

การประยุกต์ขั้นตอนอานานิคมผึ้งเทียมเพื่อแก้ปัญหาการจัดตาราง  
การทำงานของเครื่องเสมือนในการประมวลผลแบบกลุ่มเมฆ

APPLIED ARTIFICIAL BEE COLONY ALGORITHM FOR SOLVING  
A VIRTUAL MACHINE SCHEDULING IN CLOUD COMPUTING



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาคณะหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2557

KMITL-2014-SC-M-002-025

การประยุกต์ขั้นตอนอณานิคมผึ้งเทียมเพื่อแก้ปัญหาการจัดตาราง  
การทำงานของเครื่องเสมือนในการประมวลผลแบบกลุ่มเมฆ

**APPLIED ARTIFICIAL BEE COLONY ALGORITHM FOR SOLVING  
A VIRTUAL MACHINE SCHEDULING IN CLOUD COMPUTING**



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต  
สาขาวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
พ.ศ. 2557

**KMITL-2014-SC-M-002-025**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**APPLIED ARTIFICIAL BEE COLONY ALGORITHM FOR SOLVING  
A VIRTUAL MACHINE SCHEDULING IN CLOUD COMPUTING**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
MASTER OF SCIENCE IN COMPUTER SCIENCE  
FACULTY OF SCIENCE  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2014**

**KMITL-2014-SC-M-002-025**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**COPYRIGHT 2014**

**FACULTY OF SCIENCE**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์

การประยุกต์ขั้นตอนอาณานิคมผึ้งเทียมเพื่อแก้ปัญหการจัดตาราง  
การทำงานของเครื่องเสมือนในการประมวลผลแบบกลุ่มเมฆ  
Applied Artificial Bee Colony Algorithm for Solving A Virtual  
Machine Scheduling in Cloud Computing

นักศึกษา

นางสาวบุญททัย เครือแก้ว

รหัสประจำตัว

53650853

ปริญญา





วิทยาศาสตรมหาบัณฑิต

สาขาวิชา

วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์

ดร.วรางคณา กัมปาน

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.จิรพร	วีระพันธุ์	
ดร.รุ่งรัตน์	เวียงศรีพนาวัลย์	
ผศ.ดร.กาญจนา	ชินสาร	
ดร.วรางคณา	กัมปาน	

วัน / เดือน / ปี ที่สอบ 14 พฤษภาคม พ.ศ. 2556 เวลา 13.30 – 16.30 น.  
สถานที่สอบ ณ ห้อง 302 ชั้น 3 อาคารปฏิบัติการใหม่

คณะวิทยาศาสตร์รับรองแล้ว

(รองศาสตราจารย์ ดร. ดุษณี ธีระบริพัฒน์)  
คณบดีคณะวิทยาศาสตร์

วันที่ 23 เดือน พค. พ.ศ. 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การประยุกต์ขั้นตอนอาณานิคมผึ้งเทียมเพื่อแก้ปัญหาการจัดตารางการทำงานของเครื่องเสมือนในการประมวลผลแบบกลุ่มเมฆ
นักศึกษา	นางสาวบุญหทัย เครือแก้ว
รหัสประจำตัว	53650853
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์
พ.ศ.	2557
อาจารย์ที่ปรึกษา	ดร.วรางคณา กิมปาน

### บทคัดย่อ

การออกแบบการจัดตารางการทำงานเพื่อเข้าใช้ทรัพยากรในการประมวลผลแบบกลุ่มเมฆ (Cloud Computing) ถือเป็นปัญหาที่สำคัญ โดยสิ่งที่จะต้องคำนึงถึงในการแก้ปัญหานี้คือ ค่าใช้จ่าย การเพิ่มประสิทธิภาพการให้บริการ เวลาในการเข้าใช้บริการ และเงื่อนไขในการร้องขอเพื่อเข้าใช้บริการ จากปัญหาในข้างต้นจึงได้เสนอขั้นตอนการจัดตารางการทำงานและการกระจายงานแบบสมดุลในการเข้าใช้ทรัพยากรที่มีอยู่อย่างจำกัด การจัดตารางเวลาเข้าทำงานที่ได้นั้นควรปรับรูปแบบการเข้าทำงานให้เหมาะสมกับสภาพแวดล้อม ในงานวิจัยนี้จึงได้เสนอขั้นตอนการจัดตารางเวลาเพื่อเข้าใช้เครื่องเสมือน (Virtual Machine) ที่มีสภาพแวดล้อมภายในเหมือนกันและต่างกันในการประมวลผลแบบกลุ่มเมฆ โดยประยุกต์ใช้วิธีการหาคำตอบแบบขั้นตอนอาณานิคมผึ้งเทียม (Artificial Bee Colony Algorithm) ควบคู่กับวิธีการจัดลำดับงานแบบฮิวริสติก (Heuristics) ในการรักษาสมดุลการทำงานในระบบและลดเวลารวมในการทำงาน (Makespan) ให้น้อยลง ในขั้นตอนของการทดลองได้มีการใช้ CloudSim ในการสร้างแบบจำลอง ผลจากการทดลองแสดงให้เห็นว่าในกรณีที่สภาพแวดล้อมภายในของเครื่องเสมือนมีสภาพเหมือนกัน (Homogeneous) ควรใช้ขั้นตอนอาณานิคมผึ้งเทียมที่นำเสนอทำงานร่วมกับการจัดตารางงาน โดยพิจารณาจากขนาดของงานที่มีขนาดใหญ่ที่สุดเข้าทำงานก่อน (Longest Job First) และในกรณีที่สภาพแวดล้อมภายในของเครื่องเสมือนมีสภาพต่างกัน (Heterogeneous) ควรใช้วิธีการที่นำเสนอควบคู่กับการจัดลำดับงาน โดยพิจารณาจากขนาดของงานที่มีขนาดเล็กที่สุดเข้าทำงานก่อน (Shortest Job First) จะทำให้ประสิทธิภาพในการจัดตารางการทำงานเหมาะสมที่สุด ในการกระจายงานแบบสมดุลและลดเวลารวมในการทำงาน เมื่อปริมาณงานในระบบเพิ่มมากขึ้นและสภาพแวดล้อมระบบมีการเปลี่ยนแปลง

**คำสำคัญ:** การประมวลผลแบบกลุ่มเมฆ การจำลองการทำงานของคอมพิวเตอร์เสมือน การจัด

ตารางการทำงาน การกระจายงานแบบสมดุล ขั้นตอนอาณานิคมผึ้งเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
I  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>Thesis Title</b>	Applied Artificial Bee Colony Algorithm for Solving a Virtual Machine Scheduling in Cloud Computing
<b>Student</b>	Boonhatai Kruekaew
<b>Student ID</b>	53650853
<b>Degree</b>	Master of Science
<b>Program</b>	Computer Science
<b>Year</b>	2014
<b>Thesis Advisor</b>	Dr. Warangkhana Kimpan

## ABSTRACT

Resource scheduling management design using Cloud computing is an important problem. To eliminate such a problem, the factors to be focused are scheduling model, cost, quality of service, time, and conditions of the request for access to services. A good task scheduler should adapt its scheduling strategy to the environment changing and load balancing Cloud task scheduling policy. In this research, Artificial Bee Colony (ABC) algorithm and Heuristic scheduling algorithm are applied to optimize the scheduling task for Virtual Machine (VM) on homogeneous and heterogeneous environment platforms on Cloud computing. The main contribution of work is to analyze the difference of VMs load balancing algorithm and to reduce the makespan of data processing time. The scheduling strategy was simulated using CloudSim tools. The experimental results indicated that, in the case of homogeneous VMs environment, the best result is the combination of the proposed ABC algorithm and the scheduling based on the size of task with the Longest Job First (LJF) scheduling algorithm. On the other hand, in the case of heterogeneous VMs environment, the best result is the combination of the proposed ABC algorithm and the scheduling based on the size of task with the Shortest Job First (SJF) scheduling algorithm. Both cases performed a good performance scheduling strategy in changing environment and balancing work load which can reduce the makespan of data processing time.

**Keywords:** Cloud Computing, Virtualization Machine, Scheduling, Load Balance, Artificial Bee Colony (ABC) Algorithm

# กิตติกรรมประกาศ

วิทยานิพนธ์นี้ ตั้งแต่เริ่มต้นจนสำเร็จลุล่วงนั้น นับว่าเป็นความภาคภูมิใจของผู้วิจัยเป็นอย่างมาก หากลำพังผู้วิจัยเพียงผู้เดียวคงไม่สามารถดำเนินงานจนประสบความสำเร็จเช่นนี้ได้ ทั้งนี้เพราะได้รับความกรุณาจากบุคคลหลายๆ ท่านดังนี้

ขอขอบพระคุณ ดร.วรางคณา กิมปาน อาจารย์ที่ปรึกษา ที่ได้สละเวลาให้คำแนะนำ ดูแลเอาใจใส่ รวมทั้งให้ข้อคิด ความรู้ที่มีประโยชน์ที่ใช้ประกอบการทำงานวิจัยจนสำเร็จลุล่วง

ขอขอบพระคุณ รองศาสตราจารย์ ดร.จิรพร วีระพันธุ์ ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์ และผู้ช่วยศาสตราจารย์ ดร.กฤษณะ ชินสาร คณะกรรมการสอบหัวข้อวิทยานิพนธ์ ที่กรุณาให้คำแนะนำ ตลอดจนข้อชี้แนะจนในที่สุดทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลงได้

ขอขอบคุณพระคุณบิดา มารดา และพี่ ที่สนับสนุนให้ได้เรียนในระดับที่ได้ตั้งใจ อีกทั้งยังได้สนับสนุนดูแลเรื่องค่าใช้จ่ายต่างๆ ระหว่างการศึกษาก็คืออย่างดียิ่งด้วย

ท้ายสุดขอขอบคุณเพื่อนๆ ร่วมรุ่น และพี่น้องทุกคนที่ให้คำปรึกษา และช่วยอำนวยความสะดวกในด้านต่างๆ

สำหรับคุณงามความดีและประโยชน์อันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดา มารดา อาจารย์ทุกท่านซึ่งเป็นที่เคารพรักยิ่ง ตลอดจนญาติพี่น้อง รุ่นพี่และเพื่อนๆ ที่รักทุกคน

บุญหทัย เครือแก้ว

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญรูป.....	IX
<b>บทที่ 1 บทนำ.....</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	2
1.3 สมมติฐานของงานวิจัย.....	2
1.4 ขอบเขตของงานวิจัย.....	3
1.5 ขั้นตอนการศึกษาและดำเนินงานวิจัย.....	3
1.6 ประโยชน์ที่ได้รับจากงานวิจัย.....	4
<b>บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....</b>	<b>5</b>
2.1 การประมวลผลแบบกลุ่มเมฆ.....	5
2.1.1 หลักการของการประมวลผลแบบกลุ่มเมฆ.....	5
2.1.2 โครงสร้างการประมวลผลแบบกลุ่มเมฆ.....	8
2.1.3 ประเภทของการประมวลผลแบบกลุ่มเมฆ.....	9
2.1.4 การให้บริการของระบบการประมวลผลแบบกลุ่ม.....	10
2.1.5 ประโยชน์ของการประมวลผลกลุ่มเมฆ.....	14
2.1.6 ข้อจำกัดของระบบประมวลผลกลุ่มเมฆ.....	15
2.1.7 ข้อดีและข้อเสียของการประมวลผลแบบกลุ่มเมฆ.....	16
2.2 เทคโนโลยีการจำลองเครื่องคอมพิวเตอร์เสมือน.....	17
2.2.1 ตัวอย่างของผลิตภัณฑ์ที่ใช้ในการจำลองเทคโนโลยีเครื่องคอมพิวเตอร์เสมือน.....	17
2.2.2 ประโยชน์ที่ได้รับจากการจำลองเทคโนโลยีเครื่องคอมพิวเตอร์เสมือน.....	18
2.3 ขั้นตอนวิธีอาณานิคมผึ้งเทียม.....	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)

	หน้า
2.4 วิธีการจัดลำดับงานแบบฮิวริสติก.....	22
2.5 เครื่องมือที่ใช้ในการสร้างแบบจำลอง CloudSim-3.0.1.....	23
2.5.1 ลักษณะทั่วไปของ CloudSim.....	23
2.5.2 รูปแบบการทำงานของ CloudSim.....	24
2.6 การวัดการกระจายข้อมูล.....	25
2.6.1 การวัดแนวโน้มเข้าสู่ส่วนกลาง.....	25
2.6.1.1 ค่าเฉลี่ยเลขคณิต.....	26
2.6.1.2 มัชยฐาน.....	26
2.6.1.3 ฐานนิยม.....	27
2.6.2 ลักษณะของการกระจายข้อมูล.....	27
2.7 งานวิจัยที่เกี่ยวข้อง.....	28
<b>บทที่ 3 วิธีการดำเนินงานวิจัย.....</b>	<b>30</b>
3.1 การประยุกต์ใช้ขั้นตอนวิธีอาณานิคมฟังก์ชัน.....	30
<b>บทที่ 4 การทดลองและผลการทดลอง.....</b>	<b>35</b>
4.1 ข้อมูลที่ใช้ในการทดสอบประสิทธิภาพ.....	35
4.2 การเตรียมข้อมูล.....	36
4.3 เครื่องมือที่ใช้ในงานวิจัย.....	36
4.4 ผลการทดลองกรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกัน.....	37
4.4.1 การตั้งค่าพารามิเตอร์สภาพแวดล้อมในการทดลอง.....	37
4.4.2 การตั้งค่าพารามิเตอร์ในขั้นตอนวิธีอาณานิคมฟังก์ชัน.....	38
4.4.3 ผลการทดสอบประสิทธิภาพกรณีทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลง.....	38
4.4.3.1 ผลการทดสอบประสิทธิภาพ เมื่อชุดข้อมูลมีการเปลี่ยนแปลง.....	38
4.4.3.2 ผลการทดสอบประสิทธิภาพ เมื่อขนาดของงานมีการเปลี่ยนแปลง.....	40
4.4.3.3 ผลการทดสอบประสิทธิภาพ เมื่อจำนวน PE ของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง.....	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

หน้า

4.4.3.4 ผลการทดสอบประสิทธิภาพ เมื่อขนาด MIPS ของ PE ในเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง.....	46
4.4.3.5 ผลการทดสอบประสิทธิภาพ เมื่อจำนวน RAM ของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง.....	47
4.4.3.6 ผลการทดสอบประสิทธิภาพ เมื่อขนาดของแบนด์วิธของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง.....	49
4.4.3.7 ผลการทดสอบประสิทธิภาพ เมื่อข้อมูลมีการกระจายแบบโค้งปกติ.....	51
4.4.3.8 ผลการทดสอบประสิทธิภาพ เมื่อข้อมูลมีการกระจายแบบเบ้ขวา.....	52
4.4.3.9 ผลการทดสอบประสิทธิภาพ เมื่อข้อมูลมีการกระจายแบบเบ้ซ้าย.....	54
4.4.3.10 ผลการทดสอบประสิทธิภาพ เมื่อจำนวนฝั่งในระบบมีการเปลี่ยนแปลง ..	55
4.4.4 ผลการทดสอบประสิทธิภาพกรณีทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง.....	57
4.4.4.1 ผลการทดสอบประสิทธิภาพเมื่อชุดข้อมูลมีการเปลี่ยนแปลง.....	58
4.4.4.2 ผลการทดสอบประสิทธิภาพเมื่อขนาดของงานมีการเปลี่ยนแปลง .....	60
4.5 ผลการทดลองกรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในต่างกัน.....	64
4.5.1 การตั้งค่าพารามิเตอร์สภาพแวดล้อมในการทดลอง .....	64
4.5.2 การตั้งค่าพารามิเตอร์ในขั้นตอนวิธีอาณานิคมฝั่งเทียบ.....	65
4.5.3 ผลการทดสอบประสิทธิภาพกรณีทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลง.....	65
4.5.3.1 ผลการทดสอบประสิทธิภาพ เมื่อชุดข้อมูลมีการเปลี่ยนแปลง.....	65
4.5.3.2 ผลการทดสอบประสิทธิภาพ เมื่อขนาดของงานมีการเปลี่ยนแปลง .....	67
4.5.3.3 ผลการทดสอบประสิทธิภาพ เมื่อข้อมูลมีการกระจายแบบโค้งปกติ.....	71
4.5.3.4 ผลการทดสอบประสิทธิภาพ เมื่อข้อมูลมีการกระจายแบบเบ้ขวา.....	72
4.5.3.5 ผลการทดสอบประสิทธิภาพ เมื่อข้อมูลมีการกระจายแบบเบ้ซ้าย.....	73
4.5.3.6 ผลการทดสอบประสิทธิภาพ เมื่อจำนวนฝั่งในระบบมีการเปลี่ยนแปลง ....	74
4.5.4 ผลการทดสอบประสิทธิภาพกรณีทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง.....	76
4.5.4.1 ผลการทดสอบประสิทธิภาพเมื่อชุดข้อมูลมีการเปลี่ยนแปลง .....	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
4.5.4.2 ผลการทดสอบประสิทธิภาพเมื่อนำขนาดของงานมีการเปลี่ยนแปลง .....	78
<b>บทที่ 5 บทสรุป.....</b>	<b>83</b>
เอกสารอ้างอิง.....	85
ภาคผนวก ก ตัวอย่างชุดข้อมูลที่ใช้ในการทดลอง .....	88
ภาคผนวก ข งานวิจัยที่ตีพิมพ์ .....	99
ประวัติผู้เขียน .....	114



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

ตารางที่	หน้า
4.1 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพ .....	35
4.2 ตัวอย่างในการจัดเรียงข้อมูล .....	36
4.3 ค่าพารามิเตอร์ของแบบจำลองการประมวลผลแบบกลุ่มเมฆ กรณีเครื่องคอมพิวเตอร์เสมือน มีสภาพแวดล้อมภายในเหมือนกันทั้งหมด.....	37
4.4 ค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมฟังก์ชันเทียม (Homogenous).....	38
4.5 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของกรณีที่ 4.4.3.1 .....	38
4.6 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของการทดสอบกรณีที่ 4.4.3.2.....	41
4.7 เปรอ์เซ็นต์ค่าความต่างของเวลาที่ลดลงในการจัดตารางงานแบบฮิวริสติกของข้อมูลชุดที่ 8 (Homogenous) .....	52
4.8 เปรอ์เซ็นต์ค่าความต่างของเวลาที่ลดลงในการจัดตารางงานแบบฮิวริสติกของข้อมูลชุดที่ 9 (Homogenous) .....	53
4.9 เปรอ์เซ็นต์ค่าความต่างของเวลาที่ลดลงในการจัดตารางงานแบบฮิวริสติกของข้อมูลชุดที่ 10 (Homogenous) .....	54
4.10 ค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมฟังก์ชันเทียม (Homogenous).....	55
4.11 สภาพแวดล้อมของเครื่องคอมพิวเตอร์เสมือน กรณีสภาพแวดล้อมภายในเหมือนกัน.....	57
4.12 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของการทดสอบกรณีที่ 4.4.4.1.....	58
4.13 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของการทดสอบกรณีที่ 4.4.4.2.....	60
4.14 ค่าพารามิเตอร์ของแบบจำลองการประมวลผลแบบกลุ่มเมฆกรณีเครื่องคอมพิวเตอร์เสมือน มีสภาพแวดล้อมภายในต่างกัน.....	64
4.15 ค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมฟังก์ชันเทียม (Heterogeneous).....	65
4.16 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของการทดสอบกรณีที่ 4.5.3.1.....	65
4.17 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของการทดสอบกรณีที่ 4.5.3.2.....	68
4.18 ค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมฟังก์ชันเทียม (Heterogeneous).....	74
4.19 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของการทดสอบกรณีที่ 4.5.4.1.....	76
4.20 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของการทดสอบกรณีที่ 4.5.4.2.....	79

# สารบัญรูป

รูปที่	หน้า
2.1	หลักการทํางานของระบบการประมวลผลแบบกลุ่มเมฆ.....5
2.2	ระบบการประมวลผลแบบกลุ่มเมฆแบ่งตามขอบเขตการใช้งาน .....9
2.3	การให้บริการของระบบการประมวลผลแบบกลุ่มเมฆ .....11
2.4	ตัวอย่างผู้ให้บริการในระดับโครงสร้างฮาร์ดแวร์พื้นฐาน .....12
2.5	ตัวอย่างผู้ให้บริการในระดับแพลตฟอร์ม .....13
2.6	ตัวอย่างผู้ให้บริการในระดับซอฟต์แวร์ .....13
2.7	สถาปัตยกรรมของเทคโนโลยีการจำลองเครื่องคอมพิวเตอร์เสมือนด้วยซอฟต์แวร์ .....18
2.8	การเดินแกว่งของฝั่งเพื่อสื่อสารข้อมูลของแหล่งอาหาร .....20
2.9	สถาปัตยกรรมลำดับชั้นของ CloudSim (Layered CloudSim Architecture).....23
2.10	การทำงานของ CloudSim.....24
2.11	การกระจายของข้อมูลแบบการกระจายแบบสมมาตร .....27
2.12	การกระจายของข้อมูลแบบการกระจายที่เบ้ทางขวา.....28
2.13	การกระจายของข้อมูลแบบการกระจายที่เบ้ทางซ้าย .....28
3.1	ผังงานแสดงการจัดตารางงานเพื่อเข้าใช้เครื่องคอมพิวเตอร์เสมือนและระบบมีการกระจายงานแบบสมดุล โดยประยุกต์ใช้ขั้นตอนวิธีอาณานิคมฝั่งเทียม.....31
4.1	เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 เมื่องานมีปริมาณเพิ่มขึ้น (Homogenous) .....39
4.2	เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 2 เมื่องานมีปริมาณเพิ่มขึ้น (Homogenous) .....39
4.3	เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 3 เมื่องานมีปริมาณเพิ่มขึ้น (Homogenous) .....40
4.4	เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 เมื่องานมีขนาด 5000 – 20000 (Homogenous).....41
4.5	เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 4 เมื่องานมีขนาด 15000 – 35000 (Homogenous).....42
4.6	เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 5 เมื่องานมีขนาด 25000 – 45000 (Homogenous).....42
4.7	เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 6 เมื่องานมีขนาด 35000 – 45000 (Homogenous).....43
4.8	เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 7 เมื่องานมีขนาด 45000 – 65000 (Homogenous).....43
4.9	เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและกำหนดค่า PE = 1 (Homogenous) .....44
4.10	เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและกำหนดค่า PE = 2 (Homogenous) .....45
4.11	เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและกำหนดค่า PE = 4 (Homogenous) .....45
4.12	เวลาทำงานรวมเฉลี่ย เมื่องานเพิ่มขึ้นและกำหนดค่า MIPS = 1000 (Homogenous) .....46

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.13 เวลาทำงานรวมเฉลี่ย เมื่องานเพิ่มขึ้นและกำหนดค่า MIPS = 2000 (Homogenous) .....	47
4.14 เวลาทำงานรวมเฉลี่ย เมื่องานเพิ่มขึ้นและกำหนดค่า RAM = 1024 (Homogenous) .....	48
4.15 เวลาทำงานรวมเฉลี่ย เมื่องานเพิ่มขึ้นและกำหนดค่า RAM = 2048 (Homogenous) .....	48
4.16 เวลาทำงานรวมเฉลี่ย เมื่องานเพิ่มขึ้นและกำหนดขนาดแบนด์วิธ = 1000 (Homogenous)....	50
4.17 เวลาทำงานรวมเฉลี่ย เมื่องานเพิ่มขึ้นและกำหนดขนาดแบนด์วิธ = 2000 (Homogenous)....	50
4.18 เวลาทำงานรวมเฉลี่ย เมื่อข้อมูลมีการกระจายแบบโค้งปกติ (Homogenous) .....	51
4.19 เวลาทำงานรวมเฉลี่ย เมื่อข้อมูลมีการกระจายแบบเบ้ขวา (Homogenous) .....	53
4.20 เวลาทำงานรวมเฉลี่ย เมื่อข้อมูลมีการกระจายแบบเบ้ซ้าย (Homogenous) .....	54
4.21 เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและกำหนดค่าพารามิเตอร์ของขั้นตอน วิธีอณานิคมฝูงเทียม Value I (Homogenous).....	56
4.22 เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและกำหนดค่าพารามิเตอร์ของขั้นตอน วิธีอณานิคมฝูงเทียม Value II (Homogenous) .....	56
4.23 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 เมื่อจำนวน VM เพิ่มขึ้น (Homogenous) .....	58
4.24 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 2 เมื่อจำนวน VM เพิ่มขึ้น (Homogenous) .....	59
4.25 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 3 เมื่อจำนวน VM เพิ่มขึ้น (Homogenous) .....	59
4.26 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 ขนาดของงาน 5000 – 20000 (Homogenous) .....	61
4.27 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 4 ขนาดของงาน 15000 – 35000 (Homogenous) .....	61
4.28 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 5 ขนาดของงาน 25000 – 45000 (Homogenous) .....	62
4.29 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 6 ขนาดของงาน 35000 – 55000 (Homogenous) .....	62
4.30 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 7 ขนาดของงาน 45000 – 65000 (Homogenous) .....	63
4.31 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 เมื่องานมีปริมาณเพิ่มขึ้น (Heterogeneous) .....	66
4.32 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 2 เมื่องานมีปริมาณเพิ่มขึ้น (Heterogeneous) .....	66
4.33 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 3 เมื่องานมีปริมาณเพิ่มขึ้น (Heterogeneous) .....	67
4.34 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 ขนาดของงาน 5000 – 20000 (Heterogeneous) .....	68
4.35 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 4 ขนาดของงาน 15000 – 35000 (Heterogeneous).....	69
4.36 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 5 ขนาดของงาน 25000 – 45000 (Heterogeneous).....	69
4.37 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 6 ขนาดของงาน 35000 – 55000 (Heterogeneous).....	70

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.38 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 7 ขนาดของงาน 45000 – 65000 (Heterogeneous).....	70
4.39 เวลาทำงานรวมเฉลี่ยของข้อมูล เมื่อข้อมูลมีการกระจายแบบ โค้งปกติ (Heterogeneous).....	71
4.40 เวลาทำงานรวมเฉลี่ยของข้อมูล เมื่อข้อมูลมีการกระจายแบบเบ้ขวา (Heterogeneous).....	72
4.41 เวลาทำงานรวมเฉลี่ยของข้อมูล เมื่อข้อมูลมีการกระจายแบบเบ้ซ้าย (Heterogeneous).....	73
4.42 เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและกำหนดค่าพารามิเตอร์ของขั้นตอน วิธีอานานิคมผึ้งเทียม Value I (Heterogeneous).....	75
4.43 เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและกำหนดค่าพารามิเตอร์ของขั้นตอน วิธีอานานิคมผึ้งเทียม Value II (Heterogeneous).....	75
4.44 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 เมื่อจำนวน VM เพิ่มขึ้น (Heterogeneous).....	77
4.45 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 2 เมื่อจำนวน VM เพิ่มขึ้น (Heterogeneous).....	77
4.46 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 3 เมื่อจำนวน VM เพิ่มขึ้น (Heterogeneous).....	78
4.47 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 ขนาดของงาน 5000 – 20000 (Heterogeneous).....	79
4.48 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 4 ขนาดของงาน 15000 – 35000 (Heterogeneous).....	80
4.49 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 5 ขนาดของงาน 25000 – 45000 (Heterogeneous).....	80
4.50 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 6 ขนาดของงาน 35000 – 55000 (Heterogeneous).....	81
4.51 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 7 ขนาดของงาน 45000 – 65000 (Heterogeneous).....	81

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันอินเทอร์เน็ตเข้ามามีบทบาทในการทำกิจกรรมในสังคมเพิ่มมากขึ้น โดยเฉพาะการติดต่อสื่อสารผ่านอินเทอร์เน็ตกลายเป็นสิ่งที่จำเป็นในชีวิตประจำวัน อินเทอร์เน็ตจึงกลายเป็นศูนย์กลางในการเชื่อมโยงข้อมูลต่างๆ แต่เนื่องจากผู้ใช้งานมีปริมาณเพิ่มมากขึ้นประกอบกับข้อมูลที่มีการเปลี่ยนแปลงตลอดเวลา ทำให้ภาระการจัดการงานของคอมพิวเตอร์มีปริมาณเพิ่มมากขึ้นเช่นกัน ซึ่งส่งผลให้ไม่สามารถตอบสนองความต้องการของผู้ใช้ได้อย่างทั่วถึง จากปัญหาดังกล่าว จึงได้มีการนำเทคโนโลยีการให้บริการการประมวลผลแบบกลุ่มเมฆ (Cloud Computing) มาใช้เพื่อมุ่งเน้นการให้บริการทางเครือข่ายและคอมพิวเตอร์ (Network and Computing) การจัดเก็บข้อมูล (Storage) และการบริการทรัพยากรข้อมูล (Data Service Resource) ไปด้วยกัน

การประมวลผลแบบกลุ่มเมฆมีรูปแบบการประมวลผลที่อิงตามความต้องการของผู้ใช้ ผู้ใช้สามารถระบุความต้องการไปยังซอฟต์แวร์ของระบบ จากนั้นซอฟต์แวร์จะมีการร้องขอให้ระบบจัดสรรทรัพยากรและบริการให้ตรงกับความต้องการของผู้ใช้ อีกทั้งระบบสามารถเสนอบริการให้ตรงกับความต้องการของผู้ใช้ได้ โดยผู้ใช้ไม่จำเป็นต้องทราบรูปแบบการทำงานหรือการให้บริการข้อดีของการประมวลผลแบบกลุ่มเมฆ คือ การลดความซับซ้อนของผู้ให้บริการ ประหยัดพลังงาน และลดค่าใช้จ่ายในการจัดการ เพราะระบบจะทำงานผ่านเทคโนโลยีเสมือน (Virtualization Technology) ส่งผลให้ระบบไม่ถูกจำกัดในเรื่องของขีดความสามารถในการประมวลผลในระบบต่างๆ

ในกรณีที่มีการร้องขอเพื่อเข้าใช้บริการในระบบไปยังเครื่องแม่ข่ายเดียวกัน ในบางครั้งเครื่องแม่ข่ายตัวอื่นๆ ไม่มีการเข้าใช้บริการ ซึ่งเรียกได้ว่าเกิดการกระจายงานแบบไม่สมดุลในระบบ (System Load Imbalance) ซึ่งสามารถแก้ไขโดยการทำการจัดตารางการทำงาน (Scheduling) ก่อนเข้าใช้บริการ ซึ่งการจัดตารางการทำงานที่ดีจะมีประโยชน์ในแง่ของการเพิ่มประสิทธิภาพการใช้ทรัพยากรมากที่สุด คือ เป็นการลดการสูญเสียเวลาในการรอเข้าทำงานในระบบซึ่งเป็นการลดเวลาการทำงานในระบบให้น้อยที่สุด โดยการกระจายการทำงานแบบสมดุลในระบบ (System Load Balance) เพื่อให้เกิดการสูญเสียเวลาน้อยลงและระบบสามารถทำงานได้อย่างต่อเนื่อง

การจัดตารางเข้าทำงานเพื่อเข้าใช้ทรัพยากรในระบบประมวลผลแบบกลุ่มเมฆ เพื่อให้ระบบเกิดการกระจายงานแบบสมดุลนั้น จำเป็นต้องคำนึงถึงเงื่อนไขต่างๆ เช่น ความเสถียรของระบบ ความรวดเร็วในการทำงานในระบบ การเปลี่ยนแปลงของระบบเมื่อสภาพแวดล้อมภายในระบบมี

การเปลี่ยนแปลง ฯลฯ ซึ่งจากเงื่อนไขที่ต้องคำนึงถึงในการจัดการนั้นจะมีลักษณะคล้ายคลึงกับการทำงานของขั้นตอนวิธีอาณานิคมผึ้งเทียม (Artificial Bee Colony Algorithm: ABC Algorithm) ในส่วนของพฤติกรรมการออกหาอาหารของผึ้ง ดังนั้นจึงได้เสนอขั้นตอนวิธีการจัดการตารางการทำงาน และการกระจายงานแบบสมดุลเพื่อเข้าใช้ทรัพยากรในการประมวลผลแบบกลุ่มเมฆ โดยประยุกต์ใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับวิธีการจัดลำดับงานแบบฮิวริสติก (Heuristics) เพื่อเพิ่มประสิทธิภาพในการเข้าใช้ทรัพยากรของระบบและลดเวลารวมในการรอคอยเข้าใช้ทรัพยากร

## 1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อเพิ่มประสิทธิภาพการทำงานในระบบการประมวลผลแบบกลุ่มเมฆ
2. เพื่อนำเสนอขั้นตอนวิธีการจัดการตารางเวลาเพื่อเข้าใช้ทรัพยากรเครื่องคอมพิวเตอร์เสมือน (Virtual Machine: VM) ที่มีสภาพแวดล้อมภายในเหมือนกันทั้งหมด (Homogeneous Environment) และสภาพแวดล้อมภายในต่างกันทั้งหมด (Heterogeneous Environment) ในระบบการประมวลผลแบบกลุ่มเมฆ
3. ประยุกต์ใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับวิธีการจัดลำดับงานแบบฮิวริสติก เพื่อหารูปแบบการเข้าทำงานให้เหมาะสมกับสภาพแวดล้อมภายในของระบบการประมวลผลแบบกลุ่มเมฆ
4. ลดเวลารวมในการทำงาน (Makespan) ในระบบการประมวลผลแบบกลุ่มเมฆให้น้อยลง
5. จัดลำดับการเข้าใช้ทรัพยากรเครื่องคอมพิวเตอร์เสมือน เพื่อให้มีการกระจายการทำงานในระบบแบบสมดุล (Load Balance)

## 1.3 สมมติฐานของงานวิจัย

การใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมมาประยุกต์ใช้ควบคู่กับการพิจารณาการจัดการตารางการทำงาน โดยคำนึงถึงลำดับการมาถึงและขนาดของงานในการจัดการ เพื่อเข้าใช้ทรัพยากรในระบบการประมวลผลแบบกลุ่มเมฆ จะทำให้มีการกระจายงานในระบบแบบสมดุลและช่วยลดเวลารวมของการทำงานในระบบ

## 1.4 ขอบเขตของงานวิจัย

1. จำลองระบบการประมวลผลแบบกลุ่มเมฆเป็นแบบเฉพาะที่ (Private Cloud) โดยใช้ CloudSim Toolkit 3.0.1
2. งานวิจัยนี้ได้เสนอขั้นตอนวิธีการจัดตารางเวลาเข้าทำงานเพื่อเข้าใช้ทรัพยากรเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกันและต่างกันในการประมวลผลแบบกลุ่มเมฆ โดยการนำขั้นตอนวิธีอาณานิคมผึ้งเทียมมาประยุกต์ใช้ควบคู่กับวิธีการจัดลำดับงานแบบฮิวริสติก โดยพิจารณาจากลำดับการมาถึงของงานและพิจารณาจากขนาดของงาน คือ งานที่มาก่อนเข้าทำก่อน (FCFS) งานที่มีขนาดใหญ่ที่สุดเข้าทำงานก่อน (LJF) และงานที่มีขนาดเล็กที่สุดเข้าทำก่อน (SJF)
3. กำหนดรูปแบบการใช้หน่วยประมวลผลกลาง (Central Processing Unit: CPU) เป็นแบบที่ไม่สามารถขัดจังหวะหรือแทรกกลางกันได้ (Non-Preemptive Process)
4. ทำการศึกษาการจัดลำดับการทำงานภายใต้สมมติฐานที่ว่า เครื่องจักร ไม่มีการเสีย (Down Time) ระหว่างดำเนินการ
5. เครื่องคอมพิวเตอร์เสมือนทุกเครื่องอยู่ในสถานะพร้อมทำงาน ไม่มีค้างในสถานะรอทำงาน และเครื่องเริ่มทำงาน ณ เวลาที่ 0 วินาที
6. ชุดข้อมูลที่ใช้ในการทดสอบมี 4 รูปแบบ คือ ชุดข้อมูลแบบสุ่มอิสระ ชุดข้อมูลที่มีการกระจายแบบโค้งปกติ ชุดข้อมูลที่มีการกระจายตัวแบบเบ้ซ้ายและชุดข้อมูลที่มีการกระจายตัวแบบเบ้ขวา

## 1.5 ขั้นตอนการศึกษาและดำเนินงานวิจัย

1. ศึกษาและวิเคราะห์รูปแบบการประมวลผลแบบกลุ่มเมฆ
2. ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการประมวลผลแบบกลุ่มเมฆ วิธีการจัดการการเข้าใช้งานระบบ และขั้นตอนวิธีอาณานิคมผึ้งเทียม
3. ศึกษาเครื่องมือที่ใช้ในการจำลองการประมวลผลแบบกลุ่มเมฆ
4. ตั้งสมมติฐานเกี่ยวกับการประยุกต์ขั้นตอนวิธีอาณานิคมผึ้งเทียมในการจัดการการเข้าใช้งานทรัพยากรเครื่องคอมพิวเตอร์เสมือนของระบบการประมวลผลแบบกลุ่มเมฆ จะทำให้เกิดการกระจายงานในระบบแบบสมดุลและช่วยลดเวลาการทำงานในระบบ ในกรณีที่เครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมเหมือนกันและต่างกัน
5. ออกแบบขั้นตอนวิธีการจัดการการเข้าใช้ทรัพยากรในระบบการประมวลผลแบบกลุ่มเมฆ โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาจากลำดับการมาถึงของงาน ขนาดของงาน และสภาพแวดล้อมภายในระบบการประมวลผลแบบกลุ่มเมฆ

6. พัฒนาโปรแกรมและทดลองตามกระบวนการที่นำเสนอ
7. วิเคราะห์ผลการทดลอง
8. สรุปผลการทดลองพร้อมเสนอแนวทางการพัฒนางานวิจัย
9. เขียนวิทยานิพนธ์

## 1.6 ประโยชน์ที่ได้รับจากงานวิจัย

1. สามารถจัดตารางเวลาเพื่อเข้าใช้ทรัพยากรเครื่องคอมพิวเตอร์เสมือนที่มีคุณสมบัติของสภาพแวดล้อมภายในที่เหมือนกันและต่างกันของระบบการประมวลผลแบบกลุ่มเมฆได้
2. ทราบถึงขั้นตอนวิธีการประยุกต์ใช้ขั้นตอนวิธีอาณานิคมฝังเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก เพื่อใช้ในการจัดการการทำงานของระบบการประมวลผลแบบกลุ่มเมฆ
3. สามารถลดเวลารวมของการเข้าใช้งานทรัพยากรในระบบการประมวลผลแบบกลุ่มเมฆ
4. สามารถประยุกต์ใช้ขั้นตอนวิธีอาณานิคมฝังเทียมในการกระจายการเข้าทำงานให้กับทรัพยากรเครื่องคอมพิวเตอร์เสมือน เพื่อให้ระบบสามารถกระจายงานได้แบบสมดุล
5. ได้แบบจำลองที่ช่วยในการวางแผนการเข้าใช้เครื่องคอมพิวเตอร์เสมือนในการประมวลผลแบบกลุ่มเมฆ
6. สามารถประเมินสถานการณ์ของการเข้าใช้เครื่องคอมพิวเตอร์เสมือนให้ทราบล่วงหน้าได้

## บทที่ 2

# ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้ได้แบ่งเนื้อหาออกเป็น 5 ส่วน ได้แก่ การประมวลผลแบบกลุ่มเมฆ เทคโนโลยีการจำลองเครื่องคอมพิวเตอร์เสมือน ขั้นตอนวิธีอาณานิคมฟังก์ชันเชิงควม วิธีการจัดลำดับงานแบบฮิวริสติก เครื่องมือที่ใช้ในการสร้างแบบจำลอง CloudSim-3.0.1 การวัดการกระจายข้อมูลและงานวิจัยที่เกี่ยวข้อง

### 2.1 การประมวลผลแบบกลุ่มเมฆ

#### 2.1.1 หลักการของการประมวลผลแบบกลุ่มเมฆ

การประมวลผลแบบกลุ่มเมฆ (Cloud Computing) [1-2] คือ ระบบคอมพิวเตอร์ที่มีการประมวลผลอยู่ในระบบอินเทอร์เน็ต บนรูปแบบของโครงสร้างการประมวลผลขนาดใหญ่ที่ทำงานร่วมกัน มีการแบ่งปันทรัพยากรในการประมวลผลร่วมกันบนเครือข่ายอินเทอร์เน็ต ราวกับว่าเป็นหน่วยประมวลผลเดียวกัน



รูปที่ 2.1 หลักการทำงานของระบบการประมวลผลแบบกลุ่มเมฆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.1 แสดงถึงตัวอย่างการใช้งานระบบการประมวลผลแบบกลุ่มเมฆ ปัจจุบันมีการให้นิยามจากผู้พัฒนาแนวคิดนี้ได้แก่

บริษัท Gartner ได้ให้นิยามว่า “Cloud computing is a style of computing where massively scalable IT-related capabilities are provided ‘as a service’ across the Internet to multiple external customers” หรือ ระบบการประมวลผลแบบกลุ่มเมฆ คือ แนวทางการประมวลผลที่พลังของโครงสร้างทางไอทีขนาดใหญ่ที่ขยายตัว ได้ถูกนำเสนออย่างลูกค้าภายนอกจำนวนมากในรูปแบบของบริการ

ฟอร์สเตอร์กรุป ได้ให้นิยามว่า “Cloud computing: A pool of abstracted, highly scalable, and managed infrastructure capable of hosting end-customer applications and billed by consumption” หรือ กลุ่มของโครงสร้างพื้นฐานที่ถูกบริหารจัดการและขยายตัวได้อย่างมาก ซึ่งมีขีดความสามารถในการรองรับโปรแกรมประยุกต์ต่างๆ ของผู้ใช้และเก็บค่าบริการตามการใช้งาน

การประมวลผลแบบกลุ่มเมฆจะมีรูปแบบการประมวลผลที่อิงกับความต้องการของผู้ใช้ โดยผู้ใช้สามารถระบุความต้องการไปยังซอฟต์แวร์ของระบบการประมวลผลแบบกลุ่มเมฆ จากนั้นซอฟต์แวร์จะร้องขอให้ระบบจัดสรรทรัพยากรและบริการให้ตรงกับความต้องการของผู้ใช้งาน ทั้งนี้ระบบสามารถเพิ่มและลดจำนวนของทรัพยากร รวมถึงเสนอบริการให้พอเหมาะกับความ ต้องการของผู้ใช้ได้ตลอดเวลา โดยที่ผู้ใช้ไม่จำเป็นต้องทราบการทำงานหรือเหตุการณ์เบื้องหลัง เป็นเช่นไร

การประมวลผลแบบกลุ่มเมฆเป็นเทคโนโลยีที่พัฒนาขึ้นมา เพื่อตอบสนองรูปแบบการทำงานและการใช้งานระบบสารสนเทศผ่านเครือข่ายอินเทอร์เน็ต โดยเน้นไปที่การจัดทำระบบประมวลผลและ โครงสร้างพื้นฐานคอมพิวเตอร์ขนาดใหญ่ เพื่อรองรับการใช้งานของผู้ใช้จำนวนมากผ่านโปรแกรมประยุกต์ที่ทำงานแบบบริการด้านซอฟต์แวร์ (Software as a Service: SaaS) ผู้ใช้งานการประมวลผลแบบกลุ่มเมฆไม่จำเป็นต้องรับภาระการดำเนินการจัดทำระบบคอมพิวเตอร์ หรือการขยายระบบเมื่อองค์กรมีจำนวนผู้ใช้งานมากขึ้น ซึ่งจะเป็นการลดต้นทุนในส่วนของคอมพิวเตอร์ประมวลผลและค่าบำรุงรักษา แนวคิดของระบบการประมวลผลแบบกลุ่มเมฆนี้ เปรียบเสมือนกับการบริการไฟฟ้า โดยมองการบริการประมวลผลด้วยคอมพิวเตอร์เป็นเหมือน โครงสร้างสาธารณูปโภคพื้นฐานที่องค์กรหรือหน่วยงานสามารถใช้งานได้โดยไม่มีขอบเขต [1-3]

นิยามความหมายของคำหลักๆ 3 คำที่เกี่ยวข้องกับการประมวลผลแบบกลุ่มเมฆมีดังนี้

**1. ความต้องการ (Requirement)** คือ โจทย์ปัญหาที่ผู้ใช้ต้องการให้ระบบคอมพิวเตอร์ แก้ไขปัญหาหรือตอบปัญหาตามที่ผู้ใช้กำหนดได้ ยกตัวอย่างเช่น ความต้องการพื้นที่จัดเก็บข้อมูล ขนาด 1,000,000 GB ความต้องการประมวลผล โปรแกรมแบบขนานเพื่อค้นหาพยากรณ์โรคไข้หวัดนกให้ได้สูตรยาภายใน 90 วัน ความต้องการโปรแกรมและพลังการประมวลผลสำหรับสร้าง ภาพยนตร์แอนิเมชันความยาว 2 ชั่วโมงให้แล้วเสร็จภายใน 4 เดือน และความต้องการค้นหาข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ท่องเที่ยวและโปรแกรมทัวร์ในประเทศอิตาลีในราคาที่ถูกที่สุดในโลก แต่ปลอดภัยในการเดินทางด้วย เป็นต้น

2. **ทรัพยากร (Resource)** หมายถึง บั๊กจ็อบหรือสิ่งที่เกี่ยวข้องกับการประมวลผล หรือเกี่ยวข้องกับการแก้ไขปัญหาตาม โจทย์ที่ความต้องการของผู้ใช้ได้ระบุไว้ อาทิเช่น หน่วยประมวลผลกลาง (Central Processing Unit: CPU) หน่วยความจำ เช่น แรม (Random Access Memory: RAM) อุปกรณ์จัดเก็บข้อมูล เช่น ฮาร์ดดิสก์ (Harddisk) ฐานข้อมูล (Database) สารสนเทศ (Information) ข้อมูล (Data) เครือข่าย (Network) ซอฟต์แวร์ประยุกต์ (Application Software) เซ็นเซอร์ที่สามารถรับส่งสัญญาณอินพุตและเอาต์พุต (Remote Sensor) เป็นต้น

3. **บริการ (Service)** ถือว่าเป็นทรัพยากรและในทางกลับกันก็สามารถบอกได้ว่า ทรัพยากรก็คือบริการ โดยเฉพาะอย่างยิ่งในด้านการประมวลผลแบบกลุ่มเมฆแล้ว สามารถจะใช้คำว่าบริการ แทนคำว่าทรัพยากร คำว่าบริการหมายถึงการกระทำ (Operation) เพื่อให้เกิดผลลัพธ์ที่สนองต่อ ความต้องการ (Requirement) แต่การกระทำของบริการจะเกิดขึ้นได้จำเป็นต้องพึ่งพา ทรัพยากร โดยการใช้ทรัพยากรที่เกี่ยวข้องเพื่อแก้ปัญหาให้เกิดผลลัพธ์สนองต่อความต้องการ

ระบบประมวลผลกลุ่มเมฆ [3] ได้รับความสนใจอย่างมาก เนื่องจากความสามารถของการขยายหรือลดขนาดของระบบ และความยืดหยุ่นโดยปราศจากการลงทุนในเรื่องของการติดตั้ง และการดำเนินงานในส่วนของการวางโครงสร้างพื้นฐานทางด้านเทคโนโลยีสารสนเทศและศูนย์รวมข้อมูล กระบวนการของระบบประมวลผลกลุ่มเมฆนี้คาดว่าจะทำให้องค์กรสามารถขยายหรือลดขนาดของระบบในส่วนและเทคโนโลยีพื้นฐานขององค์กรได้ตามความต้องการ

ผู้คนส่วนใหญ่มักสับสนเกี่ยวกับความหมายของระบบประมวลผลกลุ่มเมฆ เพราะมักให้ความคิดเห็นที่แตกต่างกันออกไป ทั้งผู้ที่ริเริ่มทำกลุ่มเมฆและนักวิจัยหลายท่านจะใช้คำนิยามของระบบประมวลผลกลุ่มเมฆที่นิยามโดย US National Institute of Standards and Technology (NIST) นั่นคือ

“ระบบประมวลผลกลุ่มเมฆคือโมเดลสำหรับความสะดวก (ให้บริการ) ผ่านระบบเครือข่ายเพื่อแบ่งปันการใช้ทรัพยากรที่มีอยู่ (เช่น เครือข่าย เครื่องแม่ข่าย หน่วยความจำ แอปพลิเคชัน และบริการ) ซึ่งสามารถจัดหาและใช้งานได้อย่างรวดเร็วด้วยการจัดการเพียงเล็กน้อยหรือเพียงแคโต้ตอบกับผู้ใช้บริการ” [3]

### 2.1.2 โครงสร้างการประมวลผลแบบกลุ่มเมฆ

สถาปัตยกรรมในมุมมองการตลาดของการประมวลผลในกลุ่มเมฆ ประกอบไปด้วยองค์ประกอบพื้นฐาน 4 ส่วน [4] ดังนี้

1. **ผู้ใช้ (User/Broker)** ในกรณีนี้หมายถึงผู้ที่กระทำการส่งคำร้องขอใช้บริการซึ่งจะถูกส่งมาจากที่ใดก็ได้ในโลก

2. **ข้อตกลงในการจัดสรรทรัพยากรในการให้บริการของผู้ให้บริการ (Service Level Agreement (SLA) Resource Allocator)** ที่เปรียบเสมือนส่วนเชื่อมต่อประสาน (Interface) ระหว่างผู้บริการประมวลผลในกลุ่มเมฆกับผู้ใช้บริการที่อยู่นอกกลุ่มเมฆ การทำงานข้อตกลงในการจัดสรรทรัพยากรในการให้บริการของผู้ให้บริการ ประกอบไปด้วย 6 ส่วนย่อยดังนี้

1) **Service Request Examiner and Admission Control:** ทำหน้าที่ในการแปลความหมายของคำร้องขอใช้บริการที่ถูกส่งมาจากผู้ใช้บริการ คำร้องขอใช้บริการจะถูกนำไปเปรียบเทียบกับคุณภาพของบริการ (Quality of Service: QoS) ซึ่งเป็นข้อตกลงในการใช้บริการ (Service Level Agreement: SLA) ที่ได้กระทำไว้ล่วงหน้า เพื่อพิจารณาว่า คำร้องขอใช้บริการจะได้รับการยอมรับหรือปฏิเสธ รวมทั้งการจัดสรรทรัพยากรประมวลผลอย่างมีประสิทธิภาพโดยไม่โหลตมากเกินไป จากนั้นจึงส่งคำร้องขอใช้บริการไปยังเครื่องคอมพิวเตอร์เสมือน (Virtual Machine: VM) และกำหนดเครื่องสำหรับการจัดการเครื่องคอมพิวเตอร์เสมือน

2) **Pricing:** กลไกในส่วนนี้มีไว้เพื่อกำหนดค่าใช้จ่ายในการใช้บริการตามเงื่อนไขของการใช้บริการ เช่น เวลาที่ใช้คำร้องขอใช้บริการเป็นช่วงเวลาสูงสุดหรือไม่ อัตราค่าบริการเป็นอัตราคงที่หรือผันแปร เป็นต้น

3) **Accounting:** เป็นกลไกสำหรับใช้บรรจุข้อมูลบนร่องของงานบันทึก (Track) การใช้ทรัพยากรที่ถูกร้องขอและมีการคิดค่าใช้จ่ายกับผู้ใช้บริการ กลไกนี้จะมีหน้าที่บันทึกประวัติการใช้งานทรัพยากรต่างๆ เพื่อนำไปปรับปรุงการจัดสรรทรัพยากรในการทำทรัพยากรเสมือน

4) **Virtual Machine Monitor:** เป็นกลไกที่เก็บบันทึกข้อมูลบนร่องของงานบันทึกของเครื่องคอมพิวเตอร์เสมือนที่จัดให้พร้อมใช้งานรวมทั้งทรัพยากรที่นำมาจัดเป็นเครื่องคอมพิวเตอร์เสมือน

5) **Dispatcher:** เป็นโปรแกรมเลือกจ่ายงานซึ่งถูกใช้เป็นกลไกในการเริ่มต้นการทำงานตามคำร้องขอบริการที่ได้รับการยอมรับให้ทำงานบนเครื่องเสมือนที่จัดสรรไว้กับงานนั้นๆ

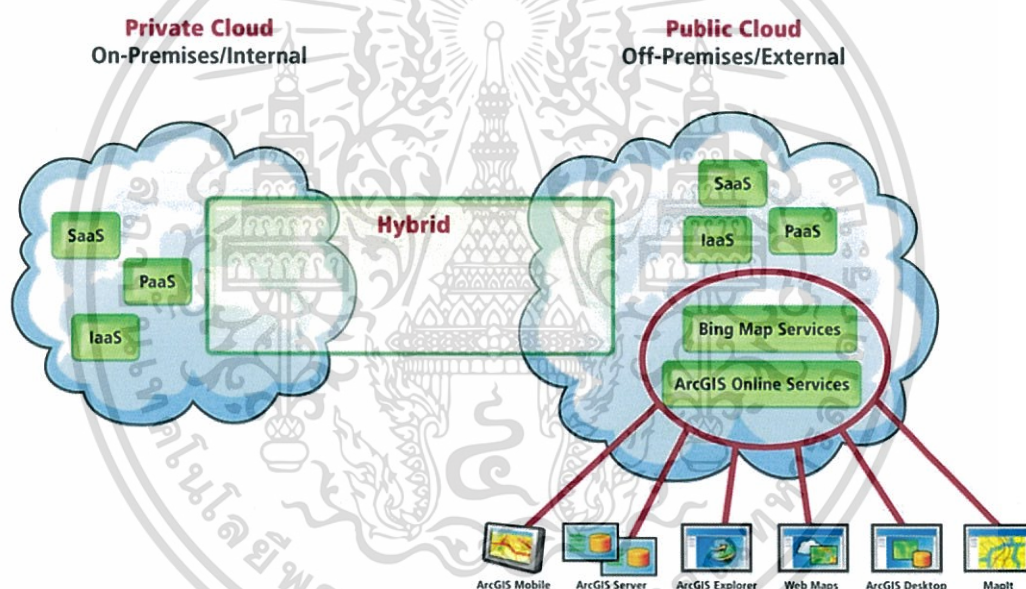
6) **Service Request Monitor:** เป็นกลไกในการเก็บรักษาข้อมูลบนร่องของงานบันทึกของความก้าวหน้าในการดำเนินงานตามคำร้องขอบริการ

3. เครื่องคอมพิวเตอร์เสมือน (Virtual Machine: VM) เครื่องคอมพิวเตอร์ที่มีอยู่จริงหนึ่งเครื่องสามารถจัดเป็นเครื่องคอมพิวเตอร์เสมือนได้หลายเครื่อง ทั้งนี้ขึ้นอยู่กับระดับค่าธรรมเนียมบริการ เครื่องคอมพิวเตอร์เสมือนหลายๆ เครื่องสามารถดำเนินงานตามชุดคำสั่งงานได้พร้อมกันบนเครื่องคอมพิวเตอร์ที่มีอยู่จริงเพียงเครื่องเดียว ทั้งนี้เพราะว่าเครื่องคอมพิวเตอร์เสมือนแต่ละเครื่องจะทำงานเป็นอิสระจากกัน แม้จะอยู่บนเครื่องคอมพิวเตอร์ที่มีอยู่จริงเครื่องเดียวกัน

4. เครื่องคอมพิวเตอร์ที่มีอยู่จริง (Physical Machines) ประกอบไปด้วยเครื่องแม่ข่าย (Server) จำนวนมากที่เตรียมไว้สำหรับให้บริการ

### 2.1.3 ประเภทของการประมวลผลแบบกลุ่มเมฆ

ระบบประมวลผลกลุ่มเมฆสามารถแบ่งตามขอบเขตการใช้งาน [3] แสดงดังรูปที่ 2.2 โดยสามารถแบ่งออกเป็น 3 ประเภทด้วยกัน คือ



รูปที่ 2.2 ระบบการประมวลผลแบบกลุ่มเมฆแบ่งตามขอบเขตการใช้งาน [3]

#### 1. แบบเฉพาะที่ (Private Cloud)

บริการของผู้ให้บริการที่นำการประมวลผลแบบกลุ่มเมฆไปทำงานบนเครือข่ายส่วนบุคคล (Private Network) ของผู้ให้บริการหรือเครือข่ายที่เปิดให้ใช้บริการเฉพาะผู้ให้บริการเฉพาะรายเป็นรายๆ ไปบนหลักการของเทคโนโลยีการจำลองเครื่องคอมพิวเตอร์เสมือน (Virtualization Technology) ผู้ให้บริการสามารถควบคุมและจัดการระบบได้ด้วยตนเอง การประมวลผลแบบกลุ่มเมฆแบบเฉพาะที่สามารถแก้ปัญหาในเรื่องของความเสถียรและความน่าเชื่อถือในการใช้งานระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. แบบสาธารณะ (Public Cloud)

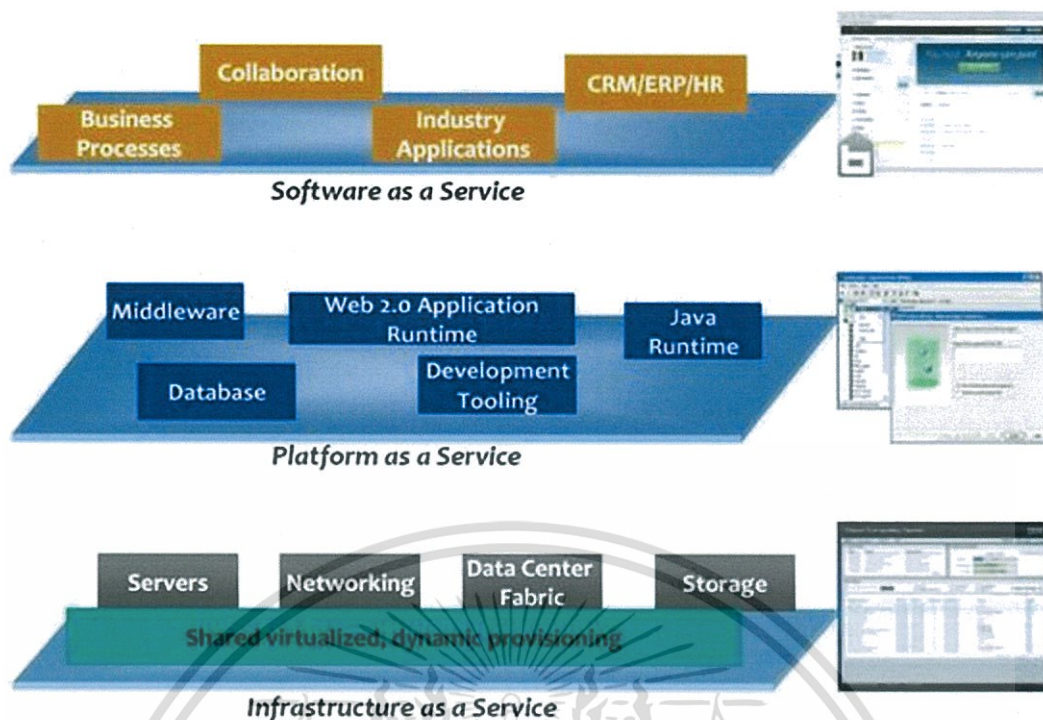
การประมวลผลกลุ่มเมฆแบบสาธารณะ เป็นการประมวลผลผ่านบริการทางเว็บจากผู้ให้บริการต่อผู้ให้บริการโดยทั่วไป ตามหลักการของคลาวด์ ที่จัดให้มีการแบ่งปันการใช้ทรัพยากรในการประมวลผลศูนย์ข้อมูล คำสั่งงานประยุกต์ด้วยวิธีการของเทคโนโลยีการจำลองเครื่องคอมพิวเตอร์เสมือน โดยที่ผู้ให้บริการสามารถระบุความต้องการได้ด้วยตนเองและจ่ายค่าบริการตามปริมาณการใช้งานจริง เช่น กลุ่มเมฆของ Amazon เป็นต้น

## 3. แบบผสม (Hybrid Cloud)

การนำเอารูปแบบของการประมวลผลแบบกลุ่มเมฆแบบเฉพาะที่และการประมวลผลแบบกลุ่มเมฆแบบสาธารณะมาผสมผสานกัน นั่นคือการใช้ทรัพยากรที่มีอยู่มาสร้างเป็นกลุ่มเมฆขึ้นเพื่อใช้ภายในองค์กร แต่เนื่องจากทรัพยากรที่มีอยู่นั้น ไม่เพียงพอกับระบบทั้งหมด จึงได้มีการใช้บริการของกลุ่มเมฆแบบสาธารณะมาร่วมด้วยซึ่งจะทำให้สามารถเพิ่มทรัพยากรขึ้นพื้นฐานได้เพียงพอกับระบบที่จะนำมาใช้

### 2.1.4 การให้บริการของระบบการประมวลผลแบบกลุ่ม

ทุกสิ่งทุกอย่างที่อยู่ในคลาวด์จะถูกนำเสนอในรูปแบบของบริการ ไม่ว่าจะเป็นทรัพยากรพื้นฐาน เช่น แหล่งเก็บข้อมูล หน่วยความจำ หน่วยประมวลผล รวมทั้งเครื่องมือที่ใช้ในการพัฒนาและจัดการแอปพลิเคชัน ตัวแอปพลิเคชันเอง หรือแม้แต่ข้อมูลต่างๆ [5] ซึ่งการที่ทุกสิ่งทุกอย่างในระบบการประมวลผลแบบกลุ่มเมฆนั้นมีรากฐานมาจากเซอร์วิส (Services) จะเห็นได้ว่ามีผู้เชี่ยวชาญจำนวนมากมองว่า การประมวลผลแบบกลุ่มเมฆเป็นเทคโนโลยีที่พัฒนามาจากเว็บเซอร์วิส (Web services) ซึ่งมุ่งเน้นในเรื่องของการให้บริการข้อมูลและแอปพลิเคชันในลักษณะของเซอร์วิสบนอินเทอร์เน็ตนั่นเอง [6] แต่เมื่อพัฒนามาเป็นการประมวลผลแบบกลุ่มเมฆบริการที่ถูกนำเสนอทางอินเทอร์เน็ตนั้น ไม่ได้ถูกจำกัดอยู่ที่ข้อมูลและแอปพลิเคชันอีกต่อไป แต่บริการที่ถูกนำเสนอในการประมวลผลแบบกลุ่มเมฆสามารถแบ่งออกได้เป็น 3 กลุ่มใหญ่ๆ ได้แก่ Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) และ Infrastructure-as-a-Service (IaaS) ดังแสดงในรูปแบบที่ 2.3 [3], [7]



รูปที่ 2.3 การให้บริการของระบบการประมวลผลแบบกลุ่มเมฆ [3]

### 1. บริการด้านโครงสร้างฮาร์ดแวร์พื้นฐาน (Infrastructure as a Service : IaaS)

เป็นบริการโครงสร้างพื้นฐานคอมพิวเตอร์ที่สามารถเข้าถึงได้ผ่านทางอินเทอร์เน็ต ผู้ให้บริการโครงสร้างพื้นฐาน (Infrastructure Provider: IP) จัดทรัพยากรประมวลผลไม่ว่าจะเป็น ส่วนของการจัดเก็บ (Storage) หรือประมวลผลผ่านระบบเสมือนที่ผู้ใช้บริการสามารถกำหนดขนาดของทรัพยากรให้ตรงกับความต้องการที่เปลี่ยนแปลงไป ผู้ให้บริการประเภทนี้มีทั้งผู้ใช้บริการทั่วไปและผู้ที่ใช้บริการ ในระดับโครงสร้างพื้นฐานเหมาะสมกับองค์กรที่ไม่ต้องการลงทุนทางด้านฮาร์ดแวร์ (Hardware) ซึ่งก็คือ การให้บริการในด้านต่างๆ ดังนี้

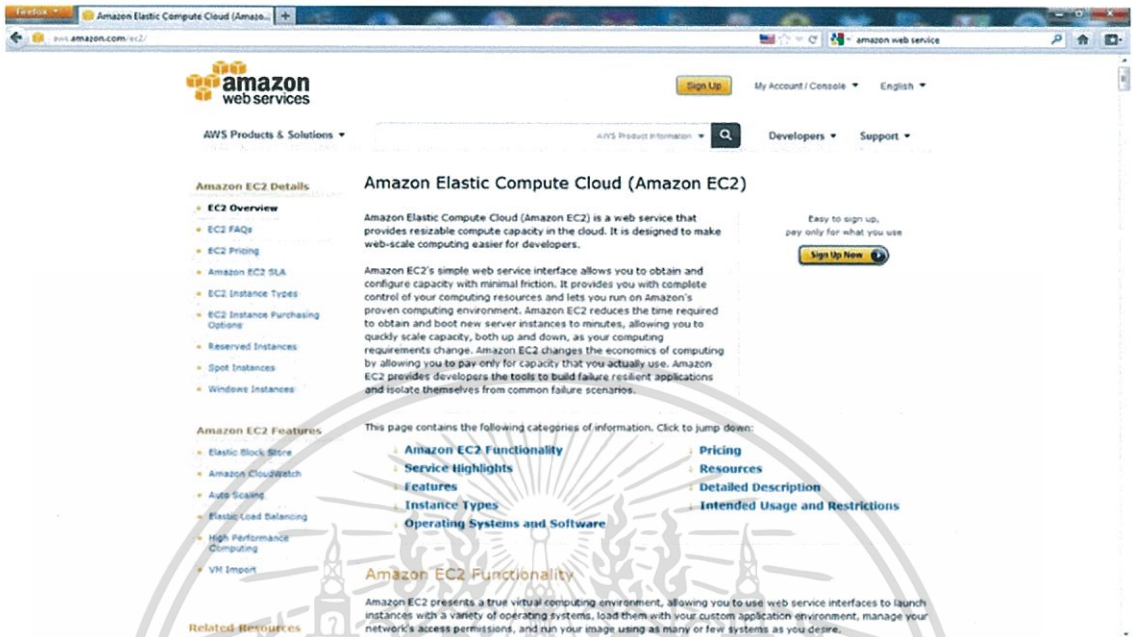
1) ทรัพยากร (Resource) ต่างๆ ในรูปของบริการ เช่น พวงเครื่องแม่ข่าย (Server) หน่วยความจำ (Memory) หน่วยประมวลผลกลาง (CPU) พื้นที่เก็บข้อมูลบนฮาร์ดดิสก์ (Disk Space) หรือ อุปกรณ์เครือข่าย (Network Equipment) เป็นต้น

2) การขยายขนาดของโครงสร้างพื้นฐานซึ่งสามารถทำให้เล็กหรือใหญ่ได้ขึ้นอยู่กับความต้องการของแอปพลิเคชัน

3) การกำหนดราคาในการให้บริการนั้นจะขึ้นอยู่กับทางเลือกใช้บริการทรัพยากร ความสามารถในการขยายหรือลดขนาดของระบบ (Scalability) คือคุณสมบัติที่โดดเด่นของระบบประมวลผลกลุ่มเมฆ บริการในระดับ IaaS นั้นจะตอบสนองความต้องการในเรื่องของทรัพยากรด้านพื้นที่ในการเก็บข้อมูล (Storage Resource) เช่น Amazon's Simple Storage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Service และความยืดหยุ่นของการใช้ทรัพยากร เช่น GoGrid, RightScale, Linode และ Amazon Compute Elastic (EC2) ดังรูปที่ 2.4



รูปที่ 2.4 ตัวอย่างผู้ให้บริการในระดับโครงสร้างฮาร์ดแวร์พื้นฐาน เช่น Amazon EC2 [8]

## 2. บริการด้านแพลตฟอร์ม (Platform as a Service: PaaS)

การให้บริการแบบ PaaS นั้นจะคล้ายกับการให้บริการ IaaS ซึ่งผู้ใช้ต้องมีการเช่าฮาร์ดแวร์ไปด้วย แต่ IaaS จะมีอิสระในการเลือกใช้งานมากกว่า ซึ่งการให้บริการ PaaS นั้นผู้ให้บริการจะมีการสร้างชุด API (Application Programming Interface) ขึ้นเพื่อให้ลูกค้านำไปใช้และใช้งานอยู่บนแพลตฟอร์มของผู้ให้บริการ จุดที่เป็นข้อดีในมุมมองของลูกค้าคือ API นั้นให้ความสะดวกผ่านการทดสอบแล้ว และมีการปรับปรุงให้ดีขึ้นเรื่อยๆ นอกจากนั้นยังไม่ต้องกังวลกับฮาร์ดแวร์หรือการปรับเพิ่มลดขนาดของระบบตามจำนวนผู้ใช้ที่เพิ่มมากขึ้น เพราะ PaaS จะทำให้เองอัตโนมัติ แต่มีข้อเสียในส่วนที่ว่าถ้าผู้ใช้เขียนโปรแกรมโดยผูกติดกับ API ของผู้ให้บริการ PaaS รายใดรายหนึ่ง หากโปรแกรมมีความซับซ้อนมาก ก็ยากที่จะย้ายไปใช้งานที่อื่นได้ จากรูปที่ 2.5 แสดงถึงตัวอย่างผู้ให้บริการในระดับแพลตฟอร์ม เช่น Google Apps, Heroku Mosso และ Engine Yard เป็นต้น

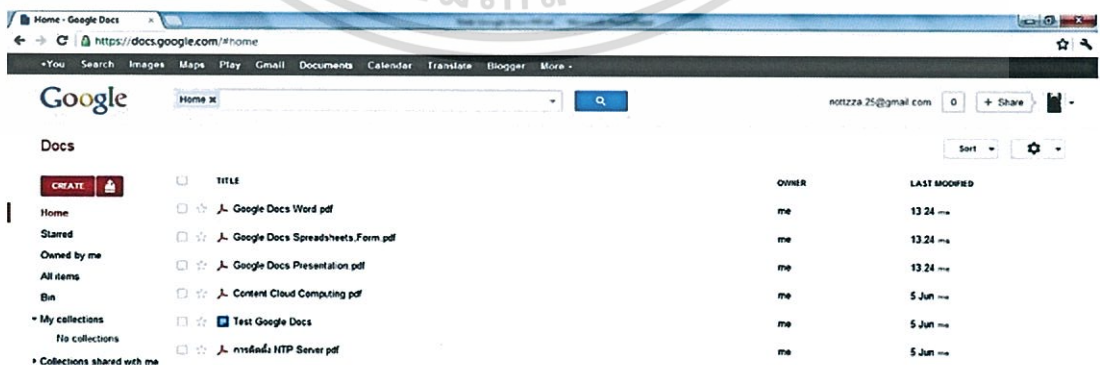
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ตัวอย่างผู้ให้บริการในระดับแพลตฟอร์ม เช่น Google Apps [8]

### 3. บริการด้านซอฟต์แวร์ (Software as a Service: SaaS)

การให้บริการด้านแอปพลิเคชันซึ่งใครก็สามารถเข้ามาใช้ได้ โดยที่ผู้ใช้ไม่จำเป็นต้องเป็นเจ้าของ และไม่จำเป็นต้องรู้ว่าแอปพลิเคชันนั้นถูกติดตั้งที่ใด หรือมีการใช้ทรัพยากรอะไรบ้าง ทางผู้ให้บริการจะเป็นผู้จัดเตรียมทั้งหมด ผู้ใช้นั้นเพียงแค่อเข้าไปใช้งานเท่านั้น และผู้ให้บริการก็จะคิดค่าบริการตามการใช้งานจริงของผู้ใช้แอปพลิเคชัน ในระดับแอปพลิเคชันนั้นมีซอฟต์แวร์ ให้เลือกใช้อย่างมากมาย โดยไม่ต้องใช้เงินลงทุนจำนวนมาก นั่นคือการให้บริการซอฟต์แวร์ในรูปแบบเซอร์วิสเล็กๆ ที่ทำหน้าที่เฉพาะทาง ทำงานตามฟังก์ชันที่กำหนดไว้ ไม่ได้ทำงานเหมือนแอปพลิเคชันใหญ่ๆ ที่มีความสามารถมากมายรวมอยู่ในตัวเดียว โดยการใช้งานนั้น ผู้ใช้ SaaS เองไม่จำเป็นต้องเป็นเจ้าของเซิร์ฟเวอร์นั้น เพราะจะสามารถเลือกใช้บริการเซิร์ฟเวอร์ของผู้ให้บริการก็ได้ในลักษณะของการเช่า หรือสมัครเป็นสมาชิกเพื่อใช้งานเซิร์ฟเวอร์นั้นๆ ผู้ให้บริการมีหน้าที่บริการและสร้างเซิร์ฟเวอร์ใหม่ๆ ขึ้นมา และดูแลระบบต่างๆ ให้สามารถให้บริการตัวเซิร์ฟเวอร์นั้นได้ตามความต้องการของลูกค้า การให้บริการแอปพลิเคชันต่างๆ เช่น Google Apps หรือ การให้บริการ E-mail ดังรูปที่ 2.6



รูปที่ 2.6 ตัวอย่างผู้ให้บริการในระดับซอฟต์แวร์ เช่น การให้บริการ E-mail

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.1.5 ประโยชน์ของการประมวลผลกลุ่มเมฆ

การประมวลผลแบบกลุ่มเมฆมีประโยชน์หลายประการดังนี้ [3], [7]

### 1. ทรัพยากรไอทีถูกใช้อย่างคุ้มค่า

เนื่องจากการแบ่งปัน (Share) ทรัพยากรทั้งส่วนจัดเก็บข้อมูล แอปพลิเคชันและการประมวลผลบนเครื่องอินเทอร์เน็ต จึงเกิดการขนาดที่ประหยัด (Economies of Scale) กล่าวคือคั้งนั้นผู้ให้บริการจึงสามารถนำเสนอบริการแอปพลิเคชันในราคาที่ต่ำกว่ากรณีที่ใช้จะลงทุนติดตั้งแอปพลิเคชันเพื่อใช้งาน นอกจากนี้ยังมีความสามารถในการปรับลดขยายขนาดของประสิทธิภาพของระบบประมวลผลได้ ตามลักษณะการใช้งานจริง ผู้ใช้จึงไม่จำเป็นต้องลงทุนติดตั้งระบบไอทีเพื่อสำรองการใช้งานในบางช่วงเวลา แต่สามารถเรียกใช้ความสามารถจากผู้ให้บริการ สำหรับผู้ให้บริการการสำรองความสามารถไว้ให้บริการจะสามารถแบ่งปันความสามารถดังกล่าวให้แก่ลูกค้าหลายรายจึงเกิดการใช้ทรัพยากรอย่างเหมาะสม

### 2. ผู้ใช้มีโอกาสเข้าถึงเทคโนโลยีได้มากขึ้น

โดยอัตราค่าบริการด้านไอทีผ่านสภาพแวดล้อมการประมวลผลแบบกลุ่มเมฆ จะต่ำกว่าการลงทุนติดตั้งระบบไอทีเองทั้งหมด จึงมีโอกาสที่จะเข้าถึงเทคโนโลยีที่หลากหลาย ตลอดจนเทคโนโลยีใหม่ๆ ได้มากกว่าในสภาพแวดล้อมแบบเดิม ทั้งนี้ในกรณีที่มีการพัฒนาเทคโนโลยีใหม่ ผู้ให้บริการมีความยืดหยุ่นที่จะนำเทคโนโลยีใหม่มาให้บริการบนสถาปัตยกรรม SOA (Service-Oriented Architecture)

### 3. ผู้พัฒนาเทคโนโลยีไอทีมีโอกาสให้นำผลงานออกสู่ตลาดได้มากขึ้น

บนสถาปัตยกรรมที่แยกเซิร์ฟเวอร์ในส่วนของแอปพลิเคชัน ทำให้แอปพลิเคชันบนเครือข่ายมีความยืดหยุ่นสูง ผู้ให้บริการจะกำหนดแพลตฟอร์มของแอปพลิเคชันบนกลุ่มเมฆของตน เช่น แพลตฟอร์มสำหรับ Google App Engine เพื่อเป็นมาตรฐานสำหรับผู้พัฒนาแอปพลิเคชันสามารถนำผลงานไปให้บริการบนกลุ่มเมฆ โดยการเพิ่มบริการแอปพลิเคชันจะไม่กระทบในส่วนของผู้ใช้

### 4. ผู้ให้บริการสามารถบริหารจัดการได้อย่างมีประสิทธิภาพ

บริการถูกนำเสนอบนเครือข่ายอินเทอร์เน็ต คั้งนั้นผู้ให้บริการสามารถติดตั้งอุปกรณ์ในทำเลที่ตั้งที่ห่างไกลซึ่งมีต้นทุนต่ำกว่าการตั้งกิจการในย่านธุรกิจ แต่สามารถเสนอบริการให้แก่ลูกค้าได้ทั่วโลก อีกทั้งหน่วยจัดเก็บข้อมูล หน่วยประมวลผล และแอปพลิเคชันต่างๆ จะถูกเก็บแยกออกจากส่วนผู้ใช้ คั้งนั้นผู้ให้บริการจึงสามารถดูแลรักษาระบบได้อย่างมีประสิทธิภาพ

## 2.1.6 ข้อจำกัดของระบบประมวลผลกลุ่มเมฆ

แม้ว่าหลักการของระบบประมวลผลกลุ่มเมฆ จะมีประโยชน์ต่อภาพรวมทั้งส่วนของ ผู้ใช้ ผู้พัฒนา และผู้ให้บริการ แต่ยังคงมีข้อจำกัดบางประการที่มีผลต่อการให้บริการบน สภาพแวดล้อมแบบกลุ่มเมฆ [3],[7], [9]

### 1. ความปลอดภัยและความเป็นส่วนตัว

เมื่อข้อมูลและแอปพลิเคชันถูกส่ง ไปยังระบบการประมวลผลแบบกลุ่มเมฆผ่าน เครือข่ายอินเทอร์เน็ต อีกทั้งสภาพแวดล้อมที่กระบวนการทำงานต้องอาศัยความสามารถของระบบ การประมวลผลแบบกลุ่มเมฆ เซิร์ฟเวอร์หลายกลุ่มบนเครือข่าย องค์กรธุรกิจและผู้ใช้ระดับบุคคล อาจไม่มั่นใจและมีความกังวลเกี่ยวกับมาตรฐานความปลอดภัยของข้อมูลบนเครือข่าย อินเทอร์เน็ต นอกจากนี้โดยสภาพแวดล้อมของกลุ่มเมฆที่ข้อมูลการสนทนา และประวัติการเข้าใช้ บริการเครือข่าย (Log) จะไม่ได้ถูกจัดเก็บบนระบบไอทีขององค์กร แต่กระจายไปบนเครือข่าย อินเทอร์เน็ต จึงมีความเสี่ยงหากข้อมูลการติดต่อสื่อสารระหว่างองค์กรธุรกิจ ซึ่งเป็นความลับทาง การค้าอาจถูกโจรกรรมจากเครือข่าย

### 2. ไม่มีมาตรฐานของแพลตฟอร์ม

ผู้ให้บริการกลุ่มเมฆมีมาตรฐานแพลตฟอร์มที่แตกต่างกัน โดย Amazon Web Service เป็นแบบซอฟต์แวร์ฟรีโปรแกรม (Linux, Apache, MySQL, Perl/PHP: LAMP) ขณะที่ Google App Engine เป็นแบบมาตรฐานเฉพาะผลิตภัณฑ์ (Proprietary Formats) และผู้ใช้ วินโดวส์มักจะใช้บริการจาก Go Grid ดังนั้นสำหรับผู้พัฒนาแอปพลิเคชัน หากต้องการให้ผลงาน ครอบคลุมตลาดผู้ใช้หลายๆ กลุ่ม ก็ต้องพัฒนาแอปพลิเคชันบนหลายแพลตฟอร์มซึ่งเป็นเรื่อง ยุ่งยาก

### 3. ความเชื่อถือได้ (Reliability)

โอกาสที่บริการระบบการประมวลผลแบบกลุ่มเมฆจะล้มหรือไม่สามารถให้บริการ ได้ในบางขณะจะส่งผลกระทบต่อผู้ใช้ในระบบ

### 4. คุณสมบัติด้านการเคลื่อนย้ายข้อมูล (Portability)

ตามหลักการทำงานแบบแบ่งปันประสิทธิภาพของระบบไอทีบนระบบการ ประมวลผลแบบกลุ่มเมฆหลายๆ กลุ่ม หมายถึงกระบวนการประมวลผลแต่ละชิ้นงานอาจเริ่มต้น และสิ้นสุดลงโดยผ่านการทำงานบนกลุ่มเมฆ (เครื่องแม่ข่าย) มากกว่า 1 กลุ่ม ในขั้นตอนการ เคลื่อนย้ายข้อมูลจากกลุ่มเมฆหนึ่งไปยังอีกกลุ่มเมฆหนึ่ง แม้จะเกิดความคุ้มค่าของการใช้ ประสิทธิภาพของคอมพิวเตอร์แต่อาจสิ้นเปลือง ทรัพยากรด้านการสื่อสาร (Bandwidth) บน เครือข่ายอินเทอร์เน็ตในมูลค่าที่สูงกว่า

## 5. เซิร์ฟเวอร์ทางกายภาพ

ระบบประมวลผลกลุ่มเมฆให้บริการผ่านเทคโนโลยีเสมือน อย่างไรก็ตามในความเป็นจริงอุปกรณ์ฮาร์ดแวร์ที่ทำหน้าที่เซิร์ฟเวอร์ทางกายภาพยังคงมีอยู่จริง ซึ่งมีโอกาสที่จะติดตั้งกระจายอยู่ในประเทศต่างๆ ทั่วทุกมุมโลก ประเด็นที่ยังคงเป็นกังวลคือข้อมูลทางธุรกิจและข้อมูลที่มีผลต่อความมั่นคง ของประเทศ อาจถูกจัดเก็บบนเซิร์ฟเวอร์ในประเทศอื่น และมีความเสี่ยงที่รัฐบาล หรือทางการ ตลอดจนภาคเอกชนของประเทศที่เป็นที่ตั้งของเซิร์ฟเวอร์จะสามารถเข้าถึงข้อมูลเหล่านั้นได้

### 2.1.7 ข้อดีและข้อเสียของการประมวลผลแบบกลุ่มเมฆ

**ข้อดี** คือ ลดความซับซ้อน ยุ่งยาก อีกทั้งยังช่วยประหยัดพลังงานและลดค่าใช้จ่ายเพราะไม่ต้องซื้อเครื่องแม่ข่ายเอง การทำฐานข้อมูลในบริษัทหรือในองค์กรเองไม่จำเป็นต้องซื้อซอฟต์แวร์เพื่อจัดการฐานข้อมูลหรือบรรจุข้อมูล นอกจากนั้นแล้วการประมวลผลแบบกลุ่มเมฆยังสามารถรองรับความต้องการที่หลากหลายกว่า ซึ่งแตกต่างกับเทคโนโลยีการประมวลผลแบบกริด (Grid Computing) ที่ค่อนข้างเน้นการทำงานเฉพาะด้าน เนื่องจากการประมวลผลแบบกลุ่มเมฆทำงานผ่านเทคโนโลยีเสมือน (Virtualization) ระบบจึงไม่ได้ถูกจำกัดในเรื่องของสมรรถนะและขีดความสามารถของคอมพิวเตอร์จากระบบต่างๆ ในระยะหลังนิยมใช้กันทั่วโลกและมีการพยายามประยุกต์ใช้ในงานต่างๆ มากมายทั่วไป โดยเฉพาะการทำธุรกิจทุกประเภทที่เห็นเด่น เช่น Google App ซึ่งมีฟังก์ชันใช้ได้มากมาย ตั้งแต่ การพยากรณ์การตลาด การจัดการการขาย การสร้างความสัมพันธ์กับลูกค้า หรือซีอาร์เอ็ม (Customer Relation Management: CRM) เชื่อมต่อการดำเนินการด้านธุรกิจอื่นๆ ในบริษัทสะดวกและประหยัดมาก จึงถือได้ว่าการประมวลผลแบบกลุ่มเมฆเป็นนวัตกรรมทางด้านคอมพิวเตอร์รูปแบบใหม่แห่งอนาคตนั่นเอง [3], [7]

**ข้อเสีย** คือเรื่องความปลอดภัยของข้อมูล ตัวอย่างเช่น ฐานข้อมูลอีเมลที่ผู้ใช้ติดต่อกับบุคคลต่างๆ บนอินเทอร์เน็ต หรือการติดต่อผ่านบริการเครือข่ายสังคม เช่น มายสเปซ เฟซบุ๊ก ข้อมูลของผู้ใช้ถูกบรรจุอยู่ในเครื่องผู้อื่น ซึ่งขณะที่ผู้ใช้ใช้งานข้อมูลเหล่านั้น ผู้ใช้ก็ไม่รู้เลยว่าใครมาใช้ฐานข้อมูลของผู้ใช้บ้าง มีใครช่วยปกป้องข้อมูลให้ผู้ใช้บ้าง หรือผู้ใช้เองก็ไม่รู้ว่าฐานข้อมูลเหล่านั้นอยู่ที่ใด ประกอบกับแฮกเกอร์มีความสามารถสูง ดังนั้นจึงไม่ใช่เรื่องยากที่แฮกเกอร์จะสามารถเจาะเอาข้อมูลของท่านไปได้ ทำให้ผู้ใช้ต้องมีความระมัดระวังด้วย โดยเฉพาะข้อมูลที่เป็นความลับหรือมีความสำคัญมากๆ เช่น ข้อมูลส่วนตัว ข้อมูลทางการเงิน เป็นต้น

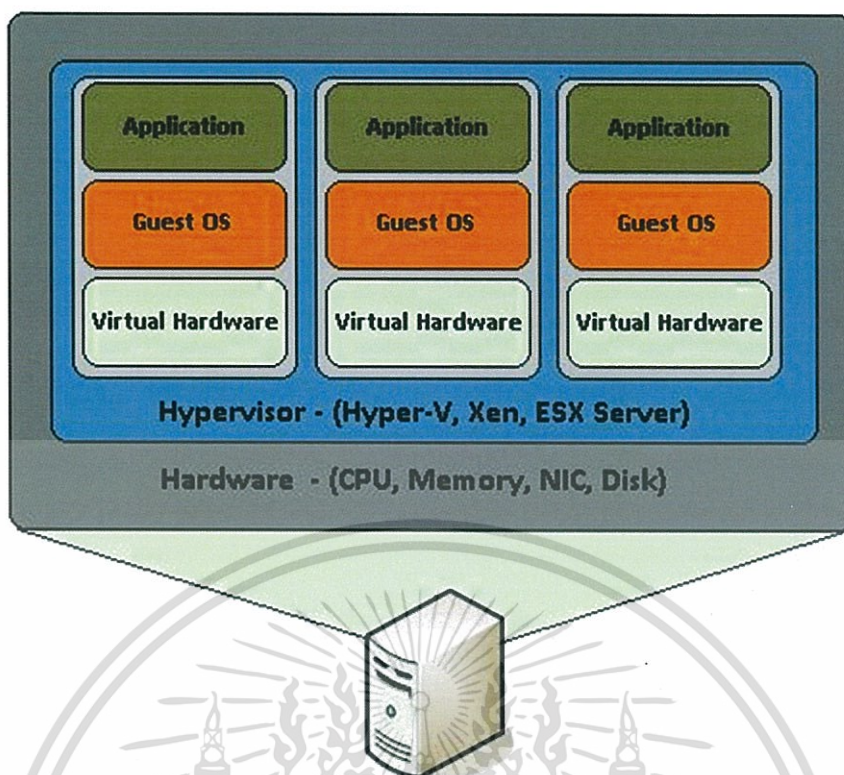
## 2.2 เทคโนโลยีการจำลองเครื่องคอมพิวเตอร์เสมือน (Virtualization Technology)

ปัจจุบันเทคโนโลยีทางด้านฮาร์ดแวร์นั้นก้าวหน้าไปมาก ในส่วนของประสิทธิภาพและความเร็วในการประมวลผล ทำให้เกิดแนวการสร้างระบบจำลองเสมือนจริงขึ้นที่เราเรียกว่าเทคโนโลยีการจำลองเครื่องคอมพิวเตอร์เสมือน [3] เทคโนโลยีที่กล่าวไปนี้ใช้สำหรับสร้างทรัพยากรเสมือนหรือทรัพยากรแบบนามธรรมของระบบคอมพิวเตอร์ เช่น สร้างหน่วยประมวลผลกลาง (Central Processing Unit: CPU) การ์ดแสดงผล (Video Card) การ์ดแลน (Local Area Network Card: LAN Card) เสมือน ซึ่งสิ่งที่ถูกสร้างขึ้นทั้งหมดนั้นเรียกว่า เครื่องคอมพิวเตอร์เสมือน (Virtual Machine: VM) [3], [10-11] โดยระบบเทคโนโลยีเสมือนจะควบคุมและจัดการทรัพยากรภายในระบบ โดยมีการแบ่งจัดสรรทรัพยากรให้กับแต่ละเครื่องคอมพิวเตอร์เสมือนอย่างเหมาะสมตามความต้องการ ดังนั้นแต่ละเครื่องคอมพิวเตอร์เสมือนจะมียุคประกอบเปรียบเสมือนเป็นเครื่องคอมพิวเตอร์ที่สมบูรณ์เครื่องหนึ่ง ทำให้สามารถรองรับระบบปฏิบัติการได้หลากหลายบนเครื่องแม่ข่าย (Server) หรือคอมพิวเตอร์เพียงเครื่องเดียว

### 2.2.1 ตัวอย่างของผลิตภัณฑ์ที่ใช้ในการจำลองเทคโนโลยีเครื่องคอมพิวเตอร์เสมือน

1. ผลิตภัณฑ์จากบริษัท VMware ได้แก่ ESX, ESXi Server
2. ผลิตภัณฑ์จากบริษัท Microsoft ได้แก่ Hyper-V, Virtual Server 2005
3. ผลิตภัณฑ์จากบริษัท Oracle ได้แก่ Oracle VM, VirtualBox
4. นอกเหนือจากนี้ก็จะมีผลิตภัณฑ์ของบริษัท Proxmox และ Eucalyptus ซึ่งเป็น Open Source

ซอฟต์แวร์เหล่านี้จะถูกเรียกว่า ไฮเปอร์ไวเซอร์ (Hypervisor) ซึ่งหมายถึงซอฟต์แวร์ที่ติดตั้งเพื่อใช้ในการจัดสรรทรัพยากรหรือเพื่อการสร้างเครื่องคอมพิวเตอร์เสมือน สถาปัตยกรรมของเทคโนโลยีการจำลองเครื่องคอมพิวเตอร์เสมือนด้วยซอฟต์แวร์แสดงดังรูปที่ 2.7



รูปที่ 2.7 สถาปัตยกรรมของเทคโนโลยีการจำลองเครื่องคอมพิวเตอร์เสมือนด้วยซอฟต์แวร์ [3]

### 2.2.2 ประโยชน์ที่ได้รับจากการจำลองเทคโนโลยีเครื่องคอมพิวเตอร์เสมือน

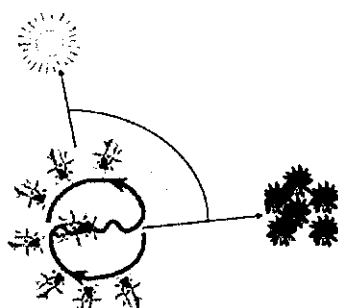
1. สัมเลิกระบบเก่าที่กำหนดให้ 1 แอปพลิเคชันต้องทำงานอยู่บน 1 เครื่องแม่ข่าย โดยการนำเอางานของหลายเครื่องแม่ข่ายมาทำงานร่วมกันทำให้สามารถใช้งานเครื่องแม่ข่ายได้เต็มประสิทธิภาพมากขึ้น เช่น การรวมงานของเครื่องแม่ข่าย 2 เครื่องที่ใช้งาน CPU เพียงแค่ 40% ให้มาทำงานอยู่บนเครื่องแม่ข่ายเครื่องเดียว ซึ่งจะทำให้เครื่องแม่ข่ายอีกเครื่องนั้นว่าง และสามารถนำไปใช้กับงานอื่นได้มากขึ้น
2. ลดค่าใช้จ่ายของศูนย์รวมข้อมูล (Data Center) โดยลดจำนวนของโครงสร้างพื้นฐาน นั่นคือลดจำนวนของเครื่องแม่ข่ายลง ทำให้ต้องการพื้นที่น้อยลง ลดความจำเป็นในการใช้ไฟฟ้า และการทำความเย็น บริหารจัดการได้ง่ายขึ้นเนื่องจากมีจำนวนน้อย
3. ทำให้ฮาร์ดแวร์และซอฟต์แวร์นั้นว่างและอยู่ในสถานะพร้อมใช้งานมากขึ้น ส่งผลให้เกิดความต่อเนื่องในการดำเนินงาน
4. มีความยืดหยุ่นในการปฏิบัติงาน สามารถตอบสนองความต้องการทางธุรกิจได้แบบไดนามิก เพราะสามารถเพิ่มหรือลดทรัพยากรให้กับระบบงานที่ต้องการเฉพาะช่วงเวลาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 ขั้นตอนวิธีอาณานิคมผึ้งเทียม (Artificial Bee Colony Algorithm: ABC Algorithm)

ขั้นตอนอาณานิคมผึ้ง (Bee Colony Algorithm) [12 - 13] ได้ถูกนำเสนอโดย Pham และคณะ โดยวิธีบีเอส เป็นขบวนการในการหาค่าที่เหมาะสมที่สุด ที่เลียนแบบมาจากพฤติกรรมในการหาอาหารของฝูงผึ้ง และเป็นกระบวนการหาค่าตอบชนิดหนึ่งที่อยู่ในกลุ่มที่มีแนวคิดมาจากการเลียนแบบพฤติกรรมของฝูงแมลง (Swarm Based Optimisation Algorithm) โดยลักษณะเด่นของกระบวนการหาค่าตอบในกลุ่มนี้ จะใช้หลักการในการสุ่มค่าตัวแปรด้วยจำนวนประชากรจำนวนหนึ่งเข้าไป แล้วนำผลลัพธ์มาทำการคำนวณ การคัดเลือกและการปรับค่า แล้วทำการคำนวณรอบซ้ำจนกระทั่งได้คำตอบที่เหมาะสมในที่สุด ลักษณะของฝูงผึ้งในธรรมชาติ ผึ้งจะบินออกไปหาอาหารในระยะทางไกลมากกว่า 14 กิโลเมตรในหลายๆ ทิศทางพร้อมๆ กัน โดยมีเป้าหมายในการสำรวจหาแหล่งอาหารเพื่อให้ได้อาหารจำนวนมาก ผึ้งผึ้งจะถูกส่งให้ไปสำรวจหาแหล่งอาหารที่มีอาหารอุดมสมบูรณ์ ตามทฤษฎีแหล่งที่มีดอกไม้ซึ่งให้ปริมาณน้ำหวานจากเกสรสูงและมีคุณภาพดี ก็จะมีผึ้งจำนวนมากกว่าที่บริเวณอื่น เช่นเดียวกันกับบริเวณที่มีแหล่งอาหารน้อยกว่าและคุณภาพต่ำกว่า ก็จะมีผึ้งจำนวนที่น้อยกว่าในบริเวณนั้น

ขบวนการในการหาอาหารของผึ้งเริ่มต้นจาก การส่งผึ้งสังเกตการณ์ (Scout Bees) ไปสำรวจหาแหล่งดอกไม้ ผึ้งสังเกตการณ์จะบินสุ่มไปยังที่หนึ่งและสุ่มย้ายไปยังที่อื่นๆ ในช่วงฤดูเก็บเกี่ยวอาหาร ผึ้งผึ้งจะสุ่มสำรวจหาแหล่งอาหารเมื่อผึ้งสังเกตการณ์บินกลับมาที่รัง ผึ้งตัวที่พบแหล่งอาหาร ที่มีคุณภาพสูงกว่ามาตรฐานเช่น มีส่วนประกอบของน้ำตาลสูงจะนำน้ำหวานมาเก็บ แล้วไปยังลานเต้น (Dance Floor) เพื่อทำการเต้นที่เรียกว่า “เต้นแกว่ง” (Waggle Dance) การเต้นนั้นมีความสำคัญสำหรับการสื่อสารของผึ้งผึ้งเป็นอย่างมาก เนื่องจากผึ้งจะใช้ช่วงระหว่างการเต้นสื่อสารข้อมูลสามชุดที่เกี่ยวกับแหล่งของดอกไม้ นั่นคือ ทิศทางของดอกไม้ ระยะทางที่ห่างจากรัง และระดับคุณภาพของน้ำหวาน ซึ่งข้อมูลเหล่านี้จะทำให้ผึ้งตัวอื่นบินออกไปหาแหล่งดอกไม้ได้อย่างแม่นยำโดยไม่ต้องมีแผนที่ ผึ้งผึ้งจะนำข้อมูลที่ได้จากการสำรวจภายนอกแต่ละชุดมาสื่อสารระหว่างกันในช่วงที่ทำการเต้นแกว่ง และทำการเปรียบเทียบทั้งในด้านของคุณภาพของอาหาร และพลังงานที่จะต้องใช้ในการออกไปหาอาหาร หลังจากการเต้นแกว่งในลานเต้น ผึ้งสังเกตการณ์ก็จะกลับไปยังแหล่งดอกไม้ร่วมกับผึ้งตัวอื่นๆ ซึ่งอยู่ในรัง ผึ้งจำนวนมากกว่าจะถูกส่งไปยังบริเวณที่เชื่อว่าจะมีอาหารที่มีคุณภาพสูงกว่า และเช่นเดียวกัน ผึ้งจำนวนน้อยกว่าก็จะถูกส่งไปยังพื้นที่ที่เชื่อว่าจะมีอาหารคุณภาพต่ำกว่าซึ่งจากวิธีการนี้จะทำให้ผึ้งผึ้งหาอาหารได้อย่างรวดเร็วและมีประสิทธิภาพ ซึ่งการเต้นแกว่งเพื่อสื่อสารข้อมูลของแหล่งอาหารของผึ้งแสดงได้ดังรูปที่ 2.8



รูปที่ 2.8 การเดินแกว่งของผึ้งเพื่อสื่อสารข้อมูลของแหล่งอาหาร [14]

ขั้นตอนวิธีอาณานิคมผึ้งเทียม (Artificial Bee Colony Algorithm: ABC Algorithm) [15] เป็นขั้นตอนวิธีที่ใช้ในการแก้ปัญหาการหาค่าหรือวิธีการที่เหมาะสมที่สุด จากค่าหรือวิธีการที่เป็นไปได้ทั้งหมดเพื่อให้ได้คำตอบที่ต้องการ เช่น ค่าน้อยที่สุด หรือมากที่สุดของปัญหาที่ต้องการแก้ปัญหา ขั้นตอนวิธีอาณานิคมผึ้งเทียมได้เลียนแบบพฤติกรรมการออกหาอาหารของอาณานิคมผึ้ง ประชากรของอาณานิคมผึ้งเทียมแบ่งออกเป็น 2 ส่วน จำนวนเท่ากัน คือ ผึ้งงาน (Employed bees) และผึ้งเฝ้าดู (Onlooker bees) โดยผึ้งงานจะออกหาอาหารแล้วกลับมาให้ข้อมูลเกี่ยวกับแหล่งอาหารที่พบให้กับผึ้งเฝ้าดูที่รออยู่ในรัง ผึ้งเฝ้าดูจะเลือกผึ้งงานเพื่อตามออกไปเก็บอาหารขึ้นอยู่กับข้อมูลที่ได้รับจากผึ้งงาน ถ้าอาหารแหล่งใดถูกเก็บจนหมด ผึ้งงานที่เป็นเจ้าของแหล่งอาหารนั้นจะกลายเป็นผึ้งสำรวจ (Scout bees) เพื่อสำรวจหาแหล่งอาหารใหม่ ขั้นตอนวิธีอาณานิคมผึ้งเทียมสำหรับปัญหาการหาค่าพารามิเตอร์ที่เหมาะสมที่สุดมีขั้นตอนหลักดังนี้

- 1) เริ่มต้นสร้างเวกเตอร์พารามิเตอร์ให้กับผึ้งงาน
- 2) ผึ้งงานสร้างเวกเตอร์พารามิเตอร์ใหม่ จากเวกเตอร์พารามิเตอร์เดิมแล้วเลือกเวกเตอร์พารามิเตอร์ที่ให้ค่าความเหมาะสมมากกว่า
- 3) ผึ้งเฝ้าดูแต่ละตัวเลือกผึ้งงาน โดยขึ้นอยู่กับค่าความเหมาะสมของผึ้งงาน แล้วสร้างเวกเตอร์พารามิเตอร์ใหม่จากเวกเตอร์พารามิเตอร์ของผึ้งงานที่เลือก ซึ่งผึ้งงานจะเลือกเวกเตอร์พารามิเตอร์ที่ให้ค่าความเหมาะสมมากกว่า
- 4) จดจำเวกเตอร์พารามิเตอร์ที่ดีที่สุดตั้งแต่เริ่มต้นจนถึงขั้นตอนนี้
- 5) หาเวกเตอร์พารามิเตอร์ของผึ้งงานที่ต้องละทิ้งแล้วกำหนดให้เป็นผึ้งสำรวจที่สร้างเวกเตอร์พารามิเตอร์ขึ้นมาใหม่
- 6) ทำซ้ำขั้นตอนที่ 2-5 จนกระทั่งครบตามจำนวนรอบที่กำหนดไว้

**ขั้นตอนที่ 1** เวกเตอร์พารามิเตอร์หรือตำแหน่งแหล่งอาหารของฝูงงาน  $\vec{x}_i$  ประกอบด้วยสมาชิก  $x_{ij}$  เมื่อ  $i = 1, 2, \dots, SN$  ;  $SN$  คือ จำนวนของฝูงงานทั้งหมดภายในอาณานิคมผึ้งเทียม และ  $j = 1, 2, \dots, D$  ;  $D$  จำนวนพารามิเตอร์ทั้งหมดที่ต้องการหาค่าที่เหมาะสมโดย  $x_{ij}$  สร้างโดยสมการที่ (2.1) แล้วคำนวณค่าฟังก์ชันวัตถุประสงค์และค่าความเหมาะสมโดยสมการที่ (2.2)

$$x_{ij} = l_j + rand(0,1) * (u_j - l_j) \quad (2.1)$$

เมื่อ  $l_j$  และ  $u_j$  คือ ค่าต่ำสุดและสูงสุดของพารามิเตอร์  $x_{ij}$  ตามลำดับ

$rand$  คือ ค่าสุ่มที่มีค่าระหว่าง 0 ถึง 1

$$fit_i(\vec{x}_i) = \begin{cases} \frac{1}{1 + f_i(\vec{x}_i)} & , f_i(\vec{x}_i) \geq 0 \\ 1 + |f_i(\vec{x}_i)| & , f_i(\vec{x}_i) < 0 \end{cases} \quad (2.2)$$

