

การเปรียบเทียบสมรรถนะระหว่าง pfSense และ Mikrotik RouterOS ในการกระจายภาระงาน

วุฒิพงศ์ โปธิสาร¹

ชัยพร เขมะภาคะพันธ์²

บทคัดย่อ

pfSense และ Mikrotik RouterOS เป็นซอฟต์แวร์ที่ออกแบบมาเพื่อให้คอมพิวเตอร์สามารถทำหน้าที่เป็นโหนดบาลานซ์และเราเตอร์สำหรับระบบเครือข่าย ซึ่ง RouterOS ได้รับการพัฒนาโดยบริษัท Mikrotik ซึ่งเป็นบริษัทที่พัฒนาทางด้านฮาร์ดแวร์ของเราเตอร์ แม้ว่า pfSense และ Mikrotik RouterOS จะได้รับการออกแบบมาพร้อมกับแอปพลิเคชันที่คล้ายกัน แต่ก็แตกต่างกันไปบ้างซึ่งส่วนใหญ่เกี่ยวกับคุณลักษณะการทำงานของระบบ และการออกใบอนุญาตการใช้งาน ซึ่ง pfSense จะเป็นแบบโอเพ่นซอร์ส (Open source) สามารถดาวน์โหลดใช้งานได้ฟรี โดยไม่เสียค่าใช้จ่ายใด ๆ การประเมินประสิทธิภาพของ Load Balancer จึงเป็นสิ่งจำเป็นที่ผู้ออกแบบระบบเครือข่ายต้องทราบ เพื่อให้สามารถระบุค่าพารามิเตอร์สำคัญที่อาจส่งผลกระทบต่อสมรรถนะการทำงานของระบบโดยรวม การตัดสินใจเลือกใช้ให้เหมาะสมกับงานในแต่ละแบบที่มักมีความต้องการใช้งานแตกต่างกันออกไป จึงนับเป็นเรื่องที่ควรให้ความสำคัญเป็นอันดับแรก ๆ เนื่องจากโดยปกติทั่วไปการเลือกใช้ซอฟต์แวร์หรือฮาร์ดแวร์ Load Balancer มาใช้งานนั้น มักใช้วิธีการศึกษาจากคู่มือ หรือเลือกมาจากความนิยมของผู้ใช้งานส่วนใหญ่เป็นหลัก ซึ่งแนวทางพิจารณาดังกล่าวนี้อาจจะยังไม่เพียงพอที่จะนำคุณสมบัติที่มีอยู่มาใช้ให้เกิดประโยชน์ได้อย่างเต็มความสามารถที่มี ดังนั้นปัญหาความต้องการเครื่องมือสำหรับการทดสอบประสิทธิภาพ หรือคุณสมบัติด้านอื่น ๆ จึงมีบทบาทสำคัญในการพิจารณาเลือกใช้หรือเลือกซื้อเพื่อสนับสนุนประกอบการตัดสินใจ

คำสำคัญ: pfSense, RouterOS, Load Balance

1. บทนำ

จากความน่าสนใจของปัญหาพิจารณาเลือกใช้หรือเลือกซื้อเพื่อสนับสนุนประกอบการตัดสินใจ ได้นำไปสู่การพัฒนาสร้างเครื่องมือเพื่อทดสอบคุณสมบัติแต่ละด้านของเทคนิคการทำ Load Balancing ที่มาจากผู้พัฒนารายต่าง ๆ ซึ่งได้หยิบยกผลิตภัณฑ์ที่นำมาทดสอบจำนวน 2 ผลิตภัณฑ์ ได้แก่ pfSense เพราะเป็นแบบโอเพ่นซอร์ส สามารถดาวน์โหลดเพื่อใช้งานได้ฟรี และอีกผลิตภัณฑ์หนึ่งก็คือ Mikrotik RouterOS ที่

¹ นักศึกษาปริญญาโท หลักสูตรวิศวกรรมคอมพิวเตอร์และโทรคมนาคม วิทยาลัยนวัตกรรมการศึกษาและเทคโนโลยีและวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์

² ที่ปรึกษาวิทยานิพนธ์

ได้รับความนิยมในปัจจุบัน ในการทดลองนั้นจะนำเสนอในรูปแบบการจำลองสร้างเป็นระบบเครื่องเสมือน (Virtual machine) สำหรับติดตั้งใช้งานซอฟต์แวร์เพื่อทดลองสร้างสถานการณ์จำลอง (Scenario) เลียนแบบสภาพแวดล้อมที่สมมติจากกรณีต่าง ๆ เสมือนการใช้งานจริงเพื่อวัดสมรรถนะการทำงานในแต่ละด้าน ก่อนนำข้อมูลที่ได้มาวิเคราะห์เพื่อเปรียบเทียบหาข้อสรุปในเชิงประสิทธิภาพการทำงานด้านต่าง ๆ ของแต่ละผลิตภัณฑ์ ซึ่งผลจากการทดลองจะช่วยให้ผู้ดูแลระบบหรือผู้ใช้งานสามารถพิจารณาคัดเลือกผลิตภัณฑ์ให้เหมาะสมกับงานได้ตามต้องการ และการออกแบบทดลองระบบเสมือนจะแบ่งออกเป็นสามส่วนหลัก ๆ ได้แก่ โคล์เอนท์ เพื่อใช้จำลองการสร้างแพคเกจในการทดสอบ ส่วนที่สองเพื่อทดสอบคุณสมบัติการทำงานของ Load Balancing แต่ละผลิตภัณฑ์และเป็นตัวกลางการสื่อสารระหว่างโคล์เอนท์และเครื่องเซิร์ฟเวอร์ ส่วนสุดท้ายจะเป็นเซิร์ฟเวอร์เป็นตัวรับแพคเกจ และติดตั้งอพาเช่เว็บเซิร์ฟเวอร์ (Apache web server) ไว้ ปัจจุบันจัดได้ว่าเป็นมีความน่าเชื่อถือมากเนื่องจากเป็นที่นิยมใช้กันทั่วโลก

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

Server Load Balancing [1][2][3] หลากหลายรูปแบบการทำ Load Balance เพื่อเตรียมความพร้อมให้บริการแอปพลิเคชันต่าง ๆ โหลดที่ทำงานหนักมากบนเครือข่ายรวมทั้งเซิร์ฟเวอร์ซึ่งเป็นสาเหตุหลักที่ทำให้การบริการไม่ประสบผลสำเร็จส่งผลถึงความไม่พึงพอใจแก่ลูกค้า รวมทั้งภาพลักษณ์ขององค์กร การจัดการเว็บไซต์ต้องการประสิทธิภาพและความพร้อมที่ให้บริการข้อมูลแก่ลูกค้าอยู่เสมอ รวมทั้งความน่าเชื่อถือของเว็บไซต์นั้น ๆ อย่างไรก็ตามความพร้อมให้บริการของแอปพลิเคชันเหล่านี้จะถูกคุกคามโดยปัญหาเกี่ยวกับโหลดที่หนักมากของเครือข่ายรวมทั้งเซิร์ฟเวอร์ที่เป็นสาเหตุหลักที่ทำให้การให้บริการไม่ประสบผลสำเร็จ ปัญหาที่เกิดขึ้นบ่อย คือปริมาณการใช้งานทรัพยากรบนเซิร์ฟเวอร์ อย่างเช่น ปริมาณการใช้งานซีพียู รวมทั้งหน่วยความจำสูงเกินกว่าเซิร์ฟเวอร์จะรับได้ ผลก็คือประสิทธิภาพในการให้บริการของเซิร์ฟเวอร์ลดลง หรืออาจจะล่มได้ในที่สุด แนวทางการแก้ปัญหาได้แก่ การเพิ่มจำนวนเซิร์ฟเวอร์ที่มีเนื้อหาเหมือนกันหรือต่างกัน หรือมีความเกี่ยวข้องกันก็ได้มากกว่า 1 ตัวขึ้นไป ซึ่งอาจมีตั้งแต่ 2 ถึง 10 ตัวหรือมากกว่านี้ก็ได้ ซึ่งจะเรียกลักษณะเช่นนี้ว่าเซิร์ฟเวอร์ฟาร์ม (Server Farm) และวิธีการที่กระจายการร้องขอให้ใช้บริการของไคลเอนต์ (Client) ไปตามเซิร์ฟเวอร์ต่าง ๆ ลักษณะนี้จะเรียกว่าเซิร์ฟเวอร์โหลดบาลานซ์ (Server Load Balancing) จึงมีความสำคัญ และอุปกรณ์ที่เกี่ยวข้องนี้จะเรียกว่าเซิร์ฟเวอร์โหลด บาลานซ์ เซอร์ (Server Load Balancer) จุดประสงค์ของการเพิ่มเซิร์ฟเวอร์เข้ามาให้บริการเป็นกลุ่มหรือเรียกว่าเซิร์ฟเวอร์ฟาร์ม ก็เพื่อให้การบริการข้อมูลข่าวสารแก่ผู้ใช้งานมีความต่อเนื่อง น่าเชื่อถือ มีความรวดเร็ว ถึงแม้ว่าเซิร์ฟเวอร์ตัวใดตัวหนึ่งเกิดล่ม ตัวอื่น ๆ ก็สามารถทำงานแทนได้ทันที นอกจากนี้ในมุมมองประสิทธิภาพคือการลดภาระการใช้ทรัพยากรบนเซิร์ฟเวอร์ได้ โดยการกระจายคำร้องขอใช้บริการแก่ผู้ใช้งานหลายคนได้ในเวลาเดียวกัน

pfSense [4][5] เป็นโปรเจกต์ที่พัฒนาโดย Chris Buechler และ Scott Ullrich ถูกพัฒนาขึ้นจาก Linux สายพันธ์ FreeBSD เมื่อปี 2004 จุดประสงค์เพื่อใช้งานเป็นไฟร์วอลล์ และเราเตอร์ และต้องสามารถจัดการตัวอุปกรณ์ ได้ผ่านหน้า Browser (เช่น IE, Firefox, Chrome ฯลฯ) ได้ และด้วยเนื่องจากตัว pfSense ถูกพัฒนามาจาก Linux ทำให้เราสามารถใช้งานได้ฟรีโดยไม่มีค่าใช้จ่ายใด ๆ แบบไม่ต้องกังวลเรื่อง License และมันยังสามารถใช้งานได้ค่อนข้างเกือบจะเทียบเท่าตัวที่เป็นแบบ Next Generation Firewall แบบ Enterprise ที่ใช้กันตามบริษัทต่าง ๆ (ตอนนี้ตามตลาดไฟร์วอลล์ระดับสูง ก็จะเป็นพวก Palo Alto, Checkpoint, Cisco, Juniper, Fortinet เป็นต้น)

Mikrotik RouterOS [6][7][8] เป็นบริษัทที่ผลิตซอฟต์แวร์ และ ฮาร์ดแวร์ Router โดย Mikrotik มาจากภาษา Latvia ประเทศ Latvia เป็นประเทศขนาดเล็กที่อยู่ทางยุโรปตะวันออก ซึ่งตัว Mikrotik เกิดจากการผสมคำสองคำด้วยกัน ระหว่าง Mikro (ภาษาอังกฤษคือ Micro) + Tik (ภาษาอังกฤษคือ Network) ถ้าแปลง่าย ๆ จากภาษาอังกฤษแปลว่าเครือข่ายขนาดเล็ก (Micro Network) โดยบริษัทได้ก่อตั้งขึ้นในปี 1996 โดย John Trully & Arnis Reikstins โดย Mikrotik (ไมโครติก) หรือ เราเตอร์บอร์ด คือ อุปกรณ์อัจฉริยะที่มีความสามารถในการใช้งานที่หลากหลายรูปแบบทั้งยังสามารถปรับเปลี่ยนหรือสามารถดัดแปลงการตั้งค่าต่าง ๆ ได้ค่อนข้างอิสระ จึงทำให้ในปัจจุบันเป็นที่นิยมอย่างแพร่หลายในการนำไปใช้งานในระบบเครือข่าย ทั้งในระดับบุคคลหรือองค์กร อาทิเช่น มหาวิทยาลัย, บริษัทต่าง ๆ เป็นต้น ซึ่งอุปกรณ์ของ MikroTik ก็มีให้เลือกหลากหลายรุ่นตามวัตถุประสงค์ในการใช้งานหรือการรองรับการใช้งานที่แตกต่างกันไป โดยจุดเด่นที่สำคัญของ Mikrotik คือ ผู้ใช้สามารถปรับแต่งและตั้งค่าฟังก์ชันต่าง ๆ ได้อย่างอิสระเพื่อให้สอดคล้องกับการใช้งาน ตัว Mikrotik Router มีการออกแบบให้รองรับกับงานด้านบริหารจัดการด้าน Network โดยตัว Mikrotik Router มีราคาถูก ด้านการใช้งานไม่จำเป็นต้องพึ่ง Sever PC ทำให้ประหยัดไฟและประหยัดพื้นที่การใช้งาน นอกจากนี้ยังมีขนาดเราท์เตอร์ ที่เล็กและสะดวกต่อการติดตั้งได้ง่าย รวมทั้งมีฟังก์ชัน Authenticate ที่ทำให้ออกก่อนการใช้งาน Hotspot wifi ต้องมีการ Login ด้วย Username และ Password เพื่อเข้าสู่ระบบ หรือเรียกง่าย ๆ ว่าต้องล็อกอินเพื่อออกเน็ตก่อนเพื่อความปลอดภัยในการใช้งาน Internet wifi ในองค์กรต่าง ๆ ในส่วนของการออกใบอนุญาตของ Mikrotik RouterOS มีให้บริการในรูปแบบการออกใบอนุญาตที่แตกต่างกัน 6 แบบโดยมีราคาตั้งแต่ฟรีจนถึงประมาณ 250 (เหรียญสหรัฐ ณ วันที่เผยแพร่) ถ้าแปลงเป็นเงินไทยก็ราว ๆ 8,000 กว่าบาท ดังนั้นจะมีระบุจำนวนคุณสมบัติรวมไว้ด้วย

Apache JMeter [9] เป็น Java Application ที่ถูกพัฒนามาเพื่อใช้สำหรับการทดสอบโหลดของพฤติกรรมการใช้งาน และวัดประสิทธิภาพ JMeter สามารถจำลองการสถานการณ์ในการทดสอบ โดยการกำหนดจำนวน Request ค่าต่าง ๆ ได้ และสามารถแสดงผลออกมาในรูปแบบของกราฟ หรือเป็นไฟล์ CSV ได้ ในส่วนของโปรโตคอลที่เป็นจุดเด่นของ JMeter ค่อนข้างหลากหลาย เช่น Web (HTTP,HTTPS), SOAP, FTP, Database, LDAP, JMS, TCP, Mail (SMTP) และในงานวิจัยนี้นำมาเป็นเครื่องมือในการทดสอบ Load Balancing คำจำกัดความของ Load testing, Performance testing และ Stress testing นั้นแตกต่างกันไปตามลักษณะการทำงาน และการใช้งานของแต่ละบริษัท ซึ่งจะอธิบายดังต่อไปนี้

Performance testing คือการทดสอบใด ๆ ที่วัดการตอบสนองและความเสถียร ภายใต้ปริมาณการใช้งานที่กำหนดไว้

Load Testing คือการทดสอบระบบด้วยการสร้างปริมาณการใช้งานเข้าไปและวัดการตอบสนองของระบบ

Stress testing ระบบรองรับโหลดที่มากเกินไปจนเกินขีดความสามารถได้สูงสุดเท่าไรก่อนที่ระบบจะล่ม หรือล่มแล้วสามารถกลับมาอยู่ในสภาพพร้อมทำงานใหม่ได้หรือไม่

2.2 งานวิจัยที่เกี่ยวข้อง

เครือข่ายคอนเทนเนอร์เน็ตเวิร์ก [10]

พรรรรักษา กมลเวชช์ และชัยพร เขมะภาคะพันธ์ได้ทำการประเมินประสิทธิภาพการทำงานของวิธีการเชื่อมโยงของคอนเทนเนอร์เน็ตเวิร์ก พบว่าวิธีการเชื่อมโยงแบบ Calico มีประสิทธิภาพการทำงานที่ดีกว่าทางด้านเวลาในการตอบสนองกลับมายังผู้ใช้บริการ ปริมาณการดาวน์โหลดข้อมูล ปริมาณ Throughput ยกเว้นการทำงานของ CPU ที่เครื่องมาสเตอร์และเครื่องมินิเนียนที่วิธี Calico มีการทำงานมากกว่าเนื่องจากลักษณะของ Calico เป็นการจำลองเน็ตเวิร์กในระดับเลเยอร์ 3 และไม่มีการทำงาน Encapsulation หรือ Tunnel ทำให้มี Overhead ส่งผลให้มีประสิทธิภาพการทำงานที่ดีกว่า และยังเป็นวิธีที่เหมาะสมกับกรณีที่ผู้ใช้งานมีเป็นจำนวนมาก จากการวิเคราะห์งานวิจัยนี้ได้ทำการติดตั้งระบบที่ใช้ในการทดสอบ โดยใช้โปรแกรม Apache JMeter เพื่อใช้เป็นเครื่องมือที่ใช้สำหรับจำลองพฤติกรรมการร้องขอเข้าใช้บริการ (Request) จากเครื่องโฮสต์ไปยังเครื่องมาสเตอร์ผ่านทางโปรโตคอล HTTP โดยมีการกำหนดให้มีการร้องขอเข้าใช้บริการไปยัง Web Server จำนวน ต่าง ๆ และนำมาวิเคราะห์ผลการทดลอง

Load Balance Web Server [11]

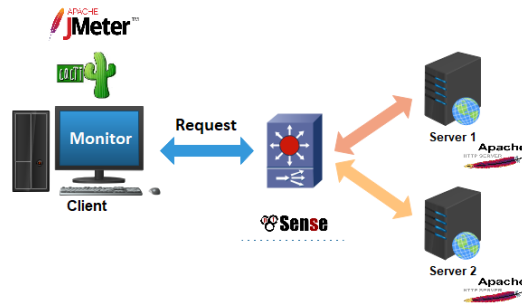
โชติมา ชาญนุกุลได้ทำการทดสอบการกระจายงานของโหลดบาลานซ์ ได้ทำการทดสอบอยู่ 2 ลักษณะ คือการเพิ่มจำนวนการเข้าใช้บริการของ Client และเพิ่ม Private Server โดยการจำลองระบบเพื่อใช้ในการทดสอบการกระจายการร้องขอของ Web Server ให้สมดุล และใช้ทฤษฎี Round Robin DNS นำมาประยุกต์ใช้บางส่วนร่วมกับโปรแกรมการนับ Session ที่เชื่อมต่อกับเซิร์ฟเวอร์ ทำให้ประสิทธิภาพของโปรแกรมในระบบนั้น สามารถทำการกระจายจำนวนการเข้าใช้บริการไปยัง Private Server ได้ แต่ไม่สามารถเช็คภาระการทำงานของระบบได้ เช่น สถานะของ CPU Usage และ Memory Usage ได้ ข้อดีที่ได้คือสามารถนับจำนวนการเข้าใช้บริการได้ โปรแกรมใช้งานง่ายและสะดวกต่อการติดตั้ง รวมถึงประหยัดค่าใช้จ่ายในการลงทุนซื้ออุปกรณ์โหลดบาลานซ์ที่มีราคาแพง จากการวิเคราะห์งานวิจัยนี้ เพื่อศึกษาการเปรียบเทียบและทำงานของการแสดงการเข้าใช้บริการต่อการกระจาย Load Balancing โดยมีการเปรียบเทียบมีการเข้าใช้งานพร้อมกัน และไม่พร้อมกัน ผลที่ได้คือมีการกระจายโหลดได้ดี สามารถนับจำนวนการเข้าใช้บริการได้ แต่ไม่สามารถเช็คสถานะของ CPU และ RAM ได้

3. ขั้นตอนการดำเนินงานวิจัย

แผนการดำเนินงานนำเสนอเป็นไปตามลำดับที่เริ่มจากภาพโดยรวมของระบบ เครื่องมือที่นำมาประยุกต์ใช้ และขั้นตอนการดำเนินงาน โครงสร้างและภาพรวมของระบบแบ่งออกเป็น 2 แบบ ดังต่อไปนี้

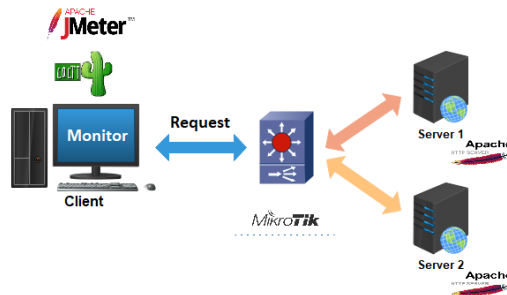
3.1 โครงสร้างและภาพรวมของระบบ

1. Load Balancing โดยใช้ซอฟต์แวร์ pfSense



ภาพที่ 1 โครงสร้างภาพรวมระบบโดยใช้ pfSense

2. Load Balancing โดยใช้ RouterOS



ภาพที่ 2 โครงสร้างภาพรวมระบบโดยใช้ RouterOS

3.2 แผนการดำเนินงาน

แผนการดำเนินงาน เพื่อเป็นกรอบการดำเนินงานวิจัยเพื่อให้เสร็จลุล่วงได้ตามวัตถุประสงค์ ซึ่งใช้เป็นแนวทางปฏิบัติงาน

1. เตรียมการกำหนดหัวข้อสารนิพนธ์
2. ศึกษาเอกสาร และงานวิจัยที่เกี่ยวข้อง
3. ค้นคว้าและออกแบบเครื่องมือในการทดสอบ โดยเริ่มจากตัวอย่างเครื่องมือที่ใช้ในการทดสอบอุปกรณ์อื่น ๆ
4. กำหนดกรอบเวลาในการดำเนินงาน
5. ดำเนินงานพัฒนาและจัดสร้างระบบจำลองเพื่อการทดสอบ

6. ทดสอบและบันทึกผลการทดลอง

7. วิเคราะห์ผลการทดลอง ช่วงนี้อาจจะการทดลองผิดพลาดถูกเครื่องมือที่นำมาใช้ในการทดสอบ เมื่อผลที่ได้ออกมาไม่เป็นไปตามแนวทางที่ต้องการ

8. สรุปผลการทดลอง

9. จัดทำสารนิพนธ์ฉบับสมบูรณ์

3.3 เครื่องมือที่ใช้ในงานวิจัย

ประเมินความต้องการของระบบ อ้างอิงมาจากโครงสร้างพื้นฐานของเครื่องมือสำหรับทดสอบ เป็นไปตามแนวทางที่ออกแบบไว้ โดยได้อธิบายถึงวิธีการปฏิบัติงาน ที่เริ่มจากการจัดหาและการจัดเตรียมความพร้อม ทั้งในส่วนของฮาร์ดแวร์ ที่เป็นการกำหนดคุณสมบัติให้เพียงพอสำหรับรองรับการเป็นพื้นที่หลักสำหรับจัดวางเครื่องคอมพิวเตอร์จำลองในส่วนต่าง ๆ และส่วนการติดตั้งซอฟต์แวร์ เพื่อกระบวนการในการทดสอบ ซึ่งระบุไว้ตามโครงสร้างของระบบในทุก ๆ ส่วน และจะนำเสนออยู่ 3 ส่วน คือส่วนแรกเป็นส่วนของฮาร์ดแวร์ เช่นเครื่องคอมพิวเตอร์จำลอง ที่ใช้ในการทดสอบ ส่วนที่สองคือการตั้งค่าทรัพยากรที่จะใช้ แบ่งตามความต้องการของเครื่องคอมพิวเตอร์จำลองที่จะใช้ในการทดสอบ และส่วนที่สามคือซอฟต์แวร์ จะระบุว่าในงานวิจัยนี้จะใช้ซอฟต์แวร์ตัวไหน เวอร์ชันไหนบ้าง รวมถึงหน้าที่การทำงานของแต่ละส่วนด้วย ดังแสดงในตารางที่ 1

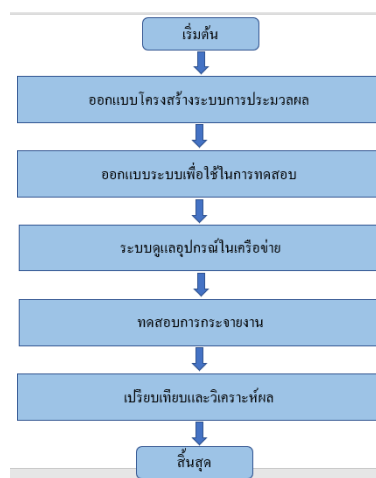
ตารางที่ 1 รายละเอียดข้อมูลอุปกรณ์ฮาร์ดแวร์และการกำหนดค่า

อุปกรณ์	ผลิตภัณฑ์	รายละเอียด	หน้าที่
ฮาร์ดแวร์	DELL Notebook	CPU: Inter® Core™ i5-5200U Ram: 8 GB HDD: 1 TB OS: Window 10 (64 Bit)	โครงสร้างพื้นฐานหลักของระบบ ใช้ติดตั้งซอฟต์แวร์
Virtual machine	pfSense Version 2.3.3	CPU: 1 Core (X86), RAM: 1 GB, HDD: 20 GB Lan to Client: 1 Gbps Lan to Server 1: 1 Gbps Lan to Server 2: 1 Gbps	สำหรับ Load Balancing เพื่อกระจายงาน ให้กับ Server 1 และ Server 2
	Mikrotik RouterOS Version 2.3.3	CPU: 1 Core (X86), RAM: 1 GB, HDD: 20 GB Lan to Client: 1 Gbps Lan to Server 1: 1 Gbps Lan to Server 2: 1 Gbps	สำหรับ Load Balancing เพื่อกระจายงาน ให้กับ Server 1 และ Server 2
	Client (Windows 7) Version Starter	CPU: 1 Core (X86), RAM: 512 MB, HDD: 20 GB Lan to LB: 1 Gbps	สำหรับการทดสอบรับและส่ง Packet (User Request)
	Server 1	CPU: 1 Core (X86), RAM: 1 GB, HDD: 20 GB Lan to LB: 1 Gbps	สำหรับเป็น Web Server
	Server 2	CPU: 1 Core (X86), RAM: 1 GB, HDD: 20 GB Lan to LB: 1 Gbps	สำหรับเป็น Web Server

ตารางที่ 2 รายละเอียดข้อมูลซอฟต์แวร์ต่าง ๆ

อุปกรณ์	ผลิตภัณฑ์	หน้าที่
ซอฟต์แวร์	VMware Workstation Pro 12.0	สำหรับสร้างเครื่องคอมพิวเตอร์จำลองทำงานแบบเครื่องเสมือน เพื่อทดสอบและพัฒนาระบบ
	Apache JMeter Version 4.0	ใช้จำลองการสร้างแพคเกจ และวัดผลการทดลอง
	Cacti Version 1.1.38	ใช้ในการมอนิเตอร์ CPU และ Memory Usage
	Apache Web Server Version 2.2.8	ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์สำหรับรับเว็บแอปพลิเคชัน
	Win box Version 3.7	สำหรับการตั้งค่า RouterOS (Mikrotik)

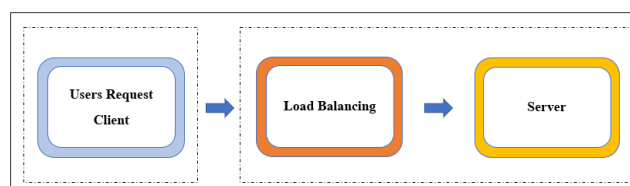
3.4 ขั้นตอนและวิธีการดำเนินงาน



ภาพที่ 3 แผนผังการดำเนินงานโดยรวมของระบบ

1. ออกแบบโครงสร้างระบบการประมวลผล

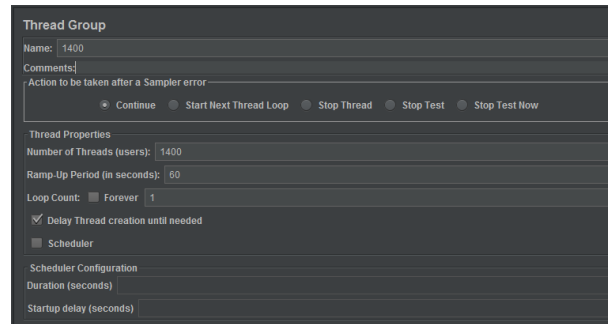
แนวทางกำหนดการทำงานออกเป็น 3 ส่วน อธิบายถึงความเกี่ยวข้องการทำงานร่วมกันของทั้งระบบ จำเป็นต้องแสดงให้เห็นถึงการเชื่อมต่อกันตามตำแหน่งที่ถูกระบุไว้ ให้มีการประสานทำงานร่วมกันของระบบ เกิดเป็น โครงสร้าง โดยรวมของเครื่องมือที่ใช้วัดทดสอบประสิทธิภาพการทำงานได้อย่างครบถ้วน รวมไปถึงการแสดงผลภาพรวมที่เกิดขึ้นในกระบวนการทำงานจากต้นทางจนกระทั่งถึงปลายทาง ดังแสดงในภาพที่ 4



ภาพที่ 4 แนวคิดออกแบบ โครงสร้างหลักของระบบ

2. ออกแบบระบบเพื่อใช้ในการทดสอบ

การกำหนด Number of Threads (users) จำนวนผู้ใช้งานที่ต้องการ ณ เวลานั้น Ramp-up Period (in seconds) คือเวลาหน่วงในการเพิ่มผู้ใช้งานจนถึงจำนวน Number of Threads ที่ตั้งไว้ ยกตัวอย่างเช่น Number of Threads = 1400, Ramp-up Period = 60 แสดงว่า ภายใน 60 วินาที จะมีจำนวนผู้ใช้งานระบบรวม 1400 คน (ค่อย ๆ เพิ่มเข้าไปจนครบ 1400 ใน 60 วินาที) และ Loop Count รวมไปถึงการทดสอบหาค่า Response time ของระบบด้วย



ภาพที่ 5 การกำหนดค่าทดสอบโดยใช้ Apache JMeter

3. เครื่องมือดูแลอุปกรณ์บนระบบเครือข่าย

Cacti เป็นหนึ่งในเครื่องมือการดูแลอุปกรณ์บนระบบเครือข่าย ที่เป็น Open source สามารถดาวน์โหลดได้โดยที่จะแสดงผลเป็นลักษณะของกราฟที่สามารถเข้าใจได้ง่ายและสามารถจัดการกราฟของแต่ละอุปกรณ์ให้เป็นหมวดหมู่รวมถึงเลือกดูกราฟย้อนหลังได้ โดยมีการส่งผ่านข้อมูลจากเครื่องในระบบเครือข่ายมายังเครื่องที่ติดตั้ง Cacti ด้วยโปรโตคอล SNMP

4. ทดสอบการกระจายงานของ Load Balancing

ในส่วนนี้จะทำการนับ Connection ของการรับส่งแพคเกจระหว่าง Load Balancer กับเครื่อง Server โดยใช้ Command ในการนับจำนวน Request ใน Apache access log โดยมีลำดับการดำเนินการดังต่อไปนี้

1. Stop apache

```
$sudo /etc/init.d/apache2 stop
```

2. Remove access log

```
$sudo rm /var/log/apache2/access.log
```

3. Start apache

```
$sudo /etc/init.d/apache2 start
```

4. JMeter request

5. Count access log

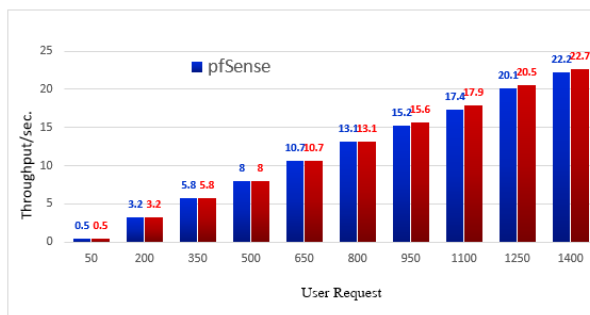
```
$cat /var/log/apache2/access.log | wc -l
```


4.1 วิเคราะห์ผลด้าน Throughput ต้องการผลของปริมาณงานสำหรับการรับส่งข้อมูล

การทดสอบกำหนดให้มีการเข้าใช้บริการ (Number of Threads (user)) โดยกำหนดค่าดังต่อไปนี้ 50, 200, 350, 500, 650, 800, 950, 1100, 1250 และ 1400 Request เพื่อป้องกันความคลาดเคลื่อนและความถูกต้องของข้อมูลที่สุด จึงมีการทดสอบจำนวน 10 ครั้ง และเพื่อให้ง่ายต่อการบันทึกและวิเคราะห์ผล ซึ่งจะยกตัวอย่างผลการทดสอบที่ได้จากโปรแกรม JMeter มาแสดงผล

ตารางที่ 3 บันทึกผลทดสอบทางด้าน Throughput

Users Request	Throughput/sec.	
	pfSense	Mikrotik
50	0.5	0.5
200	3.2	3.2
350	5.8	5.8
500	8.0	8.0
650	10.7	10.7
800	13.1	13.1
950	15.2	15.6
1100	17.4	17.9
1250	20.1	20.5
1400	22.2	22.7



ภาพที่ 6 กราฟแสดงผลการทดสอบด้าน Throughput

จากการทดสอบดังกล่าวถ้าเปรียบเทียบ User request ตั้งแต่ 950 ขึ้นไป จะเห็นว่า Mikrotik RouterOS จะเริ่มมี Throughput มากกว่า pfSense จึงสามารถสรุปได้ว่า Mikrotik RouterOS สามารถทำงานในด้าน Load Balancing ได้เร็วกว่า pfSense เนื่องจากตัวซอฟต์แวร์ของ Mikrotik RouterOS ออกแบบมาเพื่อที่จะทำงานเป็น Router โดยเฉพาะ จึงมีความสามารถในการทำงานด้าน Throughput ได้ดีกว่า pfSense เนื่องจาก

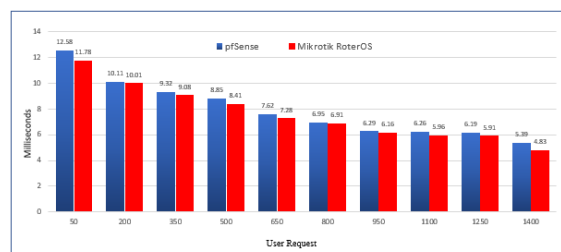
pfSense พัฒนามาจาก FreeBSD ซึ่งเป็นระบบปฏิบัติการสำหรับทำเป็นเซิร์ฟเวอร์ จึงทำให้การทำงานของ pfSense มีความล่าช้าเมื่อมี Packet เพิ่มมากขึ้น

4.2. วิเคราะห์ผลด้านเวลา (Response time) ในการตอบสนองกลับมายังผู้ใช้บริการ

การทดลองกำหนดให้มีการเข้าใช้บริการจำนวน (Number of Threads (user)) โดยกำหนดค่าดังต่อไปนี้ 50, 200, 350, 500, 650, 800, 950, 1100, 1250 และ 1400 Request โดยมีการทดสอบจำนวน 10 ครั้ง ในแต่ละครั้งจะทำการทดสอบจำนวน 3 ครั้ง แล้วหาค่าเฉลี่ยเพื่อให้ได้ข้อมูลที่ถูกต้อง ครบถ้วน และสมบูรณ์ที่สุดในการบันทึกผล

ตารางที่ 4 บันทึกผลการทดสอบทางด้านเวลา (Response time)

Users Request	Response time (ms)	
	pfSense	Mikrotik
50	12.58	11.78
200	10.11	10.01
350	9.32	9.08
500	8.85	8.41
650	7.62	7.28
800	6.95	6.91
950	6.29	6.16
1100	6.26	5.96
1250	6.19	5.91
1400	5.39	4.83



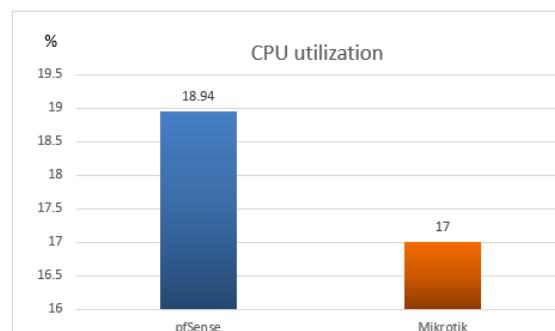
ภาพที่ 7 วิเคราะห์ผลทางด้านเวลา Response time

จากการทดสอบดังกล่าวเป็นการแสดงค่า Response time ที่อยู่ในรูปแบบกราฟ ซึ่งกราฟที่แสดงนั้นได้มาจากโปรแกรม Apache JMeter จะเก็บค่า Response time ทุก ๆ ช่วงเวลา 10 Millisecond และเพื่อการวิเคราะห์ผลการทดสอบสามารถวิเคราะห์ได้ง่ายขึ้น จะทำการบันทึกผลที่ได้จากโปรแกรม Apache JMeter เป็นรูปแบบของไฟล์ Excel โดยนำมาหาค่าเฉลี่ยเพื่อใช้ในการวิเคราะห์ที่ง่ายขึ้น ผลการทดสอบด้านเวลา (Response Time) ที่ได้เก็บข้อมูลมา จะเห็นได้ว่า Mikrotik RouterOS มีการใช้เวลาในการตอบสนองเร็วกว่า pfSense ที่ใช้ เนื่องจากการทำงานของซอฟต์แวร์ของ Mikrotik (เป็น Router OS) ส่วน pfSense ถูก

พัฒนามาจาก FreeBSD ซึ่งเป็น UNIX ออกแบบมาเพื่อที่จะทำงานเป็นเซิร์ฟเวอร์ จึงทำให้ Mikrotik ที่ ออกแบบมาเพื่อเป็น Router โดยตรงสามารถลดระยะเวลาในการประมวลผลได้ดีกว่า

4.3 วิเคราะห์ผลการทดสอบด้าน CPU utilization

ผลการทดสอบสมรรถนะด้าน CPU utilization จะทดสอบจากการบันทึกผลการทดลอง ณ เวลา หนึ่ง ๆ ที่มีการทดสอบแบบต่อเนื่อง ซึ่งจะ Monitor โดยใช้โปรแกรม Cacti โดยจะนำค่า CPU utilization ที่มี ค่าสูงสุดมาวิเคราะห์ผลการทดลอง

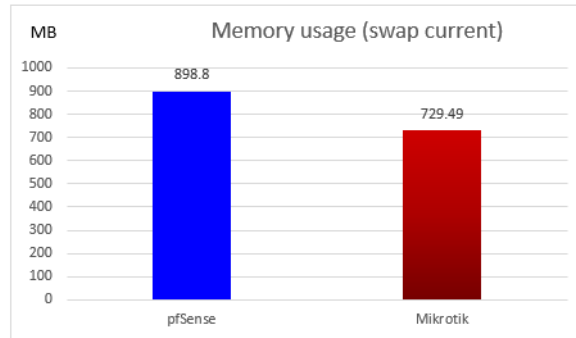


ภาพที่ 8 ผลการทดสอบสมรรถนะด้าน CPU utilization

จากการวิเคราะห์ผลการทดสอบสมรรถนะด้าน CPU utilization ที่บันทึกผลมาจาก โปรแกรม Cacti ในช่วงของการใช้งานระยะเวลาหนึ่ง จะเห็นได้ว่า Mikrotik RouterOS มีการใช้งาน CPU ที่ 17 เปอร์เซ็นต์ (%) ซึ่งมีค่าน้อยกว่าการใช้งาน CPU ของ pfSense ที่มีการใช้งาน 18.94 เปอร์เซ็นต์ ที่มีการ ทดสอบช่วงของการใช้งานระยะเวลาหนึ่ง เนื่องจากการทำงานของซอฟต์แวร์ของ Mikrotik (เป็น Router OS) ส่วน pfSense ถูกพัฒนามาจาก FreeBSD ซึ่งเป็น UNIX ออกแบบมาเพื่อที่จะทำงานเป็นเซิร์ฟเวอร์ จึงทำ ให้ Mikrotik ที่ออกแบบมาเพื่อเป็น Router โดยตรงสามารถลดระยะเวลาในการประมวลผลได้ดีกว่า

4.4 วิเคราะห์ผลการทดสอบด้าน Memory usage

ผลการทดสอบสมรรถนะด้าน Memory usage จะทดสอบจากการบันทึกผลการทดลอง ณ ช่วงเวลาหนึ่ง ๆ ที่มีการทดสอบแบบต่อเนื่อง เป็นการเก็บค่าจากการ Monitor โดยใช้โปรแกรม Cacti ซึ่งจะ นำค่า Memory usage ที่มีค่าสูงสุดมาวิเคราะห์ผลการทดลอง ดังแสดงในภาพที่ 9

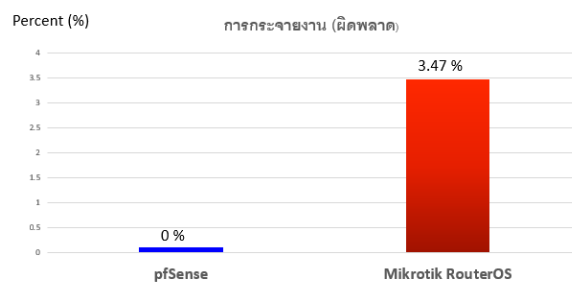


ภาพที่ 9 ผลการทดสอบสมรรถนะด้าน Memory usage

จากการวิเคราะห์ผลการทดสอบสมรรถนะด้าน Memory usage ที่บันทึกผลมาจากโปรแกรม Cacti ในช่วงของการใช้งานระยะเวลาหนึ่ง จะเห็นได้ว่า Mikrotik RouterOS มีการใช้งาน Memory ที่ 729.49 เมกะไบต์ (MB) ซึ่งมีย่านน้อยกว่าการใช้งาน Memory ของ pfSense ที่มีการใช้งาน 898.8 เมกะไบต์ ที่มีการทดสอบช่วงของการใช้งานระยะเวลาหนึ่ง เนื่องจากการทำงานของซอฟต์แวร์ของ Mikrotik (เป็น Router OS) ส่วน pfSense ถูกพัฒนามาจาก FreeBSD ซึ่งเป็น UNIX ออกแบบมาเพื่อที่จะทำงานเป็นเซิร์ฟเวอร์ จึงทำให้ Mikrotik ที่ออกแบบมาเพื่อเป็น Router โดยตรงสามารถลดระยะเวลาในการประมวลผลได้ดีกว่า

4.5 วิเคราะห์ผลการเปรียบเทียบการกระจายงานของ Load Balancing

ในส่วนนี้จะทำการวิเคราะห์ผลการทดสอบการกระจายงานของตัว pfSense และ Mikrotik RouterOS เพื่อจะรู้ว่าซอฟต์แวร์ตัวไหนจะมีการกระจายงานได้ดีกว่ากัน โดยใช้การนับ Connection ที่เครื่อง Server และผลการทดสอบดังแสดงในภาพที่



ผลการทดสอบสรุปได้ว่า pfSense มีคุณสมบัติการกระจายงานของ Load Balancing ได้ดีกว่า Mikrotik RouterOS เนื่องจากการทำงานของซอฟต์แวร์ของ pfSense มีการออกแบบมาให้มีความเสถียรภาพด้านการกระจายงานได้ดี

5. สรุปผลการวิจัย

ในส่วนนี้นำเสนอสรุปผลการทดลอง ซึ่งแบ่งออกได้เป็นสองหัวข้อย่อยคือ การสรุปผลการดำเนินงาน และการสรุปและอภิปรายผลการวิจัย ดังนี้

5.1 สรุปผลการดำเนินงาน

การดำเนินงานครั้งนี้บรรลุผลได้ตามวัตถุประสงค์ตามที่ระบุไว้ดังนี้

1. สามารถเข้าถึงวิธีการทำงานของเทคโนโลยี Load Balancing ในรูปแบบที่ใช้งานโดยซอฟต์แวร์ pfSense และ Mikrotik RouterOS

2. สามารถออกแบบและเข้าถึงวิธีการใช้งานเครื่องมือในการทดสอบ เช่น โปรแกรม Apache JMeter เพื่อทดสอบและวัดผลได้

3. ทำการประเมินและวิเคราะห์ผลการทดสอบ Load Balancing โดยใช้ซอฟต์แวร์ pfSense และ Mikrotik RouterOS ได้

5.2 สรุป และอภิปรายผลการวิจัย

การวิเคราะห์ผลการทดสอบในด้านต่าง ๆ ของ Load Balancing แต่ละผลิตภัณฑ์ สามารถสรุปและแยกการอธิบายออกเป็นส่วนต่าง ๆ ตามเกณฑ์การวัดผลที่กำหนดไว้ดังนี้

1. การทดสอบสมรรถนะทางด้าน Throughput เพื่อวัดผลปริมาณจำนวนงานที่ทำได้สำเร็จต่อหนึ่งหน่วยเวลาสรุปได้ว่า Mikrotik RouterOS มีจำนวน Throughput ที่มากกว่า Load Balancing แบบ pfSense เนื่องจากคุณสมบัติของซอฟต์แวร์ที่ออกแบบมาเป็น RouterOS จึงมีคุณสมบัติการทำงานในด้าน Throughput ที่ดีกว่า

2. การทดสอบสมรรถนะทางด้าน Response time เพื่อวัดผลเวลาที่เซิร์ฟเวอร์มีการตอบสนองจะเห็นได้ว่า Mikrotik RouterOS มี Response time ที่ดีกว่า เนื่องจากการทำงานของซอฟต์แวร์ของ Mikrotik เป็น Router OS ส่วน pfSense ถูกพัฒนามาจาก FreeBSD ซึ่งเป็น UNIX ออกแบบมาเพื่อที่จะทำงานเป็นเซิร์ฟเวอร์ จึงทำให้ Mikrotik RouterOS ที่ออกแบบมาเพื่อเป็น Router โดยตรงสามารถลดระยะเวลาในการประมวลผลได้ดีกว่า

3. การทดสอบสมรรถนะทางด้าน CPU Utilization สำหรับชี้วัดการใช้ประโยชน์จากหน่วยประมวลผลสรุปได้ว่า Mikrotik RouterOS มีการใช้ทรัพยากร CPU ในการประมวลผลจำนวนต่ำกว่าในหนึ่งหน่วยเวลาที่ทดสอบ เนื่องจากคุณสมบัติทางด้านซอฟต์แวร์ที่ออกแบบมา

4. การทดสอบสมรรถนะทางด้าน Memory usage สำหรับวัดผลการใช้ประโยชน์จากหน่วยความจำสรุปได้ว่า Mikrotik RouterOS มีการใช้ทรัพยากร Memory ในการประมวลผลจำนวนต่ำกว่าในหนึ่งหน่วยเวลาที่ทดสอบ เนื่องจากคุณสมบัติทางด้านซอฟต์แวร์ที่ออกแบบมา

5. การทดสอบการกระจายงานของ โดยใช้วิธีการทดสอบด้วยอัลกอริทึมแบบ Round Robin ซึ่งจะไม่สนใจ Connection ของเครื่อง และจะเห็นได้ว่า pfSense มีคุณสมบัติการกระจายงานได้ดีกว่า Mikrotik RouterOS เนื่องจากการทำงานของซอฟต์แวร์ในการกระจายงานที่มีความเสถียรภาพมากกว่า ซอฟต์แวร์

pfSense เหมาะกับการนำไปใช้ในองค์กรที่ทรัพยากรของเครื่องเซิร์ฟเวอร์ที่เท่ากัน ส่วน Mikrotik RouterOS ที่นำมาใช้เป็น Load Balancer นั้น ไม่สามารถควบคุม Request ให้แบ่งไปตามเซิร์ฟเวอร์ในปริมาณที่เท่า ๆ กันได้ นั่นอาจเกิดปัญหาเซิร์ฟเวอร์ทำงานเกินความสามารถ ทำให้เกิด คอขวด สำหรับ Request ที่เรียกใช้งานไปยังเซิร์ฟเวอร์ที่มีการทำงานหนักอยู่

5.3 ข้อจำกัดของงานวิจัย และแนวทางแก้ไข

เนื่องจากการจำลองทดสอบประสิทธิภาพ Load Balancing บนเครื่องเสมือนในเครื่องคอมพิวเตอร์ของผู้วิจัย ซึ่งมีหน่วยความจำและหน่วยประมวลผลจำกัด ทำให้การทดสอบในครั้งนี้ไม่สามารถทำงานได้อย่างเต็มที่ ส่งผลให้ระบบไม่สามารถตอบสนองต่อการประมวลผลได้ตามต้องการได้ เนื่องจากข้อจำกัดทางด้านฮาร์ดแวร์ ซึ่งแนวทางแก้ไขนั้น อาจจะทำการทดสอบบนเครือข่ายที่ใช้งานอยู่จริง หรือปรับปรุงประสิทธิภาพด้านฮาร์ดแวร์ให้มีทรัพยากรสูงขึ้น เพื่อที่จะทำให้สามารถวัดประสิทธิภาพการทำงานของระบบ Load Balancing ได้สมบูรณ์มากยิ่งขึ้น และสามารถทดสอบจำนวน User Request ที่สูงขึ้นได้

5.4 ข้อเสนอแนะและแนวทางการพัฒนาในอนาคต

ในงานวิจัยนี้เป็นการทดสอบบนเครื่องคอมพิวเตอร์ของผู้วิจัยซึ่งมีข้อจำกัดอยู่หลายประการ เช่น ทรัพยากรของเครื่อง จึงขอเสนอแนะแนวทางในการวิจัยต่อยอดสำหรับผู้สนใจนำไปพัฒนาต่อดังนี้

1. ทดสอบในสภาพแวดล้อมอื่น ๆ เช่น การทดสอบบนเครือข่ายที่ใช้งานอยู่จริง หรือทดสอบผ่านซอฟต์แวร์และอุปกรณ์ Load Balancer ต่าง ๆ ซึ่งปัจจุบันมีการพัฒนาอยู่หลายผลิตภัณฑ์ ทั้งแบบที่มีให้ใช้งานฟรีและแบบที่มีราคาถูก กล่าวคือนำผลิตภัณฑ์ต่าง ๆ เหล่านี้ที่ได้รับคามนิยมเลือกใช้งานมาทำการทดสอบ ตามการทดสอบที่ได้กล่าวไว้ข้างต้น

2. ทดสอบทางด้านปัจจัยอื่น ๆ เพื่อสามารถวิเคราะห์ประสิทธิภาพด้านอื่น ๆ ได้อย่างถูกต้องมากยิ่งขึ้น เช่นวิเคราะห์การทำงานด้านความปลอดภัย และความยืดหยุ่นในการใช้งาน เป็นต้น

บรรณานุกรม

- [1] ดร.วิรินทร์ เมฆประดิษฐ์สิน. (2557). รู้จักกับ Server load balancing. กรุงเทพฯ : ซีเอ็ดดูเคชั่น.
- [2] ดร.วิรินทร์ เมฆประดิษฐ์สิน. (2557). เพิ่มประสิทธิภาพการเชื่อมต่อกับอินเทอร์เน็ตด้วย Load Balancing. กรุงเทพฯ : ซีเอ็ดดูเคชั่น.
- [3] Load Balancing. [https://en.wikipedia.org/wiki/Load_balancing_\(computing\)](https://en.wikipedia.org/wiki/Load_balancing_(computing)).
- [4] pfSense. <http://thaiopensource.org/ติดตั้ง-pfsense/>.
- [5] ติดตั้ง pfSense features. <http://linuxfirewall.in.th/pfsense/88-general/79-pfsense-features.html>.
- [6] MikroTik. <https://blog.mikrotikthai.com/mikrotik>.
- [7] MikroTik License. <http://mindphp.com/บทความ/212-network/5390-mikrotik>.
- [8] MikroTik License. <https://wiki.mikrotik.com/wiki/>

Manual:License.

[9] Apache JMeter (2018). Apache JMeter. Retrieved March 20, 2018 from <https://jmeter.apache.org/>.

[10] พรรัรักษา กมลเวชช์ และชัยพร เขมระภาตะพันธ์ 2561. การประเมินสมรรถนะการเชื่อมต่อเครือข่ายคอนเทนเนอร์คูเบอร์เนตส์ สาขาวิชาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์.

[11] โชติมา ชาญนุกูล 2557. กรณีศึกษาการทำ Load Balance Web Server สาขาเทคโนโลยีสารสนเทศ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีมหานคร