

ขั้นตอนวิธีการเลือกบริการโครงสร้างพื้นฐานคลาวด์แบบหลายแพลตฟอร์มด้วย แบบจำลองเวกเตอร์สเปซ*

A Multi-Platform Infrastructure-as-a-Service Selection Algorithm with Vector Space Model

สุระชัย หัวไผ่ (Surachai Huapai)**

เทพฤทธิ์ บัณฑิตวัฒนาวงศ์ (Thepparit Banditwattanawong)***

บทคัดย่อ

งานวิจัยนี้เสนอขั้นตอนวิธีการเลือกบริการโครงสร้างพื้นฐานคลาวด์แบบหลายแพลตฟอร์มด้วยการวัดความคล้ายคลึงกันโดยใช้แบบจำลองเวกเตอร์สเปซ ระหว่างแผนแบบเครื่องบริการ(server templates) ที่มีให้บริการ และความต้องการใช้งานทรัพยากรโครงสร้างพื้นฐานคลาวด์ของผู้ใช้ โดยแผนแบบเครื่องบริการที่มีความคล้ายคลึงกับความต้องการใช้งานทรัพยากรโครงสร้างพื้นฐานคลาวด์มากที่สุดจะถูกเลือก

แบบจำลองเวกเตอร์สเปซที่ใช้ในการวัดความคล้ายคลึงกันของงานวิจัยนี้คือแบบโคไซน์ (cosine similarity) ซึ่งมีส่วนประกอบของเวกเตอร์แผนแบบเครื่องบริการเช่นเดียวกับเวกเตอร์ความต้องการใช้งานทรัพยากรโครงสร้างพื้นฐานคลาวด์อันได้แก่ 1) จำนวนเลขที่อยู่ไอพี 2) ค่าใช้จ่ายในการใช้บริการต่อชั่วโมง 3) จำนวนหน่วยประมวลผลกลาง 4) หน่วยความจำหลัก 5) ขนาดพื้นที่หน่วยเก็บข้อมูล 6) ขนาดพื้นที่หน่วยเก็บข้อมูลสูงสุด 7) ช่วงเวลาให้บริการ 8) ปริมาณการถ่ายโอนข้อมูลสูงสุดต่อเดือน และ 9) ระยะเวลาที่ต้องการใช้งานสูงสุด ในการประเมินประสิทธิภาพของขั้นตอนวิธีการเลือกบริการโครงสร้างพื้นฐานคลาวด์แบบหลายแพลตฟอร์มด้วยแบบจำลองเวกเตอร์สเปซที่เสนอถูกดำเนินการกับแผนแบบเครื่องบริการจำนวน 18 แผนแบบและความต้องการใช้งานทรัพยากรโครงสร้างพื้นฐานคลาวด์ของผู้ใช้จำนวน 2 ชุด

ผลการทดลองพบว่าค่าความคล้ายคลึงเชิงมุมโคไซน์ระหว่างแผนแบบเครื่องบริการและความต้องการใช้เครื่องบริการเว็บขนาดกลางมีค่าเท่ากับ 0.941344157 และค่าความคล้ายคลึงเชิงมุมโคไซน์ระหว่างแผนแบบเครื่องบริการและความต้องการใช้เครื่องบริการเว็บขนาดใหญ่มีค่าเท่ากับ 0.999991086 ในภาพรวมพบว่า

* บทความนี้เป็นส่วนหนึ่งของวิทยานิพนธ์เรื่องระบบนายหน้าโดยใช้ออนโทโลยีเพื่อการดำเนินการร่วมกันของการคำนวณแบบคลาวด์หลายแพลตฟอร์ม

This paper is part of thesis entitled "An Ontology-Based Broker System for The Interoperability of Multi-Cloud Computing Platforms"

** นักศึกษาหลักสูตรปรัชญาดุษฎีบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยศรีปทุม 61 ถนนพหลโยธิน แขวงเสนานิคม เขตจตุจักร กรุงเทพมหานคร 10900 อีเมล: surachai.rbac@gmail.com

Ph.D. student in Information Technology, School of Information Technology, Sripatum University 61 Paholyothin Rd., Senanikom Subdistrict, Jatujak District, Bangkok 10900, Thailand E-mail: surachai.rbac@gmail.com

*** อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก ผู้ช่วยศาสตราจารย์ ดร. คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยศรีปทุม

Advisor: Assistant Professor Dr. School of Information Technology, Sripatum University

ขั้นตอนวิธีที่เสนอมีค่าความเที่ยงเท่ากับ 100 เปอร์เซ็นต์ ค่าความระลึกได้เท่ากับ 100 เปอร์เซ็นต์และค่าเอฟ-เมเชอร์เท่ากับ 100 เปอร์เซ็นต์

คำสำคัญ: แบบจำลองเวกเตอร์เวกเตอร์สเปซ, การวัดความคล้ายคลึงกันแบบโคไซน์, บริการโครงสร้างพื้นฐาน คลาวด์, โอเพนสแตก, อาปาเช่คลาวด์สแตก, วิเอ็มแวร์ อีเอสเอ็กซ์ไอ

Abstract

This paper presents an algorithm to select cloud infrastructure-as-a-service in multi-platform by measuring similarity based on the vector space model between the server templates that are available and user requirement for cloud infrastructure resources. A server template that is most similar with the requirement of the user will be chosen.

The vector space model is used to measure the similarity of this research is cosine similarity, which cloud infrastructure-as-a-service templates have the following vector components: 1) number of IP Addresses 2) the cost per hour of the service 3) number of CPUs 4) main memory size 5) storage space 6) ephemeral storage space 7) period of service as a percentage (SLA uptime) 8) maximum amount of monthly data transfer and 9) maximum time requirement. The evaluation of our proposed multi-platform infrastructure-as-a-service selection algorithm with vector space model was conducted with 18 server templates and 2 sets of user requirements for cloud infrastructure resources.

The experimental results were that the value of the cosine similarity between a chosen server template and a user requirement for middle-scale web server was equal to 0.941344157 and the cosine similarity value between another chosen server template and a user requirement for large-scale web server was equal to 0.999991086. In overall, it was found that the algorithm delivered 100 percent precision, 100 percent recall and 100 percent F-Measure.

Keyword: Vector space model, Cosine similarity, IaaS, OpenStack, Apache CloudStack, VMware ESXi

บทนำ

Armbrust, et al., (2009) กล่าวว่า การคำนวณแบบคลาวด์เป็นการเปลี่ยนรูปแบบการให้บริการคอมพิวเตอร์ซึ่งมีความสำคัญอย่างยิ่งในวงการอุตสาหกรรมเทคโนโลยีสารสนเทศในปัจจุบันและยังมียอดการใช้งานขนาดใหญ่เป็นผู้ให้บริการการคำนวณแบบคลาวด์อันเป็นผลมาจากการยอมรับกระบวนทัศน์การคำนวณแบบคลาวด์ เช่น อะเมซอนเว็บเซอร์วิส (Amazon Web Services: AWS) ไมโครซอฟต์อาซัวร์ (Microsoft

azure) และกูเกิลคลาวด์แพลตฟอร์ม (Google cloud platform) ส่วนทิศทางในอนาคตของการคำนวณแบบคลาวด์การ์ทเนอร์ (Gartner, 2016) หักข้อมูลว่าในปี ค.ศ. 2020 องค์การธุรกิจที่มีนโยบายไม่ใช้ระบบการคำนวณแบบคลาวด์ (“no-cloud”) จะหายากหรือไม่มี ทำนองเดียวกับในปัจจุบันที่องค์การซึ่งมีนโยบายไม่ใช้อินเทอร์เน็ต (“no-Internet”) ก็หายากการพยากรณ์นี้ยังได้รับการสนับสนุนโดยรายงานการจัดอันดับการตลาดของการคำนวณแบบคลาวด์ (ITA, 2016) ได้กล่าวถึงอัตราการเจริญเติบโตและการลงทุนทางด้านบริการโครงสร้างพื้นฐานคลาวด์ (IaaS) และบริการแพลตฟอร์มคลาวด์ (PaaS) ภายในปี ค.ศ.2020 มีแนวโน้มการลงทุนมากกว่า 55 ล้านเหรียญดอลลาร์และไอดีซี (IDC, 2015) ได้คาดการณ์ว่าภายในปี ค.ศ.2020 ในองค์การธุรกิจจะมีการใช้การคำนวณแบบคลาวด์เป็นวิธีทางในการดำเนินธุรกิจซึ่งทุกองค์การมีความจำเป็นต้องใช้เหมือนเป็นโครงสร้างพื้นฐานขององค์การ

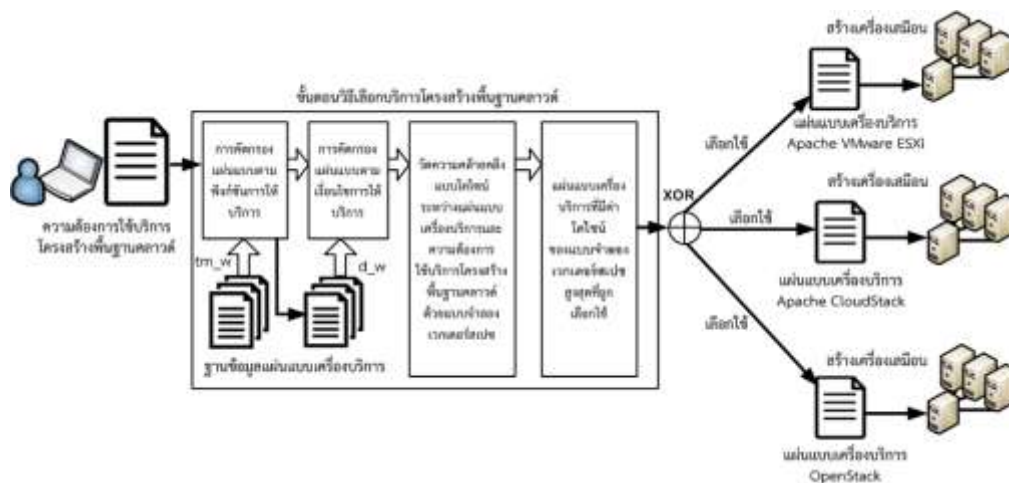
การคำนวณแบบคลาวด์มีหลายแบบจำลองบริการ (service models) และงานวิจัยครั้งนี้มุ่งเน้นแบบจำลองที่เรียกว่า บริการโครงสร้างพื้นฐานคลาวด์ (Infrastructure as a Service: IaaS) สถาบันมาตรฐานและเทคโนโลยีแห่งชาติของสหรัฐอเมริกา (NIST) (Mell and Grance, 2011) นิยามแบบจำลองบริการโครงสร้างพื้นฐาน ว่าเป็นบริการทรัพยากรการคำนวณอันได้แก่ หน่วยประมวลผล, หน่วยเก็บข้อมูล, และเครือข่ายคอมพิวเตอร์ ซึ่งผู้ใช้บริการสามารถใช้ทรัพยากรเหล่านี้เพื่อติดตั้งระบบปฏิบัติการหรือโปรแกรมประยุกต์ใดๆ ได้ตามต้องการในปัจจุบันมีผู้ให้บริการคลาวด์สาธารณะของแบบจำลองบริการโครงสร้างพื้นฐาน (public IaaS providers) หลายรายเช่น อาเมซอนเว็บเซอร์วิสอีซีทู (Amazon Web Services, Inc., 2015) กูเกิลคอมพิวเอนจิน (Google, Inc., 2016) แคทไอริส (CAT Telecom, Inc., 2015) และทรูไอดีซี (True Internet Data Center, Inc., 2015) ซึ่งสามารถตอบสนองความต้องการขององค์การที่ใช้ต้องการจัดตั้งใช้งานบริการคลาวด์ส่วนบุคคลแบบนอกที่ตั้ง (off-premise private cloud) ส่วนองค์การที่มีเหตุผลความจำเป็นในด้านความมั่นคงมักมีความต้องการบริหารจัดการคลาวด์ส่วนบุคคลแบบในที่ตั้ง (on-premise private cloud) ซึ่งต้องจัดหาติดตั้งระบบหรือแพลตฟอร์มโครงสร้างพื้นฐานคลาวด์ด้วยตนเอง นอกจากนี้บางองค์การที่มีความต้องการใช้ระบบโครงสร้างพื้นฐานคลาวด์หลายระบบเพื่อทำการเชื่อมต่อบริการคลาวด์ส่วนบุคคลแบบในที่ตั้งร่วมกับบริการคลาวด์ส่วนบุคคลแบบนอกที่ตั้งสำหรับกระจายภาระงานการประมวลผล กลายเป็นแบบจำลองคลาวด์ลูกผสม (hybrid cloud) ซึ่งในที่นี้หมายถึงโครงสร้างพื้นฐานคลาวด์ที่ประกอบด้วยคลาวด์ส่วนบุคคลแบบในที่ตั้งกับคลาวด์สาธารณะและมีความผูกพันเข้าด้วยกันโดยใช้ข้อกำหนดของเทคโนโลยีมาตรฐานหรือเทคโนโลยีเฉพาะที่ช่วยให้การทำงานของข้อมูลและซอฟต์แวร์ประยุกต์ สามารถทำงานร่วมกัน (Mell and Grance, 2011) ซึ่งช่วยลดปัญหาความเสี่ยงในการใช้บริการโครงสร้างพื้นฐานคลาวด์จากผู้ให้บริการเพียงรายเดียว และช่วยให้ผู้ใช้บริการได้รับบริการที่หลากหลายจากผู้ให้บริการที่ต่างกันจากการใช้บริการของการคำนวณแบบคลาวด์แบบหลายผู้ให้บริการ (Banditwattanawong, et al., 2016)

การใช้งานแบบจำลองคลาวด์ลูกผสมมักประกอบด้วยโครงสร้างพื้นฐานคลาวด์ที่มีแพลตฟอร์มที่แตกต่างกันเพราะแต่ละแพลตฟอร์มมีคุณลักษณะที่เหมาะสมกับผู้ใช้งานแตกต่างกันไป และในการติดตั้งใช้งานแบบจำลองคลาวด์ลูกผสมที่ประกอบด้วยแพลตฟอร์มคลาวด์หลายแพลตฟอร์มนี้มีข้อจำกัดบางประการที่สำคัญคือแพลตฟอร์มที่ใช้มีลักษณะเป็นระบบปิดที่ไม่ได้สร้างมาเพื่อสื่อสารหรือทำงานร่วมกับแพลตฟอร์มอื่น ทำให้เกิดความยากลำบากในการทำงานร่วมกัน (interoperability) (Zhang, 2012) เช่น ปัญหาการทำงานร่วมกัน

ระหว่างบริการโครงสร้างพื้นฐานคลาวด์นอกที่ตั้งที่ใช้แพลตฟอร์มคลาวด์วีเอ็มแวร์ อีเอสเอ็กซ์ไอ (VMware ESXi) (VMware, Inc., 2016) และบริการโครงสร้างพื้นฐานคลาวด์ในที่ตั้งที่ใช้แพลตฟอร์มโอเพนสแตก (OpenStack, 2015) ปัญหาการทำงานร่วมกันระหว่างบริการโครงสร้างพื้นฐานคลาวด์นอกที่ตั้งที่ใช้วีเอ็มแวร์อีเอสเอ็กซ์ไอ และบริการโครงสร้างพื้นฐานคลาวด์ในที่ตั้งที่ใช้อาปาเช่คลาวด์สแตก (Apache CloudStack, 2015) หรือปัญหาการทำงานร่วมกันระหว่างบริการโครงสร้างพื้นฐานคลาวด์นอกที่ตั้งที่ใช้โอเพนสแตกและบริการโครงสร้างพื้นฐานคลาวด์ในที่ตั้งที่ใช้อาปาเช่คลาวด์สแตก วิธีการหนึ่งที่ใช้แก้ปัญหาดังกล่าวคือ การใช้ระบบนายหน้าคลาวด์ (cloud broker) (Huapai and Banditwattanawong, 2015) ซึ่งเป็นระบบสำหรับการจัดการการทำงานร่วมกันระหว่างบริการโครงสร้างพื้นฐานคลาวด์ การจัดการดังกล่าวมีรูปแบบหนึ่งที่น่าสนใจคือ การเลือกบริการโครงสร้างพื้นฐานคลาวด์จากหลายแพลตฟอร์มที่พร้อมใช้งานได้อย่างเหมาะสมกับความต้องการของผู้ใช้เป็นอย่างดีอัตโนมัติ ซึ่งในการทำงานร่วมกันของหลายแพลตฟอร์มคลาวด์มีทั้งส่วนที่คล้ายคลึงกันและส่วนที่แตกต่างกัน ทำให้ยากต่อการตัดสินใจเลือกบริการโดยระบบนายหน้าคลาวด์

ในบทความนี้ผู้วิจัยเสนอขั้นตอนวิธีการเลือกบริการโครงสร้างพื้นฐานคลาวด์แบบหลายแพลตฟอร์มด้วยแบบจำลองเวกเตอร์สเปซ (vector space model) ซึ่งใช้วัดความคล้าย (similarity) ดังปรากฏใน Eidoon, et al., (2007) ซึ่งใช้การวัดความคล้ายคลึงเชิงมุมโคไซน์ของแบบจำลองเวกเตอร์สเปซเพื่อทำการจับคู่ออนไลน์สองชนิดที่มีความคล้ายคลึงกันโดยการกำหนดค่าน้ำหนักที่ไม่ใช่ศูนย์เป็นตัวแทนความสัมพันธ์และการสืบทอดของออนไลน์ Jovita, et al., (2015) ได้เสนอระบบตอบคำถามที่ตอบสนองต่อความต้องการของผู้ใช้โดยใช้วิธีการวัดความคล้ายคลึงเชิงมุมโคไซน์มาหาคำตอบที่ตรงคำถามมากที่สุด ZhiHao, et al., (2012) ได้นำแบบจำลองเวกเตอร์สเปซมาช่วยในการเพิ่มประสิทธิภาพในการค้นหาข้อมูลของบริการเว็บความหมายโดยการจัดลำดับความเหมือนคล้ายของบริการเว็บความหมาย Liu, et al., (2010) ได้นำแบบจำลองเวกเตอร์สเปซมาวัดความเกี่ยวข้องระหว่างคำในโครงสร้างแบบต้นไม้ โดยกำหนดให้แต่ละเวกเตอร์แทนแต่ละโหนดในต้นไม้ Intakosum and Muangon, (2007) ได้เสนอแบบอย่างการพัฒนาโปรแกรมประยุกต์สำหรับการค้นหาข้อมูลด้วยแบบจำลองเวกเตอร์สเปซโดยใช้การวัดความคล้ายคลึงเชิงมุมโคไซน์ สุภา จันทา และ นลินภัทร์ ปรพัฒน์ปรีयर (2013) ได้พัฒนาระบบค้นคืนข้อมูลเว็บกระชูดโดยใช้แบบจำลองเวกเตอร์สเปซเป็นเครื่องมือหลักในการสืบค้นข้อมูลเปรียบเทียบความคล้ายของคำค้นกับเอกสาร อย่างไรก็ตามงานวิจัยที่เกี่ยวข้องเหล่านี้มิได้เป็นการออกแบบขั้นตอนวิธีเฉพาะสำหรับการเลือกบริการโครงสร้างพื้นฐานคลาวด์แบบหลายแพลตฟอร์ม

งานวิจัยนี้จะทำการวัดความคล้ายคลึงระหว่างแผนแบบเครื่องบริการกับความต้องการใช้งานบริการโครงสร้างพื้นฐานคลาวด์ของผู้ใช้งานโดยใช้กรณีศึกษาแพลตฟอร์มโครงสร้างพื้นฐานคลาวด์คือ แพลตฟอร์มคลาวด์โอเพนสแตก แพลตฟอร์มอาปาเช่คลาวด์สแตก และแพลตฟอร์มวีเอ็มแวร์ อีเอสเอ็กซ์ไอ ดังแสดงในภาพที่ 1 โดยพิจารณาจากฟังก์ชัน (function) ของบริการโครงสร้างพื้นฐานคลาวด์ที่อยู่ในรูปแบบของเครื่องเสมือน (virtual machine) เช่น เครื่องให้บริการเว็บ หรือเครื่องให้บริการฐานข้อมูล โดยรายละเอียดของเครื่องให้บริการแต่ละเครื่อง (จำนวนเลขที่อยู่ไอพีค่าใช้จ่ายในการใช้บริการต่อชั่วโมง จำนวนหน่วยประมวลผลกลาง หน่วยความจำหลักขนาดพื้นที่หน่วยเก็บข้อมูล ขนาดพื้นที่หน่วยเก็บข้อมูลสูงสุดช่วงเวลาให้บริการเป็นเปอร์เซ็นต์ ปริมาณการถ่ายโอนข้อมูลสูงสุดต่อเดือน และระยะเวลาที่ต้องการใช้งานสูงสุด) จะถูกจัดเก็บในรูปแบบของแผนแบบเครื่องบริการ (server template)



ภาพที่ 1 แบบจำลองการเลือกใช้บริการโครงสร้างพื้นฐานคลาวด์ของระบบนายหน้าคลาวด์ด้วยวิธีการวัดความคล้ายคลึงแบบโคไซน์

วิธีการวิจัย

การวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาขั้นตอนวิธีการเลือกบริการโครงสร้างพื้นฐานคลาวด์แบบหลายแพลตฟอร์มด้วยแบบจำลองเวกเตอร์สเปซ สำหรับสามแพลตฟอร์มโครงสร้างพื้นฐานคลาวด์ได้แก่ อาปาเช่คลาวด์สแต็กโอเพนสแต็ก และวีเอ็มแวร์ อีเอสเอ็กซ์ไอ โดยมีขั้นตอนการดำเนินการวิจัยดังต่อไปนี้

ขั้นตอนที่ 1 ทำการศึกษาค้นคว้าเอกสาร ตำรา งานวิจัย และคู่มือแพลตฟอร์มบริการโครงสร้างพื้นฐานคลาวด์ทั้งสาม

ขั้นตอนที่ 2 ทำการออกแบบขั้นตอนวิธีการตัดสินใจเลือกใช้แพลตฟอร์มบริการโครงสร้างพื้นฐานคลาวด์ที่ตรงตามความต้องการของผู้ใช้งาน โดยใช้แบบจำลองเวกเตอร์สเปซสำหรับการวัดความคล้ายคลึงเชิงมุมโคไซน์ดังมีรายละเอียดต่อไปนี้

เริ่มจากการกำหนดเวกเตอร์แผ่นแบบเครื่องบริการดังสมการที่ (1)

$$tm_j = (w_{1,j}, w_{2,j}, \dots, w_{n,j}) \text{ when } \begin{cases} n = 1 \text{ to } 9 \\ j \geq 1 \end{cases} \quad (1)$$

ซึ่ง tm_j คือเวกเตอร์แผ่นแบบเครื่องบริการทั้งหมด, j คือจำนวนของแผ่นแบบเครื่องบริการ, w เป็นส่วนประกอบของเวกเตอร์ (components of a vector) และ n คือจำนวนสมาชิกทั้งหมดของส่วนประกอบของเวกเตอร์ซึ่งในที่นี้ n เท่ากับ 9 ดังนี้

- 1) จำนวนเลขที่อยู่ไอพี (IP Address)
- 2) ค่าใช้จ่ายในการใช้บริการต่อชั่วโมง (cost for use)
- 3) จำนวนหน่วยประมวลผลกลาง (cpu core)
- 4) หน่วยความจำหลัก (memory)
- 5) ขนาดพื้นที่หน่วยเก็บข้อมูล (storage)
- 6) ขนาดพื้นที่หน่วยเก็บข้อมูลสูงสุด (ephemeral storage)

7) ช่วงเวลาให้บริการ (SLA uptime) มีหน่วยเปอร์เซ็นต์

8) ปริมาณการถ่ายโอนข้อมูล (data transfer) ต่อเดือน

9) ระยะเวลาที่ต้องการใช้งานสูงสุด (usage time)

จากนั้นกำหนดเวกเตอร์ความต้องการใช้งานทรัพยากรโครงสร้างพื้นฐานคลาวด์ดังสมการที่ (2)

$$q = (w_1, w_2, \dots, w_n) \text{ when } n = 1 \text{ to } 9 \quad (2)$$

ซึ่ง q คือเวกเตอร์ความต้องการใช้งานทรัพยากรโครงสร้างพื้นฐานคลาวด์ซึ่งมีส่วนประกอบของ

เวกเตอร์เหมือนกับของเวกเตอร์ tm

จากนั้นทำการคัดเลือกเวกเตอร์แผ่นแบบที่เหมาะสมจากเวกเตอร์แผ่นแบบเครื่องบริการทั้งหมดในฐานข้อมูลตามขั้นตอนการคัดกรองแผ่นแบบตามฟังก์ชันการให้บริการดังในภาพที่ 1 โดยมีเงื่อนไขการคัดกรองแบ่งออกเป็น 2 ขั้นตอนย่อยดังนี้

1) ทำการคัดกรองแผ่นแบบเครื่องบริการที่มีฟังก์ชันตรงกับความต้องการใช้งานซึ่งถูกระบุไว้ในชื่อของเวกเตอร์แผ่นแบบเครื่องบริการเช่น เครื่องบริการเว็บ หรือเครื่องบริการฐานข้อมูล

2) ทำการคัดกรองแผ่นแบบเครื่องบริการจากขั้นตอนย่อยแรกโดยเปรียบเทียบกับเวกเตอร์ความต้องการใช้งานทรัพยากรโครงสร้างพื้นฐานคลาวด์ผลลัพธ์ที่ได้คือ แผ่นแบบเครื่องบริการที่มีส่วนประกอบของเวกเตอร์เป็นไปตามเงื่อนไขในตารางที่ 1 ครบทุกเงื่อนไขเท่านั้น โดยเงื่อนไขในตารางที่ 1 สามารถแบ่งออกเป็น 3 กลุ่มดังนี้ กลุ่มแรกเป็นเงื่อนไขที่กำหนดให้ส่วนประกอบของความ ต้องการใช้งานบริการโครงสร้างพื้นฐานคลาวด์ต้องเท่ากับส่วนประกอบของแผ่นแบบเครื่องบริการเท่านั้น ซึ่งส่วนประกอบดังกล่าวคือ จำนวนเลขที่อยู่ไอพีที่ต้องการใช้งาน กลุ่มที่สองเป็นเงื่อนไขที่กำหนดให้ส่วนประกอบของความ ต้องการใช้งานบริการโครงสร้างพื้นฐานคลาวด์ต้องน้อยกว่าหรือเท่ากับส่วนประกอบของแผ่นแบบเครื่องบริการเท่านั้น ซึ่งส่วนประกอบดังกล่าวคือค่าใช้จ่ายในการใช้บริการต่อชั่วโมง กลุ่มที่สามเป็นเงื่อนไขมากกว่าหรือเท่ากับซึ่งใช้พิจารณาเท่านั้นซึ่งส่วนประกอบดังกล่าวคือ จำนวนหน่วยประมวลผลกลาง จำนวนหน่วยความจำหลัก ขนาดพื้นที่หน่วยเก็บข้อมูล ขนาดพื้นที่หน่วยเก็บข้อมูลสูงสุด ช่วงเวลาให้บริการ ปริมาณการถ่ายโอนข้อมูลสูงสุดต่อเดือน และระยะเวลาที่ต้องการใช้งานสูงสุด

ตารางที่ 1 เงื่อนไขการการคัดกรองแบบเครื่องบริการ

ลำดับ	ส่วนประกอบของเวกเตอร์แผ่นแบบ	เงื่อนไข	ส่วนประกอบของเวกเตอร์ความต้องการใช้งาน
1	จำนวนเลขที่อยู่ไอพี ($tm_w[1,j]$)	=	จำนวนเลขที่อยู่ไอพี ($q_w[1]$)
2	ค่าใช้จ่ายในการใช้บริการต่อชั่วโมง ($tm_w[2,j]$)	<=	ค่าใช้จ่ายในการใช้บริการต่อชั่วโมง ($q_w[2]$)
3	จำนวนหน่วยประมวลผลกลาง ($tm_w[3,j]$)	>=	จำนวนหน่วยประมวลผลกลาง ($q_w[3]$)
4	หน่วยความจำหลัก ($tm_w[4,j]$)	>=	หน่วยความจำหลัก ($q_w[4]$)
5	ขนาดพื้นที่หน่วยเก็บข้อมูล ($tm_w[5,j]$)	>=	ขนาดพื้นที่หน่วยเก็บข้อมูล ($q_w[5]$)
6	ขนาดพื้นที่หน่วยเก็บข้อมูลสูงสุด ($tm_w[6,j]$)	>=	ขนาดพื้นที่หน่วยเก็บข้อมูลสูงสุด ($q_w[6]$)
7	ช่วงเวลาให้บริการ ($tm_w[7,j]$)	>=	ช่วงเวลาให้บริการ ($q_w[7]$)

8	ปริมาณการถ่ายโอนข้อมูลสูงสุดต่อเดือน (tm_w[8,j])	>=	ปริมาณการถ่ายโอนข้อมูลสูงสุดต่อเดือน (q_w[8])
9	ระยะเวลาที่ต้องการใช้งานสูงสุด (tm_w[9,j])	>=	ระยะเวลาที่ต้องการใช้งานสูงสุด (q_w[9])

ผลลัพธ์การคัดกรองเวกเตอร์แผ่นแบบเครื่องบริการที่ได้จากขั้นตอนย่อยที่ 2) คือเวกเตอร์ d_j ดังสมการที่ (3) โดยที่ j คือจำนวนแผ่นแบบเครื่องบริการที่ผ่านการคัดกรองทั้งหมดข้างต้นแล้ว

$$d_j = (w_{1,j}, w_{2,j}, \dots, w_{n,j}) \text{ when } \begin{cases} n = 1 \text{ to } 9 \\ j \geq 1 \end{cases} \quad (3)$$

จากนั้นนำเวกเตอร์แผ่นแบบเครื่องบริการและเวกเตอร์ความต้องการใช้งานทรัพยากรโครงสร้างพื้นฐานคลาวด์ไปทำการวัดความคล้ายคลึงเชิงมุมโคไซน์โดยใช้สมการที่ (4) (Gareth, 2011)

$$\text{sim}(d_j, q) = \frac{d_j \cdot q}{\|d_j\| \|q\|} \quad (4)$$

ซึ่งเขียนในอีกรูปแบบหนึ่งได้คือ

$$\text{sim}(d_j, q) = \frac{\sum_{i=1}^n w_{i,j} \cdot w_i}{\sqrt{\sum_{i=1}^n w_{i,j}^2} \sqrt{\sum_{i=1}^n w_i^2}} \text{ when } \begin{cases} n = 1 \text{ to } 9 \\ j \geq 1 \end{cases} \quad (5)$$

ตารางที่ 2 ขั้นตอนวิธีการหาค่าการวัดความคล้ายคลึงเชิงมุมโคไซน์ของระบบนายหน้า

algorithm: IaaS selection

1. input variables:
2. $q_w \leftarrow \{q_w_1, q_w_2, q_w_3, q_w_4, q_w_5, q_w_6, q_w_7, q_w_8, q_w_9\}$ /* a vector (1-D array) of requirement from
3. user according to the equation (2) */
4. $tm_w \leftarrow \{tm_w_1, tm_w_2, tm_w_3, tm_w_4, tm_w_5, tm_w_6, tm_w_7, tm_w_8, tm_w_9, \dots, tm_w_m\}$ /* 9 x m matrix
5. (2-D array) of IaaS template vectors according to the equation (1); m is the total number of templates */
6. local variables:
7. $d_w \leftarrow \{d_w_1, d_w_2, d_w_3, d_w_4, d_w_5, d_w_6, d_w_7, d_w_8, d_w_9, \dots, d_w_p\}$
8. /* 9 x p matrix (2-D array) of template vectors according to eq. (3); p is the number of functional templates
9. where $p < m$ */
10. norm_d /* norm of a vector template or $\|d_w_j\|$ */
11. norm_q /* norm of a vector requirement from user or $\|q_w\|$ */


```
12.   coef    /* similarity coefficientor d_w . q_w */
13.   sim[j]  /* cosine similarity */
14.   j ← 0    /* counter of templates vectors */
15.   begin
16.   d_w ← find_templates_with_matched_functional_requirement(tm_w) /* functional
      filtering of templates */
17.   /* non- functional filtering of templates based on boundary constraints */
18.   For each tm_w[i,m] in tm
19.   if tm_w[1,m] = q_w[1] /* IP address quantity of template and user requirement
      must be equal */
20.   if tm_w[2,m] <= q_w[2] /* template's cost must be less than or equal to that of user
      requirement */
21.   if tm_w[3,m] <= q_w[3] /* template's cores must be greater than or equal to that of
      user requirement */
22.   if tm_w[4,m] <= q_w[4] /* template's memory must be greater than or equal to that
      of user requirement */
23.   if tm_w[5,m] <= q_w[5] /* template's storage space must be greater than or equal
      to that of user
24.   requirement */
25.   if tm_w[6,m] <= q_w[6] /*template's ephemeral storage space must be greater than
      or equal to that of user
26.   requirement */
27.   if tm_w[7,m] <= q_w[7]/* template's SLA (uptime) must be greater than or equal to
      that of user requirement*/
28.   if tm_w[8,m] <= q_w[8]/* template's monthly data transfer amount must be greater
      than or equal to that of
29.   user requirement */
30.   if tm_w[9,m] <= q_w[9]/* template's usage hours must be greater than that of or
      equal to user requirement */
31.   then d_w[1...9,++j] ← tm_w[1...9,m]
32.   /* similarity calculation */
33.   For j ←1 to p
34.   coef ←  $\sum_{i=1}^9 d_w[i, j] * q_w[i]$ 
```



```

35. norm_d  $\leftarrow \sqrt{\sum_{i=1}^9 d\_w[i]^2}$ 
36. norm_q  $\leftarrow \sqrt{\sum_{i=1}^9 q\_w[i]^2}$ 
37. sim[j]  $\leftarrow \frac{\text{coef}}{\text{norm\_d} * \text{norm\_q}}$ 
38. return selected_template  $\leftarrow$  find_all_vectors_in_d_w_with_max(sim) /* return
    template(s) of a maximum
39. similarity value */

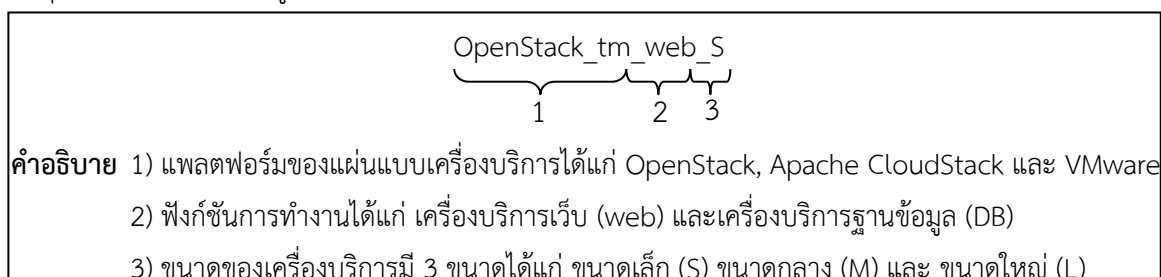
```

ค่าความคล้ายคลึงเชิงมุมโคไซน์ (sim) จะมีค่าเท่ากับ 1 ถ้าเวกเตอร์ทั้งสองอยู่ในแนวเดียวกัน (คือมีมุมระหว่างเวกเตอร์เท่ากับ 0 องศา) ซึ่งหมายความว่ามีความคล้ายคลึงกันสูงสุด หรือเหมือนกัน ซึ่งค่าความคล้ายคลึงเชิงมุมโคไซน์จะมีค่าต่ำสุดคือ 0

ขั้นตอนวิธีที่ออกแบบทั้งหมดในขั้นตอนที่ 2 นี้แสดงไว้ในตารางที่ 2 ซึ่งมีวัตถุประสงค์ในการคัดเลือกแผนแบบเครื่องบริการที่มีความเหมาะสมกับความใกล้เคียงกับความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์มากที่สุด โดยประกอบด้วยตัวแปร 2 ประเภทคือ 1) ตัวแปรนำเข้าประกอบด้วย q_w ซึ่งแทนเวกเตอร์ของความต้องการการใช้งานโครงสร้างพื้นฐานคลาวด์ตามสมการที่ (2) และ tm_w ซึ่งแทนเวกเตอร์ของแผนแบบเครื่องบริการตามสมการที่ (1), 2) ตัวแปรเฉพาะที่ซึ่งเป็นตัวแปรที่ใช้ในประมวลผลภายในชุดคำสั่งของขั้นตอนวิธี ประกอบด้วย ตัวแปร d_w คือตัวแปรที่ใช้เก็บแผนแบบเครื่องบริการที่ถูกคัดกรองตามเงื่อนไขในตารางที่ 1 โดย $d_w \subseteq tm_w$ ขั้นตอนวิธีบรรทัดที่ 16 เป็นการคัดกรองแผนแบบเครื่องบริการที่มีหน้าที่ (เช่น เครื่องบริการเว็บ, เครื่องบริการฐานข้อมูล) ให้ตรงกับความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์ แล้วนำไปคัดกรองต่อตามขั้นตอนวิธีบรรทัดที่ 18 – 31 เป็นการคัดกรองแผนแบบเครื่องบริการที่ถูกเก็บในตัวแปร tm_w ตามเงื่อนไขในตารางที่ 1 ถ้าแผนแบบใดทำให้เงื่อนไขเป็นจริงก็จะถูกนำไปเก็บในตัวแปร d จากนั้นนำตัวแปร d_w ไปทำการคำนวณหาความคล้ายคลึงเชิงมุมโคไซน์ในขั้นตอนวิธีในบรรทัดที่ 33 – 37 โดยเปรียบเทียบเวกเตอร์ q ที่เก็บในตัวแปร q_w กับเวกเตอร์ d ที่เก็บในตัวแปร d_w ทุกแผนแบบเครื่องบริการ จากนั้นหาค่าสูงสุด โดยใช้ฟังก์ชัน find_all_vectors_in_d_w_with_max() ในบรรทัดที่ 38 ซึ่งจะส่งคืนค่าของแผนแบบเครื่องบริการที่มีความเหมาะสมที่สุดออกมา กรณีที่ค่าของความคล้ายคลึงเชิงมุมโคไซน์ของหลายแผนแบบมีค่าเท่ากันฟังก์ชัน find_all_vectors_in_d_w_with_max() จะมีสองทางเลือกขึ้นอยู่กับนโยบายที่ผู้ใช้กำหนด คือ ส่งคืนแผนแบบที่มีความสำคัญ (priority) มากที่สุดที่ผู้ใช้งานกำหนดไว้ก่อนหน้า หรือส่งคืนแผนแบบที่เหมาะสมในเชิงการรักษาสมดุลภาระงาน (load balancing) ของทรัพยากรบริการโครงสร้างพื้นฐานคลาวด์ที่พร้อมใช้งาน

ขั้นตอนที่ 3 สร้างชุดข้อมูลสำหรับประเมินผลการทำงานของขั้นตอนวิธีได้แก่ ชุดข้อมูลแผนแบบเครื่องบริการและชุดข้อมูลความต้องการใช้งานบริการโครงสร้างพื้นฐานคลาวด์โดยชุดข้อมูลแผนแบบครอบคลุมฟังก์ชันการทำงานของเครื่องบริการ 2 ฟังก์ชันได้แก่ เครื่องบริการเว็บ และเครื่องบริการฐานข้อมูล ซึ่งถูกเลือกใช้

งานสำหรับระบบบริหารงานด้วยสารสนเทศโดยส่วนใหญ่ (สุจิตรา และคณะ, 2558: 92; เสกสรรค์ และคณะ, 2558: 5) โดยมีรายละเอียดดังแสดงในตารางที่ 3 ซึ่งสมมติทั้งหมด (ยกเว้นสมมติแรก) ของตารางตั้งชื่อตามตัวแปรทั้ง 9 ตัวตามสมการที่ (1) แต่ละแถวของตารางแทนหนึ่งเวกเตอร์แผนแบบเครื่องบริการในสมการที่ (1) ซึ่งชุดข้อมูลแผนแบบเครื่องบริการที่ใช้ในการทดลองได้ตั้งชื่อตามฟังก์ชันการทำงานและขนาดของเครื่องบริการดังระบุไว้ในสมมติที่ 1 โดยมีรูปแบบการตั้งชื่อดังอธิบายในภาพที่ 2



ภาพที่ 2 แสดงความหมายของชื่อแผนแบบเครื่องบริการ

ตารางที่ 3 ข้อมูลแผนแบบเครื่องบริการทั้งหมดที่ใช้ในการทดลอง

แผนแบบเครื่องบริการ	tm_w1 IP Addresses (numbers)	tm_w2 cost (USD/Hr)	tm_w3 cpu core (cores)	tm_w4 memo ry (GB)	tm_w5 stora ge (GB)	tm_w6 ephemeral storage (GB)	tm_w7 SLA(upti me) (%)	tm_w8 data transfer (GB)	tm_w9 usage time (days)
OpenStack_tm_web_S	1	1	1	2	1	100	95.00	500	30
OpenStack_tm_web_M	2	2.5	2	4	10	500	95.00	1000	365
OpenStack_tm_web_L	2	5	4	8	100	500	95.00	1000	365
OpenStack_tm_DB_S	1	1	1	2	1	100	95.00	500	30
OpenStack_tm_DB_M	2	2.5	2	4	10	500	95.00	1000	365
OpenStack_tm_DB_L	2	5	4	8	100	500	95.00	1000	365
CloudStack_tm_	1	1	1	2	1	100	98.00	500	30

web_S									
CloudStack_tm_									
web_M	2	2	2	4	50	500	98.00	1000	365
CloudStack_tm_									
web_L	2	6	4	8	100	500	98.00	1000	365
CloudStack_tm_									
DB_S	1	1	1	2	1	100	98.00	500	30
CloudStack_tm_									
DB_M	2	2	2	4	50	500	98.00	1000	365
CloudStack_tm_									
DB_L	3	6	3	8	100	500	98.00	1000	365
VMware_tm_we									
b_S	1	2	1	2	1	100	98.00	500	30
VMware_tm_we									
b_M	2	4	2	4	10	500	98.00	1000	365
VMware_tm_we									
b_L	2	8	4	16	100	500	98.00	1000	365
VMware_tm_DB_									
S	1	2	1	2	1	100	98.00	500	30
VMware_tm_DB_									
M	2	4	2	4	10	500	98.00	1000	365
VMware_tm_DB_									
L	2	8	4	16	100	500	98.00	1000	365

ขั้นตอนที่ 4 ทำการทดสอบขั้นตอนวิธีที่เสนอไว้ในขั้นตอนที่ 2 ด้วยชุดข้อมูลจากขั้นตอนที่ 3 โดยใช้โปรแกรมแผ่นตารางทำการ (spread sheet) ในการคำนวณตามขั้นตอนย่อยต่อไปนี้

1) กำหนดข้อมูลความต้องการใช้งานบริการโครงสร้างพื้นฐานคลาวด์ของผู้ใช้ ซึ่งประกอบด้วยตัวแปร q_{w1} - q_{w9} ตามสมการที่ (2) โดยมีค่าตัวแปรดังระบุในตารางที่ 4 ซึ่งประกอบด้วย 2 เวกเตอร์ (2 แถว) ได้แก่ ความต้องการใช้เครื่องบริการเว็บขนาดกลาง และความต้องการใช้เครื่องบริการเว็บขนาดใหญ่

ตารางที่ 4 ข้อมูลความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์ของผู้ใช้

No.	q_w1 IP Address (numbers)	q_w2 cost (USD/Hr)	q_w3 cpu core (cores)	q_w4 memory (GB)	q_w5 storage (GB)	q_w6 ephemeral storage (GB)	q_w7 SLA(uptime) (%)	q_w8 data transfer (GB)	q_w9 usage time (days)
1	1	2	1	4	1	100	95	1000	365
2	2	8	2	4	10	100	95	1000	365
3	2	8	4	4	100	500	95	1000	365

2) นำเข้าข้อมูลแผ่นแบบเครื่องบริการจากตารางที่ 3 เพื่อคัดกรองตามฟังก์ชันที่ตรงกับความต้องการใช้งานโครงสร้างพื้นฐานคลาวด์ตามขั้นตอนย่อยที่ 1) ในขั้นตอนที่ 2 จากนั้นคัดกรองตามเงื่อนไขดังอธิบายในขั้นตอนย่อยที่ 2) ของขั้นตอนที่ 2 (นั่นคือตามขั้นตอนวิธีในตารางที่ 2 บรรทัดที่ 18 – 31)

3) คำนวณค่าของความคล้ายคลึงเชิงมุมโคไซน์ระหว่างแผ่นแบบเครื่องบริการที่ผ่านการคัดกรองในขั้นตอนย่อยที่ 2) ข้างต้นและแต่ละความต้องการใช้บริการโครงสร้างพื้นฐานของผู้ใช้ตามขั้นตอนวิธีในตารางที่ 2 บรรทัดที่ 33 – 37

ขั้นตอนที่ 5 วิเคราะห์ข้อมูลผลการทดสอบ โดยใช้สมการความเที่ยง (precision) การระลึกได้ (recall) และ เอฟ-เมเชอร์ (F-measure) ซึ่งมีรายละเอียดดังนี้

1) ทำการทดสอบความเที่ยงโดยการรวบรวมข้อมูล ดังสมการที่ (6)

$$\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \times 100 \quad (6)$$

2) ทำการทดสอบการระลึกได้ โดยการรวบรวมข้อมูล ดังสมการที่ (7)

$$\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \times 100 \quad (7)$$

3) ทำการทดสอบเอฟ-เมเชอร์ โดยนำค่าจากสมการที่ (6) และ (7) มาคำนวณในสมการที่ (8)

$$F - \text{measure} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (8)$$

ขั้นตอนที่ 6 เป็นขั้นตอนสรุปผลการวิจัยและอภิปรายผลการวิจัยดังแสดงข้อมูลในหัวข้อถัดไป

ผลการวิจัย

ผลจากการทดลองตามขั้นตอนวิจัยที่กำหนดไว้สามารถแบ่งเป็น 2 ตอนดังนี้

ตอนที่ 1 ผลการคำนวณค่าของความคล้ายคลึงเชิงมุมโคไซน์ระหว่างแผ่นแบบเครื่องบริการและความต้องการใช้บริการโครงสร้างพื้นฐานของผู้ใช้

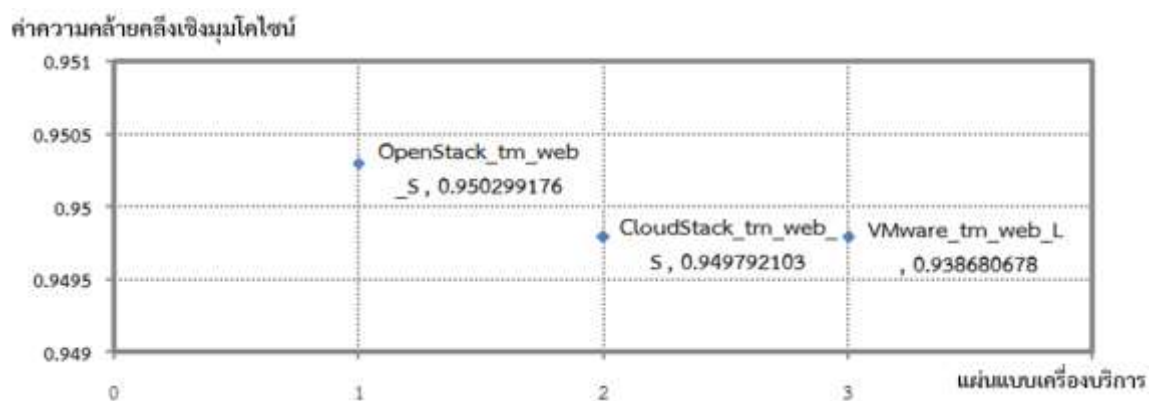
จากการหาค่าความคล้ายคลึงเชิงมุมโคไซน์ระหว่างเวกเตอร์ของความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์ของผู้ใช้งานที่มีความต้องการใช้เครื่องบริการเว็บขนาดเล็ก ขนาดกลาง และขนาดใหญ่ ซึ่งใช้ชุด

ข้อมูลตามตารางที่ 4 สามารถเขียนในรูปแบบเวกเตอร์ได้ตามลำดับดังนี้ ลำดับที่ 1 ความต้องการใช้เครื่องบริการเว็บขนาดเล็กเวกเตอร์คือ $q_w(1,2,1,4,1,100,95,1000,365)$ ลำดับที่ 2 ความต้องการใช้เครื่องบริการเว็บขนาดกลางเวกเตอร์คือ $q_w(2, 8, 2, 4, 10, 100, 95, 1000, 365)$ และลำดับที่ 3 ความต้องการใช้เครื่องบริการเว็บขนาดใหญ่เวกเตอร์คือ $q_w(2, 8, 2, 4, 10, 100, 95, 1000, 365)$ จากนั้นนำเวกเตอร์ทั้งสามไปคำนวณหาค่าความคล้ายคลึงเชิงมุมโคไซน์โดยนำไปเปรียบเทียบกับแผ่นแบบเครื่องบริการในตารางที่ 3 ตามขั้นตอนวิธีที่เสนอไว้ข้างต้น ซึ่งจะได้ผลลัพธ์ซึ่งจะได้ผลลัพธ์ดังแสดงในตารางที่ 6, 7 และ 8 ตามลำดับ ซึ่งทั้งสามตารางมีชื่อสมมติเป็นไปตามตัวแปรในสมการที่ (4)

ตารางที่ 6 ผลการคำนวณค่าของความคล้ายคลึงเชิงมุมโคไซน์ระหว่างแผ่นแบบเครื่องบริการและความต้องการใช้บริการโครงสร้างพื้นฐานที่ผู้ใช้งานเครื่องบริการเว็บขนาดเล็ก

แผ่นแบบเครื่องบริการ	$d_{w_j} \cdot q_w$	$\ d_{w_j}\ $	$\ q_w\ $	$\text{sim}(d_{w_j}, q_w)$
OpenStack_tm_web_S	529988	519.5507675	1073.44	0.950299176
CloudStack_tm_web_S	530273	520.1076812	1073.44	0.949792103
VMware_tm_web_S	530275	520.1105652	1073.44	0.949790418

จากตารางที่ 6 สามารถเปรียบเทียบค่าความคล้ายคลึงเชิงมุมโคไซน์ได้ดังภาพที่ 3 ซึ่งจะพบว่าค่าของความคล้ายคลึงเชิงมุมโคไซน์ระหว่างแผ่นแบบเครื่องบริการและความต้องการใช้บริการโครงสร้างพื้นฐานที่ผู้ใช้งานต้องการมีค่าสูงสุดเท่ากับ 0.950299176 สำหรับแผ่นแบบเครื่องบริการ OpenStack_tm_web_S

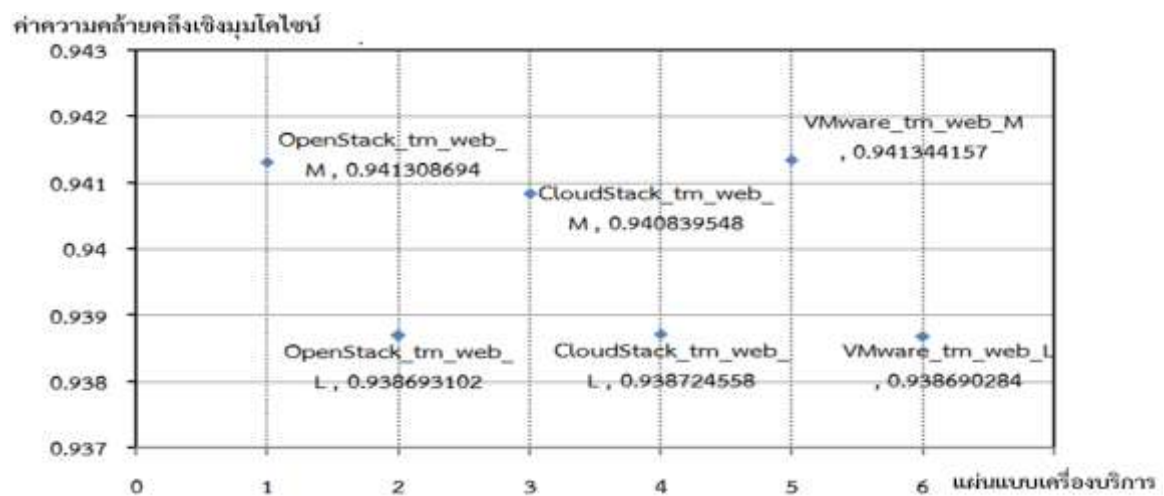


ภาพที่ 3 กราฟเปรียบเทียบความคล้ายคลึงเชิงมุมโคไซน์ระหว่างแผ่นแบบเครื่องบริการและความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์ที่ผู้ใช้งานเครื่องบริการเว็บขนาดเล็ก

ตารางที่ 7 ผลการคำนวณค่าของความคล้ายคลึงเชิงมุมโคไซน์ระหว่างแผ่นแบบเครื่องบริการและความต้องการใช้บริการโครงสร้างพื้นฐานที่ผู้ใช้งานเครื่องบริการเว็บขนาดกลาง

แผ่นแบบเครื่องบริการ	$d_{w_j} \cdot q_w$	$\ d_{w_j}\ $	$\ q_w\ $	$\text{sim}(d_{w_j}, q_w)$
OpenStack_tm_web_M	1192394	1179.991631	1073.517	0.941308694
OpenStack_tm_web_L	1193334	1184.212396	1073.517	0.938693102
CloudStack_tm_web_M	1193073	1181.252302	1073.517	0.940839548
CloudStack_tm_web_L	1193625	1184.461481	1073.517	0.938724558
VMware_tm_web_M	1192691	1180.241077	1073.517	0.941344157
VMware_tm_web_L	1193675	1184.554347	1073.517	0.938690284

จากตารางที่ 7 สามารถเปรียบเทียบค่าความคล้ายคลึงเชิงมุมโคไซน์ได้ดังภาพที่ 4 ซึ่งจะพบว่าค่าของความคล้ายคลึงเชิงมุมโคไซน์ระหว่างแผ่นแบบเครื่องบริการและความต้องการใช้บริการโครงสร้างพื้นฐานที่ผู้ใช้งานเครื่องบริการเว็บขนาดกลางมีค่าสูงสุดเท่ากับ 0.941344157 สำหรับแผ่นแบบเครื่องบริการ VMware_tm_web_M

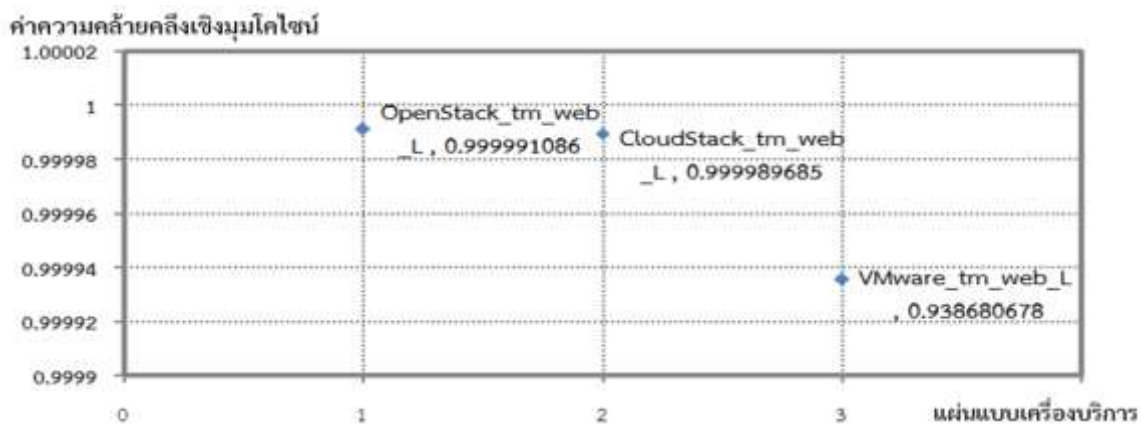


ภาพที่ 4 กราฟเปรียบเทียบความคล้ายคลึงเชิงมุมโคไซน์ระหว่างแผ่นแบบเครื่องบริการและความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์ที่ผู้ใช้งานเครื่องบริการเว็บขนาดกลาง

ตารางที่ 8 ผลการคำนวณค่าของความคล้ายคลึงเชิงมุมโคไซน์ระหว่างแผ่นแบบเครื่องบริการและความต้องการใช้บริการโครงสร้างพื้นฐานที่ผู้ใช้งานเครื่องบริการเว็บขนาดใหญ่

แผ่นแบบเครื่องบริการ	$d_{w_j} \cdot q_w$	$\ d_{w_j}\ $	$\ q_w\ $	$\text{sim}(d_{w_j}, q_w)$
OpenStack_tm_web_L	1402342	1184.212396	1184.209	0.999991086
CloudStack_tm_web_L	1402635	1184.461481	1184.209	0.999989685
VMware_tm_web_L	1402669	1184.554347	1184.209	0.999935527

จากตารางที่ 8 สามารถเปรียบเทียบค่าของความคล้ายคลึงเชิงมุมโคไซน์ได้ดังภาพที่ 5 ซึ่งจะพบว่าค่าของความคล้ายคลึงเชิงมุมโคไซน์ระหว่างแผนแบบเครื่องบริการและความต้องการใช้บริการโครงสร้างพื้นฐานที่ผู้ใช้งานมีค่าสูงสุดเท่ากับ 0.999991086 สำหรับแผนแบบเครื่องบริการ OpenStack_tm_web_L



ภาพที่ 5 กราฟเปรียบเทียบความคล้ายคลึงเชิงมุมโคไซน์ระหว่างแผนแบบเครื่องบริการและความต้องการใช้บริการโครงสร้างพื้นฐานที่ผู้ใช้งานใช้เครื่องบริการเว็บขนาดใหญ่

ตอนที่ 2 ผลการประเมินประสิทธิภาพของขั้นตอนวิธีที่เสนอ

จากขั้นตอนที่ 1 ซึ่งเป็นการนำขั้นตอนวิธีมาใช้กับข้อมูลแผนแบบเครื่องบริการ 18 แผนแบบตามตารางที่ 3 และข้อมูลความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์ตาม 3 ชุดตารางที่ 4 พบว่าผลการทำงานของขั้นตอนวิธีมีค่าจริงเชิงบวก (true positive) เท่ากับ 2 ค่าเท็จเชิงบวก (false positive) เท่ากับ 0 ค่าเท็จเชิงลบ (false negative) เท่ากับ 0 และค่าจริงเชิงลบ (true negative) เท่ากับ 17 ดังแสดงในตารางที่ 9

ตารางที่ 9 ผลการเลือกใช้แผนแบบเครื่องบริการโดยเปรียบเทียบระหว่างสิ่งที่คาดการณ์ไว้ (predicted) และสิ่งที่จริง (actual)

ที่คาดการณ์ไว้/ ที่จริง	ค่าจริง	ค่าเท็จ
ค่าจริง	3	0
ค่าเท็จ	0	17

เมื่อนำข้อมูลในตารางที่ 9 ไปคำนวณหาความเที่ยงด้วยสมการที่ (6) พบว่าค่าความเที่ยงเท่ากับ 100 เปอร์เซ็นต์ ค่าการระลึกได้โดยใช้สมการที่ (7) พบว่าค่าการระลึกได้เท่ากับ 100 เปอร์เซ็นต์และค่าเอฟ-เมเชอร์โดยใช้สมการที่ (8) พบว่าค่าเอฟ-เมเชอร์เท่ากับ 100 เปอร์เซ็นต์

การสรุปผลและอภิปราย

บทความนี้เสนอขั้นตอนวิธีการเลือกบริการโครงสร้างพื้นฐานคลาวด์แบบหลายแพลตฟอร์มด้วยแบบจำลองเวกเตอร์สเปซและประเมินผลด้วยชุดข้อมูลแผนแบบเครื่องบริการที่มีความครอบคลุมฟังก์ชันการทำงานของเครื่องบริการ 2 ฟังก์ชันได้แก่ เครื่องบริการเว็บ และเครื่องบริการฐานข้อมูล ที่รองรับสามแพลตฟอร์มบริการโครงสร้างพื้นฐานคลาวด์ได้แก่ โอเพนสแตค อาปาเช่คลาวด์สแตกและวีเอ็มแวร์ อีเอสเอ็กซ์ไอรวมทั้งสิ้น 18 แผนแบบเครื่องบริการ และชุดข้อมูลความต้องการใช้บริการโครงสร้างพื้นฐานคลาวด์ 3 ชุดประกอบด้วยความต้องการใช้เครื่องบริการเว็บขนาดเล็ก ขนาดกลาง และขนาดใหญ่ มีข้อค้นพบและการอภิปรายผลดังนี้

การเลือกแผนแบบเครื่องบริการตามค่าความคล้ายคลึงเชิงมุมโคไซน์พบว่าข้อมูลความต้องการชุดที่ 1 ได้ค่าความคล้ายคลึงเชิงมุมโคไซน์เท่ากับ 0.950299176 ที่เป็นค่าสูงสุดและตรงกับแผนแบบ OpenStack_tm_web_S ข้อมูลความต้องการชุดที่ 2 ได้ค่าความคล้ายคลึงเชิงมุมโคไซน์เท่ากับ 0.941344157 ที่เป็นค่าสูงสุดและตรงกับแผนแบบ VMware_tm_web_M และข้อมูลความต้องการชุดที่ 3 ได้ค่าความคล้ายคลึงเชิงมุมโคไซน์เท่ากับ 0.999991086 ที่เป็นค่าสูงสุดและตรงกับแผนแบบ OpenStack_tm_web_L ซึ่งตรงกับแผนแบบที่คาดการณ์ไว้ก่อนหน้านี้ทั้ง 3 ครั้ง เมื่อพิจารณาจากกราฟเปรียบเทียบในภาพที่ 4 จะพบว่าแผนแบบเครื่องบริการทั้ง 6 แบ่งออกเป็น 2 กลุ่ม กลุ่มล่างที่มีค่าน้อยกว่า 0.94 เป็นกลุ่มของแผนแบบเครื่องบริการที่รองรับการบริการขนาดใหญ่ซึ่งมีส่วนประกอบของเวกเตอร์ทรัพยากรโครงสร้างพื้นฐานคลาวด์ที่ไม่ตรงกับความต้องการใช้บริการของผู้ใช้งานค่อนข้างมาก ส่วนกลุ่มที่ 2 เป็นกลุ่มที่มีค่ามากกว่า 0.94 ซึ่งมีส่วนประกอบของเวกเตอร์ที่มีความคล้ายคลึงกับความต้องการใช้บริการของผู้ใช้งานที่ต้องการเลือกใช้บริการแผนแบบเครื่องบริการขนาดกลางส่วนกราฟเปรียบเทียบในภาพที่ 4 จะมีเพียงกลุ่มเดียวเพราะแผนแบบเครื่องบริการที่ไม่เกี่ยวข้องได้ถูกคัดกรองในขั้นตอนแรกของขั้นตอนวิธีแล้ว ในการวัดประเมินประสิทธิภาพด้วยขั้นตอนวิธีด้วยวิธี ความเที่ยง การระลึก และ เอฟ-เมเชอร์พบว่าขั้นตอนวิธีที่เสนอให้ผลลัพธ์ค่าจริงเชิงบวกเท่ากับ 2 ค่าเท็จเชิงบวกเท่ากับ 0 ค่าเท็จเชิงลบเท่ากับ 0 และค่าจริงเชิงลบเท่ากับ 17 ซึ่งสามารถคำนวณค่าความเที่ยงเท่ากับ 100 เปอร์เซ็นต์เพราะค่าเท็จเชิงบวกเป็นศูนย์ ค่าความระลึกได้เท่ากับ 100 เปอร์เซ็นต์เพราะค่าเท็จเชิงลบเป็นศูนย์ และค่าเอฟ-เมเชอร์เท่ากับ 100 เปอร์เซ็นต์ แสดงให้เห็นว่าขั้นตอนวิธีการหาค่าการวัดความคล้ายคลึงเชิงมุมโคไซน์ของระบบนายหน้าที่ออกแบบไว้มีความถูกต้องในการตัดสินใจในการเลือกแผนแบบเครื่องบริการที่มีความเหมาะสมกับความต้องการใช้บริการของผู้ใช้งาน

เอกสารอ้างอิง

ภาษาไทย

สุจิตรา อุดลย์เกษม อภิรัตน์ สุวรรณเพ็ชร นิชาภัทร ผ่องใส และ อรวรรณ เขาวลิต. (2558). “ต้นแบบระบบการควบคุมและจัดการเครื่องจักรและอะไหล่เครื่องจักร: กรณีศึกษาโรงงานผลิตน้ำตาล.”

Veridian E-Journal Science and Technology Silpakorn University ฉบับที่ 2

(กรกฎาคม-ธันวาคม): 89-101.

สุภาะ จันทา และณลินภัทร์ ปรวัฒน์ปรีयर. (2556). “ระบบจำแนกและค้นคืนข้อมูลเว็บกระหู่ขาว ด้วยโครงข่ายประสาทเทียมเปอร์เซ็ปตรอนแบบหลายชั้น.” วารสารเทคโนโลยีสารสนเทศ 9, 2 (กรกฎาคม - ธันวาคม): 14-19.

เสกสรรค์ วิสัยลักษณ์ วิภา เจริญภัณธารักษ์ และดวงดาว วิชาดากุล. (2556). “การใช้เทคนิคการทำเหมืองข้อมูลเพื่อพยากรณ์ผลการเรียนของนักเรียนโรงเรียนสาธิตแห่งมหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน ศูนย์วิจัยและพัฒนาการศึกษา.” Veridian E-Journal Science and Technology Silpakorn University ฉบับที่ 2 (กรกฎาคม-ธันวาคม): 1-17.

ภาษาต่างประเทศ

Amazon Web Services, Inc. (2016). **Amazon Elastic Compute Cloud (EC2)**. Accessed January 4, 2016, Available from <http://aws.amazon.com/ec2>

Apache CloudStack. (2016). "Apache CloudStack Open Source Cloud Computing." Apache Software Foundation, Accessed January 4, 2016, Available from <https://cloudstack.apache.org>

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I., and Zaharia, M. (2009). **Above the Clouds: A Berkeley View of Cloud Computing**. Accessed January 4, 2016, Available from <https://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>

Banditwattanawong, T., Masdisornchote, M., and Uthayopas, P. 2016. "Multi-provider cloud computing network infrastructure optimization." **Future Generation Computer Systems**, Vol 55 (February) , (ScienceDirect): 116-128.

CAT Telecom, Inc. (2015). **IRIS platform innovative cloud ecosystem by CAT**. Accessed December 7, 2015, Available from <http://iris.cattelcom.com>

Eidoon, Z., Nasser, Y., and Farhad, O. 2007. "A Vector Based Method of Ontology Matching." **Third International Conference on Semantics, Knowledge and Grid (IEEE)**: 378-381.

Gareth Williams. (2011). **Linear Algebra With Applications 6th Edition**. 76th ed. USA: Jones & Bartlett Pub.

Gartner, Inc. (2016). **Gartner's 2015 Hype Cycle for Emerging Technologies Identifies the Computing Innovations That Organizations Should Monitor**. Accessed January 4, 2016, Available from <https://www.gartner.com>

Google, Inc. (2016). **COMPUTE ENGINE Scalable, High-Performance Virtual Machines**. Accessed January 4, 2016, Available from : <https://cloud.google.com/compute/>

- Huapai, S., and Banditwattanawong, T. 2015. "An Interoperability Ontology for Multi-Cloud Computing Platforms." **International Conference on e-Business (iNCEB2015)**, November 23-24.
- Intakosum, S., and Muangon, W. 2007. " Retrieving Model for Design Patterns ECTI Transactions on Computer and Information Technology (ECTI-CIT) 3, 1 (May): 51-55.
- ITA. U.S. Department of Commerce (2016). **2016 Top Markets Report Cloud Computing**. Accessed April 1, 2016, Available from http://trade.gov/topmarkets/pdf/Cloud_Computing_Top_Markets_Report.pdf
- Jovita, Linda, Hartawan A., and Suhartono D. 2015. "Using Vector Space Model in Question Answering System." **International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)**, *Procedia Computer Science* 59, (ScienceDirect): 305-311.
- Liu, H., Bao, H., Wang, J., and Xu, D. (2010). "A Novel Vector Space Model for Tree based Concept Similarity Measurement." **International Conference on Information Management and Engineering (ICIME)**, 16-18 (April), (IEEE): 144-148.
- Mell, P. and Grance, T. (2011). **The NIST definition of cloud computing**. Special Publication 800-145, Accessed February1, 2016, Available from <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
- OpenStack. (2015). **Welcome to OpenStack Documentation**. Accessed January 4, 2016, Available from <http://docs.openstack.org/>
- True Internet Data Center, Inc. (2015). **true IDC enabling digital economy**. Accessed December 7, 2015, Available from <http://www.trueidc.com/en>
- VMware, Inc. (2016). **VMware ESXi overview**. Accessed April 1, 2016, Available from <https://www.vmware.com>
- Zhang, M., Ranjan, R., Haller, A., Georgakopoulos, D., Menzel, M., and Nepal, S. 2012. "An ontology-based system for Cloud infrastructure services discovery." **International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2012 8th)**, 14-17(October), (IEEE): 524-530.
- ZhiHao, Z., JiPing, H., Ting, D., and Yu, W. 2012. "Semantic Web Service Similarity Ranking Proposal Based on Semantic Space Vector Model." **International Conference on Intelligent Systems Design and Engineering Application**, 6-7(January), (IEEE): 917-920.