เมื่อ  $f_i(\vec{x}_i)$  คือ ค่าของฟังก์ชันวัตถุประสงค์สำหรับเวกเตอร์พารามิเตอร์  $\vec{x}_i$  ซึ่งเป็นฟังก์ชันที่ต้องการหาค่าพารามิเตอร์ที่เหมาะสม

**ขั้นตอนที่ 2** ฝูงงานสร้างเวกเตอร์พารามิเตอร์ใหม่  $\vec{v}_i$  จาก  $\vec{x}_i$  โดยหาได้จากสมการที่ (2.3) แล้วคำนวณค่าฟังก์ชันวัตถุประสงค์ และค่าความเหมาะสมเช่นเดียวกันกับขั้นตอนที่ 1

$$v_{ij} = x_{ij} + \varphi \times (x_{ij} - x_{rj}) \quad (2.3)$$

เมื่อ  $\vec{x}_r$  คือ เวกเตอร์พารามิเตอร์ที่สุ่มจาก  $\vec{x}$  ที่ไม่ใช่เวกเตอร์พารามิเตอร์เดียวกันกับ  $\vec{x}_i$

$j$  คือ จำนวนเต็มสุ่มที่มีค่าระหว่าง 1 ถึง  $D$  และ  $\varphi$  เป็นตัวเลขสุ่มที่มีค่าระหว่าง -1 ถึง 1 ฝูงงานจะเลือกเวกเตอร์พารามิเตอร์ระหว่าง  $\vec{x}_i$  และ  $\vec{v}_i$  ที่ให้ค่าความเหมาะสมที่มากกว่า

**ขั้นตอนที่ 3** ผึ้งเฝ้าดูเลือกฝูงงานโดยพิจารณาค่าความน่าจะเป็น  $p_i$  ที่คำนวณจากค่า  $fit_i(\vec{x}_i)$  โดยคำนวณจากสมการที่ (2.4)

$$p_i = \frac{fit_i(\vec{x}_i)}{\sum_{i=1}^{SN} fit_i(\vec{x}_i)} \quad (2.4)$$

กระบวนการเลือกฟังก์ชันของฟังก์ชันค่า จะเริ่มจากฟังก์ชันค่าแต่ละตัวคู่ค่าระหว่าง 0 ถึง 1 มาหนึ่งค่าแล้วเทียบกับ ค่า  $p$  ถ้าค่าคู่นั้นมีค่าน้อยกว่า ฟังก์ชันค่าก็จะเลือกฟังก์ชันตัวนั้น ฟังก์ชันค่าจะสร้างเวกเตอร์พารามิเตอร์ใหม่จากเวกเตอร์พารามิเตอร์ของฟังก์ชันที่เลือก พร้อมทั้งคำนวณค่าฟังก์ชันวัตถุประสงค์และค่าความเหมาะสม จากนั้นฟังก์ชันค่าจะเลือกเวกเตอร์พารามิเตอร์ที่ให้ค่าความเหมาะสมที่มากกว่าเช่นเดียวกันกับขั้นตอนที่ 2

**ขั้นตอนที่ 4** ฟังก์ชันค่าใดที่ไม่ถูกปรับค่าเวกเตอร์พารามิเตอร์ติดต่อกันจนถึงจำนวนที่กำหนดไว้ (Limit or Abandonment Criterion) ก็จะถูกกำหนดให้เป็นฟังก์ชันค่าแล้วกำหนดเวกเตอร์พารามิเตอร์ใหม่ด้วยสมการที่ (2.1)

## 2.4 วิธีการจัดลำดับงานแบบฮิวริสติก (Heuristics)

วิธีการจัดตารางการผลิตแบบฮิวริสติก (Heuristic) [16-17] เป็นการนำกฎต่างๆ มาใช้ในการหาผลลัพธ์ที่น่าพอใจของปัญหา เช่น กฎการจัดลำดับความสำคัญ วิธีการสุ่ม เป็นต้น วิธีนี้อาจได้ผลลัพธ์ที่น่าพอใจ แต่ไม่สามารถรับรองได้ว่าเป็นผลลัพธ์ที่ดีที่สุด ในงานวิจัยฉบับนี้ได้นำกฎเกณฑ์พื้นฐานมาใช้เป็นกฎในการจัดลำดับงาน โดยใช้กฎลำดับความสำคัญเป็นกฎในการเลือกขั้นตอนการทำงาน ดังนี้

1. การจัดงานมาก่อนทำก่อน (First Come First Serve: FCFS) เป็นการจัดลำดับงานโดยให้ทำงานที่เข้ามาก่อนเป็นอันดับแรก และทำงานที่เข้ามาหลังเป็นอันดับต่อไป ซึ่งหากพิจารณาจากเกณฑ์การวัดประสิทธิภาพการผลิตด้วยวิธี FCFS เป็นวิธีที่ไม่ค่อยดีมากนัก เนื่องจากงานที่เข้าไปจะครอบครองซีพียูจนกระทั่งทำงานไปจนเสร็จ ซึ่งซีพียูไม่มีเวลาจำกัดในการเข้าทำงาน ซึ่งจะส่งผลให้งานอื่นๆ ต้องใช้เวลาในการรอเข้าทำงาน

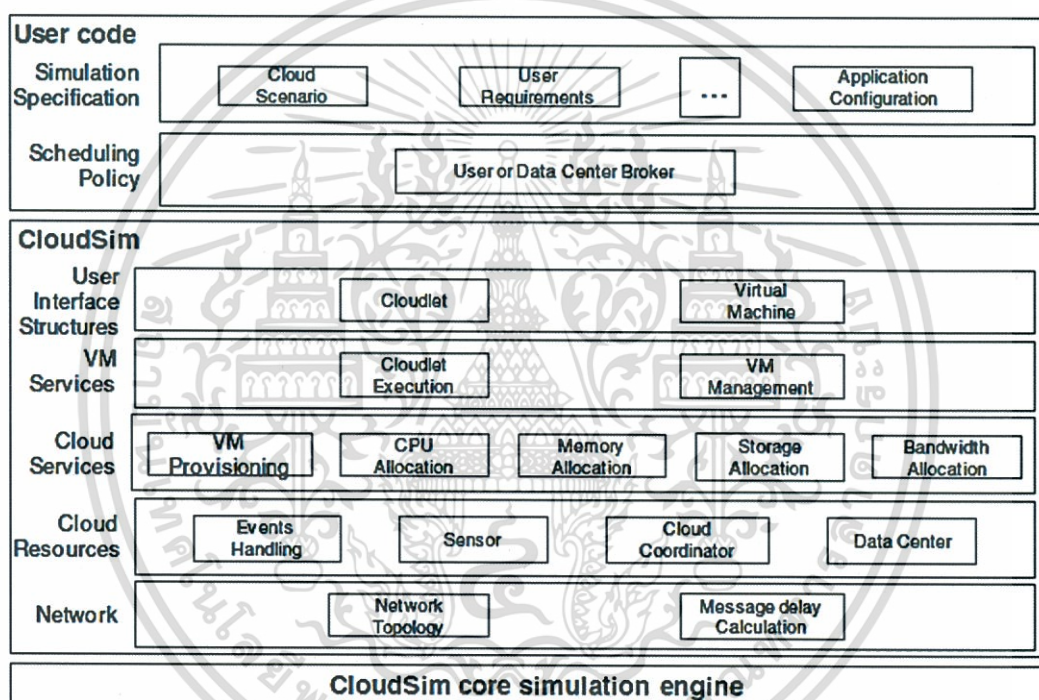
2. การจัดงานที่มีขนาดเล็กที่สุดเข้าทำงานก่อน (Shortest Job First: SJF) เป็นการจัดลำดับความสำคัญของการทำงานโดยให้ทำงานที่มีขนาดเล็กที่สุดเข้าทำงานก่อน แล้วจึงทำงานที่มีขนาดใหญ่เป็นลำดับถัดไป จะเห็นได้ว่า SJF เป็นวิธีที่มุ่งในการลดเวลาในการทำงานของแต่ละงานและพยายามทำให้งานต่างๆ ออกจากระบบประมวลผลให้เร็วที่สุด ข้อดีของการจัดงานแบบ SJF คือเวลาโดยเฉลี่ยของงานในระบบจะต่ำที่สุด แต่ข้อเสียของ SJF คือ งานที่มีขนาดใหญ่จะถูกทำงานอยู่ในอันดับท้ายๆ ทำให้เกิดการรอคอย

3. การจัดงานที่มีขนาดใหญ่ที่สุดเข้าทำก่อน (Longest Job First: LJF) เป็นการจัดลำดับความสำคัญในการทำงาน โดยให้ทำงานที่มีขนาดใหญ่ที่สุดเป็นอันดับแรก แล้วจึงทำงานที่มีขนาดเล็กกว่าเป็นอันดับต่อไป โดยทั่วไปวิธีนี้มักจะเป็นวิธีที่ส่งผลเสียต่อประสิทธิภาพโดยรวมของระบบมากกว่าวิธีการอื่นๆ เพราะการจัดงานแบบ LJF มักทำให้เวลาที่ใช้ในการทำกระบวนการทั้งหมด และยังทำให้ประสิทธิภาพการใช้งานทรัพยากรต่ำอีกด้วย

## 2.5 เครื่องมือที่ใช้ในการสร้างแบบจำลอง CloudSim-3.0.1 (CloudSim Tools)

### 2.5.1 ลักษณะทั่วไปของ CloudSim

CloudSim [18-19] เป็นเครื่องมือที่ใช้ในการสร้างแบบจำลองในระบบการประมวลผลแบบกลุ่มเมฆที่มีสภาพแวดล้อมเหมือนจริง อีกทั้งในการสร้างแบบจำลองนั้นยังสามารถทำการปรับเปลี่ยนค่าพารามิเตอร์ตามความต้องการของผู้ใช้ได้ เช่น สามารถจำลองสภาพแวดล้อมของศูนย์ข้อมูล (Data Center) ซึ่งประกอบด้วย เครื่องคอมพิวเตอร์เสมือน (VMs) หน่วยความจำ (Memory) หน่วยเก็บข้อมูล (Storages) และ แบนด์วิธ (Bandwidth) เป็นต้น จากภาพที่ 2.9 แสดงให้เห็นถึงรูปแบบลำดับชั้นของ CloudSim และสถาปัตยกรรมของ CloudSim



รูปที่ 2.9 สถาปัตยกรรมลำดับชั้นของ CloudSim (Layered CloudSim Architecture)

คุณสมบัติทั่วไปของ CloudSim มีดังนี้

1) สนับสนุนการสร้างแบบจำลองที่อยู่บนพื้นฐานของโครงสร้างของระบบการประมวลผลแบบกลุ่มเมฆ เช่นสามารถสร้างศูนย์ข้อมูลบนเครื่องหลัก 1 เครื่องได้ และสามารถจำลองการสร้างเครื่องคอมพิวเตอร์เสมือนได้

2) การทำงานใน CloudSim จะมีแพลตฟอร์มที่เป็นของตนเอง เช่น การจำลองศูนย์ข้อมูล การให้บริการของโบรกเกอร์ (Service Brokers) การจัดตารางการทำงาน (Scheduling) และ เงื่อนไขในการจัดงานและทรัพยากร (Allocations Policies)

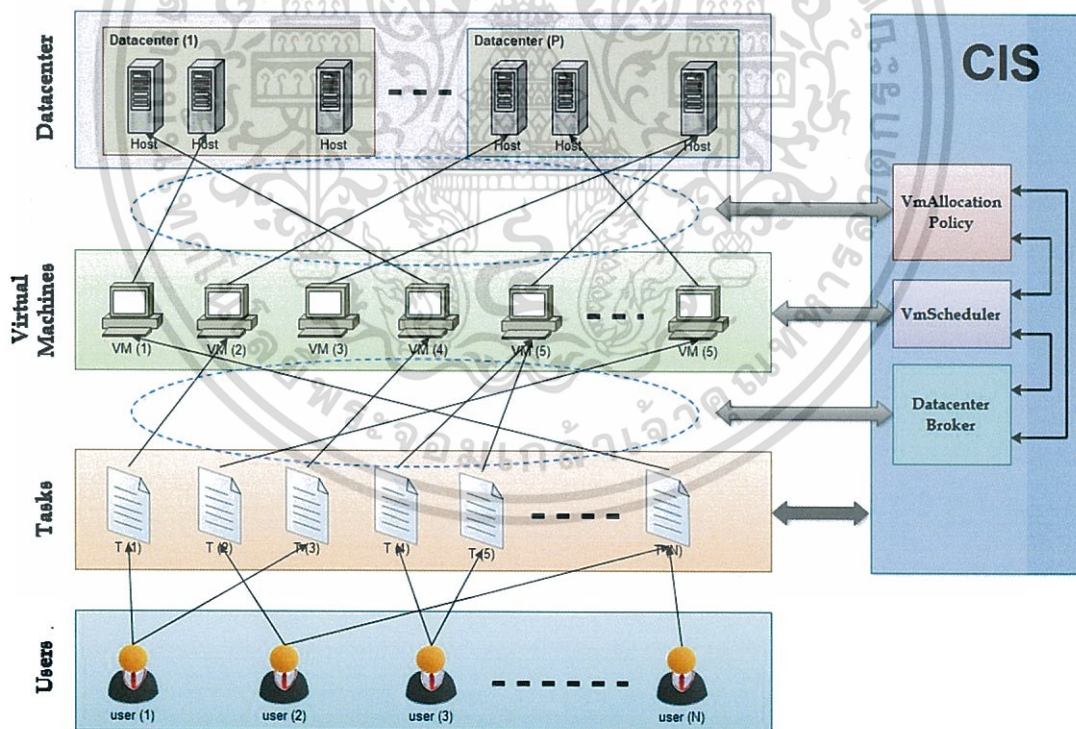
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) CloudSim เป็นเครื่องมือที่ถือได้ว่าเป็นความพร้อมในการจัดการกับเครื่องคอมพิวเตอร์เสมือน เช่น สามารถสร้างและจัดการกับบริการต่างๆ ได้เสมือนเครื่องคอมพิวเตอร์จริง และแต่ละเครื่องคอมพิวเตอร์เสมือนจะทำงานแบบที่เป็นอิสระต่อกัน

4) CloudSim จะมีความยืดหยุ่นในส่วนของ การจัดสรรการใช้พื้นที่ร่วมกัน (Space Share) และการใช้เวลาร่วมกันในการประมวลผล (Time Share) เสมือนการใช้บริการเครื่องคอมพิวเตอร์เสมือนจริง

## 2.5.2 รูปแบบการทำงานของ CloudSim

โดยทั่วไปรูปแบบการทำงานในระบบการประมวลผลแบบกลุ่มเมฆจะประกอบด้วยงานจากผู้ใช้จำนวนมากและมีรูปแบบงานที่แตกต่างกัน กำหนดให้ ผู้ใช้งานทั้งหมดมี  $m$  คน ประกอบด้วย  $\{User_1, User_2, User_3, \dots, User_m\}$  ; จำนวนงานทั้งหมดมี  $n$  งาน ประกอบด้วย  $\{T_1, T_2, T_3, \dots, T_n\}$  ; จำนวน VM มี  $k$  เครื่อง ประกอบด้วย  $\{VM_1, VM_2, VM_3, \dots, VM_k\}$  และประกอบด้วย  $p$  DataCenter คือ  $\{Datacenter_1, Datacenter_2, Datacenter_3, \dots, Datacenter_p\}$  รูปแบบการทำงานของ CloudSim สามารถแสดงได้ดังรูปที่ 2.10



รูปที่ 2.10 การทำงานของ CloudSim

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CIS (Cloud Information Service): เป็นตัวกลางในการจัดสรรงานระหว่างผู้ให้บริการและผู้ให้บริการ โดยในการจัดสรรงานนั้นจะต้องคำนึงถึงความต้องการของผู้ให้บริการ ซึ่งประกอบด้วย Datacenter Broker, VmScheduler และ VmAllocation Policy

Datacenter Broker: ดูแลในส่วนของการติดต่อระหว่าง ผู้ใช้งาน (User) และผู้ให้บริการ (Service Providers) โดยในการจัดสรรงานให้กับ VM นั้นจะต้องคำนึงถึงความต้องการของผู้ใช้งาน

VmScheduler: ดูแลในส่วนของการจัดสรรทรัพยากรให้กับแต่ละ VM ทั้งนี้ขึ้นอยู่กับข้อกำหนดการจัดสรรการใช้พื้นที่ร่วมกันและการใช้เวลารวมกันในการประมวลผล

VmAllocation Policy: ทำหน้าที่ในการเลือกศูนย์ข้อมูลให้เหมาะสมกับโฮสต์ (Host) โดยดูจาก หน่วยความจำ หน่วยเก็บข้อมูลและความต้องการใช้งานของเครื่องคอมพิวเตอร์เสมือน

## 2.6 การวัดการกระจายของข้อมูล (Measures of Dispersion)

การวัดการกระจาย (Measures of Dispersion)[20] เป็นสถิติประเภทหนึ่งที่คำนวณออกมาเป็นตัวเลข เพื่อใช้อธิบายลักษณะการกระจายข้อมูล การที่ข้อมูลชุดหนึ่งๆ ประกอบด้วยคะแนนที่มีค่าต่างกัน จะเรียกว่า เป็นข้อมูลที่มีการกระจาย ถ้าข้อมูลชุดนั้นประกอบด้วยคะแนนที่มีค่าต่างกันมาก เรียกว่า ข้อมูลที่มีการกระจายมาก ถ้าข้อมูลชุดนั้นประกอบด้วยคะแนนที่มีค่าต่างกันน้อย เรียกว่า ข้อมูลที่มีการกระจายน้อย และถ้าข้อมูลชุดนั้นประกอบด้วยคะแนนที่มีค่าเท่ากันทั้งหมด เรียกว่า ข้อมูลที่ไม่มีการกระจาย ตัวอย่างเช่น

ตารางที่ 2.1 ตัวอย่างข้อมูลที่มีการกระจายแตกต่างกัน

ข้อมูลชุดที่	คะแนนในชุดข้อมูล	ลักษณะการกระจาย
1	7 10 35 70 100	การกระจายมาก
2	50 58 60 61 67	การกระจายน้อย
3	30 30 30 30 30	ไม่มีการกระจาย

### 2.6.1 การวัดแนวโน้มเข้าสู่ส่วนกลาง

การวัดแนวโน้มเข้าสู่ส่วนกลางเป็นการหาค่ากลางของคะแนนชุดนั้น หรือหาค่าที่เป็นตัวแทนของคะแนนชุดนั้น การวัดแนวโน้มเข้าสู่ส่วนกลางที่นิยมใช้ ได้แก่ ค่าเฉลี่ยเลขคณิต (Arithmetic Mean,  $\bar{X}$ ) ค่ามัธยฐาน (Median) และค่าฐานนิยม (Mode)

### 2.6.1.1 ค่าเฉลี่ยเลขคณิต (Arithmetic mean)

ค่าเฉลี่ยเลขคณิตเป็นค่าเฉลี่ยที่นิยมใช้กันมากที่สุด ใช้สัญลักษณ์แทนด้วย  $\bar{x}$  เป็นค่ากลางที่คำนวณได้โดยการหาผลรวมของข้อมูลทุกค่า แล้วหารด้วยจำนวนข้อมูลทั้งหมด มีสูตรในการคำนวณดังสมการที่ (2.1) ใช้ในกรณีที่ข้อมูลไม่มีการแจกแจงความถี่

ถ้ามีข้อมูลเป็น  $x_1, x_2, x_3, \dots, x_N$  รวมทั้งหมด  $N$  ตัว

$$\bar{x} = \frac{\sum_{i=1}^N x_i}{N} \quad (2.1)$$

$$\sum_{i=1}^N x_i = x_1 + x_2 + x_3 + \dots + x_N \quad (2.2)$$

เมื่อ  $\bar{x}$  = ค่าเฉลี่ยเลขคณิต  
 $\frac{\sum_{i=1}^N x_i}{N}$  = ผลรวมของข้อมูลทั้งหมด คำนวณได้ดังสมการที่ (2.2)  
 $N$  = จำนวนข้อมูลทั้งหมด

ตัวอย่าง ข้อมูล 14, 16, 14, 17, 16, 14, 18, 17

$$\bar{x} = \frac{\sum_{i=1}^N x_i}{N}$$

$$\bar{x} = \frac{14+16+14+17+16+14+18+17}{8} = 15.75$$

∴ ค่าเฉลี่ยของข้อมูลชุดนี้มีค่าเท่ากับ 15.75

### 2.6.1.2 มัชยฐาน (Median)

มัชยฐาน หมายถึง ค่ากึ่งกลางของข้อมูลชุดนั้น หรือค่าที่อยู่ในตำแหน่งกึ่งกลางของข้อมูลชุดนั้น เมื่อได้จัดเรียงค่าของข้อมูลจากน้อยที่สุดไปหามากที่สุดหรือ จากค่ามากที่สุดไปหาค่าน้อยที่สุด ค่ากึ่งกลางจะเป็นตัวแทนที่แสดงว่ามีข้อมูลที่มากกว่าและน้อยกว่านี้อยู่ 50%

การหาค่ามัชยฐานของข้อมูลที่ไม่แจกแจงความถี่ สามารถหาได้ดังนี้

1. เรียงข้อมูลจากน้อยไปหามาก หรือจากมากไปหาน้อย
2. หาดำแหน่งของมัชยฐานจาก  $\frac{N+1}{2}$  เมื่อ  $N$  แทนจำนวนข้อมูลทั้งหมด

ตัวอย่าง ข้อมูล 9, 10, 5, 11, 14, 6, 16, 17, 13

เรียงข้อมูลที่มีค่าน้อยที่สุดไปหาข้อมูลที่มีค่ามากที่สุดคือ 5, 6, 9, 10, 11, 13, 14, 16, 17

$$\text{หาดำแหน่งมัชยฐาน } \frac{N+1}{2} = \frac{9+1}{2} = 5$$

∴ มัชยฐานของข้อมูลเท่ากับ 11

**ตัวอย่าง** ข้อมูล 40, 35, 24, 28, 26, 29, 36, 31, 42, 20, 23, 32

เรียงข้อมูลจากข้อมูลที่มีค่าน้อยที่สุด ไปหาข้อมูลที่มีค่ามากที่สุดคือ 20, 23, 24, 26, 28, 29, 31, 32, 35, 36, 40, 42

$$\text{หาตำแหน่งมัธยฐาน } \frac{N+1}{2} = \frac{12+1}{2} = 6.5$$

มัธยฐานอยู่ในตำแหน่ง ที่ 6.5 อยู่ระหว่าง 29 กับ 31

$$\text{มัธยฐานเท่ากับ } \frac{29+31}{2} = 30$$

∴ มัธยฐานเท่ากับ 30

### 2.6.1.3 ฐานนิยม (Mode)

ฐานนิยม หมายถึง ค่าของคะแนนที่ซ้ำกันมากที่สุดหรือ ค่าคะแนนที่มีความถี่สูงที่สุดในข้อมูลชุดนั้น

**ตัวอย่าง** ข้อมูล 3, 2, 4, 5, 6, 4, 8, 4, 7, 10

ข้อมูลที่ซ้ำกันมากที่สุดคือ 4 ∴ ฐานนิยมคือ 4

**ตัวอย่าง** ข้อมูลบางชุดอาจมีฐานนิยม 2 ค่า เช่น 10, 14, 12, 10, 11, 13, 12, 14, 12, 10

ข้อมูลที่ซ้ำกันมากที่สุดคือ 10 กับ 12 ∴ ฐานนิยม คือ 10 กับ 12

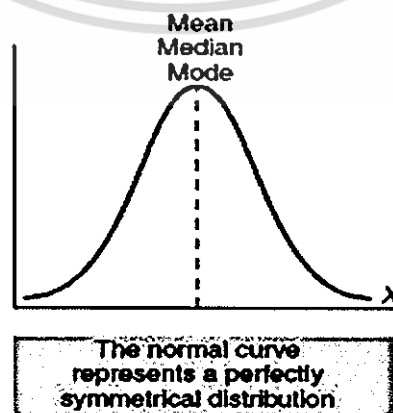
**ตัวอย่าง** ข้อมูลบางชุดอาจจะไม่มีฐานนิยมซึ่ง ได้แก่

ข้อมูลที่ไม่มีรายการซ้ำกันเลย เช่น 8, 9, 10, 11, 13, 15

### 2.6.2 ลักษณะของการกระจายของข้อมูล

ลักษณะของการกระจายของข้อมูลอาจแบ่งได้เป็น 3 แบบ ดังนี้

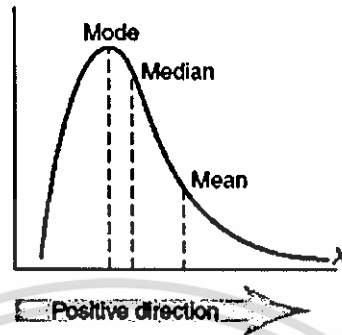
1. ลักษณะการกระจายของข้อมูลแบบการกระจายแบบสมมาตร (Symmetric Distribution) ค่าเฉลี่ยเลขคณิต มัธยฐาน และฐานนิยมมีค่าเท่ากัน หรืออยู่ที่จุดเดียวกันคือจุดที่มีความถี่สูงสุด แสดงดังรูปที่ 2.11



**รูปที่ 2.11** การกระจายของข้อมูลแบบการกระจายแบบสมมาตร [20]

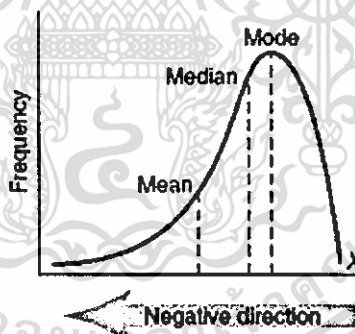
เอกสารนี้เป็นเอกสารที่โรงเรียนเจ้าพระยาฯ อนุญาตให้ใช้ฟรีโดยไม่คิดค่าใช้จ่าย ขอสงวนสิทธิ์ในเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ลักษณะการกระจายของข้อมูลแบบการกระจายที่เบ้ทางขวา (Right-Skewed Distribution หรือ Positive Curve) ค่าเฉลี่ยเลขคณิตจะมีค่ามากที่สุด รองลงมาเป็นมัธยฐาน และฐานนิยมตามลำดับ แสดงดังรูปที่ 2.12



รูปที่ 2.12 การกระจายของข้อมูลแบบการกระจายที่เบ้ทางขวา [20]

3. ลักษณะการกระจายของข้อมูลแบบการกระจายที่เบ้ทางซ้าย (Left-Skewed Distribution หรือ Negative Curve) ฐานนิยมจะมีค่ามากที่สุด รองลงมาเป็นมัธยฐาน และค่าเฉลี่ยเลขคณิตจะมีค่าน้อยที่สุด แสดงดังรูปที่ 2.13



รูปที่ 2.13 การกระจายของข้อมูลแบบการกระจายที่เบ้ทางซ้าย [20]

## 2.7 งานวิจัยที่เกี่ยวข้อง (Related Work)

ในระยะเวลาหลายปีที่ผ่านมา ได้มีผู้เสนอแนะขั้นตอนวิธีการจัดการงานเข้าใช้ทรัพยากรในระบบการประมวลผลแบบกลุ่มเมฆด้วยวิธีการต่างๆ เพื่อให้ระบบมีการกระจายการทำงานแบบสมดุลและตรงกับความต้องการของผู้ใช้ ภายใต้ทรัพยากรที่มีอยู่อย่างจำกัด และเพื่อเป็นการเพิ่มประสิทธิภาพการทำงานของระบบการประมวลผลแบบกลุ่มเมฆ [2] เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B. Mondal, K. Dasgupta และ P. Dutta [21] ได้มีการนำขั้นตอนการแก้ไขปัญหาคาร์ปปีนเขาของสโทแคสติก (Stochastic Hill Climbing) มาใช้ในการจัดการงานแบบสมดุลในระบบการประมวลผลแบบกลุ่มเมฆ

ในปี 2010 Y. Fang, F. Wang และ J. Ge [22] ได้เสนอการจัดตารางการทำงานในประมวลผลแบบกลุ่มเมฆ โดยคำนึงถึงความต้องการของผู้ใช้งานที่มีการเปลี่ยนแปลงตลอดเวลาและคำนึงถึงการกระจายงานแบบสมดุลในสภาพแวดล้อมของการประมวลผลแบบกลุ่มเมฆ เมื่อผู้ใช้งานและความต้องการในการใช้งานทรัพยากรมีปริมาณเพิ่มมากขึ้น

J. Hu, J. Gu, G. Sun และ T. Zhao [23] ได้เสนอการจัดตารางการทำงานเพื่อให้ระบบมีการกระจายงานแบบสมดุล โดยได้มีการนำโครงสร้างต้นไม้ (Tree Structure) มาใช้ในขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) เพื่อในการพิจารณาจัดตารางงาน ซึ่งมีการพิจารณาถึงข้อมูลก่อนหน้าของการทำงานและสถานะปัจจุบันเพื่อกำหนดผลการทำงานล่วงหน้า ที่มีอิทธิพลต่อการทำงานของทั้งระบบ เพื่อเป็นการช่วยแก้ปัญหาของการเกิดสถานะไม่สมดุลของระบบ

K. Li, G. Xu, G. Zhao, Y. Dong และ D. Wang [24] เสนอขั้นตอนวิธี LBACO โดยการนำขั้นตอนวิธีหาค่าเหมาะสมที่สุดด้วยระบบอาณานิคม (Ant Colony Optimization) มาใช้ในการจัดตารางการทำงานเพื่อทำให้ระบบเกิดความสมดุลในการทำงานและลดเวลารอเข้าทำงานในระบบ

D. Karaboga, B. Gorkemli, C. Ozturk และ N. Karaboga [15] ศึกษาเกี่ยวกับขั้นตอนวิธีอาณานิคมผึ้งเทียมและการนำไปประยุกต์ใช้ในการแก้ไขปัญหาคาร์ปปีนเขา ขั้นตอนวิธีอาณานิคมผึ้งเทียมเป็นกระบวนการในการหาค่าที่เหมาะสมที่สุด ที่เลียนแบบมาจากพฤติกรรมในการหาอาหารของฝูงผึ้ง และเป็นกระบวนการหาค่าตอบชนิดหนึ่ง ที่อยู่ในกลุ่มที่มีแนวคิดมาจากการเลียนแบบพฤติกรรมของฝูงแมลงมีการนำขั้นตอนวิธีอาณานิคมผึ้งเทียมไปประยุกต์ใช้ในการแก้ไขปัญหาคาร์ปปีนเขา เช่น การค้นหาค่าที่เหมาะสม (Searching) การทำเหมืองข้อมูล (Data Mining) เป็นต้น

T. Mizan [25] ได้เสนอการจัดตารางการทำงานในระบบการประมวลผลแบบกลุ่มเมฆแบบผสม (Hybrid Cloud) โดยนำขั้นตอนวิธีวงจรผึ้งและขั้นตอนวิธีประเภทละโมบ (Greedy Algorithm) เพื่อลดเวลาในการจัดการในระบบให้น้อยลง

ในปี 2012 B. Bitam [1] ได้เสนอการนำขั้นตอนวิธีวงจรผึ้ง (Bees Life Algorithm: BLA) ไปประยุกต์ใช้ในการจัดตารางการทำงานในการประมวลผลแบบกลุ่มเมฆ โดยเลียนแบบพฤติกรรมการสืบพันธุ์และการค้นหาแหล่งอาหารของผึ้ง เพื่อประเมินประสิทธิภาพการเข้าทำงานในแต่ละทรัพยากรของระบบการประมวลผลแบบกลุ่มเมฆ โดยมุ่งเน้นเพื่อลดเวลาในการทำงานและลดความซับซ้อนในงานทำงานของระบบ

## บทที่ 3

# วิธีการดำเนินงานวิจัย

ในงานวิจัยนี้เป็นการศึกษาหาวิธีการตามหลักวิชาการในการจัดการตารางการผลิต โดยจะเปรียบเทียบผลลัพธ์จากการใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก สำหรับหาคำตอบในการจัดการตารางงานในการเข้าใช้เครื่องคอมพิวเตอร์เสมือน ทั้งในกรณีที่เครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันและต่างกัน ในระบบการประมวลผลแบบกลุ่มเมฆ ในบทนี้จะกล่าวถึงรูปแบบของปัญหาของงานวิจัยนี้ รวมทั้งการนำเอาขั้นตอนวิธีอาณานิคมผึ้งเทียม เพื่อใช้ในการหาคำตอบดังกล่าว โดยจะให้ความสำคัญกับเวลาการทำงานรวม (Makespan) เป็นตัวชี้วัดประสิทธิภาพ

ในบทนี้ผู้วิจัยได้เสนอวิธีการจัดการตารางงานในการเข้าใช้เครื่องคอมพิวเตอร์เสมือน โดยประยุกต์ใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก 3 วิธี คือ

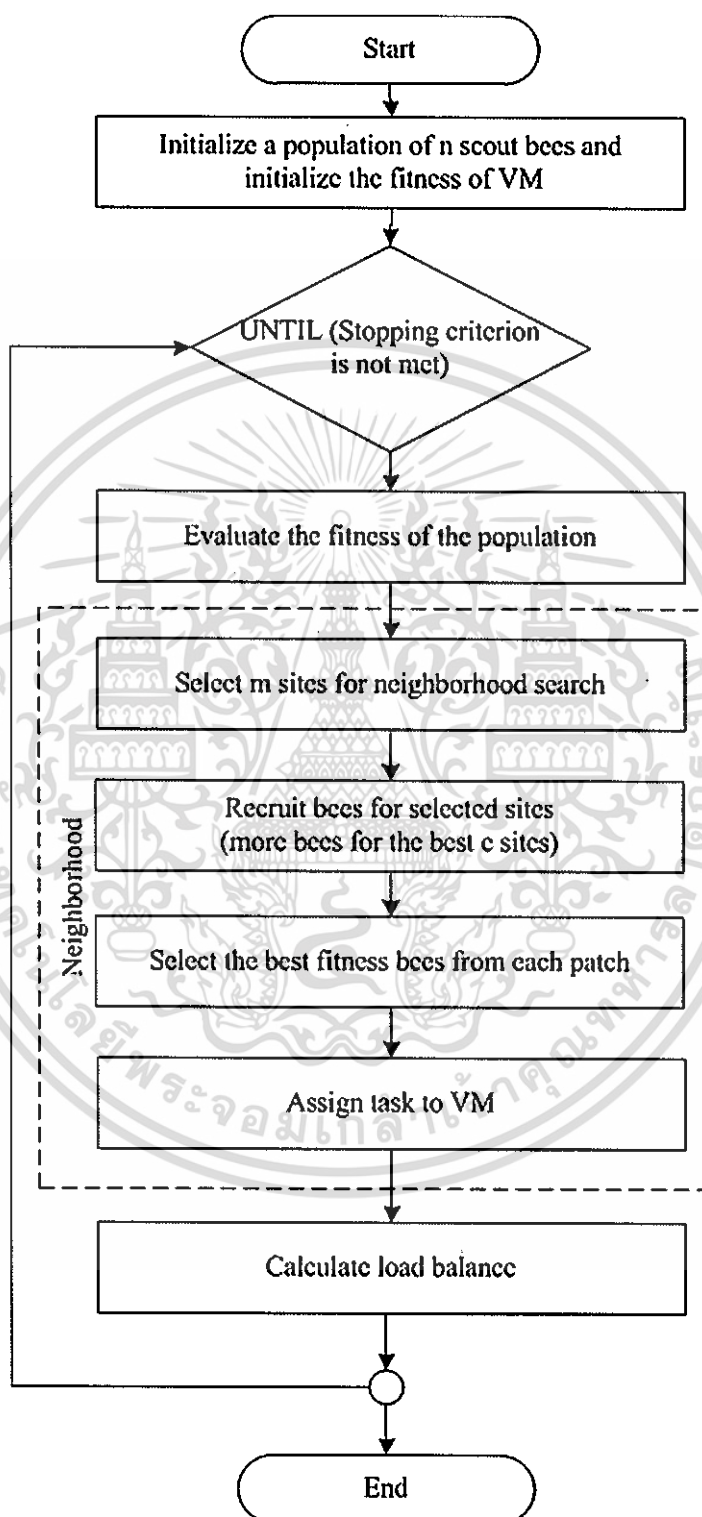
1. การประยุกต์ใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบงานมาก่อนทำก่อน (ABC\_FCFS)
2. การประยุกต์ใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบจัดงานที่มีขนาดเล็กที่สุดเข้าทำงานก่อน (ABC\_SJF)
3. การประยุกต์ใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบจัดงานที่มีขนาดใหญ่ที่สุดเข้าทำงานก่อน (ABC\_LJF)

### 3.1 การประยุกต์ใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียม (Artificial Bee Colony Algorithm)

ในงานวิจัยได้นำเอาขั้นตอนวิธีอาณานิคมผึ้งเทียมมาประยุกต์ใช้ในการหาคำตอบ สำหรับปัญหาการจัดการตารางงานเพื่อเข้าใช้เครื่องคอมพิวเตอร์เสมือนให้เหมาะสมกับสภาพแวดล้อม ทั้งในกรณีที่เครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันและต่างกัน บนระบบการประมวลผลแบบกลุ่มเมฆ และในการจัดการตารางงานนั้นจะต้องอยู่บนพื้นฐานของการกระจายงานในระบบแบบสมดุลและลดเวลาการทำงานรวมให้น้อยลง [26, 27] โดยมีขั้นตอนดังต่อไปนี้

กำหนดให้  $VM = \{VM_1, VM_2, VM_3, \dots, VM_N\}$  เมื่อ  $N$  คือจำนวนของเครื่องคอมพิวเตอร์เสมือนทั้งหมดที่มีในระบบและ  $Task = \{task_1, task_2, task_3, \dots, task_K\}$  เมื่อ  $K$  คือจำนวนของงานทั้งหมดในระบบ รูปแบบการเข้าทำงานในระบบเป็นการทำงานที่ไม่สามารถขัดจังหวะหรือแทรกกลางกันได้ (Non-Preemptive Process) ฟังก์ชันของขั้นตอนวิธีการจัดการตารางงานในการเข้าใช้

เครื่องคอมพิวเตอร์เสมือน เพื่อให้มีการกระจายงานในระบบแบบสมมูล โดยประยุกต์ใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียม แสดงดังรูปที่ 3.1



รูปที่ 3.1 ฟังงานแสดงการจัดการตารางงานเพื่อเข้าใช้เครื่องคอมพิวเตอร์เสมือนและระบบมีการ

กระจายงานแบบสมมูล โดยประยุกต์ใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียม

เอกสารนี้เป็นเอกสารทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กำหนดผึ้งสำรวจจำนวน  $n$  ตัว และกำหนดค่าที่เหมาะสม (Fitness) เริ่มต้นให้กับแต่ละเครื่องคอมพิวเตอร์เสมือน

กำหนดจำนวนผึ้งสำรวจ (Scout Bees) จำนวน  $n$  ตัว เพื่อใช้ในหา VM ที่มีความเหมาะสม และกำหนดค่าที่เหมาะสม (Fitness) เริ่มต้นให้กับแต่ละ VM ค่าที่เหมาะสมสามารถคำนวณได้จากสมการที่ (3.1)

$$Fitness_j = pe\_num_j \times pe\_mips_j + vm\_bw_j \quad (3.1)$$

เมื่อ	$Fitness_j$	=	ค่า Fitness เริ่มต้นของ VM ลำดับที่ $j$
	$pe\_num_j$	=	จำนวน Processor ใน VM ลำดับที่ $j$
	$pe\_mips_j$	=	Million Instructions Per Second ของ Processor ใน VM ลำดับที่ $j$
	$vm\_bw_j$	=	ปริมาณแบนด์วิธ (Bandwidth) ของ VM ลำดับที่ $j$

2. ส่งผึ้งสำรวจแบบสุ่มเข้าไปยัง VM และคำนวณค่าที่เหมาะสม

ส่งผึ้งสำรวจจำนวน  $n$  ตัว แบบสุ่ม เข้าไปในระบบเพื่อเลือก VM ที่เหมาะสม และคำนวณค่าที่เหมาะสม ดังสมการที่ (3.2)

$$Fitness_{ij} = \frac{\sum_{i=1}^n task\_length_{ij}}{Evaluate\ capacity\ of\ VM_j\ (capacity_j)} \quad (3.2)$$

เมื่อ	$Fitness_{ij}$	=	ค่าที่เหมาะสมของ VM ลำดับที่ $j$ จากผึ้งสำรวจตัวที่ $i$ (Capacity of $VM_j$ with bee number of $i$ )
	$\sum_{i=1}^n task\_length_{ij}$	=	ผลรวมของขนาดของงาน (Task) ทั้งหมดที่ทำงานใน VM ลำดับที่ $j$
	$capacity_j$	=	ความสามารถในการทำงานของ VM ลำดับที่ $j$ สามารถคำนวณได้จากสมการที่ (3.3)

$$capacity_j = pe\_num_j \times pe\_mips_j + vm\_bw_j \quad (3.3)$$

เมื่อ	$pe\_num_j$	=	จำนวน Processor ใน VM ลำดับที่ $j$
	$pe\_mips_j$	=	Million Instructions Per Second ของ Processor ใน VM ลำดับที่ $j$
	$vm\_bw_j$	=	ปริมาณแบนด์วิธ (Bandwidth) ของ VM ลำดับที่ $j$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. เลือก VM ที่มีค่าเหมาะสมที่สุดจำนวน $m$ เครื่อง

ผึ้งเฝ้าดู (Onlooker Bee) จะเลือกแหล่งอาหารที่มีค่า Fitness ดีที่สุดจำนวน  $M$  ตัว ซึ่ง  $m$  ที่ถูกเลือกจะกลายเป็น Select Bee และพื้นที่ที่ Select Bee ไปเยี่ยมชมมา จะเรียกเป็นพื้นที่ที่ถูกเลือก (Neighborhood Search)

### 4. ส่งผึ้งจำนวนมากเข้าค้นหาในพื้นที่ที่ถูกเลือก

กำหนดผึ้งจำนวนมากเข้าค้นหาในพื้นที่ที่ถูกเลือก ส่งผึ้งจำนวนมากกว่า  $e$  ที่เลือก จำนวนผึ้งที่ส่ง  $nep$  ตัว และคำนวณค่าที่เหมาะสม ดังสมการที่ (3.4)

$$fit_{ij} = \frac{\sum_{i=1}^n task\_length_i + InputFile\_length}{Evaluate\ capacity\ of\ VM_j\ (capacity_j)} \quad (3.4)$$

เมื่อ  $InputFile\_length$  = ขนาดของงานที่ต้องการเข้าใช้งาน VM

### 5. เลือก VM ที่ดีที่สุด

เลือก VM ที่มีค่า fitness ดีที่สุด 1 VM และส่งงาน 1 งาน เพื่อเข้าทำงานใน VM ที่ถูกเลือก

### 6. กำหนดงาน 1 งาน เพื่อเข้าทำงานใน VM ที่ถูกเลือก

ส่งงาน 1 งาน เพื่อเข้าทำงานใน VM ที่ถูกเลือก

### 7. คำนวณเพื่อดูการกระจายงานภายในกลุ่ม (Load Balance)

หลังจากมีการจัดการ task ให้เข้าทำงานใน VM จะมีการคำนวณภาระงานทั้งหมดในปัจจุบันของ VM ทั้งหมด โดยคำนวณจากค่าส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation: S.D.) เพื่อวัดภาระงานทั้งหมดใน VM สามารถคำนวณได้ดังสมการที่ (3.5 – 3.7)

$$S.D. = \sqrt{\frac{1}{n} \sum_{j=0}^n (X_j - \bar{X})^2} \quad (3.5)$$

เวลารวมในการทำงานของแต่ละ VM :

$$X_j = \frac{\sum_{i=1}^k task\_length_i}{capacity_j} \quad (3.6)$$

ค่าเฉลี่ยของเวลารวมในการทำงานของ VM ทั้งหมดในระบบ :

$$\bar{X} = \frac{\sum_{j=1}^n X_j}{n} \quad (3.7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ	S.D.	=	ค่าส่วนเบี่ยงเบนมาตรฐาน
	N	=	จำนวนงานทั้งหมดในระบบ
	$X_j$	=	เวลารวมในการทำงานของแต่ละ VM สามารถคำนวณได้จากสมการที่ (3.6)
	$\bar{X}$	=	ค่าเฉลี่ยของเวลารวมในการทำงานของ VM ทั้งหมดในระบบ สามารถคำนวณได้จากสมการที่ (3.7)

ถ้า S.D. ของภาระงานของ VM มีค่าน้อยกว่าหรือเท่ากับค่าเฉลี่ยแล้วถือได้ว่า ระบบอยู่ในสภาวะสมดุล และในทางกลับกัน ถ้า S.D. มีค่ามากกว่าค่าเฉลี่ยแล้วจะถือว่า ระบบอยู่ในสภาวะของการทำงานแบบไม่สมดุล



## บทที่ 4

### การทดลองและผลการทดลอง

จากบทที่ 3 ได้กล่าวถึงวิธีการประยุกต์ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียมเพื่อใช้ในการแก้ไขปัญหาการจัดตารางการทำงานของเครื่องเสมือนในการประมวลผลแบบกลุ่มเมฆ ทั้งในกรณีที่เครื่องเสมือนมีสภาพแวดล้อมภายในเหมือนกันและต่างกัน ในบทนี้จะแสดงรายละเอียดในการทดลองลักษณะของชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพ และวิธีการทดสอบ เพื่อเป็นการทดสอบประสิทธิภาพของขั้นตอนวิธีที่ได้นำเสนอไป ซึ่งมีรายละเอียดดังต่อไปนี้

#### 4.1 ข้อมูลที่ใช้ในการทดสอบประสิทธิภาพ

ข้อมูลที่ใช้ในการทดสอบเป็นข้อมูลที่มีจากการสุ่มแบบอิสระ โดยถูกจำกัดด้วยขนาดของงานที่ใช้ในการทดสอบประสิทธิภาพ ในการทดสอบประสิทธิภาพได้มีการแบ่งชุดข้อมูลออกเป็น 10 ชุดข้อมูล แต่ละชุดจะมีการจัดจำนวนของงานออกเป็นชุดๆ ละ 100, 200, 300, 400, 500, 600 และ 700 งานตามลำดับ แต่ละชุดข้อมูลจะมีขนาดของงานและรูปแบบการกระจายข้อมูลแสดงดังตารางที่ 4.1 และชุดข้อมูลตัวอย่างแสดงดังภาคผนวก ก

ตารางที่ 4.1 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพ

ชุดข้อมูล	ขนาดของงาน (MI)	การกระจายข้อมูล
1	5000 - 20000	สุ่มอิสระ
2	5000 - 20000	สุ่มอิสระ
3	5000 - 20000	สุ่มอิสระ
4	15000 - 35000	สุ่มอิสระ
5	25000 - 45000	สุ่มอิสระ
6	35000 - 55000	สุ่มอิสระ
7	45000 - 65000	สุ่มอิสระ
8	5000 - 20000	โค้งปกติ
9	5000 - 20000	เบ้ขวา
10	5000 - 20000	เบ้ซ้าย

หมายเหตุ: MI = Million Instruction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 การเตรียมข้อมูล

ในขั้นแรกของการทดสอบประสิทธิภาพจะทำการจัดเรียงข้อมูล ดังตารางที่ 4.2 โดยแบ่งการจัดเรียงข้อมูลเป็น 3 รูปแบบ คือ

1. จัดเรียงข้อมูลตามลำดับการมาถึงของงาน (FCFS)
2. จัดเรียงข้อมูลตามขนาดของงาน โดยเรียงข้อมูลจากเล็กที่สุดไปหาขนาดใหญ่ที่สุด (SJF)
3. จัดเรียงข้อมูลตามขนาดของงาน โดยเรียงข้อมูลจากใหญ่ที่สุดไปหาขนาดเล็กที่สุด (LJF)

ตารางที่ 4.2 ตัวอย่างในการจัดเรียงข้อมูล

FCFS	SJF	LJF
5163	5163	11965
7832	5322	11290
11290	7256	10897
10897	7832	9159
7256	9159	7832
11965	10897	7256
5322	11290	5322
9159	11965	5163

## 4.3 เครื่องมือที่ใช้ในงานวิจัย

เครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบ มีคุณสมบัติดังนี้

Notebook	:	HP ProBook 6450b
หน่วยประมวลผลกลาง (CPU)	:	Intel® Core™ i5-450M Processor 2.40 GHz
หน่วยความจำหลัก (RAM)	:	8 GB DDR3
หน่วยความจำสำรอง (Hard Disk)	:	320 GB
ระบบปฏิบัติการ (OS)	:	Windows 7 Enterprise
โปรแกรมที่ใช้ในการจำลองระบบการประมวลผลแบบกลุ่มเมฆ	:	CloudSim-3.0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4 ผลการทดลองกรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกัน

##### 4.4.1 การตั้งค่าพารามิเตอร์สภาพแวดล้อมในการทดลอง

การตั้งค่าพารามิเตอร์แบบจำลองเพื่อประเมินประสิทธิภาพของระบบการประมวลผลแบบกลุ่มเมฆ กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันทั้งหมด แสดงค่าดังตารางที่ 4.3

ตารางที่ 4.3 ค่าพารามิเตอร์ของแบบจำลองการประมวลผลแบบกลุ่มเมฆ กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันทั้งหมด

Type	Parameter	Value
Datacenter	Number of Datacenter	10
	Number of Host	5
	Type of Manager	Space_Shared, Time_Shared
	Number of PE per Host	4
	MIPS of PE	2000
	Host Memory (RAM) (MB)	10240
	Datacenter Cost (The cost of using processing in this resource)	10
Virtual Machine (VM)	Total number of VM	30 - 210
	MIPS of PE	2000
	Number of PE per VM	1
	VM Memory (RAM) (MB)	1024
	Bandwidth	1000 Bit
	Type of Manager	Time_Shared
Task	Total number of Task	100 - 700
	Length of Task (MI)	5000 - 20000
	Number of PE per requirement	1
	Type of Manager	Space_Shared

**หมายเหตุ:** PE = Processing Element, MIPS = Million Instructions Per Second,

MB = Megabyte, MI = Million Instruction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4.2 การตั้งค่าพารามิเตอร์ในขั้นตอนวิธีอาณานิคมผึ้งเทียม

การตั้งค่าพารามิเตอร์ในส่วนของขั้นตอนวิธีอาณานิคมผึ้งเทียม แสดงค่าดังตารางที่ 4.4

ตารางที่ 4.4 ค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมผึ้งเทียม (Homogenous)

Symbol	Parameter	Value
n	Number of scout bees	2000
m	Number of sites selected out of n visited sites	5
e	Number of best site out of m selected sites	1
nep	Number of bees recruited for best e sites	1600
nsp	Number of bees recruited for other (m-e) selected sites	400

#### 4.4.3 ผลการทดสอบประสิทธิภาพกรณีทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลง

การประเมินประสิทธิภาพในการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือน โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก (ABC\_FCFS, ABC\_SJF และ ABC\_LJF) ทำการเปรียบเทียบกับการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว (FCFS, SJF และ LJF) กำหนดจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าเท่ากับ 50 เครื่อง และแบ่งการทดสอบออกเป็น 10 กรณี คือ

##### 4.4.3.1 ผลการทดสอบประสิทธิภาพ เมื่อชุดข้อมูลมีการเปลี่ยนแปลง

การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลง เมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่แสดงค่าดังตารางที่ 4.5 เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางการทำงาน โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก

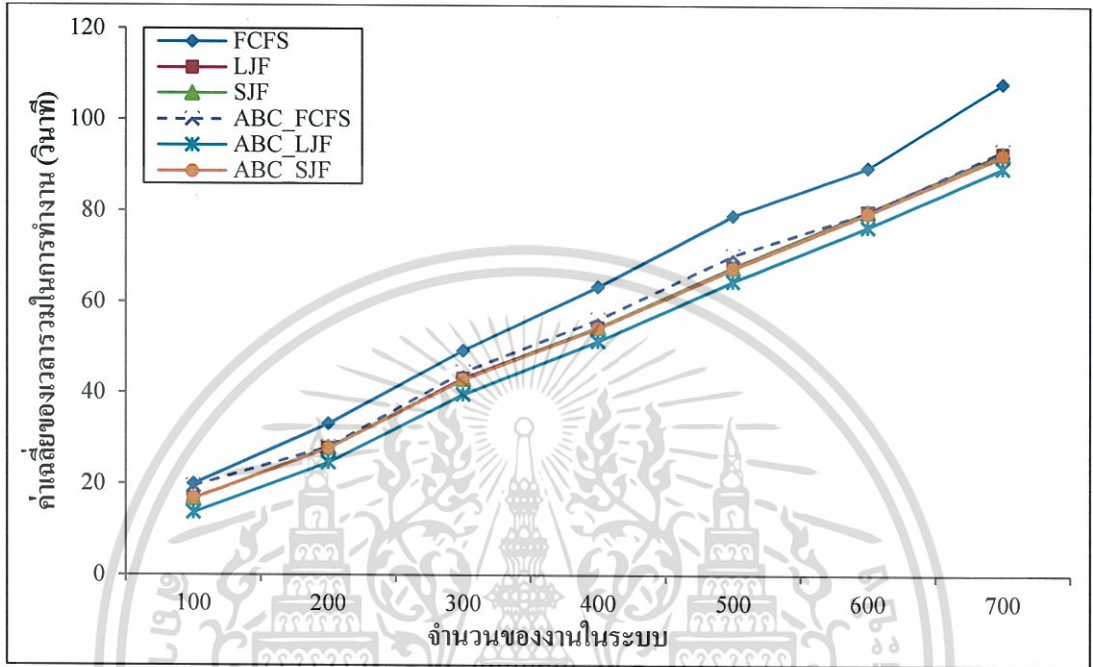
ตารางที่ 4.5 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของกรณีที่ 4.4.3.1

ชุดข้อมูล	ขนาดของงาน (MI)	การกระจายข้อมูล
1	5000 - 20000	สุ่มอิสระ
2	5000 - 20000	สุ่มอิสระ
3	5000 - 20000	สุ่มอิสระ

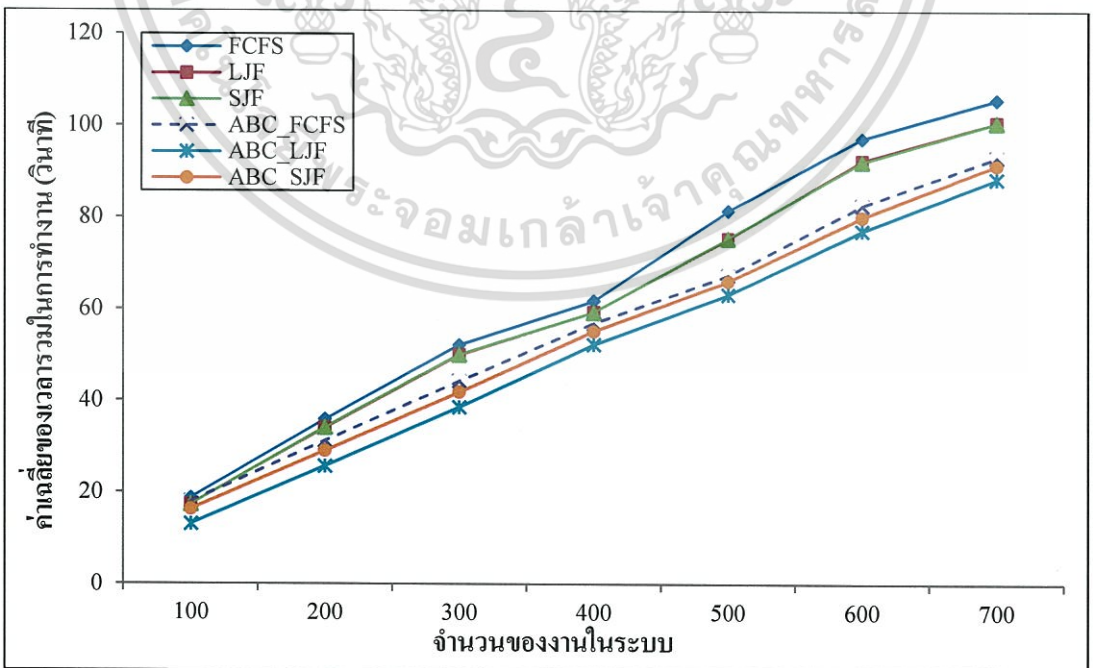
หมายเหตุ: MI = Million Instruction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานและชุดข้อมูลที่ใช้ในการทดสอบมีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันทั้งหมด โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ย (Average Processing Time) ของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.1 – 4.3

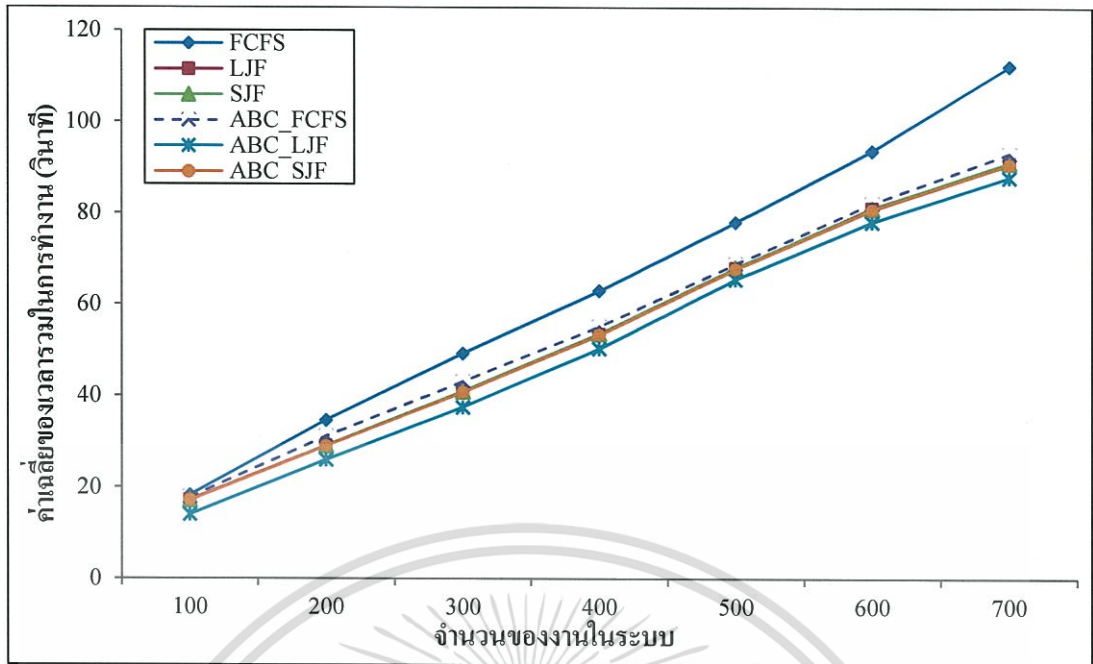


รูปที่ 4.1 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 เมื่องานมีปริมาณเพิ่มขึ้น (Homogenous)



รูปที่ 4.2 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 2 เมื่องานมีปริมาณเพิ่มขึ้น (Homogenous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 3 เมื่องานมีปริมาณเพิ่มขึ้น (Homogenous)

จากรูปที่ 4.1 – 4.3 แสดงเวลาทำงานรวมเฉลี่ย โดยทำการเปรียบเทียบขั้นตอนการจัดการงาน กรณีเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกัน โดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันความถูกต้องกับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลการทดสอบประสิทธิภาพแสดงให้เห็นว่า ABC\_LJJ มีประสิทธิภาพในการจัดการงานดีที่สุด เมื่อปริมาณงานในระบบมีปริมาณเพิ่มมากขึ้น แม้ว่าข้อมูลภายในระบบจะมีการเปลี่ยนแปลง การจัดลำดับงานแบบ ABC\_LJJ ยังคงสามารถทำงานได้อย่างมีประสิทธิภาพเช่นเดิม

#### 4.4.3.2 ผลการทดสอบประสิทธิภาพ เมื่อขนาดของงานมีการเปลี่ยนแปลง

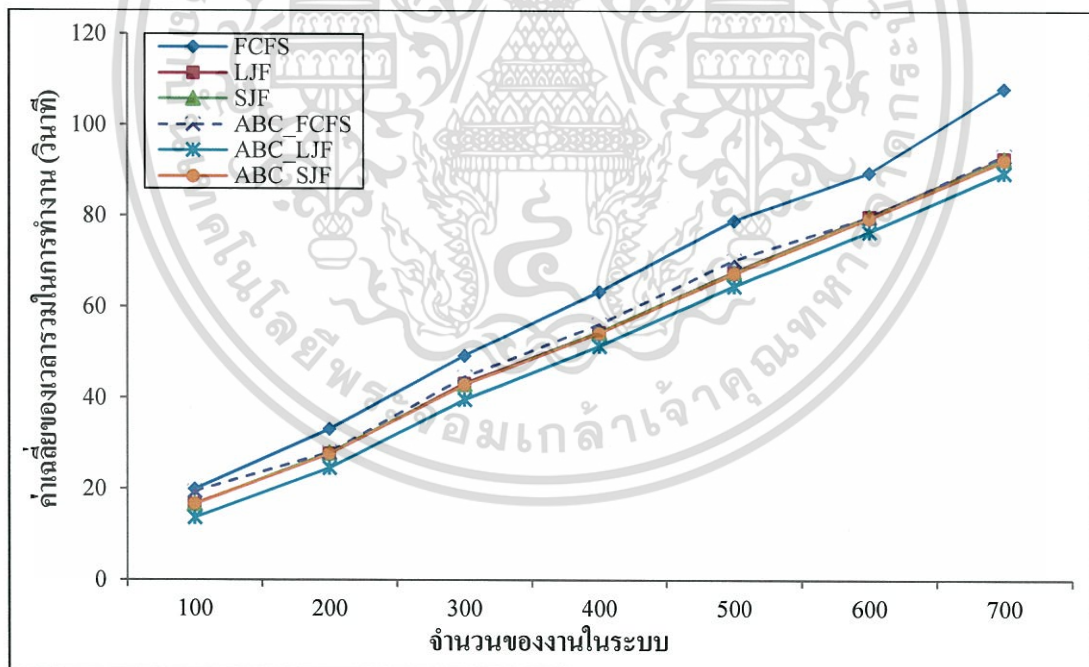
การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลง เมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลและขนาดของงาน ดังตารางที่ 4.6 เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดการการทำงานโดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันความถูกต้องกับการจัดลำดับงานแบบฮิวริสติก

ตารางที่ 4.6 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของการทดสอบกรณีที่ 4.4.3.2

ชุดข้อมูล	ขนาดของงาน (MI)	การกระจายข้อมูล
1	5000 - 20000	สุ่มอิสระ
4	15000 - 35000	สุ่มอิสระ
5	25000 - 45000	สุ่มอิสระ
6	35000 - 55000	สุ่มอิสระ
7	45000 - 65000	สุ่มอิสระ

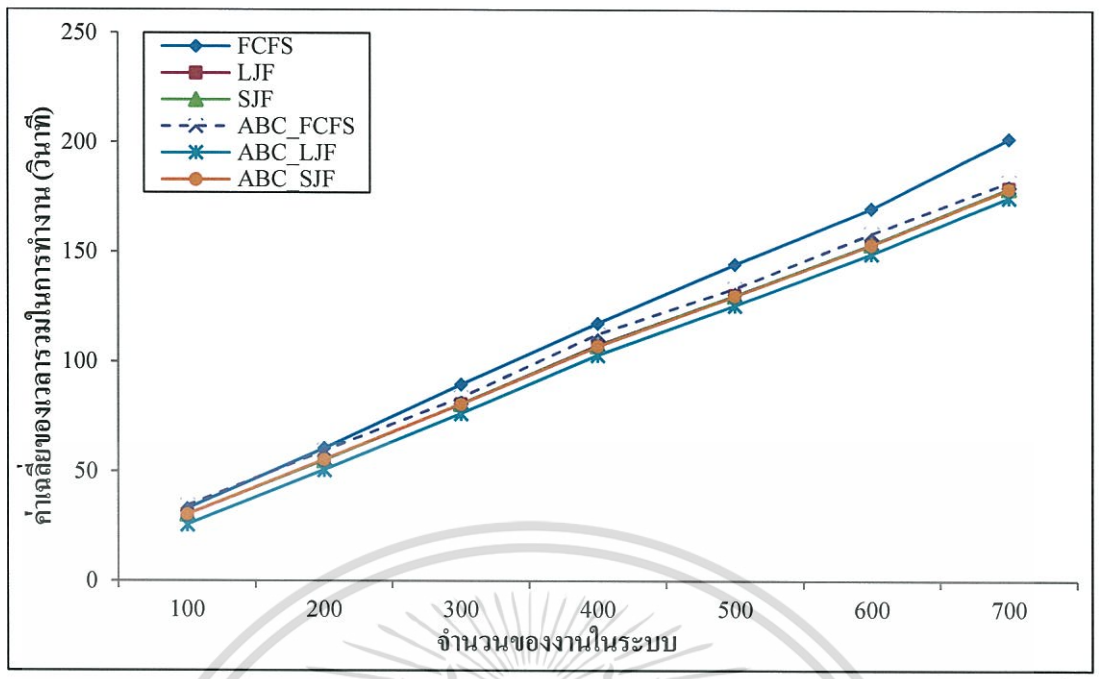
หมายเหตุ: MI = Million Instruction

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลงและขนาดของงานมีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันทั้งหมด โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.4 - 4.8

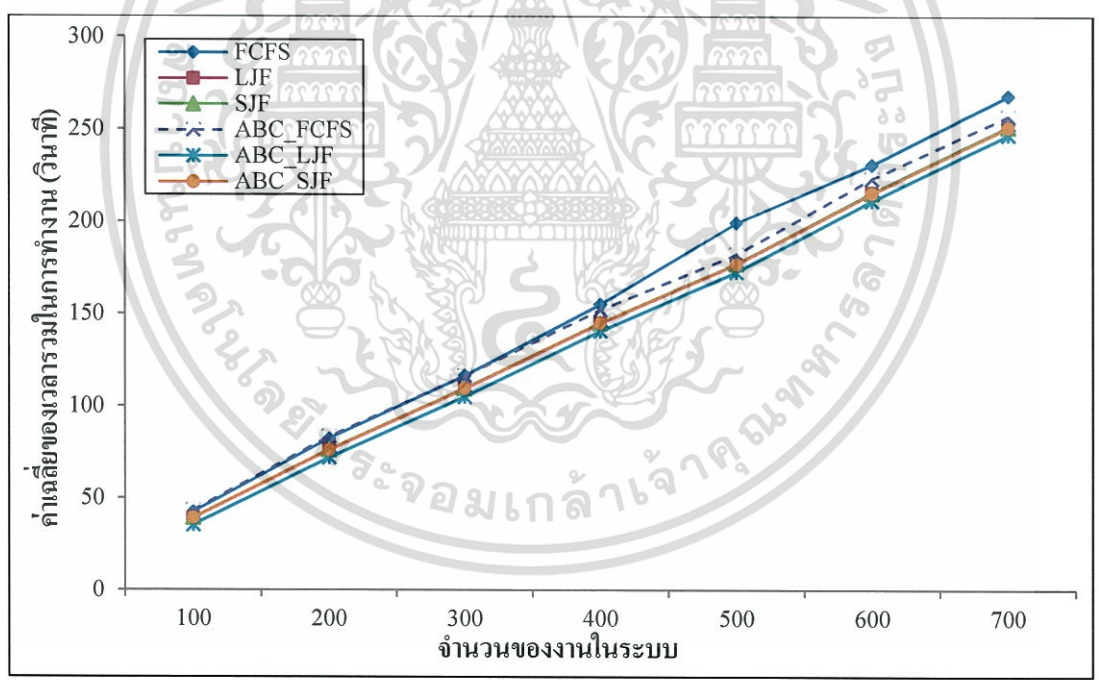


รูปที่ 4.4 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 เมื่องานมีขนาด 5000 - 20000 (Homogenous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

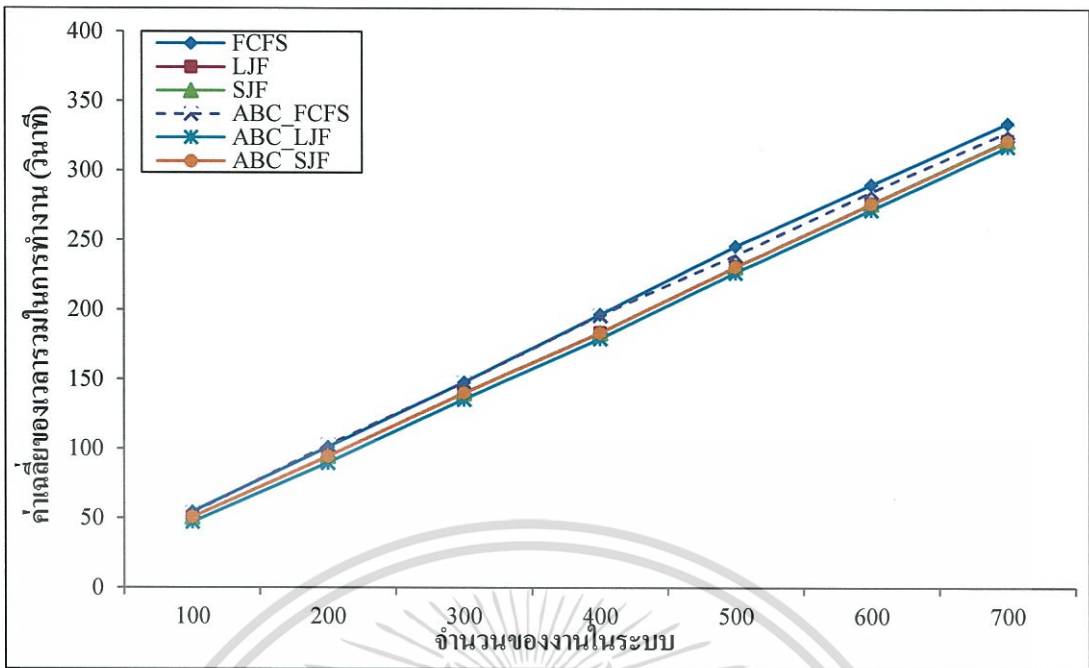


รูปที่ 4.5 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 4 เมื่องานมีขนาด 15000 - 35000 (Homogenous)

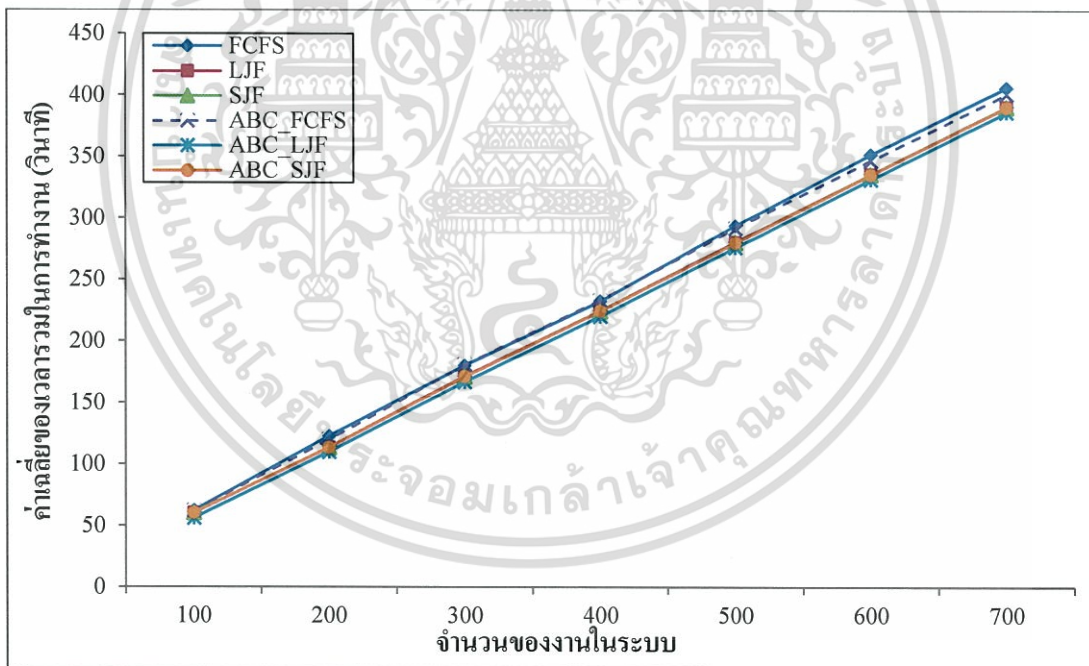


รูปที่ 4.6 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 5 เมื่องานมีขนาด 25000 - 45000 (Homogenous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 6 เมื่องานมีขนาด 35000 - 55000 (Homogenous)



รูปที่ 4.8 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 7 เมื่องานมีขนาด 45000 - 65000 (Homogenous)

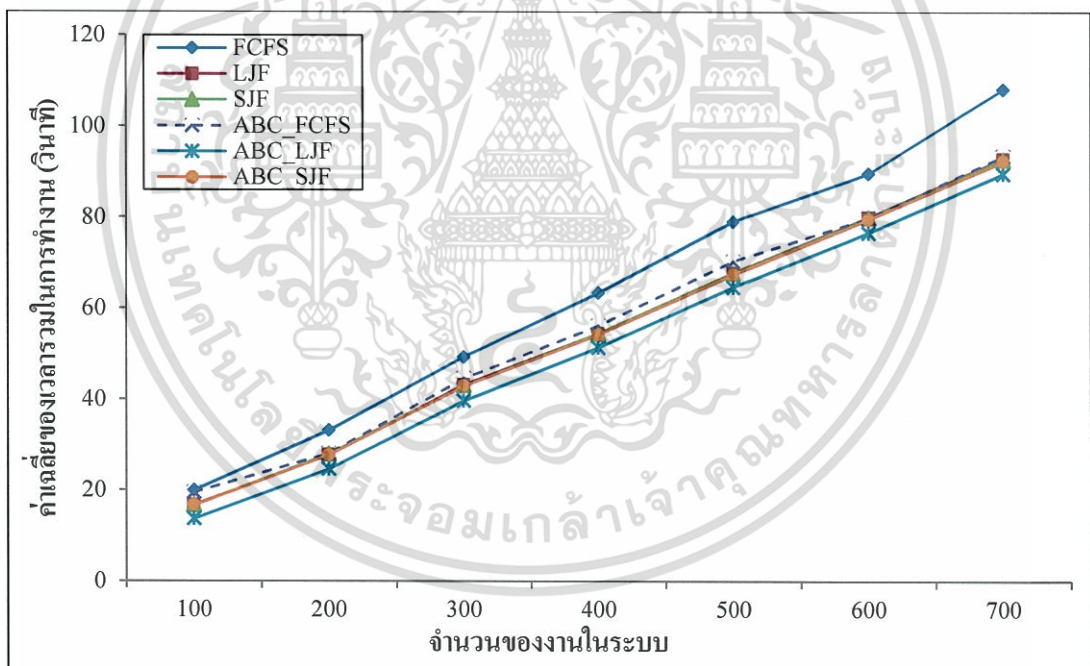
จากรูปที่ 4.4 – 4.8 แสดงเวลาทำงานรวมเฉลี่ย โดยทำการเปรียบเทียบขั้นตอนการจัดตารางงาน เมื่อเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกัน โดยใช้ขั้นตอนวิธีอาณานิคมฝังเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ABC\_LJF มีประสิทธิภาพในการจัดตารางงานดี เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่สุด เมื่อปริมาณงานในระบบมีปริมาณเพิ่มมากขึ้น แม้ว่าขนาดของงานมีการเปลี่ยนแปลง การจัดลำดับงานแบบ ABC\_LJF ยังคงสามารถทำงานได้อย่างมีประสิทธิภาพเช่นเดิม

#### 4.4.3.3 ผลการทดสอบประสิทธิภาพเมื่อจำนวน PE ของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง

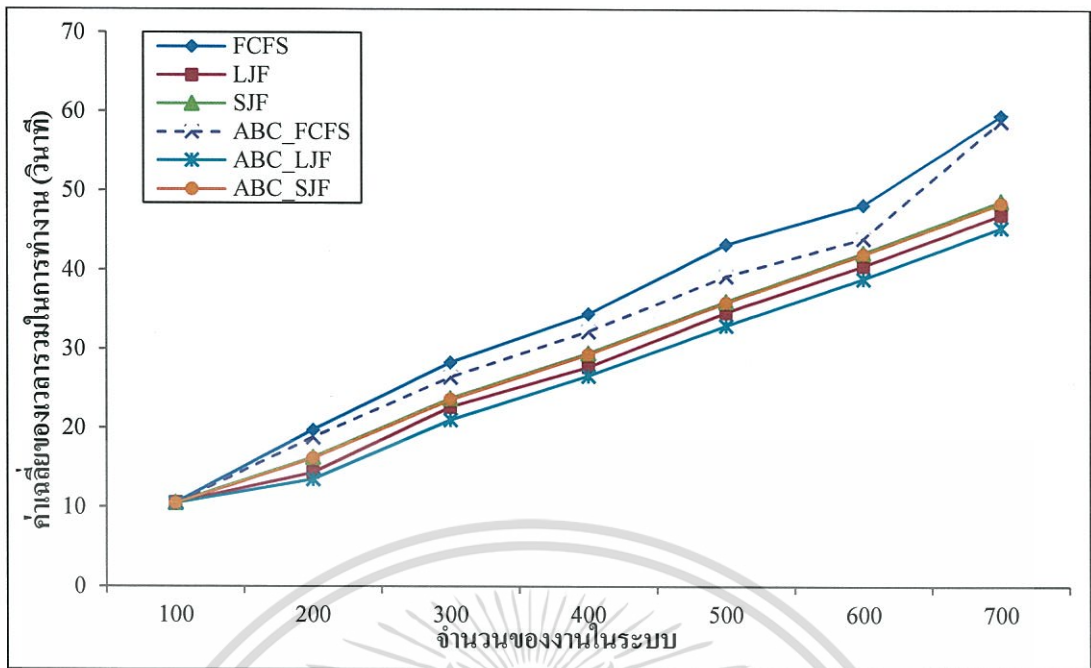
การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลง เมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่ 1 เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางงาน โดยใช้ขั้นตอนวิธีอาณานิคม ผังเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก ในการทดสอบจะทำการเปรียบเทียบการทำงานเมื่อกำหนดจำนวน PE (Processing Element) ของเครื่องคอมพิวเตอร์เสมือน 3 ค่า คือ 1, 2 และ 4

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานและจำนวน PE ของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.9 – 4.11

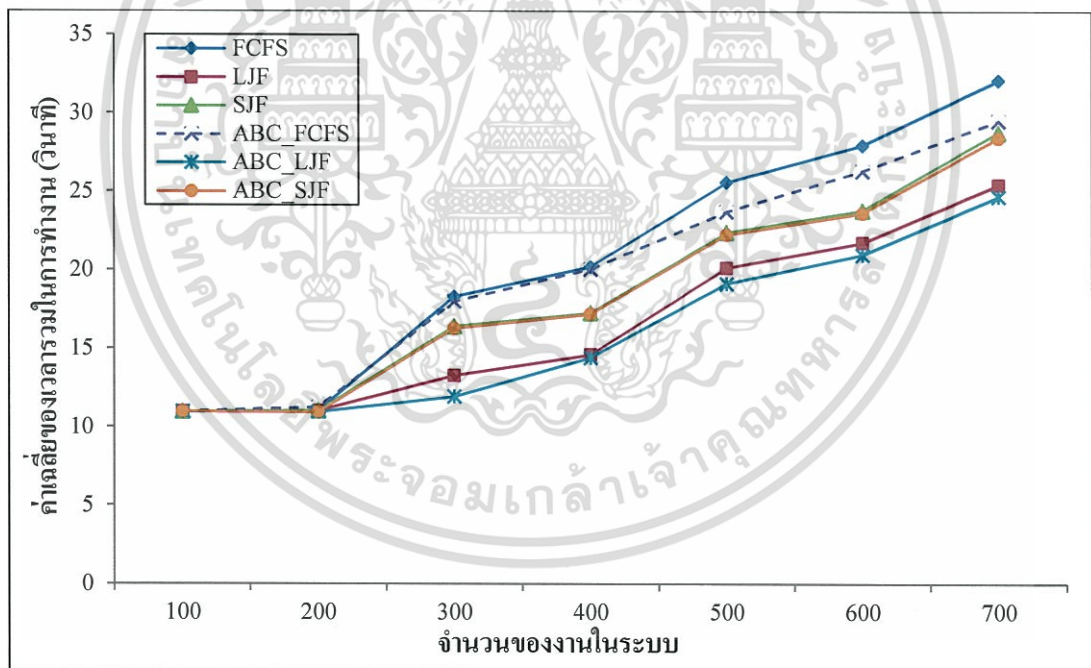


รูปที่ 4.9 เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและกำหนดค่า PE = 1 (Homogenous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและกำหนดค่า PE = 2 (Homogenous)



รูปที่ 4.11 เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและกำหนดค่า PE = 4 (Homogenous)

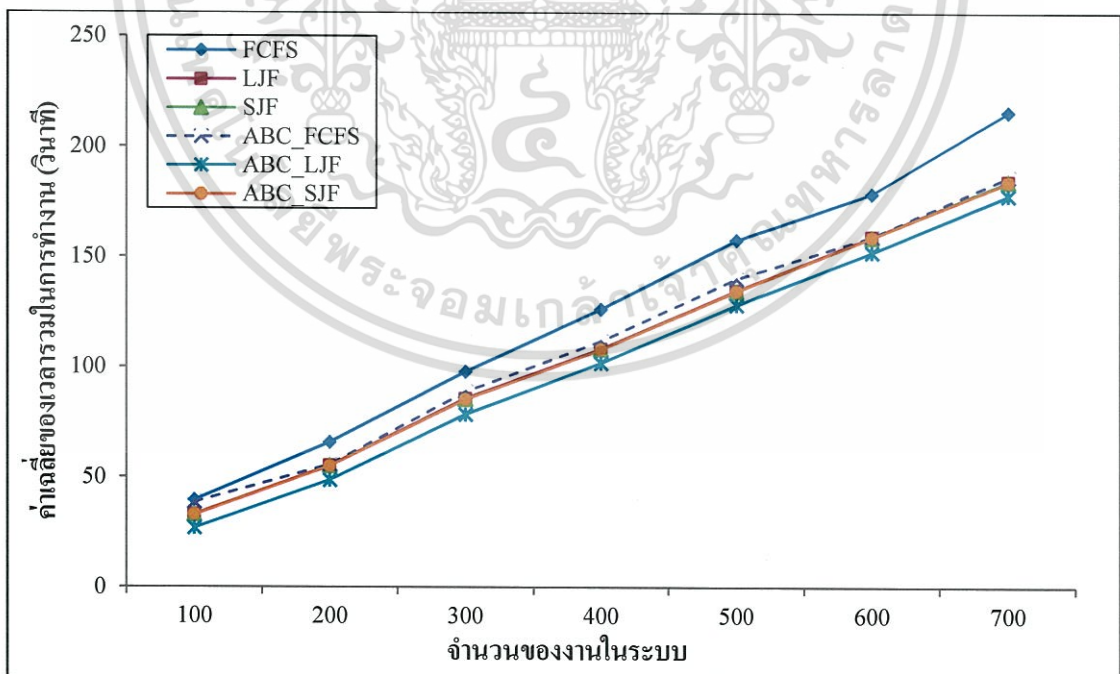
จากรูปที่ 4.9 – 4.11 แสดงเวลาทำงานรวมเฉลี่ย โดยทำการเปรียบเทียบขั้นตอนการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกัน โดยใช้ขั้นตอนวิธีอาณานิคมฝังเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า สำหรับ PE = 4 จำนวนงานที่ 100 และ 200 จะมีเอกสารเป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอญูหาเห็นาไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าเฉลี่ยของเวลารวมในการทำงานแต่ละอัลกอริทึมแตกต่างกันเพียงจุดศูนยิมเท่านั้น ซึ่งเมื่อจำนวนงานในระบบมีปริมาณมากกว่า 200 งานขึ้นไป จะเห็นได้ว่า ABC\_LJF มีประสิทธิภาพในการจัดการงานดีที่สุด จึงสรุปได้ว่า แม้ว่าจำนวน PE ของเครื่องคอมพิวเตอร์เสมือนจะมีการเปลี่ยนแปลง การจัดลำดับงานแบบ ABC\_LJF ยังคงสามารถทำงานได้อย่างมีประสิทธิภาพเช่นเดิม

#### 4.4.3.4 ผลการทดสอบประสิทธิภาพ เมื่อขนาด MIPS ของ PE ในเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง

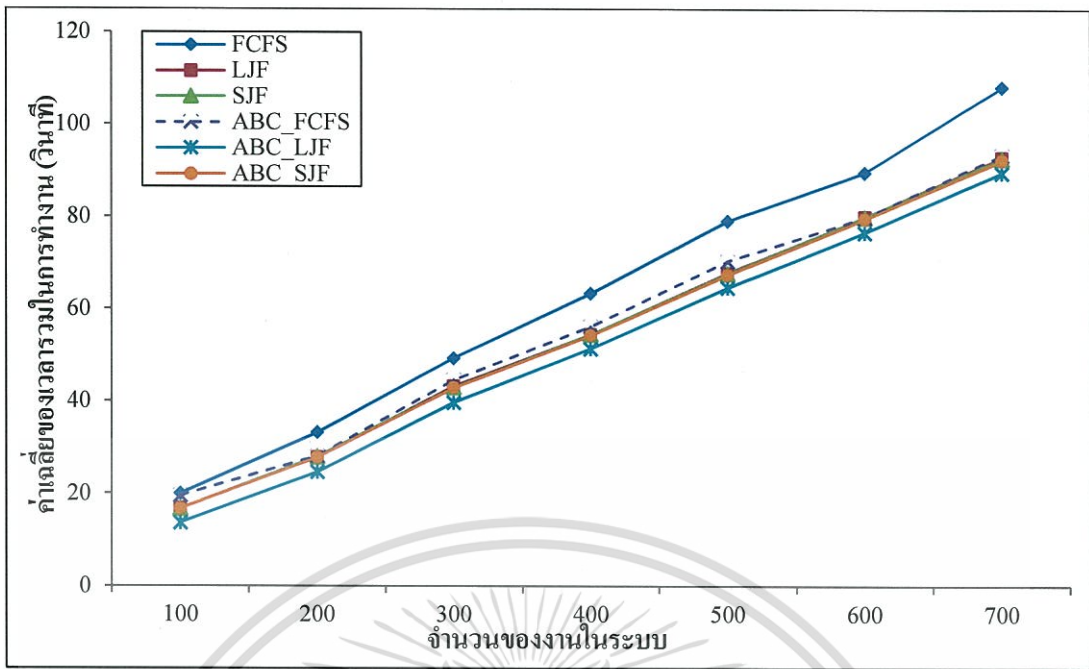
การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลงเมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองใช้ชุดข้อมูลที่ 1 เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดการการทำงาน โดยใช้ขั้นตอนวิธีอณานิคมผังเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก โดยในการทดสอบจะทำการเปรียบเทียบค่า MIPS (Million Instructions Per Second) ของ PE เมื่อค่า MIPS มีการเปลี่ยนแปลง โดยกำหนดให้ MIPS มีค่าเท่ากับ 1000 และ 2000

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลง และจำนวน MIPS ของ PE ในเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันทั้งหมด โดยแกน Y แสดงเวลารวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.12 และรูปที่ 4.13



รูปที่ 4.12 เวลาทำงานรวมเฉลี่ย เมื่องานเพิ่มขึ้นและกำหนดค่า MIPS = 1000 (Homogenous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 เวลาทำงานรวมเฉลี่ย เมื่องานเพิ่มขึ้นและกำหนดค่า MIPS = 2000 (Homogenous)

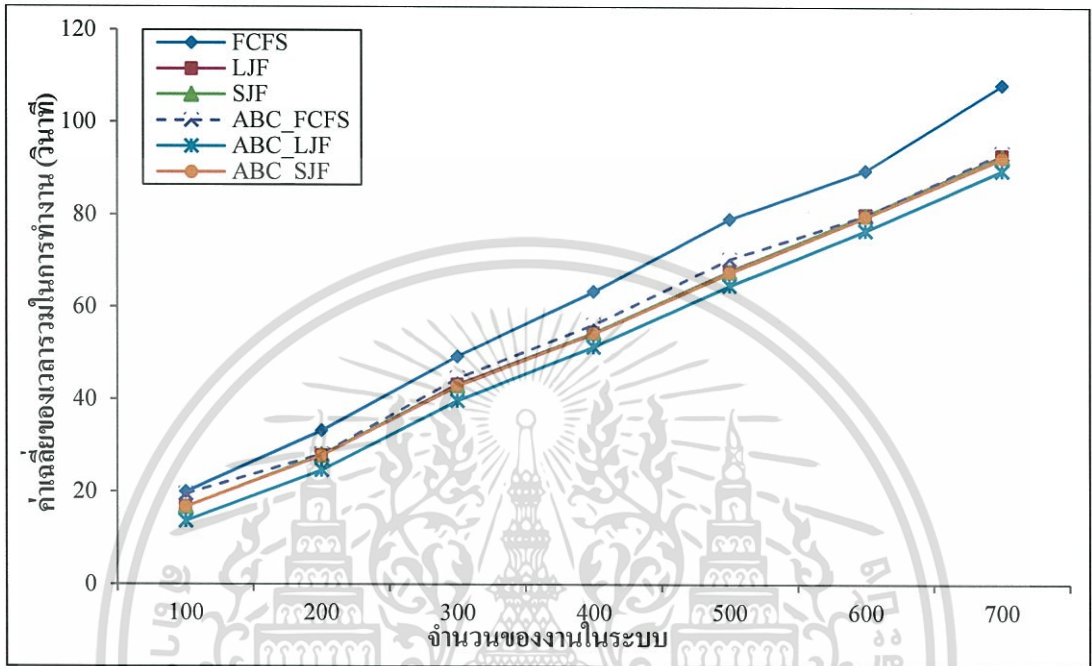
จากรูปที่ 4.12 และรูปที่ 4.13 แสดงเวลาทำงานรวมเฉลี่ย โดยทำการเปรียบเทียบขั้นตอนการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกัน โดยใช้ขั้นตอนวิธีตามนิคมผังเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า เมื่อมีการเปลี่ยนแปลงค่า MIPS ของ PE ในเครื่องคอมพิวเตอร์เสมือน ลักษณะของกราฟที่ได้จะมีลักษณะคล้ายคลึงกันแต่เวลารวมเฉลี่ยในการทำงานในระบบจะมีค่าต่างกัน ซึ่งสามารถสรุปได้ว่า ABC\_LJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด เมื่อปริมาณงานในระบบมีปริมาณเพิ่มมากขึ้น แม้ว่าจำนวน MIPS ของ PE ในเครื่องคอมพิวเตอร์จะมีการเปลี่ยนแปลง การจัดลำดับงานแบบ ABC\_LJF ยังคงสามารถทำงานได้อย่างมีประสิทธิภาพเช่นเดิม

#### 4.4.3.5 ผลการทดสอบประสิทธิภาพ เมื่อจำนวน RAM ของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง

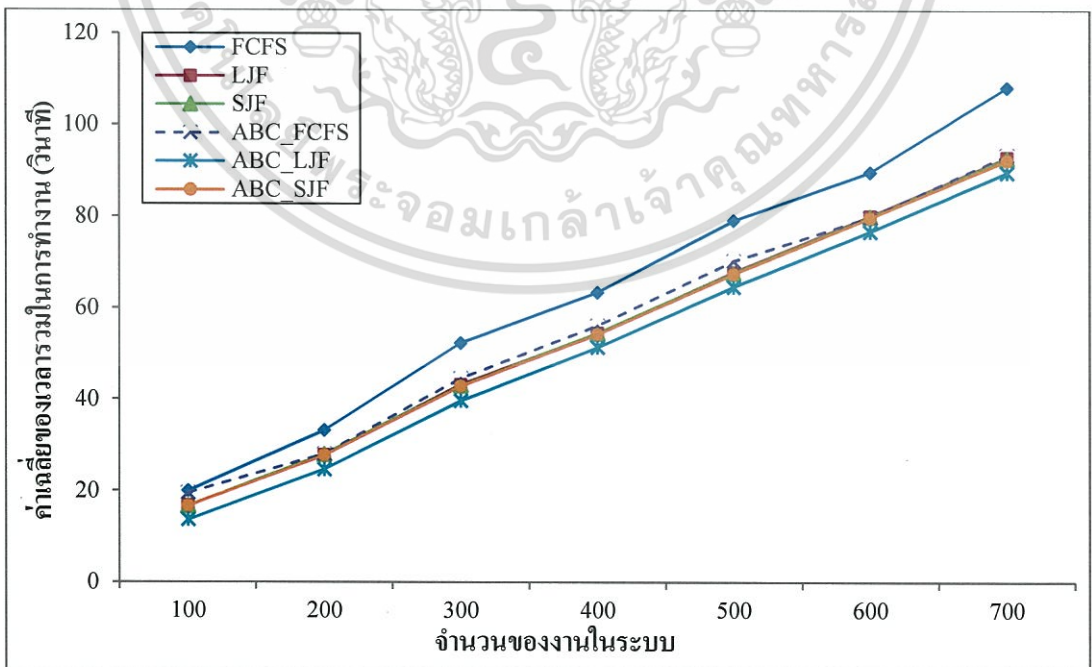
การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลงเมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่ 1 เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางการทำงานโดยใช้ขั้นตอนวิธีตามนิคมผังเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก โดยในการทดสอบจะทำการเปรียบเทียบเมื่อค่า RAM ของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง โดยทำการเปรียบเทียบระหว่าง RAM เท่ากับ 1024 และ 2048

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลงและจำนวน RAM ของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันทั้งหมด โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.14 และรูปที่ 4.15



รูปที่ 4.14 เวลาทำงานรวมเฉลี่ย เมื่องานเพิ่มขึ้นและกำหนดค่า RAM = 1024 (Homogenous)



รูปที่ 4.15 เวลาทำงานรวมเฉลี่ย เมื่องานเพิ่มขึ้นและกำหนดค่า RAM = 2048 (Homogenous)

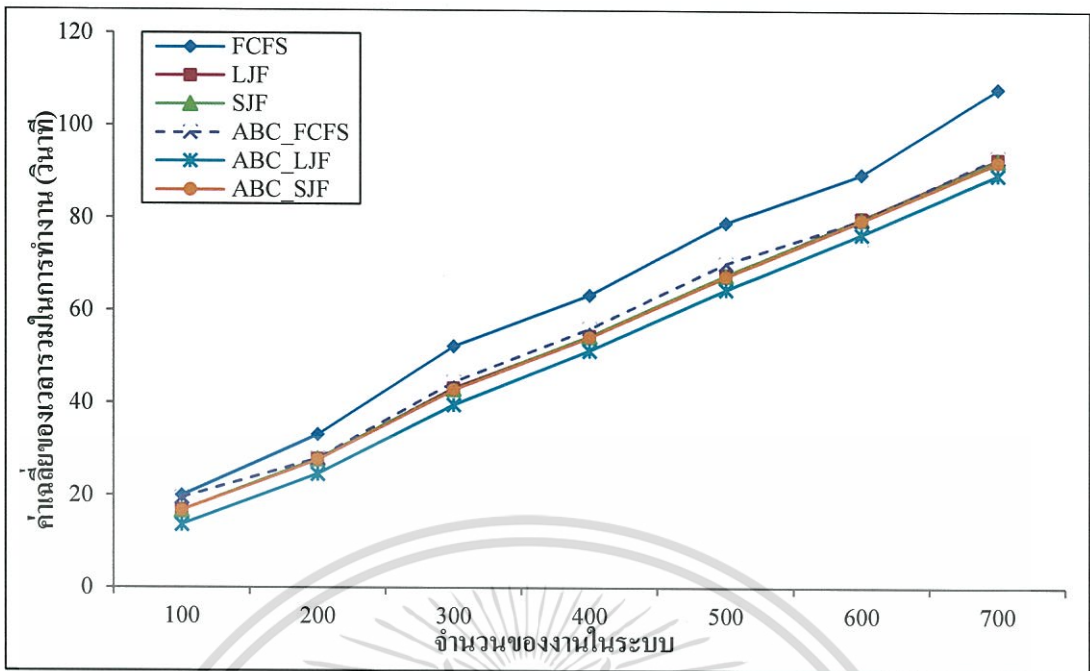
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.14 และรูปที่ 4.15 แสดงเวลาทำงานรวมเฉลี่ย โดยทำการเปรียบเทียบ ขั้นตอนการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกัน โดยใช้ ขั้นตอนวิธีอาณานิคมฟังก์ชันควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า เมื่อขนาดของ RAM มีการเปลี่ยนแปลง กราฟแสดงเวลาทำงานรวมเฉลี่ยจะมีลักษณะคล้ายคลึงกัน ซึ่งสามารถสรุปได้ว่า ABC\_LJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด เมื่อปริมาณงานในระบบมีปริมาณเพิ่มมากขึ้น แม้ว่าจำนวน RAM ในเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง การจัดลำดับงานแบบ ABC\_LJF ยังคงสามารถทำงานได้อย่างมีประสิทธิภาพเช่นเดิม

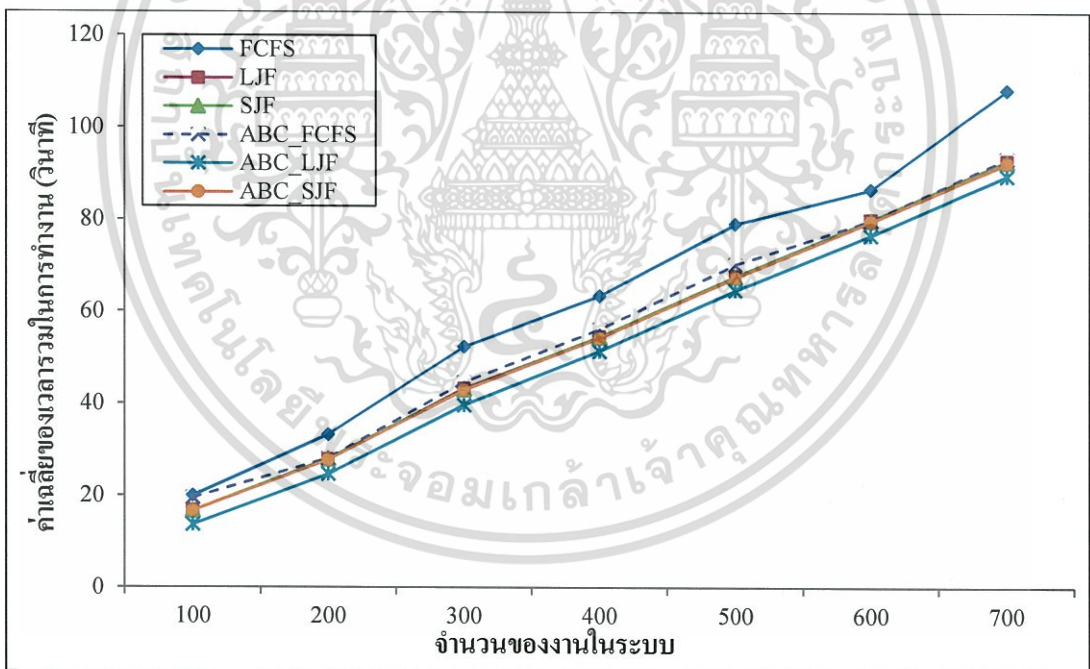
#### 4.4.3.6 ผลการทดสอบประสิทธิภาพ เมื่อขนาดของแบนด์วิธในเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง

การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลง เมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่ 1 เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางการทำงาน โดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันควบคู่กับการจัดลำดับงานแบบฮิวริสติก โดยในการทดสอบจะทำการเปรียบเทียบ เมื่อขนาดแบนด์วิธของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง โดยทำการเปรียบเทียบระหว่างแบนด์วิธเท่ากับ 1000 และ 2000

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลงและขนาดแบนด์วิธของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันทั้งหมด โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.16 และรูปที่ 4.17



รูปที่ 4.16 เวลาทำงานรวมเฉลี่ย เมื่องานเพิ่มขึ้นและกำหนดขนาดแบนด์วิธ = 1000 (Homogenous)



รูปที่ 4.17 เวลาทำงานรวมเฉลี่ย เมื่องานเพิ่มขึ้นและกำหนดค่าแบนด์วิธ = 2000 (Homogenous)

จากรูปที่ 4.16 และรูปที่ 4.17 แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบขั้นตอนการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกัน โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า กราฟแสดงเวลาทำงานรวมเฉลี่ยมีลักษณะ

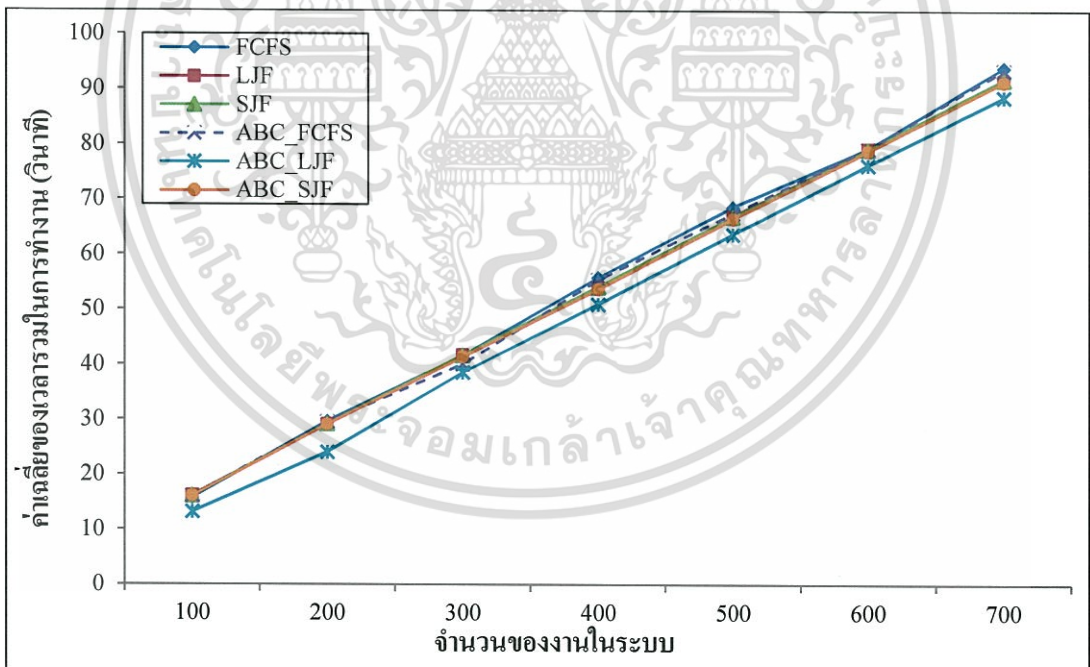
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คล้ายคลึงกันเมื่อขนาดแบนด์วิธมีการเปลี่ยนแปลง ซึ่งสามารถสรุปได้ว่า ABC\_LJF มีประสิทธิภาพในการจัดการงานดีที่สุดในที่สุด เมื่อปริมาณงานในระบบมีปริมาณเพิ่มมากขึ้น แม้ว่าขนาดแบนด์วิธภายในเครื่องคอมพิวเตอร์เสมือนจะมีการเปลี่ยนแปลง การจัดลำดับงานแบบ ABC\_LJF ยังคงสามารถทำงานได้อย่างมีประสิทธิภาพเช่นเดิม

#### 4.4.3.7 ผลการทดสอบประสิทธิภาพ เมื่อข้อมูลมีการกระจายแบบโค้งปกติ

การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลง เมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่ 8 ซึ่งมีรูปแบบการกระจายของข้อมูลแบบโค้งปกติ เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดการงาน โดยใช้ขั้นตอนวิธีอาณานิคมซึ่งเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลงและข้อมูลมีการกระจายแบบโค้งปกติ กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันทั้งหมด โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.18



รูปที่ 4.18 เวลาทำงานรวมเฉลี่ย เมื่อข้อมูลมีการกระจายแบบ โค้งปกติ (Homogenous)

ตารางที่ 4.7 เปอร์เซ็นต์ค่าความต่างของเวลาที่ลดลงในการจัดตารางงานแบบฮิวริสติกของข้อมูล ชุดที่ 8 (Homogenous)

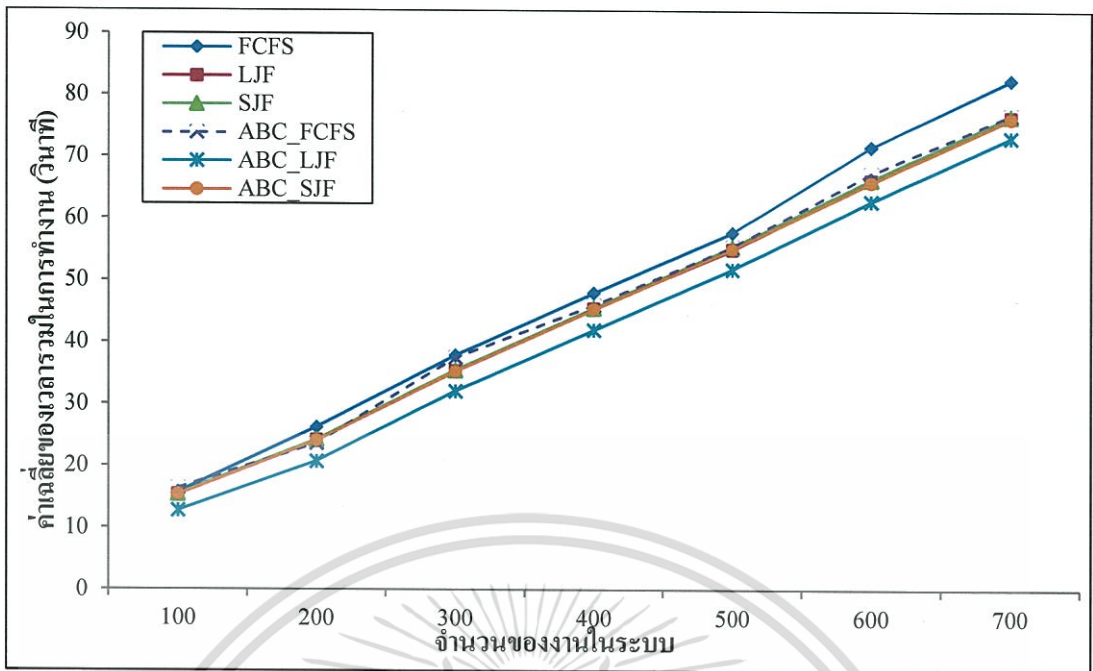
รูปแบบการ จัดตารางงาน	จำนวนงานในระบบ						
	100	200	300	400	500	600	700
ABC_FCFS	-2.10	0.34	3.68	0.66	1.68	0.02	0.51
ABC_LJF	18.57	17.44	7.58	5.67	4.81	3.70	3.24
ABC_SJF	0.37	0.34	0.74	0.73	0.53	0.43	0.48

จากรูปที่ 4.18 แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบขั้นตอนการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกัน โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ABC\_LJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด เมื่อข้อมูลมีรูปแบบการกระจายแบบโค้งปกติ และจากตารางที่ 4.7 พบว่าเมื่อทำการเปรียบเทียบข้อมูลจากเปอร์เซ็นต์ค่าความต่างของเวลาที่ลดลงในการจัดตารางงานแบบฮิวริสติกและขั้นตอนวิธีอาณานิคมผึ้งเทียบ ABC\_LJF สามารถลดเวลารวมในการจัดตารางเข้าทำงานได้ดีกว่าวิธีการอื่นๆ

#### 4.4.3.8 ผลการทดสอบประสิทธิภาพ เมื่อข้อมูลมีการกระจายแบบเบ้ขวา

การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลงเมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่ 9 ที่มีรูปแบบการกระจายของข้อมูลแบบเบ้ขวา เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางงาน โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลงและข้อมูลมีการกระจายแบบเบ้ขวา กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันทั้งหมด โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.19



รูปที่ 4.19 เวลาทำงานรวมเฉลี่ย เมื่อข้อมูลมีการกระจายแบบเบ้ขวา (Homogenous)

ตารางที่ 4.8 เปอร์เซ็นต์ค่าความต่างของเวลาที่ลดลงในการจัดการงานแบบฮิวริสติกของข้อมูลชุดที่ 9 (Homogenous)

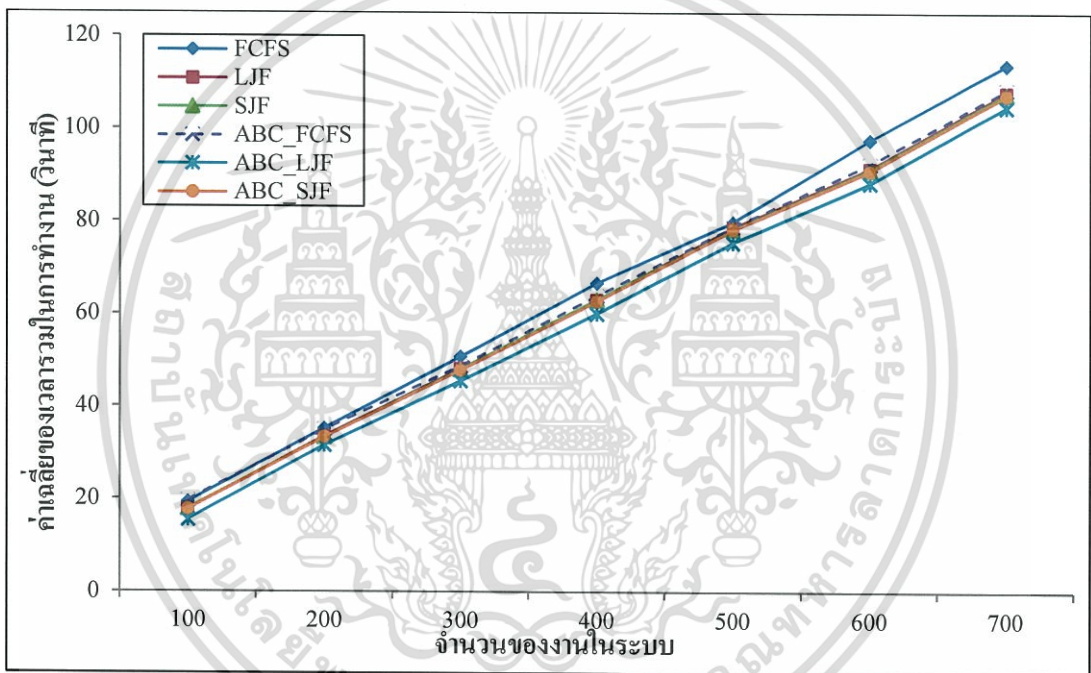
รูปแบบการ จัดการงาน	จำนวนงานในระบบ						
	100	200	300	400	500	600	700
ABC_FCFS	-4.32	9.76	0.84	4.12	3.89	5.97	6.78
ABC_LJF	16.80	14.15	9.04	7.50	5.72	5.01	4.24
ABC_SJF	0.58	0.62	0.59	0.46	0.68	0.71	0.61

จากรูปที่ 4.19 แสดงเวลาทำงานรวมเฉลี่ย โดยทำการเปรียบเทียบขั้นตอนการจัดการงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกัน โดยใช้ขั้นตอนวิธีอาณานิคมฝึ้มเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ABC\_LJF มีประสิทธิภาพในการจัดการงานดีที่สุด เมื่อข้อมูลมีรูปแบบการกระจายแบบเบ้ขวา และจากตารางที่ 4.8 พบว่าเมื่อทำการเปรียบเทียบข้อมูลจากเปอร์เซ็นต์ค่าความต่างของเวลาที่ลดลงในการจัดการงานแบบฮิวริสติกและขั้นตอนวิธีอาณานิคมฝึ้มเทียบ ABC\_LJF สามารถลดเวลารวมในการจัดการงานเข้าทำงานได้ดีกว่าวิธีการอื่นๆ

#### 4.4.3.9 ผลการทดสอบประสิทธิภาพ เมื่อข้อมูลมีการกระจายแบบเบ้ซ้าย

การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลงเมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่ 10 ที่มีรูปแบบการกระจายของข้อมูลเป็นแบบเบ้ซ้าย เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางงาน โดยใช้ขั้นตอนวิธีอาณานิคมซึ่งเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลงและข้อมูลมีการกระจายแบบเบ้ซ้าย กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันทั้งหมด โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.20



รูปที่ 4.20 เวลาทำงานรวมเฉลี่ย เมื่อข้อมูลมีการกระจายแบบเบ้ซ้าย (Homogenous)

ตารางที่ 4.9 เปอร์เซ็นต์ค่าความต่างของเวลาที่ลดลงในการจัดตารางงานแบบฮิวริสติกของข้อมูลชุดที่ 10 (Homogenous)

รูปแบบการจัดตารางงาน	จำนวนงานในระบบ						
	100	200	300	400	500	600	700
ABC_FCFS	-1.61	0.51	4.06	4.14	1.31	5.33	4.67
ABC_LJF	12.91	5.85	5.51	4.53	4.05	3.36	2.83
ABC_SJF	0.62	0.11	0.45	0.57	0.20	0.39	0.33

จากรูปที่ 4.20 แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบขั้นตอนการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกัน โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ABC\_LJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด เมื่อข้อมูลมีรูปแบบการกระจายแบบเบ้ซ้าย และจากตารางที่ 4.9 พบว่าเมื่อทำการเปรียบเทียบข้อมูลจากเปอร์เซ็นต์ค่าความต่างของเวลาที่ลดลงในการจัดตารางงานแบบฮิวริสติกและขั้นตอนวิธีอาณานิคมผึ้งเทียม ABC\_LJF สามารถลดเวลารวมในการจัดตารางเข้าทำงานได้ดีกว่าวิธีการอื่นๆ

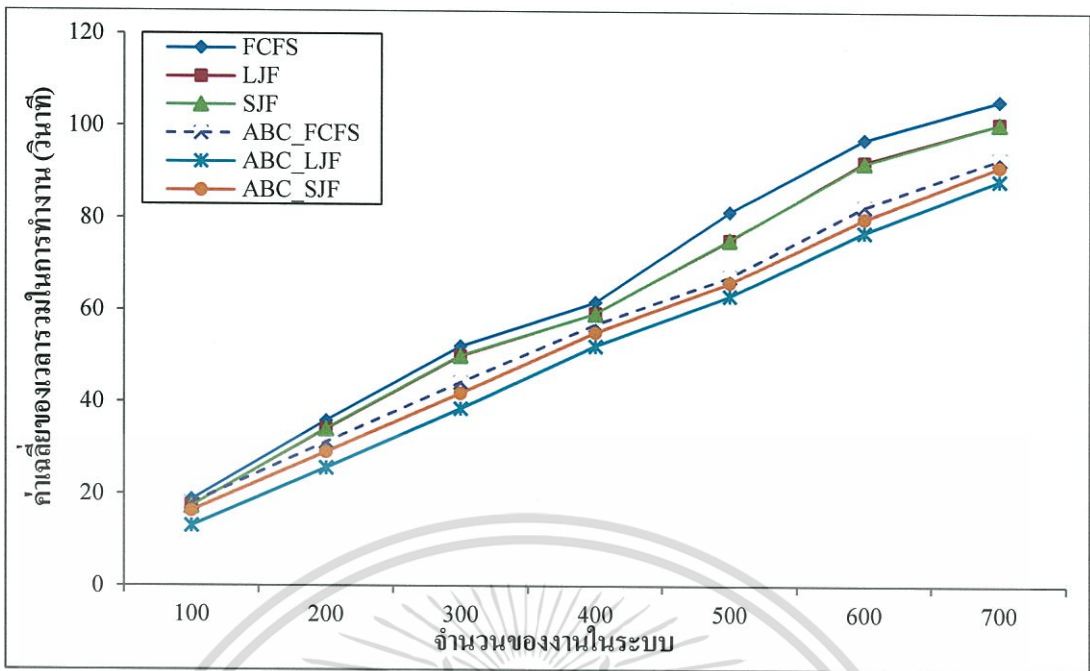
#### 4.4.3.10 ผลการทดสอบประสิทธิภาพ เมื่อจำนวนผึ้งในระบบมีการเปลี่ยนแปลง

การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลงเมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่ 2 เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางงาน โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก โดยในการทดสอบจะทำการเปรียบเทียบค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมผึ้งเทียมที่มีการเปลี่ยนแปลง แสดงค่าดังตารางที่ 4.10

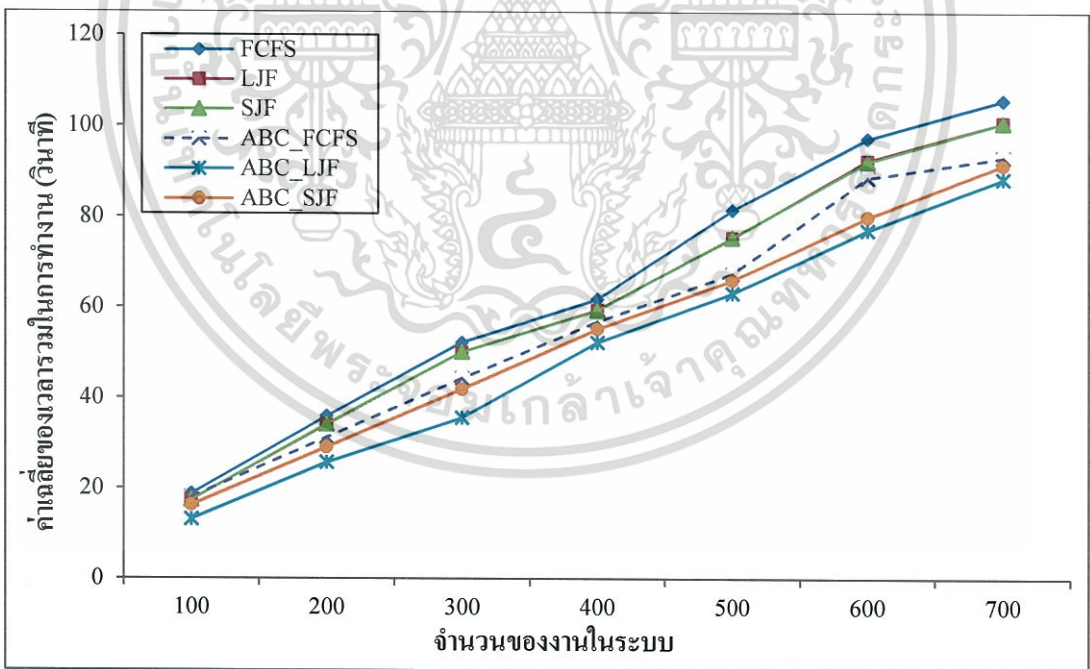
ตารางที่ 4.10 ค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมผึ้งเทียม (Homogenous)

Symbol	Parameter	Value I	Value II
n	Number of scout bees	2000	1500
m	Number of sites selected out of n visited sites	5	10
e	Number of best site out of m selected sites	1	1
nep	Number of bees recruited for best e sites	1600	1000
nsp	Number of bees recruited for other (m-e) selected sites	400	500

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลงและจำนวนผึ้งในระบบมีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันทั้งหมด โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.21 และรูปที่ 4.22



รูปที่ 4.21 เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและกำหนดค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมฟังก์ชัน Value I (Homogenous)



รูปที่ 4.22 เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและกำหนดค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมฟังก์ชัน Value II (Homogenous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.21 และรูปที่ 4.22 แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบ ขั้นตอนการจัดการงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกัน โดยใช้ ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ABC\_LJF มีประสิทธิภาพในการจัดการงานได้ดีที่สุด เมื่อปริมาณงานในระบบมีปริมาณเพิ่มมากขึ้น แม้ว่าค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมฟังก์ชันจะมีการเปลี่ยนแปลง การจัดลำดับงานแบบ ABC\_LJF ยังคงสามารถทำงานได้อย่างมีประสิทธิภาพเช่นเดิม

#### 4.4.4 ผลการทดสอบประสิทธิภาพกรณีทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง

การประเมินประสิทธิภาพในการจัดการงานในเครื่องคอมพิวเตอร์เสมือน โดยใช้ ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก (ABC\_FCFS, ABC\_SJF และ ABC\_LJF) จะทำการเปรียบเทียบกับการจัดลำดับงานแบบฮิวริสติก เพียงอย่างเดียว (FCFS, SJF และ LJF) การตั้งค่าสภาพแวดล้อมของเครื่องคอมพิวเตอร์เสมือน กรณีสภาพแวดล้อมภายในเหมือนกัน แสดงค่าดังตารางที่ 4.11 โดยในการทดสอบจะกำหนดจำนวนของงานเท่ากับ 700 งาน และแบ่งการทดสอบออกเป็น 2 กรณี

ตารางที่ 4.11 สภาพแวดล้อมของเครื่องคอมพิวเตอร์เสมือนกรณีสภาพแวดล้อมภายในเหมือนกัน

Type	Parameter	Value
Virtual Machine (VM)	Total number of VM	30 - 180
	MIPS of PE	2000
	Number of PE per VM	1
	VM Memory (RAM) (MB)	1024
	Bandwidth	1000 Bit
	Type of Manager	Time_shared

**หมายเหตุ:** PE = Processing Element, MIPS = Million Instructions Per Second,

MB = Megabyte

#### 4.4.4.1 ผลการทดสอบประสิทธิภาพเมื่อชุดข้อมูลมีการเปลี่ยนแปลง

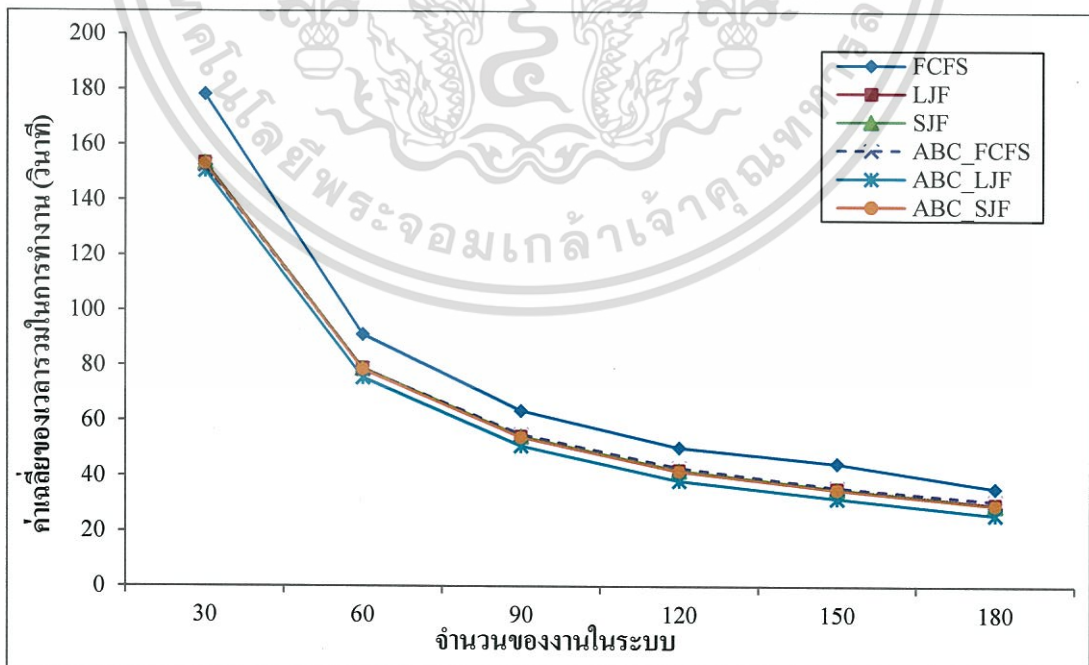
การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง และจำนวนงานมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่แสดงค่าดังตารางที่ 4.12 เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางการทำงานโดยใช้ขั้นตอนวิธีอาณานิคมพึงเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก

ตารางที่ 4.12 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของการทดสอบกรณีที่ 4.4.4.1

ชุดข้อมูล	ขนาดของงาน (MI)	การกระจายข้อมูล
1	5000 - 20000	กลุ่มอิสระ
2	5000 - 20000	กลุ่มอิสระ
3	5000 - 20000	กลุ่มอิสระ

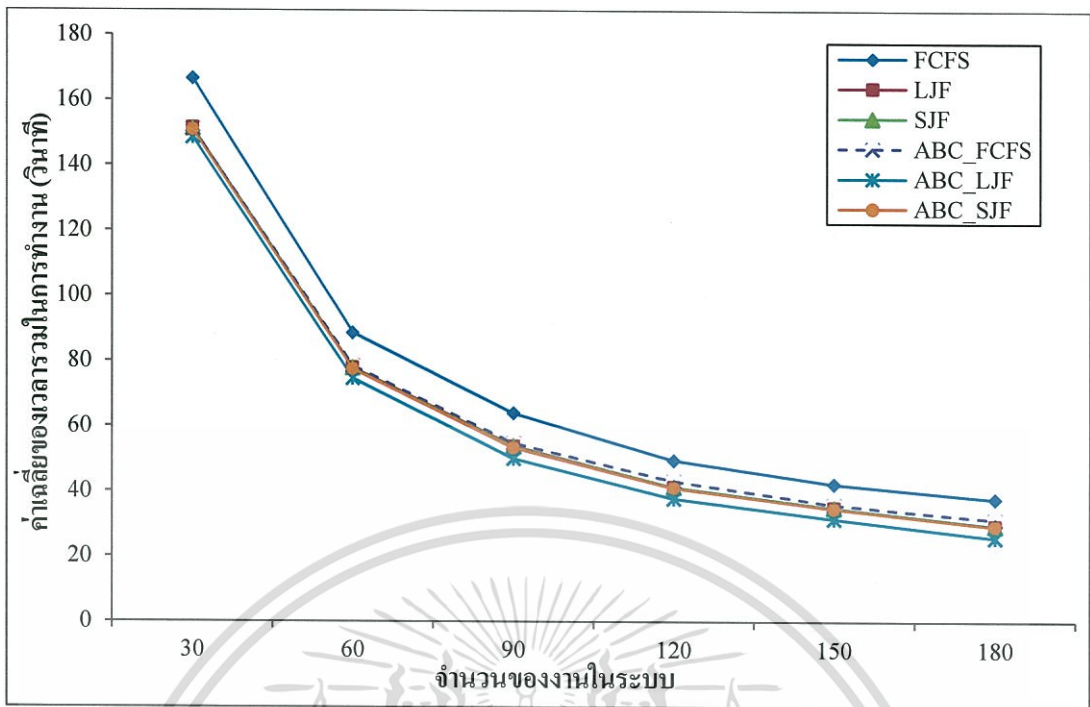
หมายเหตุ: MI = Million Instruction

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนเครื่องคอมพิวเตอร์เสมือนและชุดข้อมูลที่ใช้ในการทดสอบมีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันทั้งหมด โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.23 – 4.25

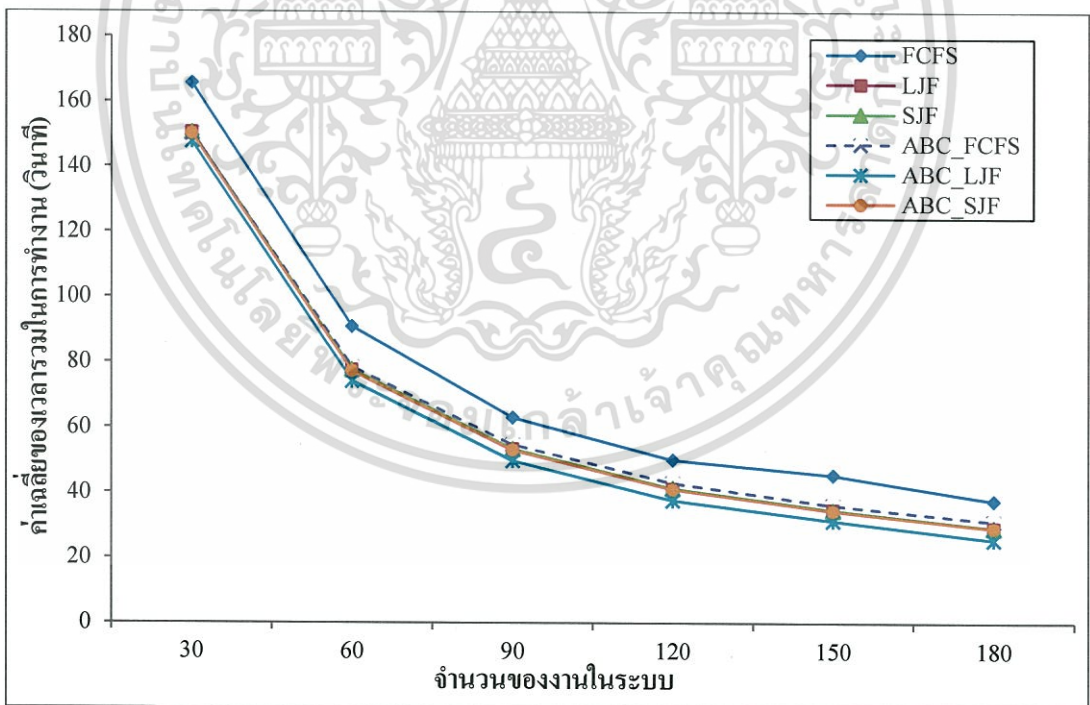


รูปที่ 4.23 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 เมื่อจำนวน VM เพิ่มขึ้น (Homogenous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 2 เมื่อจำนวน VM เพิ่มขึ้น (Homogenous)



รูปที่ 4.25 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 3 เมื่อจำนวน VM เพิ่มขึ้น (Homogenous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.23 - 4.25 แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบขั้นตอนการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกัน โดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันที่เชื่อมควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า รูปกราฟที่ได้ในข้อมูลแต่ละชุดมีลักษณะคล้ายคลึงกัน โดย ABC\_LJF มีประสิทธิภาพในการจัดตารางงานที่ดีที่สุด เมื่อปริมาณเครื่องคอมพิวเตอร์เสมือนในระบบมีปริมาณเพิ่มมากขึ้น แม้ว่าข้อมูลภายในจะมีการเปลี่ยนแปลง การจัดลำดับงานแบบ ABC\_LJF ยังคงสามารถทำงานได้อย่างมีประสิทธิภาพเช่นเดิม

#### 4.4.4.2 ผลการทดสอบประสิทธิภาพเมื่อขนาดของงานมีการเปลี่ยนแปลง

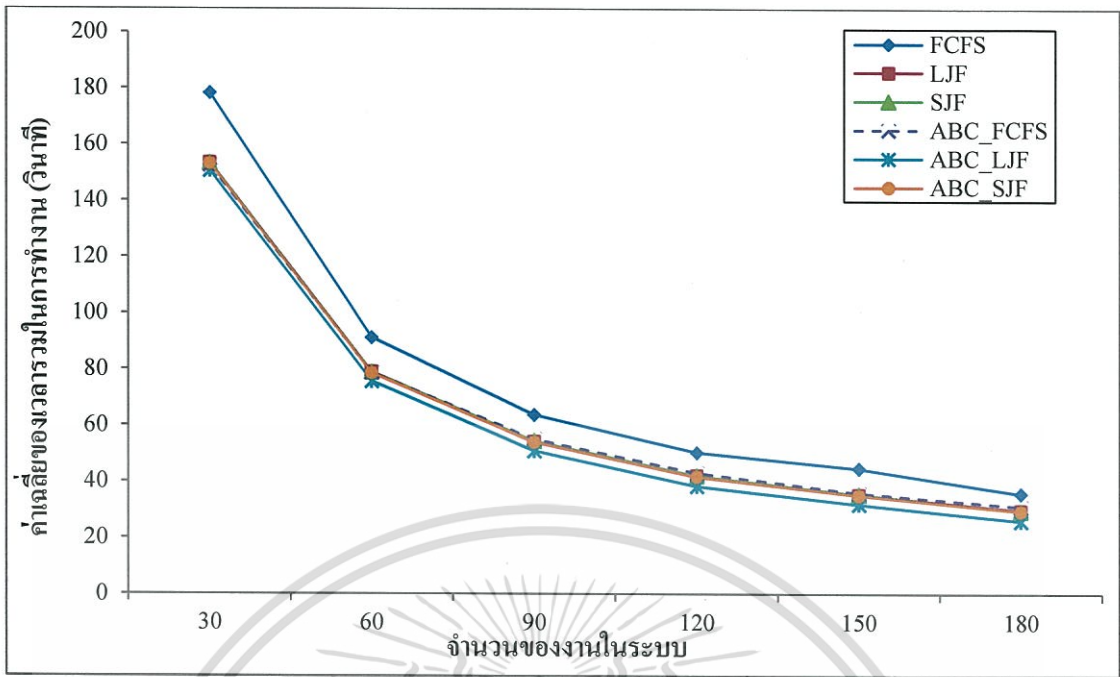
การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง และจำนวนของงานมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลและขนาดของงาน ดังตารางที่ 4.13 เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางการทำงาน โดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันที่เชื่อมควบคู่กับการจัดลำดับงานแบบฮิวริสติก

ตารางที่ 4.13 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของการทดสอบกรณีที่ 4.4.4.2

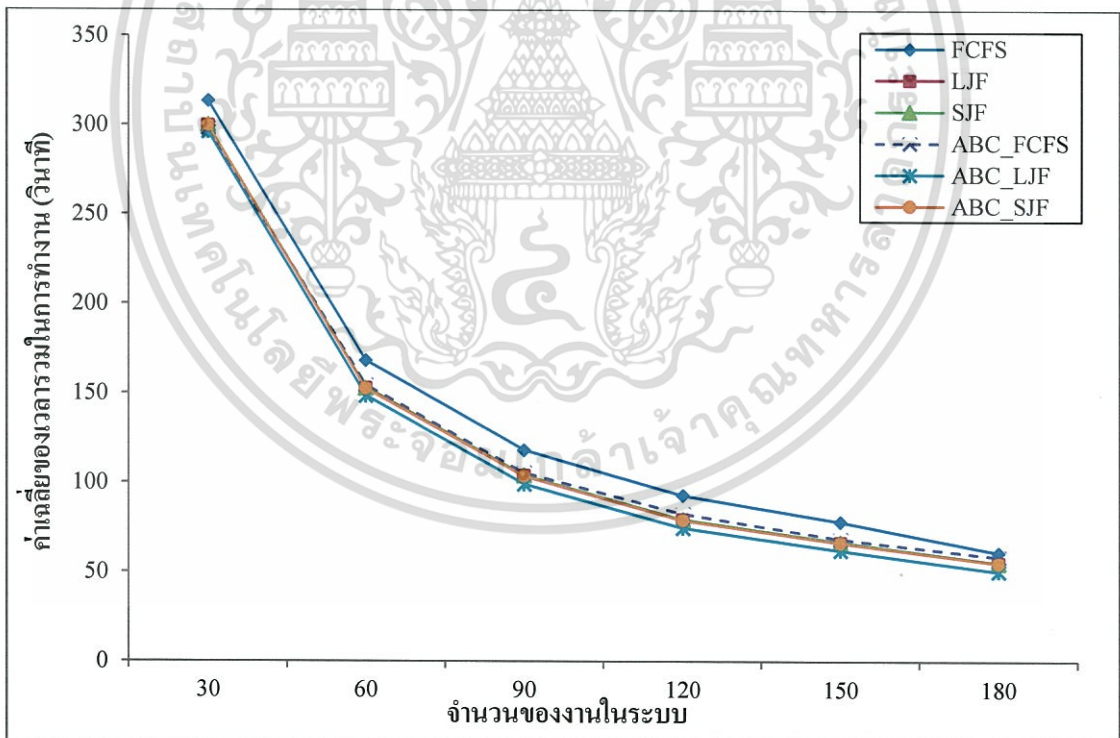
ชุดข้อมูล	ขนาดของงาน (MI)	การกระจายข้อมูล
1	5000 - 20000	สุ่มอิสระ
4	15000 - 35000	สุ่มอิสระ
5	25000 - 45000	สุ่มอิสระ
6	35000 - 55000	สุ่มอิสระ
7	45000 - 65000	สุ่มอิสระ

หมายเหตุ: MI = Million Instruction

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลงและชุดข้อมูลที่ใช้ในการทดสอบมีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันทั้งหมด โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.26 – 4.30

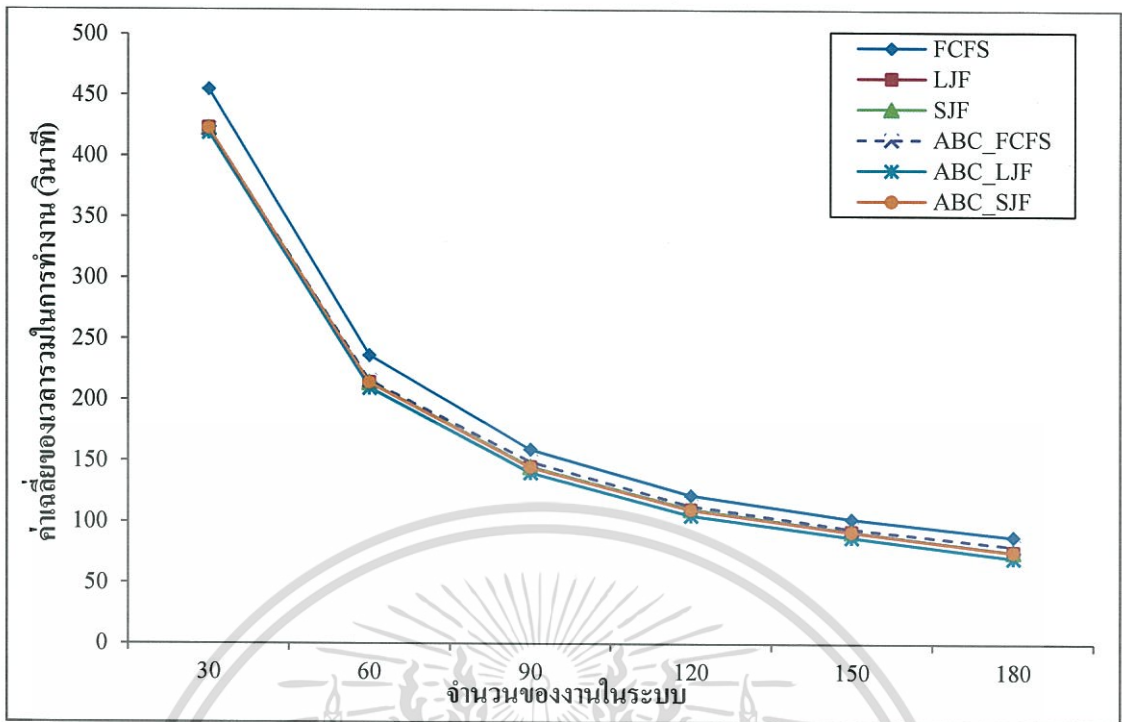


รูปที่ 4.26 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 ขนาดของงาน 5000 - 20000 (Homogenous)

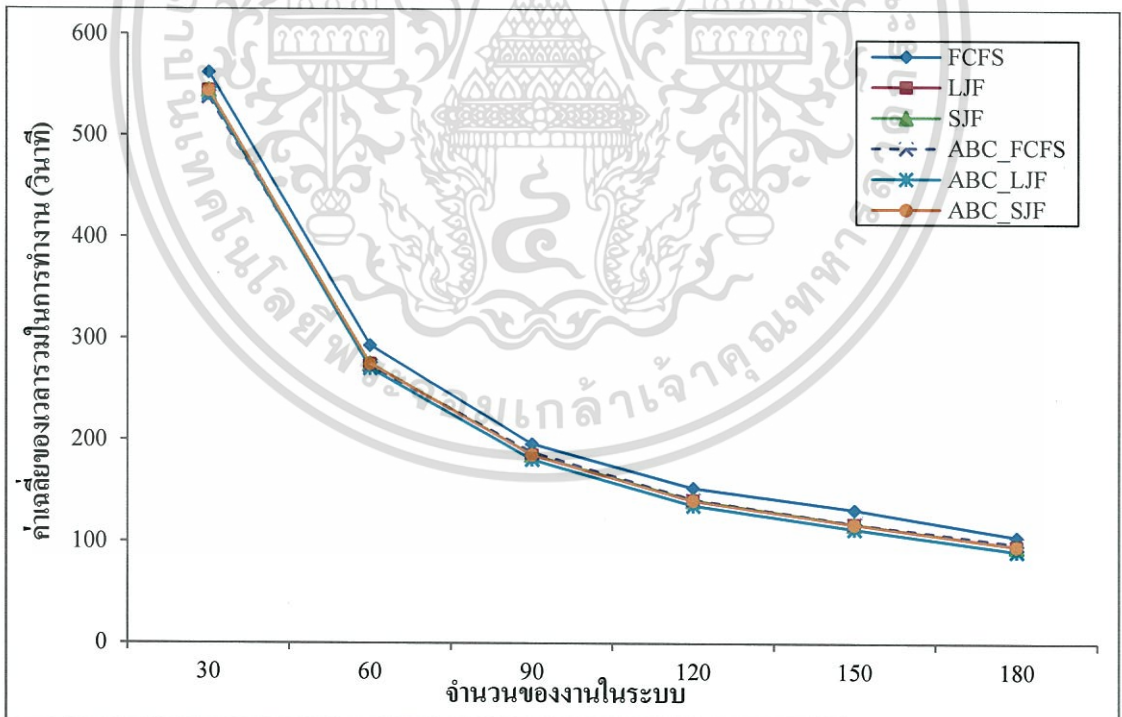


รูปที่ 4.27 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 4 ขนาดของงาน 15000 - 35000 (Homogenous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

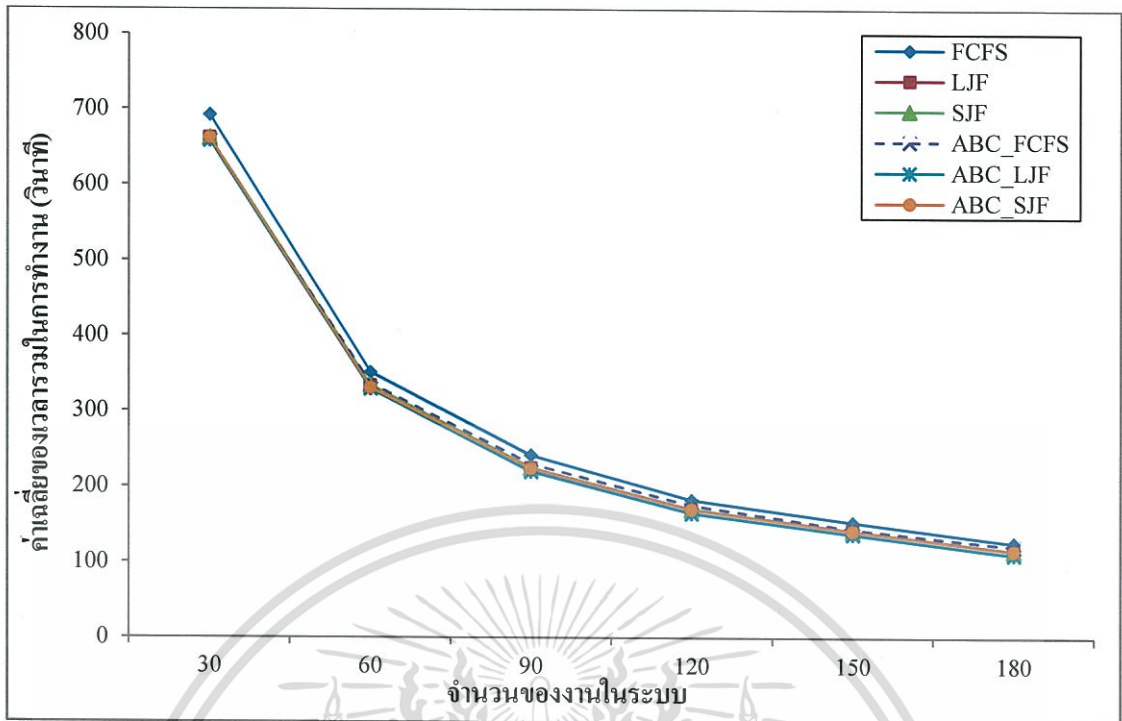


รูปที่ 4.28 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 5 ขนาดของงาน 25000 - 45000 (Homogenous)



รูปที่ 4.29 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 6 ขนาดของงาน 35000 - 55000 (Homogenous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 7 ขนาดของงาน 45000 - 65000 (Homogenous)

จากรูปที่ 4.26 - 4.30 แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบขั้นตอนการจ้ดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกัน โดยใช้ขั้นตอนวิธีอาณานิคมซึ่งเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ถ้างานมีขนาดระหว่าง 5000 - 65000 การจ้ดตารางงานแบบ ABC\_LJF มีแนวโน้มในการจ้ดตารางงานดีที่สุดใน แต่ถ้างานมีขนาดมากกว่า 65000 ขั้นตอนการจ้ดตารางงานแบบ ABC\_LJF อาจมีประสิทธิภาพในการจ้ดตารางงานไม่ดีไปกว่าขั้นตอนวิธีการจ้ดตารางงานแบบอื่นๆ เมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนเพิ่มมากขึ้น

## 4.5 ผลการทดลองกรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในต่างกัน

### 4.5.1 การตั้งค่าพารามิเตอร์สภาพแวดล้อมในการทดลอง

การตั้งค่าพารามิเตอร์แบบจำลองเพื่อประเมินประสิทธิภาพของระบบการประมวลผลแบบกลุ่มเมฆ กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในต่างกัน แสดงดังตารางที่ 4.14

ตารางที่ 4.14 ค่าพารามิเตอร์ของแบบจำลองการประมวลผลแบบกลุ่มเมฆ กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในต่างกัน

Type	Parameter	Value
Datacenter	Number of Datacenter	10
	Number of Host	5
	Type of Manager	Space_Shared, Time_Shared
	Number of PE per Host	4
	MIPS of PE	2000
	Host Memory (RAM) (MB)	10240
	Datacenter Cost	10
Virtual Machine (VM)	Total number of VM	20 - 120
	MIPS of PE	1000 - 2000
	Number of PE per VM	1 - 6
	VM Memory (RAM) (MB)	512 - 2048
	Bandwidth	1000 - 2000 Bit
	Type of Manager	Time_Shared
Task	Total number of Task	100 - 700
	Length of Task (MI)	5000 - 20000
	Number of PE per requirement	1
	Type of Manager	Space_Shared

**หมายเหตุ:** PE = Processing Element, MIPS = Million Instructions Per Second,

MB = Megabyte, MI = Million Instruction

#### 4.5.2 การตั้งค่าพารามิเตอร์ในขั้นตอนวิธีอาณานิคมผึ้งเทียม

การตั้งค่าพารามิเตอร์ในส่วนของขั้นตอนวิธีอาณานิคมผึ้งเทียม แสดงค่าดังตารางที่ 4.15

ตารางที่ 4.15 ค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมผึ้งเทียม (Heterogeneous)

Symbol	Parameter	Value
n	Number of scout bees	2000
m	Number of sites selected out of n visited sites	5
e	Number of best site out of m selected sites	1
nep	Number of bees recruited for best e sites	1600
nsp	Number of bees recruited for other (m-e) selected sites	400

#### 4.5.3 ผลการทดลอง กรณีทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลง

การประเมินประสิทธิภาพในการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือน โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก (ABC\_FCFS, ABC\_SJF และ ABC\_LJF) จะทำการเปรียบเทียบกับการจัดลำดับงานแบบฮิวริสติก เพียงอย่างเดียว (FCFS, SJF และ LJF) กำหนดจำนวนของเครื่องคอมพิวเตอร์เสมือนจำนวน 50 เครื่อง โดยแบ่งการทดสอบออกเป็น 6 กรณี

##### 4.5.3.1 ผลการทดสอบประสิทธิภาพเมื่อชุดข้อมูลมีการเปลี่ยนแปลง

การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลงเมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่แสดงค่าดังตารางที่ 4.16 เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางการทำงาน โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก

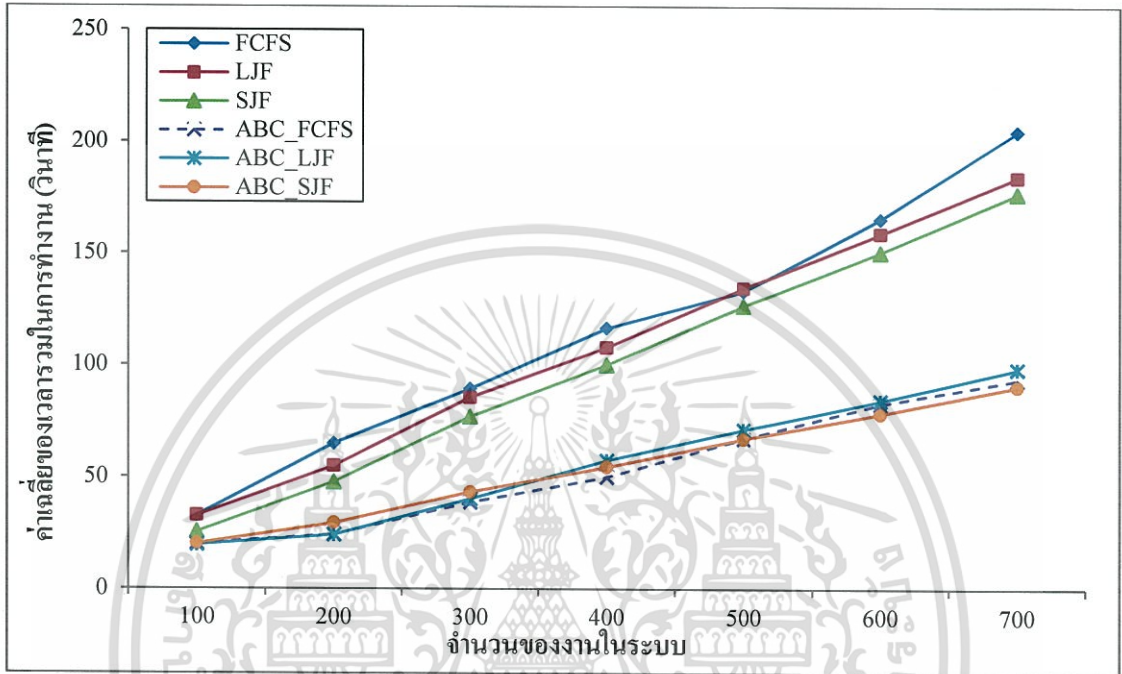
ตารางที่ 4.16 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของการทดสอบกรณีที่ 4.5.3.1

ชุดข้อมูล	ขนาดของงาน (MI)	การกระจายข้อมูล
1	5000 - 20000	สุ่มอิสระ
2	5000 - 20000	สุ่มอิสระ
3	5000 - 20000	สุ่มอิสระ

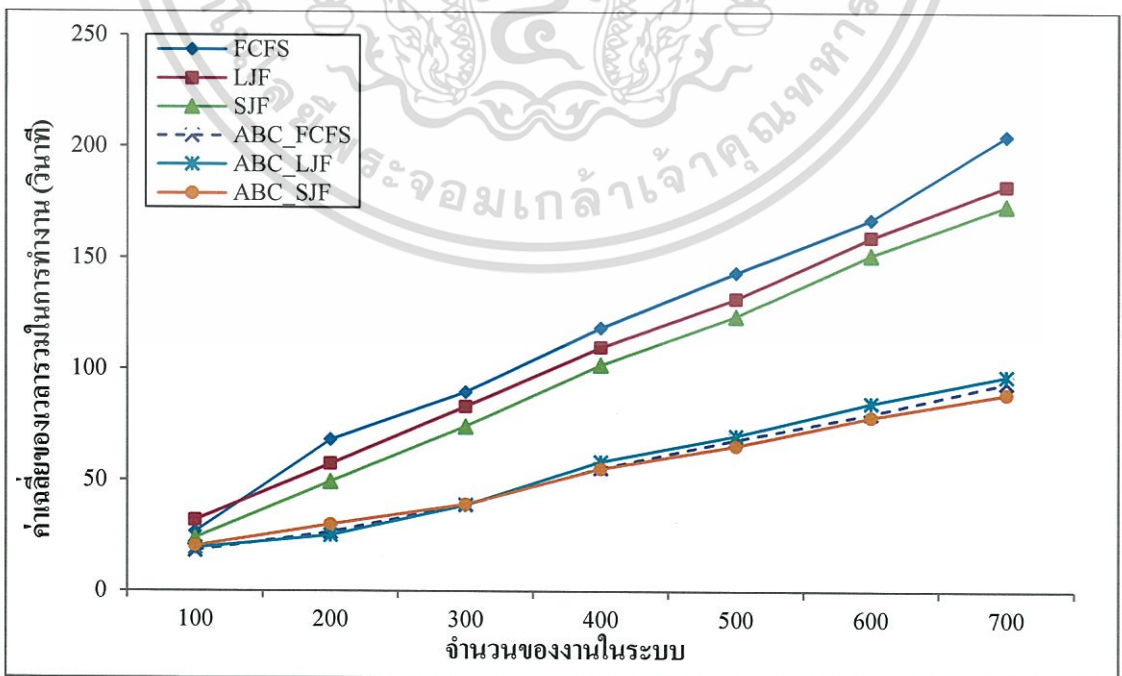
หมายเหตุ: MI = Million Instruction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

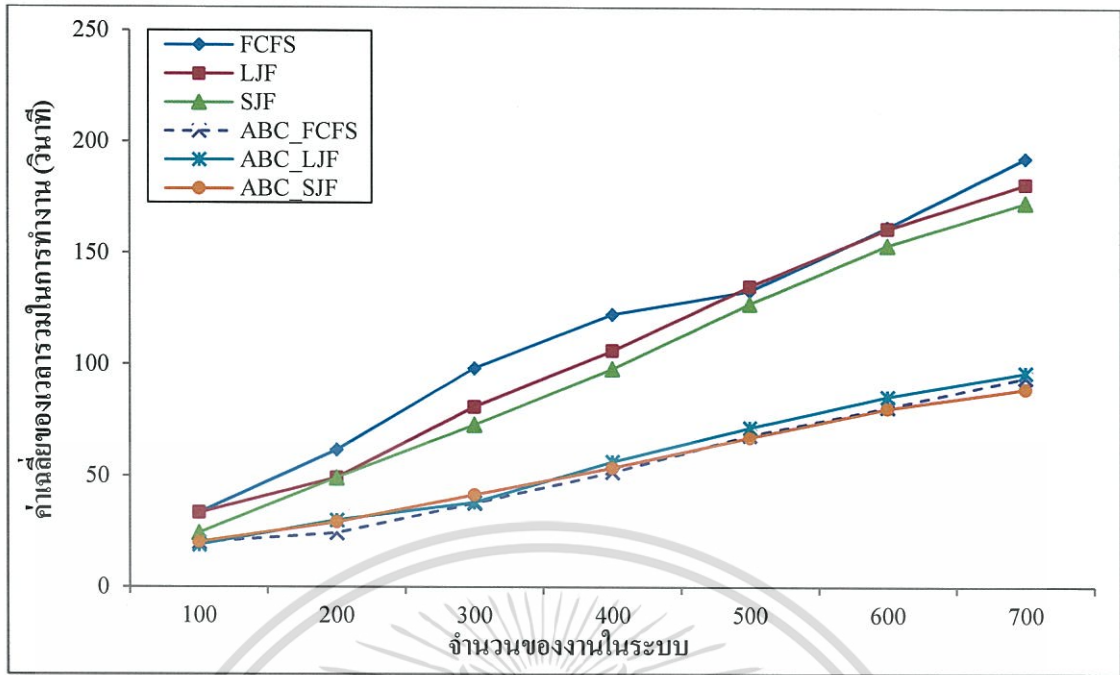
ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลงและชุดข้อมูลที่ใช้ในการทดสอบมีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในต่างกัน โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.31 -4.33



รูปที่ 4.31 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 เมื่องานมีปริมาณเพิ่มขึ้น (Heterogeneous)



เอกสารนี้เป็นรูปที่ 4.32 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 2 เมื่องานมีปริมาณเพิ่มขึ้น (Heterogeneous) การค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.33 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 3 เมื่องานมีปริมาณเพิ่มขึ้น (Heterogeneous)

จากรูปที่ 4.31 – 4.33 แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบขั้นตอนการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในต่างกัน โดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ABC\_SJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด เมื่อปริมาณงานในระบบมีปริมาณเพิ่มมากขึ้นและงานในระบบมีปริมาณมากกว่า 500 งานขึ้นไป แม้ว่าข้อมูลภายในจะมีการเปลี่ยนแปลง การจัดลำดับงานแบบ ABC\_SJF ยังคงสามารถทำงานได้อย่างมีประสิทธิภาพ เมื่องานในระบบมีปริมาณมากกว่า 500 งานขึ้นไป

#### 4.5.3.2 ผลการทดสอบประสิทธิภาพ เมื่อขนาดของงานมีการเปลี่ยนแปลง

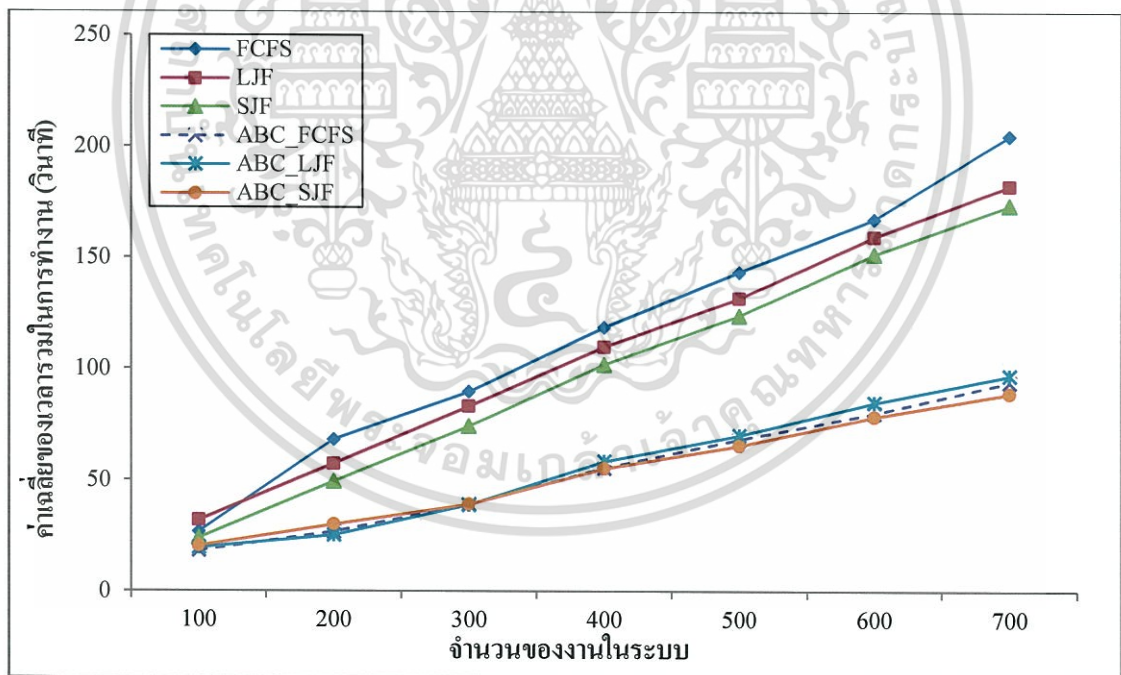
การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลงเมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลและขนาดของงาน ดังตารางที่ 4.17 เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดการการทำงานโดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก

ตารางที่ 4.17 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของการทดสอบกรณีที่ 4.5.3.2

ชุดข้อมูล	ขนาดของงาน (MI)	การกระจายข้อมูล
2	5000 - 20000	สุ่มอิสระ
4	15000 - 35000	สุ่มอิสระ
5	25000 - 45000	สุ่มอิสระ
6	35000 - 55000	สุ่มอิสระ
7	45000 - 65000	สุ่มอิสระ

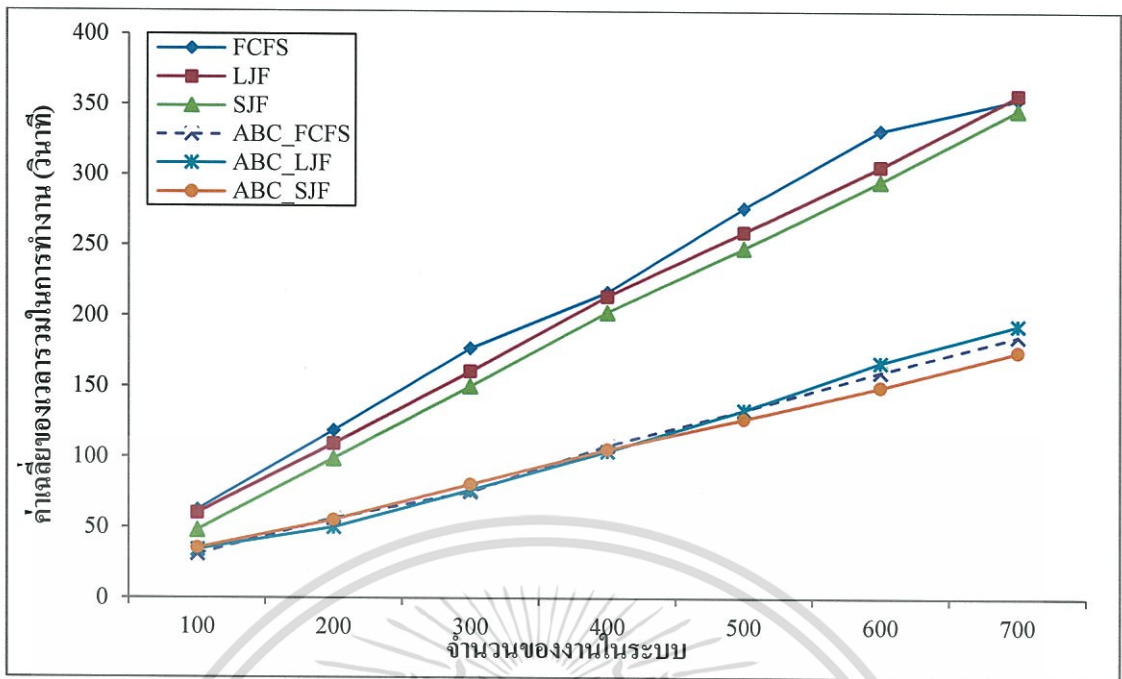
หมายเหตุ: MI = Million Instruction

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลงและขนาดของงานมีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในต่างกัน โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.34 – 4.38

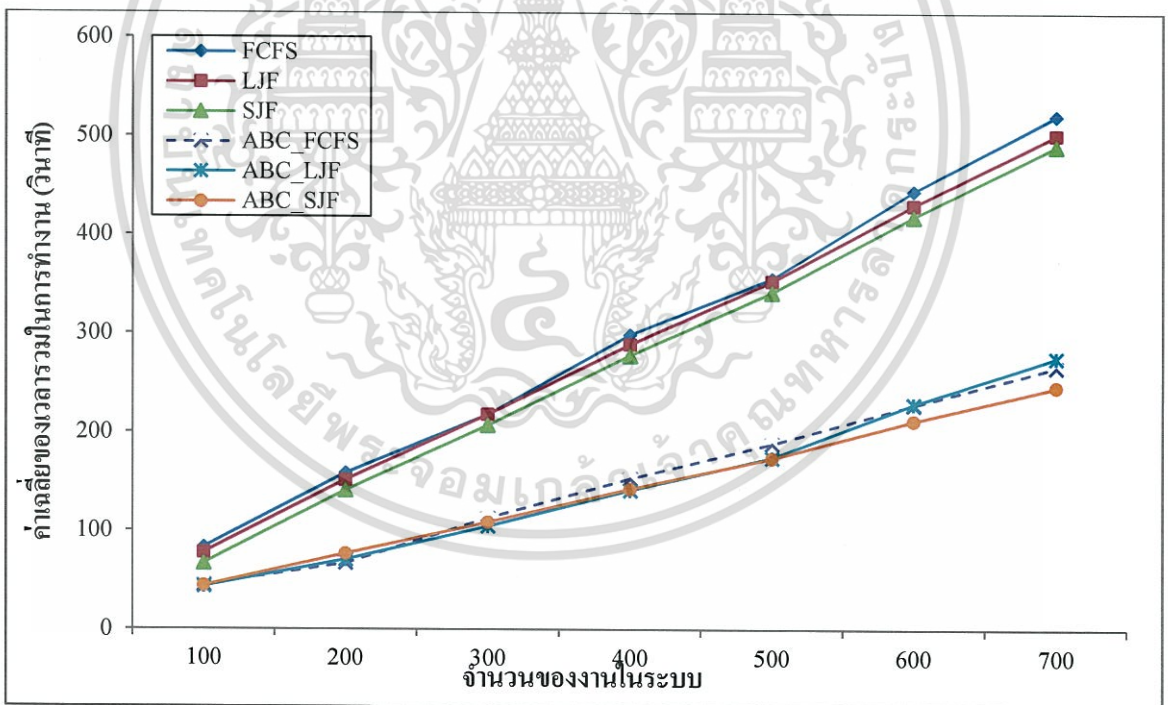


รูปที่ 4.34 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 2 ขนาดของงาน 5000 – 20000 (Heterogeneous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

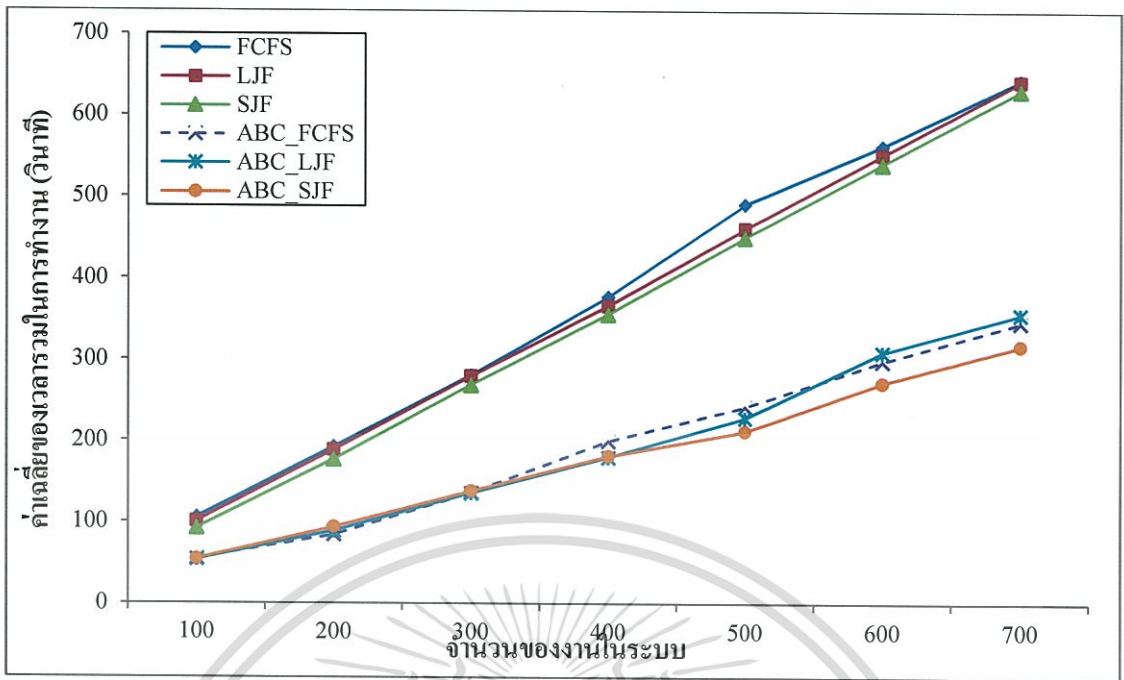


รูปที่ 4.35 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 4 ขนาดของงาน 15000 – 35000 (Heterogeneous)

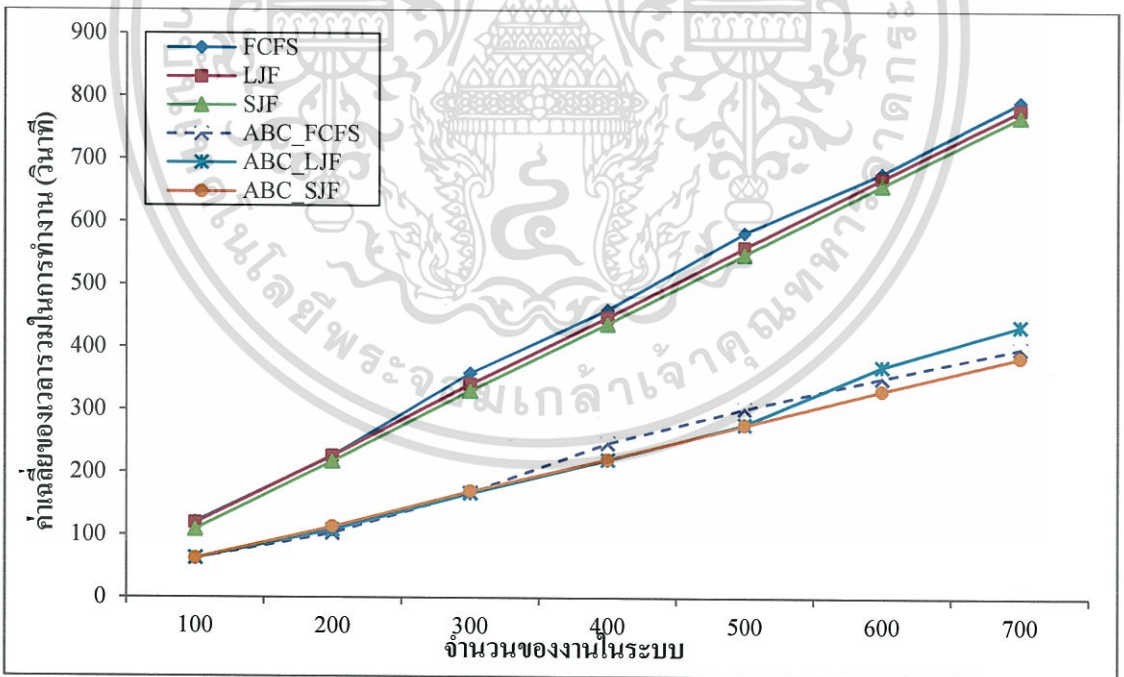


รูปที่ 4.36 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 5 ขนาดของงาน 25000 – 45000 (Heterogeneous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.37 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 6 ขนาดของงาน 35000 – 55000 (Heterogeneous)



รูปที่ 4.38 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 7 ขนาดของงาน 45000 – 65000 (Heterogeneous)

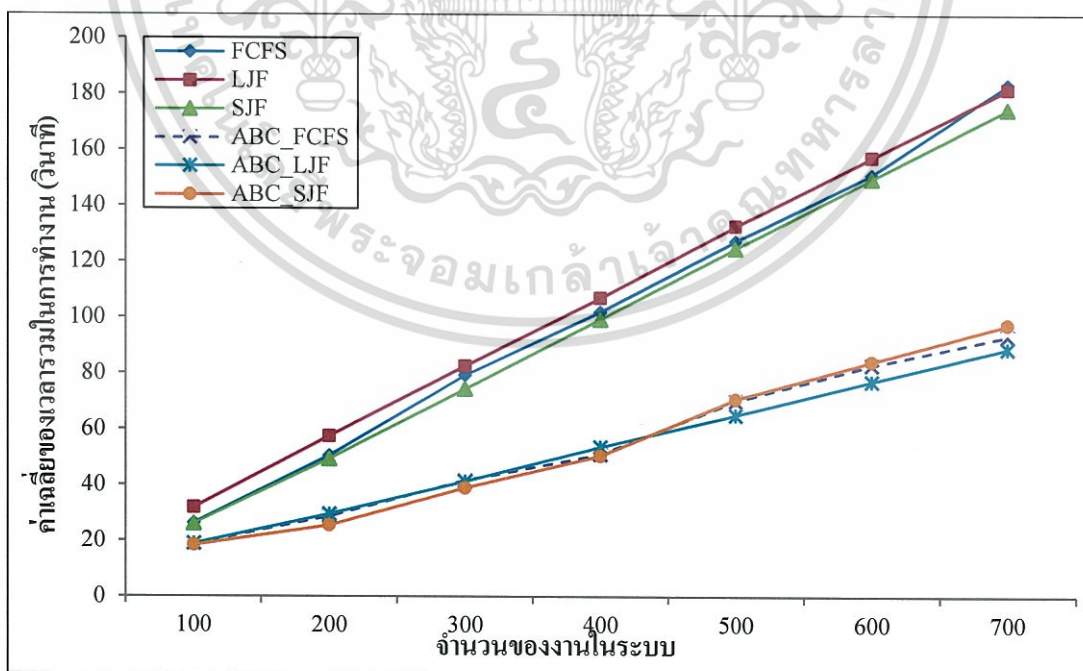
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.34 – 4.38 แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบ ขั้นตอนการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในต่างกัน โดยใช้ ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ABC\_SJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด เมื่อปริมาณงานในระบบมีปริมาณเพิ่มมากขึ้นและงานในระบบมีปริมาณมากกว่า 300 งานขึ้นไป แม้ว่าขนาดข้อมูลภายในจะมีการเปลี่ยนแปลง การจัดลำดับงานแบบ ABC\_SJF ยังคงสามารถทำงานได้อย่างมีประสิทธิภาพเช่นเดิม

#### 4.5.3.3 ผลการทดสอบประสิทธิภาพ เมื่อข้อมูลมีการกระจายแบบโค้งปกติ

การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลง เมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่ 8 ที่มีรูปแบบการกระจายของข้อมูลแบบโค้งปกติ เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางงาน โดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลงและข้อมูลมีการกระจายแบบโค้งปกติ กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในต่างกัน โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.39



รูปที่ 4.39 เวลาทำงานรวมเฉลี่ยของข้อมูล เมื่อข้อมูลมีการกระจายแบบ โค้งปกติ (Heterogeneous)

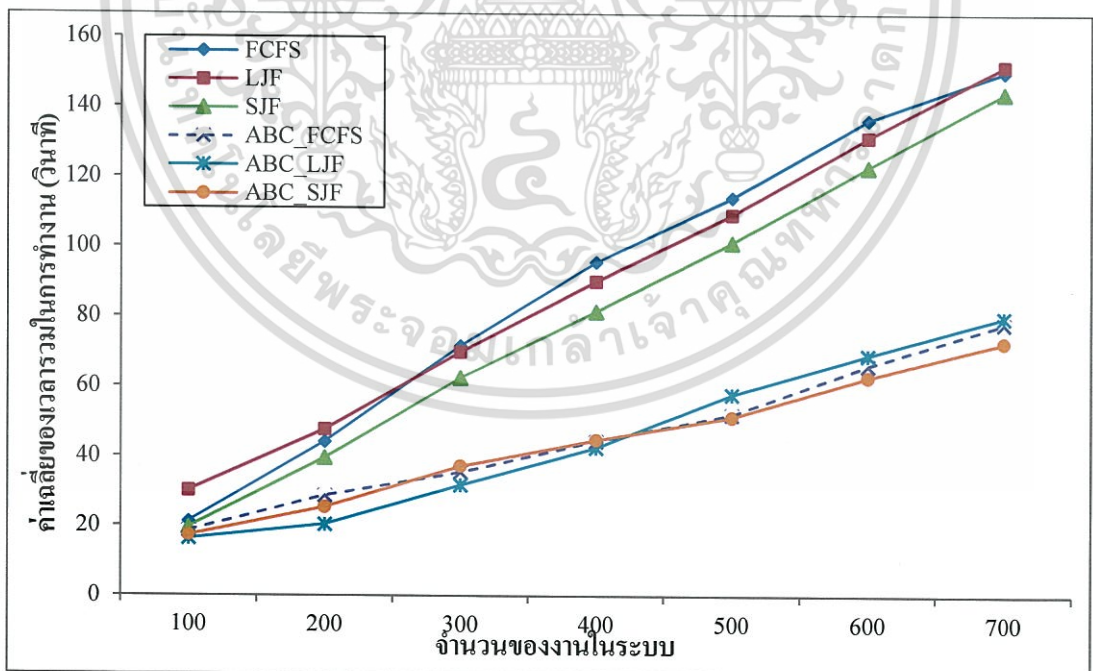
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.39 แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบขั้นตอนการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในต่างกัน โดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ผลจากการทดลองแสดงให้เห็นว่า ABC\_SJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด เมื่อมีปริมาณงานในระบบจำนวน 100 - 400 งาน และถ้าระบบมีปริมาณงานมากกว่า 400 งานขึ้นไป ABC\_LJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด

#### 4.5.3.4 ผลการทดสอบประสิทธิภาพ เมื่อข้อมูลมีการกระจายแบบเบ้ขวา

การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลงเมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่ 9 ที่มีรูปแบบการกระจายของข้อมูลแบบเบ้ขวา เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางงาน โดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลงและข้อมูลมีการกระจายแบบเบ้ขวา กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในต่างกัน โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.40



รูปที่ 4.40 เวลาทำงานรวมเฉลี่ยของข้อมูล เมื่อข้อมูลมีการกระจายแบบเบ้ขวา (Heterogeneous)

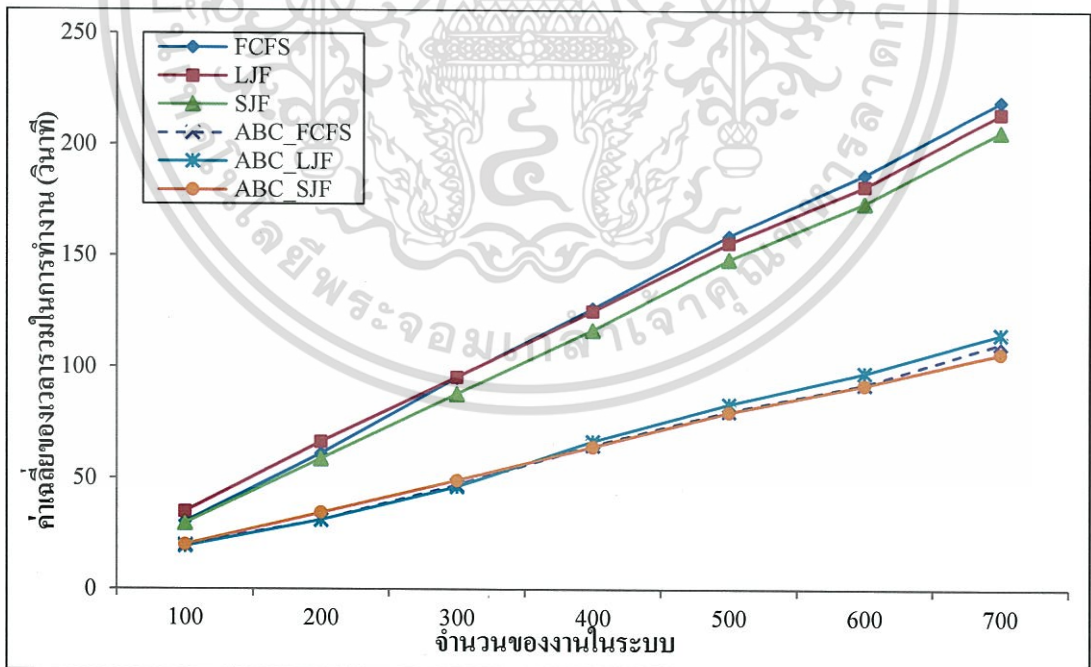
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.40 แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบขั้นตอนการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในต่างกัน โดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียบความคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ผลจากการทดลองแสดงให้เห็นว่า ABC\_LJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด เมื่อมีปริมาณงานในระบบจำนวน 100 - 400 งาน และถ้าระบบมีปริมาณงานมากกว่า 400 งานขึ้นไป ABC\_SJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด

#### 4.5.3.5 ผลการทดสอบประสิทธิภาพ เมื่อข้อมูลมีการกระจายแบบเบ้ซ้าย

การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลงเมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่ 10 ที่มีรูปแบบการกระจายของข้อมูลแบบเบ้ซ้าย เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางงาน โดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียบความคู่กับการจัดลำดับงานแบบฮิวริสติก

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลงและข้อมูลมีการกระจายแบบเบ้ซ้าย กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในต่างกัน โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.41



รูปที่ 4.41 เวลาทำงานรวมเฉลี่ยของข้อมูล เมื่อข้อมูลมีการกระจายแบบเบ้ซ้าย (Heterogeneous)

จากรูปที่ 4.41 แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบขั้นตอนการ จัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในต่างกัน โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ผลจากการทดลองแสดงให้เห็นว่า ABC\_LJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด เมื่อมีปริมาณงานในระบบจำนวน 100 - 300 งาน และถ้าระบบมีปริมาณงานมากกว่า 400 งานขึ้นไป ABC\_SJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด

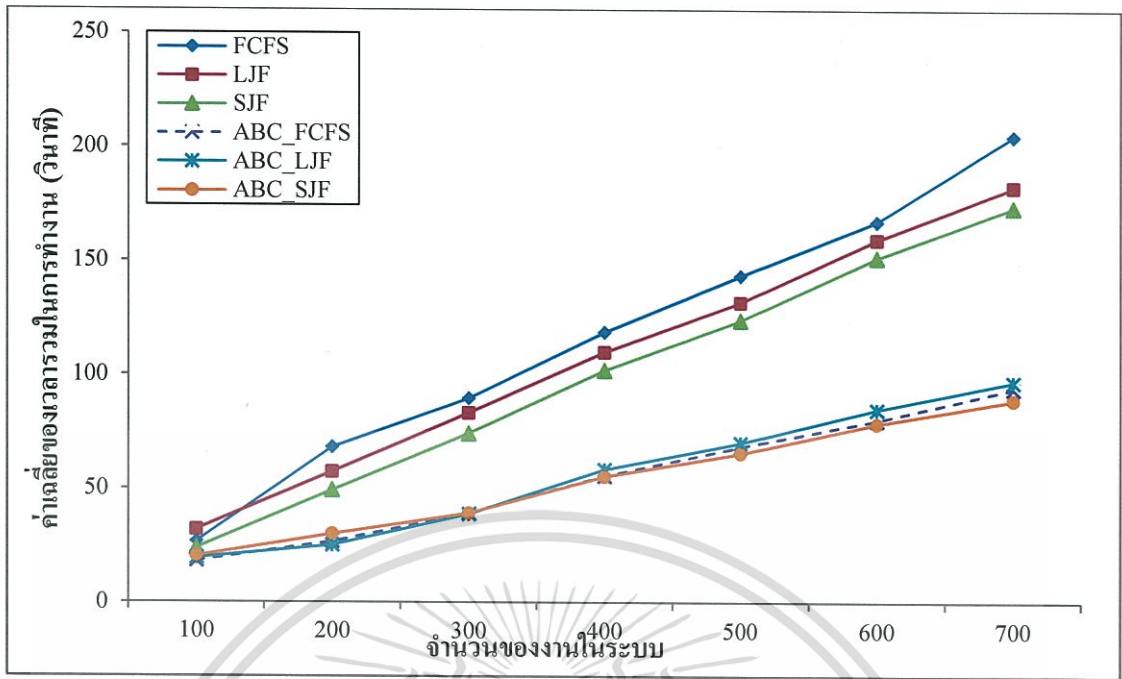
#### 4.5.3.6 ผลการทดสอบประสิทธิภาพ เมื่อจำนวนผึ้งในระบบมีการเปลี่ยนแปลง

การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลงเมื่อจำนวนเครื่องคอมพิวเตอร์เสมือนมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่ 2 เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางการทำงาน โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก โดยในการทดสอบจะทำการเปรียบเทียบ เมื่อค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมผึ้งเทียมมีการเปลี่ยนแปลง ดังตารางที่ 4.18

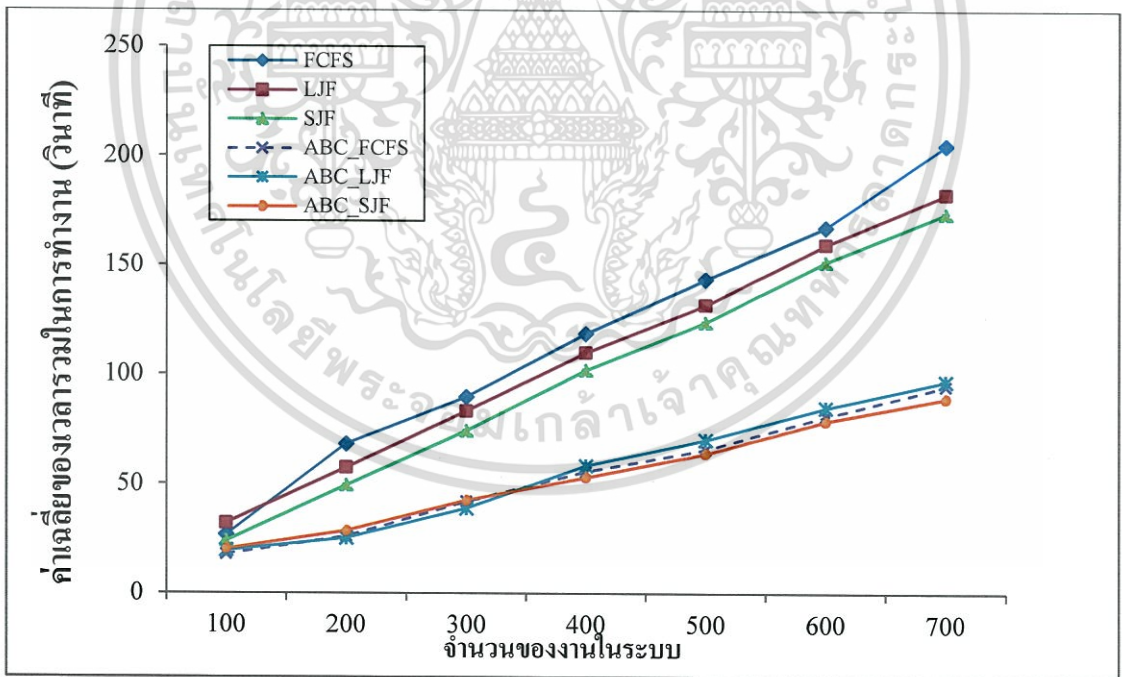
ตารางที่ 4.18 ค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมผึ้งเทียม (Heterogeneous)

Symbol	Parameter	Value I	Value II
n	Number of scout bees	2000	1500
m	Number of sites selected out of n visited site	5	10
e	Number of best site out of m select site	1	1
nep	Number of bees recruited for best e site	1600	1000
nsp	Number of bees recruited for other (m-e) selected sites	400	500

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลงและขนาดของงานมีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในต่างกัน โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.42 และรูปที่ 4.43



รูปที่ 4.42 เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและกำหนดค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมผึ้งเทียม Value I (Heterogeneous)



รูปที่ 4.43 เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและกำหนดค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมผึ้งเทียม Value II (Heterogeneous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.42 และรูปที่ 4.43 แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบขั้นตอนการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในต่างกัน โดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ABC\_SJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด เมื่อปริมาณงานในระบบมีปริมาณเพิ่มมากขึ้นและงานในระบบมีปริมาณมากกว่า 400 งานขึ้นไป แม้ว่าค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมฟังก์ชันเทียบจะมีการเปลี่ยนแปลงการจัดลำดับงานแบบ ABC\_SJF ยังคงสามารถทำงานได้อย่างมีประสิทธิภาพเช่นเดิม เมื่องานในระบบมีปริมาณมากกว่า 400 งานขึ้นไป

#### 4.5.4 ผลการทดลองกรณีทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลง

การประเมินประสิทธิภาพในการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือน โดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก (ABC\_FCFS, ABC\_SJF และ ABC\_LJF) จะทำการเปรียบเทียบกับการจัดลำดับงานแบบฮิวริสติก เพียงอย่างเดียว (FCFS, SJF และ LJF) การตั้งค่าสภาพแวดล้อมของเครื่องคอมพิวเตอร์เสมือน กรณีสภาพแวดล้อมต่างกัน โดยในการทดสอบจะกำหนดจำนวนของงานเท่ากับ 700 งาน และแบ่งการทดสอบออกเป็น 2 กรณี

##### 4.5.4.1 ผลการทดสอบประสิทธิภาพเมื่อชุดข้อมูลมีการเปลี่ยนแปลง

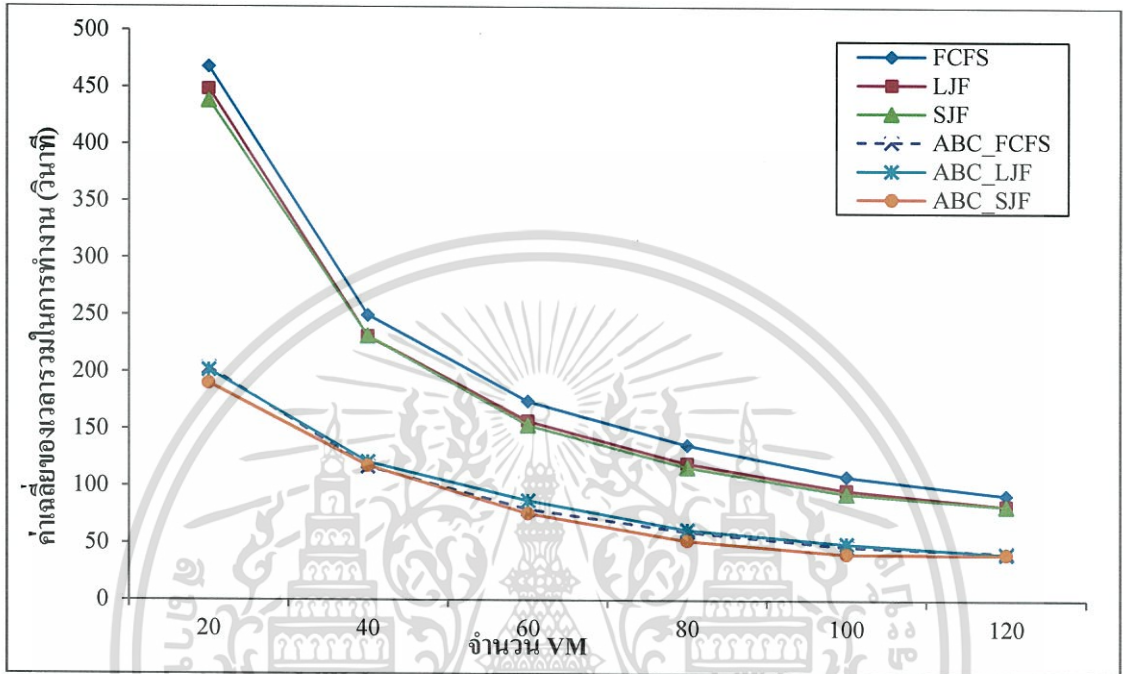
การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของเครื่องคอมพิวเตอร์เสมือนและชุดข้อมูลมีการเปลี่ยนแปลง และจำนวนงานมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลที่แสดงค่าดังตารางที่ 4.19 เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางการทำงาน โดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก

ตารางที่ 4.19 ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของการทดสอบกรณีที่ 4.5.4.1

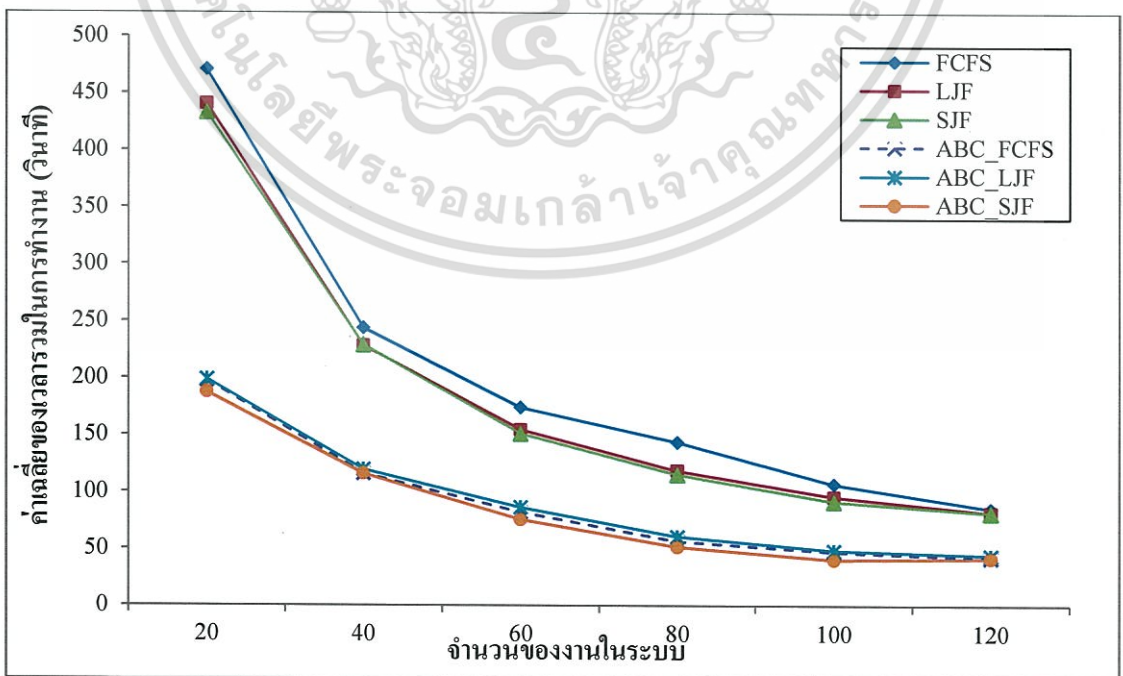
ชุดข้อมูล	ขนาดของงาน (MI)	การกระจายข้อมูล
1	5000 - 20000	สุ่มอิสระ
2	5000 - 20000	สุ่มอิสระ
3	5000 - 20000	สุ่มอิสระ

หมายเหตุ: MI = Million Instruction

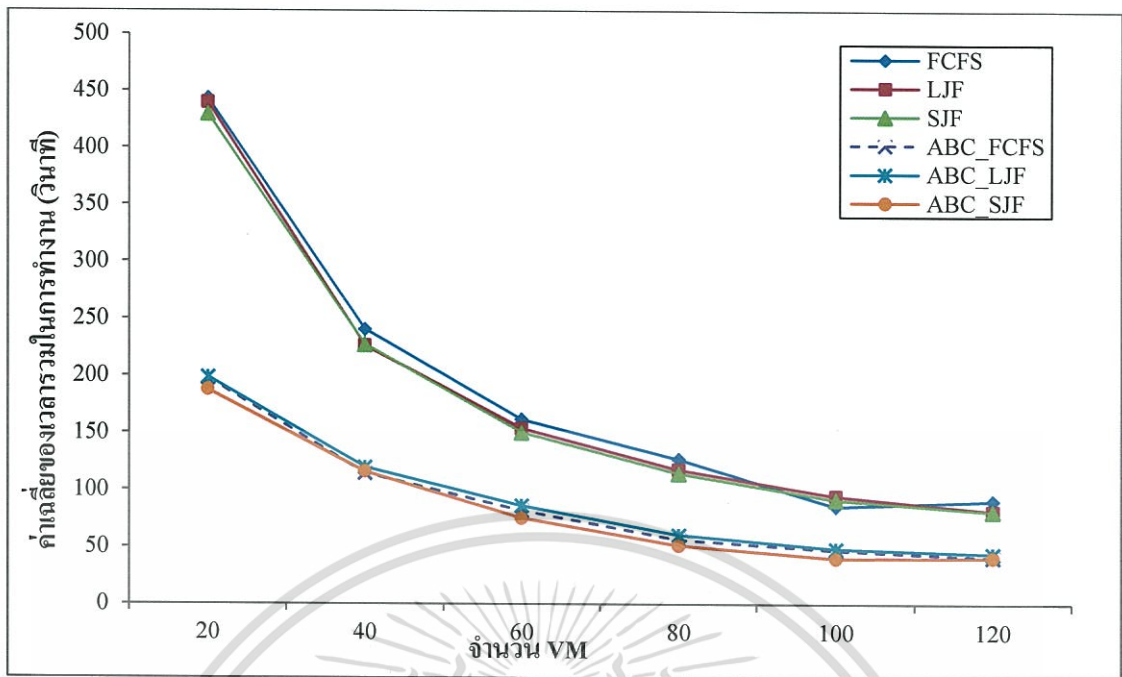
ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนเครื่องคอมพิวเตอร์เสมือนมีการเปลี่ยนแปลงและชุดข้อมูลที่ใช้ในการทดสอบมีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในต่างกัน โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.44 - 4.46



รูปที่ 4.44 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 เมื่อจำนวน VM เพิ่มขึ้น (Heterogeneous)



เอกสารนี้เป็นรูปที่ 4.45 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 2 เมื่อจำนวน VM เพิ่มขึ้น (Heterogeneous) การคำนวณค่าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.46 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 3 เมื่อจำนวน VM เพิ่มขึ้น (Heterogeneous)

จากรูปที่ 4.44 – 4.46 แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบขั้นตอนการ จัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในต่างกัน โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ABC\_SJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด เมื่อปริมาณเครื่องคอมพิวเตอร์เสมือนในระบบมีปริมาณเพิ่มมากขึ้น แม้ว่าข้อมูลการเปลี่ยนแปลง การจัดลำดับงานแบบ ABC\_SJF ยังคงสามารถทำงานได้อย่างมีประสิทธิภาพเช่นเดิม

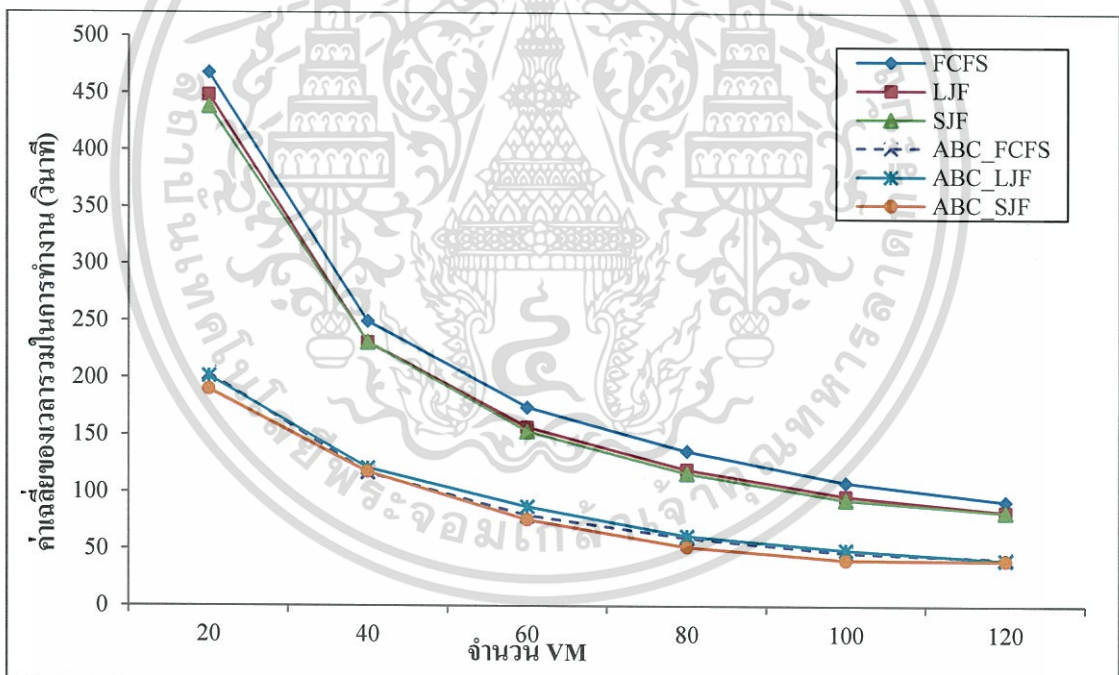
#### 4.5.4.2 ผลการทดสอบประสิทธิภาพ เมื่อขนาดของงานมีการเปลี่ยนแปลง

การทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของเครื่องคอมพิวเตอร์เสมือนและขนาดของงานมีการเปลี่ยนแปลง เมื่อจำนวนงานมีค่าคงที่ ในการทดลองจะใช้ชุดข้อมูลและขนาดของงาน ดังตารางที่ 4.20 เพื่อทำการเปรียบเทียบความสามารถในการทำงานของการจัดตารางงาน โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก

ตารางที่ 4.20 แสดงชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของการทดสอบกรณีที่ 4.5.4.2

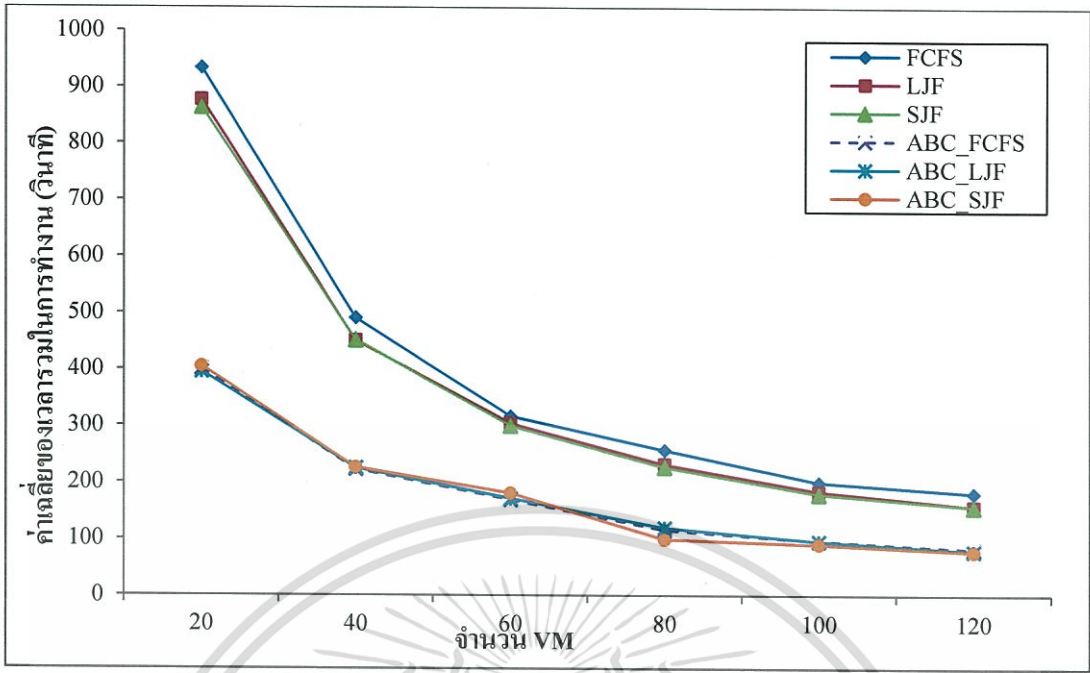
ชุดข้อมูล	ขนาดของงาน (MI)	การกระจายข้อมูล
1	5000 - 20000	สุ่มอิสระ
4	15000 - 35000	สุ่มอิสระ
5	25000 - 45000	สุ่มอิสระ
6	35000 - 45000	สุ่มอิสระ
7	45000 - 65000	สุ่มอิสระ

ผลการทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนงานที่มีการเปลี่ยนแปลงและขนาดของงานมีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในต่างกัน โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของระบบ และแกน X แสดงจำนวนงานในระบบ แสดงดังรูปที่ 4.47 - 4.51

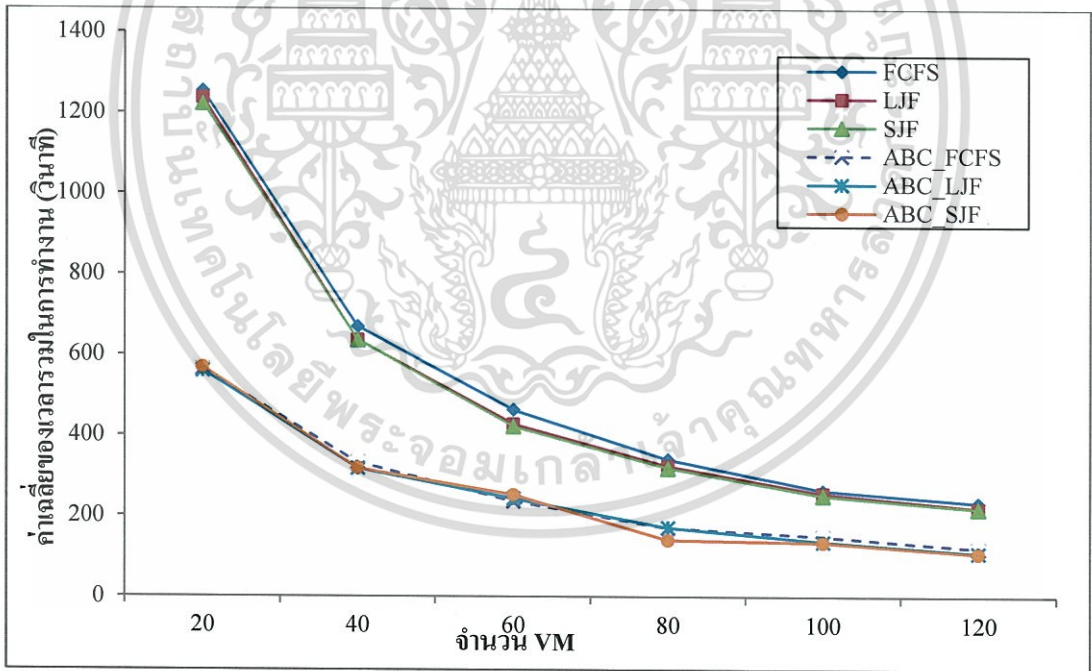


รูปที่ 4.47 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 1 ขนาดของงาน 5000 – 20000 (Heterogeneous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

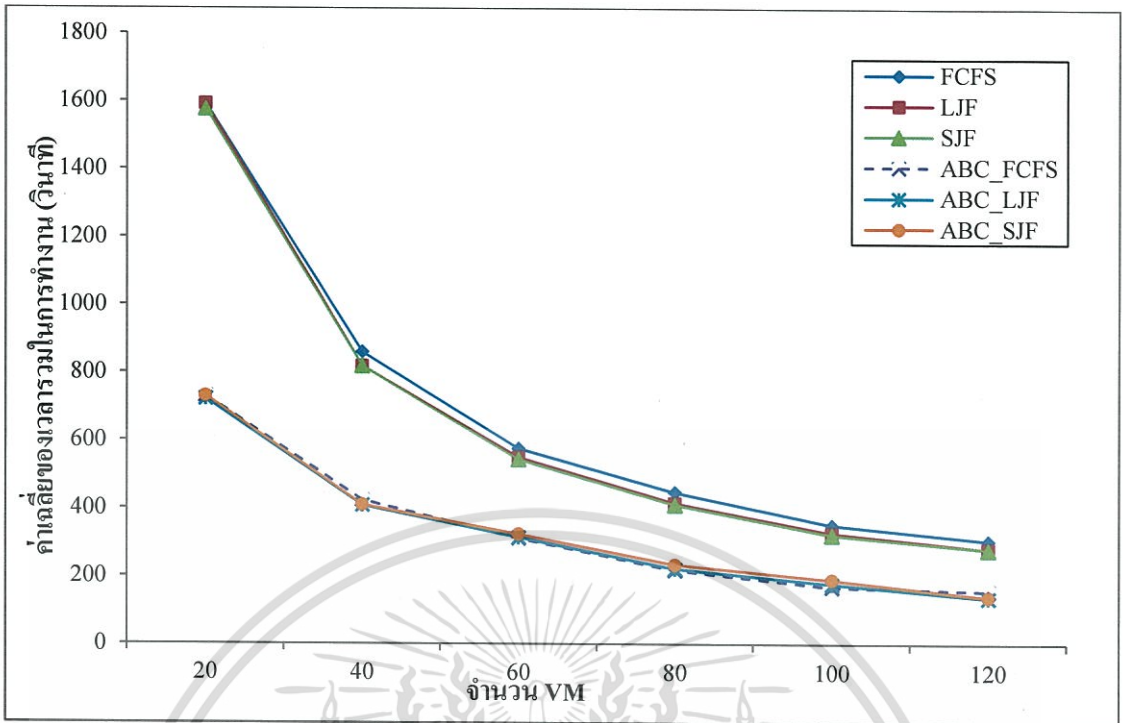


รูปที่ 4.48 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 4 ขนาดของงาน 15000 - 35000 (Heterogeneous)

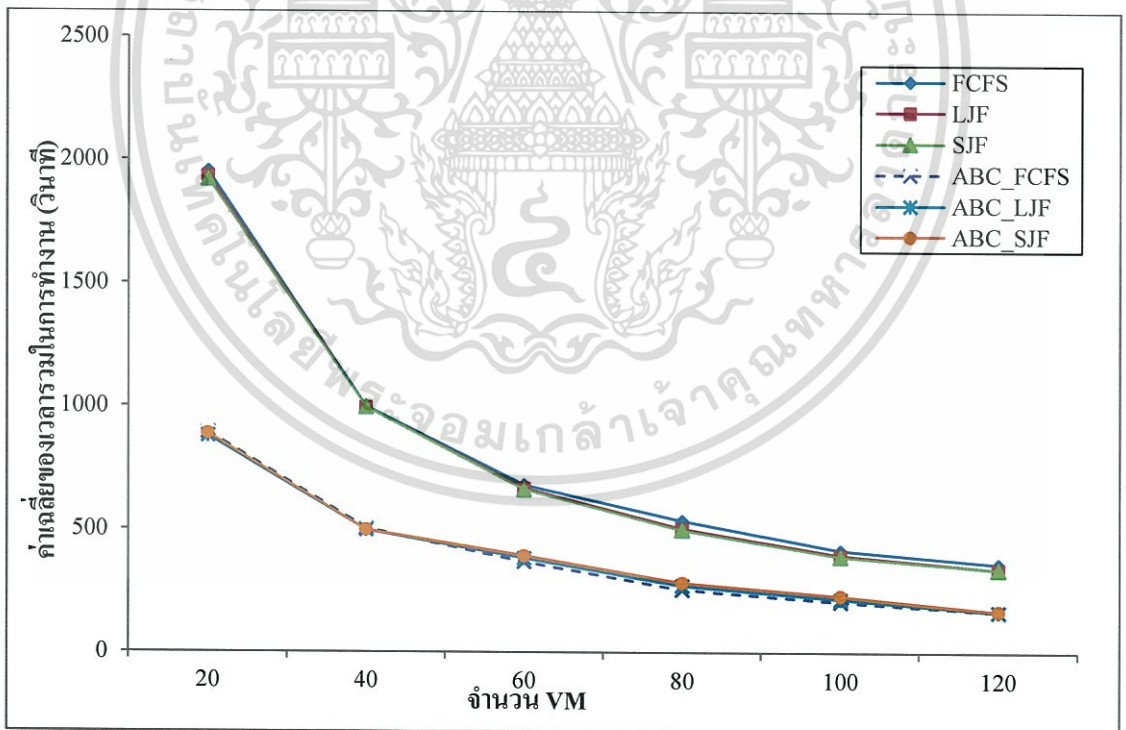


รูปที่ 4.49 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 5 ขนาดของงาน 25000 - 45000 (Heterogeneous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.50 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 6 ขนาดของงาน 35000 - 55000 (Heterogeneous)



รูปที่ 4.51 เวลาทำงานรวมเฉลี่ยของข้อมูลชุดที่ 7 ขนาดของงาน 45000 - 65000 (Heterogeneous)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.47 – 4.51 แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบ ขั้นตอนการจัดการงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในต่างกัน โดยใช้ ขั้นตอนวิธีอาณานิคมซึ่งเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า เมื่อนำขั้นตอนอาณานิคมซึ่งเทียบมาทำงานควบคู่กับการจัดการงานแบบฮิวริสติก ผลการทดลองจะมีประสิทธิภาพใกล้เคียงกัน ทำให้ไม่สามารถสรุปได้ว่าการจัดการงานโดยขั้นตอนอาณานิคมซึ่งเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติกแบบใดดีที่สุด เนื่องจากต้องทำงานทดลองเพิ่มเติม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### บทสรุปและข้อเสนอแนะ

#### 5.1 สรุปผลและวิเคราะห์ผลการทดลอง

งานวิจัยนี้มีวัตถุประสงค์เพื่อขั้นตอนวิธีการจัดตารางงานบนพื้นฐานของการกระจายงานแบบสมดุลในการประมวลผลแบบกลุ่มเมฆ โดยใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก โดยผลการทดลองแสดงให้เห็นว่า ในกรณีที่สภาพแวดล้อมภายในของเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกัน ควรประยุกต์ใช้ขั้นตอนวิธีอาณานิคมฟังก์ชันที่นำเสนอทำงานร่วมกับการจัดตารางงาน โดยพิจารณาจากงานที่มีขนาดใหญ่ที่สุดเข้าทำก่อน (ABC\_LJF) ถึงแม้ว่าลักษณะข้อมูลจะแตกต่างกันออกไป เช่น ข้อมูลจะเป็นข้อมูลแบบสุ่ม ข้อมูลที่มีการกระจายแบบโค้งปกติ ข้อมูลที่มีการกระจายแบบเบ้ขวาและข้อมูลที่มีการกระจายแบบเบ้ซ้าย ABC\_LJF ยังคงสามารถทำงาน ได้มีประสิทธิภาพเช่นเดิม และในกรณีที่สภาพแวดล้อมภายในของเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมในแตกต่างกัน ควรใช้วิธีการที่นำเสนอควบคู่กับการจัดลำดับงาน โดยพิจารณาจากงานที่มีขนาดเล็กที่สุดเข้าทำงานก่อน (ABC\_SJF) เมื่อปริมาณของงานมีมากกว่า 500 งานและข้อมูลมีลักษณะการสุ่มแบบอิสระ แต่ในกรณีที่ข้อมูลมีรูปแบบการกระจายแบบโค้งปกติ ในช่วงข้อมูล 100 – 400 งาน ควรใช้ ABC\_SJF ในการจัดตารางเข้าทำงาน และในทางตรงข้าม ถ้าข้อมูลมีปริมาณมากกว่า 400 งานขึ้นไป ควรใช้ ABC\_LJF ในการจัดตารางเข้าทำงาน และในกรณีที่ข้อมูลมีรูปแบบการกระจายงานแบบเบ้ซ้ายและเบ้ขวา ABC\_LJF มีประสิทธิภาพในการจัดตารางงานดีที่สุดในเมื่อมีปริมาณงานในระบบประมาณ 100 - 400 งาน และถ้าปริมาณงานมากกว่า 400 งานขึ้นไป ควรใช้ ABC\_SJF ในการจัดตารางงาน จะทำให้ประสิทธิภาพในการจัดตารางการทำงานเหมาะสมที่สุด ในการกระจายงานแบบสมดุลและเวลารวมในการทำงานน้อยที่สุด เมื่อปริมาณงานในระบบเพิ่มมากขึ้นและสภาพแวดล้อมภายในของระบบมีการเปลี่ยนแปลง

## 5.2 ข้อเสนอแนะ

- 1) ทำการทดสอบกับชุดข้อมูลที่มีความหลากหลายและมีปริมาณข้อมูลมากขึ้น เพื่อทดสอบสมมติฐานว่า วิธีการจัดตารางการเข้าทำงานเครื่องคอมพิวเตอร์เสมือนที่นำเสนอจะสามารถลดเวลารวมในการทำงานให้น้อยลง
- 2) ทดสอบว่าระบบมีการกระจายงานแบบสมดุลดีหรือไม่ เมื่อข้อมูลมีความหลากหลายและข้อมูลมีปริมาณเพิ่มมากขึ้น
- 3) เนื่องจากการทำงานของเครื่องคอมพิวเตอร์เสมือนกรณีศึกษาที่สภาพแวดล้อมภายในแตกต่างกัน ยังมีพารามิเตอร์ที่มีความหลากหลาย ดังนั้นเปอร์เซ็นต์ค่าความต่างของเวลาที่ลดลงในการจัดตารางงานแบบฮิวริสติกจึงเป็นประเด็นที่ต้องทำการศึกษาคืบต่อไป



## เอกสารอ้างอิง

- [1] Bitam B. “Bees Life Algorithm for Job Scheduling in Cloud Computing” **Conf. on Computing and Information Technology (ICIT 2012)**. 2012. pp. 186 – 191
- [2] Hsu Y.-C., Liu P. and Wu Jan-Jan. “Job Sequence Scheduling for Cloud Computing” **IEEE Int. Conf. on Cloud and Service Computing**. 2011. December 2011. pp. 212 – 219
- [3] Tang K., Zang J. M., and Feng C. H. “Application Centric Lifecycle Framework in Cloud” **IEEE 8th Int. Conf. on e-Business Engineering (ICEBE)**. 2011. pp. 329 – 334
- [4] Buyya R., Yeo C. S., Venugopal S., Broberg J. and Brandic I. “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility” **Journal Future Generation Computer Systems**. 2009, Vol. 25 Issue 6, June 2009. pp. 599 – 616
- [5] Carey W. P. (4 June, 2008) “**Cloud Computing: The Evolution of Software-as-a-Service.**” สืบค้นเมื่อ 20 ก.พ. 2557. จาก <http://knowledge.wpcarey.asu.edu/article.cfm?articleid=1614>
- [6] Robison S.(2007) “**The next wave: Everything as a service, Executive Viewpoint.**” สืบค้นเมื่อ 19 ก.พ. 2557. จาก <http://www.hp.com/hpinfo/execteam/articles/robison/>
- [7] สุวันดี ศิลปศร. “**Cloud Computing**” สืบค้นเมื่อ 19 ก.พ. 2557. จาก <https://sites.google.com/site/suwandee15510/-cloud-computing/rup-baeb-kar-hi-brikar-khxng-cloud-computing>.
- [8] Anonymous. (ม.ป.ป.) “**Amazon**” สืบค้นเมื่อ 20 ก.พ. 2557. จาก <http://aws.amazon.com/ec2/>
- [9] jongjin.(2 Sep, 2008) “**ระบบการประมวลผลแบบกลุ่มเมฆ (Cloud Computing)**” สืบค้นเมื่อ 20 ก.พ. 2557. จาก <http://www.vcharkarn.com/vblog/38378/1>.
- [10] Dutta K. “A Smart Job Scheduling System for Cloud Computing Service Providers and Users: Modeling and Simulation” **1st Int’l Conf. on Recent Advances in Information Technology RAIT-2012**. 2012.
- [11] Sindhu S. and Mukherjee S. “Efficient Task Scheduling Algorithms for Cloud Computing Environment” **High Performance Architecture and Grid Computing Communications in Computer and Information Science**. 2011, vol. 169. July 2011. pp. 79-83
- [12] Phan D.T., Ghanbarzadeh A., Koc E., Otri S., Rahim S. and Zaidi M. “The Bees Algorithm. Technical Note.” **Intelligent Production Machines and Systems**. UK. 2005

## เอกสารอ้างอิง (ต่อ)

- [13] Phan D.T., Ghanbarzadeh A., Koc E., Otri S., Rahim S. and Zaidi M. “The Bees Algorithm - A Novel Tool for Complex Optimisation Problems.” **Proc. of IPROMS 2006 Conf.** 2006. pp. 454-461
- [14] Guney and Oney. “**Expert System with Applications**” Vol. 37, Issue 4. April 2010. pp. 2707-3528
- [15] Karaboga D. “An Idea Based on Honey Bee Swarm for Numerical Optimization.” **Technical Report TR06**, Vol. 200. Erciyes University, Engineering Faculty, Computer Engineering Department. October 2005.
- [16] Azmi Z.R.M., Bakar K.A., Shamsir M.S., Abdullah A.H., Shamsir S. and Manan W.N.W. “Performance Comparison of Priority Rule Scheduling Algorithms Using Different Inter Arrival Time Jobs in Grid Environment” **Int. Journal of Grid and Distributed Computing.** 2011, Vol. 4, No.3. September 2011. pp. 61-70
- [17] Pas.(Aug 2551) “**บทที่ 5. Cpu scheduling.**” สืบค้นเมื่อ 20 ก.พ. 2557. จาก <http://www.oknation.net/blog/print.php?id=304778>.
- [18] Calheiros R. N., Ranjan R., Rose C. A. F. D. and Buyya R. “CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services” **arXiv preprint arXiv: 0903.2525.** 2009. pp. 1-9
- [19] Calheiros R. N., Ranjan R., Rose C. A. F. D. and Buyya R., “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms” **Softw. Pract. Exper.** 2010, Vol. 41, No.1. August. 2010. pp. 23-50
- [20] Anonymous. (ม.ป.ป.) “**6. การจัดการกระจายข้อมูล.**” สืบค้นเมื่อ 20 ก.พ. 2557. จาก [http://www.mwit.ac.th/~math/E\\_Learning/MATH30203/sources/Statistics\\_06.pdf](http://www.mwit.ac.th/~math/E_Learning/MATH30203/sources/Statistics_06.pdf)
- [21] Mondal B., Dasgupta K. and Dutta P. “Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach,” **Procedia Technology.** 2012, vol. 4. 2012. pp. 783-789
- [22] Fang Y., Wang F. and Ge J. “A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing” **Web Information Systems and Mining, Lecture Notes in Computer Science (WISM 2010, LNCS).** 2010, Vol. 6318. 2010. pp. 271-277

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง (ต่อ)

- [23] Hu J., Gu J., Sun G. and Zhao T. “A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment” **3rd Int. Symp. on Parallel Architectures, Algorithms and Programming (PAAP)**. 18-20 December, 2010. pp. 89-96
- [24] Li K., Xu G., Zhao G., Dong Y. and Wang D. “Cloud Task Scheduling based on Load Balancing Ant Colony Optimization” **6th Annual ChinaGrid Conf.** August, 2011. pp. 3-9.
- [25] Mizan T., Masud S. M. R. A., and Latip R. “Modified Bees Life Algorithm for Job Scheduling in Hybrid Cloud” **Int. Journal of Engineering and Technology (IJET)**. 2012, Vol. 2, no.6. June 2012. pp. 974-979
- [26] Kruekaew B. and Kimpan W. “Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony” **Proc. of the Int. MultiConference of Engineers and Computer Science 2014 (IMECS 2014)**. 2014, Vol. I. Hong Kong. March 12-14, 2014. pp. 18-22
- [27] บุญหทัย เครือแก้ว และ วราภรณ์ กิมปาน. “การจัดตารางงานบนพื้นฐานของการกระจายงานสมดุลในการประมวลผลแบบกลุ่มเมฆโดยขั้นตอนวิธีอาณานิคมผึ้งเทียม” **The 10<sup>th</sup> Nat. Conf. on Computing and Information Technology (NCCIT2014)**. Phuket, Thailand, May 8-9, 2014. pp. 700 – 705



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

## ตัวอย่างชุดข้อมูลที่ 2

## ข้อมูลจำนวน 100 งาน

19539	16946	8825	9320	18427	10919	19140	15891	6094	9734
16656	17413	13104	15524	7015	6511	13669	13238	18482	14014
15344	18646	12112	15947	9344	8958	8784	10660	13888	9478
11678	5965	18354	6512	19434	7660	12207	7990	9423	11689
10666	13043	12120	9819	8577	9135	19989	15318	14244	16496
8402	19976	14346	5696	9217	19748	5912	14810	17907	9376
17365	7388	10128	6546	9561	7222	15281	5668	10127	11278
6317	8161	19343	5694	9852	19210	15537	12508	17368	14143
19832	12706	15385	12965	13324	11217	6103	6125	8444	18281
15612	5297	19849	5726	5592	13389	5091	16983	13816	11003

## ข้อมูลจำนวน 200 งาน

14587	10824	15010	8775	19384	12113	17189	6624	16406	13821
10498	9962	18495	8484	14172	16833	7017	5953	12503	19351
19885	13445	11420	18140	16167	9202	8451	6921	9242	9515
19041	13294	11957	18509	15740	5130	11009	11782	9732	10585
15212	14566	6644	14282	9192	16076	7683	18302	18018	9341
16281	10810	5869	9099	14510	19429	8047	15089	11010	6017
16804	9479	12492	5266	6831	14688	13562	15168	10633	9465
13409	5164	11764	7263	12166	18757	17193	15056	18368	17840
15523	5683	11776	7896	6066	13678	18561	14167	12141	18589
7722	14347	18348	19397	15181	9329	17060	15236	9967	14353
7390	18387	18161	8648	16480	14485	19625	5654	16672	7269
8562	15893	7191	7125	13130	17475	19453	5044	12753	6213
7904	16600	6429	18689	17003	13353	12205	8247	7886	8499
7640	13134	5080	8818	5951	13840	17886	13161	6969	6994

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13005	9876	14392	9336	9980	12476	16469	7865	12893	16489
18453	6797	15908	17545	8186	18340	19597	5604	10617	7246
10300	14662	18095	6742	13447	17926	12938	12953	6279	5362
8213	13042	15765	15763	14116	17074	8719	5888	7145	14213
16927	6531	16955	12072	14565	12261	8823	7830	6836	17221
19516	14429	17989	15796	16956	18861	15416	12938	6545	8904

### ข้อมูลจำนวน 300 งาน

16884	15865	15866	14561	8922	7729	10270	17377	18993	12965
11538	17816	10728	10149	18764	16270	17367	12380	13720	8691
18030	12015	9022	8704	13296	8681	8521	15504	17061	5342
19277	18709	19151	7840	5667	17401	10995	18253	19182	19015
6455	13783	7399	6090	5464	7686	13606	14237	19561	10819
5241	10043	11575	17413	7842	12805	7730	9661	15426	8448
5971	5024	7995	12847	19512	7276	18370	15875	16478	13796
19904	11196	18310	5030	17422	8330	7485	11018	17413	13130
15919	11230	18946	11391	13763	15847	16184	15801	6065	9838
9319	14905	10095	11442	19717	12718	15102	11793	16821	13446
18475	14787	15566	19370	16564	18451	12527	5101	7159	15770
5813	18941	11520	12368	18054	16008	12114	15045	16937	9776
7344	8254	6409	14922	12306	18325	19160	13355	11522	7011
12153	17943	5846	11133	17649	6913	11737	10009	12771	14963
9288	6246	10921	14055	13475	5768	7800	16885	7762	9970
13346	15932	15050	18192	10296	7356	9591	8677	5439	11002
8807	5710	7517	15378	5404	16325	10664	15278	19919	18810
14695	6034	13683	12581	17554	5723	16010	6866	12772	9893
5109	16385	15166	5443	5099	19543	10935	15442	6801	16149
19022	19468	12321	15095	13013	5105	18036	11308	13873	13112
5291	10579	5332	5958	7943	13175	12692	12324	9483	9520
18229	13525	11570	14892	12632	16943	16530	14254	17950	19002
15994	19235	10957	13516	16669	8407	16082	7699	7001	10246

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15970	10223	7551	5069	12653	16303	7926	15683	14051	5206
10023	15463	18525	8404	15708	18526	6272	11227	7277	5665
12100	18301	19982	13523	10681	7916	17169	16068	17019	19738
15381	13774	8355	13895	9784	10254	6537	16359	8230	15989
14531	8463	8354	7761	15972	9118	12326	14602	5359	16750
15529	7499	11691	9942	9668	11382	19507	19368	6944	8955
12977	12269	10803	13774	18222	9815	18205	11546	17299	19767

### ข้อมูลจำนวน 400 งาน

16887	8880	6958	15636	16335	6491	7407	18405	18595	16315
17985	18705	12003	11089	11163	19856	12595	6555	15575	12211
7789	8042	7911	15658	17745	19775	19924	13357	9876	5793
5194	5525	19150	12128	16350	5216	5005	13973	10048	16715
14283	15372	18077	9801	18090	19072	13323	8565	12387	6921
12133	10681	7855	8194	13355	18565	18529	8295	13389	7205
15869	14031	10780	12140	13861	5459	6513	10556	15725	10010
14537	15819	17780	9123	11346	18713	13256	18775	11527	19106
8037	17904	11246	12783	5397	13684	18255	12120	19178	18646
8133	10728	15174	19943	14135	11916	13107	12330	17867	15124
19597	16705	9056	14600	15134	17563	11845	17880	12313	10038
6813	10014	19006	19148	14979	9178	13101	9735	18669	11406
11694	17326	16534	11629	14136	9678	6643	16836	15034	18143
6820	15278	17543	7443	15256	16305	9620	15920	17942	19920
5683	6722	8246	17039	11144	10434	17985	11400	16947	18118
18849	15046	13663	15347	18049	6389	16344	17956	7552	14679
13695	18743	9297	16206	16159	9589	5085	15391	8776	18365
19239	7868	10969	9269	13171	15917	12905	18283	13732	8101
5887	16191	11412	12947	7911	19062	9348	12106	11625	17593
5543	7593	6473	15066	5666	12895	7847	8090	6827	13338
16156	12743	7082	18566	13957	15475	17804	15970	14766	14154
18254	17506	13759	5863	16729	10694	10308	15534	7177	5042

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14255	6295	10937	10406	18451	12632	14763	7364	7755	10154
10693	8984	11195	11528	19363	15559	18135	18764	5316	19412
18478	15911	12439	10750	10968	6547	11202	15669	16875	10798
16870	5259	13808	16904	18494	7493	18061	19639	10418	16622
9866	7817	12436	19196	9560	6653	12258	9902	12525	16970
17171	12655	16331	17646	15215	13108	11811	12178	9494	18457
15143	19240	8883	6299	15172	17452	17993	7679	19492	5688
14222	6113	15704	8826	16088	9384	10917	11508	13725	9100
16896	16085	11462	14019	6777	17789	7344	11029	8983	15496
19340	17259	9141	13732	13181	7613	6843	9474	8313	19859
6964	6372	13994	10865	17184	8614	6872	7164	8793	13967
9880	18662	16631	12222	13366	12381	13208	9644	6979	18183
17996	9530	14680	12443	12391	16928	5323	8039	13578	8859
16379	15358	8441	9534	12368	13349	16253	10831	11192	18521
17244	9281	7751	9909	15085	19509	16586	12059	7240	12256
17869	7008	12700	12409	10775	9377	5877	8049	6978	7474
16031	18454	18095	10451	9126	17231	7841	13857	11407	19651
9683	7953	7977	8290	15195	15640	18801	10315	11909	7851

### ข้อมูลจำนวน 500 งาน

14334	19422	16405	6916	13698	15970	17026	5864	7082	11398
13462	5893	10670	16081	10550	10755	6096	8342	15711	12480
16424	18704	16485	10617	13579	9525	5494	19869	12080	12401
7555	7649	19339	18315	15546	18540	17422	15530	6228	16954
9879	5746	12162	19026	6045	18230	9048	17902	9222	18397
18198	18523	12437	17258	15391	13837	14861	6536	12068	16430
18438	16521	8306	11318	14197	12401	9117	7835	14774	11769
19984	17665	13328	18742	7089	7635	11093	6069	10122	8578
5848	17963	12618	14852	7128	10270	12514	8184	18502	19129
15020	10812	11913	13342	17293	12734	10859	15687	6714	15310
16713	5265	12779	6269	17399	18640	15997	12746	6027	7243

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12568	6222	7881	11924	16137	11840	7148	6999	7714	8283
10450	8507	6809	6624	7814	15254	5276	5919	17739	6860
6793	13807	12829	5115	9691	9145	9028	14849	12727	16328
6841	6991	16620	6560	10518	7569	14664	16120	6235	15460
19449	13997	18026	17440	8276	19684	19053	11545	15557	18787
10330	12481	6084	12474	14049	10432	5607	10750	18325	8893
19999	8771	5325	19444	10441	16325	13595	12692	6913	12135
18465	5617	9786	12278	8897	18202	5127	9461	10297	14809
10924	10959	14663	19335	17808	15601	12667	12434	18951	7955
6835	16735	13600	8575	13262	18422	19920	6181	5477	10622
17266	6867	15094	5630	17947	10297	6504	19704	5576	15475
9096	6725	13797	8424	13292	7598	8735	5123	5674	18640
11177	10949	15608	14510	7362	5468	19558	9525	16211	9965
6778	7817	15328	19792	6392	6793	12154	18586	8045	12314
15601	9294	9037	8427	18116	7283	18011	10657	14479	13706
18564	12678	11167	7041	15905	10170	18535	16493	19550	19681
7891	5568	17887	6344	18419	12215	17169	14912	18256	12473
9127	9172	12171	7295	11892	8599	16578	14731	13489	12892
10397	15750	16754	6522	7971	12034	19212	17742	14009	14184
6390	18040	8210	9163	8343	13730	5594	14118	13770	11713
9648	10647	10054	10240	9167	18998	14932	6106	9976	5831
5484	6152	16528	14666	19140	9370	19080	13436	19466	7560
14958	7337	12873	17282	18641	12279	14198	8854	19563	14107
9495	5876	8410	13131	9603	18428	6360	17085	10744	14543
7148	6171	17052	6505	18621	12220	16323	18774	9439	7413
9672	14239	11490	5603	5905	17587	16651	18334	10295	17270
14742	10745	10868	17742	12054	12345	7271	11981	11036	8498
11266	17238	12051	19911	15505	16087	14310	7656	16480	9385
16842	7993	11283	14515	5514	11287	5210	11254	12849	10907
15354	16216	11787	16633	7744	15691	15318	18611	19557	7811
19887	8511	9381	13468	12255	9185	13296	15410	9012	12378
5333	13946	14915	11176	12977	14981	10966	17054	18585	8996

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15445	16963	7018	5502	17660	16530	15502	8680	17481	17141
15304	13031	17623	6983	16596	12389	13514	13101	9129	9653
13725	5532	12375	5819	8553	11585	16704	14598	18786	12882
13213	18684	14553	12171	8096	13645	16717	16174	12292	15110
16023	8258	16109	19567	16796	11003	16072	14896	16366	9492
6701	5665	15656	5269	16332	9260	10689	5243	10121	16581
14038	14671	19539	19458	18983	13222	11971	10033	6754	9163

### ข้อมูลจำนวน 600 งาน

17660	17612	10441	9410	18413	5360	15683	19868	5517	7947
15042	6589	16363	9756	12531	13336	6457	8831	9284	10305
16216	13558	7568	7473	11060	7780	9976	11641	6553	9546
6178	5779	6472	15957	15122	17609	19884	9651	9635	13760
8503	10093	6449	14159	10982	18539	16703	14534	19544	16180
17769	17433	11646	17909	11280	6671	19252	17283	19887	8761
14473	14739	7771	16235	19661	17918	11746	18556	17068	14266
10381	14764	7936	12183	18901	5948	14021	13648	15409	17634
11452	13770	5352	10022	14834	13887	5875	18138	7389	6857
6688	17001	5659	7776	17789	6027	10435	11320	14462	18581
18091	15230	5745	9702	6613	19014	10842	12160	5038	13336
7391	10276	6237	9111	17467	17357	19472	6894	12358	12315
5702	11197	19345	7651	6899	12357	11709	17364	8781	15397
19883	13385	12320	15983	13819	18406	9010	5436	13568	8929
7907	7057	8106	7786	19264	15414	6442	11194	12809	5211
12883	8714	5571	16804	15371	18251	14238	6715	17380	7051
9666	14727	17770	11973	12599	13423	9204	18871	17471	10403
18599	9369	6660	11123	12230	15758	14822	9359	19803	19858
15834	11296	15503	7256	18516	16383	8704	19997	13695	10500
12024	5089	19842	17539	17279	17755	16733	10961	15391	7421
9097	18377	13264	16178	14822	15169	16675	18406	18604	11161
13060	18275	9765	16831	11447	17510	8442	6483	12039	19886

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8120	15555	12350	12701	13938	15161	6467	17378	15339	16900
7081	9151	18387	15721	16856	7305	11491	10598	11673	14105
16246	8176	12053	15848	6317	5451	17664	12446	11517	15914
15886	11520	18995	16463	8865	13689	5933	7301	8932	19573
12599	13141	17547	8936	16459	16978	14995	5876	17629	16209
16298	11073	16766	6318	11438	17823	19576	15220	12242	12462
8861	11437	10998	8416	8805	10949	13785	7691	13207	16587
9734	16825	8967	10625	9456	14110	17325	19332	15066	5532
12561	8598	5913	12589	12333	18597	8245	15797	12391	6413
17093	18122	14078	5112	9238	6437	7582	19049	15206	7986
14986	14623	5231	16416	7326	18228	8658	10074	11128	11073
5377	13894	19977	18264	18527	10067	6997	17625	13977	18485
12898	16875	17145	7168	8189	17702	15406	13120	11330	5674
16547	9351	19720	11197	18848	8522	18737	10341	12167	14562
17151	14136	6884	6706	9197	13748	12941	16829	17873	10088
10292	11559	10910	15723	19597	10272	17660	6775	17378	7297
15187	13863	9120	5536	6104	14040	19550	8124	15957	6805
18661	17760	5639	6148	15494	16264	5796	14146	16582	6079
17667	14475	7604	12067	8071	12996	15720	17471	9610	17000
15896	6654	18369	16071	14529	17188	12368	6356	14219	8603
12312	19605	15479	16255	19729	5861	12496	12204	18405	5559
6211	8394	19963	9870	14906	17209	8504	12147	15490	17291
15277	13827	5174	11183	15381	7637	8321	7326	15015	15888
13837	15520	7518	13136	13449	13664	6020	16969	11765	8349
15167	6858	18300	10788	7514	19803	8983	19041	5276	13781
10015	11701	14504	5514	8821	5152	17982	18787	7344	16826
12475	18214	12642	8211	13541	13792	17789	15504	11660	15408
18248	8755	18281	16009	6613	12620	8915	13553	8598	7688
18882	17186	10733	17041	17175	14884	15431	17894	7754	9137
5152	18913	6902	9933	5991	6348	11928	5064	11613	19060
17303	6861	6954	18695	7950	14195	8989	12946	12665	13800
11595	18895	17748	5356	9254	14697	15383	6313	6437	5870

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18862	19105	11344	6167	16439	16828	13563	6348	5360	16783
12176	5533	13210	6930	9146	16410	12607	15370	15549	17001
13409	13455	8117	14682	17942	7138	16481	15359	12540	18687
6692	8839	6189	17669	18441	10101	10141	12478	18847	7243
7794	18218	16803	19030	9166	12409	17217	12437	12821	19965
6461	14900	8424	8079	16649	12337	14441	18274	19935	5576

### ข้อมูลจำนวน 700 งาน

17749	13829	7352	6530	14230	19433	18453	12720	19317	12079
14813	11754	7464	17617	5706	11274	16758	7474	17339	6897
10973	7604	15789	19807	9419	5481	14317	7435	10983	15857
18354	18162	16149	17573	6253	5965	9957	17497	13959	8591
8281	5860	12832	10746	10320	17106	11343	15176	14062	7487
19888	12627	6567	7507	9093	18652	19372	9321	13512	19833
17150	10156	15752	15781	19645	14944	18667	15389	12950	5641
18322	8788	8841	9019	9074	16551	13475	11422	16259	13804
9318	8476	17613	18623	13109	11659	7197	12481	15910	7353
7670	18960	16690	11408	17670	18954	11561	5723	13309	8604
11143	10704	18053	19010	11827	14647	15734	13673	18165	19501
10799	14056	9826	14695	9679	18114	5371	10428	16299	10080
6423	16714	18287	7088	10658	11686	16590	16245	15979	5728
18586	10901	8340	12089	19999	7572	11647	14382	8948	9084
7397	15152	10648	7352	5904	12030	11453	18247	8577	6368
18647	11765	8430	16568	12778	8281	9643	5452	5005	18441
7723	10429	5654	16068	13243	10383	18900	9024	5281	9104
14456	8146	15452	8868	12492	8349	14424	19195	10352	5515
14040	9102	12142	19123	14080	12647	5224	17991	9546	6156
12741	19230	6910	5210	5668	5367	19601	16811	17796	14557
17532	8729	7978	17013	12301	13606	18591	9751	10455	16793
9502	17388	18098	15420	9911	9703	7068	7922	14298	19306
16033	12568	16914	12166	19768	11917	8635	12080	15780	10252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8996	13159	13525	13745	10990	14295	9672	8183	16615	18601
5470	6116	8260	9675	11874	9964	5778	13020	12100	5834
16659	11249	7419	12329	11948	8651	6546	12803	6515	9737
8764	13847	6641	5100	15061	6700	11789	12852	6695	19410
15962	18030	11965	19866	11661	14380	17259	9929	18981	15151
8045	10500	8492	10734	7694	8619	9421	13569	13407	5981
7844	19176	8110	5859	7794	12512	6795	19582	16252	10135
18914	10549	16490	16327	15008	6059	14132	13298	5279	9042
7842	15378	14697	9171	15584	12807	19649	16033	18837	12737
8067	10801	14395	15015	16528	12090	6052	8437	18333	16045
8871	17508	14124	6951	14197	16457	15853	14595	12987	17906
14670	11501	13999	14154	6305	5224	5536	15026	7980	5689
13468	18233	9255	18037	15026	5914	14073	10534	12770	12475
8174	9381	11960	17026	8328	7960	13864	10381	12522	13471
17774	6391	9644	5658	17149	7016	16049	17266	6201	5419
17510	6186	13393	11369	17190	8899	12114	11676	6342	14579
11050	17644	11593	15321	11404	19786	10596	16942	12658	6132
17823	16795	7322	17394	5954	19649	11232	16992	16467	9449
14568	10118	19903	5970	18179	12173	14448	16675	11489	5774
7709	18783	7469	15012	14337	14409	19755	14192	19230	6660
11418	17203	19260	9438	15634	17080	12780	12886	16732	16035
13664	8508	9028	10417	11744	12708	10330	10723	19021	7092
15762	15098	19954	9342	18047	7835	15865	10809	9674	8202
5534	11493	13604	6778	16426	6983	16827	8129	17361	14316
8936	16446	14443	10539	17298	8638	19875	17098	15718	12316
6014	15232	16624	15504	17195	14086	5453	10576	15349	10384
13321	15801	19259	17182	8625	13195	7044	10601	15175	17157
9713	11968	9553	11597	7750	13539	10659	19333	14694	18108
19372	10097	12770	16967	12471	6005	11707	16729	18863	17240
14378	10336	11724	6603	6221	7727	16943	16676	11719	6506
10301	13798	10671	18102	17806	13809	17908	12958	5445	17012
14299	11924	13767	16295	13783	17766	5550	17763	13763	7407

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17628	12316	13199	5576	5296	14773	13454	15074	6486	16625
12467	19484	17292	9972	14052	5613	10240	14166	10198	18988
8968	13351	19290	14873	7422	6149	9229	12690	5044	5789
19325	8836	18782	17030	19027	18130	12941	10079	9565	10642
5545	18379	12385	7105	7233	13503	7092	17848	11055	9803
13070	14157	17745	16634	7806	8912	5542	13041	19310	18738
11846	13988	19622	8745	10421	15894	15605	7545	10704	6386
14858	9631	14966	18286	10687	16507	17189	19749	18877	11028
15822	10581	13533	5387	15292	18903	9498	10404	12772	17058
5329	7087	11794	16802	6023	9357	16899	10422	7075	19531
6999	16000	18561	5131	15851	9305	8221	13430	12649	7458
7078	14703	10464	12627	9808	13642	12956	8938	9015	11243
17523	15651	9986	7997	11604	19942	18992	8594	18234	17028
6553	13837	6474	9279	8094	11211	15182	11621	5531	18824
5637	8968	11796	9549	8679	12788	8620	18388	12424	8365

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Lecture Notes in Engineering and Computer Science**

**IMECS 2014**

International MultiConference of  
**Engineers and Computer  
Scientists 2014**

**Hong Kong  
12-14 March, 2014**

S. I. Ao  
Oscar Castillo  
Craig Douglas  
David Dagan Feng  
Jeong-A Lee (Eds.)

**IAENG**

International Association of Engineers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษา ISBN: 978-988-19252-5-1 ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
ISSN: 2078-0958

# Table of Contents

## International MultiConference of Engineers and Computer Scientists IMECS 2014

Preface.....	xxix
Committees.....	xxx
Abstracts of Invited Lectures.....	1

### VOLUME I

#### The 2014 IAENG International Conference on Artificial Intelligence and Applications

Island-Model-based Distributed Modified Extremal Optimization with Tabu Lists for Reducing Crossovers in Reconciliation Graph <i>Keiichi Tamura, and Hajime Kitakami</i> .....	1
Recurrent-Fuzzy-Neural System-based Adaptive Controller for Nonlinear Uncertain Systems <i>Tzu-Wei Hu, and Ching-Hung Lee</i> .....	7
Cluster-based Particle Swarm Algorithm for Solving the Mastermind Problem <i>Dan Partynski</i> .....	13
Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony <i>B. Kruekaew, and W. Kimpan</i> .....	18
An Application of Fuzzy Inference System Composed of Double-Input	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony

B. Kruekaew and W. Kimpan

**Abstract**—Resource scheduling management design on Cloud computing is an important problem. Scheduling model, cost, quality of service, time, and conditions of the request for access to services are factors to be focused. A good task scheduler should adapt its scheduling strategy to the changing environment and load balancing Cloud task scheduling policy. Therefore, in this paper, Artificial Bee Colony (ABC) is applied to optimize the scheduling of Virtual Machine (VM) on Cloud computing. The main contribution of work is to analyze the difference of VM load balancing algorithm and to reduce the makespan of data processing time. The scheduling strategy was simulated using CloudSim tools. Experimental results indicated that the combination of the proposed ABC algorithm, scheduling based on the size of tasks, and the Longest Job First (LJF) scheduling algorithm performed a good performance scheduling strategy in changing environment and balancing work load which can reduce the makespan of data processing time.

**Index Terms**—Artificial Bee Colony, Cloud Computing, Scheduling Management, Virtualization Machine

## I. INTRODUCTION

RECENTLY, internet communication has become one of the most important factors in daily life activities. Consequently, internet is the center of data sharing. Due to the large amount of increasing users, the data is often changed or modified. This causes the heavy task scheduling for computers while less services can be processed to serve numerous users. However, this problem can be solved by using new technology of Cloud computing which focuses on networking and computing, storage, and data service resources.

Cloud computing is an emerging concept of information technology service. It consists of both infrastructure as well as the application services and focuses on types of users' requirements. The users can identify their needs through the software in Cloud.

Cloud computing [1], [2] is a terminology used in describing various computing concepts that involve a large number of computers connected through a real-time communication network such as the internet. Cloud computing is the result of evolution and adoption of existing technologies and paradigms. Its goal is to allow the users to

take advantages from these technologies without the requirements of deep knowledge or expertise. The Cloud aims to reduce costs and to provide the ease of resource management. Virtualization is the principle technology for Cloud computing. It provides the agility in operations, and reduces cost by increasing infrastructure utilization. On the other hand, Cloud computing automates the process through which the users can utilize resources on-demand.

Virtualization [3], [4] is critical to Cloud computing because it delivers services by providing a platform for optimizing complex Information Technology resources in a scalable manner. The main aim of the virtualization is an ability to run the multiple VMs on a single machine by sharing all the resources that belong to the hardware.

The problem of load balancing services occurs when the services from the users make a request to access to the same server while other servers have no request from the services. This phenomenon is called distributed load imbalance system. In this situation, it can be solved by scheduling the tasks or the services before using the system. Therefore, a good task scheduler can increase the performance of resource utilization and can reduce the makespan of assigned tasks which is called distributed load balance system.

Task scheduling in distribution system such as Cloud is used for balancing work load. It requires some conditions for example, stability of the system, makespan of work, ability to adapt to the environment changing, and etc. Those conditions in task management are similar to the behaviors of Artificial Bee Colony algorithm (ABC algorithm). ABC is inspired by the behavior of a honey bee colony in nectar collection. Bee colony based on the social behavior has the ability to adapt to the environment changing.

The suggestion for enhancing the performance of cloud task scheduling is to reduce the makespan of a given task set and increase performance of resource utilization by using improved ABC algorithm. It can be used to solve the problem and the appropriate value can be found. Moreover, it has the ability to adapt to the environment changing [5]-[7].

The structure of this paper is organized as follows: the section II presents an overview of the related work. The Artificial Bee Colony is presented in section III. Section IV presents Artificial Bee Colony inspired scheduling and load balancing algorithm. Section V is the experimental results and discussion. Finally the conclusion and future work are presented in section VI.

Manuscript received November 21, 2013; revised December 3, 2013.

B. Kruekaew is with the Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang (KMITL), Chalokkrung Rd., Ladkrabang, Bangkok, 10520, Thailand (e-mail: s3650853@kmitl.ac.th).

W. Kimpan is with the Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang (KMITL), Chalokkrung Rd., Ladkrabang, Bangkok, 10520, Thailand (e-mail: knwarang@kmitl.ac.th).

## II. RELATED WORK

Scheduling and load balancing techniques are crucial for implementing the efficient parallel and distributed applications in making the appropriate use of parallel and distributed system. Task scheduling can increase the efficiency of Cloud computing if it is able to work under the reasonable resources [4], [8], [9].

Li *et al.* [10] proposed the tree structure method in order to solve Job-Shop Scheduling in supporting the needs of a complex and flexibility to Job-Shop Scheduling problem. It also appropriately managed the specific features of model management and scheduling algorithm by using Reusable Scheme and Multi-Agent system. Later, Fang *et al.* [11] presented task scheduling in Cloud computing considering the requirements of the users and load balance in Cloud environment when the users increase and change the environment. The scheduling strategy was simulated by using CloudSim toolkit package.

Calheiros *et al.* [12], [13] developed the CloudSim simulation for modeling and simulation of virtualized Cloud-based datacenter environments, including dedicated management interface for VMs, memory, storage, and bandwidth. Simulation-based approaches can evaluate Cloud computing system and application behaviors offering significant benefits.

Mondal *et al.* [14] proposed troubleshoot of load balance in Cloud computing using Stochastic Hill Climbing. In 2011, Hsu *et al.* [9] presented a focus on energy efficiency in datacenter by considering the task scheduling to physical server and reducing energy in the system. The criterion used to measure the performance of an energy loss was to prepare computer exceeds the requirement of the job. Experiment strategies for allocating server to a sequence of jobs were a largest machine first heuristic, a best fit method, and a mixed method. The experiment results indicated that all three algorithms waste less energy consumption in over-provision.

In 2010, Hu *et al.* [15] introduced the scheduling strategy on load balancing of VM resource in Cloud computing environment by adopting a tree structure using Genetic algorithm for scheduling. It considered previous data and the current state of work in advance to the performance behavior of the system which can solve the problem of load imbalance in Cloud computing.

In 2012, Wei *et al.* [16] presented Genetic algorithm for scheduling in Cloud computing to increase the system performance. Li *et al.* [17] proposed a Load Balancing Ant Colony Optimization (LBACO) Algorithm. ACO was used to schedule on load balancing and reduce makespan in Cloud.

Karaboga *et al.*, in 2012 [7], presented ABC algorithm which was used to solve the problem and find the most appropriate value within environment changing. ABC algorithm based on the swarm-based optimization algorithm and it is used to solve the optimization problems in searching, electrical engineering, data mining, and etc. In the same year, Bitam *et al.* [18] proposed Bee Life algorithm which was used for scheduling in Cloud computing. Bee Life algorithm is inspired by the behavior and reproduction of bee to find food source. The algorithm evaluated the performance of the resources and it has the aim to reduce time and complexity of work.

Mizan *et al.* [19] presented job scheduling in Hybrid cloud by modifying Bee Life algorithm and Greedy algorithm to achieve an affirmative response from the end users and utilize the resources.

## III. ARTIFICIAL BEE COLONY ALGORITHM

Artificial Bee Colony (ABC) algorithm was proposed by Karaboga [5], [6], [20], [21]. It is the method to find the appropriate value. This method is inspired by the foraging behavior of honey bees. In ABC model, there are three kinds of honey bee to search food sources, which include scout bees search for food source randomly, employed bees search around the food source and share food information to the onlooker bees, and onlooker bees calculate the fitness and select the best food source. In the nature, bees can extend themselves over long distances in multiple directions. After scout bees find the food source and return to the hive, they compare the quality of food source and go to the dance floor to perform a dance known as “waggle dance”. The waggle dance is the communication of bees to shares the information about direction of the food source, distance from the hive, and the nectar amount of the food source. While sharing information, bees evaluate the nectar quality and energy waste. After sharing information on the dance floor, onlooker bees select the best food source and then scout bees will return to the food source to bring nectar back to the hive. A pseudo code of ABC algorithm is described in Fig. 1.

---

### Begin

Initialize the population of the scout bees, generate randomly scout bees into the food sources and calculate the fitness values.

### Repeat

- Each the employed bees search around the food sources and update the new fitness, if the new fitness is better than the old values.
- Select employed bees and recruit onlooker bees to search around the food sources and calculate on their fitness values.
- Choose the onlooker bees with have the best fitness value.
- Send scout bees into the food sources to discover new food sources.

Until (Stopping criterion is not met)

---

### End

Fig. 1. Basic of ABC algorithm.

## IV. ARTIFICIAL BEE COLONY INSPIRED SCHEDULING AND LOAD BALANCING ALGORITHM

Cloud computing provides a dynamic resource pool of VM according to different requirements from the users or the system. The routing of services which request to the diverse servers, depend on the Cloud management policies based on load of individual server. Scheduling management on Cloud computing is different degrees of loads on each and every VM. This may lead to different load among the VMs. Load balancing techniques are effective in reducing the makespan and response time.

Let  $VM = \{VM_1, VM_2, VM_3, \dots, VM_N\}$  is a set of  $N$  virtual machines and  $Task = \{task_1, task_2, task_3, \dots, task_K\}$  is a set of  $K$  task to be scheduled and processed in  $VM$ . All the machines are unrelated but are paralleled. Scheduling is non-preemptive which means that the processing of the tasks on VMs cannot be interrupted. The flowchart of the VM scheduling and load balancing using ABC algorithm is shown in Fig. 2.

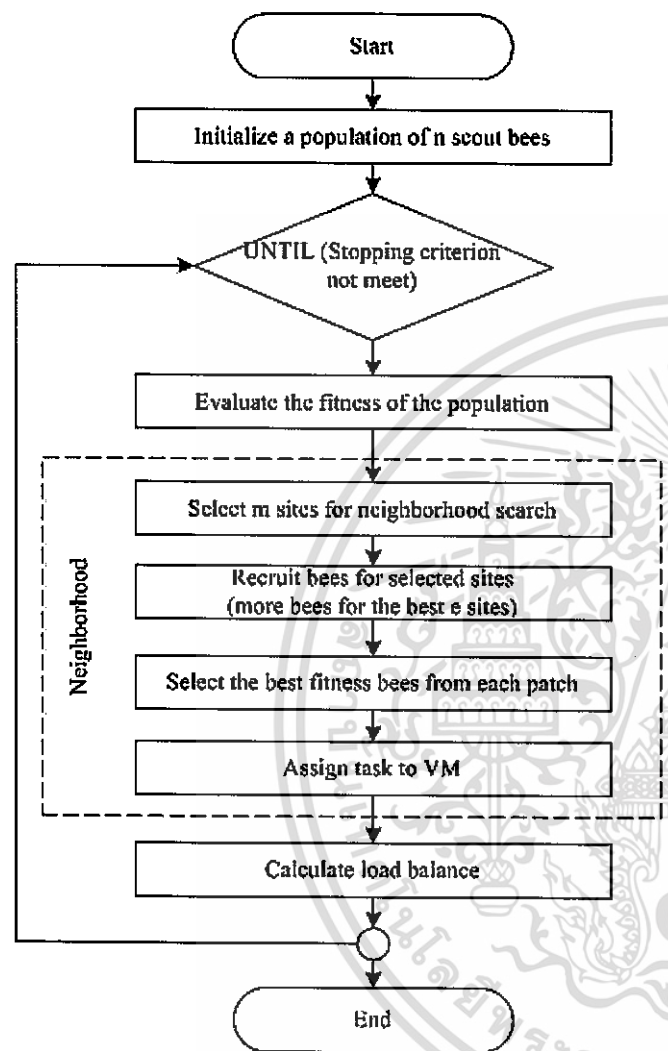


Fig. 2. Flowchart of the VM scheduling and load balancing using ABC algorithm.

**A. Initialize Population**

At the beginning, the initial  $n$  scout bees are placed randomly in  $VMs$  on Cloud computing and  $n$  is the number of scout bees.

**B. Evaluate the Fitness of the Population**

Fitness is calculated based on (1).

$$fit_{ij} = \frac{\sum_{i=1}^n task\_length_i}{Evaluate\ capacity\ of\ VM_j\ (capacity_j)} \quad (1)$$

Where  $fit_{ij}$  is the fitness of the bees population of  $i$  in  $VM_j$  or capacity of  $VM_j$  with bee number of  $i$ .  $task\_length$  is the length of the task that has been submitted in  $VM_j$  and  $capacity_j$  is the capacity of  $VM_j$  calculating based on (2).

$$capacity_j = pe\_num_j \times pe\_mips_j + vm\_bw_j \quad (2)$$

Where  $capacity_j$  is a capacity of  $VM_j$ ,  $pe\_num_j$  (processing element) is the number processor in  $VM_j$ ,  $pe\_mips_j$  is million instructions per second of each processor in  $VM_j$ , and  $vm\_bw_j$  is the network bandwidth ability of  $VM_j$ .

**C. Select m Sites for Neighborhood Search**

Scout bees with the highest fitness are chosen as "Selected Bee" and the sites visiting by them are chosen from neighborhood of  $m$  VMs.

**D. Recruit Bees for Selected Site**

Send more bees to neighborhood of the best  $e$  VM, then evaluate the fitness based on (3).

$$fit_{ij} = \frac{\sum_{i=1}^n task\_length_i + InputFile\_length}{Evaluate\ capacity\ of\ VM_j\ (capacity_j)} \quad (3)$$

Where  $InputFile\_length$  is the length of the task before execution.

**E. Select the Best Fitness Bees from Each Patch and Assign Task to  $VM_j$**

For each round, the bee with the best fitness will be chosen to assign task in  $VM_j$ .

**F. Calculate Load Balance**

After submitting tasks to the under loaded  $VM_j$ , the current workload of all available VMs can be calculated by using the information that received from the datacenter. Thus, Standard Deviation (S.D.) is calculated in order to measure the deviations of load on VMs. Standard deviation of load can be defined as (4):

$$S.D. = \sqrt{\frac{1}{n} \sum_{j=0}^n (X_j - \bar{X})^2} \quad (4)$$

Processing time of a VM:

$$X_j = \frac{\sum_{i=1}^k task\_length_i}{capacity_j} \quad (5)$$

Mean of processing time of all VMs:

$$\bar{X} = \frac{\sum_{j=1}^n X_j}{n} \quad (6)$$

Where  $S.D.$  is Standard deviation of load,  $n$  is number of all VM.  $X_j$  is processing time of  $VM_j$  which can be calculated based on (5) and  $\bar{X}$  is mean processing time of  $N$  virtual machines which can be calculated based on (6).

If the S.D. of the loaded VM is less than or equal to the mean, then the system is in a balance state. On the other hand, if the S.D. is higher than the mean, then the system is in an imbalance state.

V. EXPERIMENTAL RESULTS

A. Implementation Environment

According to the algorithm described above, the simulation using CloudSim-3.0.1 Tools will be addressed. The experiments consist of 10 datacenters and 100-700 tasks under the simulation platform. The parameters setting on the CloudSim-3.0.1 is shown in Table I.

TABLE I  
PARAMETERS SETTING OF CLOUD SIMULATOR

Type	Parameter	Value	
Datacenter	Number of Datacenter	10	
	Number of Host	5	
	Type of Manager	Space_shared, Time_shared	
	Number of PE per Host	2 - 4	
	Bandwidth	2000	
	Host Memory (MB <sup>a</sup> )	2048 - 10240	
	Datacenter Cost (The cost of using processing in this resource)	10	
	Virtual Machine (VM)	Total number of VMs	30 - 210
		MIPS <sup>b</sup> of PE <sup>c</sup>	1000 - 2000
		Number of PE per VM	1
VM Memory (MB)		512 - 2048	
Bandwidth (Bit)		1000	
Type of Manager		Time_shared	
Task		Total number of Tasks	100 - 700
	Length of Task (MI <sup>d</sup> )	5000 - 20000	
	Number of PE per requirement	1	
	Type of Manager	Space_shared	

<sup>a</sup>MB = Megabyte, <sup>b</sup>PE = Processing Element, <sup>c</sup>MIPS (Million Instructions Per Second) is a measure of the processing speed of the computer, <sup>d</sup>MI is Million Instruction.

B. Parameters Setting of ABC Algorithm

The parameters setting of improved ABC algorithm is shown in Table II.

TABLE II  
PARAMETERS SETTING OF IMPROVED ABC ALGORITHM

Symbol	Parameter	Value
n	Number of scout bees	1000
m	Number of sites selected out of n visited site	5
e	Number of best site out of m select site	1
nep	Number of bees recruited for best e site	800
nsp	Number of bees recruited for other (m-e) selected sites	200

C. Experimental Results

The evaluation of the performance of the proposed method can be described and compared with the scheduling algorithm as the followings:

- First Come First Serve (FCFS) is considering the sequence of arrival of the job.
- Shortest Job First (SJF) is considering the size of the job by selecting the shortest job first.
- Longest Job First (LJF) is considering the size of the job by selecting the longest job first.

There are two experiments; the first experiment is the comparison of the average makespan with the number of requests changing, the second phase is the comparison of the average makespan with the number of VMs changing.

In the first experiment, the comparison of the average makespan with the number of requests changing from the different scheduling algorithm is performed. The number of VMs is fixed as 50 VMs and the number of requests gradually increased between 100 – 700 tasks. The x axis shows the effect on performance of increased requests as shown in Fig. 3.

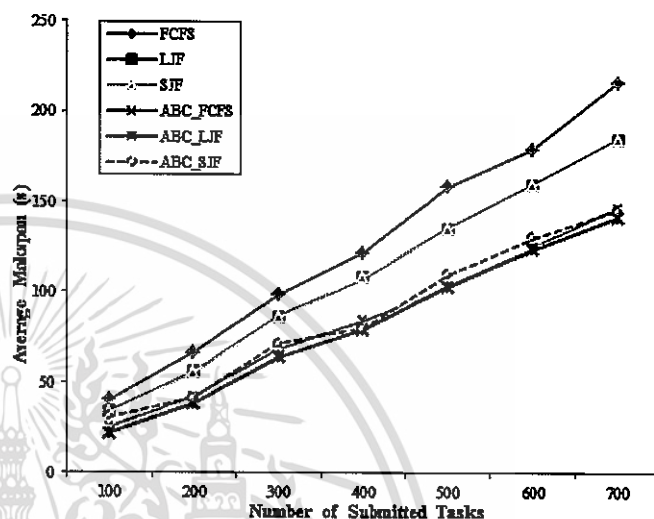


Fig. 3. Comparison of average makespan among FCFS, LJF, SJF, ABC\_FCFS, ABC\_LJF, and ABC\_SJF algorithm based on the fixed Number of VMs and the increased number of the requests.

Fig. 3 shows the average makespan of the scheduling using improved ABC algorithm with FCFS, LJF, and SJF (ABC\_FCFS, ABC\_LJF, and ABC\_SJF), comparing with the original scheduling (FCFS, LJF, and SJF) which was based on 50 fixed VMs and the number of increasing requests. The experimental results indicated that, while the number of requests was increasing, the average makespan consequently increased. It can be concluded that ABC\_LJF performed the effective results in the optimal scheduling on the loaded system.

In the second experiment, the comparison of the average makespan with the number of VMs changing from the different scheduling algorithm is performed. The number of tasks was fixed as 500 tasks and gradually increased the number of VMs from 30 to 210 VMs. The x axis shows the effect on performance in increasing the system size (VMs).

Fig. 4 shows the average makespan of the scheduling using ABC\_FCFS, ABC\_LJF, and ABC\_SJF comparing with FCFS, LJF, and SJF which was based on the fixed number of tasks and the number of increasing VMs. It can be concluded that ABC\_LJF performed better than all methods and its performance is more prominent in scalability.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

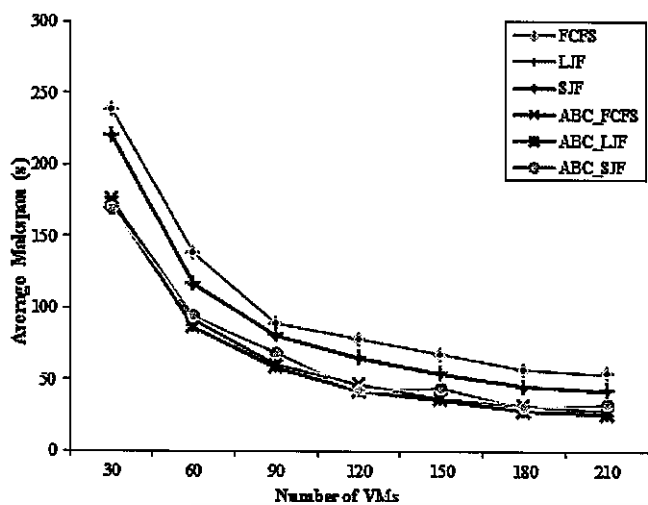


Fig. 4. Comparison of average makespan between FCFS, LJF, SJF, ABC\_FCFS, ABC\_LJF, and ABC\_SJF algorithm based on the fixed number of the requests and the increased number of VMs.

## VI. CONCLUSION AND FUTURE WORK

This paper presents ABC optimization algorithm which can solve the Virtual machine scheduling management under the environmental changing of the number of VMs and requests on Cloud computing. Even in changing environment, Cloud computing needs to be operated in a stable system. Therefore, ABC algorithm is suitable for Cloud computing environment because the algorithm is able to effectively utilize the increased system resources and reduce makespan. The experimental results illustrated that the proposed methods of ABC\_LJF performed effective results than all methods and its performance is more prominent in scalability. Under the circumstance of increasing or decreasing the number of servers, the load balancing algorithm should be done by using ABC\_LJF algorithm in order to maintain system stability and scheduling and to prevent the system crash. For further studies, the preemptive Virtual machine scheduler operating with independent and heterogeneous tasks on Cloud computing will be focused.

## REFERENCES

- [1] K. Tang, J. M. Zhang, and C. H. Feng, "Application centric lifecycle framework in cloud," *IEEE 8th Int. Conf. on e-Business Engineering (ICEBE)*, 2011, 19-21 Oct. 2011, pp. 329-334.
- [2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," in *Communications of the ACM*, vol. 53, no. 4, April 2010, pp. 50-58.
- [3] K. Dutta, "A smart job scheduling system for cloud computing service providers and users: modeling and simulation," in *1st Int'l Conf. on Recent Advances in Information Technology RAIT-2012*, 2012.
- [4] S. Sindhu, and S. Mukherjee, "Efficient task scheduling algorithms for cloud computing environment," in *High Performance Architecture and Grid Computing, Communications in Computer and Information Science*, 2011, vol. 169, 2011, pp. 79-83.
- [5] D. Karaboga, and B. Basturk, "Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems," in *IFSA 2007, LNAI 4529*, 2007, pp. 789-798.
- [6] D. Karaboga, and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," in *Applied Soft Computing*, 2008, pp. 687-697.
- [7] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and

- applications," in *Artificial Intelligence Review 2012, Springer Science Business Media B.V.* 2012, March 2012.
- [8] S. T. Maguluri, R. Srikant, and L. Ying, "Stochastic models of load balancing and scheduling in cloud computing clusters," in *Proc. IEEE Infocom*, 2012, pp. 702-710.
- [9] Y. C. Hsu, P. Liu, and J. J. Wu, "Job sequence scheduling for cloud computing," in *Int. Conf. on Cloud and Service Computing (CSC 2011)*, 2011, Dec. 2011, pp. 212-219.
- [10] N. Li, Z. Hong, and H. Xiaoting, "Dynamic integration mechanism for job-shop scheduling model base using Multi-agent," in *2009 Int. Conf. on Information Management, Innovation Management and Industrial Engineering*, 2009, vol. 4, Dec. 2009, pp. 421-425.
- [11] Y. Fang, F. Wang, and J. Ge, "A task scheduling algorithm based on load balancing in cloud computing," in *Web Information Systems and Mining, Lecture Notes in Computer Science 2010*, Vol. 6318, 2010, pp. 271-277.
- [12] R. N. Calheiros, R. Ranjan, C. A. F. D. Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, Vol. 41, No.1, Jan. 2011, pp. 23-50.
- [13] R. N. Calheiros, R. Ranjan, C. A. F. D. Rose, and R. Buyya, "CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services," *arXiv preprint arXiv: 0903.2525*, 2009.
- [14] B. Mondal, K. Dasgupta, and P. Dutta, "Load balancing in cloud computing using Stochastic Hill Climbing-A soft computing approach," in *Procedia Technology*, vol. 4, 2011, pp. 783-789.
- [15] J. Hu, J. Gu, G. Sun, and T. Zhao, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in *3rd Int. Symp. on Parallel Architectures, Algorithms and Programming (PAAP)*, 2010, 18-20 Dec. 2010, pp. 89-96.
- [16] Y. Wei, and L. Tian, "Research on cloud design resources scheduling based on genetic algorithm," in *2012 Int. Conf. on Systems and Informatics (ICSAI 2012)*, 2012, May 2012, pp. 2651-2656.
- [17] K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, "Cloud task scheduling based on load balancing Ant Colony Optimization," in *6th Annual ChinaGrid Conf.*, 2011, Aug. 2011, pp. 3-9.
- [18] S. Bitam, "Bees life algorithm for job scheduling in cloud computing," in *Conf. on Computing and Information Technology (ICCIT 2012)*, 2012, pp. 186-191.
- [19] T. Mizan, S. M. R. A. Masud, and R. Latip, "Modified bees life algorithm for job scheduling in hybrid cloud," in *Int. Journal of Engineering and Technology (IJET)*, 2012, vol. 2, no.6, June 2012, pp. 974-979.
- [20] D. Karaboga, "An Idea based on honey bee swarm for numerical optimization," in *Technical Report TR06*, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [21] D.T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, "The Bees Algorithm-A novel tool for complex optimisation problems," in *Proc. of IPROMS 2006 Conf.*, 2006, pp. 454-461.

การศึกษานี้เป็นการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



# THE 10<sup>TH</sup> NATIONAL CONFERENCE ON COMPUTING AND INFORMATION TECHNOLOGY

PROCEEDINGS OF NCCIT 2014

THE 10<sup>TH</sup> NATIONAL CONFERENCE ON COMPUTING AND INFORMATION TECHNOLOGY

8<sup>TH</sup>-9<sup>TH</sup> MAY 2014

ANGSANA LAGUNA PHUKET, THAILAND

[WWW.NCCIT.NET](http://WWW.NCCIT.NET)

FACULTY OF INFORMATION TECHNOLOGY

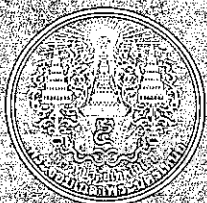
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

บทความวิจัย

การประชุมวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ  
ครั้งที่ 10

8-9 พฤษภาคม 2557

โรงแรมอังกูรา ลากูน ภูเก็ต



คณะเทคโนโลยีสารสนเทศ

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

# การจัดตารางงานบนพื้นฐานของการกระจายงานสมดุลในการประมวลผล แบบกลุ่มเมฆโดยขั้นตอนวิธีอาณานิคมผึ้งเทียม Task Scheduling Based on Load Balancing in Cloud Computing Using Artificial Bee Colony Algorithm

บุญหทัย เครือแก้ว (Boonhatai Kruekaew)<sup>1</sup> และวารงคณา กิมปาน (Warangkha Kimpan)<sup>2</sup>

สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

<sup>1</sup>bs3650853@kmitl.ac.th, <sup>2</sup>knwarang@kmitl.ac.th

## บทคัดย่อ

เครื่องคอมพิวเตอร์เสมือนถือเป็นส่วนประกอบสำคัญของเทคโนโลยีการประมวลผลแบบกลุ่มเมฆ ดังนั้นการจัดตารางงานเพื่อให้เครื่องคอมพิวเตอร์เสมือนมีการกระจายงานแบบสมดุลและระบบการประมวลผลแบบกลุ่มเมฆสามารถทำงานได้อย่างมีประสิทธิภาพจึงเป็นปัญหาที่สำคัญ บทความนี้ได้เสนอการใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก สำหรับจัดตารางงานในการเข้าใช้เครื่องคอมพิวเตอร์เสมือน ทั้งในกรณีที่เครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันและต่างกันบนการประมวลผลแบบกลุ่มเมฆ โดยมีจุดมุ่งหมายเพื่อหาขั้นตอนวิธีในการกระจายงานแบบสมดุลให้กับเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมต่างกันและใช้เวลารวมในการทำงานน้อยที่สุด การทดลองได้ทำการเปรียบเทียบเพื่อหาวิธีการที่เหมาะสมสำหรับการจัดตารางงานในการเข้าใช้เครื่องคอมพิวเตอร์เสมือน โดยใช้ CloudSim ในการสร้างแบบจำลอง

**คำสำคัญ:** การประมวลผลแบบกลุ่มเมฆ การกระจายงานแบบสมดุล การจัดตารางงาน ขั้นตอนวิธีอาณานิคมผึ้งเทียม วิธีการจัดลำดับงานแบบฮิวริสติก

## Abstract

Virtual Machine is a key component of Cloud computing Technology. Therefore, the developing an optimal scheduling mechanism for balancing Virtual Machines operations at Cloud computing framework is an intriguing issue for Cloud computing services performance. This paper proposed Artificial Bee Colony

(ABC) algorithm and Heuristic scheduling algorithms which are applied to optimize the scheduling tasks for Virtual Machines on homogeneous and heterogeneous environment platforms on Cloud computing. The main contribution of this work is to analyze the difference of Virtual Machines load balancing algorithms. The experimental results demonstrated the comparison of many algorithms to find the suitable one for task scheduling on Virtual Machines and to reduce the makespan of data processing time. The scheduling strategy was simulated by using CloudSim tool.

**Keyword:** Cloud Computing, Load Balance, Scheduling, Artificial Bee Colony Algorithm, Heuristics.

## 1. บทนำ

การประมวลผลแบบกลุ่มเมฆ (Cloud Computing) [1] ถือเป็นระบบคอมพิวเตอร์ที่มีการประมวลผลในระบบอินเทอร์เน็ต มีรูปแบบการประมวลผลที่อิงตามความต้องการของผู้ใช้ ผู้ใช้สามารถระบุความต้องการไปยังซอฟต์แวร์ของระบบ จากนั้นซอฟต์แวร์จะมีการร้องขอให้ระบบจัดสรรทรัพยากรและบริการให้ตรงกับความต้องการของผู้ใช้ นอกจากนี้ผู้ใช้งานสามารถเพิ่มและลดจำนวนทรัพยากรได้ตามความต้องการ โดยไม่จำเป็นต้องทราบถึงรูปแบบการทำงานหรือการให้บริการ ข้อดีของการประมวลผลแบบกลุ่มเมฆ คือ การลดความซับซ้อนของผู้ให้บริการ ประหยัดพลังงานและลดค่าใช้จ่ายในการจัดการ เพราะระบบจะทำงานผ่านเทคโนโลยีเสมือน (Virtualization Technology) ส่งผลให้ระบบจะไม่ถูกจำกัดในเรื่องของขีดความสามารถในการประมวลผลไปใช้ประโยชน์ด้านการค้า

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ การเผยแพร่โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคโนโลยีเสมือนเป็นการจำลองการสร้างเครื่องคอมพิวเตอร์เสมือน (Virtualization Machine: VM) [2,3] ด้วยซอฟต์แวร์ที่ทำให้คอมพิวเตอร์เครื่องหนึ่งสามารถทำงานเสมือนมีหลายระบบ โดยแต่ละเครื่องคอมพิวเตอร์เสมือนจะมีทรัพยากรที่เป็นอิสระต่อกัน ซึ่งถือเป็นการลดปริมาณเครื่องคอมพิวเตอร์ที่ใช้เป็นเครื่องแม่ข่าย (Server) และเป็นการใช้ทรัพยากรอย่างคุ้มค่า

ในกรณีที่มีการร้องขอเข้าใช้บริการในระบบไปยังเครื่องแม่ข่ายเดียวกัน ในบางครั้งเครื่องแม่ข่ายอื่นอาจไม่มีการเข้าใช้บริการ ซึ่งเรียกได้ว่าระบบเกิดการกระจายงานแบบไม่สมดุล (Load Imbalance) สามารถแก้ไขได้โดยการจัดการตารางงาน (Scheduling) ก่อนเข้าใช้บริการ ซึ่งการจัดการตารางงานที่คิ่นั้นจะมีประโยชน์ในแง่ของการเพิ่มประสิทธิภาพการใช้ทรัพยากรคือ ลดเวลาขอเข้าทำงานในระบบและระบบจะมีการกระจายการทำงานแบบสมดุล (Load Balance)

การจัดการตารางงานในระบบประมวลผลแบบกลุ่มเมฆ เพื่อให้ระบบเกิดการกระจายงานแบบสมดุลนั้น จำเป็นต้องคำนึงถึงเงื่อนไขต่างๆ เช่น ความเสถียรของระบบ ความรวดเร็วในการทำงาน การเปลี่ยนแปลงของสภาพแวดล้อมในระบบ เป็นต้น เงื่อนไขที่ต้องคำนึงถึงในการจัดการตารางงานนั้นจะคล้ายคลึงกับการทำงานของขั้นตอนวิธีอาณานิคมผึ้งเทียม (Artificial Bee Colony Algorithm: ABC Algorithm) ส่วนของพฤติกรรมการหาอาหารของผึ้ง เนื่องจากผึ้งสามารถปรับตัวได้ตามสภาพแวดล้อมที่อาศัยอยู่ [4,5] การประมวลผลแบบกลุ่มเมฆมีความคล้ายคลึงกับขั้นตอนวิธีอาณานิคมผึ้งเทียม ในส่วนของเป้าหมายของการทำงาน เนื่องจากระบบการประมวลผลแบบกลุ่มเมฆต้องการให้เครื่องคอมพิวเตอร์สามารถประมวลผลงานให้ได้ปริมาณมากที่สุด ในขณะที่ผึ้งต้องการแหล่งอาหารที่มีปริมาณน้ำหวานมากที่สุดเช่นกัน

ดังนั้นงานวิจัยนี้ได้นำเสนอขั้นตอนวิธีการจัดการตารางงานเพื่อเข้าใช้เครื่องคอมพิวเตอร์เสมือน ในกรณีที่เครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันและต่างกัน (Homogeneous and Heterogeneous) โดยประยุกต์ใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับการทำงานแบบฮิวริสติก (Heuristics) [6] เพื่อหาเวลารวมน้อยที่สุดในการทำงาน (Processing Time) และทำให้ระบบอยู่ในสถานะสมดุล

## 2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

### 2.1 การประมวลผลแบบกลุ่มเมฆ

การประมวลผลแบบกลุ่มเมฆ มีรูปแบบการให้บริการที่นำเสนอทรัพยากรสารสนเทศ สำหรับการประมวลผลตั้งแต่โครงสร้างพื้นฐานไปจนถึงชุดคำสั่งประยุกต์ เสมือนหนึ่งเป็นบริการที่ผู้ใช้สามารถบริการตนเองได้ โดยเลือกใช้เฉพาะสิ่งที่ตนต้องการ ผู้ใช้บริการระบบการประมวลผลแบบกลุ่มเมฆจะมองกลุ่มเมฆแตกต่างกันออกไป เนื่องจากบริการที่นำเสนอในแต่ละผู้ให้บริการนั้นมีความแตกต่างกันออกไป รวมทั้งผู้ให้บริการสามารถเลือกใช้บริการเฉพาะที่ตนต้องการได้ [1]

การให้บริการบนระบบการประมวลผลแบบกลุ่มเมฆสามารถแบ่งรูปแบบของการให้บริการเป็น 3 รูปแบบ [4] คือ

1) การบริการ โครงสร้างพื้นฐานของคอมพิวเตอร์ (Infrastructure as a Service: IaaS) มีประโยชน์ในการประมวลผลทรัพยากรจำนวนมาก ผู้ให้บริการโครงสร้างพื้นฐาน เช่น Amazon's EC2 เป็นต้น

2) การบริการแพลตฟอร์ม (Platform as a Service: PaaS) เป็นการประมวลผลซึ่งจะมีระบบปฏิบัติการและสนับสนุนเว็บแอปพลิเคชันเข้าร่วมด้วย เช่น Google Apps เป็นต้น

3) การบริการซอฟต์แวร์ (Software as a Service: SaaS) จะให้บริการการประมวลผลผ่านแอปพลิเคชันแม่ข่ายของผู้ให้บริการและเปิดให้บริการทางด้านซอฟต์แวร์ต่างๆ โดยผู้ใช้งานสามารถเรียกใช้งานซอฟต์แวร์ผ่านระบบอินเทอร์เน็ต

ในระบบการประมวลผลแบบกลุ่มเมฆนั้นจะมีการจัดสรรทรัพยากรของระบบแตกต่างกัน ขึ้นอยู่กับความต้องการของระบบ กรณีที่มีการจัดสรรทรัพยากรเหมือนกันทั้งระบบ จะเรียกสภาพแวดล้อมภายในระบบนี้ว่า Homogeneous และในกรณีที่มีการจัดสรรทรัพยากรในระบบแตกต่างกัน จะเรียกสภาพแวดล้อมนี้ว่า Heterogeneous

ตลอดระยะเวลาหลายปีที่ผ่านมาได้มีผู้เสนอแนะขั้นตอนวิธีการจัดการตารางงานเข้าใช้ทรัพยากรในระบบการประมวลผลแบบกลุ่มเมฆด้วยวิธีการต่างๆ เพื่อให้ระบบมีการกระจายการทำงานแบบสมดุลและตรงกับความต้องการของผู้ใช้ ภายใต้ทรัพยากรที่มีอยู่อย่างจำกัด และเพื่อเป็นการเพิ่มประสิทธิภาพการทำงานของระบบการประมวลผลแบบกลุ่มเมฆ [3] เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปี 2010 Y. Fang, F. Wang และ J. Ge [7] ได้เสนอการจัดตารางการทำงานในการประมวลผลแบบกลุ่มเมฆ โดยคำนึงถึงความต้องการของผู้ใช้งานที่มีการเปลี่ยนแปลงตลอดเวลาและคำนึงถึงการกระจายงานแบบสมดุลในสภาพแวดล้อมของการประมวลผลแบบกลุ่มเมฆ เมื่อผู้ใช้งานและความต้องการในการใช้งานทรัพยากรมีปริมาณเพิ่มมากขึ้น โดยใช้ CloudSim [8] ซึ่งเป็นชุดเครื่องมือในการสร้างแบบจำลองของระบบการประมวลผลแบบกลุ่มเมฆ สามารถใช้ทดสอบประสิทธิภาพของระบบที่พัฒนาขึ้นใหม่ เป็นเครื่องมือที่สะดวกในการกำหนดสภาพแวดล้อมตามที่ผู้ใช้ต้องการ เช่น การตั้งค่าของดาต้าเซ็นเตอร์ (Data Center) และเครื่องคอมพิวเตอร์เสมือน เป็นต้น

จากนั้นในปี 2011 B. Mondal และ K. Dasgupta [9] ได้มีการนำขั้นตอนการแก้ไขปัญหาค้นหาของสโตนคลิมบิง (Stochastic Hill Climbing) มาใช้ในการจัดการงานเพื่อให้มีการกระจายงานแบบสมดุลในการประมวลผลแบบกลุ่มเมฆ

## 2.2 การจัดลำดับงานแบบฮิลลิสติก

วิธีนี้เป็นวิธีการนำกฎต่างๆ มาใช้ในการหาผลลัพธ์ที่น่าพอใจของปัญหา เช่น กฎการจัดลำดับความสำคัญ วิธีการสุ่ม เป็นต้น วิธีการนี้อาจได้ผลลัพธ์ที่น่าพอใจ แต่ไม่สามารถรับรองได้ว่าเป็นผลลัพธ์ที่ดีที่สุด ในบทความนี้ได้นำกฎการจัดลำดับงานโดยใช้ลำดับความสำคัญเป็นเกณฑ์ในการเลือกขั้นตอนการทำงาน [6] ได้แก่

- 1) การจัดให้งานมาก่อนทำก่อน (First Come First Serve: FCFS) เป็นการจัดลำดับงานโดยทำงานที่เข้ามาก่อนเป็นอันดับแรกและทำงานที่เข้ามาหลังเป็นลำดับถัดไป
- 2) การจัดให้งานที่มีขนาดเล็กที่สุดทำงานก่อน (Shortest Job First: SJF) เป็นการจัดลำดับงานโดยทำงานที่มีขนาดเล็กที่สุดก่อน และเลือกงานที่มีขนาดใหญ่เป็นลำดับถัดไป
- 3) การจัดให้งานที่มีขนาดใหญ่ที่สุดทำก่อน (Longest Job First: LJF) เป็นการจัดลำดับงานโดยทำงานที่มีขนาดใหญ่ที่สุดก่อน และเลือกงานที่มีขนาดงานเล็กเป็นลำดับถัดไป

## 2.3 ขั้นตอนวิธีอาณานิคมผึ้งเทียม

ขั้นตอนวิธีอาณานิคมผึ้งเทียม [4,5] เป็นขบวนการในการหาค่าที่เหมาะสม โดยเลียนแบบจากพฤติกรรมในการหาอาหารของฝูงผึ้ง และเป็นกระบวนการหาคำตอบประเภทที่อยู่ในกลุ่ม

ที่มีแนวคิดมาจากการเลียนแบบพฤติกรรมของฝูงแมลง ได้มีการนำขั้นตอนวิธีอาณานิคมผึ้งเทียมไปประยุกต์ใช้ในการแก้ไขปัญหาดังกล่าว เช่น การค้นหาค่าที่เหมาะสม การทำเหมืองข้อมูล เป็นต้น

ขั้นตอนวิธีอาณานิคมผึ้งเทียมเป็นขั้นตอนวิธีที่ใช้ในการแก้ไขการหาค่าที่เหมาะสม เพื่อให้ได้คำตอบที่ต้องการ โดยการเลียนแบบพฤติกรรมการออกหาอาหารของผึ้ง ประชากรของขั้นตอนวิธีอาณานิคมผึ้งเทียมแบ่งออกเป็น 3 ประเภท คือ ผึ้งสำรวจ (Scout Bees) มีหน้าที่ในการสุ่มสำรวจหาแหล่งอาหารแหล่งใหม่ ผึ้งงาน (Employed Bees) มีหน้าที่ออกหาแหล่งอาหารแล้วกลับมาให้ข้อมูลเกี่ยวกับแหล่งอาหารที่พบให้กับผึ้งเฝ้าดูที่รออยู่ที่รัง ผึ้งเฝ้าดู (Onlooker Bees) มีหน้าที่คำนวณค่าความเหมาะสม (Fitness) และเลือกแหล่งอาหารเพื่อส่งผึ้งงานไปเก็บอาหาร ถ้าแหล่งอาหารแหล่งใดถูกเก็บจนหมด ผึ้งงานที่เป็นเจ้าของแหล่งอาหารนั้นจะกลายเป็นผึ้งสำรวจ เพื่อค้นหาแหล่งอาหารแหล่งใหม่ โดยในธรรมชาติ ผึ้งจะมีการเปรียบเทียบคุณภาพของแหล่งอาหารที่ลานเต้น (Dance Floor) โดยทำการเต้นที่เรียกว่า “เต้นแกว่ง (Waggle Dance)” เพื่อแสดงข้อมูลเกี่ยวกับทิศทาง ระยะทางและปริมาณน้ำหวานของแหล่งอาหาร โดยผึ้งเฝ้าดูจะทำการเปรียบเทียบคุณภาพของแหล่งอาหารที่ดีที่สุด

หลายปีที่ผ่านมาได้มีนักวิจัยที่นำขั้นตอนวิธีเกี่ยวกับผึ้งไปประยุกต์ใช้ในการแก้ปัญหาต่างๆ เช่น ในปี 2012 S. Bitam [10] ได้เสนอการนำขั้นตอนวิธีวงจรผึ้ง (Bees Life Algorithm: BLA) ไปประยุกต์ใช้ในการจัดตารางการทำงานในการประมวลผลแบบกลุ่มเมฆ โดยเลียนแบบพฤติกรรมการสืบพันธุ์และการค้นหาแหล่งอาหารของผึ้ง เพื่อประเมินประสิทธิภาพการเข้าทำงานในแต่ละทรัพยากรของระบบการประมวลผลแบบกลุ่มเมฆ โดยมุ่งเน้นเพื่อลดเวลาในการทำงานและลดความซับซ้อนในการทำงานของระบบ และในปีเดียวกัน T. Mizan [11] ได้เสนอการจัดตารางงานในการประมวลผลแบบกลุ่มเมฆแบบผสม (Hybrid Cloud) โดยใช้ขั้นตอนวงจรผึ้งและขั้นตอนวิธีประเภทละโมภ (Greedy Algorithm) เพื่อลดเวลารวมในระบบให้น้อยลง

ดังนั้นผู้วิจัยจึงได้เสนอแนวคิดในการจัดตารางงานบนพื้นฐานของการกระจายงานแบบสมดุล และเวลารวมในการ

การศึกษานี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานน้อยที่สุด สำหรับการประมวลผลแบบกลุ่มเมฆ โดยเน้นในกรณีที่เครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันและต่างกัน โดยประยุกต์ใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติก

### 3. วิธีการดำเนินงานวิจัย

การประมวลผลแบบกลุ่มเมฆจะมีการใช้ทรัพยากรที่แตกต่างกันตามความต้องการของผู้ใช้ในระบบ อีกทั้งการร้องขอเข้าใช้บริการต้องคำนึงถึงภาระงานของเครื่องแม่ข่ายที่จะให้บริการด้วย การจัดตารางเข้าใช้ทรัพยากรที่มีประสิทธิภาพจะส่งผลให้สามารถลดเวลารวมในการทำงานในระบบและส่งผลให้เกิดการกระจายงานแบบสมดุล ได้มีการนำขั้นตอนอาณานิคมผึ้งเทียมมาประยุกต์ใช้ในการจัดตารางงานบนพื้นฐานของการกระจายงานแบบสมดุลในระบบ และลดเวลารวมในการทำงานของระบบในการประมวลผลแบบกลุ่มเมฆ [12] โดยจุดมุ่งหมายของการวิจัยครั้งนี้ เพื่อหาขั้นตอนวิธีการจัดตารางงานให้เหมาะสมกับสภาพแวดล้อมภายในของเครื่องคอมพิวเตอร์เสมือน ทั้งในกรณีที่เครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันและต่างกัน และในการจัดตารางงานนั้นจะต้องอยู่บนพื้นฐานของการกระจายงานแบบสมดุลและเวลารวมในการทำงานน้อยที่สุดในระบบการประมวลผลแบบกลุ่มเมฆ

กำหนด  $VM = \{VM_1, VM_2, VM_3, \dots, VM_N\}$ ;  $N$  คือ จำนวนของคอมพิวเตอร์เสมือน;  $Task = \{task_1, task_2, task_3, \dots, task_K\}$ ;  $K$  คือ จำนวนของงานในระบบ รูปแบบการเข้าทำงานในระบบเป็นการทำงานที่ไม่สามารถขัดจังหวะหรือแทรกกลางคันได้ (Non-Preemptive Process) ขั้นตอนวิธีจัดตารางงานเพื่อเข้าใช้เครื่องคอมพิวเตอร์เสมือน โดยใช้ขั้นตอนอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติกแสดงดังภาพที่ 1

### 4. ผลการทดลอง

#### 4.1 การตั้งค่าพารามิเตอร์สภาพแวดล้อมในการทดลอง

เครื่องมือที่ใช้ในสร้างแบบจำลองเพื่อประเมินประสิทธิภาพการทำงานของวิธีการที่นำเสนอ คือ CloudSim-3.0.1 การตั้งค่าของเครื่องคอมพิวเตอร์เสมือนแบ่งได้เป็น 2 กรณี คือ

กรณีที่ 1: ค่าพารามิเตอร์ของเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกัน คือ MIPS (Million Instructions Per Second) is a measure of the processing speed of the

#### เริ่มต้น

- กำหนดค่าที่เหมาะสม (Fitness) เริ่มต้นให้กับแต่ละ  $VM_i$  ดังสมการที่ (1)

$$Fit_i = pe\_num_i \times pe\_mips_i + vm\_bw_i \quad (1)$$

โดยที่  $Fit_i$  = ค่า Fitness ของแต่ละ VM เครื่องที่  $i$

#### ทำซ้ำ

- ส่งผึ้งสังเกตการณ์ แบบสุ่มเข้าไปยัง  $VM_i$  และคำนวณค่า Fitness ดังสมการที่ (2) และ (3)

$$Fit_i = \frac{\sum_{j=1}^n task\_length \ h_j}{Evaluate\ capacity\ of\ VM\ (Capa_i)} \quad (2)$$

โดยที่  $task\_length$  = ขนาดของงาน

$$Capa_i = pe\_num_i \times pe\_mips_i + vm\_bw_i \quad (3)$$

โดยที่  $Capa_i$  = ความสามารถในการทำงานของ VM เครื่องที่  $i$

- ผึ้งนางจู่จะเลือก VM ที่มีค่า Fitness ต่ำที่สุดจำนวน  $m$  ตัว ซึ่ง  $m$  จะกลายเป็น Select Bee และ Site ที่ Select Bee ไปเยี่ยมชมมาแล้วจะถูกเรียกเป็นพื้นที่ที่ถูกเลือก (Neighborhood Search)
- กำหนดคั้งจำนวนมาดเข้าค้นหาในพื้นที่ที่ถูกเลือก ส่งคั้งจำนวนมากกว่า  $c$  ที่เลือก จำนวนคั้งที่ส่ง  $nep$  ตัว และคำนวณค่า Fitness ดังสมการที่ (4)

$$Fit_i = \frac{\sum_{j=1}^n task\_length \ h_j + InputFile\_length \ h}{Evaluate\ capacity\ of\ VM\ (Capa_i)} \quad (4)$$

โดยที่  $InputFile\_length$  = ขนาดของงาน

- คั้งนำข้อมูล Fitness กลับมาที่รังเพื่อทำการเลือก Site ที่ดีที่สุด
- กำหนดให้ งานจำนวน 1 งานเข้าทำงานใน VM ที่ถูกเลือก
- คำนวณเพื่อคั้ง Load balance ของระบบหลังจากมีการจัดงานให้เข้าทำงานใน VM จะมีการคำนวณภาระงานทั้งหมดในปัจจุบันของ VM ทั้งหมด โดยคำนวณจากค่าส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation: S.D.) เพื่อวัดภาระงานทั้งหมดใน VM ดังสมการที่ (5) - (7)

$$S.D. = \sqrt{\frac{1}{n} \sum_{j=0}^n (X_j - \bar{X})^2} \quad (5)$$

$$X_j = \frac{\sum_{i=1}^n task\_length \ h_i}{capacity \ j} \quad (6) \quad , \quad \bar{X} = \frac{\sum_{j=1}^n X_j}{n} \quad (7)$$

โดยที่  $X_j$  = เวลารวมในการทำงานของแต่ละ VM สามารถคำนวณได้จากสมการที่ (6);  $\bar{X}$  = ค่าเฉลี่ยของเวลารวมในการทำงานของ VM ทั้งหมดในระบบสามารถคำนวณได้จากสมการที่ (7)

ถ้า  $S.D \leq \bar{X}$  แสดงว่าระบบอยู่ในสภาวะสมดุล และ

ถ้า  $S.D > \bar{X}$  แสดงว่าระบบอยู่ในสภาวะไม่สมดุล

#### จนกระทั่ง (ทุกงานเข้าทำงานใน VM)

#### จบการทำงาน

ภาพที่ 1: รหัสเทียม (Pseudocode) แสดงขั้นตอนการจัดตารางงานในระบบการประมวลผลแบบกลุ่มเมฆ โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียม

computer) of PE = 2000, Number of PE per VM = 1, RAM = 1024, Bandwidth = 1000 Bit

กรณีที่ 2: ค่าพารามิเตอร์ของเครื่องคอมพิวเตอร์เสมือนที่มี

สภาพแวดล้อมภายในต่างกัน คือ MIPS of PE = 1000- 2000,

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Number of PE per VM = 1- 8, RAM = 512 - 6144,

Bandwidth = 1000 - 2000 Bit

การตั้งค่าพารามิเตอร์แบบจำลองเพื่อประเมินประสิทธิภาพของระบบการประมวลผลแบบกลุ่มเมฆแสดงดังตารางที่ 1

ตารางที่ 1: ค่าพารามิเตอร์ของแบบจำลองการประมวลผลแบบกลุ่มเมฆ

Type	Parameter	Value
Data Center	Number of Data Center	10
	Number of Host	5
	Type of Manager	Space_shared, Time_shared
	Number of PE <sup>a</sup> per Host	4
	Bandwidth	2000
	Host Memory (RAM) (MB <sup>b</sup> )	10240
	Data Center Cost (The cost of using processing in this resource)	10
Virtual Machine (VM)	Total number of VM	50
	Type of Manager	Time_shared
Task	Total number of Task	100 - 700
	Length of Task (MI <sup>c</sup> )	5000 - 20000
	Number of PE per requirement	1
	Type of Manager	10

PE<sup>a</sup> = Processing Element, MB<sup>b</sup> = Megabyte, MI<sup>c</sup> is Million Instruction.

#### 4.2 การตั้งค่าพารามิเตอร์ในขั้นตอนวิธีอาณานิคมผึ้งเทียม

การตั้งค่าพารามิเตอร์ที่ใช้ในส่วนของขั้นตอนวิธีอาณานิคมผึ้งเทียม แสดงค่าดังตารางที่ 2

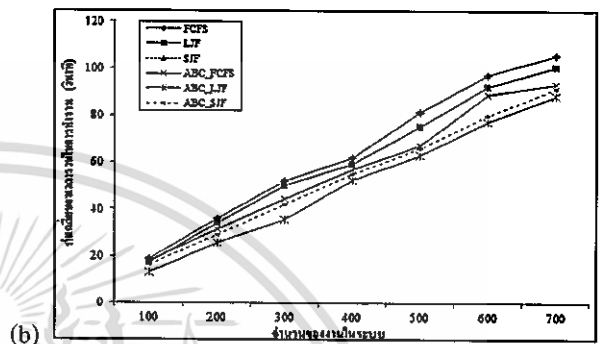
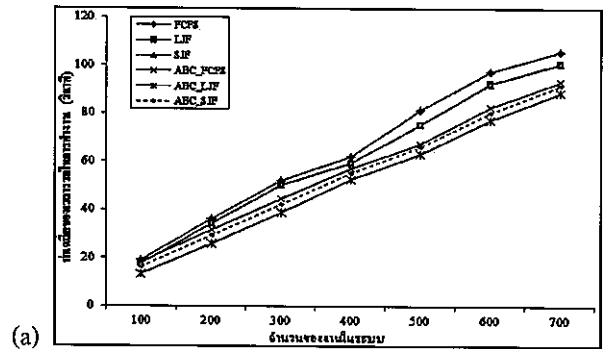
ตารางที่ 2: ค่าพารามิเตอร์ของขั้นตอนวิธีอาณานิคมผึ้งเทียม

Symbol	Parameter	Case I	Case II
n	Number of scout bees	2000	1500
m	Number of sites selected out of n visited site	5	10
e	Number of best site out of m select site	1	1
nep	Number of bees recruited for best e site	1600	1000
nsp	Number of bees recruited for other (m-e) selected sites	400	500

#### 4.3 ผลการทดลอง

การประเมินประสิทธิภาพของวิธีที่นำเสนอ (ABC\_FCFS, ABC\_SJF และ ABC\_LJF) จะทำการเปรียบเทียบกับการจัดลำดับงานแบบฮิวริสติก (FCFS, SJF และ LJF) โดยแบ่งการทดลองออกเป็น 2 กรณี คือ

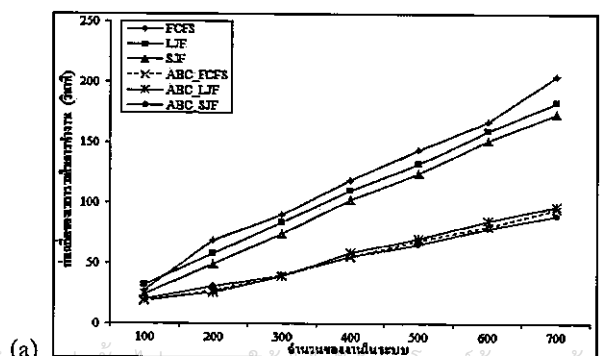
กรณีที่ 1 : ทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในเหมือนกันทั้งหมด เมื่อจำนวนของ VM มีค่าคงที่ โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ย (Average Processing Time) ของงานในระบบ



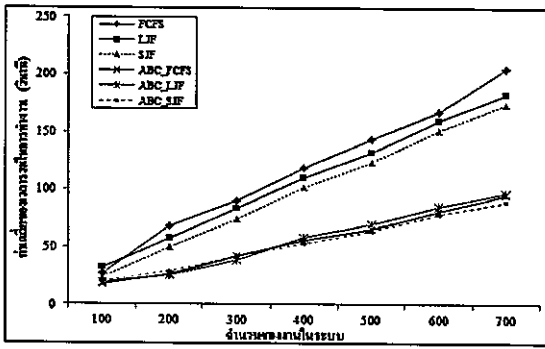
ภาพที่ 2: เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและ VM มีสภาพแวดล้อมภายในเหมือนกัน โดยที่ (a) พารามิเตอร์ ABC Case I (b) พารามิเตอร์ ABC Case II

จากภาพที่ 2(a) และ 2(b) แสดงเวลาทำงานรวมเฉลี่ยโดยทำการเปรียบเทียบขั้นตอนการจัดตารางงานในเครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในเหมือนกัน โดยใช้ขั้นตอนวิธีอาณานิคมผึ้งเทียมควบคู่กับการจัดลำดับงานแบบฮิวริสติกและการจัดลำดับงานแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ABC\_LJF มีประสิทธิภาพในการจัดตารางงานดีที่สุด เมื่อปริมาณงานในระบบมีปริมาณเพิ่มมากขึ้น

กรณีที่ 2 : ทดสอบความสามารถในการทำงานเปรียบเทียบกับจำนวนของงานที่มีการเปลี่ยนแปลง กรณีเครื่องคอมพิวเตอร์เสมือนมีสภาพแวดล้อมภายในต่างกัน เมื่อ VM มีจำนวนคงที่ โดยแกน Y แสดงเวลาทำงานรวมเฉลี่ยของงานในระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปเผยแพร่โดยไม่ได้รับอนุญาตเห็นว่าเป็นการละเมิดลิขสิทธิ์และไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3: เวลาทำงานรวมเฉลี่ย เมื่องานมีปริมาณเพิ่มขึ้นและ VM มีสภาพแวดล้อมภายในต่างกัน โดยที่ (a) พารามิเตอร์ ABC Case I (b) พารามิเตอร์ ABC Case II

จากภาพที่ 3(a) และ 3(b) แสดงเวลาทำงานรวมเฉลี่ย โดยการเปรียบเทียบขั้นตอนการ จัดตารางงาน เพื่อเข้าใช้เครื่องคอมพิวเตอร์เสมือนที่มีสภาพแวดล้อมภายในต่างกัน โดยใช้ขั้นตอนวิธีอาณานิคมซึ่งเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก และการจัดตารางแบบฮิวริสติกเพียงอย่างเดียว ผลจากการทดลองแสดงให้เห็นว่า ABC\_SJF มีประสิทธิภาพในการจัดตารางการทำงานที่ดีที่สุด เมื่องานในระบบมีปริมาณมากกว่า 400 งานขึ้นไป

## 5. บทสรุป

งานวิจัยนี้ได้เสนอวิธีการจัดตารางงานบนพื้นฐานของการกระจายงานแบบสมดุลในการประมวลผลแบบกลุ่มเมฆโดยใช้ขั้นตอนวิธีอาณานิคมซึ่งเทียบควบคู่กับการจัดลำดับงานแบบฮิวริสติก โดยผลจากการทดลองแสดงให้เห็นว่า ในกรณีที่สภาพแวดล้อมภายในของเครื่องคอมพิวเตอร์เสมือนมีสภาพเหมือนกันควรใช้ขั้นตอนวิธีอาณานิคมซึ่งเทียบที่นำเสนอทำงานร่วมกับการจัดตารางงานโดยพิจารณาจากงานที่มีขนาดใหญ่ที่สุดเข้าทำงานก่อน (ABC\_LJF) และในกรณีที่สภาพแวดล้อมภายในของเครื่องคอมพิวเตอร์เสมือนมีสภาพต่างกันควรใช้วิธีการที่นำเสนอควบคู่กับการจัดลำดับงานโดยพิจารณาจากงานที่มีขนาดเล็กที่สุดเข้าทำงานก่อน (ABC\_SJF) เมื่อปริมาณของงานมีมากกว่า 400 งาน จะทำให้ประสิทธิภาพในการจัดตารางการทำงานเหมาะสมที่สุด ในการกระจายงานแบบสมดุลและเวลารวมในการทำงานน้อยที่สุด เมื่อปริมาณงานในระบบเพิ่มมากขึ้นและสภาพแวดล้อมภายในของระบบมีการเปลี่ยนแปลง

## เอกสารอ้างอิง

- [1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the acm*, vol. 53, no. 4, pp. 50-58, 2010.
- [2] K. Dutta, "A smart job scheduling system for cloud computing service providers and users: modeling and simulation," in *1st Int'l Conf. on Recent Advances in Information Technology RAIT-2012*, 2012.
- [3] S. Sindhu, and S. Mukherjee, "Efficient task scheduling algorithms for cloud computing environment," *High Performance Architecture and Grid Computing, Communications in Computer and Information Science*, vol. 169, pp. 79-83, 2011.
- [4] D. Karaboga, and B. Basturk, "Artificial Bee Colony optimization algorithm for solving constrained optimization problems," *IFSA 2007*, LNAI 4529, pp. 789-798, 2007.
- [5] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review 2012, Springer Science Business Media B.V. 2012*, 2012.
- [6] Z.R.M. Azmi, K.A. Bakar, M.S. Shamsir, W.N.W. Manan, A.H. Abdullah., "Performance comparison of priority rule scheduling algorithms using different inter arrival time jobs in grid environment," *International Journal of Grid and Distributed Computing*, Vol. 4, No. 3, 61-70, 2011
- [7] Y. Fang, F. Wang, and J. Ge, "A task scheduling algorithm based on load balancing in cloud computing," in *Web Information Systems and Mining, Lecture Notes in Computer Science 2010*, Vol. 6318, pp. 271-277, 2010.
- [8] R. N. Calheiros, R. Ranjan, C. A. F. D. Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software : Practice and Experience*, Vol. 41, No.1, pp. 23-50, Jan 2011.
- [9] B. Mondal, K. Dasgupta, and P. Dutta, "Load balancing in cloud computing using Stochastic Hill Climbing-A soft computing approach," in *Procedia Technology*, vol. 4, pp. 783-789, 2011.
- [10] S. Bitam, "Bees life algorithm for job scheduling in cloud computing," in *ICCIT 2012*, pp. 186-191, 2012.
- [11] T. Mizan, S. M. R. A. Masud, and R. Latip, "Modified bees life algorithm for job scheduling in hybrid cloud," in *IJET*, vol. 2, no.6, pp. 974-979, June 2012.
- [12] B. Kruekaew and W. Kimpan, "Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony," *IMECS 2014*, HK, pp. 18-22, March 12-14, 2014.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

- ชื่อ – สกุล นางสาวบุญหทัย เครือแก้ว
- วัน เดือน ปีเกิด 29 พฤศจิกายน 2528
- ที่อยู่ 66 หมู่ที่ 2 ถ.เจ้าฟ้า ต.วิจิต อ.เมือง จ.ภูเก็ต 83000
- E-mail s3650853@kmitl.ac.th, iced\_milk3@hotmail.com
- ประวัติการศึกษา 2557 วิทยาศาสตรมหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ เกรดเฉลี่ย 3.83  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ  
2550 วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ เกรดเฉลี่ย 3.20  
มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ สงขลา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้