สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# หมากรุกไทยบนกริดคอมพิวติ้ง

# THAI CHESS ON GIRD COMPUTING

ประโยชน์ รุจิรา

กมณโลจน์ พรหมผาง

วพ.
ว368ห
2550

.b 11959307
.i

เลขหมู่.....
เลขทะเบียน 83126
วัน,เดือน,ปี - 5 ส.ค. 2551

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หมากรุกไทยบนกริดคอมพิวติ้ง

THAI CHESS ON GIRD COMPUTING

ผู้จัดทำ

1. นายกมณโลจน์ พรหมผาง รหัสนักศึกษา 48015323
2. นายประโยชน์ รุจิรา รหัสนักศึกษา 48015346

ชุติเมษฎ์ ศรีนิลทา อาจารย์ที่ปรึกษา

(ดร. ชุติเมษฎ์ ศรีนิลทา )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# หมากรุกไทยบนกริดคอมพิวติ้ง

นายกมณโลจน์ พรหมผาง 48015323
นายประโยชน์ รุจิรา 48015346
อาจารย์ชุติเมษฏ์ศรีนิลทา อาจารย์ที่ปรึกษา
ปีการศึกษา 2550

**บทคัดย่อ**

งานวิจัยนี้เป็นการนำเอาเทคโนโลยีกริดคอมพิวติ้งมาช่วยในการประมวลผลการคำนวณตาเดินของการเล่นเกมหมากรุกไทย โดยใช้โปรแกรม MATLAB และ Distributed computing engine ในการพัฒนา โดยแบ่งเนื้อหาออกเป็นส่วนสำคัญสองส่วนคือ การพัฒนาโปรแกรมระบบการเล่นของเกมหมากรุก และการสร้างระบบปัญญาประดิษฐ์ของเกมโดยใช้หลักการของการพัฒนาโปรแกรมแบบขนาน มาแก้ปัญหาในส่วนของขั้นตอนการค้นหา โดยใช้วิธีการที่เรียกว่า Minimax search เพื่อทดลองดูว่า สามารถทำให้การคำนวณตาเดินของเกมหมากรุก ที่ต้องใช้พลังในการประมวลผลและทรัพยากรที่สูงมากๆนั้น จะสามารถแก้ปัญหาเหล่านี้ด้วยเทคโนโลยีกริดได้หรือไม่ ต้องใช้เครื่องคอมพิวเตอร์จำนวนกี่เครื่อง สมรรถภาพของเครื่องคอมพิวเตอร์แต่ละเครื่อง และทำการทดสอบและปรับแต่งจนกว่าจะได้จำนวนที่เหมาะสมที่สุดในการคำนวณแต่ละรอบของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Thai chess on grid computing


Mr. Kamonlojn Phrompang 48015323

Mr. Prayoch Rujira 48015346

Dr. Chutimet Srinilta Advisor

Academic Year 2007



**ABSTRACT**

This project implement Thai chess game on grid using MATLAB distributed computing toolbox. There usually exists a number of possible moves at any point during the chess game. Minimax algorithm is used for searching and evaluating the best move. This Minimax algorithm is mostly used in 2 players game such as chess, othello, tic tac toe, go and any two or multiplayer games. Minimax algorithm normally consumes lots of processing power and resources while searching into the game tree. There are many ways to reduce the computation, in this project we choose parallel programming. Lastly, we compare performance when run on different number of machines.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ขอขอบคุณ ดร. ชุติเมษฏ์ ศรีนิลทา ที่ปรึกษาโปรเจคซึ่งคอยติดตามและแนะนำสิ่งที่ควรปฏิบัติที่เกี่ยวข้องกับโปรเจคและอื่นๆ ทำให้สามารถปฏิบัติงานได้จนประสบผลสำเร็จได้

ขอขอบคุณ นพรัตน์ พันธ์เสนา และ สมโชค กิ้มปาน ในการให้ความรู้เบื้องต้นเกี่ยวกับเทคโนโลยีกริดคอมพิวติ้ง และการพัฒนาโปรแกรมแบบขนาน

ขอขอบคุณ ผู้อำนวยการสำนักวิจัยและบริการคอมพิวเตอร์ รศ.ดร.มนัส สังวรศิลป์ ที่เอื้อเฟื้อระบบคอมพิวเตอร์และซอฟต์แวร์สำหรับใช้ในการดำเนินโครงการ

ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ที่เอื้อเฟื้อสถานที่ที่ทำงาน และขอขอบคุณเพื่อนๆพี่ๆและน้องๆ ห้อง OLALA ทุกท่านที่เป็นแรงผลักดันให้ทำงานได้ลุล่วงไป

ขอขอบคุณ คุณพ่อคุณแม่ ญาติพี่น้อง ตลอดจนมิตรสหายทุกท่าน ที่คอยเป็นกำลังใจที่ดีและให้ความช่วยเหลือซึ่งกันและกันตลอดมา

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

นายประโยชน์ รุจิรา

นายกมณโลจน์ พรหมผาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า


บทคัดย่อภาษาไทย ..... I

บทคัดย่อภาษาอังกฤษ ..... II

สารบัญ ..... III

สารบัญตาราง ..... VII

สารบัญรูป ..... VIII

บทที่ I บทนำ ..... I

1.1 หลักการและเหตุผล ..... 1

1.2 วัตถุประสงค์ของโครงงาน ..... 1

1.3 ประโยชน์ที่คาดว่าจะได้รับ ..... 2

1.4 ส่วนประกอบของวิทยานิพนธ์ ..... 2

บทที่ 2 หมากรุกไทย ..... 3

2.1 ตัวหมากรุก ..... 3

2.1.1 ขุน ..... 3

2.1.2 เม็ด ..... 3

2.1.3 โคน ..... 4

2.1.4 ม้า ..... 4

2.1.5 เรือ ..... 5

2.1.6 เบี้ย ..... 5

2.2 กติกาการเล่น ..... 5

2.3 การนับศักดิ์ ..... 6

2.3.1 การนับศักดิ์กระดาน ..... 6

2.3.2 การนับศักดิ์หมาก ..... 6

2.4 Minimax Search ..... 7

2.5 Evaluation function ..... 13

2.5.1 Material balance ..... 13

2.5.2 Mobility and board control ..... 14

2.5.3 Quiescence search ..... 15

2.6 Makemove and Unmakemove ..... 16



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)

หน้า


2.7 Generate legal move & legal capture ..... 16

2.8 สรุป ..... 16

บทที่ 3 Grid computing technology ..... 18

3.1Grid Computing ..... 18

3.2Multiprocessor ..... 19

3.3 Clustering system ..... 20

3.4 การนำGrid computing ใช้งาน ..... 20

บทที่ 4 MATLAB Distributed computing toolbox & engine ..... 23

4.1 MATLAB Distributed Computing Toolbox & Engine ..... 23

4.2 Distributed Computing Toolbox ..... 24

4.3 Distributed Computing Engine ..... 24

4.4 ส่วนประกอบของ MATLAB Distributed computing engine(MDCE) ..... 25

4.5 การใช้งาน MATLAB distributed computing toolbox ..... 26

4.5.1 Job ..... 26

4.5.2 Task ..... 26

4.6 ขั้นตอนการใช้งาน MATLAB distributed computing toolbox ..... 27

บทที่ 5 การออกแบบและพัฒนา ..... 32

5.1 Class Diagram ..... 34

5.2 การออกแบบโปรแกรมหมากรุก ..... 35

5.3 Activity Diagram ..... 39

5.4 การใช้งานโปรแกรม ..... 39

5.4.1 เริ่มต้นสร้างกระดาน ..... 39

5.4.2 แสดงผล GUI ..... 40

5.4.3 เริ่มต้นเดินหมาก ..... 41

5.4.4 เช็คเงื่อนไข ..... 41

บทที่ 6 การทดลอง ..... 42

6.1 ทดลอง Search ตาเดิน ..... 42

6.2 วิเคราะห์ผลการทดลอง ..... 46

6.3 ทำการทดสอบในการเดินหมากกับ A.I. ..... 46



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)

หน้า


บทที่ 7 บทวิจารณ์และสรุป ..... 48

7.1 สรุปผลการทดลอง ..... 48

7.2 แนวทางในการพัฒนาต่อ ..... 49

บรรณานุกรม ..... 50

ภาคผนวก ..... 51



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง


ตาราง หน้า

2.1 ตัวอย่างการกำหนดค่า material cost ..... 14

6.1 ผลการทดลองกับ 1 worker ..... 42

6.2 ผลการทดลองกับ 2 workers ..... 43

6.3 ผลการทดลองกับ 3 workers ..... 43

6.4 ผลการทดลองกับ 4 workers ..... 43

6.5 ผลการทดลองกับ 5 workers ..... 44

6.6 ผลการทดลองกับ 6 workers ..... 44

6.7 ผลการทดลองกับ 7 workers ..... 44

6.8 ผลการทดลองกับ 8 workers ..... 45

6.9 ผลการทดลองกับ 1 เครื่องโดยไม่ผ่าน Distributed Tool box ..... 46



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป


| รูปที่ | หน้า |
|---|---|
| 2.1 ตาเดินของขุน | 3 |
| 2.2 ตาเดินของเม็ด | 3 |
| 2.3 ตาเดินของโคน | 4 |
| 2.4 ตาเดินของม้า | 4 |
| 2.5 ตาเดินของเรือ | 5 |
| 2.6 ตาเดินของเบี้ย | 5 |
| 2.7 แสดงขั้นตอนการทำงานของ Minimax search | 7 |
| 2.8 อัลกอริธึมของ Minimax | 8 |
| 2.9 สถานการณ์ตัวอย่าง (1) | 8 |
| 2.10 โครงสร้าง tree ที่ได้จากการสังเคราะห์ตาเดิน | 9 |
| 2.11 ผลลัพธ์ที่ได้จากการเลือกเส้นทาง Rxb7 | 9 |
| 2.12 tree จากการสังเคราะห์ตาเดินของฝ่ายตรงข้าม | 10 |
| 2.13 ผลลัพธ์จากการที่ฝ่ายดำ เลือกเดินหมากไปที่ Qg7 ในกรณีที่ฝ่ายขาวเลือกเดิน Rxb7 | 11 |
| 2.14 การสังเคราะห์ตาเดิน | 12 |
| 2.15 อัลกอริธึม Minimax(2) | 12 |
| 3.1 ระบบ Symmetric multi processor | 19 |
| 3.2 การเชื่อมต่อแบบ Cluster | 20 |
| 3.3 การเชื่อมต่อแบบ Grid โดยนำ Cluster มาต่อเชื่อมต่อกัน | 21 |
| 3.4 ส่วนประกอบหลักของระบบกริดคอมพิวติ้ง | 21 |
| 4.1 ภาพรวมของระบบ MATLAB distributed computing | 25 |
| 4.2แสดงภาพรวมของระบบเมื่อมีการเชื่อมต่อที่มากขึ้น | 26 |
| 4.3 แสดงการทำงานของระบบ MATLAB distributed computing | 27 |
| 4.4 แผนภาพแสดงขั้นตอนการใช้งาน MATLAB Distributed computing toolbox | 28 |
| 4.5 คำสั่งในการติดต่อกับ jobmanager | 28 |
| 4.6 คำสั่งที่ใช้ในการสร้าง job | 29 |
| 4.7 ตัวอย่างคำสั่งที่ใช้ในการสร้าง task | 29 |
| 4.8 คำสั่งเริ่มการทำงาน | 30 |
| 4.9 คำสั่งสำหรับสั่งให้รอจนกว่าการประมวลผลจะเสร็จสิ้น | 30 |
| 4.10 คำสั่งที่ใช้ในการเรียกดูผลลัพธ์ทั้งหมด | 30 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป(ต่อ)

รูปที่ หน้า


4.11 ผมลัพธ์ที่ได้จากการคำนวณ 31

5.1 แสดง use-case diagram ของโปรแกรม 32

5.2 แสดงการทำงานแบบปกติ 33

5.3 การทำงานแบบขนาน 33

5.4 class diagram ของโปรแกรมหมากรุกไทย 33

5.5 Activity Diagram ของโปรแกรมหมากรุก 39

5.6 แสดงภาพเมื่อพิมพ์คำว่า run2 เพื่อสร้างตัวหมากและกระดาน 40

5.7 แสดงภาพเมื่อพิมพ์คำว่า gui ก็จะแสดง GUI ของตัวโปรแกรมหมากรุกขึ้นมา 40

5.8 แสดงการเดินของหมากที่เราเลือก 41

6.1 แผนภูมิเปรียบเทียบเวลากับคอมพิวเตอร์ที่ใช้ในการ Search ในแต่ละ ระดับความลึก 45

6.2 Message box ของโปรแกรมหมากรุกเมื่อขุนของสีดำโดนรุกฆาต 46

6.3 การหนีของขุนเมื่อมีการรุก 47

ก.1 ภาพรวมของระบบ MATLAB Distributed Computing 51

ก.2 การกรอก PLP 52

ก.3 ข้อความสำหรับไฟล์ license.dat 52

ก.4 การระบุที่อยู่ของไฟล์ license.dat 53

ก.5 เลือกรูปแบบการติดตั้ง CUSTOM 53

ก.6 การเลือกติดตั้ง License Manager 54

ก.7 การเลือก Configure license manager service 54

ก.8 ที่อยู่ของไฟล์ license.dat สำหรับโปรแกรม LMTOOLS 55

ก.9 ข้อมูลภายในไฟล์ liense.dat สำหรับโปรแกรม LMTOOLS 55

ก.10 Textbox ที่ระบุถึงชื่อของ Server ที่ถูกต้อง 56

ก.11 ที่อยู่ของไฟล์ license.dat สำหรับโปรแกรม MATLAB 56

ก.12 ข้อมูลภายในไฟล์ license.dat สำหรับโปรแกรม MATLAB 57

ก.13 หน้าต่างโปรแกรม LMTOOLS 57

ก.14 แท็บ Server status ของโปรแกรม LMTOOLS 58

ก.15 การติดตั้ง MATLAB Distributed Computing Toolbox 58

ก.16 การสั่งให้ MATLAB Distributed Computing Toolbox เริ่มการทำงาน 58

ก.17 ความสัมพันธ์ของ worker 59



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป(ต่อ)

รูปที่ หน้า

ก.18 การติดตั้ง MATLAB Distributed Computing Engine ..... 60
ก.19 การเลือกรูปแบบการติดตั้ง MATLAB Distributed Computing Engine ..... 60
ก.20 การเลือกติดตั้ง License Manager สำหรับ MATLAB Distributed Computing Engine ..... 61
ก.21 Configure license manager service ของ MATLAB Distributed Computing Engine ..... 61
ก.22 การติดตั้ง MATLAB Distributed Computing Engine ..... 62
ก.23 การสั่งให้ MATLAB Distributed Computing Engine เริ่มการทำงาน ..... 62
ก.24 โปรแกรม LMTOOLS สำหรับ Jobmanager และ worker ..... 63
ก.25 หน้าต่าง Property Inspector แสดงรายละเอียดของ Jobmanager ..... 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 หลักการและเหตุผล

ในปัจจุบันที่เทคโนโลยีก้าวหน้าไปอย่างมาก อุปกรณ์คอมพิวเตอร์ในทุกวันนี้มีความเร็วสูงและราคาที่ถูกลงจึงเข้าถึงคนทั่วไปได้โดยง่าย ถึงแม้ว่าคอมพิวเตอร์จะราคาถูกลงแต่ว่าการทำงานที่ใช้ทรัพยากรมากๆนั้นก็จำเป็นต้องใช้เครื่องคอมพิวเตอร์ที่ระดับสูง(Super Computer) ที่ยากแก่การบำรุงรักษา และยังมีราคาสูง ดังนั้น นักวิชาการจึงพัฒนาเทคโนโลยีที่เรียกว่า "กริด คอมพิวติ้ง (Grid Computing)" ขึ้นมาเพื่อใช้ในการประมวลผลงานใหญ่ๆ ได้นำเอาข้อดีของคลัสเตอร์ และ Peer-to-Peer รวมเข้าไว้ด้วยกัน กล่าวคือ คอมพิวเตอร์แต่ละตัวสามารถร่วมกันประมวลผลงานชิ้นเดียวกัน ซึ่งคล้ายกับคลัสเตอร์ แต่ไม่ต้องจำเป็นต้องมีศูนย์กลางที่ใช้ในการแจกจ่ายงาน ซึ่งคล้ายกับ Peer-to-Peer ทำให้สามารถใช้งานได้อย่างยืดหยุ่น

ระบบปัญญาประดิษฐ์(A.I.) ก็เป็นการประมวลผลรูปหนึ่งที่สามารถนำเอาเทคโนโลยี Grid มาใช้ในการคำนวณได้ โดยนำรูปแบบหนึ่งในการสร้างปัญญาประดิษฐ์ ซึ่งต้องมีการกระจายการประมวลผลออกเป็นงานย่อยๆจำนวนมาก เพื่อหารูปแบบการกระทำที่เหมาะสมที่สุดจากการกระทำนับร้อยรูปแบบที่อาจเกิดขึ้นได้เมื่อระบบตัดสินใจอย่างใดอย่างหนึ่ง

ในโครงการนี้ จะใช้เกมหมากรุกไทยในการนำเสนอ เนื่องจากการเล่นมีความซับซ้อนและรูปแบบการเดินหมากที่หลากหลาย และเป็นเกมที่ผู้เล่นจะคาดเดาพฤติกรรมของผู้เล่นฝ่ายตรงข้ามได้ยาก ระบบปัญญาประดิษฐ์จะต้องคิดหารูปแบบการเดินหมากที่เหมาะสมที่สุดเพื่อเอาชนะอีกฝ่ายให้ได้ ซึ่งในขั้นตอนการประมวลผลมีการทำงานที่ซับซ้อนและใช้ทรัพยากรมาก จึงนำเทคโนโลยีกริดเข้ามาช่วยลดเวลาในการประมวลผลของระบบลง โดยการให้เครื่องคอมพิวเตอร์ที่มากกว่าหนึ่งเครื่องขึ้นไปช่วยกันประมวลผล และพิสูจน์ว่า Grid computing ช่วยลดเวลาในการคำนวณลงได้จริงหรือไม่ และต้องใช้เครื่องคอมพิวเตอร์จำนวนเท่าใดจึงจะเหมาะสมที่สุด

## 1.2 วัตถุประสงค์ของโครงงาน

1.2.1. เพื่อศึกษาเทคโนโลยี Grid Computing

1.2.2. ศึกษาและพัฒนาการเขียนโปรแกรมแบบขนาน(Parallel Programming)

1.2.3. เพื่อศึกษาและพัฒนา Grid Computing โดยใช้ MATLAB Distributed Toolbox

1.2.4. พัฒนาเกมหมากรุกไทยโดยใช้เทคโนโลยีกริดมาใช้ในการประมวลผลของระบบปัญญาประดิษฐ์

1.2.5. เพื่อนำเอาเทคโนโลยีมาประยุกต์ เพื่อช่วยในการอนุรักษ์เกมการเล่นของคนไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.3 ประโยชน์ที่ขาดว่าจะได้รับ

1.3.1. มีความรู้ความเข้าใจในเทคโนโลยี Grid Computing สามารถนำไปประยุกต์และพัฒนาต่อไป

1.3.2. มีความรู้ความเข้าใจในการเขียนโปรแกรมแบบขนาน

1.3.3. สามารถใช้ MATLAB Distributed Toolbox ในการพัฒนา Grid Computing

1.3.4. มีความรู้ความเข้าใจในกฎและกติกาของหมากรุกไทยและนำระบบปัญญาประดิษฐ์ไปประยุกต์ใช้งานต่อไป

## 1.4 ขอบเขตของโครงงาน

1.4.1 พัฒนาเกมหมากรุกไทย เพื่อทำงานบน Grid โดยทำการแข่งขันระหว่างผู้เล่นกับคน

1.4.2 พัฒนา A.I. ของหมากรุกไทยโดยใช้ Algorithm Minimax Search

14.3 ประเมินประสิทธิภาพและผลการทำงานโปรแกรมหมากรุกไทย บน Grid

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

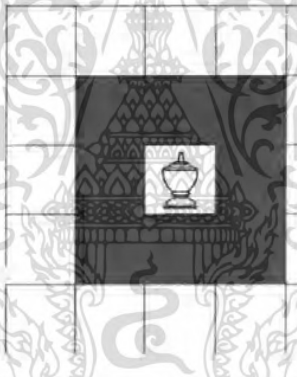
# บทที่ 2
# หมากรุกไทย

หมากรุก เป็นเกมกระดานชนิดหนึ่ง มีลักษณะจำลองจากการทำสงครามในยุคโบราณ ใช้ผู้เล่นสองคน แต่ละฝ่ายต้องพยายามรุกจนขุนของอีกฝ่ายให้ได้ โดยกติกาและตัวหมากอื่นๆจะแตกต่างกันไปแล้วแต่ประเทศหรือชนิดของหมากรุก สำหรับหมากรุกของไทย มีต้นแบบมาจากหมากรุกของอินเดีย มีลักษณะใกล้เคียงกับหมากรุกฝรั่ง

## 2.1 ตัวหมากรุก

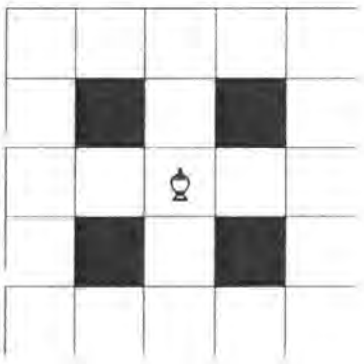
ผู้เล่นแต่ละฝ่ายจะมีตัวหมากคนละ 16 ตัว โดยประกอบไปด้วย

2.1.1 ขุน เป็นตัวหมากรุกที่สำคัญที่สุด มีการเดินและกินไปในทิศทางรอบตัวได้ แต่ไม่สามารถเดินเข้าไปในตากินของฝ่ายตรงข้ามได้



**รูปที่ 2.1 ตาเดินของขุน**

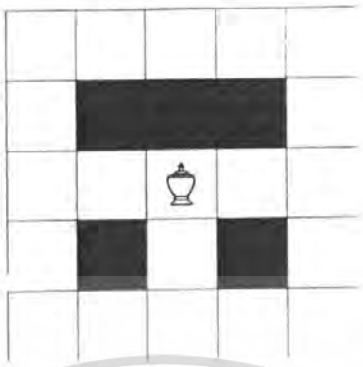
2.1.2 เม็ด มีการเดินและกินไปในแนวทแยงทั้ง 4 ด้าน



**รูปที่ 2.2 ตาเดินของเม็ด**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 โคน มีการเดินและกินไปในแนวทแยงทั้ง 4 ด้านและเดินไปทางข้างหน้า



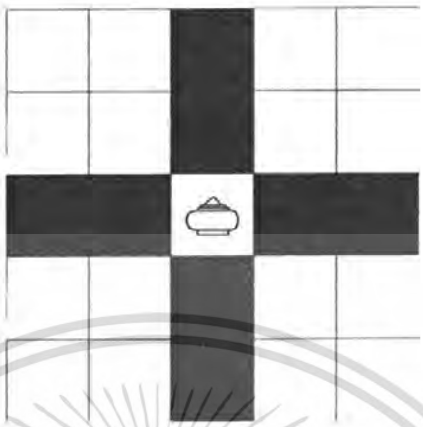
**รูปที่ 2.3 ตาเดินของโคน**

2.1.4 ม้า มีการเดินและกินเป็นรูปตัว L ในทิศทางรอบตัว สามารถข้ามหมากตัวอื่นได้

**รูปที่ 2.4 ตาเดินของม้า**

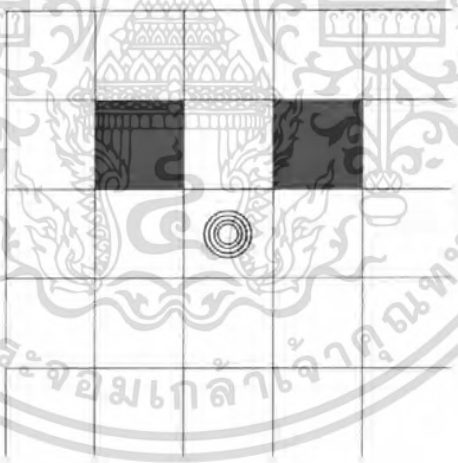
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 เรือ มีการเดินและกินไปในแนวตั้ง-แนวนอน ระยะยาว ไม่สามารถเดินข้ามตัวอื่นๆ ได้



**รูปที่ 2.5 ตาเดินของเรือ**

2.1.6 เบี้ย มีการเดินไปทางข้างหน้าและกินในแนวทแยงด้านหน้า เมื่อเบี้ยไปถึงแนววางเบี้ยของฝ่ายตรงข้ามจะกลายเป็นเบี้ยหงาย และมีการเดินและกินเช่นเดียวกับเม็ดทุกประการ



**รูปที่ 2.6 ตาเดินของเบี้ย**

## 2.2 กติกาการเล่น

- ผู้เล่นแต่ละคนผลัดกันเดินหมากของตัวเอง ครั้งละ 1 ตัว
- ถ้าเดินตัวหมากของตนเองไปในตำแหน่งที่ตัวหมากของฝ่ายตรงข้ามตั้งอยู่ ตัวหมากของฝ่ายตรงข้ามจะถูกกินและนำออกไปจากกระดาน ถ้าตัวหมากของฝ่ายตรงข้ามเป็นขุน จะถือว่าแพ้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ้าเดินหมากไปอยู่ในตำแหน่งที่สามารถกินขุนของฝ่ายตรงข้ามในตาถัดไปได้ จะเรียกว่าการ "รุก" ซึ่งฝ่ายตรงข้ามจะต้องเดินตัวขุน หรือหาวิธีการป้องกันไม่ให้ตัวขุนถูกกิน
- ถ้าตัวขุนถูกรุกอยู่ และไม่สามารถเดินหนีหรือป้องกันการถูกกินได้ จะถือว่า "รุกจน" และเป็นฝ่ายแพ้
- ถ้าขุนไม่ถูกรุก แต่ในตาต่อไปไม่สามารถเดินตัวหมากใดๆได้เลย จะถือว่า "อับ" และมีผลให้ผู้เล่นทั้งสองฝ่าย เสมอกัน

## 2.3 การนับศักดิ์

2.3.1.การนับศักดิ์กระดาน

การนับโดยวิธีนี้ไม่ว่าจะมีตัวหมากอยู่บนกระดานกี่ตัวก็ตามให้เริ่มนับตั้งแต่ 1 เป็นต้นไป โดยฝ่ายเป็นรองเป็นผู้นับฝ่ายเดียว เมื่อทั้งสองฝ่ายไม่มี**เบี้ยคว่ำ**โดยฝ่ายเป็นรองเหลือหมากตั้งแต่สองตัวขึ้นไปให้เริ่มนับศักดิ์กระดานตั้งแต่ 1 ถึง 64 โดยฝ่ายเป็นต่อจะต้องรุกให้จนใน 64 ตามิเช่นนั้นให้เสมอกัน ถ้าระหว่างนับศักดิ์กระดานฝ่ายเป็นรองถูกกินหมากตัวอื่นจนเหลือขุนเพียงตัวเดียวให้เปลี่ยนมานับศักดิ์หมาก เมื่อเริ่มนับศักดิ์กระดานแล้วฝ่ายเป็นต่อกลับกลายเป็นรองก็มีสิทธ์นับศักดิ์หมากเพื่อหาเสมอได้

2.3.2.การนับศักดิ์หมาก

การนับโดยวิธีนี้ให้นับตัวหมากของทั้งสองฝ่ายที่อยู่บนกระดานได้จำนวนเท่าใดก็ให้เริ่มนับศักดิ์หมากต่อไป เช่นมีตัวหมากอยู่บนกระดาน 7 ตัวก็ให้เริ่มนับ 8 โดยฝ่ายเป็นรองเป็นผู้นับฝ่ายเดียว เมื่อฝ่ายเป็นต่อมีหมากดังนี้

เรือลำเดียว ต้องรุกให้ฝ่ายเป็นรองจนภายใน 16 ตาเดิน ไม่เช่นนั้นให้ถือว่าเสมอกัน
เรือสองลำ ต้องรุกให้ฝ่ายเป็นรองจนภายใน 8 ตาเดิน ไม่เช่นนั้นให้ถือว่าเสมอกัน
ม้าตัวเดียว ต้องรุกให้ฝ่ายเป็นรองจนภายใน 64 ตาเดิน ไม่เช่นนั้นให้ถือว่าเสมอกัน
ม้าสองตัว ต้องรุกให้ฝ่ายเป็นรองจนภายใน 32 ตาเดิน ไม่เช่นนั้นให้ถือว่าเสมอกัน
โคนตัวเดียว ต้องรุกให้ฝ่ายเป็นรองจนภายใน 44 ตาเดิน ไม่เช่นนั้นให้ถือว่าเสมอกัน
โคนสองตัว ต้องรุกให้ฝ่ายเป็นรองจนภายใน 22 ตาเดิน ไม่เช่นนั้นให้ถือว่าเสมอกัน
เม็ดและเบี้ยหงาย นับ 64 ตาเดิน ไม่เช่นนั้นให้ถือว่าเสมอกัน

เมื่อได้เริ่มนับศักดิ์หมากแล้วถ้าฝ่ายเป็นรองกินหมากตัวใดตัวหนึ่งของฝ่ายเป็นต่อก็มิให้เปลี่ยนแปลงการนับเป็นอย่างอื่น

เกมหมากรุก เป็นเกมกระดานที่มีกฎกติกาการเล่นที่รัดกุม ถือได้ว่าเป็น Perfect information ในลักษณะของทฤษฎีเกม หรือ 2 persons zero sum game จึงสามารถนำเอากฎ กติกาพื้นฐานทั้งหมายมาเขียนเป็นเกมได้ โดยใช้ทฤษฎีที่เกี่ยวข้องดังนี้

1.) Minimax search

2.) Alpha-Beta cutoff

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.) Quiescence search

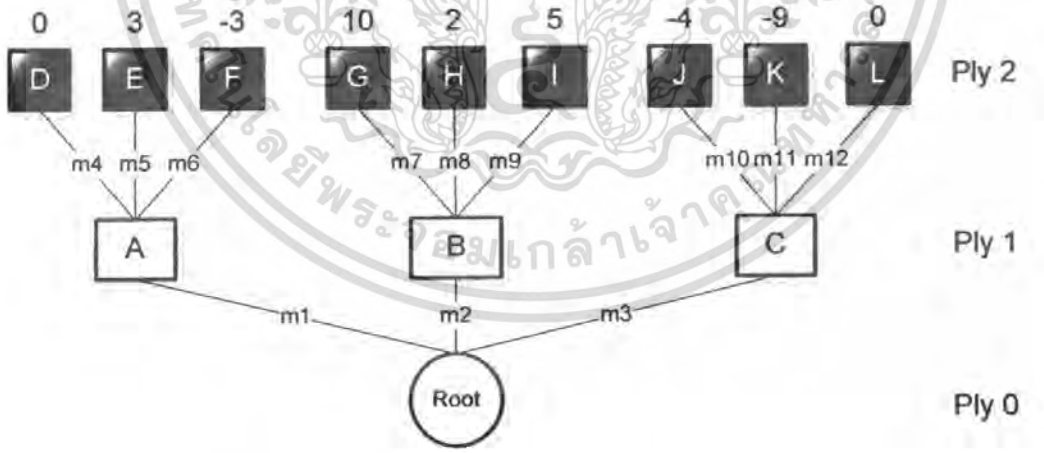
4.) Static/Dynamic Evaluation

5.) Make move & Unmake move

**2.4 Minimax search**

เป็นขั้นตอนวิธีการค้นหาที่ใช้กับโครงสร้างข้อมูลแบบต้นไม้ ( Tree ) โดยเริ่มค้นจากราก( root ) ลึกลงไปเรื่อยๆจนถึงชั้นที่ลึกที่สุด( leaf ) เพื่อเปรียบเทียบหาค่าที่มากที่สุดจากค่าน้อยที่สุดที่ได้มาจากการค้นหาในทุกๆส่วนของต้นไม้

เมื่อพิจารณาที่ตำแหน่ง root สมมติให้เป็นเหตุการณ์ของฝ่ายขาวเริ่มเล่นก่อน ซึ่งฝ่ายขาวจะมีทางเลือกอยู่สามทางคือ m1,m2 และ m3 ถ้าหากฝ่ายขาวเลือกเส้นทาง m1 หรือเดินไปยังโหนด A ฝ่ายดำ จะต้องเลือกเดินระหว่าง m4,m5 หรือ m6 ซึ่งฝ่ายดำ จะต้องเลือกเส้นที่ที่มีการสูญเสียน้อยที่สุด ก็คือ m6 ที่มีค่า -3 แต่ถ้าฝ่ายขาวเลือกเส้นทาง m2 เดินไปยังโหนด B ฝ่ายดำก็จะต้องเลือกเส้นทาง m7,m8 หรือ m9 ซึ่งต้องเป็นเส้นที่ที่มีการสูญเสียที่น้อยที่สุดอีกเช่นกัน นั่นก็คือเส้นทาง m8 ทีมีค่า 2 และเช่นเดียวกัน ถ้าฝ่ายขาว เลือกที่จะใช้เส้นทาง m3 ฝ่ายดำ ก็จะต้องเลือกเส้นทาง m11 ที่มีค่า -9 ซึ่งเป็นเส้นทางที่มีการสูญเสียน้อยที่สุดจากทั้งสามเส้นทางที่สามารถเดินไปได้ ด้วยการประเมินจากเส้นทางทั้งหมด ฝ่ายขาวควรเลือกเส้นทาง m2 เนื่องจากคาดหวังว่าฝ่ายดำ จะเลือกเดิน m8 ซึ่งเป็นการเลือกที่มีการสูญเสียน้อยที่สุด



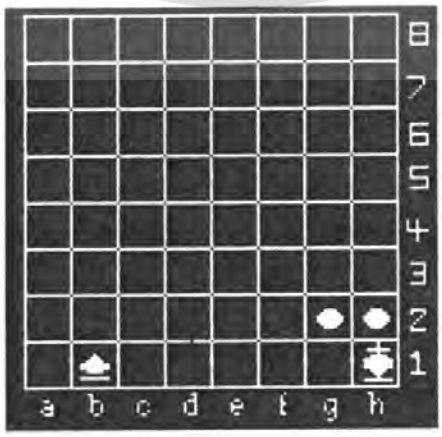

**รูปที่ 2.7 แสดงขั้นตอนการทำงานของ Minimax search**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Search (depth, ply){
      Score , bestscore = -∞
      If (depth < 1){
        ถือว่าเป็น leaf node ให้ทำการเรียกฟังก์ชั่น evaluation เพื่อหาค่า score ของ
ตำแหน่งนี้
        Return ค่า score
       }
      สร้างรายการของตาเดินที่สามารถเดินได้ในตานั้น
      for (everymoves){
       makemove( )
       score = -Search( depth – 1, ply +1 )
       unmakemove( )
       if (score > bestsocre ){
         bestscore = score
         the best move = current move
         update the principal continuation
       }
       }
      return bestscore
      }
```

**รูปที่ 2.8 อัลกอริธึมของ Minimax**

พิจารณาจากรูปตัวอย่างต่อไปนี้




**รูปที่ 2.9 สถานการณ์ตัวอย่าง(1)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 9 พิจารณาที่ความลึก 3 ชั้น ( depth = 3 ) ณ จุดนี้ เรียกว่าเป็น root ดังนั้น ค่า ply จึงเท่ากับศูนย์ ตัวแปรเริ่มต้นถูกกำหนดค่าดังนี้

ply = 0

depth = 3

bestscore = -∞

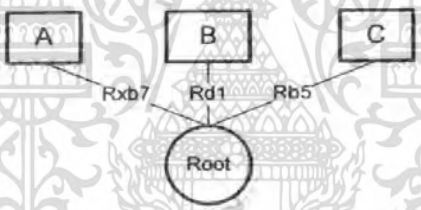
ในส่วนแรกของ Pseudo code คือการเปรียบเทียบค่า depth ว่าน้อยกว่า 1 หรือไม่ เพื่อหาคำตอบว่ามาถึง leaf แล้วหรือยัง ซึ่งในขณะนี้ ค่า depth = 3 จึงยังไม่ใช่ leaf ขั้นตอนต่อไปคือ สร้างรายการตาเดินทั้งหมดที่สามารถเดินได้ในตานั้น เป็นการเรียกฟังก์ชันในการสั่งเคราะห์ตาเดินของตัวหมากทั้งหมดของผู้เล่น จากตัวอย่างที่กำลังพิจารณา สมมติว่าได้ตาเดิน คือ

Rxb7 นำเรือ กินเบี้ย ที่ตำแหน่ง B7

Rd1+ นำเรือ รุกขุน ที่ตำแหน่ง D1

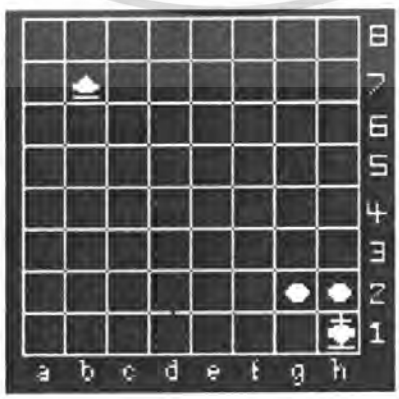
Rb5 นำเรือ ไปที่ตำแหน่ง B5

ซึ่งสามารถนำมาสร้างเป็น tree ได้ดังรูปต่อไปนี้




**รูปที่ 2.10 โครงสร้าง tree ที่ได้จากการสังเคราะห์ตาเดิน**

พิจารณาที่โหนด A ก่อนเป็นอันดับแรก จะให้ผลลัพธ์คือ




**รูปที่ 2.11 ผลลัพธ์ที่ได้จากการเลือกเส้นทาง Rxb7**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับต่อมา จะต้องมีการเรียกใช้ฟังก์ชันแบบ recursive

score = -Search( depth – 1, ply +1 )

สาเหตุที่ต้องให้ค่า score = -search เนื่องจาก ฝ่ายดำจะต้องเป็นผู้เดินในตาถัดไป ดังนั้น จะต้องคำนึงถึงหลักความจริง ตามสมการต่อไปนี้

- opponent score = our score

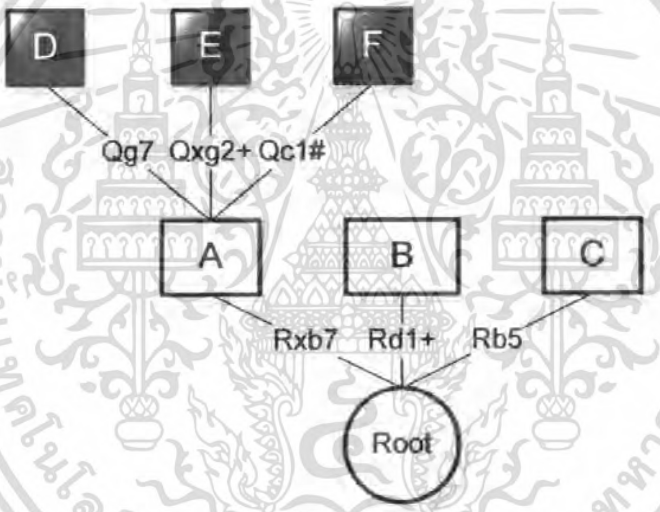
opponent score = - our score

ซึ่งในการทำ recursive จะมีค่าเริ่มต้นคือ

ply = 1

depth = 2

bestscore = -∞

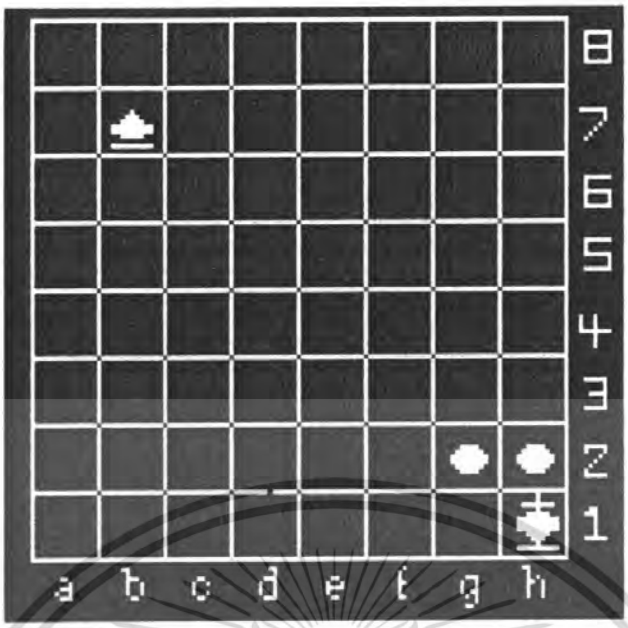



**รูปที่ 2.12 tree จากการสังเคราะห์ตาเดินของฝ่ายตรงข้าม**

ซึ่งค่าความลึก( depth )ก็ยังคงไม่เท่ากับ 1 ก็ต้องเรียกฟังก์ชันในการสังเคราะห์ตาเดินต่อไป ซึ่งรอบนี้จะเป็นการสังเคราะห์ตาเดินของฝ่ายดำ โดยสมมติว่าสังเคราะห์ได้เส้นทาง Qg7,Qxg2+, Qc1#

ตรงจุดนี้ จะเลือกไปที่ เส้นทาง Qg7 หรือโหนด D ก่อน เนื่องจากใช้อัลกอริธึม depth first search ซึ่งจะทำการค้นหาไปยังทุกๆเส้นทางที่มี ผลลัพธ์จากการที่ฝ่ายดำ เลือกเดินหมากไปที่ Qg7 ในกรณีที่ฝ่ายขาวเลือกเดิน Rxb7 ก็คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.13 ผลลัพธ์จากการที่ฝ่ายดำ เลือกเดินหมากไปที่ Qg7 ในกรณีที่ฝ่ายขาวเลือกเดิน Rxb7**

จากนั้นก็จะเกิดการ recursive ขึ้นอีกครั้ง โดยคราวนี้ก็จะมีการกำหนดตัวแปรเริ่มต้นใหม่อีกเช่นเดียวกัน

ply = 2

depth = 1

bestscore = -∞

ซึ่งค่า depth ก็ยังคงไม่น้อยกว่า 1 ดังนั้นก็จะเกิดการสังเคราะห์ตาเดินขึ้นอีก โดยสมมติว่าได้ตาเดินของฝ่ายขาวออกมาเป็น Rxg7, Rb8+ และ Rd7+

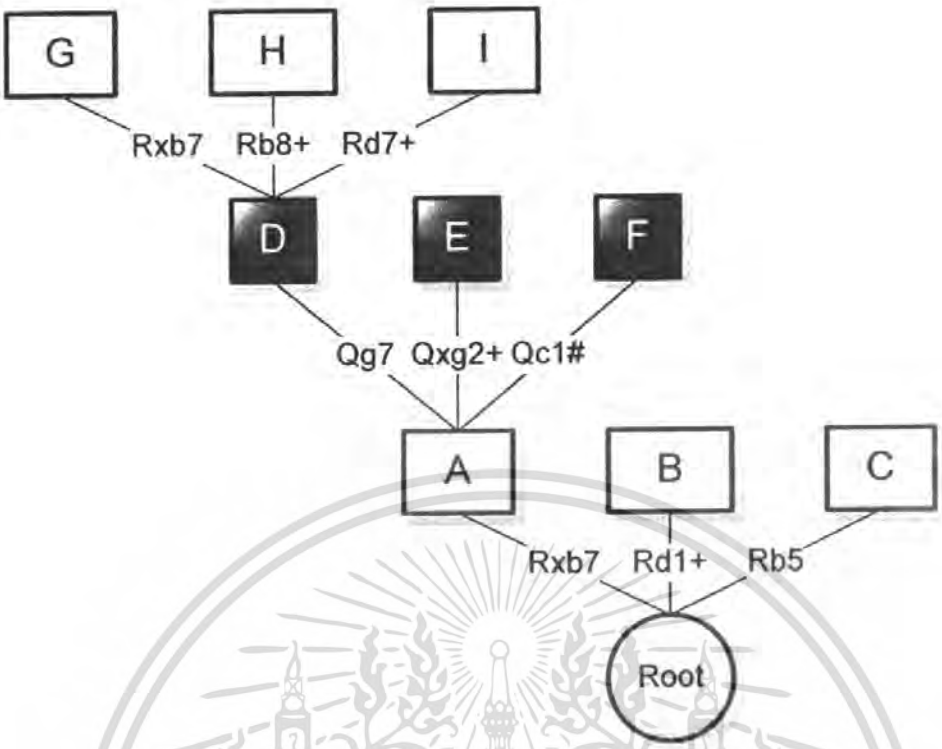
พิจารณาที่ตาเดิน Rg7 ซึ่งแน่นอนว่าจะต้องเรียกฟังก์ชัน search แบบ recursive อีกครั้ง

ply = 3

depth = 0

bestscore = -∞

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.14 การสังเคราะห์ตำแหน่ง**

ในรอบนี้ ค่า depth จะเหลือ 0 ซึ่งตรงตามเงื่อนไข depth < 1 แล้ว ทำให้เกิดการเรียกใช้ฟังก์ชัน evaluation โดยฟังก์ชันนี้ จะทำหน้าที่ในการประเมินค่าความได้เปรียบเสียเปรียบ เรียกอีกอย่างว่า heuristic function ซึ่งจะกล่าวถึงในลำดับต่อๆ ไป สมมติว่าในที่นี้ จะได้ค่าจากการประเมิน 6500 ซึ่งค่านี้จะถูกส่งกลับมาที่โหนด D และจะกลับมาประมวลผลต่อในส่วนที่เหลือ ก็คือ

```
if (score > bestsocre )
      {
       bestscore = score
       the best move = current move
       update the principal continuation
      }
```

**รูปที่ 2.15 อัลกอริธึม Minimax(2)**

ซึ่งจากตัวอย่าง จะได้ค่า score เป็น 6500 ทำให้ค่า bestscore ในขณะนี้มีค่า 6500 และจะทำการ search อีกครั้งในเส้นทาง Rb8+ และ Rd7+ และจะทำอย่างนี้ไปเรื่อยๆ จนครบทุกเส้นทาง ซึ่ง

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรงจุดนี้เป็นปัญหาสำคัญของวิธีการ minimax search เพราะในขั้นตอนการทำงานจริง แต่ละฝ่ายจะมีตัวหมากถึง 16 ตัว และแต่ละตัวนั้นก็มีเส้นทางอีกหลายเส้นทาง โดยเฉลี่ยและจะมีประมาณ 30 เส้นทางเดิน ถ้าหากทำการค้นหาในระดับความลึกมากๆ เช่น ตั้งแต่เริ่มเกม เพียงประเมินอย่างคร่าวๆ ก็จะพบว่ามีโหนดของ tree จำนวนมหาศาลที่จะถูกสร้างขึ้นมา เป็นสาเหตุสำคัญ ที่ทำให้การทำงานมีความล่าช้าเป็นอย่างมาก ดังนั้น minimax search เพียงอย่างเดียวจึงยังไม่เพียงพอต่อการนำมาเขียนโปรแกรมหมากรุก จำเป็นต้องนำวิธีการอื่นๆ เข้ามาช่วยลดจำนวนหรือเวลาในการคำนวณลง ซึ่งจะกล่าวถึงในลำดับต่อๆไป

## 2.5 Evaluation function

ในการค้นหานั้น ทุกครั้งที่เราค้นไปยังจุดๆหนึ่งแล้ว จะต้องมีการประเมินค่าผลลัพธ์ที่ได้ ในโปรแกรมหมากรุกก็เช่นเดียวกัน เมื่อโปรแกรมค้นหาไปจนถึงจุดปลายทาง ก็ต้องมีการประเมินค่า ซึ่งสามารถทำได้ 2 ลักษณะคือ

1. Static evaluation
2. Dynamic evaluation

ลักษณะที่นิยมใช้ในโปรแกรมหมากรุกทั่วๆไปคือ static evaluation สาเหตุมากจากการค้นหาในแต่ละครั้ง จะเกิดเส้นทางและโหนดลูกจำนวนมาก เมื่อค้นหาลึกลงไปมากๆ จำนวนของมันก็จะยิ่งเพิ่มขึ้นมหาศาล สำหรับหน่วยประมวลผลทั่วๆไปต้องเสียทรัพยากรและเวลาไปกับการค้นหาตรงนี้ค่อนข้างมาก หากยังต้องเสียเวลาไปกับการคำนวณ evaluation function อีก จะยิ่งทำให้การค้นหาทำได้ช้าลงไปอีก ดังนั้น หลายๆโปรแกรมจึงได้ทำการกำหนดค่าลงไปเลยว่า หากตัวหมากตัวนี้อยู่ในช่องนี้แล้ว ควรจะมีค่าประมาณเท่าใด ซึ่งค่าๆนี้จะต้องเป็นค่าที่มีช่วงค่อนข้างกว้างหรือมีความยืดหยุ่นสูง อย่างไรก็ตาม ในปัจจุบันยังไม่มีการกำหนดค่าที่แน่นอนของตัวหมากแต่ละตัวให้กับหมากรุกไทย

### 2.5.1 Material balance

คือการกำหนดค่าให้กับตัวหมากแต่ละตัวที่ยังเหลืออยู่บนกระดาน แล้วหาผลรวมเพื่อเปรียบเทียบระหว่างผู้เล่นแต่ละฝ่าย

$$M = sum(Np \times Vp)$$

โดยที่ $M$ คือค่า material balance ของฝ่ายใดฝ่ายหนึ่ง

$Np$ คือ จำนวนของตัวหมากนั้นที่อยู่บนกระดาน

$Vp$ คือ ค่าของตัวหมากนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการกำหนดค่าให้กับตัวหมากแต่ละตัว อาจกำหนดได้ ตามตารางต่อไปนี้

**ตารางที่ 2.1 ตัวอย่างการกำหนดค่า material cost**

| ตัวหมาก | ค่า material |
|---|---|
| เบี้ย | 1 |
| ม้า | 3 |
| เรือ | 5 |
| เม็ด | 7 |
| โคน | 9 |
| ขุน | ∞ |

สาเหตุที่ขุนมีค่า ∞ เนื่องจากขุนจะถูกกินไม่ได้ ในการคำนวณ แต่ละครั้ง จะนำเอาค่า material balance ของแต่ละฝ่ายมาเปรียบเทียบกันหากฝ่ายใดมีค่า material balance มากกว่า แสดงว่าฝ่ายนั้นเป็นฝ่ายได้เปรียบและมีโอกาสชนะมากกว่า และ search agent จะเลือกเอาเส้นทางที่ทำให้ material balance ของเรามีค่ามากที่สุดและให้ฝ่ายตรงข้ามมีค่าน้อยที่สุด

อย่างไรก็ตาม วิธีการเปรียบเทียบค่า material balance ยังไม่เพียงพอสำหรับโปรแกรมหมากรุก ในการคำนวณให้มีประสิทธิภาพมากที่สุดนั้น ยังต้องมีการคำนึงถึงตัวแปรอีกหลายตัว เช่น ความได้เปรียบเสียเปรียบ, ความคล่องตัวในการเดินหมาก, ความปลอดภัยของขุน เป็นต้น

ในกรณีที่เราต้องการเสียสละตัวหมากบางตัวเพื่อแลกเปลี่ยนกับผลประโยชน์บางกระการ เช่น ยอมให้เบี้ยถูกกินเพื่อจะนำตัวหมากอีกตัวไปรุกขุน ถ้าหากยังคงใช้ค่า material balance เพียงอย่างเดียวแล้ว จะไม่สามารถรองรับรูปแบบการเล่นในลักษณะนี้ได้ นอกจากนี้ ยังมีรูปแบบการเล่นในลักษณะอื่นๆอีก ซึ่งในบางครั้งอาจจะต้องค้นหาในระดับที่ลึกลงไปมากกว่า ชั้นที่ต้องการ ซึ่งขึ้นอยู่กับความสามารถและสมรรถภาพของทรัพยากร

**2.5.2 Mobility and board control**

สาเหตุหนึ่งของการที่ตัวหมากตัวใดตัวหนึ่งถูกกินก็คือ ตัวหมากที่ตกเป็นเหยื่อนั้น ไม่สามารถขยับหนีไปที่อื่นได้ โดยสัญชาติญาณแล้ว การมีทางเลือกที่หลากหลายกว่าย่อมจะดีกว่าที่ทางเลือกเพียงทางเดียวหรือไม่มีเลย

ในเกมหมากรุก mobility สามารถประเมินได้โดยการนับจำนวนตาเดินทั้งหมดของตัวหมากแต่ละตัวของแต่ละฝ่าย อย่างไรก็ตาม วิธีการนี้เป็นเพียงการประเมินขั้นต้นเท่านั้น เนื่องจากตาเดินส่วนใหญ่เป็นตาเดินที่ไม่เกิดประโยชน์ หรืออาจทำให้แพ้เกมได้ ในบางครั้ง การพยายามทำให้ฝ่ายตรงข้ามสูญเสียความคล่องตัวอาจจะไปทำลายตำแหน่งที่วางเอาไว้สำหรับการป้องกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งที่สัมพันธ์กับ mobility ก็คือ board control ในเกมหมากรุกฝ่ายที่คุมเกมก็คือการที่มีตัวหมากที่สามารถกินฝ่ายตรงข้ามได้มากกว่า ซึ่งเป็นการรับประกันได้ในระดับหนึ่งว่า สามารถเดินหมากได้อย่างปลอดภัย ในทางกลับกัน เราจะเสียเปรียบอย่างมาก หากเดินหมากในสถานการณ์ที่ฝ่ายตรงข้ามเป็นผู้คุมเกม

จากที่กล่าวมา mobility อาจจะไม่มีความสำคัญมากนักในเกมหมากรุก แต่มีผลมากกับเกม Othello หรือ เป็นเงื่อนไขในการชนะของเกมโกะเลยทีเดียว

**2.5.3 *Quiescence search***

ในการเล่นหมากรุกที่เทคนิคที่สำคัญมากๆอยู่หลายอย่าง ยกตัวอย่างเช่นในกรณีที่มีการกินกันเกิดขึ้น ฝ่ายหนึ่ง อาจเอาเรือ ไปกินม้าของอีกฝ่าย เพราะเห็นว่าสามารถกินได้ฟรีๆ นอกจากนี้เมื่อกินตัวหมากฝ่ายตรงข้ามไปแล้วยังสามารถรุกขุนของฝ่ายตรงข้ามได้อีก เป็นต้น เหตุการณ์เหล่านี้ในการค้นหาด้วย alpha-beta นั้น ไม่สามารถรับมือได้ จึงต้องมีการใช้ quiescence search สิ่งสำคัญสำหรับฟังก์ชันนี้ก็คือการหาค่าที่เหมาะสม(Evaluation) ซึ่งเป็นการประเมินว่า หากมีการกินกันจริง ฝ่ายใดจะได้เปรียบเสียเปรียบอย่างไร หรือมีผลอย่างไร มีอัลกอริธึมดังต่อไปนี้

```
int Quies( int alpha, int beta )
{
      Val = evaluate( );
      if( val >= beta )
            return beta;
      if( val > alpha )
            alpha = val;
      GenerateGoodCaptures( );
      while( CapturesLeft( ) ){
            MakeNextCapture( );
            val = -Quies(-alpha, -beta )
            Unmakemove( );
            if( val >= beta )
                  return beta;
            if( val > alpha )
                  alpha = val;
      }
      return alpha
}
```

**รูปที่ 2.16 อัลกอริธึม Quiescence search**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากอัลกอริธึมโปรแกรม จะเห็นว่าใกล้เคียงกับฟังก์ชัน alpha-beta แต่ก็จะมีจุดแตกต่างกันหลายจุด เช่นการเรียกฟังก์ชั่นประเมินค่า ( Evaluate function ) เพื่อดูคะแนนว่าเหมาะสมหรือคุ้มค่าพอที่จะทำการกินตัวหมากตัวนั้นหรือไม่ ซึ่งโดยทั่วไปแล้วจะมีการนำเอาเทคนิคอื่นมาใช้ควบคู่กันไปด้วย เช่น Most valuable victim/Least valuable attacker แนวคิดหลักๆก็คือ มีการเรียงลำดับการค้นหาและการกินจากคะแนนเพื่อดูว่ากินตัวใดจะได้เปรียบที่สุด ซึ่งง่ายในการนำมาใช้งานจริง แต่ถ้าการค้นหาไม่มีประสิทธิภาพเพียงพอ จะเสียเวลาไปอย่างมากกับกับการประเมินค่าในทุกๆรอบของโปรแกรม

**2.6 Makemove and Unmakemove**

ฟังก์ชั่นสำหรับทดสอบการเคลื่อนย้ายตัวหมาก โดยจะต้องมีการเก็บข้อมูลดังต่อไปนี้

1. ตำแหน่งปัจจุบัน
2. ตำแหน่งที่จะเคลื่อนย้ายตัวหมาก
3. มีการกินตัวอะไรไป
4. มีการหงายเบี้ยหรือไม่
5. คะแนนสำหรับตาเดินนี้คือเท่าไหร่(ใช้สำหรับการตัดสินใจเลือกเส้นทางในการค้นหา)

ทั้ง makemove และ unmakemove จึมีการทำงานคล้ายๆกับ เพียงแต่ unmakemove จะเป็นการทำงานย้อนหลังกลับ สิ่งที่สำคัญก็คือ ในการทดลองเดินหมากนั้น จำเป็นจะต้องตรวจสอบอยู่เสมอว่า ตัวเดินที่เป็นขุนจะเดินไปตำแหน่งที่ถูกรุกไม่ได้ และต้องตรวจสอบกฎอยู่ตลอดเวลา

**2.7 Generate legal move & legal capture**

เป็นฟังก์ชันที่ทำหน้าที่ในการสังเคราะห์ตาเดินโดยดูว่าฝ่ายใดเป็นฝ่ายเดินก็จะทำการสังเคราะห์ตาเดินของฝ่ายนั้นๆเก็บเอาไว้ในโครงสร้างข้อมูลเพื่อทำการทดลองเดินในฟังก์ชัน makemove หรือ unmakemove

**2.8 สรุป**

หลักสำคัญในการสร้างโปรแกรมหมากรุกก็คือจะต้องมีโครงสร้างข้อมูลดังต่อไปนี้

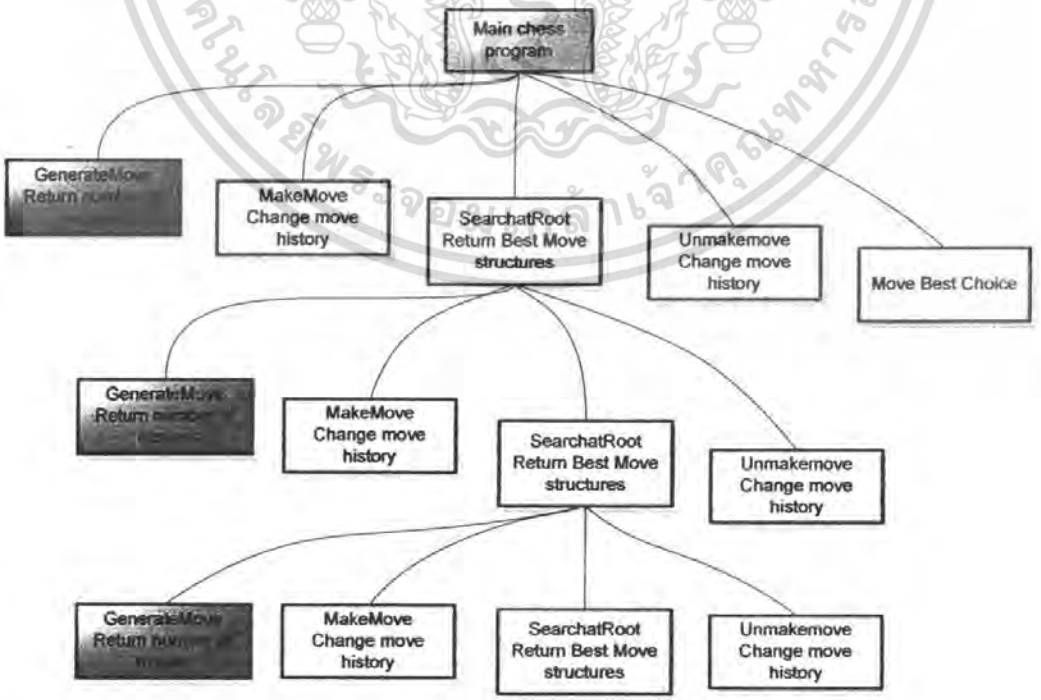
1. กระดาน เพื่อใช้เก็บค่าตัวหมากรุก
2. โครงสร้างข้อมูลในการเก็บตำแหน่งที่สามารถเดินได้ สำหรับกรณีเรียกใช้ฟังก์ชั่น generate legal moves หรือ generate legal captures ซึ่งจะใช้ควบคู่กับบอร์ดเพื่อดูว่ามีตัวหมากอะไรอยู่ในกระดานบ้าง และตัวหมากนั้นสามารถเดินไปตำแหน่งไหนได้บ้าง หรือกินตัวหมากของฝ่ายตรงข้ามตัวไหนได้บ้าง ซึ่งทั้งหมดจะต้องเป็นไปตามกฎของหมากรุก เช่น เบี้ยจะเดินถอยหลังไม่ได้ หรือ เบี้ยจะกินตรงไม่ได้ หรือ ตัวมากไม่สามารถกินกันเองได้เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง


3. ฟังก์ชันในการตรวจสอบตัวขุนว่าถูกรุกอยู่หรือไม่ ซึ่งถ้าหากว่าถูกรุกอยู่ต้องเดินขุนอย่างเดียวเท่านั้น ห้ามเดินตัวอื่น
4. ฟังก์ชันในการ make move และ unmake move เป็นฟังก์ชันเพื่อใช้ในการทดลองเดินตัวหมาก โดยเลือกเดินจากตาเดินที่ทำการสังเคราะห์ออกมาได้เพื่อทดสอบดูว่าเมื่อเดินแล้วจะเกิดผลอย่างไร
5. ฟังก์ชันในการค้นหาเส้นทาง ในการทดลองเดินตัวหมากต่างๆ ก็เพื่อการค้นหาการเดินหมากที่เหมาะสมที่สุด ซึ่งถือว่าเป็นหัวใจหลักของโปรแกรม ยิ่งค้นหาลึกเท่าไหร่ก็ยิ่งใช้เวลาและทรัพยากรมากขึ้นเท่านั้น ดังนั้น จะต้องนำเอาวิธีการต่างๆเข้ามาใช้เพื่อให้โปรแกรมสามารถคำนวณได้เร็วขึ้น
6. ฟังก์ชันสำหรับค้นหาทางเลือกในการกินตัวหมากของฝ่ายตรงข้าม (Quiescence search) อยู่ในขั้นตอนการค้นหา เมื่อค้นหาลงไปจนถึงโหนดสุดท้ายแล้ว ก่อนที่จะเรียกใช้ evaluation function จะต้องประเมินดูก่อนว่าการกินฝ่ายตรงข้ามหรือไม่กินการกระทำแบบไหนจะคุ้มค่ามากที่สุด
7. ฟังก์ชัน Evaluation สำหรับประเมินค่า เป็นฟังก์ชันที่ถูกเรียกใช้ในเวลาที่การค้นหามาถึง leaf ส่งค่ากลับคืนไปยังโปรแกรมว่าการเดินมาถึงตำแหน่งนั้นมีค่าเท่าไหร่

จากวิธีการที่กล่าวมาทั้งหมดสามารถสรุปแผนผังการทำงานของโปรแกรมได้ดังรูป




**รูปที่ 2.17 สรุปการทำงานของโปรแกรม**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
83126

# บทที่ 3

# Grid computing technology

Grid มีความหมายแรก คือ ตารางหรือตะแกรงหรือตาข่าย เรารู้จักกันดี เช่น ตาราง Excel ตาข่ายร่างแห ตารางหมากรุก ตารางหมากฮอร์ส โดยถ้าพิจารณา 2 ตารางหลังนี้ จะมีอุปกรณ์การเล่น(ถือว่าเป็นทรัพยากร)และมีวิธีการเล่น(process)ต่างกัน ซึ่งตารางหมากรุก และตารางหมากฮอร์ส จะมีเส้นทางการเดินทรัพยากรหรือเรียกว่าเส้นทางเดินหมากที่เชื่อมโยงติดต่อถึงกันระหว่างจุดต่อจุดรวมกันเป็นเครือข่ายหรือตาข่าย แต่ละจุดหรือช่อง(cell)ของตารางใดๆ จะกำหนดให้เป็นหนึ่งแฟลตฟอร์ม ถ้าจะบังคับทางเดินทรัพยากรหรือเดินหมากไปมา(distribute)ระหว่างแฟลตฟอร์ม ก็จะมีวิธีการเดินตาม process ที่ได้กำหนดไว้ สามารถกำหนดการเดินหมากให้กระโดดจากแฟลตฟอร์มหนึ่งไปยังอีกแฟลตฟอร์มหนึ่งที่อยู่ห่างไกลได้ ซึ่งผู้เล่นหรือ user หรือ End User จะต้องเรียนรู้ และเข้าใจวิธีการ(Process)

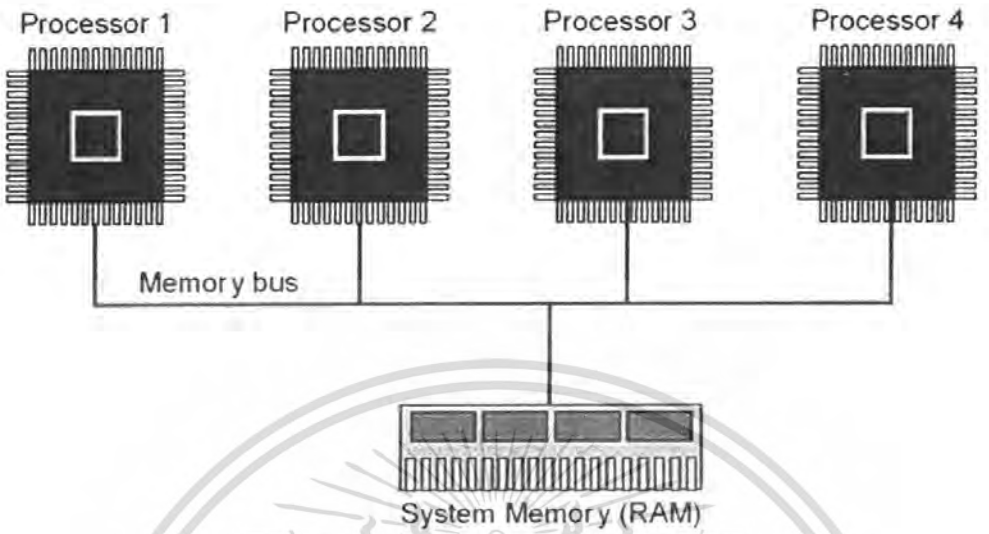
Grid มีบัญญัติความหมายไว้อีกอย่างว่า คือระบบการจ่ายกระแสไฟฟ้า ซึ่งลักษณะที่เราเห็นก็คือหากผู้ใช้ (user) ต้องการใช้อุปกรณ์ไฟฟ้าใดๆ เช่น ตู้เย็น ทีวี ก็เพียงแค่เราเสียบปลั๊กของอุปกรณ์นั้นเข้ากับเต้าเสียบที่ตำแหน่งต่างๆ ตามกำหนดไว้ในบ้านเรือน อุปกรณ์ไฟฟ้าก็จะได้รับพลังงานจากกระแสไฟฟ้านั้น เพื่อเปลี่ยนแปลงรูปแบบให้ผู้ใช้ได้บริโภคตามวัตถุประสงค์ โดยทุกๆ ครัวเรือน สามารถใช้กระแสไฟฟ้าจากแหล่งทรัพยากรเดียวกันได้ คือจากแหล่งผลิตกระแสไฟฟ้า เช่น เขื่อน หรือโรงผลิตไฟฟ้า นั่นเอง

## 3.1 Grid Computing

Grid computing คือ เป็นเทคโนโลยี (Grid Technology) หรือนวัตกรรม (Innovation) ที่ได้ผ่านการวิจัยและพัฒนาขึ้นมา เพื่อให้ระบบทำการคำนวณ หรือประมวลผลข้อมูลที่ซับซ้อน ด้วยสมรรถนะสูง โดยได้จัดเอาทรัพยากรด้านคำนวณหรือทรัพยากรประมวลผลด้านคอมพิวเตอร์ ซอฟต์แวร์ เครื่องมือ อุปกรณ์ต่างๆ มาทำการต่อเชื่อมโยงให้ถึงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 Multiprocessor



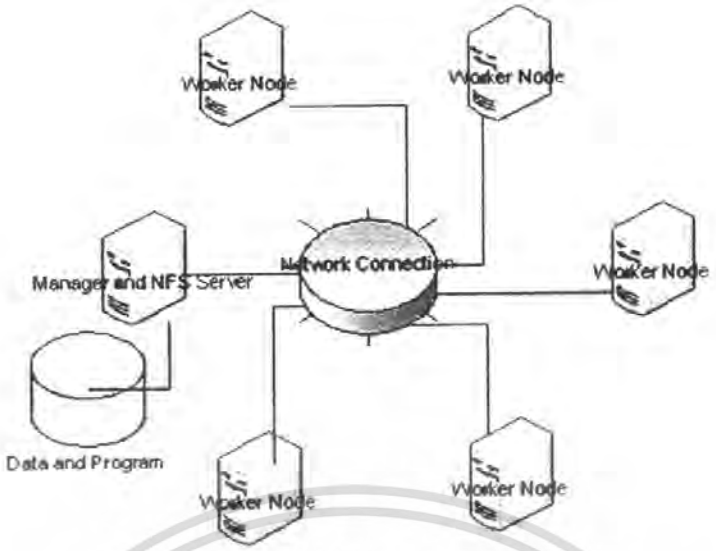

**รูปที่ 3.1 ระบบ Symmetric multi processor**

Multiprocessor เป็นการนำตัวประมวลผล CPU (Central Processing Unit) หลายๆตัวมาต่อกันเพื่อสามารถเพิ่มประสิทธิภาพการประมวลผลได้มากขึ้น โดยตัวประมวลผลแต่ละตัวจะได้รับงานแต่ละอย่างเพื่อประมวลพร้อมกันในเวลาเดียวกัน

### 3.3 Clustering system

Cluster มีความหมายกลางๆ ว่า เป็นกลุ่มคนหรือกลุ่มสิ่งของที่อยู่รวมกันเป็นกระจุกหรือหนาแน่น อยู่บนขอบเขตพื้นที่จำกัดในแฟลตฟอร์มเดียวกัน จะติดต่อเชื่อมโยงหรือประสานงานทรัพยากรทำงานร่วมกันได้เฉพาะในแฟลตฟอร์มเดียวกัน หรือสถานที่เดียวกันเท่านั้น เช่น การนำคอมพิวเตอร์หลายๆ เครื่อง(แต่ละเครื่องอาจจะมี CPU มากกว่า 2 ตัวขึ้นไป) ที่ไม่ถูกใช้งานบางช่วงเวลา หรือไม่ได้ใช้งาน ไม่ว่าจะเป็นรุ่นเก่า หรือรุ่นใหม่ มาต่อรวมกันให้ทำงานตามวัตถุประประสงค์ ของมหาวิทายลัยหนึ่งๆ ก็ถือว่าเป็น Cluster หนึ่งของมหาวิทยาลัยนั้นๆ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

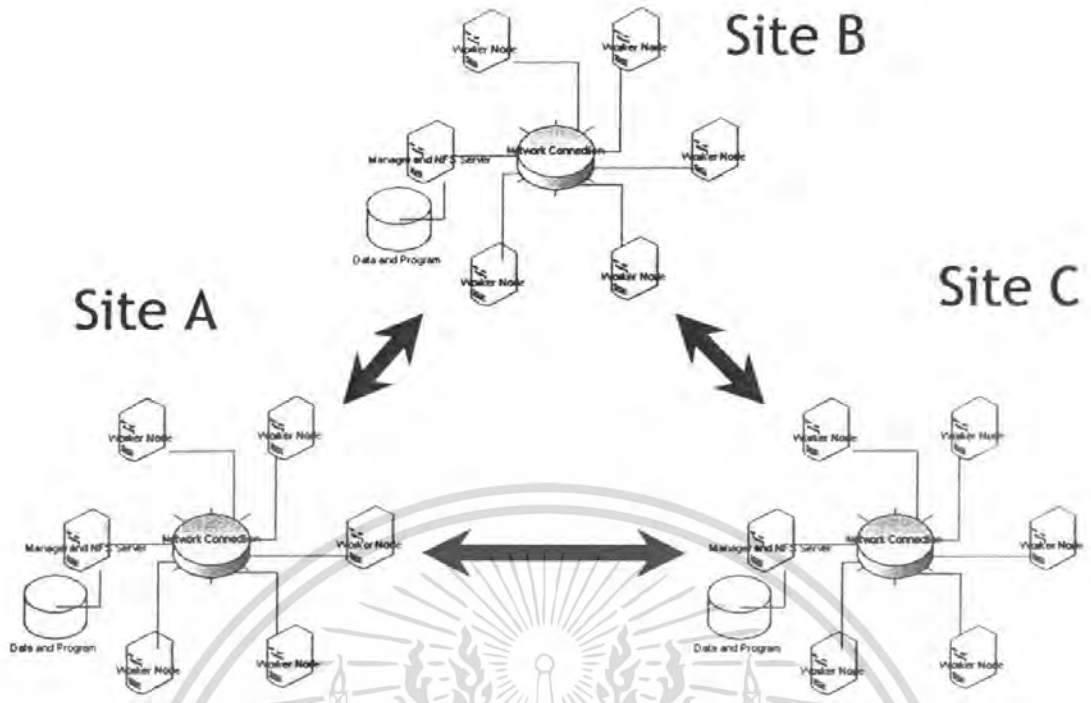



**รูปที่ 3.2 การเชื่อมต่อแบบ Cluster**

## 3.4 การนำGrid computing ใช้งาน

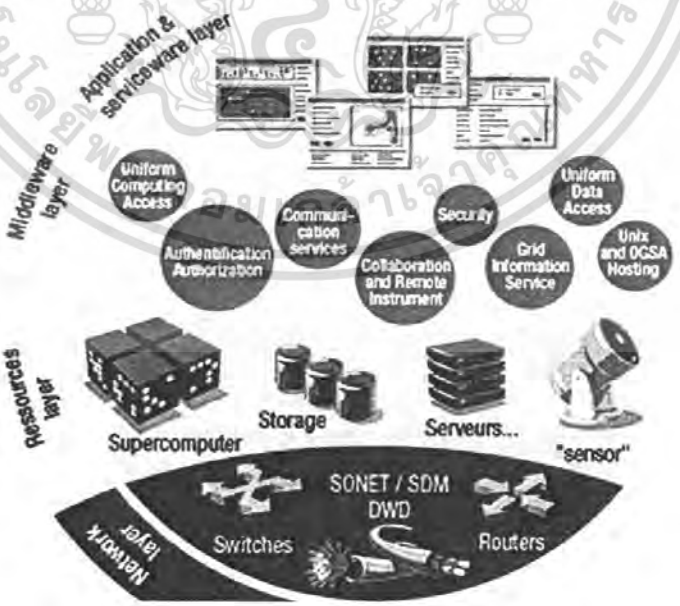
Grid computing ถูกออกแบบให้ผู้ที่อยู่ในเครือข่ายเพียงเข้ามาและใส่รหัสก็เข้าถึงทรัพยากรคอมพิวเตอร์ได้อย่างสะดวก และเรียกใช้ทรัพยากรได้มาตามต้องการช่วยการประมวลผลอันซับซ้อน โดยการจัดเอาทรัพยากรด้านคำนวณหรือทรัพยากรประมวลผลด้านคอมพิวเตอร์ software เครื่องมือ อุปกรณ์ต่างๆ มาทำการเชื่อมต่อให้ถึงกันให้ทำงานร่วมกับระบบเครือข่ายขนาดใหญ่เพียงระบบเดียว ในรูปแบบ Grid เพื่อทำการคำนวณพร้อมกันในเวลาเดียวกัน โอนถ่ายข้อมูลระหว่างกัน โอนถ่ายข้อมูลระหว่างกันไม่ว่าทรัพยากรดังกล่าว จะมีลักษณะแตกต่างกัน เก่าหรือใหม่ ไม่ได้ใช้งานบางช่วงเวลา หรือไม่ได้ถูกใช้งาน จะอยู่ใน Cluster เดียวกัน หรือไม่ สถานที่ไหลแค่ไหน ก็สามารถใช้ประมวลผลร่วมกันได้เป็นซุปเปอร์คอมพิวเตอร์ ขนาดใหญ่เครื่องเดียว ที่มีราคาต้นทุนต่ำประมวลผลแบบของ Grid computing คือ จัดให้ประมวลผลแบบขนาน (Parallel Processing หรือ Parallel Computer) เพื่อให้ทำงานพร้อมกัน หากส่วนใดในระบบขัดข้องหรือไม่ทำงาน ระบบก็ยังทำงานต่อไปได้ เพราะมีซอฟต์แวร์กลางพิเศษช่วยจัดการดูแลตรวจสอบสถานะของระบบ Grid ตลอดเวลา ซึ่งเรียกว่า Middleware

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.3 การเชื่อมต่อแบบ Grid โดยนำ Cluster มาต่อเชื่อมต่อกัน**

Grid Computing แบ่งส่วนประกอบหลักของมันมีอยู่ 4 ส่วน โดยแต่ละส่วนแบ่งเป็นชั้นๆ (layer) ดังนี้




**รูปที่ 3.4 ส่วนประกอบหลักของระบบกริดคอมพิวติ้ง**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Network Layer คือ ส่วนของอุปกรณ์ Network ที่ใช้ในการเชื่อมต่อระบบ Grid เข้าด้วยกัน เช่น Gigabit Switch และ Router เป็นต้น

Resource Layer คือ ส่วนของ Hardware ที่ใช้ในการทำระบบ Grid เช่น Server หรือ Computing Devices, และ Storage Devices

Middleware Layer คือ software ที่ใช้ในการควบคุมการทำงานของอุปกรณ์ต่างๆ ให้ทำงานเข้ากันได้ไม่ว่าจะเป็น Server, Storage, และ Networks ที่มีความแตกต่างกัน เช่น NPACI ROCK เป็นต้น

Application & Serviceware Layer คือ เป็นส่วนที่ประกอบไปด้วย Tools และ Application ต่างๆ ที่ช่วยให้ผู้ใช้งาน สามารถใช้งาน Grid ได้อย่างสะดวก เช่น โปรแกรมพยากรณ์อากาศ, โปรแกรมทางด้าน Bioinformatics, Drug Discovery เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

# MATLAB Distributed computing toolbox & engine

## 4.1 MATLAB Distributed Computing Toolbox & Engine

MATLAB เป็นโปรแกรมคอมพิวเตอร์สมรรถนะสูงเพื่อใช้ในการคำนวณทางเทคนิค MATLABได้รวมการคำนวณ การเขียนโปรแกรมและการแสดงผลรวมกันอยู่ในตัวโปรแกรมเดียวได้อย่างมีประสิทธิภาพ และอยู่ในลักษณะที่ง่ายต่อการใช้งาน นอกจากนี้ลักษณะของการเขียนสมการในโปรแกรมก็จะเหมือนการเขียนสมการคณิตศาสตร์ที่เราคุ้นเคยคือยู่แล้ว งานที่ทั่วไปที่ใช้ MATLAB ก็เช่น การคำการคำนวณทั่วไป การสร้างแบบจำลองและการทดสอบแบบจำลอง การวิเคราะห์ข้อมูล การแสดงผลในรูปกราฟทั้งโดยทั่วไปและกราฟทางด้านทางวิทยาศาสตร์และวิศวกรรม สามารถสร้างโปรแกรมในลักษณะที่ติดต่อกับผู้ใช้ทางกราฟฟิกส์

การทำงานของ MATLAB จะสามารถทำงานได้ทั้งในลักษณะของการติดค่อโดยตรง (Interactive)คือการเขียนคำสั่งเข้าไปทีละคำสั่ง เพื่อให้ MATLAB ประมวลผลไปเรื่อยๆ หรือสามารถที่จะรวบรวมชุดคำสั่งเรานั้นเป็นโปรแกรมก็ได้ ข้อสำคัญอย่างหนึ่งของ MATLAB ก็คือข้อมูลทุกตัวจะถูกเก็บในลักษณะของ array คือในแต่ละตัวแปรจะได้รับการแบ่งเป็นส่วนย่อยเล็กๆขึ้น (หรือจะได้รับการแบ่งเป็น element นั่นเอง) ซึ่งการใช้ตัวแปรเป็น array ในMATLAB นี้เราไม่จำเป็นที่จะต้องจอง dimension เหมือนกับ การเขียนโปรแกรมในภาษาขั้นต่ำตัวไป ซึ่งทำให้เราสามารถที่จะแก้ปัญหาของตัวแปรที่อยู่ในลักษณะของ matrix และ vector ได้โดยง่าย ซึ่งทำให้เราลดเวลาการทำงานลงได้อย่างมากเมื่อเทียบกับการเขียนโปรแกรมโดย C หรือ Fortran

สำหรับในปัจจุบันนี้ MATLAB ได้ถูกเขียนขึ้นโดยใช้ภาษา C โดยบริษัท MathWorks ภายใต้โครงการ LAPACK และ ARPACK ถ้าหากเราจะเริ่มนับจากโปรแกรมที่ออกเผยแพร่เป็นครั้งแรกที่มีผู้ร่วมเขียนโปรแกรมไม่กี่คน จนกระทั่งทุกวันนี้มีทีมงานขนาดใหญ่ที่ทำงานในการพัฒนาโปรแกรมให้มีประสิทธิภาพสูงขึ้น ซึ่งทำให้ทุกวันนี้ MATLAB เป็นโปรแกรมที่สุดยอดในการคำนวณที่คำนวณด้านmatrix สำหรับงานทางวิทยาศาสตร์และวิศวกรรมโปรแกรมหนึ่ง นับจากวันแรกที่ได้เริ่มโครงการขึ้นจนกระทั่งในไตรมาสสุดท้ายของปี ค.ศ.2000 ได้พัฒนาเป็น MATLAB 6.0 ซึ่งเป็นการปรับปรุงใหม่และออกสู่ตลาดเป็นครั้งที่ 12 สำหรับในมุมมองของการศึกษานั้น MATLAB ถือได้ว่าเป็นเครื่องมือที่สำคัญอันหนึ่งสำหรับนักศึกษาทางด้านวิทยาศาสตร์และเทคโนโลยีที่จะใช้เป็นเครื่องมือในการคำนวณ เป็นภาษาสำหรับการใช้งานทางด้านเทคโนโลยี นั่นคือมีระดับเป็นภาษาเหมือนกับภาษา C หรือ Fortran นั่นเอง

นอกเหนือจากตัวโปรแกรม MATLAB เองแล้ว บริษัท MathWorks ผู้ผลิต MATLAB ยังได้ผลิตเครื่องมือหรือที่เรียกว่า toolbox ซึ่งเป็นโปรแกรมที่เขียนขึ้นเพื่อประกอบกับการใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MATLAB สำหรับงานที่จำเพาะเจาะจงหลายประเภท Toolbox นั้นเป็นการนำเอาโปรแกรมที่เขียนขึ้นเป็นฟังก์ชันสำหรับMATLAB เพื่อรวมเข้าเพื่อให้ผู้ใช้งานมีความสะดวกในการเรียกใช้มากขึ้น ทำให้ผู้ใช้ไม่จำเป็นที่จะต้องสร้างโปรแกรมขึ้นมาใช้งานเอง โดย toolbox ที่สร้างขึ้นจะครอบคลุมการทำงานด้านต่างๆมากมายเช่น signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation เป็นต้น

ในปัจจุบัน ระบบคลัสเตอร์ หรือระบบการทำงานแบบหลายหน่วยประมวลผล ได้รับความนิยมมากขึ้นในงานวิจัยต่างๆ ตัว MATLAB ก็มี Distributed computing toolbox ออกมาเพื่อรองรับการทำงานในรูปแบบหลายหน่วยประมวลผลเช่นเดียวกัน โดย ประกอบไปด้วยสองส่วนคือ toolbox และ engine

**4.2 Distributed Computing Toolbox**

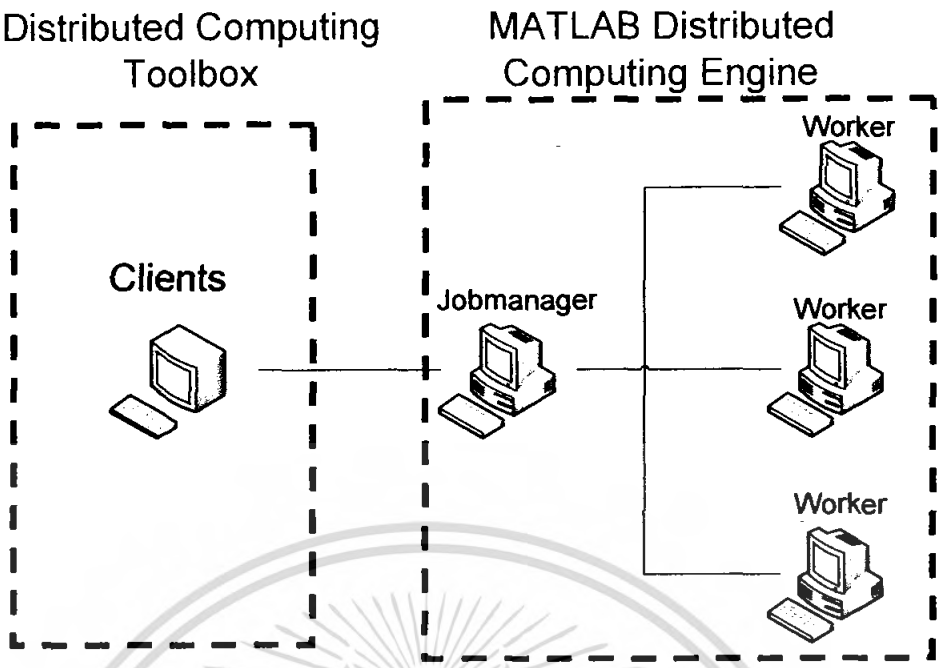
คือส่วนที่มีไว้ติดต่อกับผู้ใช้งาน เมื่อผู้ใช้เขียนโค้ดโปรแกรมขึ้นมาเสร็จแล้วต้องการประมวลผล ก็จะต้องสั่งงานผ่านเครื่องที่ติดตั้ง Toolbox ในส่วนนี้เอาไว้ เราสามารถเรียกเครื่องคอมพิวเตอร์เหล่านี้ว่า Clients ก็ได้ ซึ่งตัวโปรแกรมเองก็จะทำหน้าที่ในการติดต่อกับส่วนของ Engine อีกต่อหนึ่ง

**4.3 Distributed Computing Engine**

มีชื่อเรียกเต็มๆว่า MATLAB Distributed Computing Engine (MDCE) คือส่วนที่คอยรับคำสั่งจากผู้ใช้ เพื่อนำมาประมวลผล โดยมีสองส่วนคือ jobmanager และ worker

Jobmanager ทำหน้าที่ในการรับงานจากผู้ใช้ แล้วกระจายงานออกไปให้กับ worker ที่เชื่อมต่ออยู่ เมื่อ worker ประมวลผลเสร็จแล้วก็ส่งกลับมาให้กับ jobmanager ซึ่งจะรวบรวมผลลัพธ์ที่ได้จากการประมวลผล ส่งกลับคืนไปยังผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.1 ภาพรวมของระบบ MATLAB distributed computing**

## 4.4 ส่วนประกอบของ MATLAB Distributed computing engine(MDCE)

ประกอบไปส่วนส่วนสำคัญสองส่วนคือ Job manager และ worker เชื่อมต่อกันด้วยระบบ LAN หรือเพื่อการติดต่อที่มีประสิทธิภาพมากขึ้น อาจใช้การเชื่อมต่อแบบ Gigabit Ethernet ก็ได้

Job manager

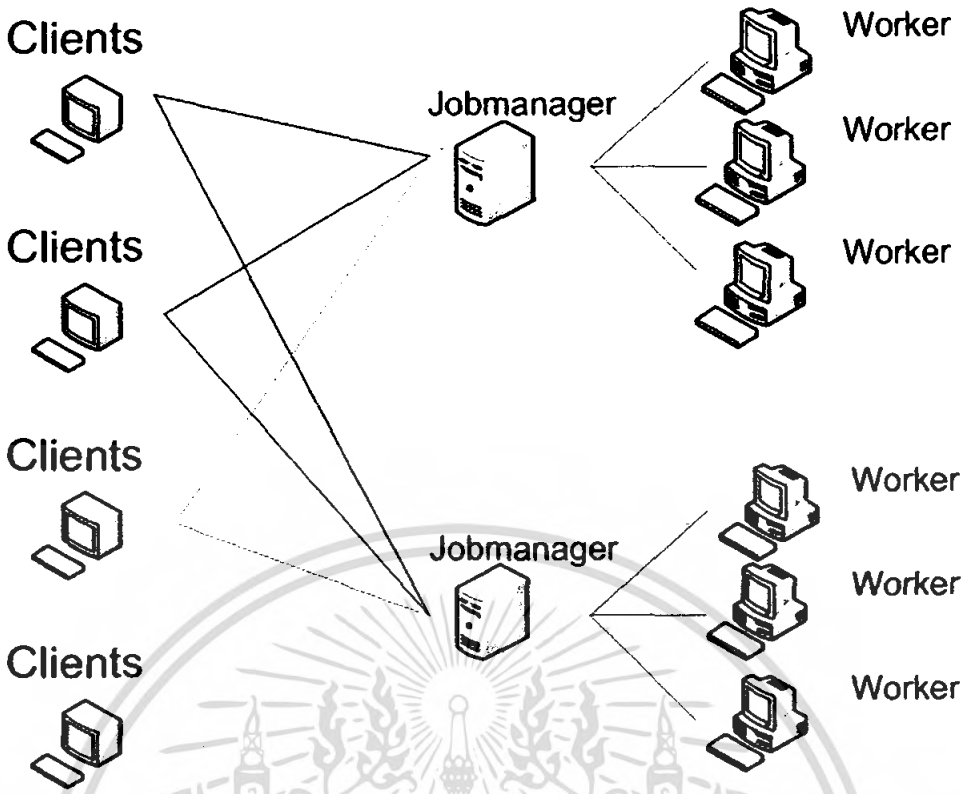
ส่วนสำคัญของระบบ ทำหน้าที่ในการรับข้อมูลหรืองานทั้งหมดจากผู้ใช้ แล้วกระจายงานไปให้กับ worker แต่ละตัวที่ขึ้นกับ job manager ตัวนั้น กล่าวคือ เครื่องคอมพิวเตอร์ที่ติดตั้ง MDCE สามารถมี jobmanager และ worker กี่ตัวก็ได้ และ jobmanager กับ worker สามารถอยู่บนเครื่องเดียวกันได้

Worker

คือส่วนที่ทำหน้าที่ในการรับงานย่อยๆ ( เรียกว่า task ) ที่ได้รับมาจาก jobmanager ในขั้นตอนของการติดตั้ง จำเป็นต้องกำหนดให้ worker ที่สร้างขึ้นมาใหม่ขึ้นกับ jobmanager ตัวใดตัวหนึ่ง ซึ่งสามารถสร้าง worker กี่ตัวก็ได้ในเครื่องคอมพิวเตอร์ 1 เครื่อง

ถึงแม้ว่าระบบ จะอนุญาตให้เราทำการสร้าง jobmanager และ worker กี่ตัวก็ได้ แต่ก็ต้องคำนึงถึงสมรรถภาพด้วยว่า สามารถรองรับการทำงานได้มาก น้อยเพียงใด จากการทดลองด้วยโปรแกรม benchmark พบว่า เครื่องที่มี worker 2 ตัวใน job manager ตัวเดียวกัน จะมีเวลาในการทำงานที่เท่ากัน แต่จะช้ากว่าเครื่องที่มี worker เพียงตัวเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.2แสดงภาพรวมของระบบเมื่อมีการเชื่อมต่อที่มากขึ้น**

## 4.5 การใช้งาน MATLAB distributed computing toolbox

ในการเข้าใช้งานระบบ distributed computing ผู้ใช้ จะต้องทำการเขียนประแกรมที่เครื่อง client ซึ่งโปรแกรมที่เขียนขึ้นนั้น ต้องมีการออกแบบให้ใช้งาน distributed computing มาแล้ว กล่าวคือ เราไม่สามารถสร้างโปรแกรมแบบขนานโดยอัตโนมัติได้ในระบบนี้

ในขั้นตอนการออกแบบโปรแกรม จะต้องคำนึงถึงสิ่งต่างๆ ดังต่อไปนี้

### 4.5.1 Job

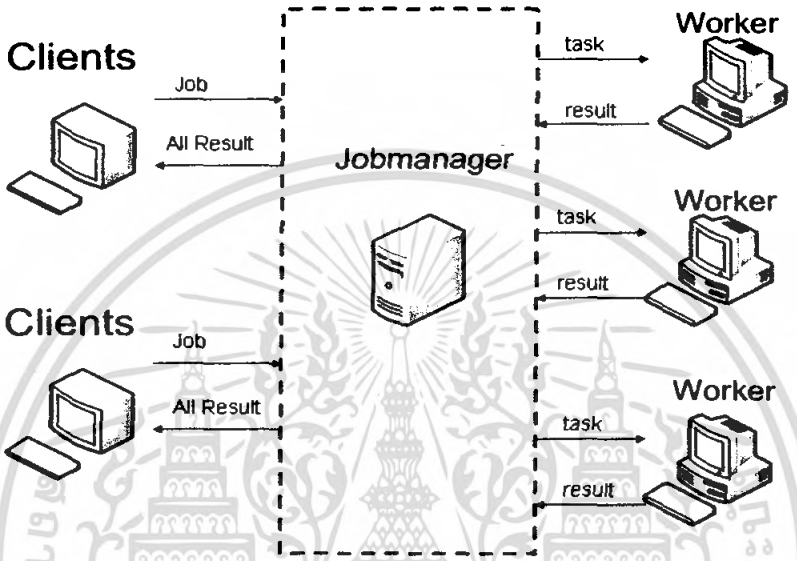
Job คือ ออบเจ็กต์ชนิดหนึ่งใน MATLAB ที่ครอบคลุมงานทั้งหมดของ MDCE หน้าที่ของ job คือการเก็บ jobmanager เอาไว้ และสร้าง task ขึ้น ซึ่งผู้ใช้จะต้องสร้างขึ้นเองในขั้นตอนของการเขียนโปรแกรม และส่งงานไปให้ jobmanager เพื่อทำการประมวลผล

### 4.5.2 Task

Task คือ งานที่ถูกแบ่งออกเป็นส่วนย่อยๆเรียบร้อยแล้วโดยผู้เขียนโปรแกรม task 1 ตัว จะทำงานกับ 1 worker หรือ 1 processor นั่นเอง ถ้าหาก task มีมากกว่าจำนวน worker ก็จะต้องมีการต่อคิวกัน โดย task ที่ถูกสร้างขึ้นมาก่อน จะนำไปทำงานก่อน ส่วนที่เหลือต้องรอจนกว่า task ที่ทำงานอยู่จะทำเสร็จ จึงจะทำงานเป็นลำดับต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานทั้งหมดจะเริ่มขึ้นหลังจากได้รับคำสั่ง submit ก่อนหน้านั้น task ทั้งหมดจะไปเก็บอยู่ในตัว jobmanager อยู่แล้ว จะทยอยส่งไปให้กับ worker ที่เชื่อมต่ออยู่ เมื่อประมวลผลเสร็จ ผลลัพธ์ที่ได้จากการประมวลผล จะส่งกลับมาที่ jobmanager การเก็บผลลัพธ์สามารถเก็บแบบ realtime หรือรอจนงานทั้งหมดเสร็จก่อนแล้วค่อยทำการเก็บผลลัพธ์ทั้งหมดก็ได้ ผลลัพธ์ที่ได้จาก taskแต่ละtaskจะอยู่ในตัวแปร getAllOutputArguments ซึ่งอยู่ภายในออบเจ็กต์ job

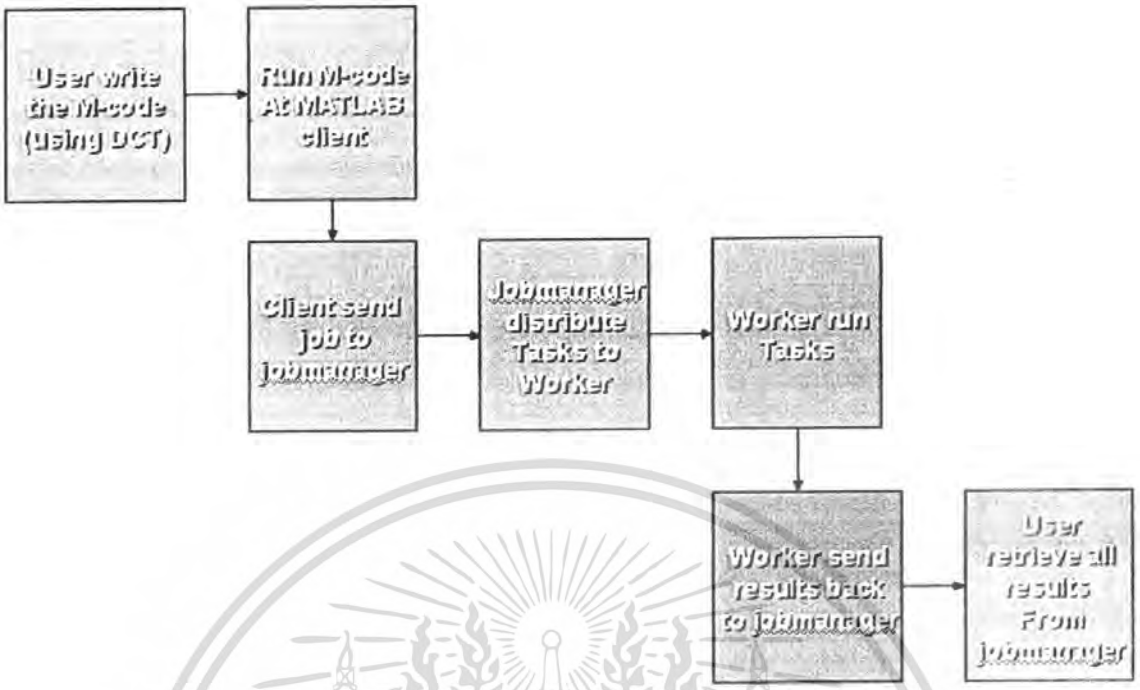



**รูปที่ 4.3 แสดงการทำงานของระบบ MATLAB distributed computing**

## 4.6 ขั้นตอนการใช้งาน MATLAB distributed computing toolbox

เมื่อทราบการทำงานของระบบแล้ว ขั้นตอนต่อไปคือการเข้าใช้งานตัว toolbox ซึ่งประกอบไปด้วยขั้นตอนต่างๆ ต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.4 แผนภาพแสดงขั้นตอนการใช้งาน MATLAB Distributed computing toolbox**

จากรูปที่ 4.4 แสดงให้เห็นถึงขั้นตอนการทำงานของโปรแกรมที่มีการเรียกใช้ Distributed computing toolbox ( ต่อไปจะเรียกว่า DCT ) ผู้ใช้จะต้องเชื่อมต่อเข้ามาในระบบ หรือในเครือข่ายเดียวกันกับเครื่องที่เป็น jobmanager และโปรแกรม MATLAB ที่ใช้ จะต้องเป็นเวอร์ชันเดียวกับตัว Distributed computing engine หากว่าเป็นคนละเวอร์ชันกัน จะไม่สามารถติดต่อกับ jobmanager ได้ การติดต่อกับ jobmanager ใช้คำสั่ง

```
jobManager = findResource('jobmanager');
```

**รูปที่ 4.5 คำสั่งในการติดต่อกับ jobmanager**

เมื่อติดต่อกับ jobmanager ได้แล้ว จะได้รับออบเจ็กต์หนึ่งตัว ทำหน้าที่เป็นตัวแทนของ jobmanager โดยจะมีค่าคุณสมบัติต่างๆของ jobmanager ให้ผู้ใช้สามารถเรียกดูและกำหนดค่าที่ต้องการบางอย่างได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนต่อมาคือการสร้าง job ซึ่งทำหน้าที่เป็นออบเจ็กต์ที่รองรับงานทั้งหมด ผู้ใช้สามารถกำหนดไฟล์หรือไดเรกทอรี่ที่เกี่ยวข้องกับงาน หรือ เก็บผลลัพธ์ที่ได้จากการคำนวณทั้งหมด จากออบเจ็กต์นี้ สามารถสร้างโดยใช้คำสั่ง

```
myjob = createJob(jobManager);
```

**รูปที่ 4.6 คำสั่งที่ใช้ในการสร้าง job**

ต่อมาคือการสร้าง task เพื่อรองรับงานย่อยที่ได้รับการออกแบบหรือกำหนดวิธีการคำนวณในรูปแบบขนานไว้ก่อนแล้ว ทั้งหมด จะถูกเก็บอยู่ใน job เพื่อรอเวลาให้ผู้ใช้เริ่มสั่งให้ทำการประมวลผล

```
taskID(1) = createTask(myjob,@sum,1,{[1,1]});
taskID(2) = createTask(myjob,@sum,1,{[2,2]});
taskID(3) = createTask(myjob,@sum,1,{[3,3]});
```

**รูปที่ 4.7 ตัวอย่างคำสั่งที่ใช้ในการสร้าง task**

จากรูปที่ 4.6 เป็นสร้าง task ที่รองรับฟังก์ชัน sum (เป็นฟังก์ชันพื้นฐานที่มีอยู่ในMATLAB อยู่แล้ว) โดยกำหนดให้แต่ละ task ทำหน้าที่ในการบวกเลขสองจำนวนเข้าด้วยกัน โดยที่มีแต่ละ task จะมี output คืนกลับมาเพียงตัวเดียว ถึงแม้ว่าฟังก์ชันในตัวอย่างจะเป็นฟังก์ชันพื้นฐาน แต่ในการทำงานจริง ก็สามารถนำเอาฟังก์ชันที่เขียนขึ้นมาเองมาใช้ได้ รวมไปถึงข้อมูลที่เกี่ยวของต่างๆ เช่น ไฟล์ข้อความหรือรูปภาพได้อีกด้วย

เมื่อสร้าง task จนครบตามที่ต้องการแล้ว ก็ต้องสั่งให้ job เริ่มทำการประมวลผล โดยใช้คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Submit(myJob);
```

**รูปที่ 4.8 คำสั่งเริ่มการทำงาน**

หลังจากที่เริ่มทำงานไปแล้ว ผู้ใช้มีสองทางเลือกคือรอให้การประมวลผลเสร็จสิ้นลงเสียก่อน แล้วจึงค่อยเรียกดูผลลัพธ์ที่ได้จากการคำนวณทั้งหมด หรือ เรียกดูผลลัพธ์ที่ได้จากการประมวลผลของแต่ละ task แบบ real-time ก็สามารถกระทำได้ โดยใช้คำสั่ง

```
waitForState(myJob);
```

**รูปที่ 4.9 คำสั่งสำหรับสั่งให้รอจนกว่าการประมวลผลจะเสร็จสิ้น**

เมื่อการประมวลผลสิ้นสุดแล้ว สามารถเรียกดูผลลัพธ์ของการประมวลผลได้โดยใช้คำสั่ง

```
result = getAllOutputArguments(myJob);
```

**รูปที่ 4.10 คำสั่งที่ใช้ในการเรียกดูผลลัพธ์ทั้งหมด**

ในเวลานี้ตัวแปร result จะเป็น แมทริกซ์ ขนาด 1x3 ที่เก็บคำตอบที่ได้จากการประมวลผลแต่ละ task ผู้ใช้สามารถนำไปแสดงผลหรือนำไปประมวลผลอย่างอื่นต่อตามแบบฉบับของโปรแกรม MATLAB ได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Result =
     [2]
     [4]
     [6]
```

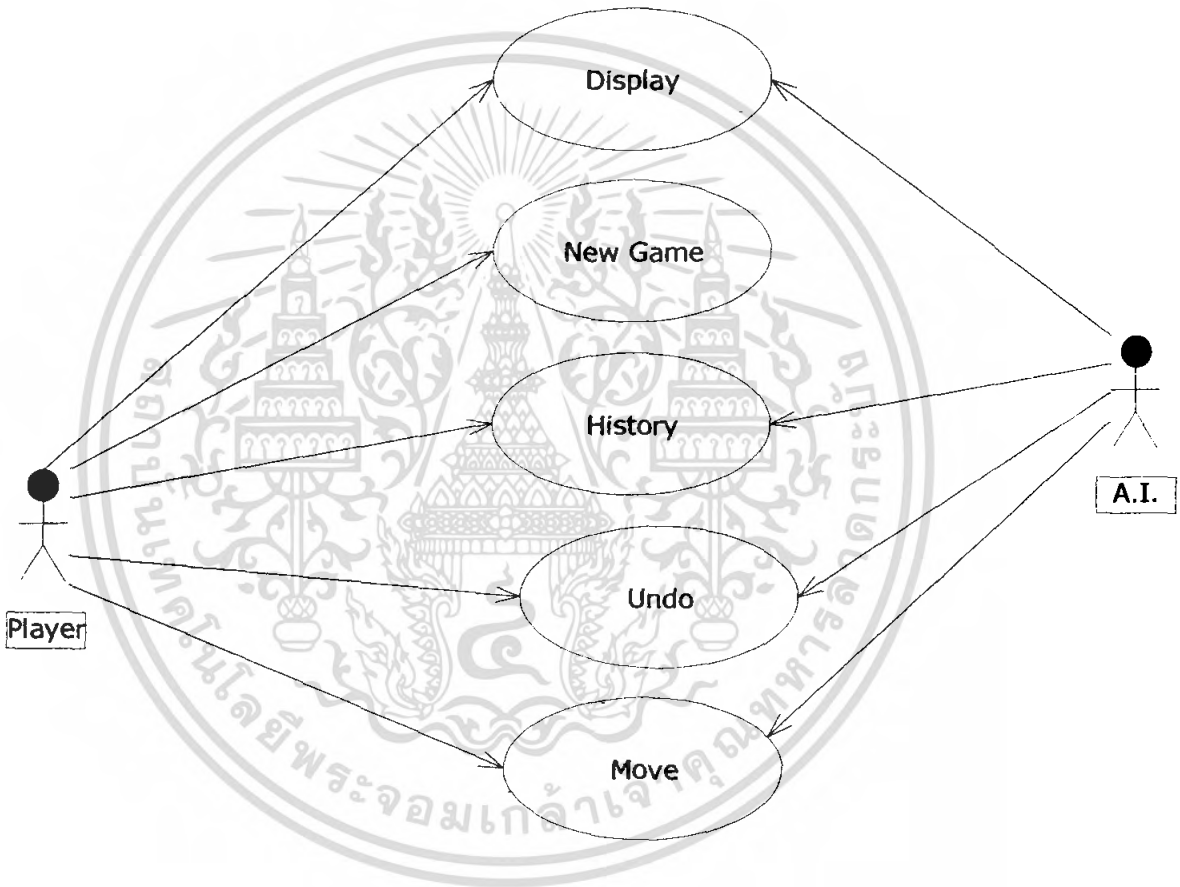
**รูปที่ 4.11 ผมลัพธ์ที่ได้จากการคำนวณ**

จากที่กล่าวมาทั้งหมดนั้นเป็นเพียงการใช้งานพื้นฐานเท่านั้น ตัว toolbox นี้ยังมีรูปแบบการใช้งาน และฟังก์ชั่นที่เกี่ยวข้องอีกมากมาย โดยสามารถศึกษาเพิ่มเติมได้จากเว็บไซต์, เอกสารคู่มือ และ ส่วนของ help ภายในโปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# การออกแบบและพัฒนา

โครงงานนี้ถูกแบ่งออกเป็นสองส่วนหลักๆ คือ การสร้างระบบเกมหมากรุกไทยด้วยเครื่องมือสำหรับสร้าง GUI และการเขียนโปรแกรมเชิงวัตถุในโปรแกรม MATLAB ในขั้นต้นของการพัฒนา ได้ทำการออกแบบ Use-case diagram เอาไว้สำหรับการพัฒนาโปรแกรมในลำดับต่อไปหลังจากนี้




**รูปที่ 5.1 แสดง use-case diagram ของโปรแกรม**

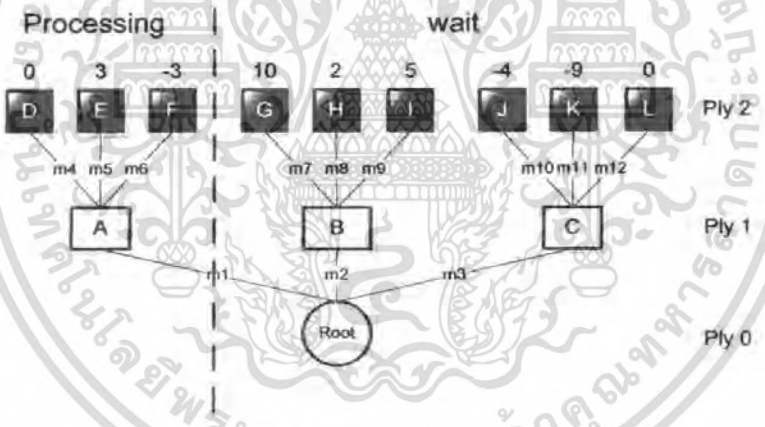
ส่วนที่เป็นหัวใจหลักของโครงการคือการสร้างระบบปัญญาประดิษฐ์ให้กับโปรแกรมหมากรุกไทย จากบทที่สองของรายงาน ได้แสดงปัญหาให้เห็นชัดเจนแล้วว่า ในขั้นตอนการค้นของ minimax algorithm จะเกิดโหนดลูกของ tree จำนวนมาก ตั้งแต่ช่วงต้นเกมไปจนถึงช่วงกลางของเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากจำนวนเส้นทางในการค้นที่มีจำนวนมหาศาลแล้ว ยังมีการใช้ evaluation function ในขั้นตอนสุดท้ายของการค้น เพื่อทำการประเมินค่าให้กับเส้นทางนั้นๆ ใช้สำหรับพิจารณาเลือกเส้นทางของระบบปัญญาประดิษฐ์

วิธีการแก้ปัญหาที่นิยมใช้ในโปรแกรมหมากรุกทั่วไป ก็คือการลดขนาดหรือจำนวนเส้นทางที่ใช้ในการค้นหาลง เรียกว่า alpha beta pruning นำมาใช้ในการพิจารณาตัดเอาเส้นทางที่ไม่เหมาะสมออกไป หรือการลดจำนวนชั้นในการค้นหาลง ตามความเหมาะสมและทรัพยากรที่มีอยู่อย่างจำกัด

การใช้หลักการโปรแกรมแบบขนานร่วมกับเทคโนโลยีกริดคอมพิวติ้ง ก็อาจเป็นทางเลือกหนึ่งในการพัฒนาการเล่นหมากรุกของคอมพิวเตอร์ได้ เมื่อพิจารณาตามหลักการของการออกแบบโปรแกรมแบบขนานแล้ว พบว่า ในขณะที่กำลังประมวลผลเส้นทางใดเส้นทางหนึ่งอยู่นั้น เมื่อย้อนกลับไปดูที่ชั้นบนสุดจะพบว่า เส้นทางที่เหลืออยู่ จะถูกเก็บไว้ในคิวเพื่อรอให้เครื่องคอมพิวเตอร์ประมวลผลเส้นทางของโหนดใดโหนดหนึ่งเสร็จเสียก่อน โหนดหรือเส้นทางที่เหลือจึงจะได้รับการประมวลผล

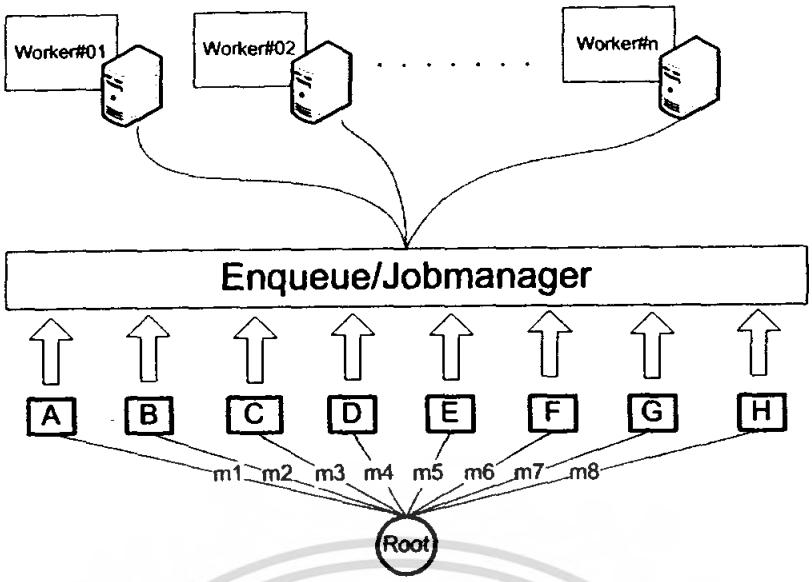



**รูปที่ 5.2 การทำงานแบบปกติ**

ถ้าเป็นในช่วงต้นเกม รูปที่ 5.2 จะมีโหนดที่รออยู่ในคิวจำนวนหนึ่ง ในเครื่องคอมพิวเตอร์ทั่วๆไป โหนดที่เหลืออยู่ก็จะค้างอยู่ในหน่วยความจำ และจำนวนของมันก็จะเพิ่มขึ้นไปเรื่อยๆ จนกว่าการค้นจะสิ้นสุดลง

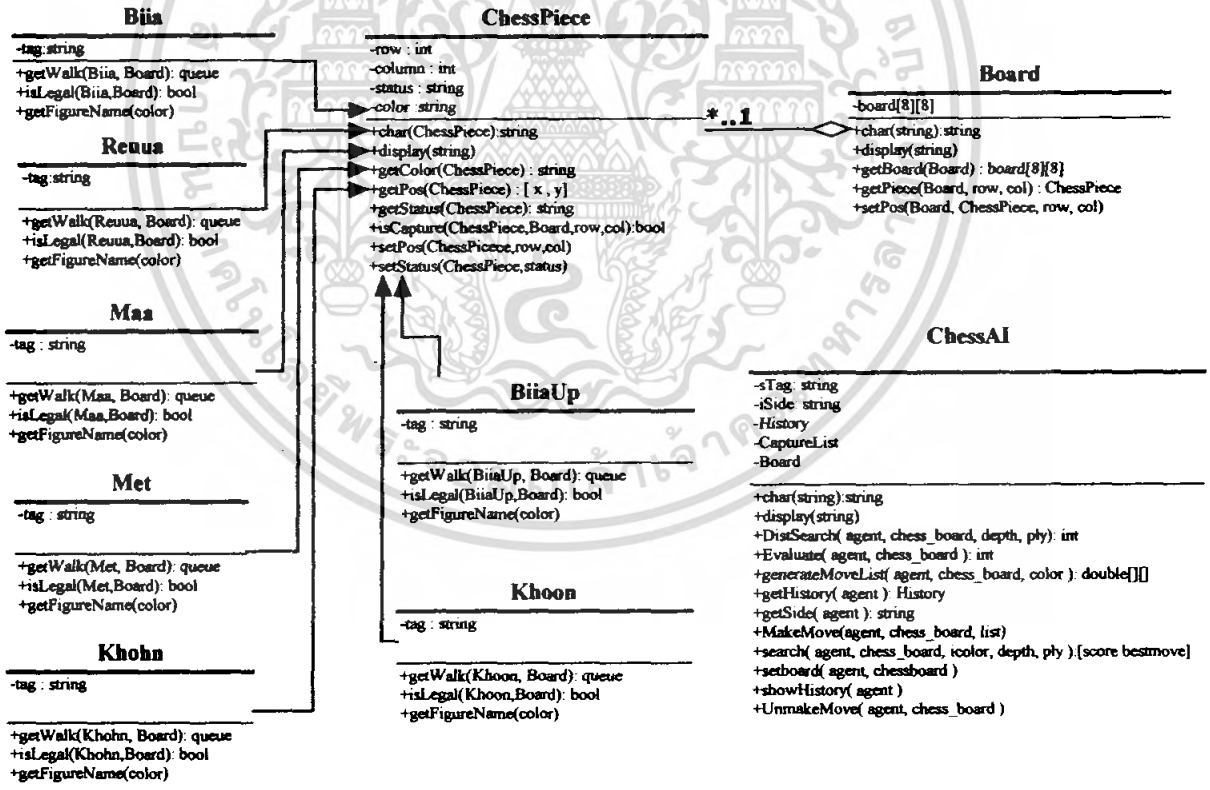
เมื่อสามารถกำหนดให้การค้นในแต่ละเส้นทางสามารถทำไปพร้อมกันได้ ตามจำนวนของ worker ที่มีอยู่ในระบบ นอกจากจะพัฒนาโปรแกรมและระบบปัญญาประดิษฐ์ให้สามารถทำงานได้แล้ว ยังจะต้องทำการทดลองหาจำนวน worker ,วิธีการและการปรับแต่งตัวระบบที่มีความเหมาะสมที่สุดในการแก้ปัญหา ในการดำเนินโครงการลำดับต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 5.3 การทำงานแบบขนาน**

## 5.1 Class diagram




**รูปที่ 5.4 class diagram ของโปรแกรมหมากรุกไทย**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.2 การออกแบบโปรแกรมหมากรุก

5.2.1 ChessPiece เป็น Abstract Class ที่เก็บค่า Attribute พื้นฐานต่างๆของตัวหมาก เพื่อสืบทอดคลาส

Attribute:

- row เป็น attribute ใช้เก็บค่าตำแหน่งแถวของตัวหมาก
- col เป็น attribute ใช้เก็บค่าตำแหน่งคอลัมน์ของตัวหมาก
- status เป็น attribute ใช้เก็บสถาณะของตัวหมากว่ามีชีวิตอยู่หรือถูกกินไปแล้ว
- color เป็น attribute ใช้เก็บค่าสีของตัวหมาก

Method:

- char เป็น Method รับ Object ของ ChessPiece เพื่อดึงค่า attribute ต่างๆ ออกมาแสดงผล
- display เป็น Method รับ ค่าจาก char อีกทีเพื่อแสดงผลออกทางหน้าจอ
- getColor เป็น Method รับ Object ของ ChessPiece เพื่อที่จะ return ค่าสีของตัวหมาก
- getPos เป็น Method รับ Object ของ ChessPiece เพื่อที่จะ return ค่าตำแหน่งของตัวหมาก
- getStatus เป็น Method รับ Object ของ ChessPiece เพื่อที่จะ return ค่าสถานะ
- isCapture เป็น Method รับ Object ของ ChessPiece , Board และตำแหน่ง row กับ col เพื่อทำการเช็คว่าตำแหน่งนี้สามารถกินได้หรือไม่ ถ้าสามารถกินได้ ก็เปลี่ยนสถานะของตัวหมากทันที
- setPos เป็น Method รับ Object ของChessPiece และตำแหน่ง row กับ col เพื่อที่จะตั้งค่าตำแหน่งของตัวหมากใหม่
- setStatus เป็น Method รับ Object ของ ChessPiece และสถานะเป็น string เพื่อที่จะตั้งค่าสถานะของตัวหมากใหม่

5.2.2 Biia เป็นคลาสของตัวเบี้ยซึ่งสืบทอดมากจากคลาส ChessPiece

Attribute:

- tag เป็น attribute ตัวหมากอะไร

Method:

- getWalk เป็น Method รับ Object ของ Biia และ Board เพื่อ return queue ของตาเดินของ Biia
- IsLegal เป็น Method รับ Object ของ Biia , Board และ row กับ col เพื่อเช็คว่าเดินถูกต้องตากฎกติกาหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- getFigureName เป็น Method รับ Object ของ Biia เพื่อ return string ที่เก็บ path ของคลาสนั้น

5.2.3 Khohn เป็นคลาสของตัวโคนซึ่งสืบทอดมากจากคลาส ChessPiece

Attribute:

- tag เป็น attribute ตัวหมากอะไร

Method:

- getWalk เป็น Method รับ Object ของ Khohn และ Board เพื่อ return queue ของตาเดินของ Khohn
- IsLegal เป็น Method รับ Object ของ Khohn , Board และ row กับ col เพื่อเช็คว่าเดินถูกต้องตากฎกติกาหรือไม่
- getFigureName เป็น Method รับ Object ของ Khohn เพื่อ return string ที่เก็บ path ของคลาสนั้น

5.2.4 Maa เป็นคลาสของตัวเบี้ยซึ่งสืบทอดมากจากคลาส ChessPiece

Attribute:

- tag เป็น attribute ตัวหมากอะไร

Method:

- getWalk เป็น Method รับ Object ของ Maa และ Board เพื่อ return queue ของตาเดินของ Maa
- IsLegal เป็น Method รับ Object ของ Maa , Board และ row กับ col เพื่อเช็คว่าเดินถูกต้องตากฎกติกาหรือไม่
- getFigureName เป็น Method รับ Object ของMaa เพื่อ return string ที่เก็บ path ของคลาสนั้น

5.2.5 Reuua เป็นคลาสของตัวเบี้ยซึ่งสืบทอดมากจากคลาส ChessPiece

Attribute:

- tag เป็น attribute ตัวหมากอะไร

Method:

- getWalk เป็น Method รับ Object ของ Reuua และ Board เพื่อ return queue ของตาเดินของ Reuua
- IsLegal เป็น Method รับ Object ของReuua , Board และ row กับ col เพื่อเช็คว่าเดินถูกต้องตากฎกติกาหรือไม่
- getFigureName เป็น Method รับ Object ของ Reuua เพื่อ return string ที่เก็บ path ของคลาสนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.6 Met เป็นคลาสของตัวเบี้ยซึ่งสืบทอดมากจากคลาส ChessPiece

Attribute:

- tag เป็น attribute ตัวหมากอะไร

Method:

- getWalk เป็น Method รับ Object ของ Met และ Board เพื่อ return queue ของตาเดินของ Met
- IsLegal เป็น Method รับ Object ของ Met , Board และ row กับ col เพื่อเช็คว่าเดินถูกต้องตากฎกติกาหรือไม่
- getFigureName เป็น Method รับ Object ของ Met เพื่อ return string ที่เก็บ path ของคลาสนั้น

5.2.7 Khoon เป็นคลาสของตัวเบี้ยซึ่งสืบทอดมากจากคลาส ChessPiece

Attribute:

- tag เป็น attribute ตัวหมากอะไร

Method:

- getWalk เป็น Method รับ Object ของ Khoon และ Board เพื่อ return queue ของตาเดินของ Khoon
- IsLegal เป็น Method รับ Object ของ Khoon , Board และ row กับ col เพื่อเช็คว่าเดินถูกต้องตากฎกติกาหรือไม่
- getFigureName เป็น Method รับ Object ของ Khoon เพื่อ return string ที่เก็บ path ของคลาสนั้น

5.2.8 BiiaUp เป็นคลาสของตัวเบี้ยซึ่งสืบทอดมากจากคลาส ChessPiece

Attribute:

- tag เป็น attribute ตัวหมากอะไร

Method:

- getWalk เป็น Method รับ Object ของ BiiaUp และ Board เพื่อ return queue ของตาเดินของ Biia
- IsLegal เป็น Method รับ Object ของ BiiaUp, Board และ row กับ col เพื่อเช็คว่าเดินถูกต้องตากฎกติกาหรือไม่
- getFigureName เป็น Method รับ Object ของ BiiaUp เพื่อ return string ที่เก็บ path ของคลาสนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.9 Board เป็นคลาสของตัวกระดานหมกรุก

Attribute:

- board เป็น cell Array 8x8 ไว้เก็บตัวหมากแต่ละช่องแทนกระดาน

Method:

- display เป็น Method ไว้แสดงค่าตัวกระดานหรือ Array 8x8 ที่ได้กำหนดไว้
- getBoard เป็น Method ที่จะ return ค่า Board กลับมา
- getPiece เป็น Method ที่จะ return ค่าคลาสตัวหมากกลับมาเมื่อข้างในนั้นมีหมากอยู่
- setPos เป็น Method ที่ไว้สำหรับตั้งค่าตัวหมากแต่ละตัวลงไปในกระดาน

5.2.10 ChessAI เป็นคลาสของระบบปัญญาประดิษฐ์ของตัวเกม ทำหน้าที่ในการค้นหาตาเดินของฝ่ายตรงข้ามกับผู้เล่น

Attribute:

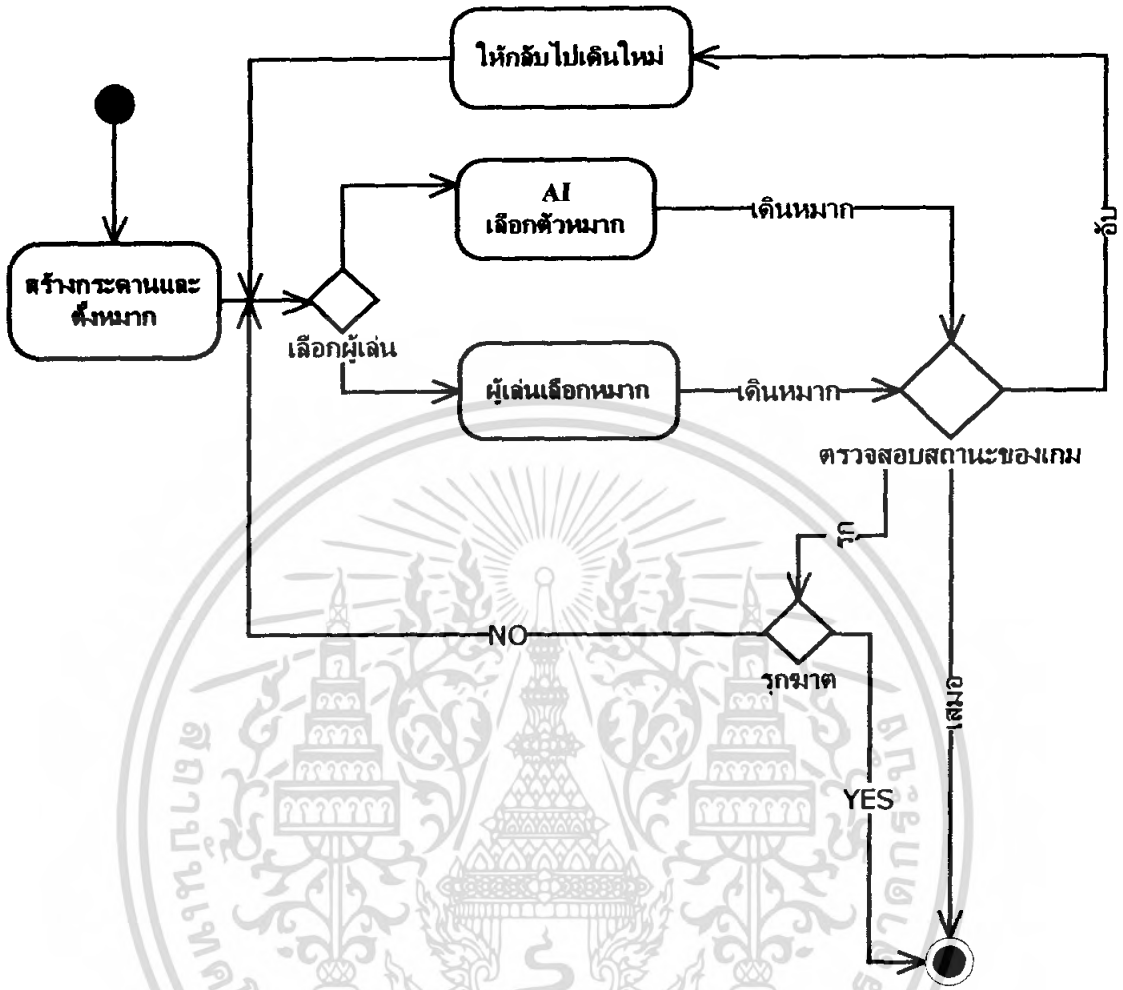
- iSide ใช้สำหรับเก็บข้อความสำหรับแสดงฝ่ายของ A.I.
- History ใช้ในขั้นตอนการ search สำหรับเก็บตาเดินที่ได้มีการทดลองเดินไปแล้ว และเพื่อให้สามารถยกเลิกการเดินหลังจากการ search เสร็จสิ้นได้
- CapturedList เป็น cell array สำหรับเก็บ Object ของตัวหมากที่ถูกกินในขณะที่มีการทดลองเดิน ในขั้นตอนของโปรแกรมจะถูกใช้งานในลักษณะ stack

Method:

- Evaluate ใช้สำหรับการประเมินค่าสำหรับการให้คะแนนในการ search
- generateMoveList ใช้สำหรับสร้างรายการตาเดินของตัวหมากทุกตัวที่สามารถเป็นไปได้ รวมไปถึงตากินของตัวหมากแต่ละตัวด้วย
- makemove ใช้สำหรับการทดลองเดินตัวหมาก ใช้ควบคู่กับ unmakemove
- unmakemove ใช้สำหรับการยกเลิกการเดินจากเมธอด makemove
- search เมธอดหลัก ของ A.I. ใช้หลักการ Minimax search โดยกำหนด ฝ่าย และความระดับความลึกตามความเหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3 Activity Diagram



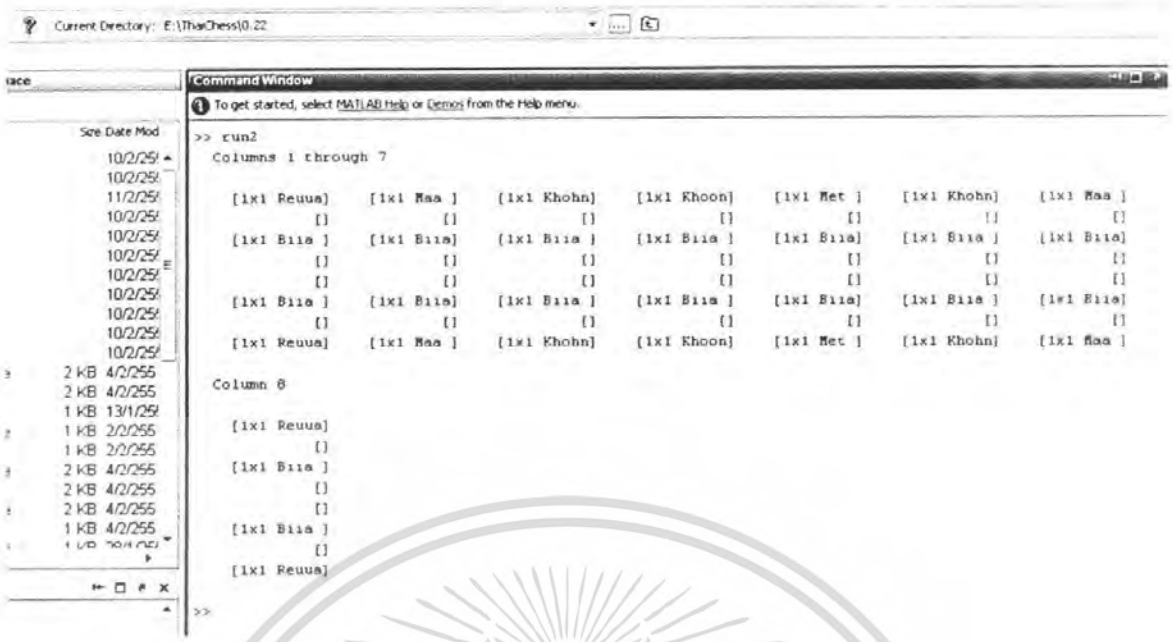

**รูปที่ 5.5 Activity Diagram ของโปรแกรมหมากรุก**

## 5.4 การใช้งานโปรแกรม

#### 5.4.1 เริ่มต้นสร้างตัวหมากและกระดาน

เมื่อเข้าโปรแกรม MATLAB แล้วให้พิมพ์คำว่า run2 เพื่อเป็นการสร้างตัวหมาก และติดตั้งหัวหมากลงไปใน Array 8x8

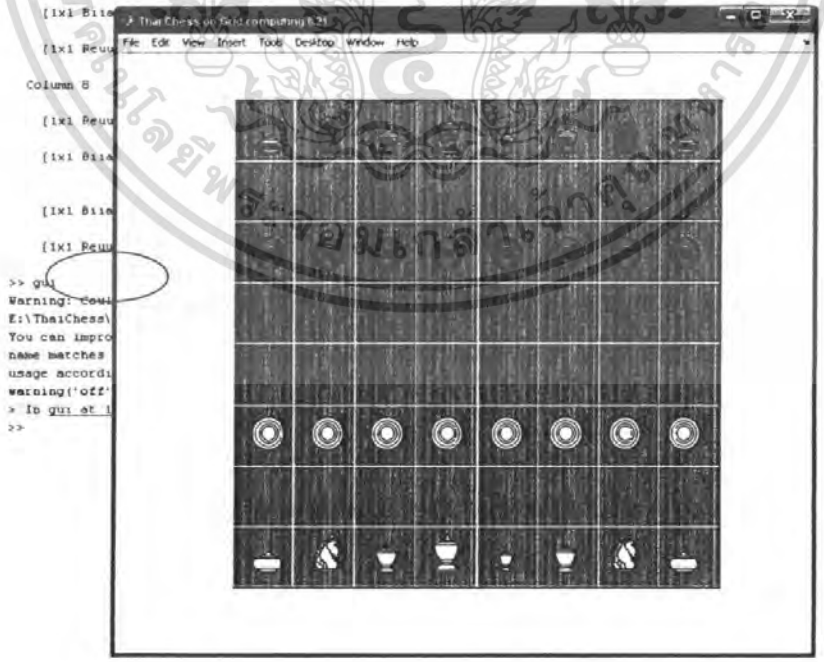
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 5.6 แสดงภาพเมื่อพิมพ์คำว่า run2 เพื่อสร้างตัวหมาและกระดาน**

### 5.4.2 แสดงผล GUI

เมื่อทำการสร้างตารางและตัวหมากเรียบร้อยแล้วให้พิมพ์ gui เพื่อเป็นการสร้างตัว GUI(Graphic Unit Interface) ของโปรแกรมหมากรุกขึ้นมา

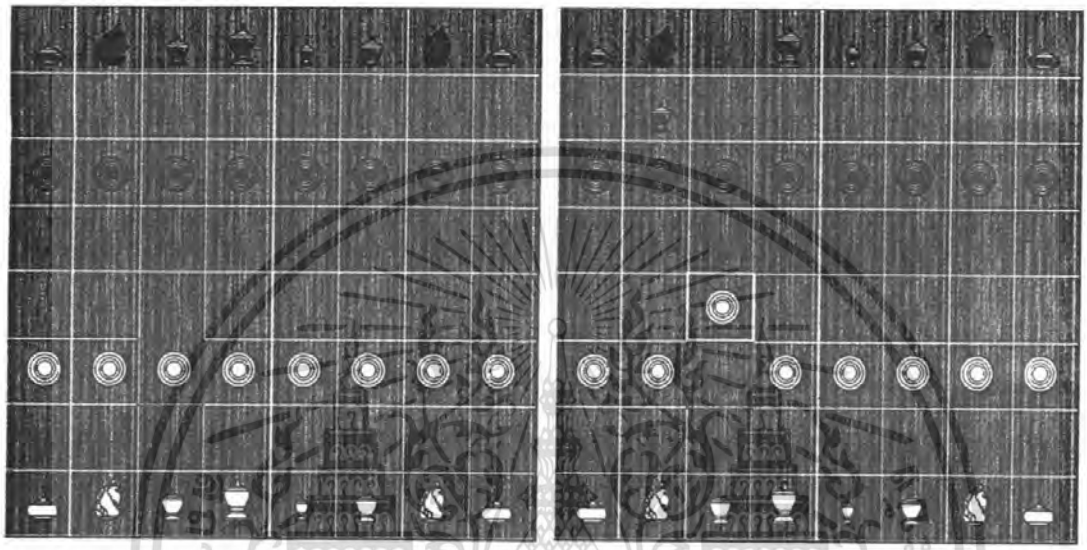



**รูปที่ 5.7 แสดงภาพเมื่อพิมพ์คำว่า gui ก็จะแสดง GUI ของตัวโปรแกรมหมากรุกขึ้นมา**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.4.3 เริ่มเดินหมาก

เมื่อทำการสร้าง GUI เรียบร้อยแล้วก็จะเริ่มเดินหมาก โดย click เลือกที่ตัวหมากที่ต้องการจะเดิน จากนั้นให้เลือกตำแหน่งที่จะเดินไป จากนั้นตัวหมากที่เราเลือกไว้จะไปอยู่ที่ตำแหน่งที่เราเลือก ถ้าทำการเลือกตาเดินที่ผิดกฎกติกา ตัวฟังก์ชันจะทำการแจ้งเตือนใน Console จะบอกมาว่า can't move แล้วให้เราทำการเลือกหมากตัวใหม่



**รูปที่ 5.8 เลือกเดินเบี้ยที่ตำแหน่ง(6,3) ไปยังตำแหน่ง (5,3)**

### 5.4.4 เช็คเงื่อนไข

เมื่อเดินเรียบร้อยแล้วก็จะเข้าสู่เงื่อนไขในการเช็คว่าตรงตามเงื่อนไขใด เช่น เมื่อมีการรุกก็จะเช็คว่ารุกฆาตไหมถ้าใช่ก็จะแสดง Message Box ขึ้นมาเพื่อบอกว่าแพ้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 6

# การทดลอง

## 6.1 ทดลอง Search ตาเดิน

ทำการทดลองวางหมากเพื่อให้ A.I. ทำการ Search ตาเดินโดยใช้ Algorithm Minimax Search โดยทำการเลือกใช้จำนวนคอมพิวเตอร์ และระดับความลึกของ Tree โดยการทดสอบ จะวัดโดยช่วงที่มีจำนวนตาเดินมากที่สุด ซึ่งก็คือตอนเริ่มเกม แต่ละฝ่ายจะมีจำนวนตาเดินถึง 23 ตัว เมื่อรวมทั้งสองฝั่งเข้าด้วยกันก็จะกลายเป็น 46 ตาเดินในระดับความลึกที่ 1 และจะเพิ่มมากขึ้นในความลึกระดับถัดไป

การ A.I. นั้นตัดสินใจเลือกเส้นทางนั้นเกิดจากการเอาผลลัพธ์จากฟังก์ชัน generateMoveList ซึ่งจะ return เส้นทาง (path) และคะแนน (score) ของแต่ละเส้นทางซึ่ง A.I. นั้นจะเลือกจากค่าที่มากที่สุดจากเส้นทางที่ได้มาทั้งหมด ในการทดลองนี้เราได้ทดลองที่ความลึกของ Tree ที่ระดับ 4 ซึ่งคิดตาแรกที่เริ่มเดิน

ทดลองประมวลผลแต่ละเครื่องโดยใช้ Algorithm ที่เหมือนกันโดยผ่าน JOBMANAGER ในการจัดสรรงานให้เครื่องคอมพิวเตอร์ โดยระดับความลึกในตาราง คือ ความลึกของ tree ณ จุดที่มีการเรียกใช้ฟังก์ชั่นประเมินค่า กำหนดให้ จุดที่เริ่มเดินคือ Root ของ tree ระดับที่ 1 ก็คือความลึกระดับที่ 1 ของ tree หรือตาเดินที่สีขาวเลือกที่จะเดินหมากตอบโต้สีดำอย่างไร ระดับที่ 2 จะประเมินที่ ระดับที่2 ก็คือตาเดินของสีดำหลังจากที่สีขาวเดินไปแล้ว ซึ่งจะสลับกันไปอย่างนี้จนถึงระดับที่กำหนด

กำหนดให้มีคอมพิวเตอร์ 1 เครื่องในการประมวลผล

**ตารางที่ 6.1 ผลการทดลองกับ 1 worker**

| ระดับความลึก | เวลา |
|---|---|
| 1 | 13.1879 |
| 2 | 143.57283 |
| 3 | 3102.6941 |
| 4 | 60358.9543 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้มีคอมพิวเตอร์ 2 เครื่องในการประมวลผล

**ตารางที่ 6.2 ผลการทดลองกับ 2 workers**

| ระดับความลึก | เวลา(วินาที) |
|---|---|
| 1 | 7.6893 |
| 2 | 73.7451 |
| 3 | 1596.7011 |
| 4 | 36039.0627 |

กำหนดให้มีคอมพิวเตอร์ 3 เครื่องในการประมวลผล

**ตารางที่ 6.3 ผลการทดลองกับ 3 workers**

| ระดับความลึก | เวลา(วินาที) |
|---|---|
| 1 | 5.3784 |
| 2 | 50.1399 |
| 3 | 1058.8602 |
| 4 | 23763.4463 |

กำหนดให้มีคอมพิวเตอร์ 4 เครื่องในการประมวลผล

**ตารางที่ 6.4 ผลการทดลองกับ 4 workers**

| ระดับความลึก | เวลา(วินาที) |
|---|---|
| 1 | 4.48775 |
| 2 | 40.0289 |
| 3 | 872.46863 |
| 4 | 19594.3478 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้มีคอมพิวเตอร์ 5 เครื่องในการประมวลผล

**ตารางที่ 6.5 ผลการทดลองกับ 5 workers**

| ระดับความลึก | เวลา(วินาที) |
|---|---|
| 1 | 3.84495 |
| 2 | 31.88895 |
| 3 | 675.13855 |
| 4 | 15234.7568 |

กำหนดให้มีคอมพิวเตอร์ 6 เครื่องในการประมวลผล

**ตารางที่ 6.6 ผลการทดลองกับ 6 workers**

| ระดับความลึก | เวลา(วินาที) |
|---|---|
| 1 | 3.44475 |
| 2 | 27.5807 |
| 3 | 602.39813 |
| 4 | 13527.3755 |

กำหนดให้มีคอมพิวเตอร์ 7 เครื่องในการประมวลผล

**ตารางที่ 6.7 ผลการทดลองกับ 7 workers**

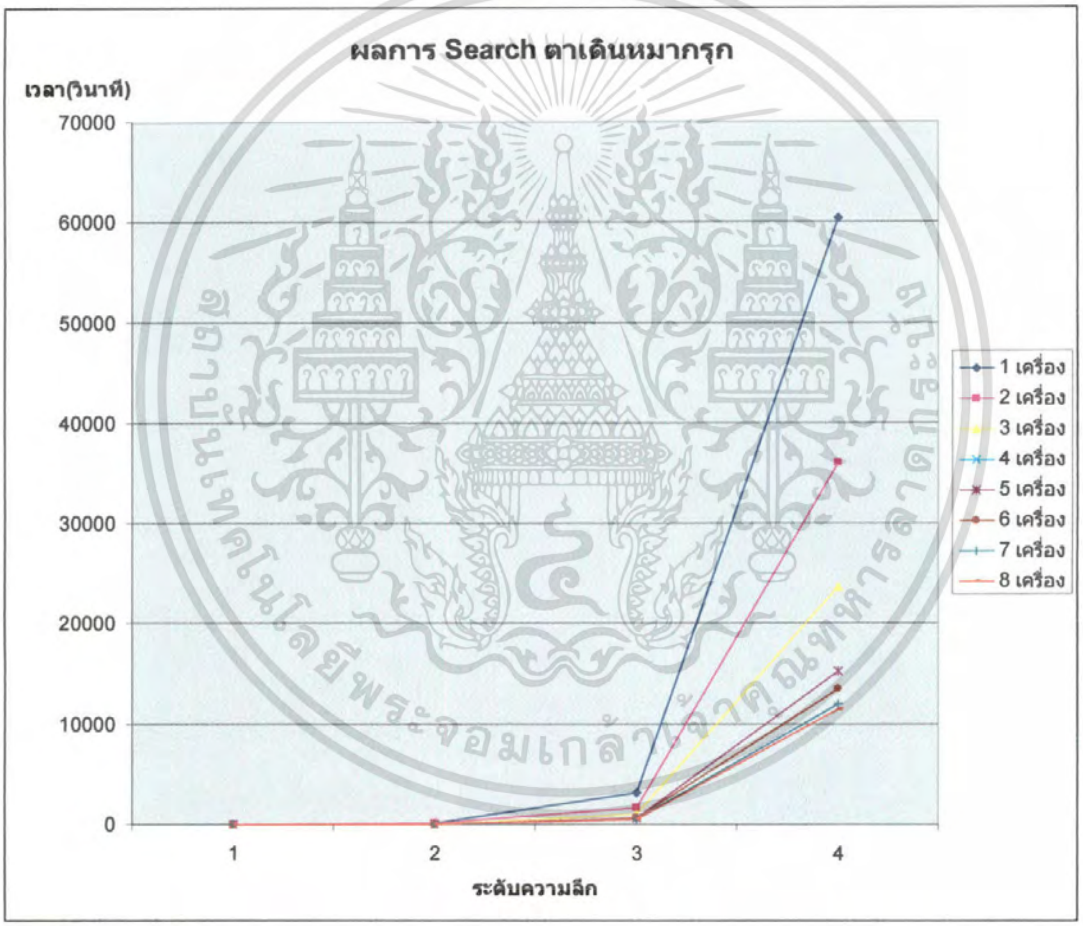
| ระดับความลึก | เวลา(วินาที) |
|---|---|
| 1 | 3.454675 |
| 2 | 25.69506 |
| 3 | 533.3653 |
| 4 | 12011.368 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้มีคอมพิวเตอร์ 8 เครื่องในการประมวลผล

**ตารางที่ 6.8 ผลการทดลองกับ 8 workers**

| ระดับความลึก | เวลา(วินาที) |
|---|---|
| 1 | 2.990325 |
| 2 | 36.1051 |
| 3 | 508.004083 |
| 4 | 11435.1186 |




**รูปที่ 6.1 แผนภูมิเปรียบเทียบเวลากับคอมพิวเตอร์ที่ใช้ในการ Search ในแต่ละ ระดับความลึก**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.2 ทดลองประมวลผลกับเครื่องเดียว โดยไม่ผ่าน JOBMANAGER

**ตารางที่ 6.9 ผลการทดลองกับ 1 เครื่องโดยไม่ผ่าน Distributed Tool box**

| ระดับความลึก | เวลา(วินาที) |
|---|---|
| 1 | 0.6638 |
| 2 | 7.9806 |
| 3 | 236.7148 |
| 4 | 2877.9082 |

## 6.2 วิเคราะห์ผลการทดลอง

จากผลการทดลอง เมื่อเปรียบเทียบความเร็วในการทำงานระหว่างการทำงานแบบ Single machine กับการทำงานผ่าน Distributed Computing Engine ในระดับความลึกขั้นแรกๆ การทำงานในแบบหลังจะใช้เวลานานกว่าค่อนข้างมาก แต่ในการทดลองครั้งต่อๆมา ความต่างของเวลาที่ใช้จะค่อยๆลดลง เมื่อ ระดับความลึกของการ search เพิ่มมากขึ้น และคาดว่าความต่างของเวลาจะใกล้เคียงกันมากขึ้นและทำงานได้เร็วกว่า เมื่อเพิ่มระดับความลึกมากยิ่งขึ้นไปอีก

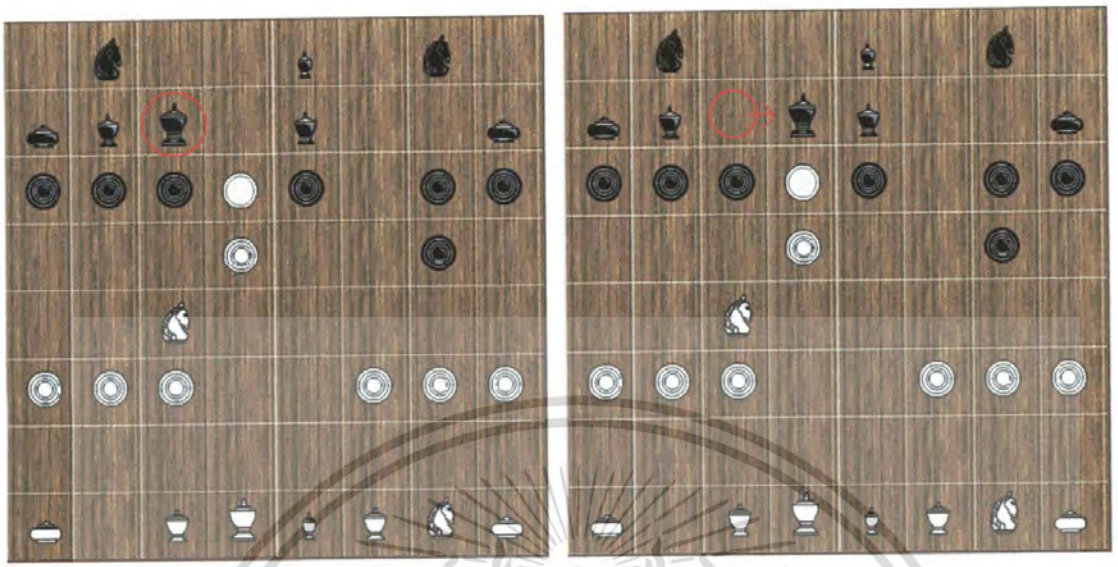
## 6.3 ทำการทดสอบในการเดินหมากกับ AI

เมื่อขุนสีดำไม่สามารถหนีได้ต่อถือว่าเป็นการรุกฆาต AI ก็จะส่งข้อความบอกเรายอมแพ้แล้วเป็นการสิ้นสุดเกม

**รูปที่ 6.2 Message box ของโปรแกรมหมากรุก เมื่อขุนของสีดำโดนรุกฆาต**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีหมากสีขาวทำการรุก ขุนสีดำ ตัว AI มีการขยับถอยเพื่อหนีออกจากตากินฝ่ายตรงข้าม



**รูปที่ 6.3 การหนีของ ขุน เมื่อมีการรุก**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 7
# บทวิจารณ์และสรุป

## 7.1 สรุปและวิเคราะห์ผลการทดลอง

จากผลการทดลอง ระบบกริดคอมพิวติ้ง ก็ยังมีจุดด้อยอยู่ที่หน่วยประมวลผลแต่ละตัวจะต้องส่งข้อมูลหากันผ่านระบบเครือข่ายเท่านั้น ทำให้ความเร็วในการติดต่อกันทำได้ค่อนข้างช้าเมื่อเทียบกับความเร็วสัญญาณบัสในเครื่องคอมพิวเตอร์แบบหลายหน่วยประมวลผล ด้วยเหตุนี้ เมื่อจำนวนของโพรเซสมีมากขึ้น เวลาที่เครื่องคอมพิวเตอร์ใช้ในการติดต่อกันก็มีมากขึ้นตามไปด้วย สรุปได้ว่า กริดคอมพิวติ้ง ไม่ได้ขึ้นอยู่กับจำนวนเครื่องคอมพิวเตอร์ที่มีเพียงอย่างเดียว แต่ต้องหาจุดที่เหมาะสมที่สุด ระหว่าง จำนวนเครื่องคอมพิวเตอร์ จำนวนโพรเซส และ ขนาดของงาน

นอกจากเวลาที่ใช้ในการติดต่อกันแล้ว ยังมีเรื่องของการเขียนโปรแกรมแบบออบเจค เพื่อนำมาประยุกต์ใช้กับระบบ ในการทำงานนั้น ผู้ใช้งาน อาจสร้างออบเจคขึ้นมาบนเครื่อง เมื่อต้องการส่งโปรแกรมนั้นไปปฏิบัติงานบนเครื่องอื่น จะพบว่าเครื่องคอมพิวเตอร์ปลายทาง ไม่รู้จักออบเจคนั้น ปัญหานี้แก้ได้โดยการแปลงออบเจคนั้นให้อยู่ในรูปแบบที่ตัวภาษา MATLAB สามารถเข้าใจได้ แล้วทำการสร้างวัตถุขึ้นมาใหม่ที่เครื่องปลายทาง แล้วค่อยแปลงข้อมูลที่ส่งมากลับมาเป็นออบเจคที่ส่งมาจริงๆอีกครั้ง กระบวนการนี้ทำให้เสียเวลาไปพอสมควรในการประมวลผลงานย่อยของระบบ

จากผลการทดลอง พบว่าเวลาที่ใช้ในการประมวลผลแต่ละรอบจะค่อยๆลดลง แต่จะลดลงไม่มากนักเมื่อมีจำนวนคอมพิวเตอร์ 5 เครื่องขึ้นไป โดยจะใช้เวลาในการประมวลผลใกล้เคียงกันมากๆ แม้จะลดลงไม่มากในความลึกชั้นต้นๆ แต่สามารถลดลงไปได้พอสมควรเมื่อกำหนดความลึกเป็นสามและสี่ชั้นขึ้นไป

อนึ่ง โครงงานนี้เป็นการนำเอาทฤษฎีเบื้องต้นของการสร้างเกมหมากรุกเท่านั้น แม้การ Search ทำได้เร็วขึ้น แต่ในด้านของการตอบสนองกับผู้ใช้ยังไม่ดีเท่าไหร่นัก ทำให้ผู้เล่นสามารถเอาชนะเกมได้อย่างง่ายดาย เนื่องจากการประเมินสถานการณ์ด้วยการใช้ dynamic evaluation เพียงอย่างเดียวนั้น ยังไม่มีประสิทธิภาพเพียงพอ ยังคงต้องมีการปรับปรุงต่อไปสำหรับการประมวลผลในส่วนนี้

จากที่กล่าวมาทั้งหมด ระบบกริดหรือคลัสเตอร์ริ่ง สามารถลดเวลาในการประมวลผลของ A.I. ได้จริง และสามารถนำไปประยุกต์ใช้งานกับงานด้านอื่นๆที่เกี่ยวข้องได้ในอนาคต แต่พบปัญหามากมายในการทำงานของ A.I. เนื่องด้วยเกมหมากรุกเป็นเกมที่ค่อนข้างอิสระในการเดิน อีกทั้งมีกลยุทธ์ที่ซับซ้อน เช่น “การผูกหมาก” เป็นวิธีการโดยนำหมากในกระดานหนึ่งตัวไปไว้ที่ตำแหน่งตากินของหมากฝ่ายเดียวกัน ซึ่งส่วนนี้เองอาจจะทำให้ค่าคะแนนที่ได้มาอาจจะผิดพลาดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือไม่อาจมีการวางกลยุทธ์ที่ซึ่งยากแก่การคาดเดาของ A.I. อาจบอกได้ว่า Minimax Search อาจไม่ใช่สิ่งดีที่สุดใน เกมหมากรุก

## 7.2 แนวทางในการพัฒนาต่อ

จากการทดลองปฏิบัติงานและปัญหาที่พบในการดำเนินงาน สามารถสรุปเป็นแนวทางในการพัฒนาต่อได้ ดังนี้

1. ปรับปรุง Evaluate function ให้มีความสามารถมากยิ่งขึ้น
2. เพิ่ม Knowledge base หรือ Expert system ให้กับระบบ
3. เสริมความเร็วในการ Search ด้วยเทคนิคใหม่ๆ เช่น Genetic algorithm
4. เพิ่มระบบ Machine learning
5. เพิ่มความสามารถในการตัดสินใจด้วย Fuzzy logic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม

รศ.ดร. มนัส สังวรศิลป์, วรรัตน์ ภัทรอมรกุล. 2546. **คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์.** สำนักพิมพ์ อินโฟเพรส. กรุงเทพ.

ผศ.ชนินทร์ วิเชียรสวรรค์. 2547. **ปัญญาประดิษฐ์.** สำนักพิมพ์มหาวิทยาลัยรามคำแหง.

โศรฎา แข็งการ, กนต์ธร ชำนิประศาสน์. **การใช้ MATLAB เพื่องานทางวิศวกรรม.** มหาวิทยาลัยเทคโนโลยีสุรนารี.

Resource Working Group. **Grid Technology and Applications**. [Online].Available: http://rswg.thaigrid.net/book/export/html/75

Thailand Board games online. **Thai Chess**. Thai Checker and etc. [Online].Available: www.thaibg.com/

วิชาการ.คอม. VChakarn.com. **Grid computing**. [Online].Available: http://www.vcharkarn.com/include/article/showarticle.php?aid=16300

Wikipedia. **หมากรุกไทย.** [Online].Available: th.wikipedia.org/wiki/หมากรุกไทย.

Franois Dominic Larame. **Chess Programming part I-IV**. [Online].Available: http://www.gamedev.net/reference/articles/article1014.asp

TheMathWork. **Distributed Computing Toolbox for use with MATLAB.** [Online].Available:http://www.mathworks.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

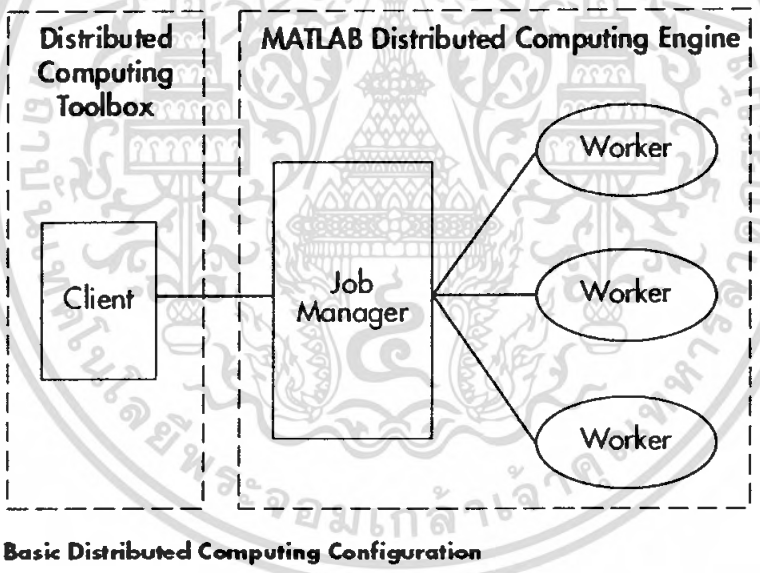
# ภาคผนวก ก

# MATLAB Distributed Computing Toolbox และ Engine

Distributed Computing คือการนำเครื่องคอมพิวเตอร์หลายๆเครื่อง มาช่วยกันประมวลผลในงานที่มีขนาดใหญ่มากๆ โดยการแบ่งงานออกเป็นส่วนย่อยๆ เรียกว่า task โดยผู้ใช้นั้นเป็นผู้กำหนดการแบ่งเอง และแจกจ่าย Tasks ที่แบ่งออกมานั้นให้เครื่องคอมพิวเตอร์แต่ละเครื่องประมวลผลเฉพาะส่วนที่ได้รับ ซึ่งช่วยให้การทำงานเพิ่มรวดเร็วขึ้น

ภายในเครือข่าย จะมีเครื่องหนึ่งทำหน้าที่เป็น Job manager ซึ่งเป็นตัวกลางระหว่างclient และเครื่องคอมพิวเตอร์เครื่องอื่นๆในเครือข่าย โดย jobmanager จะทำหน้าที่ในการแจกจ่าย task ออกไปให้กับคอมพิวเตอร์เครื่องอื่นๆในเครือข่าย เครื่องที่ทำหน้าที่ในการประมวลผล tasks ที่ได้รับ เรียกว่า worker ซึ่งผู้ใช้ไม่สามารถสั่งงานได้โดยตรง

ส่วนใหญ่แล้ว Client และ Jobmanager มักจะถูกติดตั้งอยู่บนเครื่องเดียวกัน




**รูปที่ ก.1 ภาพรวมของระบบ MATLAB Distributed Computing**

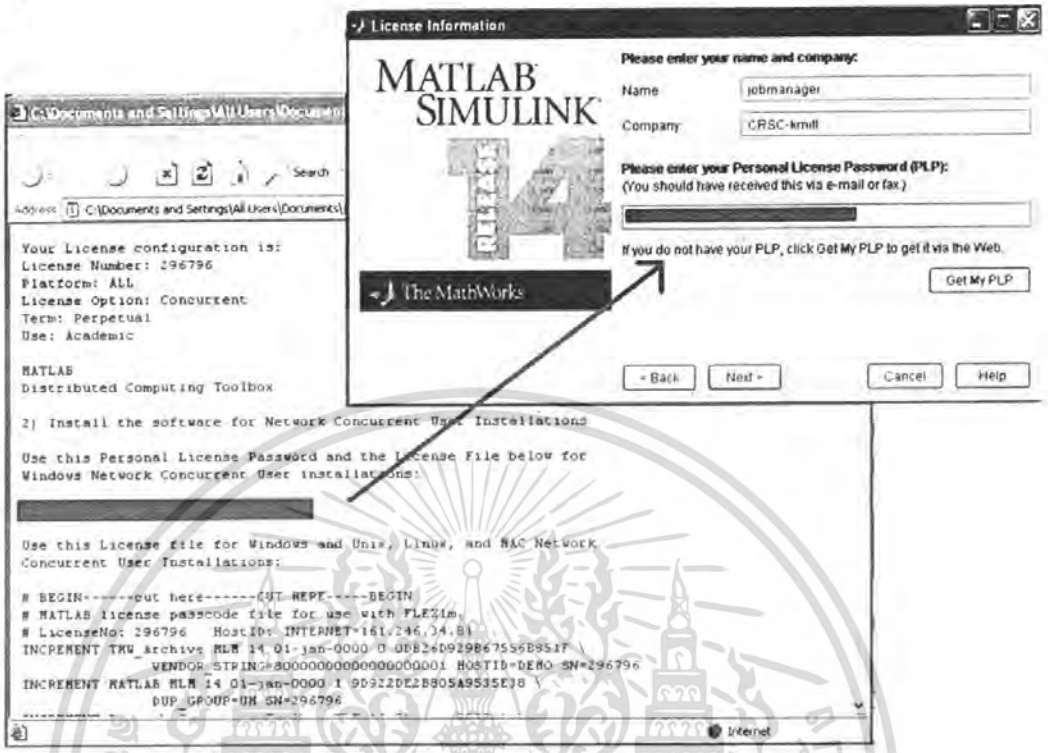
**การติดตั้ง Distributed Computing Toolbox**

Distributed computing toolbox เป็นส่วนหนึ่งของโปรแกรม Matlab สำหรับติดตั้งลงบนเครื่องที่เป็น client ( และอาจเป็น Job manager ได้ภายในเครื่องเดียวกัน ) เมื่อมีการเรียกใช้งานในส่วนของ Distributed computing Matlab จะเรียกใช้งานในส่วนนี้ เพื่อติดต่อกับ Distributed computing engine

การติดตั้งจะแบ่งออกเป็น 2 ส่วนคือ Toolbox และ Engine ซึ่งการติดตั้ง Engine จะอธิบายในลำดับต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

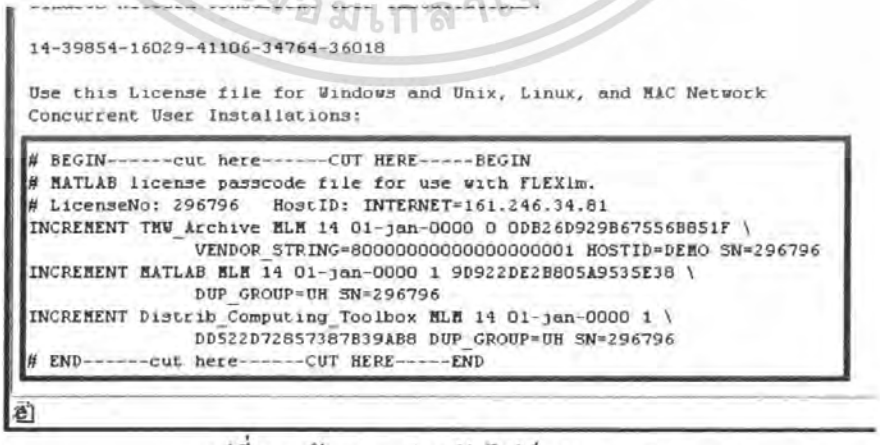
เมื่อเริ่มขั้นตอนการติดตั้ง โปรแกรมติดตั้งจะถามหา PLP ( Personal License Password ) ใส่รหัสดังกล่าวลงไป




**รูปที่ ก.2 การกรอก PLP**

Note : การติดตั้ง toolbox และ engine กำหนดโดยการใส่ PLP ที่แตกต่างกัน ซึ่ง PLP ของ engine จะมีขนาดยาวกว่า toolbox

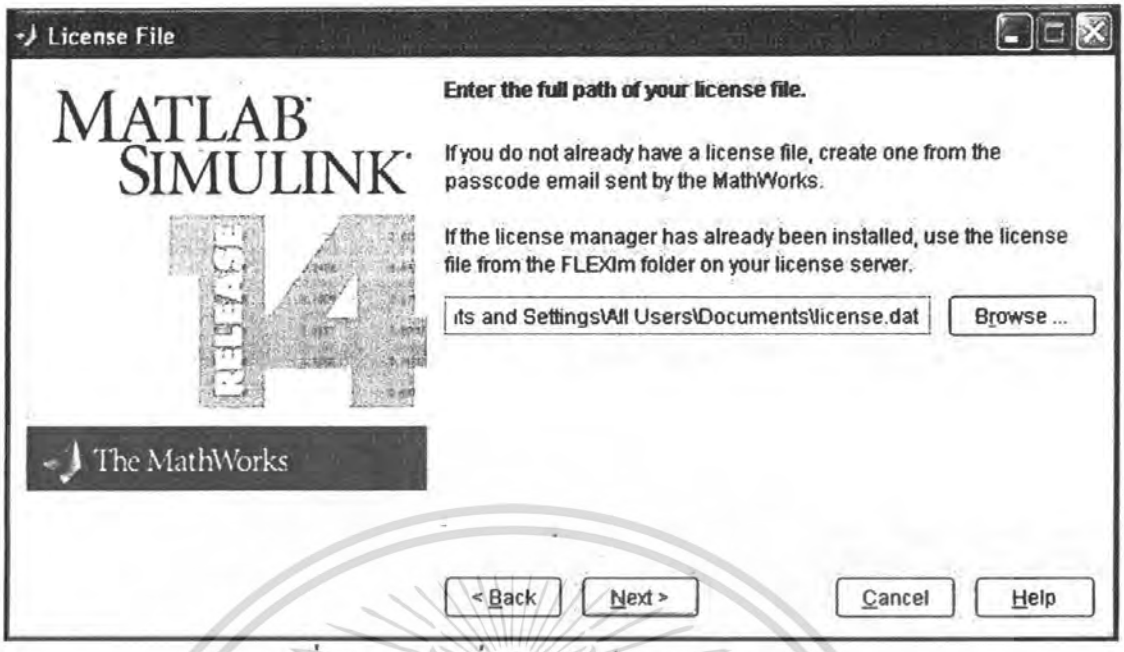
ขั้นตอนต่อมา โปรแกรมติดตั้งจะถามถึงที่อยู่ของ license file ให้สร้างไฟล์ใหม่ขึ้นมาโดยนำข้อความในกรอบสีแดงไปสร้างเป็นไฟล์ใหม่




**รูปที่ ก.3 ข้อความสำหรับไฟล์ license.dat**

หลังจากนั้นกำหนด path ที่อยู่ของไฟล์ใหม่ที่สร้างขึ้น

เอกสารนี้มีไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

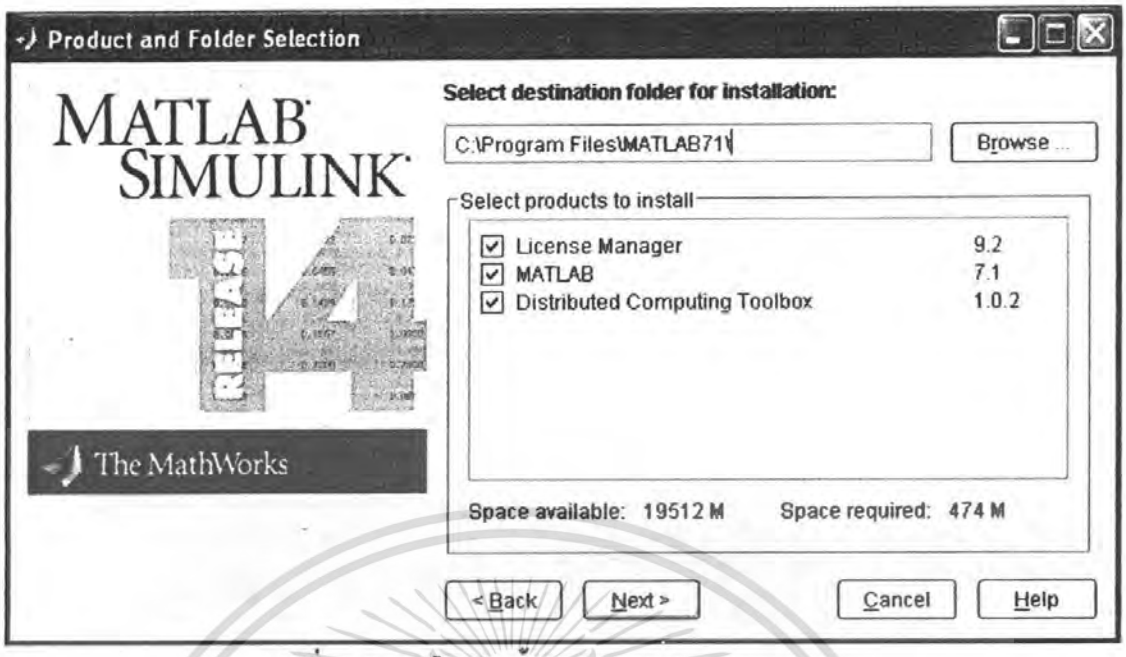



รูปที่ ก.4 การระบุที่อยู่ของไฟล์ license.dat

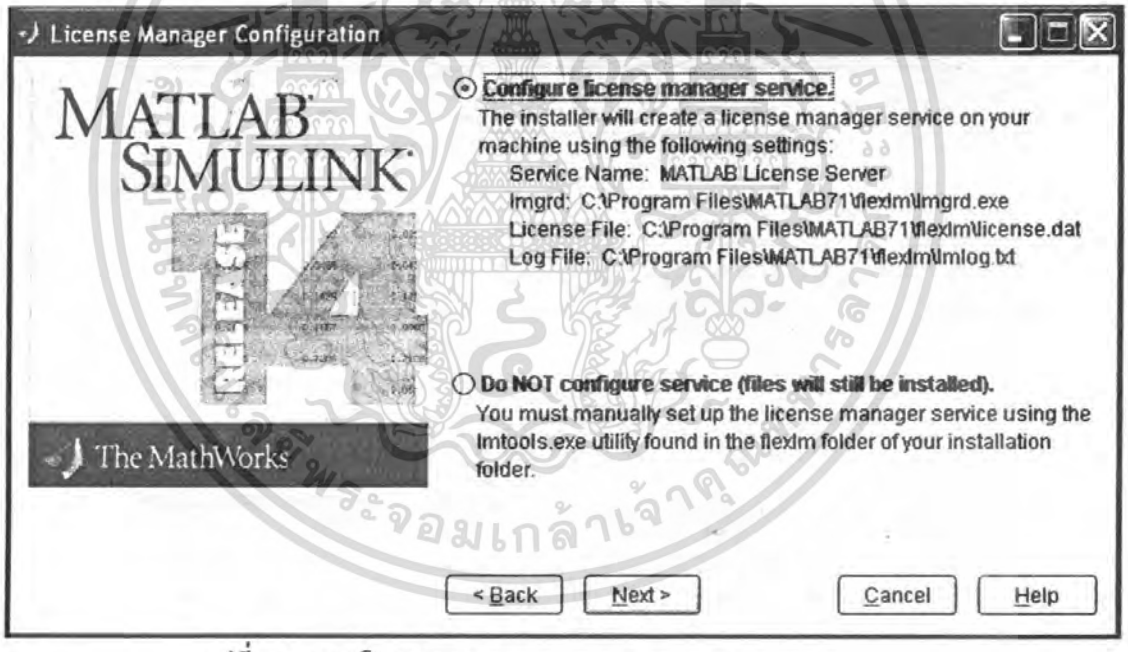
รูปที่ ก.5 เลือกรูปแบบการติดตั้ง CUSTOM

เลือกติดตั้ง license manager

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



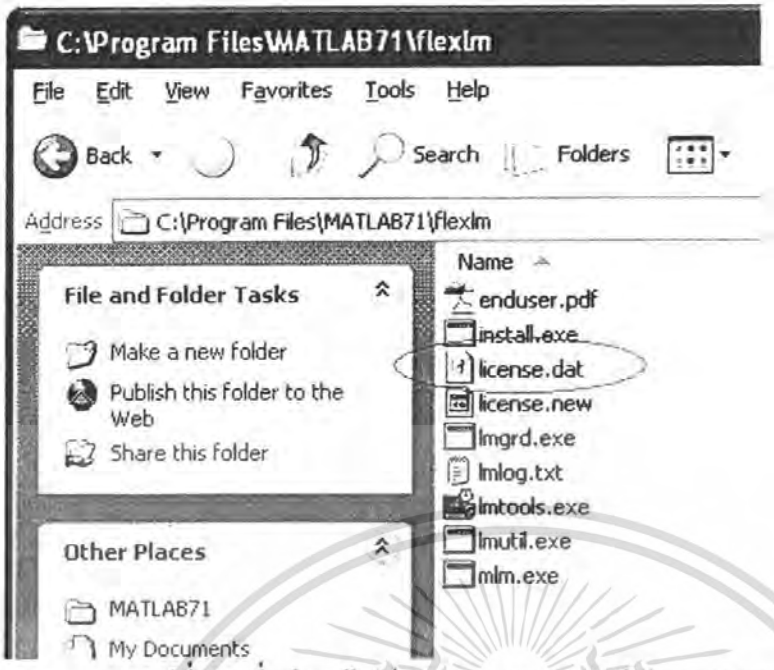

**รูปที่ ก.6 การเลือกติดตั้ง License Manager**




**รูปที่ ก.7 การเลือก Configure license manager service**

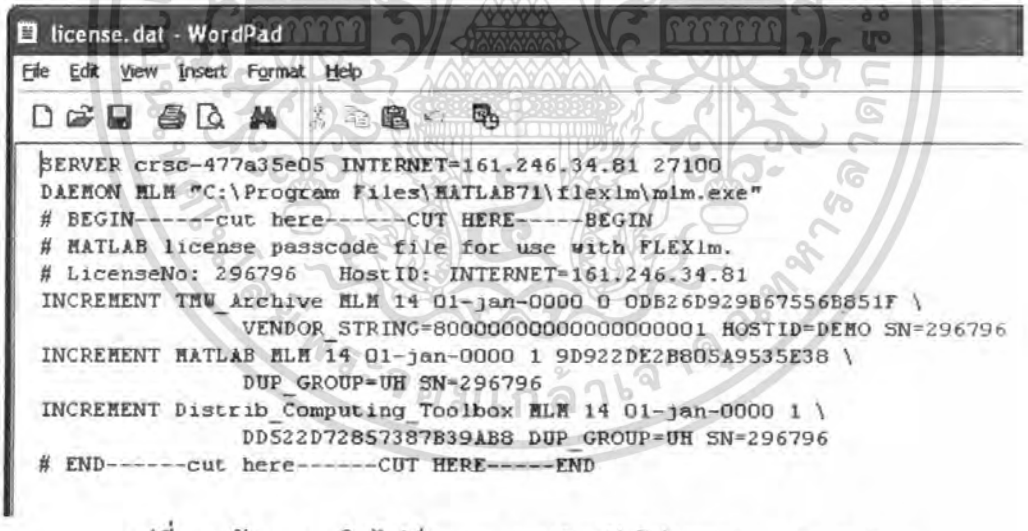
หลังจากติดตั้งเสร็จเรียบร้อยแล้ว ต้องเข้าไปทำการแก้ไข license ไฟล์ 2 ที่ ไฟล์แรกจะอยู่ใน โฟลเดอร์ matlab71/flexlm/license.dat

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ ก.8 ที่อยู่ของไฟล์ license.dat สำหรับโปรแกรม LMTOOLS**

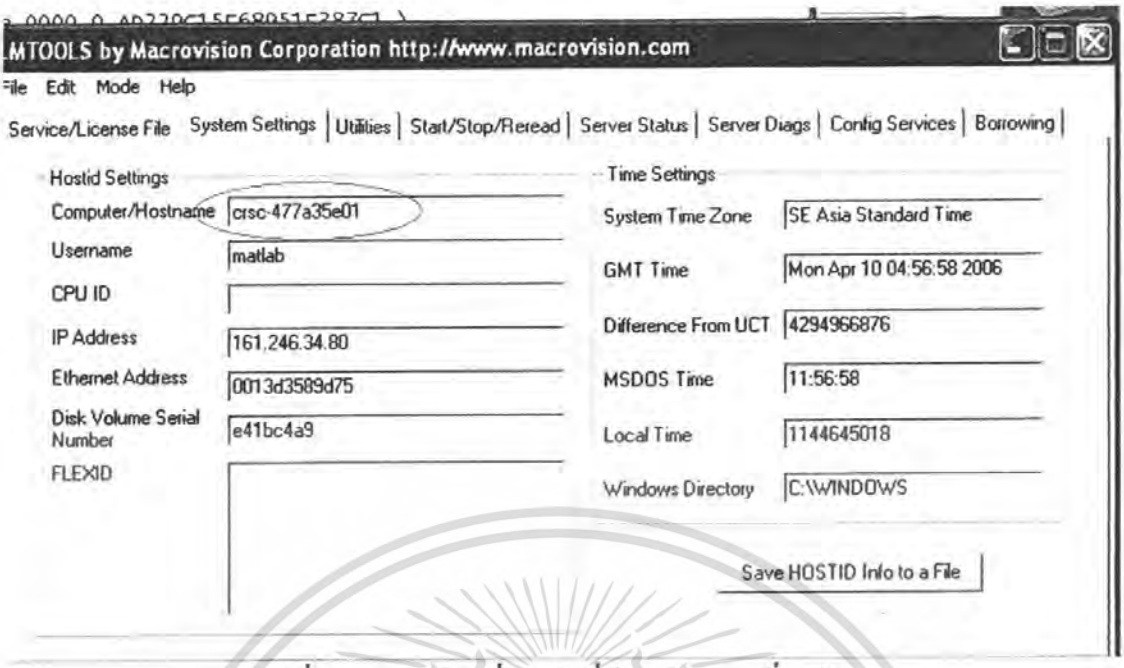
ให้ทำการเพิ่มข้อความเข้าไป ตามรูป




**รูปที่ ก.9 ข้อมูลภายในไฟล์ liense.dat สำหรับโปรแกรม LMTOOLS**

ซึ่งชื่อ SERVER ที่ถูกต้องสามารถดูได้จากโปรแกรม lmtools ที่อยู่ในโฟลเดอร์ matlab71/flexlm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

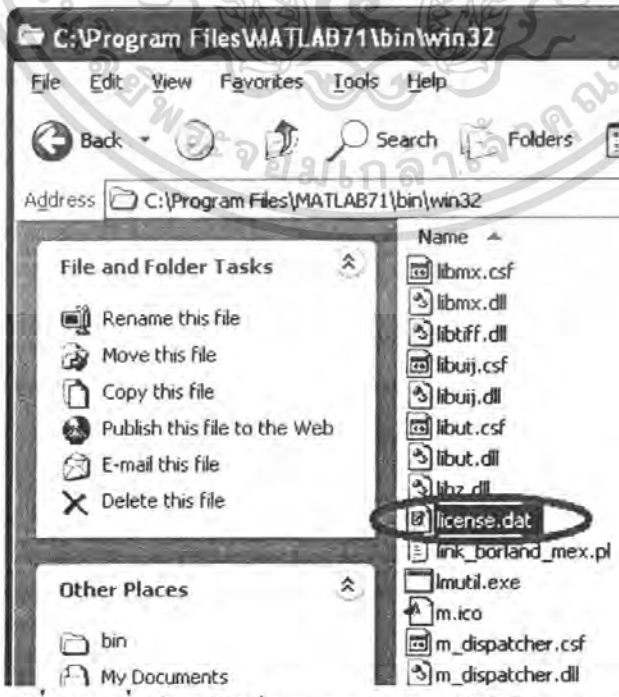



**รูปที่ ก.10 Textbox ที่ระบุถึงชื่อของ Server ที่ถูกต้อง**

ส่วนหมายเลข IP ถูกกำหนดมาแล้ว จึงต้องใส่ให้ตรง ( toolbox และ engine ให้หมายเลข ip ไม่เหมือนกัน ) และ port นั้นสามารถใช้พอร์ทอะไรก็ได้ ที่ไม่มีโปรแกรมอื่นใช้งานอยู่ และต้องตรงกันทุกไฟล์ทั้ง toolbox และ engine

บรรทัดที่สองคือ path ของโปรแกรม mlm

เมื่อแก้ไขไฟล์แรกแล้ว มาแก้ไฟล์ที่สองจะอยู่ใน matlab\bin\win32\license.dat

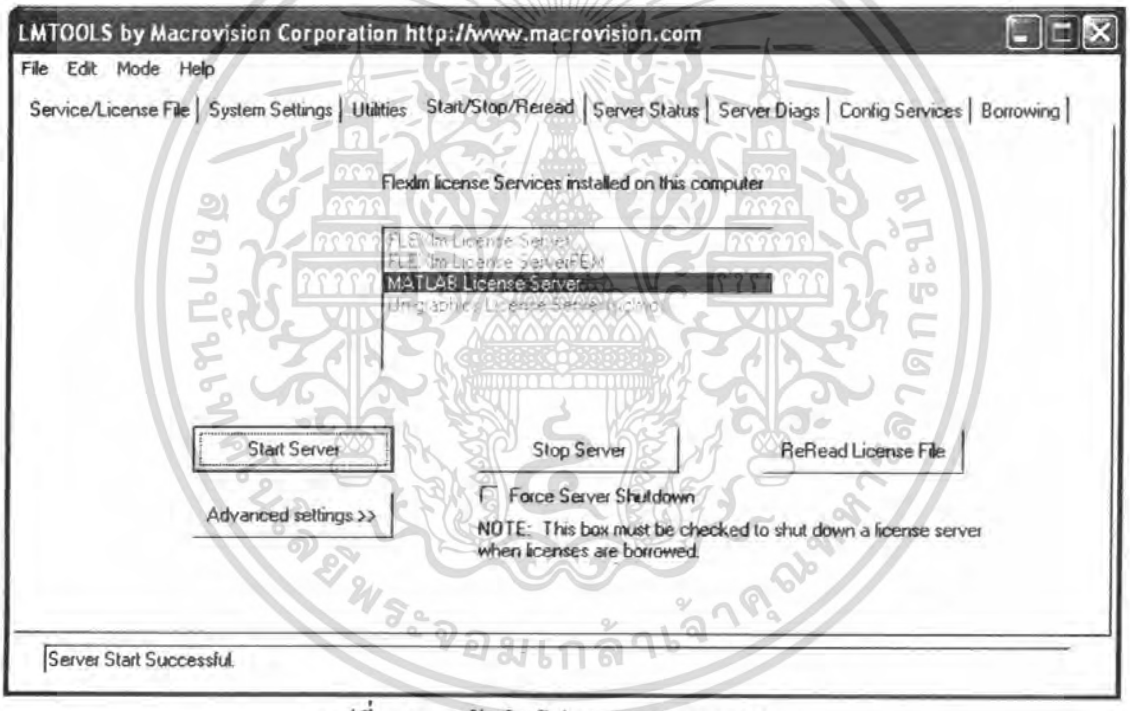



**รูปที่ ก.11 ที่อยู่ของไฟล์ license.dat สำหรับโปรแกรม MATLAB**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ ก.12 ข้อมูลภายในไฟล์ license.dat สำหรับโปรแกรม MATLAB**

หลังจากแก้ไขทั้งสองไฟล์แล้ว ผู้ใช้จะต้องทำหาร start license server เสียก่อน จึงจะเริ่มการใช้งานโปรแกรม Matlab ได้ โดยไปที่โปรแกรม lmtools เลือก start server

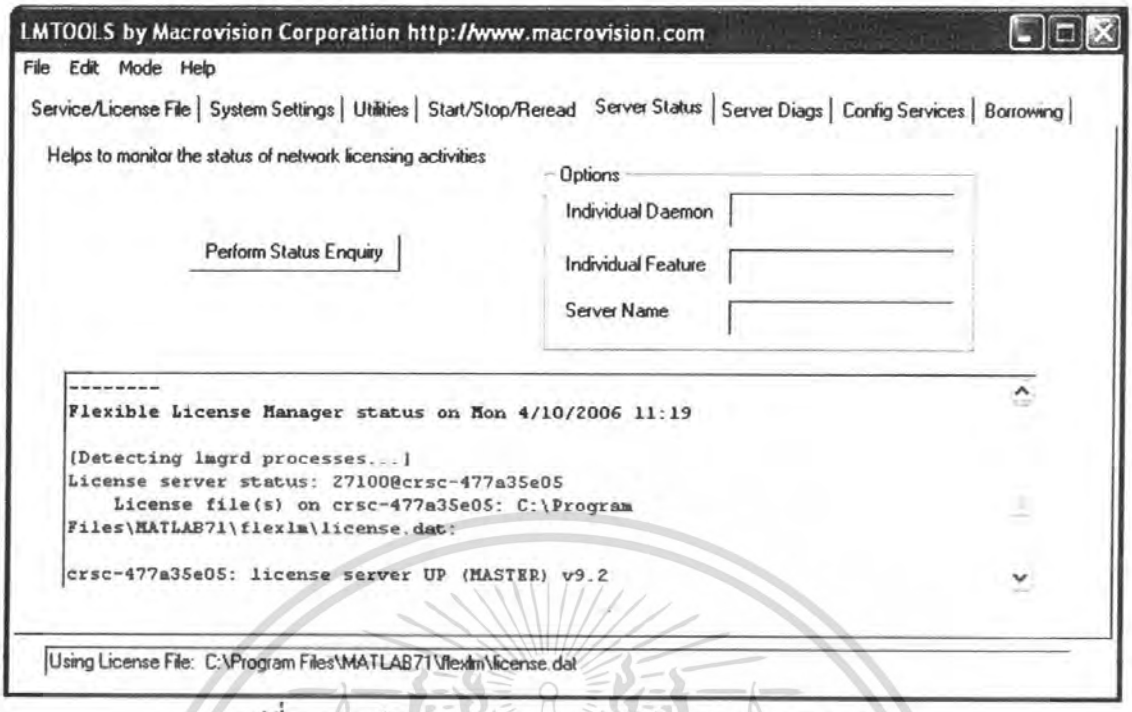



**รูปที่ ก.13 หน้าต่างโปรแกรม LMTOOLS**

หลังจากนั้นเลือก Reread license file ถ้าไฟล์ license.dat ทั้ง 2 ที่ถูกต้อง ที่ด้านล่างจะปรากฏข้อความ

เสร็จแล้วเลื่อนไปที่แท็บ Server status คลิกที่ Perform status enquiry โปรแกรมจะแสดงผลการทำงานของ license manager

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ ก.14 แท็บ Server status ของโปรแกรม LMTOOLS**

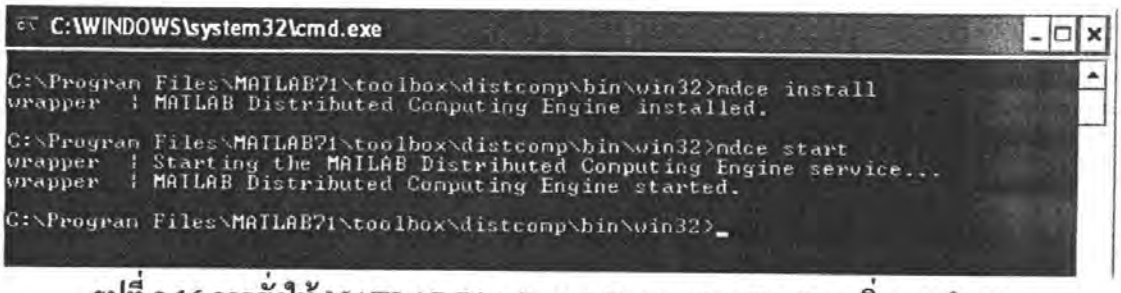
จากนั้นเข้าไปที่ dos เพื่อติดตั้ง mdce service

เข้าไปที่ matlav71\toolbox\distcomp\bin\win32

พิมพ์คำสั่ง mdce install

**รูปที่ ก.15 การติดตั้ง MATLAB Distributed Computing Toolbox**

เสร็จแล้วพิมพ์ mdce start



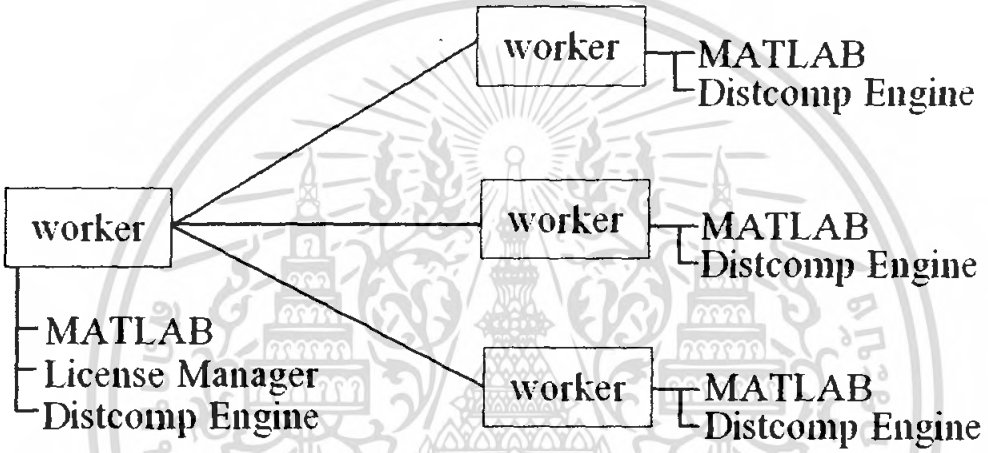

**รูปที่ ก.16 การสั่งให้ MATLAB Distributed Computing Toolbox เริ่มการทำงาน**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดท้ายพิมพ์ startjobmanager เพื่อเริ่มการทำงาน หลังจากที่ติดตั้ง toolbox เรียบร้อยแล้ว ขั้นตอนต่อไปคือการติดตั้ง engine ซึ่งจะต้องไปติดตั้งให้กับทุกเครื่องที่เหลือ โดยจะมีเครื่องหนึ่งติดตั้ง license manager เพื่อแจกจ่าย license ให้กับเครื่องอื่นๆ ซึ่งจำกัดจำนวนไว้ 8 เครื่อง

**การติดตั้ง MATLAB Distributed Computing Engine**

การติดตั้ง เพียงติดตั้ง license manager ให้กับเครื่องใดเครื่องหนึ่ง แล้วให้เครื่องอื่นๆ อ้าง license file มาที่เครื่องนี้ เมื่อติดตั้งเสร็จทุกขั้นตอนแล้วก็สามารถเริ่มดำเนินการได้

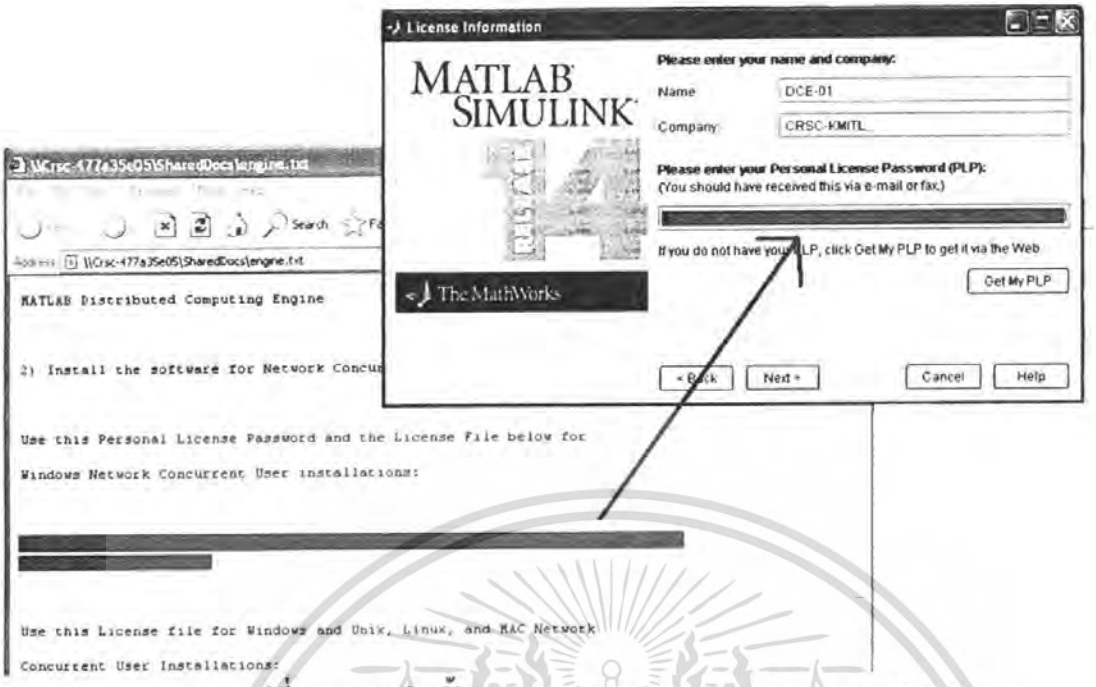



**รูปที่ ก.17 ความสัมพันธ์ของ worker**

เครื่องที่เป็น worker ไม่จำเป็นต้องใช้งาน matlab ได้ ขอเพียงให้สามารถ startworker ได้ก็เพียงพอแล้ว

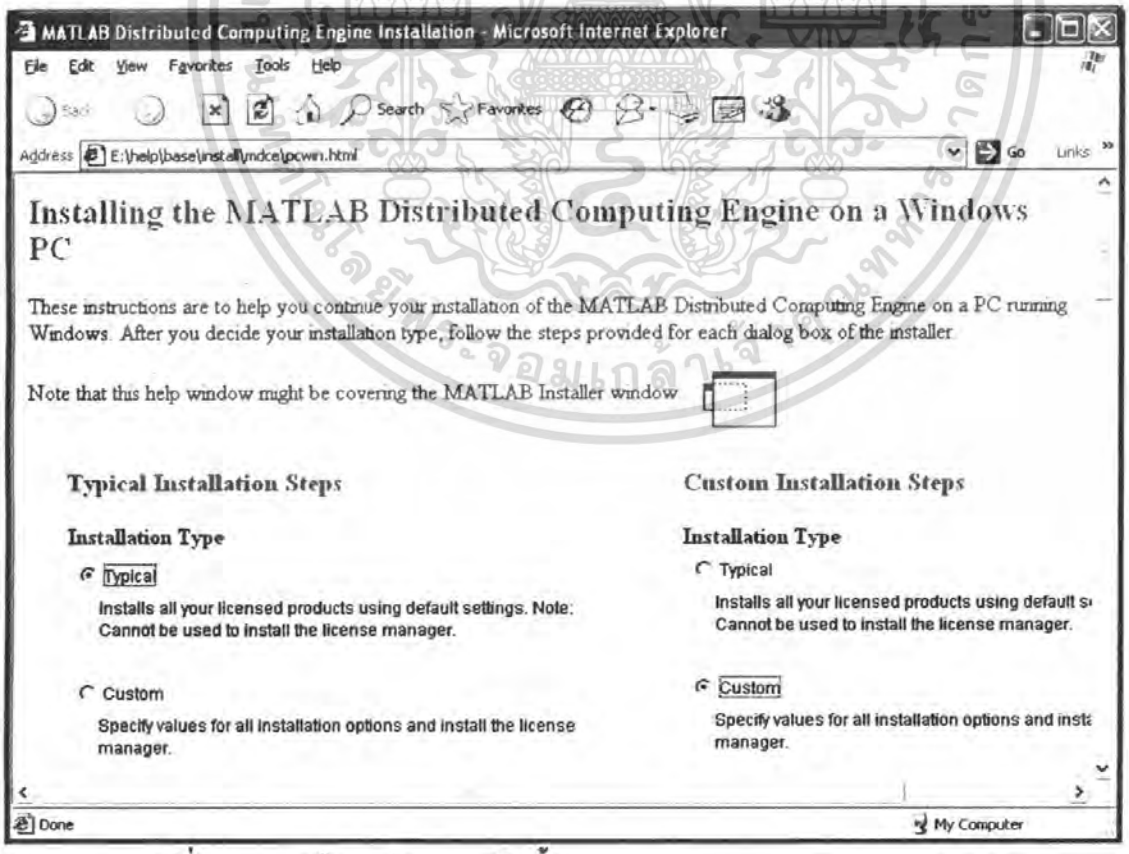
ในการติดตั้ง engine PLP ที่ใช้ในการติดตั้งจะเป็นคนละชุดกับ toolbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ ก.18 การติดตั้ง MATLAB Distributed Computing** Engine

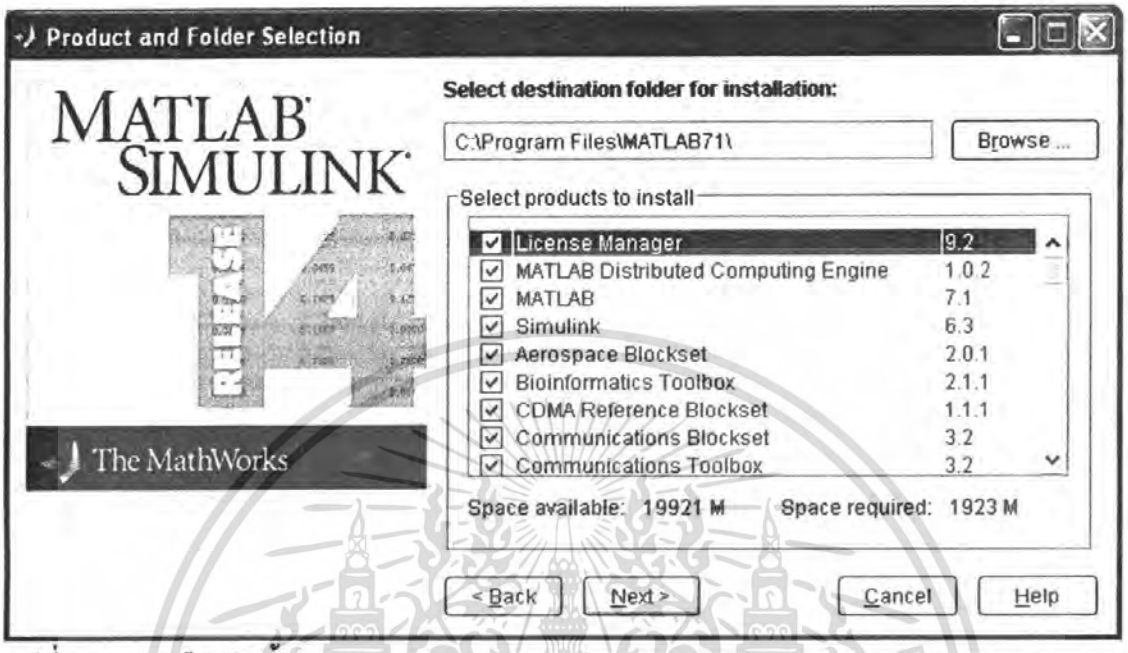
เมื่อใส่PLP ที่เป็นของ engine โปรแกรมติดตั้งจะแนะนำวิธีติดตั้งเป็น html



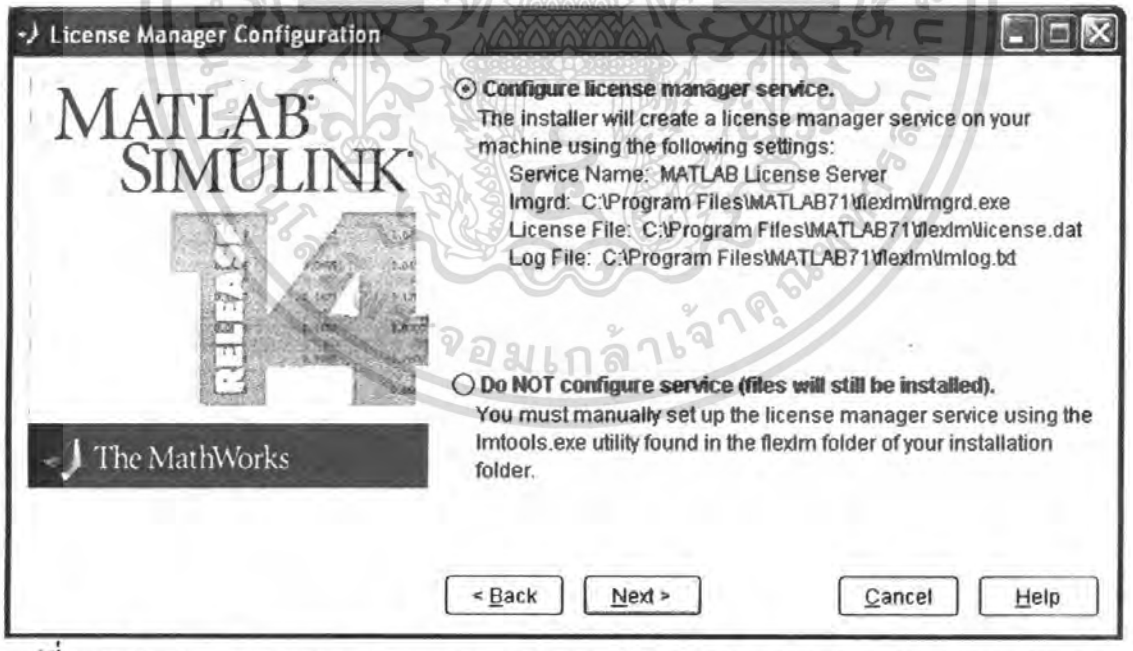

**รูปที่ ก.19 การเลือกรูปแบบการติดตั้ง MATLAB Distributed Computing Engine**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลับไปยังขั้นตอนการติดตั้งปกติ ในเครื่องแรก ต้องติดตั้ง License manager ลงไปด้วย ส่วนเครื่องที่เหลือไม่ต้องติดตั้ง




**รูปที่ ก.20 การเลือกติดตั้ง License Manager สำหรับ MATLAB Distributed Computing Engine**




**รูปที่ ก.21 Configure license manager service ของ MATLAB Distributed Computing Engine**

Worker เครื่องอื่นๆ ให้ตัวเลือก license manager ออก จากนั้นดำเนินขั้นตอนการติดตั้งไปตามปกติ จนกระทั่งการติดตั้งเสร็จสมบูรณ์ เมื่อติดตั้งเสร็จเรียบร้อยแล้ว ให้เข้าไปทำการแก้ไข license file เช่นเดียวกับการติดตั้ง toolbox ทั้งสองไฟล์ ไฟล์แรกจะมีเฉพาะเครื่องที่เป็น license managerเท่านั้น โดยจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่ในโฟลเดอร์ matlab71\flexlm\license.dat โดยเพิ่มชื่อของเครื่อง license manager และ หมายเลข ip ของเครื่องเข้าไป และ port ที่ใช้ติดต่อกัน (ต้องใช้พอร์ทเดียวกันหมดทุกเครื่อง ) ตามด้วย path ที่อยู่ของ mlm ( matlab71\flexlm\mlm.exe ) ไฟล์ที่ 2 ต้องแก้ไขทุกเครื่อง อยู่ที่ MATLAB71\bin\win32\license.dat โดยอ้างมาที่ชื่อและ ip ของเครื่องที่เป็น license manager เปิดโปรแกรม lmtool เพื่อทำการ start license server และ reread license file เข้าไปที่ command prompt เพื่อทำการติดตั้งและเริ่มการใช้งาน mdce service ( ทุกเครื่อง )

**รูปที่ ก.22 การติดตั้ง MATLAB Distributed Computing Engine**

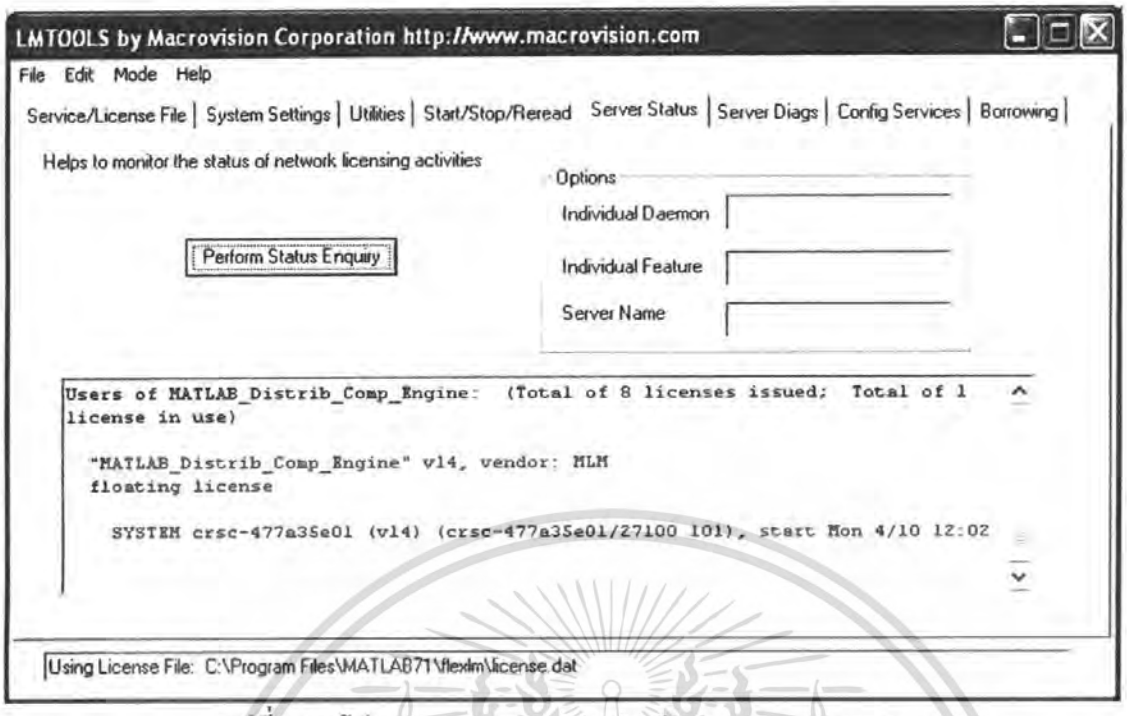
ขั้นตอนนี้คล้ายกับการติดตั้ง toolbox แต่เปลี่ยนจาก startjobmanager เป็น start worker แทน ซึ่งการกำหนด worker ในแต่ละเครื่อง ต้องตั้งชื่อไม่ให้ซ้ำกัน โดยเพิ่มออปชัน –name เข้าไป เช่น

Startworker –name worker1

**รูปที่ ก.23 การสั่งให้ MATLAB Distributed Computing Engine เริ่มการทำงาน**

เมื่อ start worker ครบทุดเครื่องแล้ว กลับมาที่ license manager เปิดโปรแกรม lmtools ที่แท็บ server status ผู้ใช้สามารถเรียกดูเครื่องที่ใช้งานอยู่ในขณะนั้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ ก.24 โปรแกรม LMTOOLS สำหรับ Jobmanager และ worker**

Note: เครื่อง worker แต่ละเครื่องจะต้องทำการ disable firewall เอาไว้ด้วย ไม่เช่นนั้นจะไม่สามารถติดต่อถึงกันได้

กลับมาที่เครื่องที่ติดตั้ง toolbox เปิดโปรแกรม matlab เพื่อทดสอบการทำงาน Distributed Computing Engine พิมพ์คำสั่ง a = findResource('jobmanager'); ตัวแปร a จะแสดงรายละเอียดต่างๆ ของ Distributed Computing Engine ขึ้น

**รูปที่ ก.25 หน้าต่าง Property Inspector แสดงรายละเอียดของ Jobmanager**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้