บทที่ 2
ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง

การวิจัยเรื่อง การประยุกต์ใช้ฟัซซีลอจิก เพื่อจัดสรรหน่วยความจำของเครื่องจักรเสมือนในระบบการประมวลผลคลาวด์แบบส่วนตัว จำเป็นที่จะต้องมีความรู้เกี่ยวกับวิธีการพยากรณ์แบบ EWMA ทฤษฎีฟัซซีลอจิก และการประมวลผลคลาวด์ ซึ่งในบทนี้จะกล่าวถึงเนื้อหาดังต่อไปนี้

2.1 วิธีการพยากรณ์แบบ EWMA (Exponentially Weighted Moving Average)

2.2 ทฤษฎีฟัซซีลอจิก (Fuzzy Logic)

2.3 การประมวลผลคลาวด์ (Cloud Computing)

2.4 OpenStack Cloud Software

2.5 โปรแกรม Apache JMeter

2.6 Memory Ballooning

2.7 งานวิจัยที่เกี่ยวข้อง

2.1 วิธีการพยากรณ์แบบ EWMA[1]

การพยากรณ์ สามารถนำมาช่วยกำหนดทรัพยากรในปัจจุบัน และทำให้ทราบว่าทรัพยากรที่มีอยู่ ได้ถูกใช้งานอย่างมีประสิทธิภาพหรือไม่ ลักษณะการใช้งานเป็นอย่างไร เพื่อที่องค์กรจะสามารถจัดหาทรัพยากรอื่น ๆ มาเพิ่มจากข้อมูลพื้นฐานที่มีอยู่ในปัจจุบันกับระยะเวลาที่กำหนดไว้ในแผน ซึ่งสามารถนำไปใช้ในการวางแผนให้สอดคล้องกับสถานการณ์ที่เกิดขึ้นใน

---

[1]http://logisticscorner.com/Docfiles/inventory/Forecasting.pdf.

การพยากรณ์ด้วยวิธีการหาค่าเฉลี่ยเคลื่อนที่ถ่วงน้ำหนักแบบเอ็กโพเนนเชียล (Exponentially Weighted Moving Average : EWMA) เป็นวิธีที่ให้ความสำคัญกับข้อมูลแต่ละค่าไม่เท่ากัน โดยจะให้ความสำคัญกับข้อมูลล่าสุดมากที่สุด และให้ความสำคัญกับข้อมูลในอดีตน้อยลงแบบเอ็กซ์โพเนนเชียล โดยมีค่าถ่วงน้ำหนัก หรือสัมประสิทธิ์ปรับให้เรียบ (เรียกว่า ค่าแอลฟา : α) และมีสูตรในการคำนวณ ดังนี้

$$F(t) = \alpha * A(t-1) + (1-\alpha) * F(t-1) \quad (1)$$

โดยที่ F(t) คือ ค่าพยากรณ์ ณ เวลา t

A(t – 1) คือ ค่าจริงที่ผ่านมาล่าสุด

F(t – 1) คือ ค่าพยากรณ์ที่ผ่านมาล่าสุด

α คือ ค่าสัดส่วน โดยที่ $\alpha = \frac{F(t-1)}{A(t-1)}$

และกำหนดค่าพยากรณ์เริ่มต้น จากค่าเฉลี่ยของค่าจริงที่ผ่านมา

ตารางที่ 2.1 ตัวอย่างการคำนวณการพยากรณ์จำนวนลูกค้าในสัปดาห์ที่ 4 เมื่อกำหนดให้ α = 0.1

สัปดาห์ที่	**จำนวนลูกค้า (คน)**	**จำนวนลูกค้า (คน) (ค่าพยากรณ์)**
1	400 ค่าเฉลี่ย	
2	380 390	
3	411	390
4	?	
5	?	

กำหนดค่าพยากรณ์เริ่มต้น โดยหาจากค่าเฉลี่ยของจำนวนลูกค้าในสองสัปดาห์ที่ผ่านมา จะได้ค่าเฉลี่ยเท่ากับ (400+380)/2 = 390 ซึ่งจะนำไปใช้เป็นค่าพยากรณ์เริ่มต้น และสามารถพยากรณ์จำนวนลูกค้าในสัปดาห์ที่ 4 ได้ดังนี้

$$F(t) = \alpha * A(t-1) + (1-\alpha) * F(t-1)$$

F(4) = 0.1 *A(3) + (1 – 0.1) * F(3)

= 0.1 * 411 + 0.9 * 390

= 392.1 หรือ 392 คน

หากกำหนดให้จำนวนลูกค้าจริงในสัปดาห์ที่ 4 เท่ากับ 415 ดังนั้น ค่าพยากรณ์ในสัปดาห์ที่ 5 จะเท่ากับ

F(5) = 0.1 * A(4) + (1 – 0.1) * F(4)

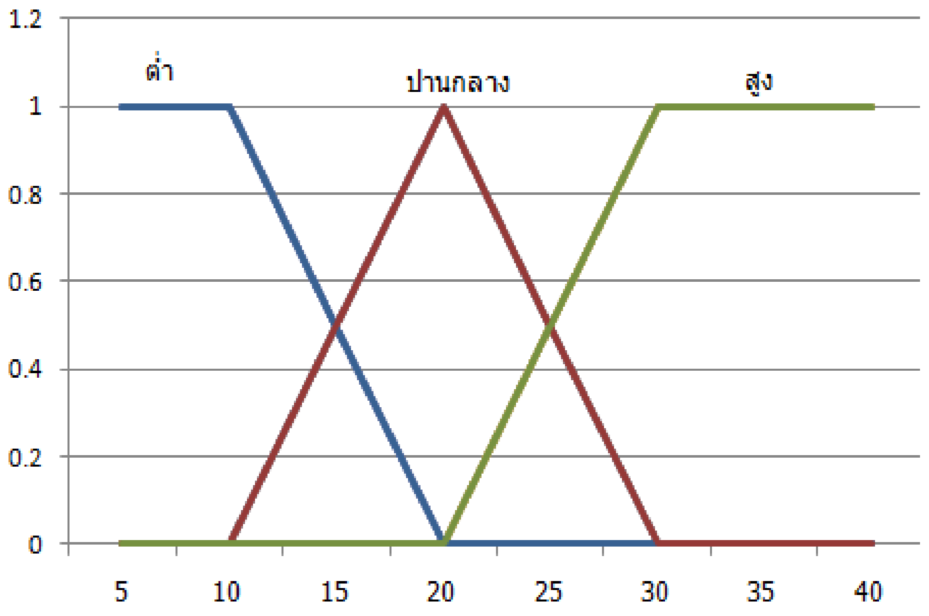
= 0.1 * 415 + 0.9 * 392.1

= 394.4 หรือ 394 คน

2.2 ทฤษฎีฟัซซีลอจิก (Fuzzy Logic)

ทฤษฎีฟัซซีลอจิก (Fuzzy Logic) เป็นทฤษฎีตรรกะทางคณิตศาสตร์ที่คิดค้น ในปี ค.ศ. 1965 โดย Lotfali Askar Zadeh (L. A. Zadeh) ศาสตราจารย์จากมหาวิทยาลัยแคลิฟอร์เนียร์ ซึ่งทฤษฎีนี้ นำมาใช้เป็นเครื่องมือช่วยในการตัดสินใจความไม่ชัดเจน ไม่แน่นอนของข้อมูล โดยมีตรรกะอยู่บนพื้นฐานความเป็นจริง ใช้หลักเหตุผลคล้ายกับวิธีการคิดของมนุษย์ และใช้ประสบการณ์ของผู้เชี่ยวชาญมากกว่าการใช้ทฤษฎี โดยทำงานภายใต้เงื่อนไข หรือข้อกำหนดของข้อมูลที่ไม่ได้มีเพียงแค่ 0 กับ 1 หรือ ใช่ กับ ไม่ใช่ แต่ข้อมูลมีการแบ่งระดับที่เรียกว่า Fuzzy Set เช่น การกำหนดระดับข้อมูลที่มีความหมายแทนคำว่า “น้อย” “ค่อนข้างน้อย” “ปานกลาง” “ค่อนข้างมาก” “มาก” เป็นต้น

ตัวอย่างการแบ่งระดับข้อมูลของเครื่องวัดอุณหภูมิที่มีการแบ่งระดับเป็น “ต่ำ” “ปานกลาง” และ “สูง”



ภาพที่ 2.1 ตัวอย่าง Fuzzy Set ของเครื่องวัดอุณหภูมิ

ในการนำทฤษฎีฟัซซีลอจิกไปใช้งานนั้น จะแสดงอยู่ในรูปแบบของการกำหนดเงื่อนไข คือ “ถ้า......แล้ว” (IF……THEN)

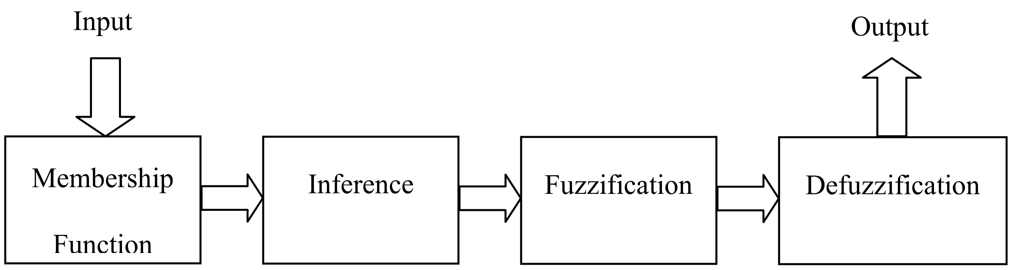
ตัวอย่าง เช่น ระบบควบคุมอุณหภูมิโดยใช้พัดลม อาจมีเงื่อนไข ดังนี้

IF <อุณหภูมิ เย็นมาก> THEN <หยุดพัดลม>

IF <อุณหภูมิ เย็น> THEN <พัดลมหมุนช้าลง>

IF <อุณหภูมิ ปานกลาง> THEN <พัดลมหมุนคงที่>

IF <อุณหภูมิ ร้อน> THEN <พัดลมหมุนเร็วขึ้น>



ภาพที่ 2.2 ขั้นตอนการประมวลผลแบบฟัซซีลอจิก

สำหรับขั้นตอนการประมวลผลแบบฟัซซีลอจิกนั้น ประกอบด้วย 4 ขั้นตอน ดังนี้

1) Membership Function เป็นขั้นตอนการแปลงข้อมูลทั่วไปที่เข้ามา ให้เป็นค่าระดับความเป็นสมาชิกที่สอดคล้องกับ fuzzy set ที่ได้ออกแบบไว้สำหรับตัวแปรนั้น ๆ หรืออาจเรียกว่าตัวแปรภาษา (Linguistic Variable) โดยในการออกแบบนั้น ไม่จำเป็นต้องมีลักษณะเดียวกัน แต่จะขึ้นอยู่กับคุณลักษณะของตัวแปรแต่ละอินพุท และความสำคัญต่อเอาท์พุทที่น่าสนใจ โดยมีลักษณะการกำหนดฟังก์ชั่นเป็นภาษาสามัญ

2) Inference เป็นการนำข้อมูลจากขั้นตอนที่ 1 มาตีความร่วมกับกลุ่มของกฎ หรือเงื่อนไขที่กำหนดไว้ อาศัยหลักการของเหตุและผล โดยมีการสร้างความสัมพันธ์ระหว่างอินพุททั้งหมดที่เกี่ยวข้องกับเอาท์พุท มาเขียนเป็นกฎในการควบคุมระบบ เพื่อให้ได้การตัดสินใจที่เหมาะสม

3) Fuzzification เป็นขั้นตอนการนำกฎที่สร้างขึ้นในขั้นตอนที่ 2 มาประมวลผลกับข้อมูลฟัซซีที่เป็นอินพุท โดยใช้วิธีการทางคณิตศาสตร์ เพื่อหาค่าการประมวลผล

4) Defuzzification เป็นขั้นตอนการแปลงค่าระดับความเป็นสมาชิก ให้เป็นค่าทั่วไปเพื่อนำค่าที่ได้มาใช้ในการตัดสินใจ เพื่อควบคุมระบบในสถานการณ์นั้น ๆ ซึ่งวิธีการทำ Defuzzification นั้น วิธีที่เป็นที่นิยมและใช้งานกันอย่างแพร่หลาย คือ วิธีหาจุดศูนย์ถ่วง (Centroid หรือ Center of Gravity : COG)

Fuzzy Logic มีบทบาทมากขึ้นในวงการวิจัยด้านคอมพิวเตอร์ และได้ถูกนำไปประยุกต์ใช้งานในด้านต่าง ๆ มากมาย เช่น ด้านการแพทย์ ด้านการทหาร ด้านธุรกิจ ด้านอุตสาหกรรม เป็นต้น เนื่องจาก Fuzzy Logic มีจุดเด่น และแตกต่างจากการคำนวณแบบอื่น ๆ ดังนี้

1) มีการใช้เหตุผลในเชิงตรรกะที่สอดคล้องกับตรรกะความคิดของมนุษย์

2) ช่วยในการตัดสินใจที่คลุมเครือ ไม่ชัดเจน ไม่ใช่แค่ผิด หรือถูกเพียง 2 สถานะ แต่เป็นดีกรีของความผิด หรือถูก ซึ่งเป็นเหตุการณ์ที่เกิดขึ้นในธรรมชาติ สอดคล้องกับสภาวะแวดล้อมในโลกจริง

3) สามารถเข้าใจได้ เนื่องจากสามารถตีความให้อยู่ในรูปแบบของ IF – THEN

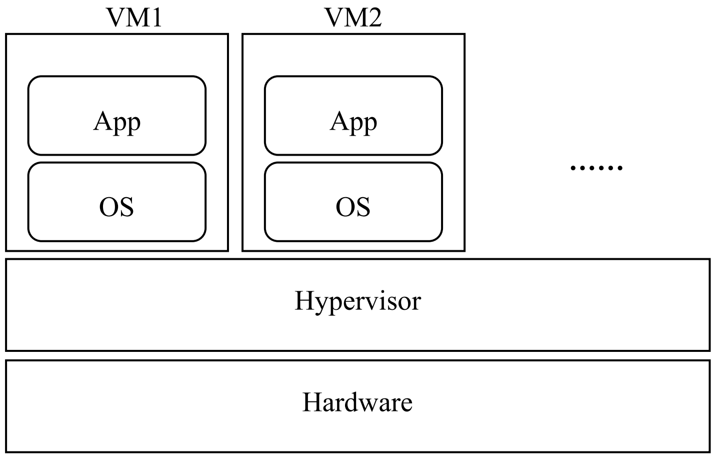
4) ผู้เชี่ยวชาญเป็นผู้กำหนดกฎ และตัวแปรต่าง ๆ ของระบบ และตรวจสอบประเมินความถูกต้องของระบบ ทำให้นำมาใช้แก้ปัญหาได้อย่างมีประสิทธิภาพ

2.3 การประมวลผลคลาวด์ (Cloud Computing)

NIST หรือ National Institute of Standards and Technology เป็นหน่วยงานภายใต้กระทรวงพาณิชย์ สหรัฐอเมริกา (U.S. Department of Commerce) ก่อตั้งในปี ค.ศ. 1901 และเป็นหน่วยงานปฏิบัติการวิจัยวิทยาศาสตร์ฟิสิกส์แห่งแรกของรัฐ ภารกิจของ NIST คือ การสนับสนุนการพัฒนานวัตกรรมและการสร้างความสามารถทางอุตสาหกรรมของสหรัฐอเมริกา ให้มีความก้าวหน้าทางวิทยาศาสตร์ด้านมาตรวิทยา มาตรฐาน และเทคโนโลยี เพื่อความมั่นคงทางเศรษฐกิจและคุณภาพชีวิตที่เพิ่มขึ้น

NIST ได้ให้คำนิยามการประมวลผลคลาวด์ (Cloud Computing) ไว้ดังนี้

การประมวลผลคลาวด์ (Cloud Computing) เป็นเทคโนโลยีที่มีการใช้งานกันอย่างแพร่หลาย โดยมีรูปแบบการให้บริการผ่านทางอินเตอร์เน็ท ที่มีผู้ให้บริการจัดสรรทรัพยากรแก่ผู้ใช้บริการ ให้ใช้งานได้ตามความต้องการของผู้ใช้งาน มีการปรับเพิ่มและลดทรัพยากร สามารถทำได้อย่างง่ายและรวดเร็ว สามารถเข้าถึงข้อมูลได้จากทุกที่ ทุกเวลา ผ่านทางอุปกรณ์ต่าง ๆ เช่น Computer, Tablet และ Smartphone เป็นต้น โดยที่ผู้ใช้บริการ ไม่จำเป็นต้องมีความรู้ หรือมีความเชี่ยวชาญทางเทคนิคสำหรับการทำงาน ๆ นั้น สามารถทำงานได้โดยอาศัยไฮเพอร์ไวเซอร์ (Hypervisor) หรือเวอร์ชวลไลเซชัน (Virtualization) เช่น Microsoft Hyper-V, VMware vSphere, Citrix Xen หรือ Linux KVM เป็นต้น เพื่อเข้าถึงทรัพยากรที่แท้จริงของแต่ละโหนด และกำหนดทรัพยากรที่มีอยู่ให้แต่ละเครื่องจักรเสมือน โดยทรัพยากรที่กำหนดให้ทั้งหมด ต้องไม่เกินกว่าที่โหนดมีอยู่ ซึ่งทรัพยากรที่ต้องจัดสรร ได้แก่ หน่วยประมวลผล (CPU) หน่วยความจำ (Memory) หน่วยจัดเก็บข้อมูล (Disk) และการเชื่อมต่อเครือข่ายคอมพิวเตอร์ (Network) เป็นต้น ซึ่งแต่ละเครื่องจักรเสมือน สามารถติดตั้ง Operating System (OS) ที่แตกต่างกัน หรือเหมือนกันก็ได้ เพื่อให้ใช้ระบบได้อย่างเต็มประสิทธิภาพและเกิดประโยชน์สูงสุด



ภาพที่ 2.3 การประมวลผลคลาวด์

คุณสมบัติที่สำคัญของการประมวลผลคลาวด์

1) On-demand Self-Service ผู้ใช้บริการสามารถระบุความต้องการ และใช้บริการทรัพยากรคอมพิวเตอร์ได้เอง โดยไม่ต้องรอผู้ให้บริการมาดำเนินการ

2) Broad Network Access เข้าถึงระบบเครือข่ายได้จากทุกอุปกรณ์ที่มีการเชื่อมต่อกับอินเตอร์เน็ท

3) Resource Pooling ทรัพยากรและการประมวลผล จะถูกรวบรวมไว้ที่ศูนย์กลาง เพื่อตอบสนองความต้องการใช้งานของผู้ใช้ที่แตกต่างกัน เป็นอิสระต่อกัน โดยที่ผู้ใช้ไม่จำเป็นต้องรู้ว่าใช้ทรัพยากรนั้นจากที่ใด

4) Rapid Elasticity มีความยืดหยุ่นในการเพิ่ม ลดทรัพยากรที่มีการปรับเปลี่ยนได้อย่างรวดเร็ว อัตโนมัติ ไม่จำกัดจำนวน และไม่จำกัดเวลา

5) Measured Services ผู้ใช้บริการและผู้ให้บริการ สามารถติดตามและควบคุมปริมาณการใช้ทรัพยากรได้ โดยเฉพาะผู้ใช้บริการ นั่นคือ ใช้งานได้ตามที่จ่ายจริง

รูปแบบการให้บริการของการประมวลผลคลาวด์

1) Software as a Service (SaaS) เป็นการใช้บริการ software หรือ application บนโครงสร้างพื้นฐานของการประมวลผลคลาวด์ ที่ผู้ให้บริการจัดหาให้ผู้ใช้บริการ สามารถเข้าถึงได้จากหลากหลายอุปกรณ์ผ่านทางเว็บบราวเซอร์ หรือหน้าโปรแกรม แต่ผู้ใช้บริการไม่สามารถ

บริหารจัดการ หรือควบคุมโครงสร้างพื้นฐานได้ รวมถึงระบบเครือข่าย เครื่องแม่ข่าย ระบบปฏิบัติการ อุปกรณ์จัดเก็บข้อมูล เป็นต้น

2) Platform as a Service (PaaS) เป็นบริการที่ผู้ใช้บริการสามารถพัฒนา software หรือ application ที่ผู้ให้บริการจัดเตรียมให้เท่านั้น โดยที่ผู้ใช้บริการไม่สามารถบริหารจัดการ หรือควบคุมโครงสร้างพื้นฐานได้ รวมถึงระบบเครือข่าย เครื่องแม่ข่าย ระบบปฏิบัติการ อุปกรณ์จัดเก็บข้อมูล เป็นต้น แต่สามารถกำหนดเงื่อนไข และตั้งค่าคุณลักษณะของเครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบ

3) Infrastructure as a Service (IaaS) เป็นบริการระดับฮาร์ดแวร์ (Hardware) ในรูปแบบของเครื่องเสมือน (Virtual Machine) เช่น ระบบเครือข่าย เครื่องแม่ข่าย อุปกรณ์จัดเก็บข้อมูล และทรัพยากรอื่น ๆ ที่ใช้งานในโครงสร้างพื้นฐาน เป็นต้น ซึ่งผู้ใช้บริการสามารถปรับเปลี่ยนระบบปฏิบัติการและ Application แต่ไม่สามารถบริหารจัดการโครงสร้างพื้นฐานของระบบประมวลผลคลาวด์ได้ มีประโยชน์ในการประมวลผลทรัพยากรจำนวนมาก

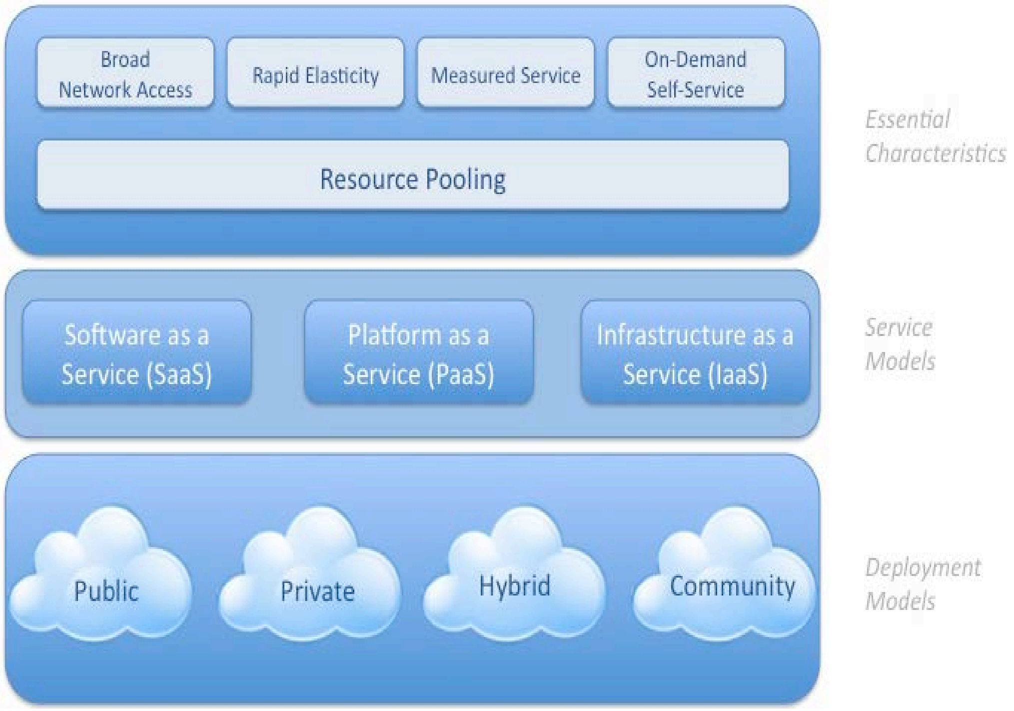
รูปแบบการใช้งานของระบบการประมวลผลคลาวด์

1) Private Cloud เป็นโครงสร้างพื้นฐานของระบบการประมวลผลคลาวด์ ที่มีการจัดเตรียมสำหรับใช้งานภายในองค์กร โดยมีองค์กรเป็นเจ้าของ และดูแลเอง หรือจัดจ้างให้องค์กรอื่นมาดูแล อาจติดตั้งไว้ภายในองค์กร หรือ นอกองค์กรก็ได้

2) Public Cloud เป็นโครงสร้างพื้นฐานของระบบการประมวลผลคลาวด์ ที่มีการจัดเตรียมสำหรับให้บุคคลทั่วไปได้ใช้งาน โดยมีเจ้าของเป็นผู้บริหารจัดการเอง อาจเป็นหน่วยงานธุรกิจ ราชการ หรือสถาบันการศึกษา โดยติดตั้งในสถานที่ของผู้ให้บริการ

3) Community Cloud เป็นโครงสร้างพื้นฐานของระบบการประมวลผลคลาวด์ ที่มีการใช้งานเฉพาะด้าน หรือทำงานในลักษณะเดียวกัน โดยอนุญาตให้หลาย ๆ องค์กรสามารถเข้าถึงและทำงานร่วมกันได้ เป็นเจ้าของเพียงองค์กรเดียว หรือหลายองค์กรร่วมกัน หรือจัดจ้างให้องค์กรอื่นมาดูแล อาจติดตั้งภายในองค์กร หรือนอกองค์กรก็ได้

4) Hybrid Cloud เป็นโครงสร้างพื้นฐานของระบบการประมวลผลคลาวด์ ที่ผสมผสานกันระหว่างโครงสร้างพื้นฐานของระบบการประมวลผลคลาวด์ที่แตกต่างกัน (Private, Public หรือ Community) โดยต้องสามารถทำงานร่วมกันได้



ภาพที่ 2.4 โครงสร้าง Cloud Computing

ที่มา: http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html

2.4 OpenStack Cloud Software

OpenStack เป็นซอฟแวร์โซลูชั่นแบบโอเพนซอร์ส ที่ใช้ในการบริหารจัดการระบบ Cloud Computing โดยเริ่มจากความร่วมมือระหว่างบริษัท Rackspace Hosting (ให้บริการเว็บโฮสติ้ง) กับ NASA โดย Rackspace นั้น สนับสนุนโค้ดในส่วนของการจัดการพื้นที่สำหรับเก็บข้อมูลบน Cloud ส่วน NASA มีระบบ Cloud พื้นฐาน ปัจจุบัน OpenStack Cloud Software อยู่ภายใต้การดูแลของ OpenStack Foundation ซึ่งเป็นองค์กรกลางที่ไม่หวังผลกำไร และไม่สังกัดบริษัทใดบริษัทหนึ่ง

OpenStack เป็นระบบปฏิบัติการ Cloud ที่ควบคุมทรัพยากรของ Compute, Storage และ Network ทั้งหมด มีการจัดการผ่านทาง Dashboard ซึ่งเป็นเว็บ ช่วยให้ผู้ดูแลระบบ สามารถควบคุมขีดความสามารถของทรัพยากรตามที่ผู้ใช้งานต้องการได้

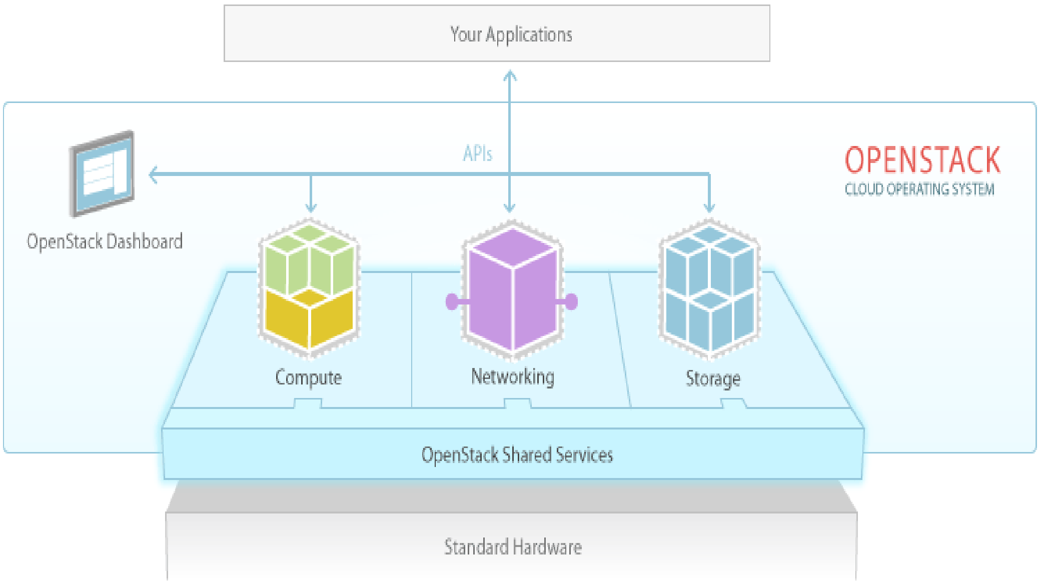
โครงสร้างระบบของ OpenStack Cloud Software มีส่วนประกอบที่สำคัญ ดังนี้

1) Compute ทำหน้าที่ในการจัดเตรียมและบริหารจัดการเครือข่ายขนาดใหญ่ของเครื่องเสมือน ซึ่งนักพัฒนาที่สร้างโปรแกรมประยุกต์บน Cloud นั้น สามารถเข้าถึงได้ผ่านทาง API ส่วนผู้ดูแลระบบและผู้ใช้บริการ สามารถเข้าถึงได้ผ่านทางเว็บ สำหรับ Hypervisor ที่ใช้งานและเป็นที่นิยม คือ KVM และ XenServer

2) Networking สามารถปรับขนาดได้ และมี API สำหรับจัดการเครือข่ายและ IP Address และผู้ใช้บริการ สามารถกำหนดค่าเครือข่ายที่จะใช้งานได้ด้วยตัวเอง เพื่อไม่ให้เกิดคอขวด หรือมีปัจจัยอื่นมาจำกัดการใช้งาน

3) Storage ใน OpenStack สนับสนุนทั้ง Object Storage และ Block Storage ให้เลือกใช้งาน โดยขึ้นอยู่กับการใช้งาน ซึ่ง Object Storage นั้น สามารถนำไปใช้สำรองข้อมูลและจัดเก็บข้อมูล ส่วน Block Storage ช่วยรักษาอุปกรณ์ที่เชื่อมต่อกับ Instance ใน Compute สำหรับจัดเก็บผลการดำเนินงานให้ดีขึ้น และทำงานร่วมกันกับอุปกรณ์จัดเก็บข้อมูลอื่นได้ เช่น NetApp

4) OpenStack Dashboard ช่วยให้ผู้ดูแลระบบและผู้ใช้งานสามารถเข้าถึงได้ผ่านทางเว็บ ในการจัดสรรทรัพยากร ใช้งานได้ง่าย และสามารถติดตาม ตรวจสอบได้



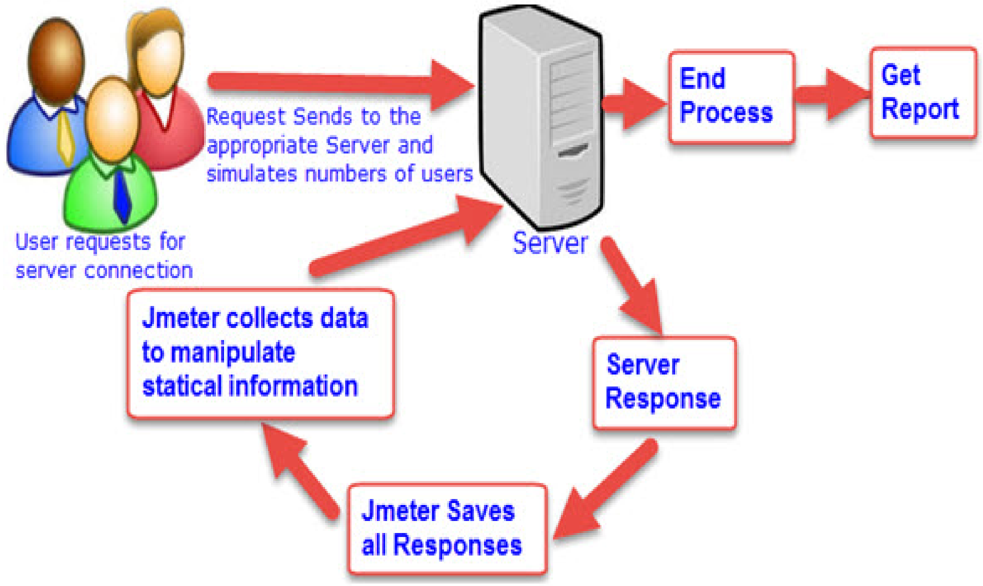
ภาพที่ 2.5 โครงสร้างระบบของ OpenStack Software

ที่มา: http://www.openstack.org/software/

OpenStack เป็น Software Solution ที่มีความสามารถหลากหลายในด้านการให้บริการ Cloud พื้นฐาน เช่น การสร้าง Image หรือ Instance และยังมีแผนการพัฒนาต่อไปในอนาคต จึงน่าจะเป็น Software ที่นำมาใช้ในการสร้าง Private Cloud ได้ดี

2.5 โปรแกรม Apache JMeter

โปรแกรม Apache JMeter เป็นซอฟต์แวร์โอเพนซอร์สที่พัฒนาโดย Apache Software Foundation ใช้ภาษา Java ในการพัฒนา และถูกออกแบบให้สามารถทำงานบน Platform ที่หลากหลาย เพื่อใช้สำหรับวัดประสิทธิภาพของระบบบริการข้อมูลบนเครื่องแม่ข่าย โดยสามารถจำลองการร้องขอการใช้บริการจากเครื่องแม่ข่ายได้หลากหลายโปรโตคอล เช่น HTTP, SMTP, POP3, IMAP เป็นต้น สามารถกำหนดจำนวนผู้ร้องขอ และอัตราการร้องขอได้ ผลลัพธ์ที่ได้จะแสดงเป็นจำนวนคำสั่งที่เครื่องแม่ข่ายสามารถให้บริการสำเร็จต่อหนึ่งวินาที จึงทำให้เป็นซอฟต์แวร์ที่มีผู้นิยมใช้งาน



ภาพที่ 2.6 ขั้นตอนการทำงานของโปรแกรม Apache JMeter

ที่มา: http://www.softwaretestingclass.com/introduction-to-apache-jmeter-tutorial-series-1/

2.6 Memory Ballooning

เป็นวิธีการดึงหน่วยความจำในส่วนที่ระบบปฏิบัติการจองไว้ แต่ไม่ได้มีการใช้งาน (Free) โดยจะนำกลับมาใช้ใหม่ จะถูกควบคุมด้วย Hypervisor สำหรับลักษณะการทำงานของ Memory Ballooning จะมีลักษณะการทำงานแบบ Virtualization คือเป็นการจัดการหน่วยความจำใน VM หลาย ๆ ตัว เมื่อ VM ต้องการใช้งานหน่วยความจำมากขึ้น จะร้องขอการใช้หน่วยความจำไปยัง Host หาก Host มีปริมาณหน่วยความจำที่เพียงพอ ก็จะจัดสรรหน่วยความจำไปให้ VM นั้น หรือหากกรณีที่ Host ต้องมีการใช้งานหน่วยความจำที่เพิ่มขึ้น Host จะขอคืนหน่วยความจำจาก VM โดยการดำเนินการดังกล่าว สามารถทำได้ทันทีโดยไม่ต้องปิดเครื่อง หรือหยุดการทำงานของ VM

ข้อดีของการใช้ Memory Ballooning

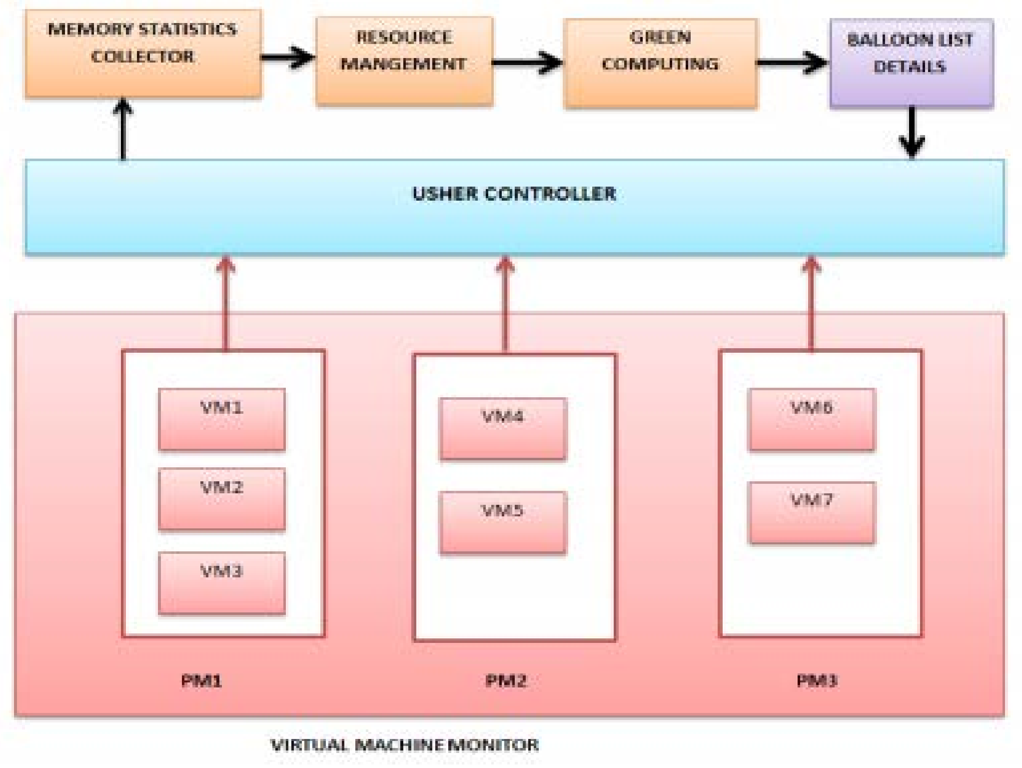
1) สามารถทำได้ทันทีโดยไม่ต้องปิดเครื่อง หรือหยุดการทำงานของ VM
2) การทำ Memory Ballooning ใน VM เป็นอิสระต่อกัน
3) เป็นวิธีการที่ง่ายในการปรับเปลี่ยนหน่วยความจำ
4) ไม่จำเป็นต้องใช้งบประมาณ หรือเสียค่าใช้จ่าย

ข้อเสียของการใช้ Memory Ballooning

1) ไม่สามารถกำหนดหน่วยความจำให้เกินกว่าที่ Host มี
2) การพยายามรักษาการกระจายตัว(Fragmentation) ในหน่วยความจำ ค่อนข้างต่ำ

2.7 งานวิจัยที่เกี่ยวข้อง

2.7.1 งานวิจัยเรื่อง Dynamic memory Allocation using ballooning and virtualization in cloud computing (V Holy Angel Jenitha, R.Veeramani, 2014) ได้นำเสนอเกี่ยวกับการปรับปรุงประสิทธิภาพของหน่วยความจำให้กับเครื่องจักรเสมือน ภายในระบบการประมวลผลคลาวด์ ที่ใช้ Xen Hypervisor โดยมีการพยากรณ์ปริมาณหน่วยความจำด้วยวิธีการพยากรณ์แบบ EWMA แล้วนำค่าที่ได้จากการพยากรณ์ มาดำเนินการจัดสรรปริมาณหน่วยความจำให้กับระบบการประมวลผลคลาวด์ด้วยวิธีบอลลูน



ภาพที่ 2.7 สถาปัตยกรรมของระบบ

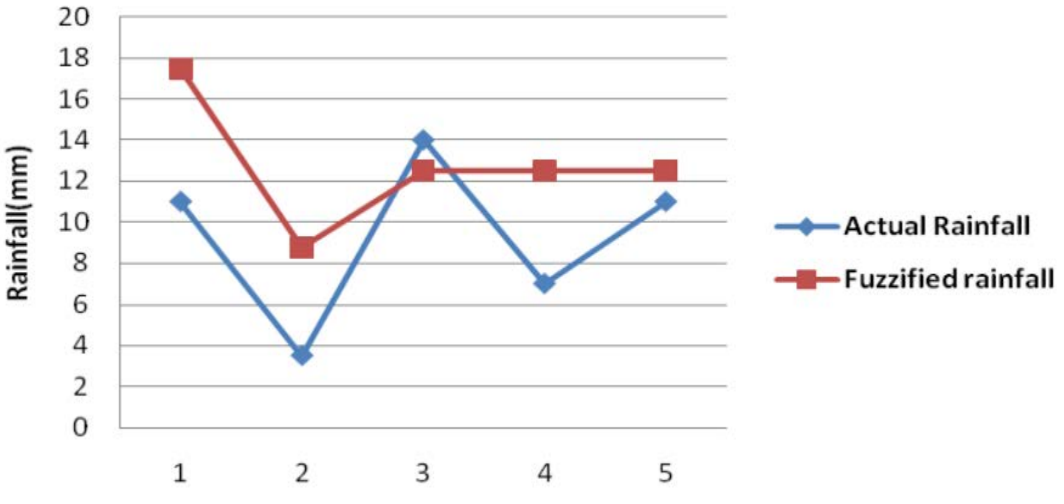
ที่มา: V Holy Angel Jenitha and R.Veeramani, “Dynamic memory Allocation using ballooning and virtualization in cloud computing” IOSR Journal of Computer Engineering(IOSR-JCE), Vol. 16, Issue 2, Ver. IV(Mar-Apr. 2014), pp.19-23.

จากภาพที่ 2.7 อธิบายการทำงานของระบบได้ว่า Usher Controller เป็นส่วนที่ใช้ในการควบคุมและจัดเก็บการใช้ทรัพยากรของ Virtual Machine แต่ละตัว และปริมาณการใช้งานหน่วยความจำในแต่ละ Virtual Machine จะถูกส่งไปจัดเก็บที่ Memory Statics Collector จากนั้นข้อมูลจะถูกส่งต่อไปยัง Resource Management ซึ่งเป็นส่วนที่ใช้สำหรับการบริหารจัดการทรัพยากร ไม่ว่าจะเป็นการพยากรณ์ความต้องการในการใช้หน่วยความจำด้วยสมการ EWMA การหา Virtual Machine ตัวถัดไป ที่ควรจะจัดสรรหน่วยความจำในครั้งต่อไปให้ โดยมี Green Computing เป็นตัวควบคุมไม่ให้มีการใช้หน่วยความจำเกินเกณฑ์ที่กำหนด หรือหลีกเลี่ยงสิ่งที่ไม่จำเป็นในการใช้หน่วยความจำ และ Balloon List Details จะเก็บรายละเอียดในการจัดสรรหน่วยความจำ และจะทำงานเมื่อ Physical Machine เริ่มขาดแคลนหน่วยความจำ ทำให้จำเป็นต้อง

บังคับ Virtual Machine ให้ปล่อยหน่วยความจำที่ยังไม่ถูกใช้งานออกมาให้ ซึ่งงานวิจัยนี้ สามารถนำไปให้ผู้ให้บริการใช้งานใน Data Center เพื่อบริหารจัดการการใช้หน่วยความจำ โดยสามารถเพิ่มประสิทธิภาพให้กับหน่วยความจำได้ด้วยวิธีการ Balloon ซึ่งเพิ่ม หรือลดหน่วยความจำได้ และวิธีการ Balloon เหมาะกับหน่วยงานที่มีงบประมาณน้อย

จากการวิเคราะห์งานวิจัยนี้ พบว่า งานวิจัยนี้ไม่ได้กล่าวถึงผลลัพธ์ที่ได้จากวิธีการพยากรณ์ปริมาณหน่วยความจำในระบบ ว่ามีผลต่อระบบอย่างไร หรือวิธีการพยากรณ์ดังกล่าวที่ใช้ในงานวิจัย มีความเหมาะสม สัมพันธ์กับงาน หรือหน่วยความจำหรือไม่

2.7.2 งานวิจัยเรื่อง Modeling Rainfall Prediction using Fuzzy Logic (Jimoh, R.G. และคณะ ,2013) ได้นำเสนอเกี่ยวกับการนำแบบจำลองฟัซซีลอจิก มาประยุกต์ใช้ในการคาดการณ์ปริมาณน้ำฝนที่จะเกิดขึ้นในอนาคต เพื่อเป็นการวางแผนป้องกันภัยพิบัติทางธรรมชาติ เช่น น้ำท่วม ภัยแล้ง เป็นต้น และแม้ว่าจะมีกระบวนการทางอุตุนิยมวิทยาที่มีการปรับปรุงแก้ไขปัญหา แต่ก็ยังไม่เป็นไปตามขอบเขตที่กำหนด จำเป็นต้องมีทางเลือกอื่นในการวิเคราะห์และคาดการณ์ ซึ่งแบบจำลองฟัซซีลอจิก ก็เป็นทางเลือกหนึ่งที่มีประสิทธิภาพในการคาดการณ์ โดยมีการกำหนดตัวแปรอินพุท 2 ตัว คือ ความเร็วลมและอุณหภูมิ และตัวแปรเอาท์พุท 1 ตัว เป็นปริมาณน้ำฝนที่คาดว่าจะเกิด ซึ่งงานวิจัยนี้ ทำให้สามารถคาดการณ์ปริมาณน้ำฝนได้

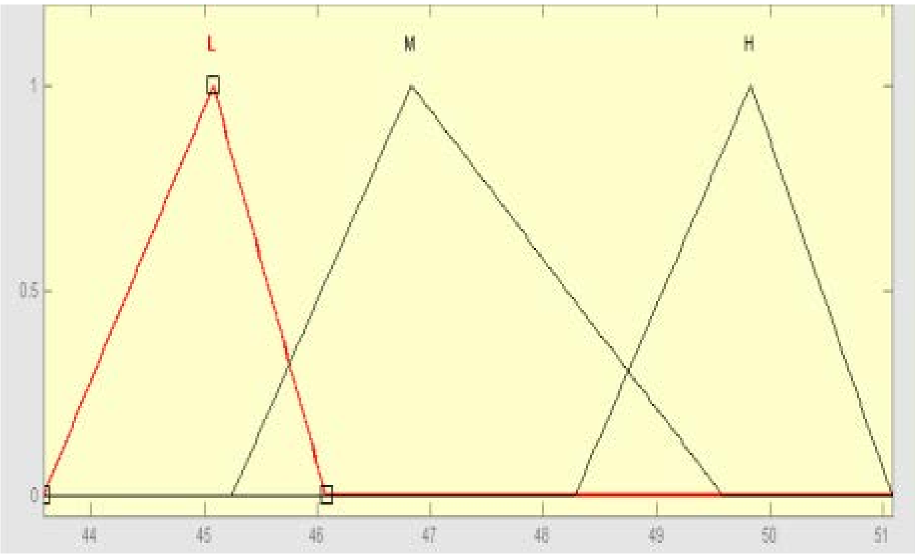


ภาพที่ 2.8 เปรียบเทียบปริมาณน้ำฝนที่เกิดขึ้นจริงกับปริมาณน้ำฝนที่ได้จากการพยากรณ์

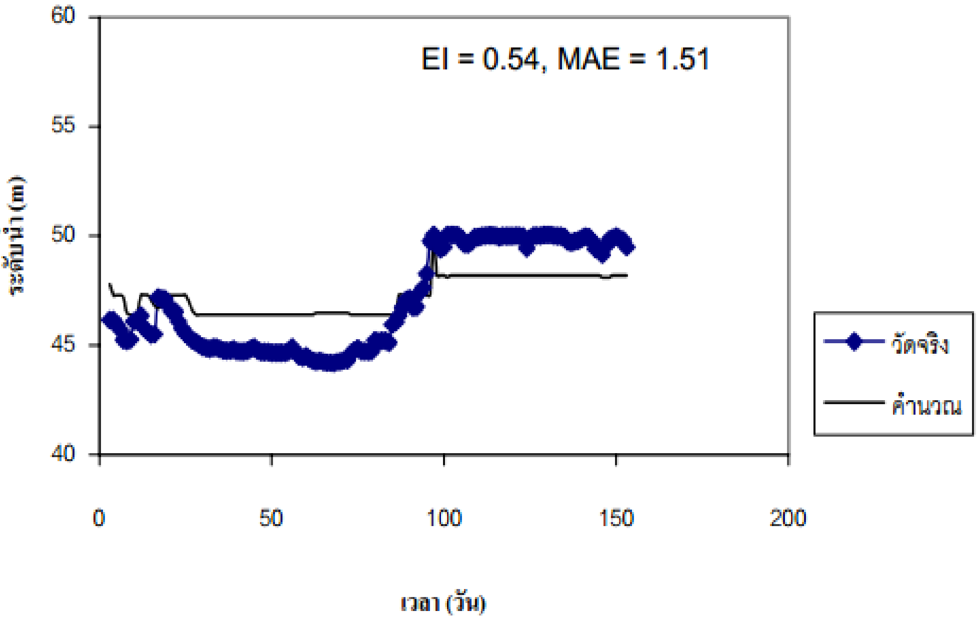
ที่มา: Jimoh, R.G., Olagunu, M., Folorunso, I.O., Asiribo and M.A., "Modeling Rainfall Prediction using Fuzzy Logic" International Journal of Innovative Research in Computer and Communication Engineering(IJIRCCE), Vol. 1, Issue 4, June 2013.

จากการวิเคราะห์งานวิจัยนี้ พบว่า ค่าการพยากรณ์ปริมาณน้ำฝนโดยส่วนใหญ่จากแบบจำลองฟัซซีลอจิก มีค่ามากกว่าปริมาณน้ำฝนที่เกิดขึ้นจริง ซึ่งสามารถนำแบบจำลองนี้ไปใช้ในการคาดการณ์ปริมาณน้ำฝนได้ แต่อย่างไรก็ตาม การใช้ตัวแปรอินพุทเพียง 2 ตัว อาจไม่เพียงพอที่จะนำไปใช้วางแผนการคาดการณ์เพื่อแก้ไขปัญหาบางปัญหา เนื่องจากยังมีปัจจัยอื่น ๆ ที่มีผลต่อปริมาณน้ำฝน เช่น ความกดอากาศ ความชื้นสัมพัทธ์ เป็นต้น หากมีการเพิ่มเติมปัจจัยที่เกี่ยวข้อง อาจทำให้การพยากรณ์ปริมาณน้ำฝน มีความแม่นยำมากยิ่งขึ้น

2.7.3 งานวิจัยเรื่อง การพยากรณ์น้ำท่วมโดยแบบจำลองฟัซซีลอจิก (ยุพา ชิดทอง, วิภาดา แซ่อึ้ง และเมธี สายมงคล, 2547) ได้นำเสนอเกี่ยวกับการประยุกต์ใช้แบบจำลองฟัซซีลอจิก เพื่อการพยากรณ์ระดับน้ำท่วมล่วงหน้าบริเวณอำเภอเมือง จังหวัดสุโขทัย โดยอาศัยผู้เชี่ยวชาญในการวิเคราะห์ความสัมพันธ์ของข้อมูลในอดีต ซึ่งพบว่า ระดับน้ำที่สถานี Y4 ในวันพรุ่งนี้ จะขึ้นกับระดับน้ำใน 2 วันที่ผ่านมา จึงทำให้มีตัวแปรอินพุท 2 ตัวแปร คือ Y4(t – 1) (ระดับน้ำที่สถานี Y4 เมื่อวาน), Y4(t) (ระดับน้ำที่สถานี Y4 วันนี้) และตัวแปรเอาท์พุท 1 ตัวแปร คือ Y4(t + 1) (ระดับน้ำที่สถานี Y4 วันพรุ่งนี้) โดยที่แต่ละตัวแปรมีค่าฟังก์ชั่นการเป็นสมาชิก 3 ค่า คือ ต่ำ ปานกลาง และสูง และเมื่อทำการตัดกฎที่ไม่จำเป็นออกไป จะเหลือเพียง 6 กฎ จากนั้นทำการคำนวณประสิทธิภาพและค่าความคลาดเคลื่อนเฉลี่ยสมบูรณ์ ซึ่งผลการศึกษาพบว่า แบบจำลองฟัซซีลอจิกให้ผลลัพธ์ด้านประสิทธิภาพมากกว่า 90% และค่าความคลาดเคลื่อนเฉลี่ยสมบูรณ์มีค่า 0.39



ภาพที่ 2.9 Membership Function ของระดับน้ำที่สถานี Y4 มีค่าระดับเป็น Low, Medium และ High



ภาพที่ 2.10 ผลการคำนวณระดับน้ำ และการคำนวณประสิทธิภาพ

จากการวิเคราะห์งานวิจัยนี้ พบว่า งานวิจัยนี้ อาศัยผู้เชี่ยวชาญในการวิเคราะห์ข้อมูลและกำหนดกฎฟัซซี โดยมีความสัมพันธ์กันของข้อมูลในอดีต ซึ่งทำให้การพยากรณ์มีประสิทธิภาพและมีความแม่นยำยิ่งขึ้น