประยุกต์ใช้เทคโนโลยีกริด

ขั้นตอนการวิจัย

1. รวบรวมข้อมูลที่จำเป็นในการพัฒนาระบบ Grid Computing และการเขียนโปรแกรมเกม 3 มิติจากแหล่งข้อมูลเว็บไซท์และเอกสารต่าง ๆ
2. ศึกษาลักษณะการทำงานของระบบ Grid Computing
3. ออกแบบและพัฒนาโปรแกรมเกม 3 มิติตัวอย่างด้วยโปรแกรม Blender เพื่อใช้ในการทดสอบ
4. ออกแบบและพัฒนาระบบการแบ่งชิ้นงาน การส่งผ่านข้อมูลเพื่อประมวลผลข้อมูลแบบ 3 มิติ และการรวบรวมข้อมูลมาแสดงผล
5. ทดสอบและปรับปรุงการทำงานของระบบการแบ่งชิ้นงาน การส่งผ่านข้อมูลเพื่อประมวลผลข้อมูลแบบ 3 มิติ และการรวบรวมข้อมูลมาแสดงผล
6. เปรียบเทียบการประมวลผลภาพ 3 มิติ ที่จำนวนเครื่องแตกต่างกันเพื่อหาประสิทธิภาพว่าแบบใดได้ผลดีที่สุด
7. สรุปผลการดำเนินการ
8. จัดทำรายงานวิทยานิพนธ์

ประโยชน์ที่คาดว่าจะได้รับ

ได้ต้นแบบในการพัฒนาระบบประมวลผลภาพเคลื่อนไหวในเกม 3 มิติ โดยการประยุกต์ใช้เทคโนโลยีกริด

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการพัฒนาระบบการประมวลผลข้อมูลแบบ 3 มิติโดยใช้เทคโนโลยีกริดประกอบไปด้วย แนวความคิดของการประมวลผลแบบกริด ทฤษฎีที่เกี่ยวข้องกับเทคโนโลยีกริด ประเภทของกริด การทำงานของกริด ทรัพยากรที่ใช้ในกริด สถาปัตยกรรมของกริด โครงสร้างของระบบการประมวลผลแบบกริด ลักษณะของแอพพลิเคชั่นที่เหมาะแก่การใช้งานกริด ตัวอย่างของงานที่ใช้การประมวลผลแบบกริด ทฤษฎีการสร้างภาพ 3 มิติ รูปแบบการให้บริการเกมออนไลน์ และงานวิจัยที่เกี่ยวข้องกับการประมวลผลแบบ 3 มิติโดยใช้กริดในปัจจุบัน ซึ่งมีรายละเอียดดังต่อไปนี้

1. อะไรคือกริดเทคโนโลยี

เทคโนโลยีกริด คือ (Foster et al. 2001 , 2002 a , 2002 b , 2002 c) เทคโนโลยีในการสร้างระบบกระจายขนาดใหญ่มากข้ามองค์กรหลากหลายที่อยู่ห่างกันเข้าหากันเพื่อให้มีการใช้งานทรัพยากรร่วมกันอย่างปลอดภัย โดยอาศัยเทคโนโลยีของเครือข่ายความเร็วสูง และเทคโนโลยีของมิดเดิลแวร์ เทคโนโลยีนี้ ทำให้เราสามารถรวมพลังการคำนวน, เนื้อที่เก็บข้อมูล ต่างๆเข้าหากันได้ และควมคุมการแจกจ่ายทรัพยากรเหล่านี้ให้กับโปรแกรมประยุกต์ต่างๆ ได้ตามความต้องการ ทำให้บางบริษัทเรียกเทคโนโลยีประเภทนี้ว่า การประมวลผลตามต้องการ (On-demand computing) อีกด้วย

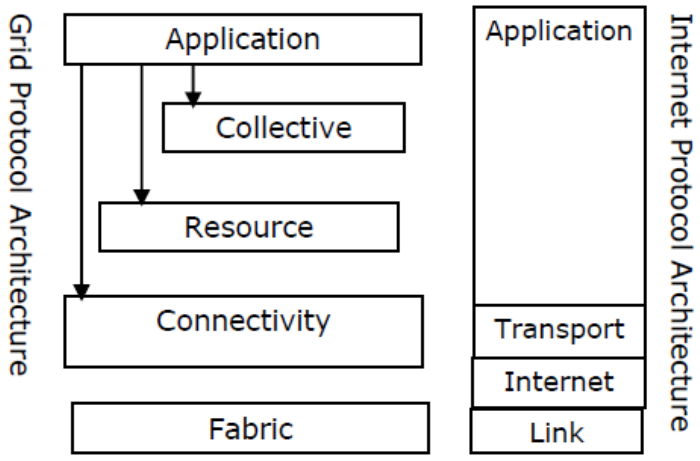
แนวความคิดของการสร้างระบบกริด มีตั้งแต่ในราวกลางทศวรรษที่ 1990 โดยเรียกว่า เมตะคอมพิวติ้ง (Meta-Computing) ก่อน แล้วจึงเปลี่ยนเป็นกริดในภายหลัง โครงการที่สำคัญหลายโครงการและแนวคิดเริ่มต้นถูกสรุปรวมไว้ในหนังสือ The Grid: Blue Print for a New Computing Infrastructure อย่างไรก็ตามแนวคิดที่สำคัญและใช้เป็นแนวทางของระบบกริดในปัจจุบัน คือ แนวคิดของแนวคิดขององค์กรเสมือน (Virtual Organization) ซึ่งเสนอโดย (Foster et al., 2001) องค์กรเสมือนเป็นการรวมกันของนักวิจัยโดยมีการใช้ทรัพยากรรวมกันอย่างใกล้ชิด โดยใช้ระบบกริด รวมทรัพยากรที่กระจายตัวตามสภาพภูมิศาสตร์ (geographically distributed) และถูกบริหารโดยองค์กรที่แตกต่างกัน (multiple administrative domain) เข้าหากันผ่านเครือข่ายสื่อสารทำให้เกิด

การทำงานที่สะดวกและรวดเร็วมากขึ้น ตัวอย่างเช่น กลุ่มนักวิจัยด้าน ไบโออินฟอร์เมติกส์ (Bio Informatics) ในมหาวิทยาลัยต่างๆทั่วประเทศสามารถสร้างกลุ่มวิจัยที่เน้นด้านนี้และใช้งานคอมพิวเตอร์ ฐานข้อมูล ต่างๆร่วมกันได้ ซึ่งเทคโนโลยีระบบกริดจะเป็น รากฐานสำคัญในการสร้างองค์กรเสมือนดังกล่าว ได้มีการสำรวจและเปรียบเทียบโครงการและแนวคิดทางเทคโนโลยีด้านนี้ไว้อย่าง กว้างขวาง

2. สถาปัตกรรมพื้นฐานของกริดเทคโนโลยี

ระบบกริดได้ผ่านการวิวัฒนาการมาสองยุคด้วยกัน (Foster et al. 2001 , 2002 a , 2002 b , 2002 c) ยุคแรกของกริดจะเน้นการใช้งานด้านวิทยาศาสตร์โดย มุ่งสร้างโครงสร้างของระบบที่รวมพลังการประมวลผลเข้าหากัน เรียกว่า คอมพิวเตชั่นนอล กริด (Computational Grid) และ ระบบที่ช่วยในการขนส่งข้อมูลจำนวนมากเรียกว่า ดาต้ากริด (Data Grid) ในปัจจุบันกริดเริ่มวิวัฒนาการเข้าสู่ยุคที่สอง คือ เซอร์วิสโอเรียนทด์ กริด (Service oriented grid) ซึ่งมุ่งเน้นในการสร้างบริการ (Services) และการพัฒนาโปรแกรมประยุกต์ในลักษณะสถาปัตยกรรมเน้นบริการ (Service Oriented Architecture หรือ SOA) ในบทความนี้จึงขอแนะนำโครงสร้างของทั้งสองยุคเพื่อให้เข้าใจได้มากขึ้น

2.1 สถาปัตยกรรมระบบกริดยุคที่หนึ่ง



ภาพที่ 1 โครงสร้างทางซอฟต์แวร์ของระบบกริด

โครงสร้างของระบบกริดยุคที่ 1 จะเป็นไปดัง ซึ่งมีโครงสร้างดังภาพที่ 1

- กริดแฟบริคเลเยอร์ (Grid Fabric Layer) ในระดับนี้จะประกอบไปด้วย ทรัพยากรต่างๆ เช่น คอมพิวเตอร์สมรรถนะสูงระบบเก็บข้อมูลขนาดใหญ่ เครือข่าย ฐานข้อมูลซึ่งทรัพยากรเหล่านี้จะกระจายตัวอยู่บนเครือข่าย

- คอนเน็ททิวีตี้ เลเยอร์ (Connectivity Layer) กำหนดมาตรฐานของการสื่อสารแบบปลอดภัยและการตรวจสอบผู้ใช้และสิทธิต่างๆในเลเยอร์นี้จะช่วยให้มีการติดต่อระหว่างซอฟต์แวร์ใน กริดแฟบริคเลเยอร์ อย่างปลอดภัยและเป็นภาพแบบเดียวกัน เทคโนโลยี อย่างเช่น Secure shell (SSH), Secure socket เลเยอร์ (SSL), Public key encryption จะเป็นแกนหลัก

- รีซอสเลเยอร์ (Resource Layer) ซอฟต์แวร์ในเลเยอร์ นี้ จะกำหนดชุดของ โพรโตคอลที่ทำให้มีการจองการใช้ การเจรจา การควบคุมและติดตาม รวมทั้งการคิดราคาการใช้งาน บนทรัพยากรต่างๆ ในเลเยอร์นี้จะต้องสนับสนุนการสอบถามและการควบคุมสถานะของทรัพยากรแต่ละอย่าง ซึ่งระบบความปลอดภัยทั้งหมดจะพึ่งพา คอนเน็ททิวีตี้เลเยอร์เป็นหลัก

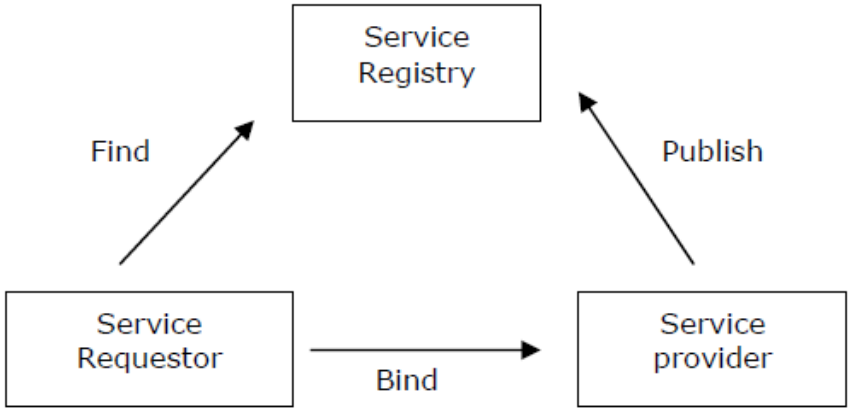
- ค็อลเลคทีฟเลเยอร์(Collective Layer) ในเลเยอร์นี้จะเป็นการประสานงานของการใช้งานทรัพยากรจำนวนมากให้เป็นไปตามความต้องการของแอ็พพลิเคชั่นบริการเหล่านี้บางส่วนได้แก่ Directory Services, Scheduling Services, Data Replication Services, CollaboratoryServices

ในการพัฒนาซอฟต์แวร์นั้นโครงสร้างทั้งหมดจะถูกรวมไว้ในซอฟต์แวร์ที่เรียกว่า กริดมิดเดิลแวร์ (Grid Middleware) ในปัจจุบันมิดเดิลแวร์ที่ใช้กันกว้างขวางและเป็นมาตรฐานที่สุดคือ โกลบัส (Globus) ซึ่ง เป็นผลงานของ Ian Foster จาก Argonne National Laboratory และ Karl Kasselmann จาก ISI, University of Southern California ในปัจจุบันเวอร์ชั่นของโกลบัสที่ใช้งานกันมากคือ รุ่น 3.2 ผู้ใช้งานระบบ กริดจะเรียกใช้ เอพีไอ (API) ของมิดเดิลแวร์หรือเรียกใช้บริการบางอย่างผ่านบรรทัดคำสั่ง (command line utility) ที่ระบบให้มา

2.2 สถาปัตยกรรมของระบบกริดยุคที่สอง

ในปัจจุบันระบบกริดกำลังก้าวเข้าสู่ความเปลี่ยนแปลงครั้งใหญ่ เนื่องจากการมาถึงของเทคโนโลยี เว็บเซอร์วิส (Web Services) ทำให้มองเห็นว่าระบบกริดในอนาคตควรจะสร้างจากรากฐานของระบบเว็บเซอร์วิสดังนั้นทีมงานโกลบัสจึงได้เสนอสถาปัตยกรรมโอเพ็นกริดเซอร์วิส (Open Grid Services Architecture หรือ OGSA) ซึ่งนับเป็นแนวคิดหลักและกำลังถูกสร้างเป็นมาตรฐานโดย Global Grid Forum อยู่ ก่อนจะอธิบายแนวความคิดของ อ็อกซ่า (OGSA) ขออธิบาย

แนวคิดการพัฒนาโปรแกรมประยุกต์แบบ SOA ก่อนโดยแนวคิดของการพัฒนาโปรแกรมประยุกต์ในแนวทาง SOA นี้สามารถแสดงได้ดังภาพที่ 2



ภาพที่ 2 แนวคิดพื้นฐานของสถาปัตยกรรม SOA

ในสถาปัตยกรรมแบบ SOA นั้นจะถือว่าระบบประกอบไปด้วยบริการ (service) จำนวนมากกระจายตัวกันอยู่บนเครือข่าย โดยมีผู้ให้บริการ (Service Provider) ทำหน้าที่ให้บริการนั้น จากนั้นโปรแกรม ประยุกต์จะเป็นผู้เรียกใช้บริการ (Service requestor) มาทำงานตามคำสั่งของผู้ใช้ การหาว่ามีผู้ให้บริการใด บ้างและอยู่ที่ไหนจะทำผ่าน ผู้ลงทะเบียน (Service registry) ซึ่งผู้ให้บริการต้องมาลงทะเบียนเอาไว้ในระบบกริดนั้น บรรดากลไกเหล่านี้จะถูกสร้างไว้เป็นพื้นฐานและมีขีดความสามารถบางอย่างเพิ่มเติมขึ้นมา ทำให้สามารถพัฒนาโปรแกรมประยุกต์ขนาดใหญ่ที่กระจายตัวผ่านเครือข่ายได้ง่าย ตัวอย่างกลไกเหล่านี้ได้แก่

- แฟคทอรี (Factory) ส่วนที่ดูแลผู้ให้บริการทำให้สร้างและเรียกใช้บริการต่างได้ง่าย
- รีจิสเตอร์ (Registry) บริการลงทะเบียนทรัพยากรและบริการ
- ดิสโคเวอร์รี่ (Discovery) ส่วนค้นหาบริการและค้นหาข้อมูลบนระบบได้ง่าย
- ไลฟ์ไซเคิล (Life Cycle) บริหารเวลาที่ให้บริการทำให้จำกัดการใช้ทรัพยากรได้ง่าย
- เซอร์วิสดาต้าอีริเม็นท์ (Service Data Element) ทำให้เก็บสถานะการทำงานของบริการ
- โนทิฟิเคชั่น (Notification) ระบบแจ้งเตือนเมื่อมีเหตุการณ์ต่างๆเกิดขึ้น
- ริไลอเบิลอินโวเคชั่น (Reliable invocation) ระบบส่งสัญญาณเตือนในกรณีมีความผิดพลาดหรือเสียหายเกิดขึ้นในระบบ

นอกจากนั้นกริดมิดเดิลแวร์ของGlobus Version 3.2 ยังให้บริการพื้นฐานอยู่อีกสามประการ คือ ระบบความปลอดภัย (Grid Security Infrastructure), ระบบบริหารระยะไกลเอ็มเอ็มเจ

เอฟเอส (Remote Execution MMJFS) และ ระบบส่งข้อมูลขนาดใหญ่ (Reliable Transfer Service) อย่างไรก็ตาม สิ่งที่สำคัญที่สุด คือ กริดได้กำหนดมาตรฐานการเขียน กริดเซอร์วิสซึ่งเป็นเว็บเซอร์วิส รูปแบบหนึ่งที่ทำงานบนกลไกความปลอดภัยของกริด ทำให้สามารถพัฒนาระบบความปลอดภัยของเว็บเซอร์วิส แอ็พพลิเคชั่น ได้โดยอาศัยกลไกดังกล่าว ดังนั้นผู้ใช้จึงสามารถพัฒนาโปรแกรมประยุกต์และบริการที่ขยายความสามารถของระบบกริดออกไปได้อย่างไม่มีขีดจำกัด

3. เทคโนโลยีที่สำคัญบนกริด

เนื่องจากกริดเทคโนโลยีประกอบด้วยหลายส่วนดังสถาปัตยกรรมที่ได้กล่าวมาแล้วในข้างต้นแต่ในส่วนเทคโนโลยีที่สำคัญที่ทำให้เกิดแนวคิดในการวิจัยนี้จะประกอบด้วยเทคโนโลยีที่สำคัญในกริดดังต่อไปนี้

3.1 เร็พพลิก้าโลเคชั่นเซอร์วิส (Replica Location Service หรือ RLS)

เร็พพลิก้าโลเคชั่น เซอร์วิส เป็นหัวข้อหลักที่เราจะกล่าวถึงส่วนนี้ คือส่วนที่ให้บริการที่เรียกว่า เร็พพลิก้าแคทตาล็อก (Replica Catalog หรือ RC) สำหรับชื่อ อาร์แอลเอส นี้เริ่มใช้ตั้งแต่ซอฟต์แวร์โกลบัสเวอร์ชั่น 3.0 เป็นต้นมา โดยจากเดิมเราจะเรียกว่า เร็พพลิก้า เมเนจเมนท์ (Replica Management) โดยในเวอร์ชั่น 3.0 นี้ได้เพิ่มคุณสมบัติใหม่ๆ มาอีกหลายประการโดยจะขอกล่าวรวมในหัวข้อถัดไปเหตุผลการพัฒนา อาร์แอลเอส ก็คือเนื่องจากปัจจุบันงานที่ใช้เทคโนโลยีการประมวลผลแบบกริดโดยมากจะเป็นงานทางด้านวิทยาศาสตร์ซึ่งประกอบด้วยประมาณข้อมูลจำนวนมาก ส่งผลให้ปริมาณของผลลัพธ์มีมากไปตามกัน ข้อมูลมากมายมหาศาลเหล่านี้หากจะจัดเก็บไว้ในที่ๆ เดียวคงทำได้ยาก เทคโนโลยีที่จะทำให้เราสามารถเข้าถึงข้อมูลที่ต้องการและจัดการกับข้อมูลจำนวนมากบนกริด จึงเกิดขึ้นและเป็นที่มาของ กริด เร็พพลิก้าแคทตาล็อก(Grid Replica Catalog) อันจะทำให้นักวิจัยที่อยู่กันคนละซีกโลกทำงานร่วมกันอย่างสะดวกขึ้น

เร็พพลิก้าแคทตาล็อกเป็นบริการหนึ่งในส่วนของการบริหารจัดการข้อมูลในระบบกริด (Data Management) เป็นเทคโนโลยีที่มุ่งเน้นในการจัดการข้อมูลขนาดใหญ่ในระบบกริดเพื่ออำนวยความสะดวกในการเข้าถึงข้อมูลที่ กระจายกันอยู่ในระบบในงานหนึ่งงานอาจประกอบด้วยไฟล์ข้อมูลหลายไฟล์ ในการจัดเก็บไฟล์หรือกลุ่มไฟล์ขนาดใหญ่จำเป็นต้องใช้พื้นที่ในการจัดเก็บจำนวนมาก เมื่อเรานำระบบกริดเข้ามาช่วยในการจัดเก็บข้อมูล ข้อมูลจะถูกจัดเก็บกระจายกันอยู่ในระบบกริด ซึ่งไฟล์ข้อมูลในชุดเดียวกันอาจถูกจัดเก็บอยู่กันคนละทวีปก็เป็นได้ ก็จะส่งผลถึงการ

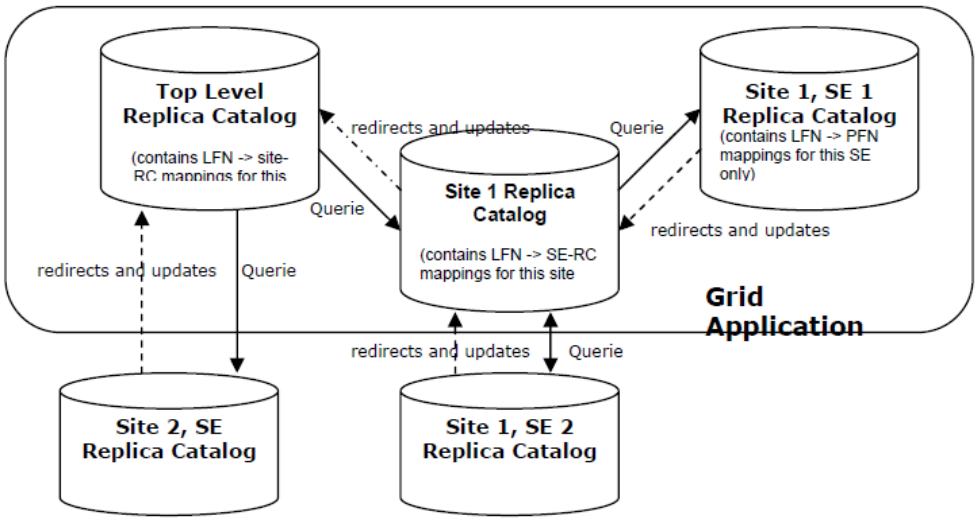
เรียกใช้และเข้าถึงไฟล์ ระบบ เร็พพลิก้าแคทตาล็อกจะเข้ามามีส่วนช่วยในการจัดการในจุดนี้ เร็พพลิก้าแคทตาล็อกส่วนหนึ่งของการจัดการข้อมูลบนกริด

ก่อนจะกล่าวถึงเรื่อง กริด เร็พพลิก้าแคทตาล็อกอย่างเต็มตัวนั้น ขอกล่าวถึงบริการในส่วนของการจัดการข้อมูล (Data Management) กันก่อนว่าประกอบด้วยอะไรบ้างเพราะ เร็พพลิก้าแคทตาล็อกนั้นก็เป็นส่วนหนึ่งภายใต้ส่วนของการจัดการข้อมูลบนกริด ซึ่งจะมีความสัมพันธ์ซึ่งกันและกัน ตลอดจนมีการเรียกใช้งานจากบริการตัวอื่นๆ ในกลุ่มเดียวกันด้วยสำหรับบริการหลักๆ ในส่วนของการจัดการข้อมูลบนกริดจะมีสามส่วน ดังต่อไปนี้

กริดเอฟทีพี (GridFTP) เป็นบริการที่ถูกพัฒนาต่อยอดมาจาก เอฟทีพี (FTP) ปรกติแต่ได้รับการพัฒนาและเพิ่มคุณสมบัติบางประการให้เหมาะกับระบบกริด เป็นบริการที่ทำหน้าที่ในการโอนถ่าย ไฟล์ข้อมูลภายในระบบกริด ใช้ภาพแบบการอ้างอิงถึงอยู่ของไฟล์ในรูปแบบของยูอาร์แอล (URLs)

รีไลอเบิล ไฟล์ ทรานส์เฟอร์ (Reliable File Transfer หรือ RFT) เป็นบริการที่เรียกใช้บริการของกริดเอฟทีพี อีกที่ แต่จะมีการความสามารถในการจัดการสถานะของไฟล์ระหว่างการโอนถ่ายไฟล์ด้วย โดยจะมีการเก็บสถานะการโอนถ่ายไฟล์ลงฐานข้อมูลเป็น ระยะ มีข้อดีก็คือทำให้ทราบว่าโอนย้าย ไฟล์มาถึงขั้นไหนแล้ว ในกรณีเกิดการผิดพลาดในการติดต่อกับเซิร์ฟเวอร์ เมื่อเริ่มดาวน์โหลดใหม่ระบบสามารถดาวน์โหลดต่อเนื่องจากที่เดิมได้ เป็นต้น แต่ก็มีข้อเสียคือต้องเสียเวลาในการปรับปรุงข้อมูลดังกล่าว ทำให้เกิดความล่าช้าในการโอนย้ายไฟล์

ระบบเร็พพลิก้าแคทตาล็อกเป็นส่วนหนึ่งที่ช่วยจัดการกับข้อมูลบนกริดด้วยกริดเป็นระบบที่แบ่งปันทรัยพากร ระหว่างองค์กรเพื่อประโยชน์ในการใช้งานระบบให้คุ้มค่ามากที่สุดและให้ได้ขีดความสามารถในด้านต่างๆ ที่สูงในด้านของพื้นที่ในการจัดเก็บข้อมูลก็เช่นกันบนระบบกริดเราสามารถแบ่งปันพื้นที่ว่างของระบบให้กับระบบกริดได้แต่ด้วยเหตุที่ระบบกริดอาจเป็นระบบขนาดใหญ่ที่ประกอบด้วยเครื่องคอมพิวเตอร์จำนวนมาก ย่อมส่งผลให้การค้นหาไฟล์ที่จัดเก็บกระจายอยู่ใน กริดนั้นเป็นไปได้ยาก การทำดัชนีของสถานที่จัดเก็บไฟล์จึงเป็นส่วนสำคัญที่จะช่วยให้การจัดเก็บและเรียกใช้งานข้อมูลบนระบบกริดได้อย่างสะดวกขึ้นแนวความคิดของการใช้เร็พพลิก้าแคทตาล็อกจะเป็นดังภาพที่ 3



ภาพที่ 3 แนวความคิดของการใช้ เร็พพลิก้าแคทตาล็อก

สิ่งที่ เร็พพลิก้าแคทตาล็อกทำให้เราก็คือการทำดัชนีของไฟล์โดยระบบจะทำการจับคู่ระหว่างชื่อทางตรรกะ (Logical File Names หรือ LFN) ซึ่งอาจเป็นชื่อที่จับคู่กับไฟล์ๆ เดียวหรือกลุ่มของไฟล์ก็เป็นได้ กับชื่อไฟล์ทางกายภาพ (Physical File Names หรือ PFN) ซึ่งก็คือชื่อไฟล์จริงๆ ที่จัดเก็บกระจายกันอยู่ในระบบรวมถึงสถานที่จัดเก็บไฟล์นั้นๆ ด้วย (Storage Element หรือSE)

เวลาที่มีผู้ต้องการใช้งานไฟล์ก็จะมาถามที่เร็พพลิก้า เมเนจเมนท์เพื่อขอที่อยู่ของไฟล์และไปติดต่อโอนถ่ายไฟล์มายังระบบของตน ตรงจุดนี้จะทำให้เกิดชุดสำเนาของไฟล์ข้อมูลขึ้นซึ่งอาจจะเป็นการสำเนาทั้งชุดข้อมูลหรือเพียงบางส่วนของกลุ่มข้อมูล ซึ่งหลังจากโอนถ่ายข้อมูลมาแล้วจะต้องทำการลงทะเบียนสำเนาข้อมูลตรงนี้ไว้กับโลคอล เร็พพลิก้าแคทตาล็อก (Local Replica Catalog) ซึ่งจะมีการอัพเดทข้อมูล แคทตาล็อกระหว่าง เร็พพลิก้า ตามลำดับชั้น ข้อมูลที่จัดเก็บในแคทตาล็อกจะเป็นข้อมูลที่จำเป็นในการเรียกใช้ไฟล์ เช่น ที่อยู่ของไฟล์ ขนาดของไฟล์ เป็นต้น

เร็พพลิก้าแคทตาล็อกจะจัดเก็บข้อมูลรายการที่อยู่ของไฟล์บนฐานข้อมูลโดยติดต่อผ่านโอดีบีซี (ODBC) และไคลเอนท์จะใช้เร็พพลิก้าแคทตาล็อกในการติดต่อกับโลคอล เร็พพลิก้าแคทตาล็อกจะอัพเดทข้อมูลกับดัชนีเร็พพลิก้าทุกช่วงเวลา ตามที่กำหนด

หลังจากที่มีสำเนาข้อมูลที่เก็บกันกระจายกันอยู่ ก็จะเป็นประโยชน์ต่อผู้ที่ต้องการใช้งานไฟล์เหล่านี้ต่อๆ ไป ที่สามารถเลือกดึงข้อมูลจากตำแหน่งที่อยู่ใกล้หรือเส้นทางที่สามารถโอนถ่าย

ข้อมูลได้ดี โดยสามารถคำนวณได้จากขนาดของไฟล์กับการประเมินค่าแบนวิทด์ของเครื่องข่ายจะส่งผลให้ใช้เวลาในการโอนถ่ายไฟล์ที่สั้นลงและเป็นการกระจายภาระงานภายในระบบอีกด้วย

เร็พพลิก้าแคทตาล็อกทำหน้าที่ดูแลเฉพาะรายการไฟล์ในระบบกริดเท่านั้น ไม่ได้รวมถึงการโอนถ่ายไฟล์ข้อมูล ซึ่งงานในส่วนนี้จะเป็นหน้าที่ของ กริดเอฟทีพีโดยจะใช้ในรายการที่เร็พพลิก้าแคทตาล็อกจัดเก็บอยู่ซึ่งจะเก็บเป็นที่อยู่ของไฟล์ในภาพแบบกริดเอฟทีพีในกรณีที่ไฟล์นั้นอยู่ต่างไซต์ เช่น gridftp://Ahost.su.ac.th:2222/nfs/path และเก็บเป็นพาธ (Path) แบบปรกติในกรณีไฟล์จัดเก็บอยู่ที่ โลคอล

จากข้างต้นเราก็อาจสรุปได้ว่าโครงสร้างของระบบเร็พพลิก้าจะประกอบด้วยหน่วยทำงานหลักๆ สองส่วนคือ ส่วนของเร็พพลิก้าที่ทำงานอยู่ในระดับโลคอลซึ่งจะคอยจัดการกับไฟล์ในส่วนของตน และส่วนที่ทำหน้าที่เป็นดัชนี (Index) ของระบบ ซึ่งจะคอยจัดลำดับโดยจะทำการปรับปรุงข้อมูลของตนโดยปรับตามข้อมูลที่ได้รับแจ้งจาก เร็พพลิก้า เมเนจเมนท์ ในระดับโลคอล ลักษณะของเร็พพลิก้าจึงเป็นแบบลำดับชั้น (Hierarchical) คล้ายกับระบบโดเมนเนม (DNS)

ตัวอย่างการประยุกต์การใช้งานเร็พพลิก้าแคทตาล็อกและกรณีศึกษาโครงการที่เกี่ยวกับดาต้า กริด ในปัจจุบันการประยุกต์ใช้งานระบบ เร็พพลิก้าแคทตาล็อกยังใช้กันมากในงานทางวิทยาศาสตร์ขั้นสูงที่ประกอบด้วยข้อมูลจำนวนมากระดับเทราไบต์ (Terabyte) ถึงพีต้าไบต์ (Petabytes) (1 เทราไบต์ เท่ากับ 1,000 เมกะไบต์ 1 พีต้าไบต์ เท่ากับ 1,000 เทราไบต์) ซึ่งเป็นที่แน่นอนว่าข้อมูลมากมายมหาศาลขนาดนี้คงจะเป็นไปได้ยากที่จะจัดเก็บในที่ที่เดียวนอกจากจะเป็นการแบ่งปันทรัพยากรในด้านของพื้นที่เก็บข้อมูลร่วมกันแล้ว ยังอำนวยความสะดวกในการแบ่งปันข้อมูลทางวิทยาศาสตร์ร่วมกันอีกด้วย

ถ้าจะกล่าวกันจริงๆ แรงจูงใจที่ทำให้เกิดระบบกริดขึ้นจะมาจากงานทางด้านวิทยาศาสตร์ที่ต้องการพลังในการประมวลผลมหาศาลและพื้นที่เก็บข้อมูลขนาดใหญ่ ในปัจจุบันจะมีงานวิจัยทางด้านฟิสิกส์ ชีววิทยา และทางดาราศาสตร์ ที่ใช้งานระบบดาต้ากริดกันอยู่ เช่น โครงการ High Energy Physic (HEP) ที่ศึกษาการชนกันของอนุภาค ซึ่งจะให้ข้อมูลดิบประมาณ 5 – 8 พีต้าไบต์ ต่อปี ตลอดจนโครงการไบโอเมติคัลและเอิธ (Biomedical and Earth) และโครงการเฝ้าสังเกตการทางวิทยาศาสตร์ (Observation Sciences) เป็นต้น ทั้งหมดได้ใช้งานระบบดาต้ากริดจากโครงการยุโรปดาต้ากริด (European Data Grid) นอกจากนี้ก็ยังมีโครงการทางด้านดาต้ากริดอีกหลายโครงการ เช่น โครงการฟิสิกส์กริด (Grid Physics) เครือข่าย (GriPhyN) ปัจจุบันมีโครงการวิจัยทางฟิสิกส์ที่ใช้งานระบบดาต้ากริดของโครงการกริดฟิซ์นี้อยู่จำนวน 4 โครงการที่ใช้งาน โครงการซีเอ็มเอส (CMS) และ แอทลาส (ATLAS) อันเป็นส่วนหนึ่งของโครงการ Large

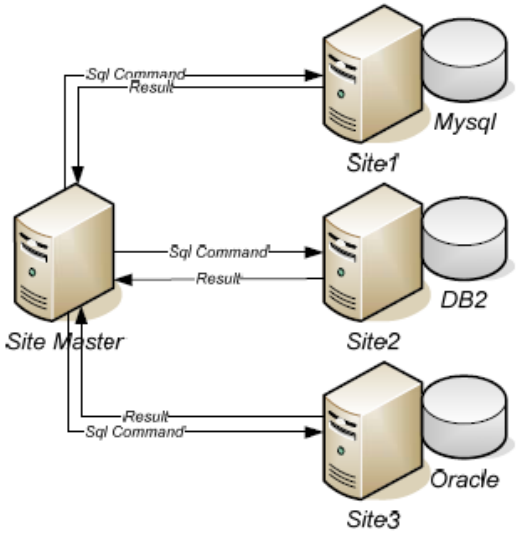
Hadron Collider (LHC) ที่สถาบันวิจัยนิวเคลียร์ของยุโรป (CERN)นครเจนีวา และมีโครงการในทำนองเดียวกันนี้อีกในประเทศสหรัฐ คือโครงการฟิสิกส์ดาต้ากริด (Particle Physics Data Grid,PPDG)

นอกจากโครงการทางด้านฟิสิกส์ยังมีโครงการทางด้านดาราศาสตร์ที่ทำการวิจัยเกี่ยวกับซุปเปอร์โนวาการค้นหาสัญญาณพัลซัล (Pulsars) และอินสไปลิงไบนารีสตาร์ และการสำรวจสโลนดิจิทัลสกาย (Sloan Digital Sky Survey, SDSS) ศึกษาเกี่ยวกับดวงดาวและกาแลกซีโครงการระบบกริดโลก (The Earth Systems Grid ,ESG) ที่ทำการศึกษาเกี่ยวกับอากาศ โดยข้อมูลจะถูกเก็บรวบรวมจากทั้งทางภาคพื้นดิน จากภาพถ่ายดาวเทียม และจากการจำลอง นักวิทยาศาสตร์ได้ใช้ เร็พพลิก้าแคทตาล็อก เผยแพร่ข้อมูลที่ได้มาให้กับนักวิจัยในกลุ่มของตน นอกจากยังมีกลุ่มที่ศึกษาวิจัยเกี่ยวการเกิดแผ่นดินไหว เช่น โครงกาจำลองการเกิดแผ่นดินไหวของโลก (Network for Earthquake Engineering Simulation, NEESgrid) เป็นต้น

3.2 อ็อกซ่าดีเอไอ (OGSA-DAI)

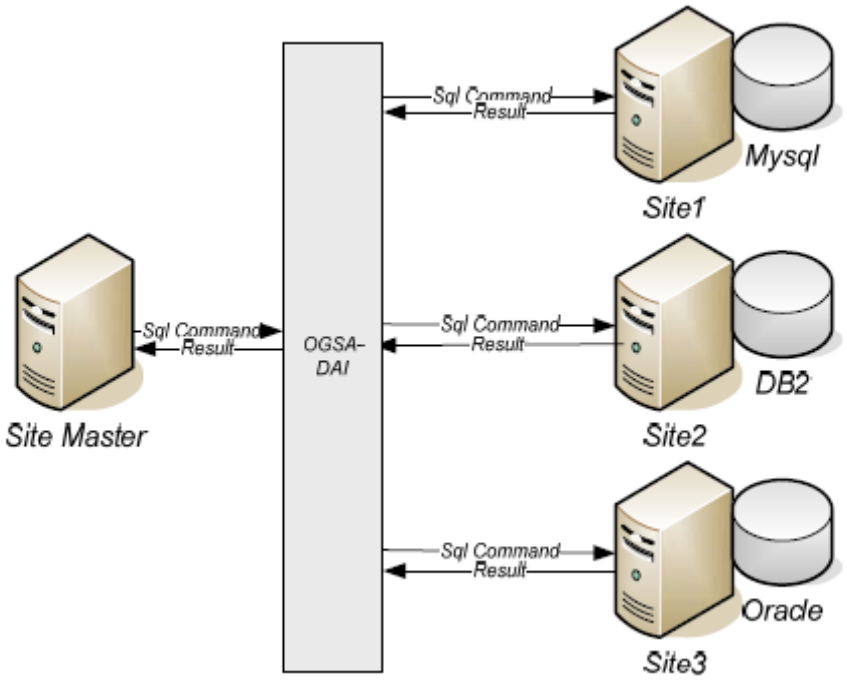
ในทุกวันนี้องค์กรต่างๆ มีความต้องการใช้ทรัพยากรทางด้านเทคโนโลยีสารสนเทศเพิ่มขึ้นอย่างรวดเร็ว ซึ่งอยู่ในภาพของความรวดเร็วในการประมวลผลและความถูกต้องของข้อมูลสารสนเทศที่จะได้รับ โดยที่ความต้องการนั้นได้ทำให้ก่อเกิดหลายๆเทคโนโลยีในปัจจุบันนี้และหนึ่งในเทคโนโลยีที่มีประสิทธิภาพที่สามารถตอบสนองความต้องการเหล่านั้นได้อย่างครบถ้วนก็คือการประมวลผลด้วยระบบกริด (Grid Computing)

ระบบกริดเป็นการใช้ทรัพยากรในการประมวลผลและข้อมูลที่อยู่ต่างองค์กรที่มีโครงสร้างแตก ต่างทั้งหน่วยประมวลผลและเครือข่ายให้เหมือนว่าเป็นองค์กรเสมือนที่มีเพียงหนึ่งเดียวซึ่งจะทำให้ง่ายต่อการใช้งานทรัพยากรที่มีอยู่ในองค์กรนั้นๆอย่างมีประสิทธิภาพโดยองค์ประกอบพื้นฐานของในแต่ละองค์กรต่างก็มีการใช้ข้อมูลจำนวนมากซึ่งได้ถูกจัดเก็บไว้ในฐานข้อมูลที่มีอยู่หลากหลายชนิดไม่ ว่าจะเป็น ออราเคิล, ดีบีทู, เอสคิวแอล เซิร์ฟเวอร์, มายเอสคิวแอล ก็ตาม โดยที่ความต้องในข้อมูลสารสนเทศในปัจจุบันนี้ได้มีเพิ่มขึ้นทุกวัน แต่ในบางครั้งข้อมูลเหล่านั้นอาจจะอยู่ต่างที่กันทำให้เกิดปัญหาในเรื่องของการจัดการฐานข้อมูล เช่น การใช้งาน, ความปลอดภัย รวมไปถึงความยุ่งยากในการเขียนโปรแกรมเพื่อติดต่อกับฐานข้อมูลที่มีอยู่อย่างกระจายกันตามองค์กรหรือหน่วยงานต่างๆ

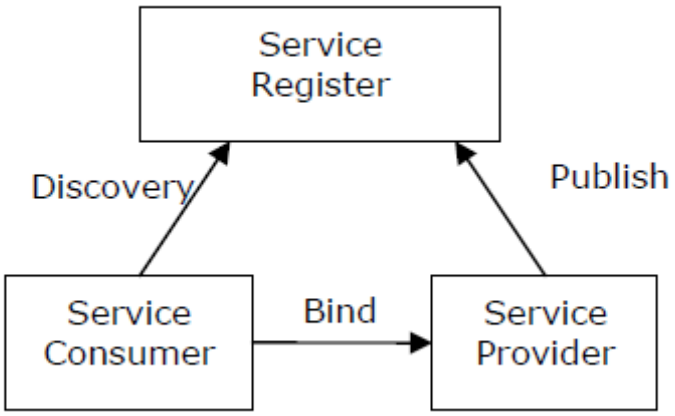


ภาพที่ 4 การคิวรี่ข้อมูลแบบไม่ใช่อ็อกซ่าดีเอไอ

ซึ่งต่อมาเมื่อระบบกริดได้เข้ามามีบทบาทจึงได้มีแนวคิดประยุกต์การใช้ฐานข้อมูลกับระบบ กริด ซึ่งประโยชน์ของการประยุกต์แนวคิดนี้ก็คือสามารถทำให้เข้าถึงและควบคุมข้อมูลที่อยู่ต่างที่กันเสมือนว่าข้อมูลหรือฐานข้อมูลเหล่านั้นอยู่ที่เดียวกัน เพื่อลดปัญหาทางด้านการจัดการฐานข้อมูลที่ได้กล่าวไว้ข้างต้น บนระบบกริดนั้นก็มีซอฟต์แวร์ในเรื่องของการจัดการฐานข้อมูลเพื่อความง่ายต่อการเข้าถึงและควบคุมฐานข้อมูลที่มีความแตกต่างกันไม่ว่าจะเป็นโครงสร้างหรือข้อมูล โดยรวมไปถึงชนิดของฐานข้อมูลบนระบบกริด นั่นก็คือ Open Grid Services Architecture Data Access and Integration (OGSA-DAI) (Kodeboyina and Plale, 2003) ซึ่งพัฒนาโดยใช้ฐานข้อมูลประเทศอังกฤษ ซึ่งทำหน้าที่เป็นส่วนเชื่อมประสาน (Interface) ระหว่าง ไซท์ (Site) ที่ต้องการคิวรี่ข้อมูลกับไซท์ ที่มีข้อมูล จากภาพที่ 4 พบว่าถ้าเราต้องการข้อมูลที่อยู่ต่างที่กันเราจะต้องทำการคิวรี่ข้อมูลไปทุกๆ ที่ที่เราเก็บข้อมูลไว้ แต่เมื่อมี อ็อกซ่าดีเอไอได้อำนวยความสะดวกในเรื่องของการรวบรวมทรัพยากรข้อมูลที่อยู่ต่างที่กันเพื่อให้ง่ายต่อการเข้าถึง โดยจากภาพที่ 5 นั้นเราสามารถทำการคิวรี่ข้อมูลผ่านอ็อกซ่าดีเอไอ เซอร์วิสได้



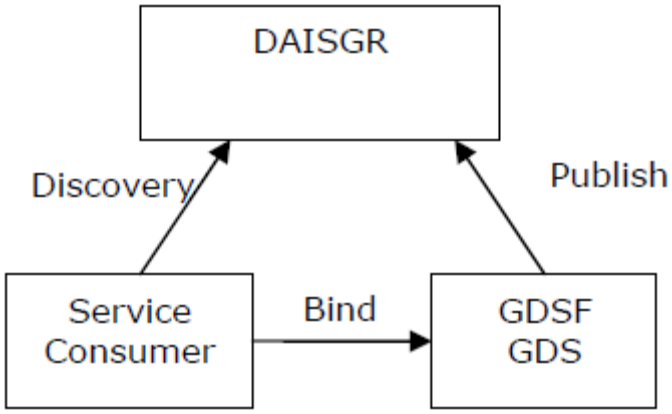
ภาพที่ 5 การคิวรี่ข้อมูลผ่านอ็อกซ่าดีเอไอ



ภาพที่ 6 การทำงานทั่วไปในแบบ SOA

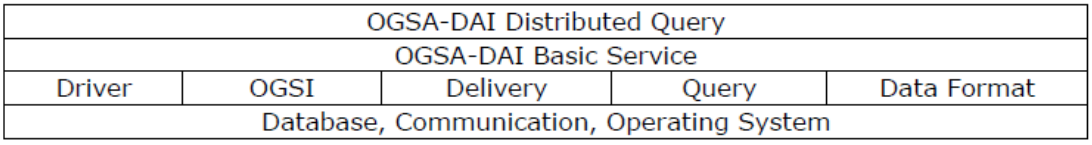
อ็อกซ่าดีเอไอมีรูปแบบการทำงานแบบ เอสโอเอ ที่ถูกสร้างโดยใช้กริดเซอร์วิสซึ่งเหมือนกับเว็บเซอร์วิส แต่ได้เพิ่มความสามารถบ้างอย่างเข้าไปเช่น ความปลอดภัย, การแบ่งภาระ

งาน เป็นต้น ซึ่ง ส่วนต่อไปเป็นการแสดงการเปรียบเทียบการทำงานของ อ็อกซ่าดีเอไอ และการทำงานของ เอสโอเอ



ภาพที่ 7 การทำงานของอ็อกซ่าดีเอไอ

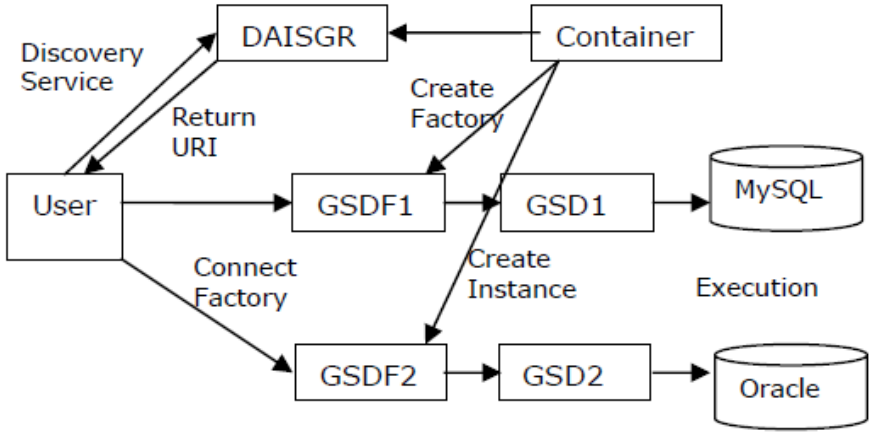
เนื่องจาก อ็อกซ่าดีเอไอ นั้นมีโครงสร้างแบบเอสโอเอดังนั้นภาพแบบการทำงานจึงมีลักษณะคล้ายคลึงกันโดยภาพที่ 6 และ 7 แสดงให้เห็นถึงความแตกต่างของโครงสร้างแบบเอสโอเอแบบทั่วไปเช่น เว็บเซอร์วิส และที่ประยุกต์กันอ็อกซ่าดีเอไอโดยการทำงานนั้นมีลักษณะที่เหมือนเดิมแต่มีชื่อที่ต่างออกไปทั้งนี้เพื่อให้เกิดความเหมาะสมและความเป็นลักษณะเฉพาะของอ็อกซ่าดีเอไอการทำงานของอ็อกซ่าดีเอไอขั้นตอนแรกคือเมื่อมีความต้องการเรียกใช้เซอร์วิสดังนั้นจะทำการค้นหาเซอร์วิสที่ต้องการโดยไปค้นที่ DAI Service Group Registry (DAISGR) ที่ได้ถูกลงทะเบียนไว้ก่อนหน้านี้จาก Grid Data Service Factory (GDSF) ขั้นตอนถัดไป เมื่อทราบถึงที่อยู่และวิธีเรียกแล้วก็จะทำการเรียกใช้งาน GDSF ตัวนั้นๆได้โดยตรง



ภาพที่ 8 แสดงสถาปัตยกรรมของอ็อกซ่าดีเอไอ

โดยภาพที่ 8 แสดงภาพสถาปัตยกรรมของอ็อกซ่าดีเอไอโดยเริ่มจากส่วนล่างสุดนั้นเป็นชั้นของการประมวลผลระบบล่างเช่น ระบบปฏิบัติการ, การติดต่อสื่อสาร และ ฐานข้อมูลซึ่งระดับ

ล่างนี้เป็นส่วนประกอบที่สำคัญที่เป็นทรัพยากรหรือการทำงานขั้นพื้นฐานของระบบชั้นต่อมาเป็นส่วนที่เป็นฟังก์ชั่นต่างๆที่เพื่ออำนวยความสะดวกในการติดต่อเซอร์วิสของอ็อกซ่าดีเอไอเช่น ไดร์เวอร์ (Driver) ที่เป็นตัวประสานการทำงานของเซอร์วิสกับฐานข้อมูลชนิดต่างๆ และ คิวรี่ (Query) นั้นทำหน้าที่ส่งคิวรี่ในระดับล่างไปยังฐานข้อมูลที่ต้องการและส่งผลกลับผ่านฟังก์ชั่นเดริเวอรี่ (Delivery) ในเซอร์วิสของอ็อกซ่าดีเอไอ ซึ่งในชั้นที่ 3 และ 4 เป็นเซอร์วิสขั้นพื้นฐานของอ็อกซ่าดีเอไอซึ่งอยู่ในภาพของเอพีไอที่ไว้คอยอำนวยความสะดวกในการเขียนโปรแกรมประยุกต์ซึ่งทำหน้าที่เป็นฟังก์ชั่นพื้นฐานของอ็อกซ่าดีเอไอรวมทั้งกระจายคิวรี่ที่ต้องการประมวลผลไประบบกริด

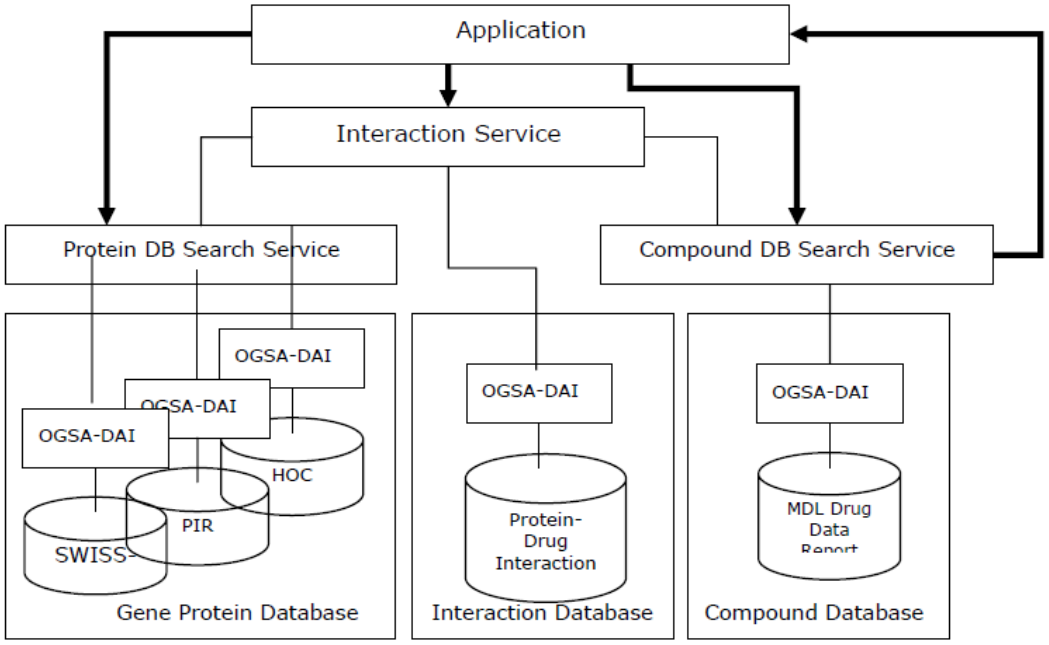


ภาพที่ 9 แสดงการทำงานของ อ็อกซ่าดีเอไอ

จากภาพที่ 9 สามารถอธิบายการทำงานของอ็อกซ่าดีเอไอได้ดังนี้การทำงานของอ็อกซ่าดีเอไอนั้นเริ่มต้นต้องมีการลงทะเบียน (Register) เซอร์วิสที่ต้องการกับDAISGR เพื่อให้ผู้ใช้ทราบถึงยูอาร์ไอ (URI) ในการเรียกใช้งาน โดยที่ฝั่งผู้ให้บริการจะทำการเปิดคอนเท็นเนอร์ (Container) ขึ้นมาเพื่อรอให้บริการซึ่งจะทำให้ แฟคทอรี่ (GSDF) ของแต่เซอร์วิสนั้นพร้อมใช้งาน เมื่อผู้ใช้ต้องการใช้เซอร์วิสของอ็อกซ่าดีเอไอจะทำการค้นหายูอาร์ไอจาก DAISGRเมื่อได้ผลลัพธ์แล้วผู้ใช้ก็สามารถจะติดต่อไปยังแฟคทอรี่ของเซอร์วิสที่ต้องการได้และหลังจากนั้นเมื่อต้องการประมวลผลเซอร์วิส แฟคทอรี่ก็จะทำการสร้าง อินสแตนท์ จีเอสดี (Instance GSD) ให้แต่ละผู้ใช้ไว้เพื่อประมวลผลกับฐานข้อมูลที่มีอยู่ในระบบ และเมื่อได้รับผลลัพธ์ก็จะถูกจะส่งผ่านมาทาง อินสแตนท์ที่ได้ทำการติดต่อไว้แล้ว

การใช้งานของอ็อกซ่าดีเอไอนั้นได้ทำการทดสอบการทำงานทั้ง วินโดว์2000 (Windows 2000), วินโดว์เอ็กซ์พี (Windows XP), เร็ดแฮช 9.0 (RedHat 9.0) หรือ ซัน โซล่าริช เวอร์ชั่น 8 (Sun Solaris Version 8) โดยผู้ใช้งานควรมีความรู้ทางด้าน จาวา (Java), เอ็กซ์เอ็มแอล(XML), อ็อกซ่า (OGSA) และ อ๊อกซิ (OGSI)

ตัวอย่างการประยุกต์อ็อกซ่ามาใช้ในงานจริง ตัวอย่างการนำอ็อกซ่ามาใช้งานทางด้านวิทยาศาสตร์ดูได้จากโครงการ ไบโอกริด (BioGrid) ของมหาวิทยาลัยโอซาก้า ได้ทำการรวบรวมฐานข้อมูลต่างทางไบโอโดยใช้อ็อกซ่าจุดประสงค์ก็เพื่อหาตัวยาโดยใช้ฐานข้อมูลที่อยู่ในความร่วมมือกับมหาวิทยาลัยทั้ง 11-แห่งที่ถูกเก็บอยู่ในระบบโครงสร้างที่แตกต่างกันเช่น ชีววิทยา,เวชกรรม และการแพทย์ ซึ่งโครงการนี้ได้ใช้ อ็อกซ่าถูกนำมาใช้เป็นมิดเดิลแวร์ เพื่อเชื่อมโยงฐานข้อมูลภาพแบบต่างเข้าด้วยกัน ดังแสดงตามภาพที่ 10



ภาพที่ 10 โครงสร้างการทำงานของโครงการไบโอกริดข้อมูลภาพจากมหาวิทยาลัยโอซาก้าประเทศญี่ปุ่น

อีกหนึ่งก็คือโครงการยีนส์กริด (GeneGrid) เป็นโครงการที่ทำการวิเคราะห์ในระดับไมโครกับข้อมูล จำนวนมากซึ่งเป็นในเรื่อง โรคมะเร็งและโรคที่ติดต่อทางพันธุกรรม ทั้งนี้ได้นำฐานข้อมูลที่เหล่านั้นมาจากโครงการอีซายน์ (e-Science) ของมหาวิทยาลัยควีน (Queen University

of Belfast) โดยได้ใช้อ็อกซ่าดีเอไอทำการรวบรวมข้อมูลเหล่านั้นเพื่อง่ายในการวิเคราะห์และประมวลผล

เนื่องจากปัจจุบันนี้ไม่ว่าทางวิทยาศาสตร์หรือทางธุรกิจต่างก็มีความจำเป็นต้องได้รับความร่วมมือกับหลายๆหน่วยงานทั้งภายนอกหรือภายในองค์กร ซึ่งตรงตามแนวคิดของการประมวลผลกริด โดยส่วนมากแล้วทรัพยากรเหล่านั้นจะเป็นข้อมูลที่ในภาพแบบของฐานข้อมูลซึ่งปัญหาโดยทั่วไปของการจัดการข้อมูลในระบบที่มีความแต่ต่างกันในเรื่องของชนิดของฐานข้อมูล, ภาระการทำงาน, ความยุ่งยากการเปลี่ยนแปลงโครงสร้างฐานข้อมูล และความปลอดภัยของการเข้าใช้งาน โดย อ็อกซ่าดีเอไอได้เป็นสิ่งที่ช่วยอำนวยความสะดวกเพื่อง่ายต่อการใช้งานฐานข้อมูลเหล่านั้นในระบบกริดอย่างมีประสิทธิภาพและทำให้สามารถเปลี่ยนแปลงโครงสร้างของฐานข้อมูลในระดับล่างได้โดยไม่มีผลกระทบต่อโปรแกรมประยุกต์ที่ได้พัฒนาซึ่งทำให้มีความคล่องตัวในการใช้งานฐานข้อมูลในระบบที่อยู่อย่างหลากหลาย โดยพบว่าอ็อกซ่าดีเอไอได้เข้ามามีบทบาทกับประมวลผลทางด้านวิทยาศาสตร์ซึ่งต่อไปในอนาคตสิ่งเหล่า นี้จะเข้ามามีบทบาทกับงานทางด้านธุรกิจมากขึ้นในเวลาอันใกล้

3.3 กริดเอฟทีพี (GridFTP)

กริดเอฟทีพีเป็นโพรโตคอลและเครื่องมือที่เกิดจากความน่าเชื่อถือของกริดซึ่งประกอบด้วยการสภาพแวดล้อมที่รวดเร็วปลอดภัยมีประสิทธิภาพและความน่าเชื่อถือของกลไกการเคลื่อนย้ายข้อมูล ดังนั้นทีมโกลบัสโปรเจ็กต์ ได้ทำการสำรวจความน่าจะเป็นของโพรโตคอลและ เทคนิค แล้วนำมาเพิ่มลงในเอฟทีพี ให้เกิดเป็นโพรโตคอลที่น่าเชื่อถือโดยสถาปัตยกรรมของกริดเอฟทีพี (GridFTP) แสดงในภาพที่ 11 ซึ่งมีคุณสมบัติดังนี้

- กริด ซิคยูริที อินฟาสตรักเจอร์ (Grid Security Infrastructure, GSI) และ เคอร์เบโรสซัพพอร์ต (Kerberos Supports) กริดเอฟทีพีมีความทนทานและยืดหยุ่นของสิทธิในการใช้งานเพราะรองรับสิทธิการใช้งานทั้งในแบบ จีเอสไอ และ เคอร์เบโรส ซัพพอร์ต ซึ่งผู้ใช้สามารถควบคุมและกำหนดค่าความหลากหลายในการเข้าถึงข้อมูลได้

- เธิดพาร์ทิ คอนโทรล (Third-party control of data transfer) ในการจัดลำดับของกลุ่มข้อมูลขนาดใหญ่ในการกระจายการติดต่อซึ่งที่จำเป็นก็คือการจัดให้มีการควบคุมแบบเธิดพาร์ทิ (third-party) ระหว่างเซิร์ฟเวอร์ที่เก็บข้อมูล โดยกริดเอฟทีพีจัดให้มีความสามารถโดยการเพิ่ม GSSAPIsecurity เพื่อใช้ในการย้ายข้อมูลแบบเธิดพาร์ทิ ซึ่งมีสามารถและข้อจำกัดเช่นเดียวกับมาตรฐานของเอฟทีพี

- แพแร็ลแล็ล ดาต้า ทรานส์เฟอร์ (Parallel data transfer) ในการเชื่อมต่อขนาดใหญ่จะมีการใช้ มัลติเปิ้ลทีซีพีสตรีม (Multiple TCP Stream) ซึ่งสามารถหาข้อตกลงเพื่อให้เกิดการเคลื่อนย้ายข้อมูลแบบซิงเกิลทีซีพี สตรีม (Single TCP Stream) โดยความต้องการของทั้ง เซิร์ฟเวอร์เดียว (Single Client) และ เซิร์ฟเวอร์เดียว และระหว่างสองเซิร์ฟเวอร์กริดเอฟทีพี มีการสนับสนุนการเคลื่อนย้ายข้อมูลแบบขนานผ่านทางขอบเขตคำสั่งเอฟทีพี และ ขอบเขตช่องทางข้อมูล

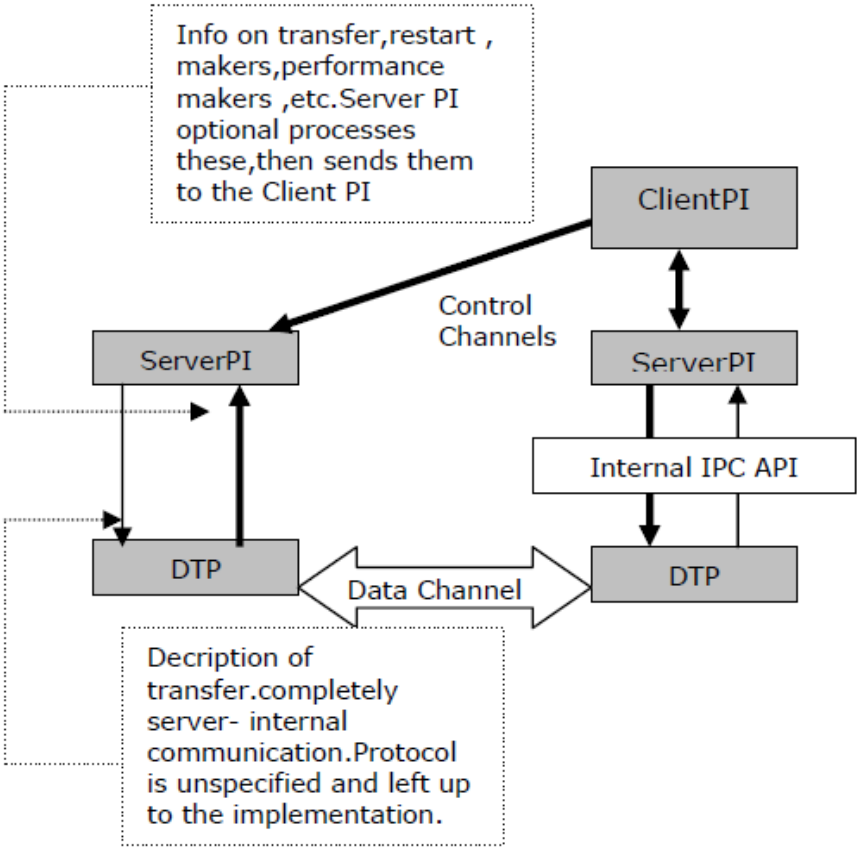
- สไทรพดาต้าทรานส์เฟอร์ (Striped data transfer) การแบ่งย่อยข้อมูลไปยังหลายเซิร์ฟเวอร์แต่สามารถส่งรวมไปในแบนด์วิทธ์กริดเอฟทีพีรองรับการสไทรพ (Striped) การเคลื่อนย้ายข้อมูลตามขอบเขตที่กำหนดในข้อตกลงของกริดได้

- พาร์เชียลไฟล์ทรานส์เฟอร์ (Partial file transfer) หลายแอพพลิเคชันต้องการย้ายข้อมูลแค่บางส่วนของไฟล์อย่างไรก็ตามเอฟทีพีมาตรฐานต้องการให้แอพพลิเคชันย้ายไฟล์หรือคงเหลือไฟล์จากตำแหน่งเริ่มต้นไปยังส่วนที่ต้องการ กริดเอฟทีพีแนะนำ เอฟทีพีใหม่ให้มีคำสั่งรองรับการย้ายจากไฟล์ดั้งเดิมได้

- รองรับความน่าเชื่อถือในการเคลื่อนย้ายไฟล์ ความน่าเชื่อถือในการเคลื่อนย้ายไฟล์เป็นสิ่งสำคัญสำหรับหลายๆแอพพลิเคชันในการจัดการกับข้อมูลโดยปรกติวิธีการในการค้นหาข้อมูลในกรณีจัดการเมื่อเครือข่ายล้มชั่วคราว, เซิรฟ์เวอร์เอาท์เอ็จ (Server Outage) และอื่นๆโดยปรกติเอฟทีพี มาตรฐานจะมีลักษณะการเริ่มต้นดึงข้อมูลให้ใหม่ กริดเอฟทีพีโพรโตคอลมีพฤติกรรมและความสามารถนี้ร่วมด้วยเช่นกัน

- สามารถควบคุมขนาดของ ทีซีพี บัฟเฟอร์ (TCP buffer) นี้เป็นพารามิเตอร์ที่มีอิทธิพลมาในการเข้าถึงขนาดแบนด์วิทธ์ให้มากที่สุดด้วย ทีซีพี/ไอพี (TCP/IP) โพรโตคอลสามารถสนับสนุนขนาดบัฟเฟอร์ ที่เหมาะสมให้อัตโนมัติและยังพัฒนาเพิ่มเติมในกริดเอฟทีพีโดยเป็นการเจรจาระหว่าง เอ็นซีเอสเอ (NCSA) กับ แอลเอเอ็นแอล (LANL) เพื่อให้สามารถทำงานร่วมกันได้

- เครื่องมือในการรวมมือกัน กริดเอฟทีพีโปรโคคอลสามารถเรียก, เริ่มต้นใหม่และมีการแสดงประสิทธิภาพเพื่อส่งข้อมูลกลับซึ่งแสดงด้วยตำแหน่งสั้นๆ



ภาพที่11 แสดงสถาปัตยกรรมของกริดเอฟทีพี

3.4 ค็อกคิท (COGKITS)

ในการพัฒนาหลายๆแอ็พพลิเคชั่นต้องการทำโปรแกรมบนกริดซึ่งมีเฟรมเวริคที่มีการเติบโตของโพรโตไทพ์ อย่างรวดเร็วจึงเกิดการพัฒนาค็อกคิทขึ้นมาใช้ในการพัฒนาโปรแกรมบนกริด โดยโครงการของค็อกคิทรวมกับกริดซอฟต์แวร์บนโกลบัสทูลคิท(Laszewski and Jackson,2002) และความร่วมมือของจาวา หรือ ไพทอล (Python) ทำให้การพัฒนาโปรแกรมกริดเซอร์วิสง่ายขึ้น, รวดเร็วขึ้นและสามารถนำโค้ดโปรแกรมเก่ามาใช้ใหม่ได้ อีกทั้งทำให้เว็บเซอร์วิสเป็นส่วนขึ้นของกริด

เนื่องจากค็อกคิทมีอินเตอร์เฟสกับโกลบัส ทูลคิท และโปรแกรมเมอร์บนกริดสามารถใช้เทคโนโลยีได้อย่างกว้างขาง เช่น การรองรับเหตุการณ์(Event) และข้อยกเว้น (Exception) บนจาวา หรือมี สวิง แว็พเพอร์ (SWIG wrapper) ในไพทอลเพื่อความรวดเร็วในการโค้ดโปรแกรมตามกฎ ดังนั้นค็อกคิทไม่ใช้เป็นแค่เพียงเอพีไอแต่เป็นการสนับสนุนการเข้าถึงเฟรมเวริคให้ได้มากยิ่งขึ้น

จาวาค็อกคิทมีการเก็บแพช (Patches) และเปลี่ยนโพรโตคอลในโกลบัสให้เป็นไปตามปัจจุบัน ค็อกคิทได้รับการสนับสนุนจาก มหาวิทยาลัยอินเดียนาในการแก้ไขโค้ดให้สามารถติดต่อกับเซอร์วิสต่างๆที่เกิดบนโกลบัสได้ โดยคุณสมบัติต่างๆนี้ได้ถูกนำมาไว้ใน จาวาค็อกคิทดังนั้นการแก้ไขโกลบัสเล็กๆน้อยๆจะไม่มีผลโดยปัจจุบันมีการพัฒนาจาวาค็อกคิทจะมีการให้บริการในเรื่องของความปลอดภัย (GSI), การควบคุมงานระยะไกล (GRAM), การเข้าถึงข้อมูลระยะไกล (GridFTP), บริการการเข้าถึงข้อมูล (MDS), และการเก็บรักษา (Certificate myProxy)

แนวคิดทฤษฎีการสร้างภาพ 3 มิติ

ในการสร้างภาพ 3 มิติ นั้นเกิดจากการคำนวณทางคณิตศาสตร์ในระดับจุดทศนิยมของชิ้นส่วนต่าง ๆ ที่เกิดขึ้นในแกน 3 แกนคือ X,Y,Z โดยชิ้นส่วนที่ได้คือ Polygon หากมีจำนวน Polygon ในแกน X,Y,Z มากก็จะทำให้ภาพ 3 มิติที่ได้มีความละเอียดสูงมากตามจำนวนของ Polygon ซึ่งการประมวลผลแบบ 3 มิติในปัจจุบันได้มีชุดคำสั่ง OpenGL เพื่อช่วยให้ผู้พัฒนานำชุดคำสั่งมาใช้งานได้สะดวกขึ้น (Mechtly, Rooker and Mast 2001)

แนวคิดทฤษฎีการคำนวณเส้นทางการเคลื่อนที่ของภาพเคลื่อนไหวโดยใช้การประมวลผลแบบกระจาย

การคำนวณหาเส้นทางการเคลื่อนที่นั้นจำเป็นต้องใช้เวลามากแต่ได้มีการนำเอาคุณสมบัติของการกระจายข้อมูลและประมวลผลแบบขนานมาช่วยจึงทำให้ใช้เวลาในการทำงานลดลง และเวลาในการคำนวณด้วยคอมพิวเตอร์เพียงเครื่องเดียว หลักการคือมีเครื่องที่ทำการกระจายข้อมูลไปยังแต่ละจุดที่ปีนจุดจัดการข้อมูลและจุดจัดการข้อมูลจะทำการแบ่งงานออกเป็นส่วนเล็ก ๆ เพื่อส่งไปประมวลผลในเครือข่ายของตนเองเมื่อประมวลผลเสร็จแล้วก็จะส่งผลลัพธ์กลับมายังจุดที่เป็นตัวจัดการ เมื่อจุดที่คอยจัดการข้อมูลได้รับข้อมูลครบก็จะรวมผลลัพธ์แล้วส่งกลับไปยังจุดกระจายเพื่อรวมผลลัพธ์ทั้งหมดและทำการแสดงผลตามที่ต้องการ (Sutaweesup and Poovarawan 2002)

งานวิจัยที่เกี่ยวข้อง

จากการสำรวจงานวิจัยในเรื่องการพัฒนาเกม 3 มิติ และเทคโนโลยีกริดพบข้อมูลดังนี้

ไพศาล สิมาเลาเต่า (2550 : บทคัดย่อ) ได้ศึกษาเกี่ยวกับ การพัฒนาโปรแกรมประยุกต์การรู้จำ โดยใช้การประมวลผลแบบกริด พบว่าการประมวลผลแบบกริดใช้เวลาในการประมวลผลน้อยกว่าการประมวลผลเพียงโปรเซสเดียว โดยถ้าแบ่งการทำงานออกเป็น 2 โปรเซส

จะมีอัตราเร็วเพิ่มขึ้นเฉลี่ย 1.59 ถึง 1.83 เท่า สำหรับการแบ่งออกเป็น 4 โปรเซส จะมีอัตราเร็วเพิ่มขึ้นเฉลี่ย 2.27 ถึง 3.22 เท่า และถ้าแบ่งออกเป็น 8 โปรเซส จะมีอัตราเร็วเพิ่มขึ้นเฉลี่ย 2.90 ถึง 6.33 เท่า ทั้งนี้ขึ้นกับตัวอักษรที่นำมาใช้เพื่อการเรียนรู้ ทั้งนี้ยังพบว่าการประมวลผลแบบกริดจะไม่เกิดประโยชน์เท่าที่ควรถ้าตัวอักษรที่นำมาใช้มีขนาดเล็กเกินไป

ศรชย โรจนวิภาต (2549 : บทคัดย่อ) ได้ศึกษาเกี่ยวกับ การพัฒนาระบบบริหารทรัพยากรกริดโดยใช้กริดโบรกเกอร์ ได้นำเสนอการออกแบบและพัฒนาสถาปัตยกรรมของกริดโบรกเกอร์ แบบ OGBA (Open Grid Broker Architecture) ซึ่งเป็นส่วนประกอบสำคัญในระบบประมวลผลแบบกริด ทำหน้าที่ในการค้นและคัดเลือกทรัพยากรให้เหมาะสมกับความต้องการของผู้ใช้ โดยทั้งระบบจะทำงานอยู่บนพื้นฐานของเว็บเซอร์วิส พร้อมทั้งนำเทคโนโลยีของมิดเดิ้ลแวร์ Globus Toolkit 4.0 มาใช้งาน เช่น OGSA-DAI, MDS เป็นต้น นอกจากนี้ยังเปรียบเทียบขั้นตอนวิธีในการแบ่งภาระงานและส่งงานข้ามแพลตฟอร์ม จากการทดลองพบว่าระบบที่นำเสนอสามารถเพิ่มขีดความสามารถในการประมวลผลให้กับระบบกริด และรองรับการขยายตัวของระบบได้เป็นอย่างดี

เปมปีย์อร วังไชยเลิศ (2549 : บทคัดย่อ) ได้ศึกษาเกี่ยวกับ การออกแบบระบบจัดเก็บและสืบค้นข้อมูลแบบกระจายโดยใช้กริดเทคโนโลยี พบว่าระบบที่ได้พัฒนาขึ้นใช้ข้อมูลมัลติมีเดียดิจิทัลเป็นข้อมูลทดสอบและใช้ทดสอบในระบบกริด จากผลลัพธ์พบว่าระบบนี้สามารถทำงานรวดเร็วกว่าระบบการค้นหาแบบลำดับอย่างชัดเจนอีกทั้งการเคลื่อนย้ายข้อมูลมัลติมีเดียดิจิทัลเพื่อจัดเก็บให้เหมาะสมกับกลุ่มผู้ใช้ที่คำนวณจากประวัติการใช้งานในอดีตทำให้ระบบสามารถทำงานได้รวดเร็วกว่าถึง 30% เมื่อเปรียบเทียบกับการจัดเก็บแบบเดิมซึ่งไม่มีการเคลื่อนย้ายข้อมูลในครั้งแรกและการเคลื่อนย้ายข้อมูลแบบสุ่มเพื่อการจัดเก็บข้อมูลในแต่ละเครื่องด้วยปริมาณข้อมูลที่เท่ากัน แนวความคิดนี้สามารถนำไปประยุกต์และขยายผลในการใช้งานได้เป็นอย่างดี

พีรพล เวทีกูล (2547 : บทคัดย่อ) ได้ศึกษาเกี่ยวกับ ต้นแบบระบบสืบค้นคืนข้อมูลแบบกระจายบนเครือข่ายการเชื่อมต่อกริด พบว่าระบบสืบค้นข้อมูลแบบกระจายที่ทำงานบนเครือข่ายการเชื่อมต่อกริด “ดีเวิร์ม” ซึ่งพัฒนาต่อเนื่องมาจากระบบสืบค้นข้อมูลแบบซีเควนท์ที่ทำงานอยู่บนโมเดลเวกเตอร์สเปส “เวิร์ม” ระบบดีเวิร์มพัฒนาอยู่บนระบบเครือข่ายเชื่อมต่อ ApGrid และมีโครงสร้างการทำงานของระบบแบบกระจาย และง่ายต่อการพัฒนา ต้นแบบดีเวิร์มที่พัฒนาได้มานี้มีความสำคัญอย่างยิ่ง เนื่องจากเป็นระบบต้นแบบสำหรับทดสอบงานวิจัยที่พัฒนาต่อเนื่องบนระบบสืบค้นข้อมูลแบบกระจายในด้านต่าง ๆ ต่อไป

บทที่ 3
วิธีการดำเนินการวิจัย

การพัฒนาเกม 3 มิติโดยใช้เทคโนโลยีกริด เป็นการพัฒนาระบบประมวลผลภาพกราฟิกแบบ 3 มิติ โดยใช้วิธีการประมวลผลแบบขนาน ซึ่งสามารถสรุปถึงขั้นตอนการวิจัยโดยมีขั้นตอนดังต่อไปนี้

1. ศึกษาข้อมูลเกี่ยวกับการพัฒนาเกมแบบ 3 มิติแบบต่าง ๆ

ผู้วิจัยเริ่มต้นทำการศึกษามาตรฐานของเกม 3 มิติแบบต่าง ๆ ที่มีใช้งานอยู่ในปัจจุบัน โครงสร้างของการสร้างภาพ 3 มิติ ศึกษาการทำงานของการสร้างภาพ 3 มิติ โดยใช้ library ต่างๆ เช่น DirectX OpenGL

จากการคัดเลือกโปรแกรมในการสร้างภาพ 3 มิติและ เกมเอนจิ้น ต่าง ๆ มากมายจนได้โปรแกรมที่เหมาะสมกับงานนี้ คือ Blender v.2.48 ซึ่งด้วยลักษณะของตัวโปรแกรมที่เป็น Open Source และความสามารถที่เหนือกว่าโปรแกรมในการสร้างภาพ 3 มิติอื่น ๆ ที่มีใช้กันอยู่ คือมีเกมเอนจิ้น เป็นของตัวเอง จึงทำให้สามารถพัฒนาผลงานเกมและทำการทดสอบไปได้ในเวลาเดียวกัน อีกทั้งในโปรแกรมที่สามารถเรียกใช้คำสั่งพื้นฐาน API จากตัวโปรแกรมได้โดยตรง หรือแม้กระทั่งเรียกใช้งานผ่าน Console ก็ตามซึ่งสามารถทำได้ง่ายและมีตัวอย่างให้เลือกใช้ได้มากมาย ที่สำคัญยังสามารถสั่งงานผ่านทาง Python Script ซึ่งเป็นภาษาคอมพิวเตอร์ใหม่และมีความหลากหลายในการเรียกใช้งาน

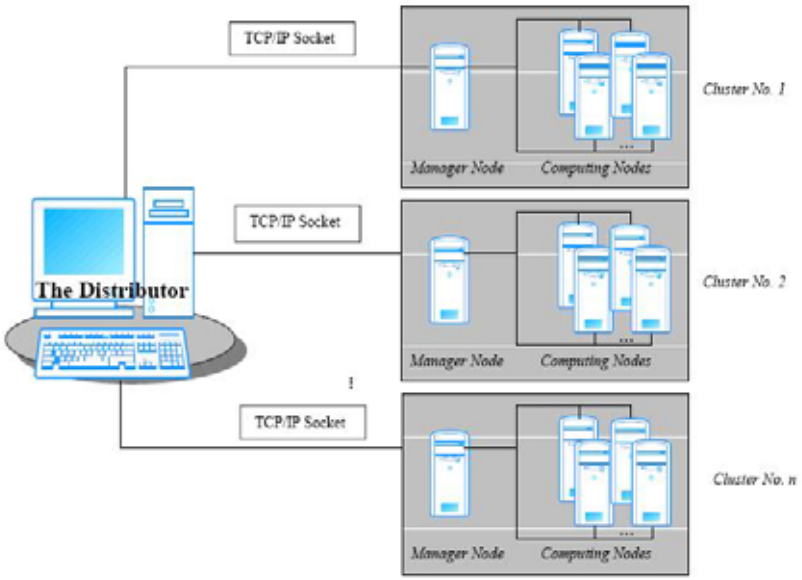
2. ศึกษาข้อมูลเกี่ยวกับระบบกริด

ผู้วิจัยได้เริ่มต้นทำการศึกษาเกี่ยวกับข้อมูลของระบบกริดโดยได้ศึกษารายละเอียดในส่วนของการจัดเตรียมระบบกริด ประเภทของกริด การทำงานของระบบกริด การใช้งานมิดเดิ้ลแวร์(Middleware) การส่งแมสเซสในระบบกริดโดยใช้ PVM (Parallel Virtual Machine) และ MPI (Message Passing Interface) เป็น library เป็นมาตรฐานในการส่งแมสเซส เพื่อให้คอมพิวเตอร์ในระบบกริดช่วยกันประมวลผลข้อมูล สถาปัตยกรรมของกริดทั้ง 3 ชนิดคือ สถาปัตยกรรมกริดแบบลำดับชั้น (N-Tier Grid Architecture) , Role-Based Grid Architecture (สถาปัตยกรรมกริดที่อ้างอิง

กับหน้าที่) และ Service-Based Grid Architecture (สถาปัตยกรรมกริดที่อ้างอิงกับเซอร์วิส) เพื่อใช้เปรียบเทียบถึงข้อดีข้อเสียของแต่ละชนิดและเลือกใช้งานได้อย่างเหมาะสมกับงานที่ทำ

จากการศึกษาระบบการทำงานของกริดพบว่าขั้นตอนการสื่อสารกันเพื่อคำนวณนั้นมีดังต่อไปนี้

- เมื่อจุดกระจายข้อมูล (Distributor) ทำการสร้างงาน (threads) ขึ้นมาก็จะทำการเชื่อมต่อไปยังจุดเชื่อมต่อ (Node) ต่าง ๆ และทำการแจกจ่ายงาน (threads) ไปยังจุดเชื่อมต่อต่าง ๆ ผ่านโปรโตคอล TCP/IP ดังแสดงในภาพที่ 12



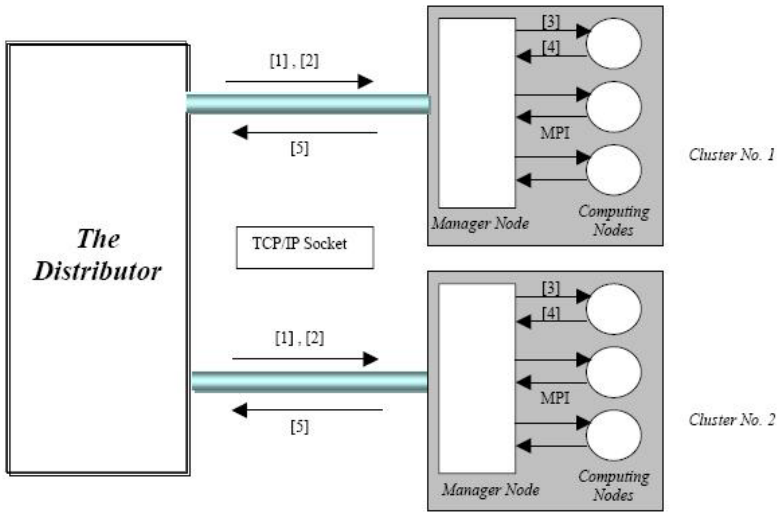
ภาพที่ 12 ระบบกริดที่จัดเตรียมไว้โดยผ่านโปรโตคอล TCP/IP ไปยังแต่ละ Node

- หลังจากสถาปนาการเชื่อมต่อแล้วจุดกระจายข้อมูล (Distributor) จะส่งข้อมูลจำนวนมากไปยังแต่ละกลุ่มย่อย (Cluster) ในขั้นตอนนี้จะทำการแบ่งข้อมูลเป็นส่วน ๆ ไปยังกลุ่มที่เล็กกว่า

- ภายในกลุ่มย่อย (Cluster) แต่ละกลุ่มจะมีจุดเชื่อมต่อ (Node) จากจุดกระจายข้อมูล (Distributor) ไว้คอยจัดการการเชื่อมต่อและแบ่งข้อมูลออกจำนวนหนึ่ง เรียกว่าจุดที่ทำการจัดการข้อมูล (Manager Node) เพื่อจัดส่งข้อมูลไปคำนวณทั้งหมด โดยที่จุดเชื่อมต่อ (Node) จะมีการคำนวณเส้นทางที่ใช้ในการรับ-ส่งข้อมูลร่วมกัน

- หลังจากที่คำนวณเสร็จสิ้นก็จะมีการส่งงานที่คำนวณเสร็จแล้วกลับมา โดยส่งกลับมายังจุดที่ทำการจัดการข้อมูล (Manager Node)

- จุดที่ทำการจัดการข้อมูล (Manager Node) จะรวบรวมผลลัพธ์ทั้งหมดและส่งต่อยังจุดกระจายข้อมูล (Distributor) ดังแสดงในภาพที่ 13



ภาพที่ 13 ขั้นตอนการสื่อสารในระบบกริดระหว่าง Cluster1 และ Cluster2

3. การเตรียมตัวอย่างสำหรับการทดลอง

ผู้วิจัยได้จัดสร้างเกมที่ใช้เป็นตัวอย่างในการทดลองด้วยโปรแกรม Blender ซึ่งเป็น Open Source โดยได้สร้างให้มีจำนวนโพลีกอนมากเพื่อใช้ในการทดสอบครั้งนี้

4. เครื่องมือและอุปกรณ์

4.1. อุปกรณ์เครือข่าย Hub/Switch 3Com 10/100

4.2. เครื่องคอมพิวเตอร์ส่วนบุคคลที่ใช้ในการวิจัยซึ่งมีคุณสมบัติดังนี้

- หน่วยประมวลผล 2.0 GHz
- หน่วยความจำ DDR RAM 1024 MB
- หน่วยความจำ HDD 160 MB
- การ์ดแสดงผล NVIDIA Geforce4 MX 440

5. ซอฟต์แวร์ที่ใช้ในการวิจัย

5.1. ระบบปฏิบัติการ Linux Ubuntu 8.0.10 Desktop Edition i368

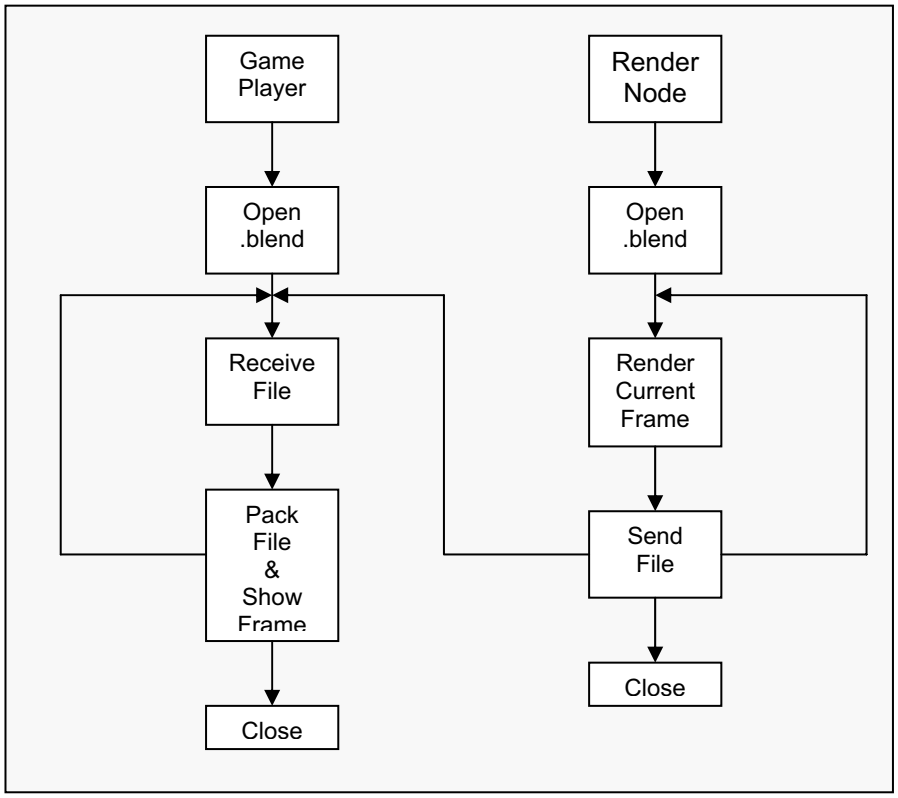
5.2. Blender v2.48 (3D Open Source) สำหรับทำภาพ 3 มิติ และนำมาสร้างเป็นเกมสำหรับใช้ทดสอบ

5.3. Python 2.5 สำหรับเรียกใช้งาน API ของ Blender v.2.48 และทำการรับส่งข้อมูลระหว่างเครื่องที่เล่นเกมดกับเครื่องที่ประมวลผล

5.4. vsftpd สำหรับรอรับการ FTP แฟ้มที่ประมวลผลแล้วจากเครื่องที่ประมวลผลภาพ 3 มิติ

6. ขั้นตอนการทำงานของโปรแกรมประมวลผล

เป็นขั้นตอนการทำงานที่ต้องทำไปพร้อม ๆ กันที่เรียกว่าการทำงานแบบขนานซึ่งมีรายละเอียดดังต่อไปนี้

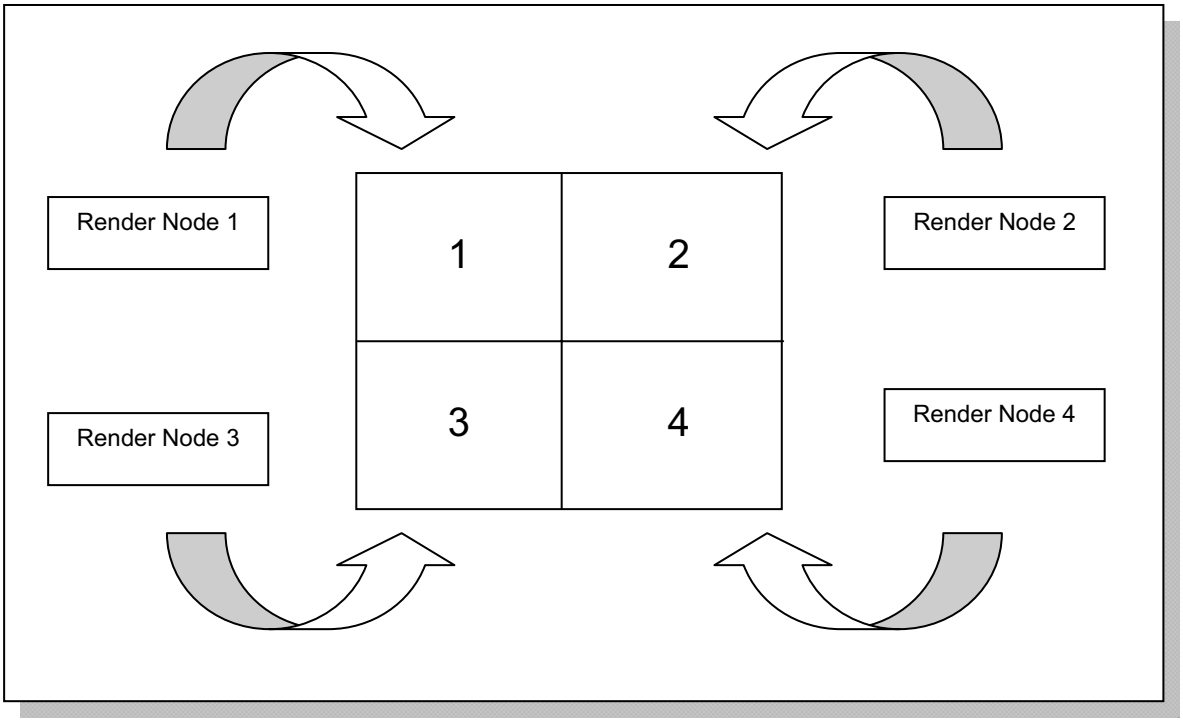


ภาพที่ 14 สรุปขั้นตอนการทำงานของโปรแกรมประมวลผลภาพ 3 มิติ จาก Game Player และ Render Node

จากภาพที่ 14 แสดงให้เห็นถึงขั้นตอนในการทำงานแบบขนานกันของโปรแกรมประยุกต์ในการประมวลผลภาพ 3 มิติ ซึ่งแบ่งออกเป็น 2 ฝั่งการทำงานและการทำงานทั้งหมดสามารถสรุปได้เป็น 3 ขั้นตอนดังนี้

6.1. ขั้นตอนการประมวลผลภาพ

เป็นขั้นตอนในการประมวลผลภาพ 3 มิติ โดยผู้ทดลองเป็นผู้นำ Script Python สำหรับทำการประมวลผลภาพ และข้อมูลเกมทั้งหมดไปใส่ไว้ที่เครื่องสำหรับประมวลผลผ่านทางระบบเครือข่ายเพื่อเตรียมการประมวลผล จากนั้นก็จะสั่งให้โปรแกรมดังกล่าวทำงานโดยแต่ละเครื่องที่รันงานอยู่นั้นก็จะรันเฉพาะในส่วนของตอนเองดังแสดงตามภาพที่ 15

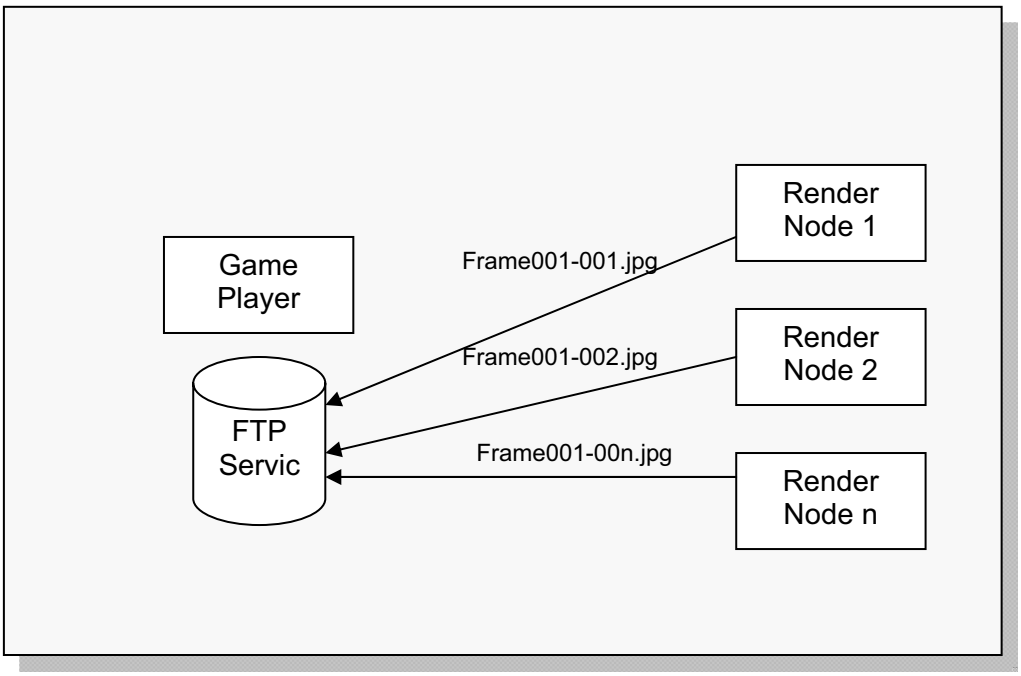


ภาพที่ 15 การแบ่งชิ้นส่วนสำหรับการประมวลผลในแต่ละ Render Node

จากภาพที่ 15 แสดงให้เห็นถึงการแบ่งชิ้นส่วนของงานออกเป็น 4 ส่วนโดยในแต่ละส่วนจะถูกทำการประมวลผลด้วยเครื่องคนละเครื่องและไม่ซ้ำกัน จากการกระทำดังกล่าวจะทำให้ลดระยะเวลาในการประมวลผลลงไปได้ ในกรณีที่ต้องการแบ่งมากขึ้นก็จำเป็นที่จะต้องใช้คอมพิวเตอร์เพื่อช่วยในการประมวลผลเพิ่มด้วย โดยผู้วิจัยได้ทดสอบแบ่งที่ 1 , 4 , 6 , 24 และ 108 เครื่องตามลำดับ

6.2. ขั้นตอนการส่งข้อมูล

เป็นขั้นตอนในการส่งผ่านข้อมูลที่ทำการประมวลผลแล้วไปยังเครื่องที่เล่นเกมต่อไปเพื่อนำภาพไปแสดง โดยในขั้นตอนการส่งข้อมูลนี้ผู้วิจัยได้ใช้คำสั่ง FTP ไปเก็บไว้ยังเครื่องที่เล่นเกมโดยตรงดังแสดงในภาพที่ 16

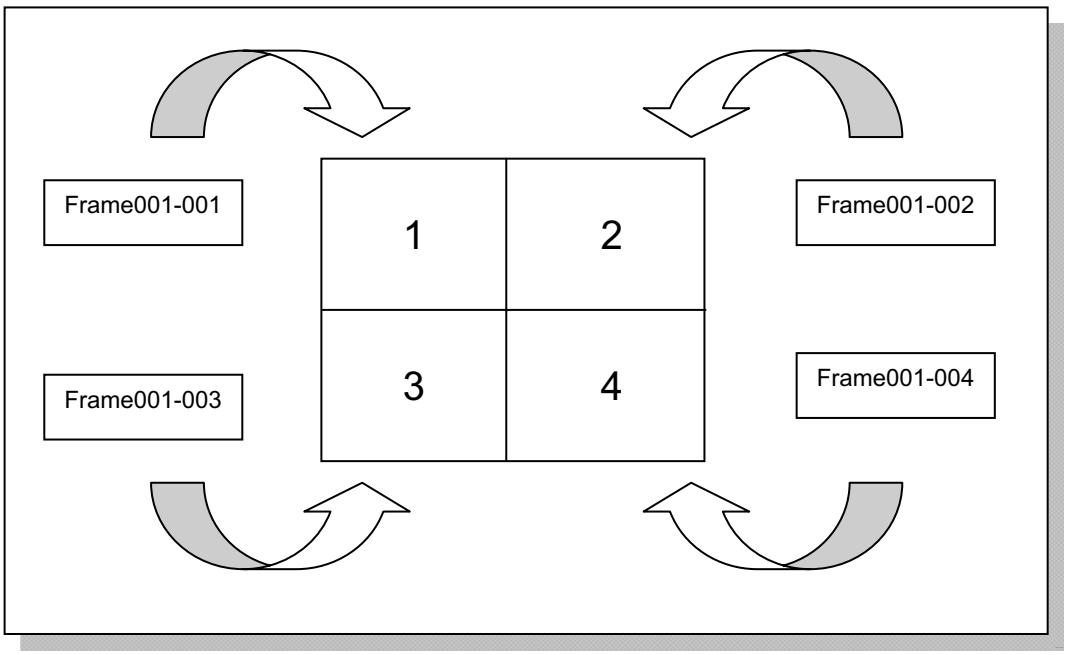


ภาพที่ 16 แสดงการส่งข้อมูลจากเครื่อง Render Node ไปยังเครื่อง Game Player โดยใช้วิธี FTP

จากภาพที่ 16 แสดงให้เห็นถึงวิธีการและขั้นตอนในการส่งผ่านข้อมูลที่ประมวลผลภาพ 3 มิติเสร็จแล้วถูกส่งไปยังเครื่องที่กำลังเล่นเกมอยู่ด้วยการ FTP แฟ้มข้อมูลเข้าไปโดยตรงและข้อมูลที่เสร็จแล้วจะถูกส่งไปโดยเพื่อไม่ให้เกิดการสับสนจึงได้มีการตั้งชื่อแฟ้มข้อมูลด้วย เช่น frame001-001.jpg หมายถึง เฟรมการแสดงผลที่ 1 จากเครื่องหมายเลข 1 เพื่อให้โปรแกรมที่ใช้ในการแสดงผลนำไปรวมได้อย่างถูกต้องและแสดงผลภาพไม่ผิดตำแหน่งอีกด้วย

6.3. ขั้นตอนในการแสดงผลข้อมูล

ในขั้นตอนนี้จะเป็นขั้นตอนการนำชิ้นส่วนภาพที่ถูกส่งมาจากเครื่องอื่น ๆ มาแสดงผลยังเครื่องที่เล่นเกมโดยต้องผ่านการประกอบชิ้นส่วนให้ตรงตำแหน่งก่อนแล้วจึงแสดงออกมาในลักษณะของรูปภาพที่เคลื่อนที่ได้ต่อไป ซึ่งมีรายละเอียดดังภาพที่ 17



ภาพที่ 17 แสดงการประกอบชิ้นส่วนของรูปภาพให้ตรงตำแหน่งเพื่อใช้แสดงผล

จากภาพที่ 17 แสดงให้เห็นถึงการประกอบรูปภาพลงในส่วนต่าง ๆ โดยอาศัยชื่อของแฟ้มข้อมูลเพื่อนำมาบอกตำแหน่งในการจัดวางชิ้นส่วนของรูปภาพให้ตรงกัน และในขั้นตอนนี้เองจะได้แฟ้มรูปภาพมาอีก 1 แฟ้มคือ Frame00x.jpg เป็นแฟ้มต้นฉบับในการแสดงผลนั่นเอง เมื่อแสดงผลเสร็จแล้วก็จะนำเอาชิ้นส่วนของเฟรมถัดไปมาแสดงต่อเรื่อย ๆ จนกว่าจะสิ้นสุดการทำงานของโปรแกรม

บทที่ 4
ผลการดำเนินการวิจัย

จากการศึกษาค้นคว้าวิจัยเกี่ยวกับการประยุกต์ใช้เทคโนโลยีกริดเพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ ซึ่งได้มีการออกแบบเกมด้วยโปรแกรม Blender 2.48 และใช้การส่งข้อมูลไปประมวลผลแบบขนานโดยใช้ Python 2.5 บนสภาวะแวดล้อม Linux Ubuntu 8 จากการทดลองดังกล่าวมีขั้นตอนในการทดลองสรุปได้ 3 ขั้นดังนี้

1. การประมวลผลภาพ (Render)
2. การส่งผ่านข้อมูล (Ftp file)
3. การรวมข้อมูลและแสดงผล (Pack & Show)

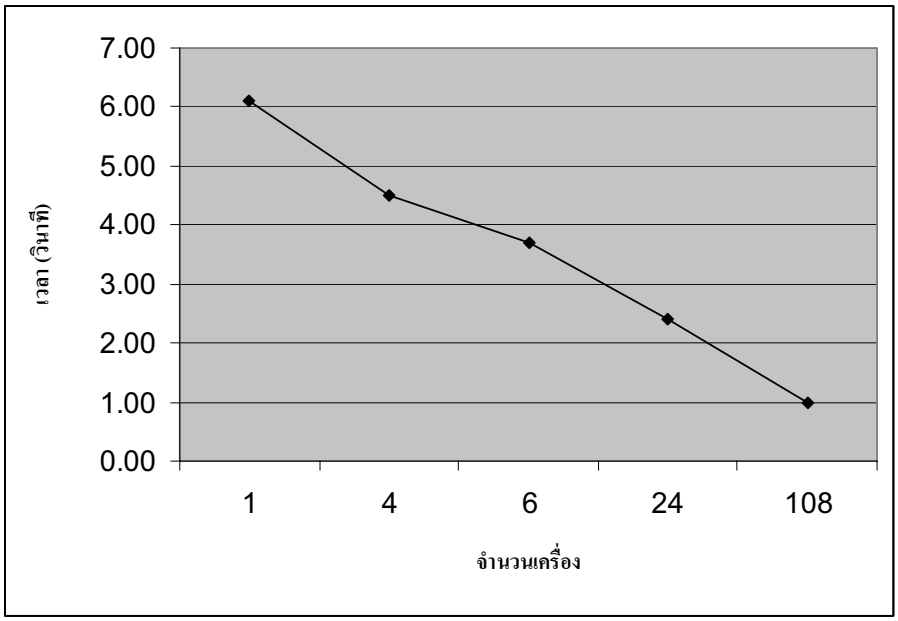
1. การประมวลผลภาพ (Render)

ผู้วิจัยได้ทำการทดลองการประมวลผลภาพ 3 มิติโดยใช้ภาพเคลื่อนไหวจำนวน 100 ภาพ ทำการทดลองกับเครื่องคอมพิวเตอร์จำนวน 1 , 4 , 6 , 24 และ 108 เครื่อง เป็นจำนวน 10 ครั้ง ดังตารางที่ 1

ตารางที่ 1 สรุปผลการทดลองขั้นการประมวลผลภาพ 3 มิติ

ครั้งที่	จำนวนเครื่อง									
	1		**4**		**6**		**24**		**108**	
1	6.12	วินาที	4.35	วินาที	3.60	วินาที	2.50	วินาที	1.00	วินาที
2	6.08	วินาที	4.40	วินาที	3.70	วินาที	2.45	วินาที	1.05	วินาที
3	6.00	วินาที	4.70	วินาที	3.60	วินาที	2.40	วินาที	1.10	วินาที
4	6.10	วินาที	4.50	วินาที	3.75	วินาที	2.30	วินาที	0.95	วินาที
5	6.10	วินาที	4.50	วินาที	3.70	วินาที	2.30	วินาที	1.00	วินาที
6	6.14	วินาที	4.55	วินาที	3.70	วินาที	2.35	วินาที	0.90	วินาที
7	6.16	วินาที	4.50	วินาที	3.70	วินาที	2.35	วินาที	1.00	วินาที
8	6.14	วินาที	4.50	วินาที	3.75	วินาที	2.40	วินาที	1.00	วินาที
9	6.04	วินาที	4.57	วินาที	3.75	วินาที	2.50	วินาที	1.00	วินาที
10	6.15	วินาที	4.43	วินาที	3.70	วินาที	2.45	วินาที	1.00	วินาที
รวมเฉลี่ย	**6.10**	วินาที	**4.50**	วินาที	**3.70**	วินาที	**2.40**	วินาที	**1.00**	วินาที

จากตารางที่ 1 สรุปผลการทดลองขั้นการประมวลผล 3 มิติ ซึ่งจากการทดลองนำภาพเคลื่อนไหวจำนวน 100 ภาพ มาทำการทดลองแบ่งไปประมวลผลกับเครื่องคอมพิวเตอร์จำนวน 1 , 4 , 6 , 24 และ 108 เครื่อง เป็นจำนวน 10 ครั้ง ผลที่ได้ในการประมวลผลแต่ละครั้งใกล้เคียงกัน โดย ผู้วิจัยได้นำข้อมูลดังกล่าวมาสร้างเป็นแผนภูมิเพื่อให้เข้าใจง่ายขึ้นดังแสดงในแผนภูมิที่ 1



แผนภูมิที่ 1 สรุปผลการทดลองขั้นการประมวลผลภาพ 3 มิติ

จากแผนภูมิที่ 1 สรุปผลการทดลองขั้นการประมวลผลภาพ 3 มิติ ผลการทดลองเป็นดังนี้ ขนาดภาพ 1024 pixel แบ่ง 1 เครื่องได้ความเร็วในการประมวลผลเฉลี่ยที่ 6.10 วินาที/ภาพ ขนาดภาพ 512 pixel แบ่ง 4 เครื่องได้ความเร็วในการประมวลผลเฉลี่ยที่ 4.50 วินาที/ภาพ ขนาดภาพ 256 pixel แบ่ง 6 เครื่องได้ความเร็วในการประมวลผลเฉลี่ยที่ 3.70 วินาที/ภาพ ขนาดภาพ 128 pixel แบ่ง 24 เครื่องได้ความเร็วในการประมวลผลเฉลี่ยที่ 2.4 วินาที/ภาพ และขนาดภาพ 64 pixel แบ่ง 108 เครื่องได้ความเร็วในการประมวลผลเฉลี่ยที่ 1.00 วินาที/ภาพ

สรุปผลการทดลองขั้นที่ 1 การประมวลผลภาพ 3 มิติ ที่ได้จากการส่งรูปภาพไปประมวลผลภาพแบบ 3 มิติในเครื่องอื่นผลที่ได้ในส่วนนี้คือ เมื่อแบ่งรูปภาพเป็นขนาดเล็กก็จะยิ่งทำให้ได้ความเร็วในการประมวลผลสูงมากยิ่งขึ้นโดยที่การตัดภาพที่ขนาด 64 pixel จะได้เวลาน้อยที่สุดคือ 1.00 นาที/ภาพนั่นเอง

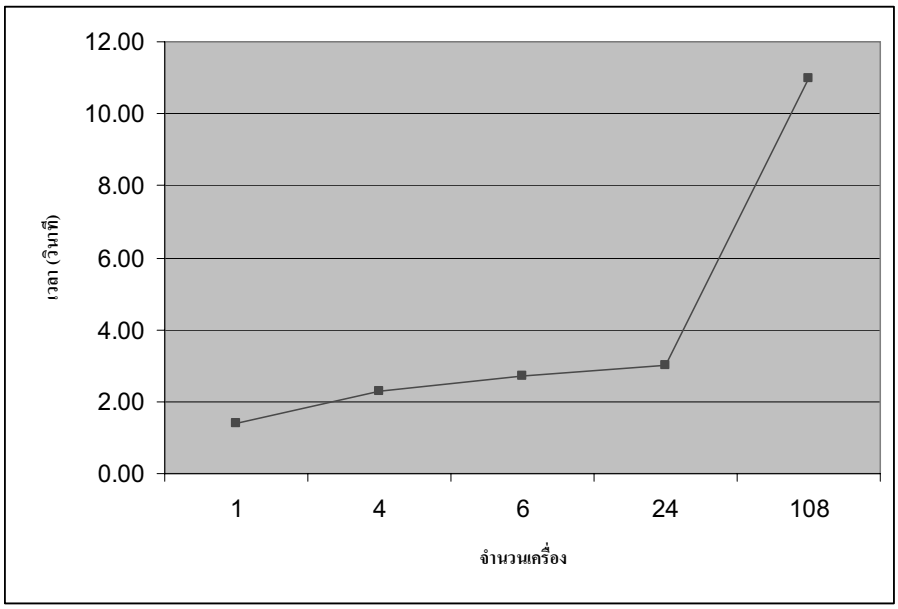
2. การส่งผ่านข้อมูล (Ftp file)

ผู้วิจัยได้ทำการทดลองการส่งผ่านข้อมูลให้เครื่องที่รอรับการแสดงผลภาพเคลื่อนไหวโดยได้ทำการทดลองนำภาพเคลื่อนไหวจำนวน 100 ทำการทดลองกับเครื่องคอมพิวเตอร์จำนวน 1 , 4 , 6 , 24 และ 108 เครื่อง เป็นจำนวน 10 ครั้ง แล้วหาค่าเฉลี่ยของเวลาที่น้อยที่สุดดังตารางที่ 2

ตารางที่ 2 สรุปผลการทดลองขั้นการส่งข้อมูลด้วย FTP

ครั้งที่	จำนวนเครื่อง				
	1	**4**	**6**	**24**	**108**
1	1.40 วินาที	2.30 วินาที	2.72 วินาที	3.00 วินาที	11.00 วินาที
2	1.43 วินาที	2.28 วินาที	2.74 วินาที	3.10 วินาที	11.05 วินาที
3	1.40 วินาที	2.28 วินาที	2.77 วินาที	3.00 วินาที	10.90 วินาที
4	1.38 วินาที	2.25 วินาที	2.70 วินาที	3.00 วินาที	10.95 วินาที
5	1.40 วินาที	2.30 วินาที	2.68 วินาที	2.95 วินาที	11.00 วินาที
6	1.39 วินาที	2.32 วินาที	2.65 วินาที	3.00 วินาที	10.90 วินาที
7	1.41 วินาที	2.30 วินาที	2.70 วินาที	3.00 วินาที	11.10 วินาที
8	1.40 วินาที	2.30 วินาที	2.60 วินาที	2.90 วินาที	11.00 วินาที
9	1.40 วินาที	2.32 วินาที	2.70 วินาที	3.05 วินาที	11.05 วินาที
10	1.42 วินาที	2.30 วินาที	2.70 วินาที	3.00 วินาที	11.00 วินาที
รวมเฉลี่ย	**1.40** วินาที	**2.30** วินาที	**2.70** วินาที	**3.00** วินาที	**11.00** วินาที

จากตารางที่ 2 สรุปผลการทดลองขั้นการส่งข้อมูลด้วย FTP ซึ่งจากการทดลองนำภาพเคลื่อนไหวจำนวน 100 ทำการทดลองกับเครื่องคอมพิวเตอร์จำนวน 1 , 4 , 6 , 24 และ 108 เครื่อง เป็นจำนวน 10 ครั้ง ผู้วิจัยได้นำข้อมูลดังกล่าวมาสร้างเป็นแผนภูมิเพื่อให้เข้าใจง่ายขึ้นดังแสดงในแผนภูมิที่ 2 ในหน้าถัดไป



แผนภูมิที่ 2 สรุปผลการทดลองขั้นการส่งข้อมูลด้วย FTP

จากแผนภูมิที่ 2 สรุปผลการทดลองขั้นการส่งข้อมูลด้วย FTP คือ การส่งรูปภาพที่ได้จากการประมวลผลภาพ 3 มิติเรียบร้อยแล้วไปยังเครื่องที่รอรับการแสดงผลภาพเคลื่อนไหวอยู่เพื่อแสดงผล จากการทดลองปรากฏว่า ขนาดภาพ 1024 pixel แบ่ง 1 เครื่องได้ความเร็วในการส่งข้อมูลเฉลี่ยที่ 1.40 วินาที/ภาพ ขนาดภาพ 512 pixel แบ่ง 4 เครื่องได้ความเร็วในการส่งข้อมูลเฉลี่ยที่ 2.30 วินาที/ภาพ ขนาดภาพ 256 pixel แบ่ง 6 เครื่องได้ความเร็วในการส่งข้อมูลเฉลี่ยที่ 2.70 วินาที/ภาพ ขนาดภาพ 128 pixel แบ่ง 24 เครื่องได้ความเร็วในการส่งข้อมูลเฉลี่ยที่ 3.00 วินาที/ภาพ และขนาดภาพ 64 pixel แบ่ง 108 เครื่องได้ความเร็วในการส่งข้อมูลเฉลี่ยที่ 11.00 วินาที/ภาพ

สรุปผลการทดลองขั้นที่ 2 การส่งผ่านข้อมูล(Ftp files) คือ การส่งรูปภาพที่ได้จากการประมวลผลภาพ 3 มิติเรียบร้อยแล้วไปยังเครื่องที่รอรับการแสดงผลภาพเคลื่อนไหวอยู่เพื่อแสดงผล ผลที่ได้ในส่วนนี้คือ เมื่อแบ่งรูปภาพเป็นขนาดเล็กก็จะยิ่งทำให้การส่งข้อมูลช้าลงไปด้วยเนื่องจากต้องมีการรอคอยจังหวะในการส่งข้อมูลคือต้องรอให้เครื่องที่เชื่อมต่อข้อมูลคืนทรัพยากรให้ระบบก่อน เครื่องอื่นที่รออยู่จึงจะสามารถเข้าไปใช้งานได้ทำให้เกิดความล่าช้า โดยขนาดภาพ 1024 pixel แบ่ง 1 เครื่องได้ความเร็วในการส่งข้อมูลเฉลี่ยที่ 1.40 วินาที/ภาพ เป็นเวลาที่น้อยที่สุด

3. การรวมข้อมูลและแสดงผล

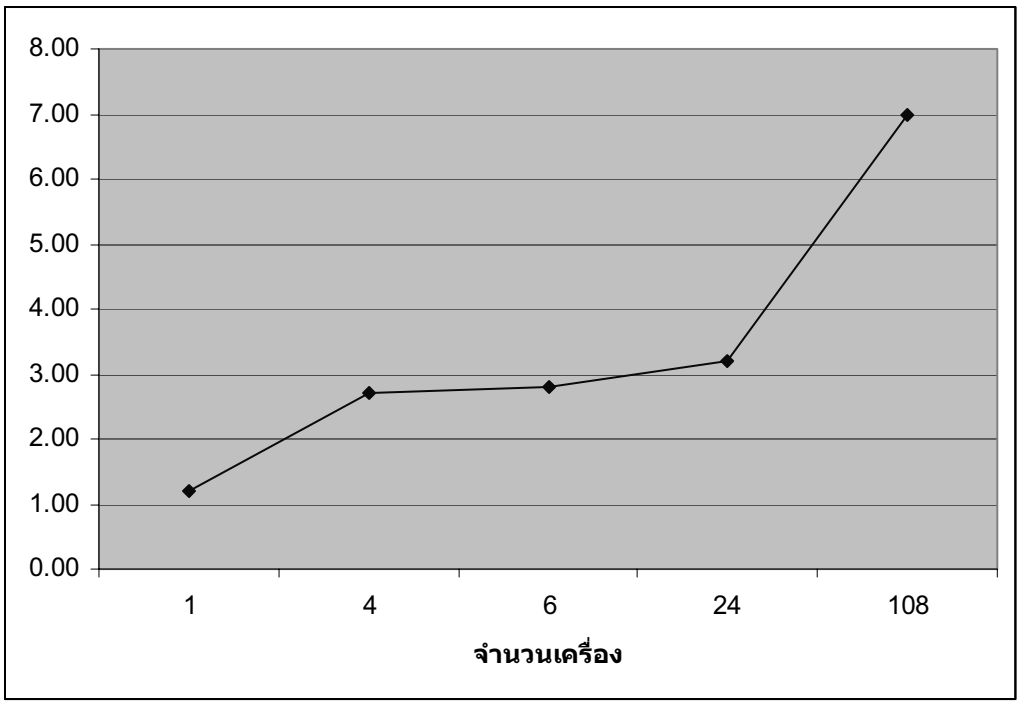
ผู้วิจัยได้ทำการทดลองการรวมข้อมูลและแสดงผล ซึ่งเป็นขั้นตอนในการนำรูปภาพมาต่อ

กันแล้วแสดงผลออกมาทางจอภาพต่อ ๆ กันไปเป็นภาพเคลื่อนไหว โดยได้ทำการทดลองกับเครื่องคอมพิวเตอร์จำนวน 1 , 4 , 6 , 24 และ 108 เครื่อง เป็นจำนวน 10 ครั้ง แล้วหาค่าเฉลี่ยของเวลาที่น้อยที่สุดดังตารางที่ 3

ตารางที่ 3 สรุปผลการทดลองขั้นการรวมข้อมูลและการแสดงผล

ครั้งที่	**จำนวนเครื่อง**				
	1	**4**	**6**	**24**	**108**
1	1.20 วินาที	2.75 วินาที	2.80 วินาที	3.20 วินาที	7.00 วินาที
2	1.20 วินาที	2.70 วินาที	2.85 วินาที	3.30 วินาที	7.10 วินาที
3	1.25 วินาที	2.65 วินาที	2.75 วินาที	3.25 วินาที	6.90 วินาที
4	1.15 วินาที	2.65 วินาที	2.80 วินาที	3.20 วินาที	7.00 วินาที
5	1.20 วินาที	2.70 วินาที	2.80 วินาที	3.15 วินาที	7.00 วินาที
6	1.15 วินาที	2.75 วินาที	2.85 วินาที	3.15 วินาที	6.90 วินาที
7	1.30 วินาที	2.70 วินาที	2.85 วินาที	3.20 วินาที	6.95 วินาที
8	1.15 วินาที	2.70 วินาที	2.75 วินาที	3.10 วินาที	7.10 วินาที
9	1.15 วินาที	2.65 วินาที	2.70 วินาที	3.25 วินาที	7.05 วินาที
10	1.20 วินาที	2.75 วินาที	2.80 วินาที	3.20 วินาที	7.00 วินาที
รวมเฉลี่ย	**1.20** วินาที	**2.70** วินาที	**2.80** วินาที	**3.20** วินาที	**7.00** วินาที

จากตารางที่ 3 สรุปผลการทดลองขั้นการรวมข้อมูลและการแสดงผล ซึ่งจากการทดลองทั้งสิ้นจำนวน 10 ครั้ง ผู้วิจัยได้นำข้อมูลดังกล่าวมาสร้างเป็นแผนภูมิเพื่อให้เข้าใจง่ายขึ้นดังแสดงในแผนภูมิที่ 3 ในหน้าถัดไป



แผนภูมิที่ 3 สรุปผลการทดลองขั้นการรวมข้อมูลและการแสดงผล

จากแผนภูมิที่ 3 สรุปผลการทดลองขั้นการรวมข้อมูลและการแสดงผล คือ การที่เครื่องที่เล่นเกมนำไฟล์รูปภาพที่ได้รับมาจากเครื่องที่ส่งไปประมวลผลมาทำการแสดงให้ผู้เล่นเกมได้รับรู้ ผลการทดลองปรากฏว่า ขนาดภาพ 1024 pixel แบ่ง 1 เครื่องได้ความเร็วในการแสดงผลเฉลี่ยที่ 1.20 วินาที/ภาพ ขนาดภาพ 512 pixel แบ่ง 4 เครื่องได้ความเร็วในการแสดงผลเฉลี่ยที่ 2.70 วินาที/ภาพ ขนาดภาพ 256 pixel แบ่ง 6 เครื่องได้ความเร็วในการแสดงผลเฉลี่ยที่ 2.80 วินาที/ภาพ ขนาดภาพ 128 pixel แบ่ง 24 เครื่องได้ความเร็วในการแสดงผลเฉลี่ยที่ 3.20 วินาที/ภาพ และขนาดภาพ 64 pixel แบ่ง 108 เครื่องได้ความเร็วในการแสดงผลที่ 7.00 วินาที/ภาพ

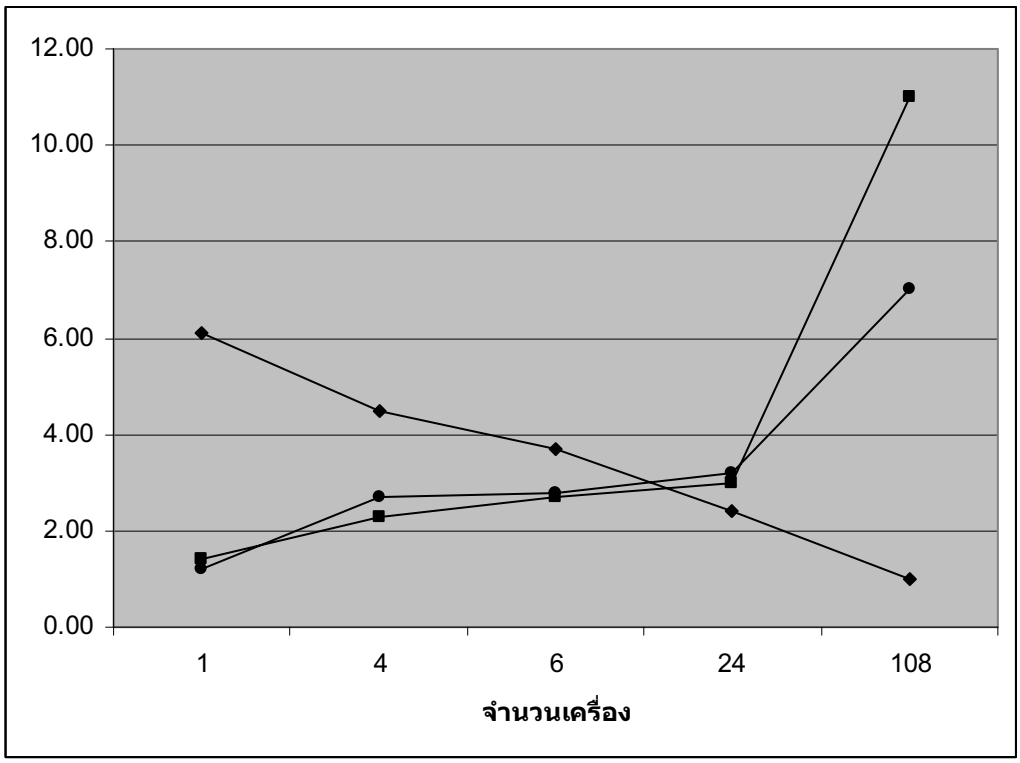
สรุปผลการทดลองขั้นที่ 3 การรวมข้อมูลและแสดงผล คือ การที่เครื่องที่เล่นเกมนำไฟล์รูปภาพที่ได้รับมาจากเครื่องที่ส่งไปประมวลผลมาทำการแสดงให้ผู้เล่นเกมได้รับรู้ ผลที่ได้ในส่วนนี้คือ การแสดงผลภาพภาพที่ไม่ได้ทำการตัดแบ่งจะสามารถแสดงผลได้เลยโดยไม่ต้องทำการรวมไฟล์เพื่อแสดงผลและให้ผลลัพท์ที่เร็วกว่า ที่ขนาดภาพ 1024 pixel แบ่ง 1 เครื่องได้ความเร็วในการแสดงผลเฉลี่ยที่ 1.20 วินาที/ภาพ ซึ่งเป็นเวลาที่เร็วที่สุด

จากการทดลองดังกล่าวทั้ง 3 ขั้นตอนผู้วิจัยได้ทำการสรุปรวมเวลาโดยเฉลี่ยของการทำงานทั้ง 3 ขั้นตอนเป็นตารางเพื่อการวิเคราะห์หาผลลัพธ์ดังแสดงข้อมูลตามตารางที่ 4

ตารางที่ 4 ผลการทดลองการประยุกต์ใช้เทคโนโลยีกริดเพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ

ขั้นตอน	1 เครื่อง	4 เครื่อง	6 เครื่อง	24 เครื่อง	108 เครื่อง
ประมวลผล	6.10 วินาที	4.50 วินาที	3.70 วินาที	2.40 วินาที	1.00 วินาที
ส่งข้อมูล	1.40 วินาที	2.30 วินาที	2.70 วินาที	3.00 วินาที	11.00 วินาที
แสดงข้อมูล	1.20 วินาที	2.70 วินาที	2.80 วินาที	3.20 วินาที	7.00 วินาที
Total	8.70 วินาที	9.50 วินาที	9.20 วินาที	8.60 วินาที	19.00 วินาที

จากตารางที่ 4 ผลการทดลองการประยุกต์ใช้เทคโนโลยีกริดเพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ ซึ่งจากการทดลองทั้งสิ้นจำนวน 10 ครั้งด้วยกันผลที่ได้ในการประมวลผลแต่ละครั้งใกล้เคียงกันอาจมีบางครั้งที่ค่าออกมามากหรือน้อยกว่าปกติแต่ก็ยังสามารถนำมาสรุปได้ ผู้วิจัยได้นำข้อมูลดังกล่าวมาสร้างเป็นแผนภูมิเพื่อให้เข้าใจง่ายขึ้นดังแสดงตามแผนภูมิที่ 4 ในหน้าถัดไป



แผนภูมิที่ 4 ผลการทดลองการประยุกต์ใช้เทคโนโลยีกริดเพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ

จากแผนภูมิที่ 4 ผลการทดลองการประยุกต์ใช้เทคโนโลยีกริดเพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ พบว่าเมื่อมีการนำผลเฉลี่ยในการประมวลผลทั้ง 3 ขั้นตอนมารวมกันแล้ว จะพบว่ามีจุดที่ตัดกันของแผนภูมิอยู่ที่ตำแหน่ง 24 เครื่อง เมื่อดูจากตารางที่ 5 ผลการทดลองการประยุกต์ใช้เทคโนโลยีกริดเพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ ในช่องผลรวมจะพบว่าใช้เวลาโดยรวมทั้ง 3 ขั้นตอนน้อยที่สุด โดยใช้เวลารวมทั้งหมดอยู่ที่ 8.60 วินาที/ภาพ

บทที่ 5
สรุปผลการวิจัย

จากการประยุกต์ใช้เทคโนโลยีกริดเพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ ทำให้ได้โปรแกรมสำหรับรับ-ส่ง ข้อมูลที่เป็นการประมวลผลภาพเคลื่อนไหว 3 มิติ แบบขนานโดยเพิ่มรูปแบบการใช้งานในโปรแกรม Blender ในส่วนของ 3D Game Engine ให้มีการทำงานแบบขนานโดยการเรียกใช้งาน Python Script ที่นำรูปภาพไปแบ่งส่วนแล้วให้เครื่องคอมพิวเตอร์เครื่องอื่นในระบบเครือข่ายร่วมกันประมวล ทำให้สามารถประมวลผลภาพเคลื่อนไหว 3 มิติแบบขนานได้ โดยสรุปได้ดังนี้

การบรรลุวัตถุประสงค์การวิจัย

ออกแบบและพัฒนาโปรแกรมเพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ โดยประยุกต์ใช้เทคโนโลยีกริด เมื่อการพัฒนาระบบเสร็จสิ้น และได้ทดสอบการทำงานของระบบ ทำให้ระบบงานนี้บรรลุวัตถุประสงค์ตามที่ตั้งไว้คือ

มีโปรแกรมสำหรับทำการประมวลผลภาพภาพเคลื่อนไหวในเกม 3 มิติ ซึ่งมีการทำงานใน 3 ขั้นตอน ดังนี้

1) การประมวลผลภาพภาพ 3 มิติโดยใช้ภาพเคลื่อนไหวจำนวน 100 ภาพ ทำการทดลองกับเครื่องคอมพิวเตอร์จำนวน 1 , 4 , 6 , 24 และ 108 เครื่อง เป็นจำนวน 10 ครั้ง ผลที่ได้ในขั้นตอนนี้คือเมื่อแบ่งรูปภาพเป็นขนาดเล็กก็จะยิ่งทำให้ได้ความเร็วในการประมวลผลสูงมากยิ่งขึ้นโดยที่การตัดภาพที่ขนาด 64 pixel ได้เวลาน้อยที่สุดคือ 1.00 นาที/ภาพ

2) การส่งผ่านข้อมูลไฟล์ภาพ โดยใช้ภาพเคลื่อนไหวจำนวน 100 ภาพทำการทดลองกับเครื่องคอมพิวเตอร์จำนวน 1 , 4 , 6 , 24 และ 108 เครื่อง เป็นจำนวน 10 ครั้ง ผลที่ได้ในขั้นตอนนี้ คือ ขนาดภาพ 1024 pixel แบ่ง 1 เครื่องได้ความเร็วน้อยที่สุดในการส่งข้อมูลเฉลี่ยคือ 1.40 วินาที/ภาพ

3) การรวมภาพเพื่อแสดงผล โดยใช้ภาพเคลื่อนไหวจำนวน 100 ภาพทำการทดลองกับเครื่องคอมพิวเตอร์จำนวน 1 , 4 , 6 , 24 และ 108 เครื่อง เป็นจำนวน 10 ครั้ง ผลที่ได้ในขั้นตอนนี้ คือ ขนาดภาพ 1024 pixel แบ่ง 1 เครื่องได้ความเร็วน้อยที่สุดในการส่งข้อมูลเฉลี่ยคือ 1.20 วินาที/ภาพ

ซึ่งจากผลการทดลองการประยุกต์ใช้เทคโนโลยีกริดเพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ เมื่อมีการนำผลเฉลี่ยในการประมวลผลทั้ง 3 ขั้นตอนมารวมกันแล้ว จะพบว่าจำนวนเครื่องที่ใช้เวลารวมเฉลี่ยในการประมวลผลน้อยที่สุด คือ 24 เครื่อง โดยใช้เวลาเฉลี่ยรวมทั้งหมดอยู่ที่ 8.60 วินาที/ภาพ

ปัญหาและอุปสรรค

การพัฒนาระบบเกิดปัญหาและอุปสรรคดังนี้

- ไม่สามารถทำงานบนระบบปฏิบัติการ WindowXP ได้ใช้ได้เฉพาะ Linux Ubuntu เท่านั้น

- เนื่องจากมีการแบ่งชิ้นงานมากจึงทำให้สูญเสียเวลาในการรวมข้อมูลภาพ 3 มิติ ทำให้เกิดความล่าช้า และยังต้องเกิดการรอคอยในระบบเครือข่ายเนื่องจากเครือข่ายมีความเร็วในการรับ-ส่งข้อมูลน้อยกว่า ความเร็วของหน่วยประมวลผลกลางของเครื่องคอมพิวเตอร์

ข้อเสนอแนะ

ข้อเสนอแนะสำหรับการวิจัย

ประยุกต์ใช้เทคโนโลยีกริดเพื่อสร้างภาพเคลื่อนไหวในเกม 3 มิติ ทำให้ได้โปรแกรมสำหรับทำการรับ-ส่งค่าข้อมูลที่เป็นการประมวลผล 3 มิติ แบบขนาน โดยขอเสนอแนะไว้ดังนี้

1. ในส่วนของการรับ-ส่งข้อมูลหากระบบเครือข่ายมีความเร็วมากยิ่งขึ้นจะทำให้การรับ-ส่งข้อมูลในระบบมีความเร็วเพิ่มขึ้นอีก

2. การแบ่งจำนวนภาพยิ่งมากจะทำให้ลดเวลาในการประมวลผลได้แต่อาจทำให้เกิดปัญหาในการรวมรวมข้อมูลซึ่งใช้เวลามาก และจากผลการทดสอบที่เครื่องจำนวน 24 เครื่อง จะได้เวลาออกมาดีที่สุด ทั้งนี้ก็ขึ้นอยู่กับปริมาณงานที่จะทำการประมวลผลด้วย

ข้อเสนอแนะในการวิจัยครั้งต่อไป

ในการศึกษาวิจัยครั้งต่อไปสิ่งที่ควรทำเพิ่มเติมมีดังนี้คือ

ควรทำการประมวลผลภาพ 3 มิติ ในลักษณะข้ามแพลตฟอร์ม เช่น Window Linux MacOS เพื่อเปรียบเทียบการทำงานของการประมวลผลภาพ 3 มิติแบบขนานในระบบปฏิบัติการที่แตกต่างกัน

บรรณานุกรม

ภาษาไทย

เปมปีย์อร วังไชยเลิศ. “การออกแบบระบบจัดเก็บและสืบค้นข้อมูลแบบกระจายโดยใช้กริดเทคโนโลยี.” วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์, 2549.

พีรพล เวทีกูล . “ต้นแบบระบบสืบค้นคืนข้อมูลแบบกระจายบนเครือข่ายการเชื่อมต่อกริด.” วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์, 2547.

ไพศาล สิมาเลาเต่า. “การพัฒนาโปรแกรมประยุกต์การรู้จำ โดยใช้การประมวลผลแบบกริด.” วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร, 2550.

ศรชย โรจนวิภาต. “การพัฒนาระบบบริหารทรัพยากรกริดโดยใช้กริดโบรกเกอร์.” วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ บัณฑิตวิทยาลัยมหาวิทยาลัยเกษตรศาสตร์, 2549.

ฮิกส์แมน. Grid Computing : เทคโนโลยี IT ของโลกอนาคต [ออนไลน์]. เข้าถึงเมื่อ 26 พฤศจิกายน 2550. เข้าถึงได้จาก http://www.vcharkarn.com/vnews/?slType=Cid&&Cafe=29)

ภาษาต่างประเทศ

Bruce Mechtly, Eric Rooker and Konrad Mast , 3D RENDERING WITH C++ AND OpenGL IN UNDERGRADUATE PROJECTS , 2001.

Dave Astle and Dave Durnil . OpenGL ES Game Development . United States of America : Thomson Course Technology , 2004.

Foster I, Kesselman C, Tuecke S. 2001. The anatomy of the Grid: Enabling scalable virtualorganizations. International Journal of Supercomputer Applications.

Foster, I. 2002. What is the Grid? A Three Point Checklist. GRIDToday 1 (6).

Foster, I., C. Kesselman, J. Nick and S. Tuecke. 2002. The Physiology of the Grid: An OpenGrid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG (Global Grid Forum).

Foster, I., C. Kesselman, M.J. Nick and S. Tuecke. 2002. Grid Services for Distributed System Integration. In Proceedings of Institute of Electrical and Electronics Engineers. 35 (6):37-46.

Harry J. Evry , Beginning Game Graphic . United States of America : Thomson Course Technology , 2005.

Ideasman . working still image renderfarm script[Online]. Access 18 December 2008 Available From http://www.blender.org/forum/viewtopic.php?t=3717 .

Kodeboyina, D. and B. Plale. 2003. Experiences with OGSA-DAI: Portlet Access and Benchmark. In Proceedings of the 9th GGF Designing and Building Grid Services Workshop.

Laszewski, G. and K. Jackson. 2002. CoG Kits: Enabling Middleware for Designing Science Applications, Web Portals, and Problem Solving Environments. Available Source: http://www.cogkits.org, Feburay 3, 2008.

Pittaya . Python Image Library [Online]. Access 14 January 2009 Available From http://www.blognone.com/node/2804 .

Sergei Savchenko . 3D Graphics Programming Game and Beyond . United States of America : Sams Publishing , 2000.

Wasara Sutaweesup and Yuen Poovarawan . Parallel Motion Path Calculation for Animated Objects inDistributed Environment . Department of Computer Engineering, Kasetsart University, Arpil 2002.

ภาคผนวก

ภาคผนวก ก
การติดตั้งและเรียกใช้งาน Blender v2.48

การติดตั้ง Blender v2.48 บน Linux Ubuntu 8.0.10 มีขั้นตอนดังนี้

1. เปิด Console สำหรับพิมพ์คำสั่ง
2. ใช้คำสั่งในการเตรียมสภาพแวดล้อมดังนี้

```
sudo apt-get install subversion openexr libopenexr-dev build-essential libjpeg-dev \
libpng12-dev libopenal-dev libalut-dev libsdl-dev libfreetype6-dev libtiff-dev python-dev \
gettext libxi-dev
```

3. ใช้คำสั่งในการ Download และ Compile ดังนี้

```
cd ~ ; \
mkdir blender-svn ; \
cd blender-svn ; \
svn checkout https://svn.blender.org/svnroot/bf-blender/trunk/blender blender; \
cd blender; \
echo "BF_OPENAL_LIB = 'openal alut'" > user-config.py; \
echo BF_PYTHON_VERSION=\"$(python -V 2>&1 | cut -d" " -f 2 | cut -d"." -f 1-2)\" >>
user-config.py; \
python scons/scons.py;
```

4. เมื่อ Compile เสร็จแล้วจะขึ้นข้อความดังนี้

```
Install file: "/home/myname/blender-svn/build/linux2/bin/blender" as
"/home/myname/blender-svn/install/linux2/blender"
scons: done building targets.
*** Success ***
```

5. จากนั้นให้เรียกใช้โปรแกรมโดยพิมพ์คำสั่งดังนี้

```
cd ~/blender-svn/blender
ln -s ../install/linux2/blender ./blender
```

6. หากว่าอยู่ภายใต้ sub directories /blender-svn/blender อยู่แล้วก็สามารถเรียกใช้โปรแกรมได้เลยโดยพิมพ์คำสั่งว่า ./blender
7. หากต้องการ Update โปรแกรมให้ใหม่ขึ้นจากเวปไซต์ให้พิมพ์คำสั่งว่า svn update จากนั้นให้ Compile ใหม่โดยพิมพ์คำสั่งว่า python scons/scons.py เครื่องก็จะทำการ Compile โปรแกรมให้ใหม่โดยอัติโนมัติ

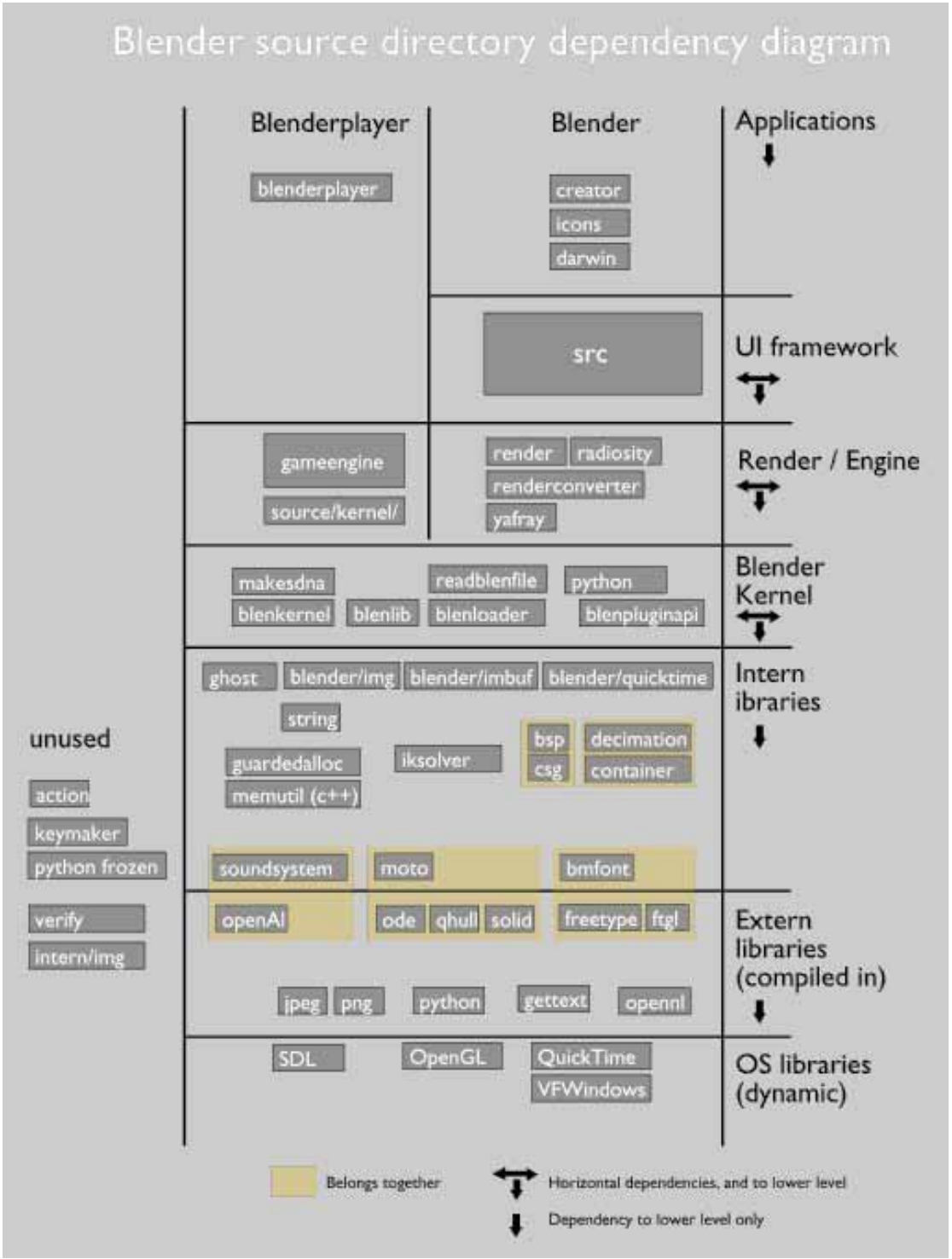
รูปที่ 18 โปรแกรม Blender v2.48

ภาคผนวก ข

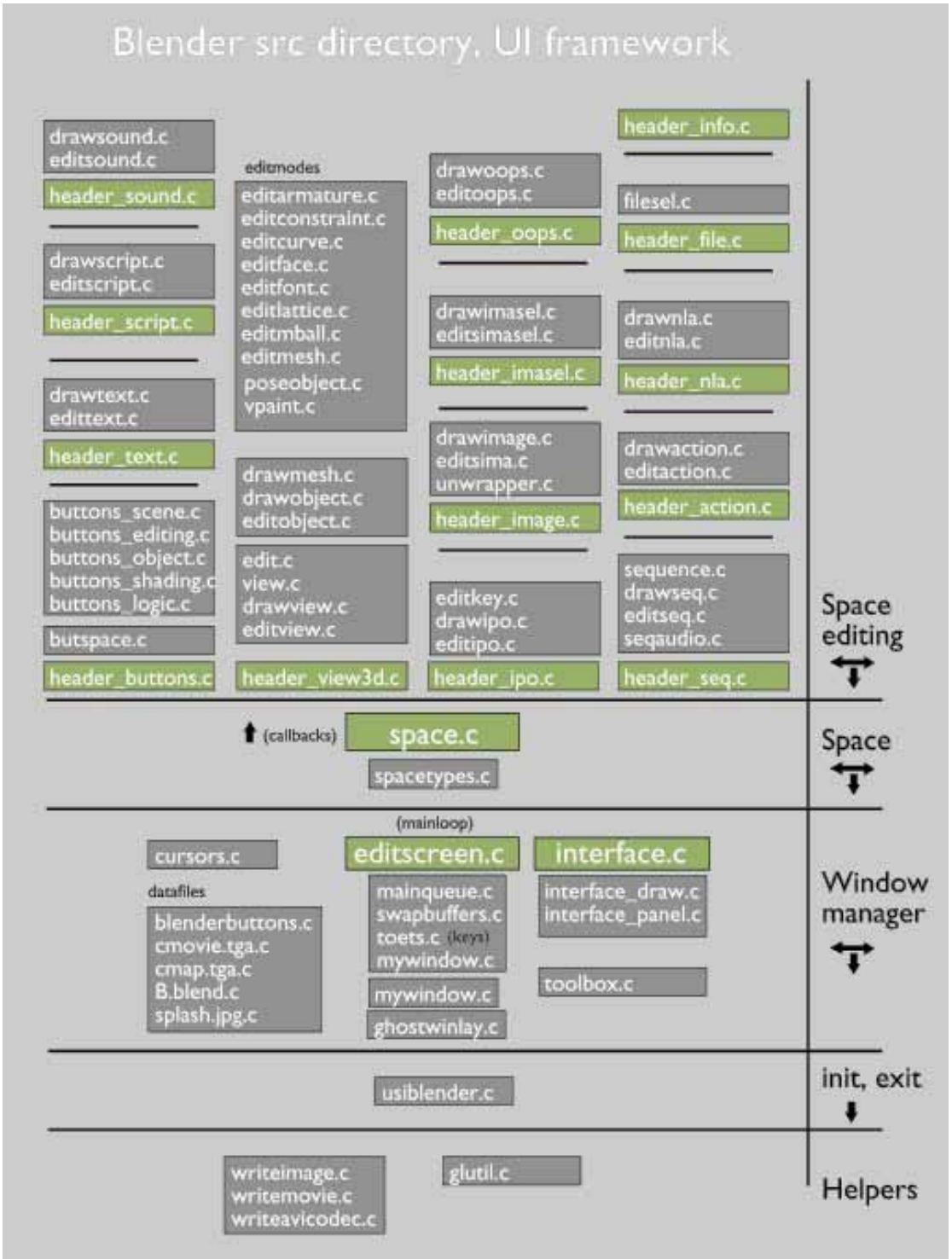
โครงสร้างภายในของ Blender v2.48 และตำแหน่งที่เก็บ Source Code

โครงสร้างภายในของ Blender v2.48

โครงสร้างภายในโปรแกรมแบ่งตามชั้นการทำงาน



รูปที่ 19 แผนภูมิแสดงโครงสร้างของ Blender v2.48



รูปที่ 20 ตำแหน่งที่เก็บ Source Code ที่จำเป็นต่อการเรียกใช้งาน

ภาคผนวก ค

Blender v2.48 API with Python Script

ต่อไปนี้เป็น Source Code ที่เขียนและทำงานบน Python 2.5 ซึ่งทำการเรียกใช้งาน API ของ Blender v2.48 สำหรับการประมวลผลภาพ 3 มิติ ซึ่งทำการแบ่งชิ้นส่วนการประมวลผลในแต่ละเครื่อง มีรายละเอียดดังนี้

```
import Blender
from Blender import sys, Scene, Noise
from Blender.Scene import Render
import string

NUM = '0123456789'
DEFAULT_TILE_SIZE=128 # px

# Clamp
def max(number):
  if number > 1.0:
    number = 1.0
  return number

def isfile(file):
  try:
    open(file)
    return 1
  except:
    return 0


def randList(list):
  randList = []
  lenList = len(list)
  while lenList != len(randList):
    randIndex = int( Noise.random() * len(list)  )
    randList.append( list[randIndex] )
    list.remove( list[randIndex] )

  return randList

# Strip all chars after a '.'
def stripExt(text):
  return text[:text.index('.')]

def getParts():

  name = Blender.Get('filename')
  name = stripExt(name)
```

```
  # Check that there is a number at the end of the filename.
  if name[-1] not in NUM:
    tileSize = DEFAULT_TILE_SIZE
  else:
    # Work out the tile size from the file name.
    tileSize = ''
    while name[-1] in NUM: # Keep pulling in the numbers as long as they are nums.
      tileSize = name[-1] + tileSize
      name = name[0:-1]

    tileSize = eval(tileSize)


  # OK either way we have a tile size.
  # Get the data
  scn = Scene.GetCurrent()
  context = scn.getRenderingContext()

  xPix = context.imageSizeX()
  yPix = context.imageSizeY()

  xParts = int(xPix / tileSize)
  yParts = int(yPix / tileSize)

  # Incase the tile size is larger then the render size.
  if xParts < 2:
    xParts = 2
  if yParts < 2:
    yParts = 2

  return xParts, yParts


# Makes a list of rects that setBorder will pass.
def makeBorderList(xparts, yparts):

  borderRectList=[] #We store all the rects here and then return them.

  xlen = 1.0 / xparts
  ylen = 1.0 / yparts
  xPoint = 0.0 # This stores the current X value, and incriments xlen each iteration
until its equel to 1.0
  yPoint = 0.0

  counter = 1 # Inde each border

  while xPoint < 0.999:
```

```
    while yPoint < 0.999:

      # Write the rect to the list
      borderRectList.append( (counter, max(xPoint), max(yPoint), max(xPoint+xlen),
max(yPoint+ylen)) )
      counter += 1 # Keep a tag of which index this one is.

      yPoint += ylen
    # Reset yPoint for the next colum.
    yPoint = 0.0
    xPoint += xlen
  return borderRectList


# SETS UP DEFAULTS NEEDED FOR OUTPUTTING AN IMAGE THAT CAN
BE COMPOSITED.
scn = Scene.GetCurrent()
context = scn.getRenderingContext()
context.enableBorderRender(1)
context.enableRGBAColor() # Save RGBA
context.setImageType(Render.PNG) # Save RGBA
context.enableExtensions(1)

# Make image name
imageName = Blender.Get('filename')

# Remove .blend
imageName = stripExt(imageName)
renderName = imageName + '_' # frameNum.png will be added.


xParts, yParts = getParts()

randBorderList = randList( makeBorderList(xParts,yParts) )


# Set the start and end frame to the current frame.
curFrame = Blender.Get( 'curframe')
context.startFrame(curFrame)
context.endFrame(curFrame)

# Keep track of frames rendered, only for a report.
renderedFrames = 0

for border in randBorderList:

  # Set the new file name WITH X/Y parts
```

```
  # blendfilename_partnum_framenum.ext
  # eg. render_01_0001.png
  partNum = str(border[0])
  while len(partNum) < 4:
    partNum = '0' + partNum
  uniqueRenderName = renderName + partNum + '_' # We add 1 so that the first
image is 1, not 0

  # CREATE THE REAL NAME OF THE OYTPUT FILE
  frameNum = str(curFrame)
  while len(frameNum) < 5:
    frameNum = '0' + frameNum
  fileToRender = uniqueRenderName + frameNum + '.png'


  # Chech that the file isnt alredy there
  if isfile(fileToRender) == 0:

    # TOUCH FILE SO NOBODY OVERWRITES IT.
    #Create a dummy file so no other nodes try to render the image.
    file = open(fileToRender,"w")
    file.close()


    # SET RENDER NAME AND PATH.
    context.setRenderPath('//' + uniqueRenderName) # // is the currentdir

    # Set border
    context.setBorder(border[1], border[2], border[3], border[4] )

    # RENDER THE IMAGE
    context.renderAnim() # This saves the pics.

    renderedFrames += 1


# Print report
print 'XParts', xParts,' YParts', yParts
print renderedFrames, ' of ', len(randBorderList), ' rendered'

# Quit
Blender.Quit()
```

เรียกใช้งานผ่าน Console ด้วยคำสั่งดังนี้คือ blender myscene_256.blend -P render_client.py

การเรียกใช้งาน FTP ใน Python 2.5 มีดังนี้คือ

```
import ftplib
#Ftp to Computer Master
        sftp = ftplib.FTP('192.168.1.xxx','username','password') # Connect
        fp = open('output/' + frameNum + '.jpg','rb') # file to send
        sftp.storbinary('STOR render_files/' + frameNum + '.jpg', fp) # Send the file
        fp.close() # Close file and FTP
        sftp.quit()
#end of Ftp
```

ภาคผนวก ง
การติดตั้ง vsftpd สำหรับ FTP

การติดตั้ง vsftpd สำหรับ FTP

$ su - root

Password:

Terminal type? [xterm]

cd /usr/ports

make search key=vsftp

Port: vsftpd-2.0.4

Path: net/vsftpd

Info: Ftp daemon which aims to be secure

Maint: Alexandre Anriot

Index: net

L-deps:

B-deps:

R-deps:

Archs: any

cd net/vsftpd

make install

แก้ไข vsftpd.conf บางส่วนดังนี้

แก้เอา # ออกหน้าบรรทัด

xferlog_file=/var/log/vsftpd.log

banned_email_file=/etc/vsftpd.banned_emails

#chroot_list_file=/etc/ftpchroot

#userlist_file=/etc/ftpusers

chroot_local_user=YES

secure_chroot_dir=/var/empty

สั่งให้ vsftp ทำงาน

/usr/local/sbin/vsftpd &

[1] 10214

ตรวจผล

```
# netstat -a | grep ftp
tcp 0 0 *.ftp *.* LISTEN
```

ทดลองเข้าไปจากอีกเครื่อง ปรากฏข้อความ

```
500 OOPS: vsftpd: cannot locate user specified in 'ftp_username':ftp
```

แก้ไข vsftpd.conf ให้ local usr ใช้งาน vsftpd ได้

```
# Standalone mode
listen=YES
anonymous_enable=NO
local_enable=YES
#local_umask=022
anon_upload_enable=NO
anon_mkdir_write_enable=NO
anon_other_write_enable=NO
anon_world_readable_only=YES
xferlog_enable=YES
hide_ids=YES
xferlog_file=/var/log/vsftpd.log
xferlog_std_format=YES
idle_session_timeout=120
data_connection_timeout=300
accept_timeout=60
connect_timeout=60
anon_max_rate=50000
nopriv_user=_vsftpd
ascii_download_enable=YES
#chroot_list_enable=YES
#chroot_list_file=/etc/ftpchroot
chroot_local_user=YES
secure_chroot_dir=/var/empty
pasv_min_port=49152
```

pasv_max_port=65535

text_userdb_names=YES

วิธีทำให้ vsftpd อนุญาต anonymous ทำดังนี้

เพิ่ม user ftp ดังนี้ # mkdir /home/ftp

/usr/sbin/useradd -d /home/ftp ftp

chown root.wheel /home/ftp

chmod og-w /home/ftp

แล้วแก้ /etc/passwd

ftp:*:1002:10:ftp user:/home/ftp:/sbin/nologin

แก้ /etc/vsftpd.conf

This FTP server is anonymous only.

อย่างอื่นเหมือนตัวอย่าง ข้างบน

local_enable=YES

anonymous_enable=YES

ftpd_banner=Welcome to blah FTP service.

nopriv_user=ftp

บันทึก แล้วสั่ง /usr/local/sbin/vsftpd &

มั่นใจว่าทุกอย่างทำงานแล้ว เอาคำสั่งนี้ใส่ใน

/etc/rc.local

สร้างโฟลเดอร์ /home/ftp/pub/ เอาไว้ใส่ไฟล์

ทดลองใช้งานจริง

$ ftp gate.sru.ac.th

220 Welcome to blah FTP service.

Name (gate.sru.ac.th:viriya): anonymous

331 Please specify the password.

Password:

230 Login successful.

Remote system type is UNIX.

Using binary mode to transfer files.

ftp> ls

229 Entering Extended Passive Mode (|||62613|)

150 Here comes the directory listing.

drwxr-xr-x 2 ftp ftp 512 Oct 01 19:34 pub

ftp> cd pub

250 Directory successfully changed.

อ้างอิง

VSFTP (Very Secure FTP) ศิริวรรณ อภิสิริเดช

http://www.thaicert.nectec.or.th/paper/unix_linux/vsftp.php

ไฟล์

/usr/local/share/doc/vsftpd/FAQ

/usr/local/share/doc/vsftpd/INSTALL

/usr/local/share/doc/vsftpd/README

/usr/local/share/examples/vsftpd/INTERNET_SITE/vsftpd.conf

ภาคผนวก จ

ส่วนของการรวมภาพและแสดงผลภาพ

ต่อไปนี้เป็น Source Code ที่เขียนและทำงานบน Python 2.5 ซึ่งทำการเรียกใช้งาน API ของ Blender v2.48 สำหรับรวมชิ้นส่วนและประมวลผลภาพ 3 มิติ ที่ได้รับมาจากเครื่องที่ประมวลผลแบบแยกชิ้นส่วน มีรายละเอียดดังนี้

```
import Image


frame = 4
xpart = 3
ypart = 2
part = xpart * ypart
x1 = 800/xpart
y1 = 600/ypart
x = 0
y = 0

newimage = Image.new("RGB", (800, 600), (255, 255, 255)) # mode, size, color

for j in range(1, frame +1):
        i = 0
        for y in xrange(0, 600, y1):
          for x in xrange(0, 800, x1):
                i = i + 1
                src_img =
Image.open("//home/blender/test//home/blender/test/myscene_000"+str(i)+"_000"+str
(j)+".png")
                piece = src_img.crop((x, y, x+x1, y+y1))
                newimage.paste(piece , (x, y))
                print "x = ",x," y = ",y

        print "Number of part = ",part
        newimage.save("//home/blender/test//home/myscene_000"+str(j)+".png")

        newimage.show()
```

ประวัติผู้วิจัย

ชื่อ – สกุล	นายธรรมรงค์ กำมะณี Mr.Thammarong Kummanee
วัน เดือน ปี เกิด	วันพฤหัสบดีที่ 6 ธันวาคม พ.ศ. 2522
สถานที่เกิด	19/3 หมู่ 4 ต.ธรรมศาลา อ.เมืองนครปฐม จ.นครปฐม 73000
ที่อยู่ปัจจุบัน	19/3 หมู่ 4 ต.ธรรมศาลา อ.เมืองนครปฐม จ.นครปฐม 73000
ประวัติการศึกษา	สำเร็จการศึกษา ระดับประถมศึกษาที่ โรงเรียนอนุบาลไผทวิทยา ต.พระปฐมเจดีย์ อ.เมืองนครปฐม จ.นครปฐม 73000 ระดับมัธยมศึกษาตอนต้นที่ โรงเรียนวัดห้วยจรเข้วิทยาคม ต.พระปฐมเจดีย์ อ.เมืองนครปฐม จ.นครปฐม 73000 ระดับประกาศนียบัตรวิชาชีพที่ โรงเรียนเทคโนโลยีนครปฐม ต.พระประโทน อ.เมืองนครปฐม จ.นครปฐม 73000 ระดับประกาศนียบัตรวิชาชีพชั้นสูงที่ วิทยาลัยอาชีวะศึกษา ต.พระปฐมเจดีย์ อ.เมืองนครปฐม จ.นครปฐม 73000 ระดับปริญญาตรีที่ สถาบันราชภัฏนครปฐม สาขาคอมพิวเตอร์ศึกษา คณะครุศาสตร์
สถานศึกษาปัจจุบัน	ระดับปริญญาวิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร