

การใช้เจนเนติกอัลกอริทึมสำหรับการประกอบกันของเว็บเซอร์วิสอย่างอัตโนมัติ

Using Genetic Algorithm for An Automated Web Services Composition

สันติ สวนศรี และ ดวงแก้ว สาวิมภัค

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยธรรมศาสตร์ ศูนย์รังสิต ปทุมธานี 12121

ดวงดาว วิชาดาภุล

ห้องปฏิบัติการ Bioinformatics หน่วยปฏิบัติการวิจัยกลางไปโอลิมปิก

ศูนย์พัฒนาวิศวกรรมและเทคโนโลยีชีวภาพแห่งชาติ (Biotec)

บทคัดย่อ

เว็บเซอร์วิสคือแนวคิดการพัฒนาซอฟต์แวร์ใหม่ที่สามารถกำหนดส่วนประกอบของซอฟต์แวร์ที่มีความเกี่ยวพันระหว่างส่วนต่างๆ ให้สามารถทำงานร่วมกันอย่างมีประสิทธิภาพ สามารถถูกเรียกใช้งานจากโปรแกรมประยุกต์อื่นผ่านระบบเครือข่ายคอมพิวเตอร์ ซึ่งในปัจจุบันมีเว็บเซอร์วิสที่ให้บริการเหมือนกันออกมากเป็นจำนวนมาก โดยแต่ละเว็บเซอร์วิสอาจมีมากกว่าหนึ่งจุดบริการ และสำหรับการประกอบกันของเว็บเซอร์วิสนั้น จำเป็นอย่างยิ่งที่จะต้องมีการคัดเลือกจุดให้บริการเพื่อนำมาระบบกันอย่างมีประสิทธิภาพ

งานวิจัยนี้จึงเสนอระบบการประกอบกันของเว็บเซอร์วิสอย่างอัตโนมัติโดยใช้เจนเนติกอัลกอริทึมในการคัดเลือกของเว็บเซอร์วิสที่มีจุดให้บริการอยู่หลายที่มาทำการประกอบกันเพื่อให้ได้รับสิ่งที่มีประสิทธิภาพดีที่สุด โดยทำการพิจารณาคุณภาพของการให้บริการซึ่งประกอบไปด้วย เวลาในการทำงาน จำนวนเงินที่ต้องจ่าย จำนวนผู้ใช้งาน และคุณภาพของเว็บเซอร์วิส เป็นต้น แล้วนำมาคำนวณโดยใช้เจนเนติกอัลกอริทึม ซึ่งจะคำนวณค่าfitness ของแต่ละจุดบริการ แล้วเลือกจุดที่มีค่าfitness มากที่สุด ซึ่งจะนำไปใช้ในการประกอบกันของเว็บเซอร์วิส

จากการทดลองพบว่าการประกอบกันของเว็บเซอร์วิสโดยใช้เจนเนติกอัลกอริทึมนั้นให้ผลที่ดีกว่าการใช้วิธีคัดเลือกจุดให้บริการ เว็บเซอร์วิสแบบ碰碰运气 ที่ต้องการบริการก่อนและแบบสุ่มในกรณีที่มีจุดให้บริการเว็บเซอร์วิสบางตัวไม่สามารถทำงานได้

คำสำคัญ : เจนเนติกอัลกอริทึม เว็บเซอร์วิส ซอฟต์แวร์

Abstract

Web service is a new software development paradigm, it can be defined as a loosely coupled collection of software components that represents a business function and can be accessed by another application over public networks using generally available protocols and transports. Currently, there are many web services that support same services and each web service has more than one instance of service. It is a necessity of web service instances selection for efficient composition.

In this paper, we propose an automated web services composition using Genetic Algorithm (GA) for selection of appropriate instances of web services based on QoS metrics regarding server and network performance. QoS metrics include response time, throughput, workload and services success rate.

Experimental result shows that our approach performs better than the selections of web service instances using First Come First Serve (FCFS) and random scheme when service instances are subjected to failure.

Keywords : web services paradigm Genetic Algorithm

1. บทนำ

ในปัจจุบันเว็บเซอร์วิส (web services) ได้เข้ามายึดบานาบทบาทสำคัญทั้งในด้านของการพัฒนาซอฟต์แวร์ และด้านของการประกอบธุรกิจ ในด้านของการพัฒนาซอฟต์แวร์นั้น เว็บเซอร์วิส ถือว่าเป็นแนวคิดใหม่ที่ช่วยให้ขัดจำกัดของการทำงานข้ามระบบ (cross platform) หมวดดังไป สามารถทำให้ซอฟต์แวร์ที่ถูกสร้างมาจากคนละภาษา ใช้รูนำข้อมูลคุณลักษณะเดียวกัน ทำงานบนระบบปฏิบัติการคนละตัว หรือทำงานบนเครื่องคอมพิวเตอร์ประเภท สามารถแลกเปลี่ยนข้อมูลซึ่งกันและกันได้ พร้อมกับทำงานร่วมกันได้ (interoperability) อย่างมีประสิทธิภาพ ซึ่งปัจจุบันมีการแบ่งประเภทเว็บเซอร์วิสเป็นสองประเภท ได้แก่ เว็บเซอร์วิสอย่างง่าย (simple web services) กับเว็บเซอร์วิสที่มีโครงสร้างซับซ้อน (complex web services) [1]

โดยทั่วไปของประเภทข้างต้นได้แก่ นักพัฒนาเว็บเซอร์วิสได้ พัฒนาออกแบบมาอย่างดี โดยที่แต่ละตัวได้ให้บริการที่แตกต่างกัน แบบเฉพาะตัว แต่ความเป็นจริงแล้วระบบต่างๆ ต้องการทำงานหลายด้านซึ่งเว็บเซอร์วิสเพียงหนึ่งตัวไม่สามารถทำงานให้แล้วเสร็จได้ จึงต้องการนำเว็บเซอร์วิสหลายๆ ตัวมาทำงานประกอบกัน จากความต้องการเหล่านี้ได้มีแนวคิดที่จะนำเว็บเซอร์วิสมาประกอบกัน (Web services composition) เพื่อทำให้เว็บเซอร์วิสอย่างง่ายสามารถทำงานเป็นเว็บเซอร์วิสที่มีโครงสร้างซับซ้อนได้

การประกอบกันของเว็บเซอร์วิสนั้นส่วนใหญ่จะเน้นไปที่ การประกอบกันเชิงความหมาย (semantic-based services composition) มากกว่า แต่ในระยะหลังได้มีการศึกษาการประกอบกันเชิงพิจารณาคุณภาพของการให้บริการ (QoS-aware services composition) [2] ด้วย แต่อย่างไรก็ตามเว็บเซอร์วิส ในปัจจุบัน มีอยู่หลายตัวที่ทำงานเหมือนกันและแต่ละตัวมีสถานที่ที่ให้บริการอยู่หลายที่ เช่นกัน ผู้สร้างเว็บเซอร์วิสที่เป็นชนิดแบบการนำมาประกอบกัน ไม่สามารถตัดสินใจได้ว่าจะใช้บริการตัวไหน และเมื่อมีการให้บริการ เว็บเซอร์วิสตัวที่ใช้อาจมี

การทำงานหน้าหรือไม่สามารถทำงานได้ ระบบการประกอบกันไม่สามารถที่จะเบลี่ยปะให้เว็บเซอร์วิสตัวอื่นอย่างอัตโนมัติได้

งานวิจัยนี้จึงมีวัตถุประสงค์ที่จะแก้ไขปัญหาดังกล่าว โดยมุ่งเน้นไปที่การประกอบกันของเว็บเซอร์วิสเชิงพิจารณาคุณภาพของการให้บริการเป็นหลัก และนำเจนเนติกอัลกอริทึมมาใช้ในการคัดเลือกเว็บเซอร์วิสที่จะนำมาใช้ในการประกอบกัน ตามคุณภาพของการให้บริการเป็นลำดับถัดไป

บทความนี้จะกล่าวถึงการนำเจนเนติกอัลกอริทึมที่เป็นวิธีการค้นหาคำตอบโดยมีพื้นฐานมาจากกระบวนการคัดเลือกทางพันธุศาสตร์มาใช้ในกระบวนการหาตัวเลือกที่ดีที่สุดในการประกอบกันอย่างอัตโนมัติของเว็บเซอร์วิส โดยเนื้อหาของบทความจะมีลำดับการนำเสนอต่อไปนี้ ทฤษฎีที่เกี่ยวข้อง สถาปัตยกรรมของระบบ รูปแบบโปรแกรมและพั้งก์ชันความเหมาะสม การทดลอง และบทสรุปของงานวิจัย

2. ทฤษฎีที่เกี่ยวข้อง

2.1 เจนเนติกอัลกอริทึม (Genetic Algorithm)

เจนเนติกอัลกอริทึมเป็นวิธีการค้นหาคำตอบโดยมีพื้นฐานมาจากกระบวนการคัดเลือกทางธรรมชาติ (natural selection) และกระบวนการคัดเลือกทางพันธุศาสตร์ (natural genetic selection) [3] ซึ่งคิดค้นโดย John Holland เมื่อปีค.ศ. 1975 โดยนำมาจากแนวคิดที่ว่าสิ่งมีชีวิตทั้งหลายมีส่วนตัวและส่วนที่ไม่ตัวซึ่งลักษณะที่ดีก็ย้อมมือก้าสอยู่รอดได้มากกว่าและจะได้รับการสนับสนุนให้มีการถ่ายทอดพันธุกรรมไปยังรุ่นลูกหลานได้มากกว่าเช่นกัน

ในการแก้ปัญหาโดยใช้เจนเนติกอัลกอริทึมนั้น ได้กำหนดปัญหาเท่ากับโครงโต้มหันง ซึ่งประกอบไปด้วยบีนส์ลักษณะต่างๆ เปรียบเหมือนกับตัวแสดงค่าคำตอบของปัญหาที่แบ่งผันไปตามการประยุกต์ใช้งานซึ่งได้แก่ ตัวแปร พารามิเตอร์ เป็นต้น หรือข้อกำหนดต่างๆ ที่เป็นองค์ประกอบของปัญหา ดังนั้น

การกำหนดครูปแบบโครงโน้มของแต่ละปัญหาโดยการแปลงตัวแปรพารามิเตอร์ ผ่านไปหรือข้อกำหนดต่างๆ ให้อยู่ในรูปแบบของยีนส์บนโครงโน้ม โดยประกอบไปด้วยบิต (bit) หรืออักษร (character) ซึ่งลักษณะต่างๆ ที่เป็นไปได้ของแต่ละยีนส์คือค่าของบิต หรือค่าตัวแปร พารามิเตอร์ ต่างๆ ที่เป็นไปได้ ส่วนฟังก์ชันความเหมาะสม (fitness function) เป็นฟังก์ชันที่กำหนดค่าความเหมาะสม (fitness) ของแต่ละโครงโน้ม เปรียบเหมือนกับค่าความสามารถในการอยู่รอดของแต่ละโครงโน้ม และยังเป็นฟังก์ชันที่กำหนดโอกาสหรือสัดส่วนที่แต่ละโครงโน้มจะถูกคัดเลือกด้วย

การทำงานของเจนเนติกอัลกอริทึมเริ่มต้นด้วยการสร้างประชากรต้นกำเนิด (initial population) จากการสุ่มสร้างค่าแต่ละบิตของแต่ละโครงโน้ม และทำการวิเคราะห์ค่าความเหมาะสมแต่ละโครงโน้ม หลังจากนั้นทำการสร้างชุดโครงโน้มต้นแบบ (mating pool) หรือชุดโครงโน้มพ่อแม่ที่สามารถอยู่รอดเป็นต้นแบบ และสุดท้ายทำการดำเนินการทางพันธุศาสตร์โดยการสุ่มจับคู่โครงโน้มต้นแบบเพื่อสร้างประชากรรุ่นใหม่ [4] ซึ่งการดำเนินการนี้มีดังต่อไปนี้

1. การวีเบรตต์กัชั่น (Reproduction) หรือการสืบพันธุ์ คือกระบวนการคัดเลือกโครงโน้มที่มีความเหมาะสมสูงเพื่อเป็นค่าต้นเริ่มต้นให้กับประชากรรุ่นต่อไป

2. การครอสโซเวอร์ (Crossover) หรือการไขว้เปลี่ยน โดยทำการแลกเปลี่ยนส่วนของโครงโน้มพ่อแม่ (parent) ตามอัตราความน่าจะเป็นในการครอสโซเวอร์ (probability of crossover: P_c) เพื่อสร้างชุดโครงโน้มรุ่นใหม่หรือโครงโน้มรุ่นลูก (offspring)

3. การมิวเทชั่น (Mutation) หรือการกลายพันธุ์ โดยการกลับค่าบางส่วนของโครงโน้มเป็นค่าใหม่ในตำแหน่งที่สุ่มได้ตามอัตราความน่าจะเป็นในการมิวเทชั่น (probability of mutation: P_m) ที่กำหนด

2.2 การประกอบกันของเว็บเซอร์วิส

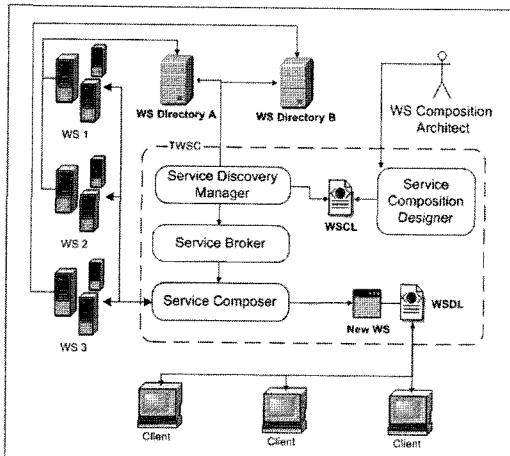
การประกอบกันของเว็บเซอร์วิส (Web Services Composition) เป็นเทคนิคของการนำเว็บเซอร์วิสชนิดง่ายๆ (simpler services) มาประกอบกันเป็นเว็บเซอร์วิสที่มีความซับซ้อน (complex services) ตามความต้องการ (arbitrarily)

ของผู้ประกอบ [5] ซึ่งมีอยู่ 2 แบบคือการประกอบกันแบบคงที่ (static) และการประกอบกันแบบไนนา米ค (dynamic) [6] การประกอบแบบคงที่จะทำการกำหนดเว็บเซอร์วิสไว้ล่วงหน้าในขั้นตอนออกแบบ ส่วนการประกอบแบบไนนา米คจะทำการ run-time คือทำการค้นหาเว็บเซอร์วิสขนาดเว็บเซอร์วิสทำงานโดยกระบวนการประกอบส่วนใหญ่จะมีกระบวนการค้นหาบริการ (discovery) กระบวนการรวมกันของเว็บเซอร์วิส (integration) และกระบวนการดำเนินงาน (execution) ซึ่งการประกอบกันนั้นทำโดยการวางแผนในรูปแบบของสายงาน (workflow) เพื่อระบุว่าเว็บเซอร์วิสใดจะต้องต่ออันเป็นเว็บเซอร์วิสตัวใด ในทิศทางใด

ต่อมาได้มีการแบ่งประเภทของการประกอบกันของบริการไว้ 2 ชุด [7] โดยชุดแรกคือ proactive และ reactive composition การประกอบกันแบบ proactive เป็นการทำงานแบบ offline คือทำการประกอบก่อนที่บริการนั้นจะเริ่มทำงาน การประกอบกันแบบ reactive เป็นการทำงานในขณะที่บริการนั้นมีการเรียกใช้งานอยู่ (real-time) ซึ่งจำเป็นต้องมีการจัดการการประกอบกัน (composition manager) และต้องคำนึงถึง bandwidth ของเครือข่ายด้วย ส่วนชุดที่สองคือ mandatory และ optional composition การประกอบกันแบบ mandatory เป็นแบบที่ทุกๆ เว็บเซอร์วิสต้องมีส่วนร่วม (participate) ต่อการทำงานนั้นด้วย แต่การประกอบกันแบบ optional เป็นการประกอบที่ไม่จำเป็นที่ทุกๆ เว็บเซอร์วิสต้องมีส่วนร่วมต่อการทำงานของเว็บเซอร์วิสที่ประกอบกันขึ้นมา

3. สถาปัตยกรรมของระบบ

งานวิจัยนี้ได้นำเสนอระบบการประกอบกันของเว็บเซอร์วิสโดยการทำงานจะเริ่มต้นที่ส่วนการค้นหาบริการทำการค้นหาบริการที่ต้องการ หลังจากนั้นทำการคัดเลือกบริการที่เหมาะสมในการนำมายประกอบกัน เมื่อได้บริการเรียบร้อยแล้วก็ทำการประกอบกันเพื่อนำไปใช้งานต่อไป ดังนั้นสถาปัตยกรรมของระบบแสดงดังรูปต่อไปนี้พร้อมกับรายละเอียดของแต่ละส่วนประกอบ



รูปที่ 1 สถาปัตยกรรมของระบบ

1. Service composition designer โดยในส่วนนี้เป็นส่วนของการทำการออกแบบสายงานของการทำงานเว็บเซอร์วิส ต่างๆ ที่ต้องการนำมาใช้งานร่วมกันหรือมาทำการประกอบกันโดยผู้ที่ทำการออกแบบนั้นจำเป็นต้องรู้ว่างานนั้นๆ มีเว็บเซอร์วิส ใดเกี่ยวข้องบ้าง อุปกรณ์ที่ได้บ้าง ผลของการออกแบบสายงาน การประกอบกันของเว็บเซอร์วิสในส่วนประมวลผลนี้จะได้เอกสารภาษา XML ซึ่งมีรายละเอียดการเรียกใช้เว็บเซอร์วิสต่างๆ ตามความต้องการ เอกสารดังกล่าวนี้เรียกว่า WSDL หรือ Web Service Composition Language

2. Service discovery manager เป็นส่วนในการจัดการค้นหาเว็บเซอร์วิสเพื่อนำมาใช้ในการประกอบกันของเว็บเซอร์วิส ในส่วนนี้จะทำการปรึกษาหรืออ่านค่าจากเอกสาร WSDL ที่ได้จากการออกแบบสายงานเกี่ยวกับการประกอบกันของเว็บเซอร์วิส ว่าในการทำงานแต่ละครั้งจำเป็นต้องใช้เว็บเซอร์วิสใดบ้าง และจากที่ไหน แล้วส่วนนี้จะทำการคิดต่อไปยังส่วนในการเก็บข้อมูล เว็บเซอร์วิสที่อยู่ภายใต้ระบบเพื่อดันหน้าเว็บเซอร์วิสตั้งกล่าว พร้อมนำข้อมูลรายละเอียดต่างๆ ของเว็บเซอร์วิส เช่นข้อมูลทางด้าน QoS ข้อมูลรายละเอียดของแต่ละ instance ของแต่ละ เว็บเซอร์วิสว่าอยู่ที่ server ใด เป็นต้น กลับเข้ามา

3. Service broker ส่วนนี้จะทำการคัดเลือก instance ของเว็บเซอร์วิสต่างๆ ตามรายละเอียดที่ได้จากส่วนก่อนหน้านี้ โดยใช้หลักการเจนเนติกอัลกอริทึม เพื่อให้ได้เส้นทางของ

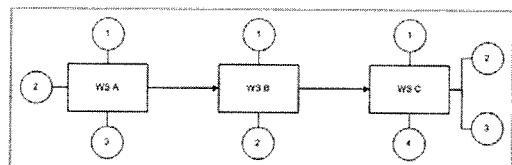
สายงานที่เหมาะสมตามคุณสมบัติทางด้าน QoS ที่ต้องการ ผลที่ได้จะเป็นตารางสายงานที่มีการคัดเลือกเส้นทางของแต่ละ instance แล้ว โดยเรียงลำดับตามคุณภาพของแต่ละสายงาน โดยคุณภาพเดิมที่สุดจะถูกเลือกใช้งานมากกว่า ส่วนนี้ถือว่าเป็นส่วนที่สำคัญส่วนหนึ่งของงานวิจัยนี้

4. Service composer ส่วนนี้จะนำผลที่ได้จากการคัดเลือกและจัดลำดับจาก service broker มาทำการประกอบกัน โดยการสร้างเว็บเซอร์วิสตัวใหม่ ที่ทำหน้าที่รับข้อมูลจากผู้ร้องขอ แล้วทำการส่งต่อไปยังเว็บเซอร์วิสต่างๆ ตามสายงานที่ทำการตั้งกล่าว เว็บเซอร์วิสตัวใหม่นี้ทำงานเสมือนเป็นแบบ template โดยที่ผู้ร้องขอไม่รู้ว่าเว็บเซอร์วิสตั้งกล่าวทำงานเพียงแค่ส่งต่อไปยังเว็บเซอร์วิสตัวอื่นแล้วนำผลมาคืนกลับให้เท่านั้น

4. รูปแบบโครงໂຄຣໂໂສມและຟັງກື້ນຄວາມເໝາະສົມ

4.1 รูปแบบโครงໂຄຣໂໂສມ

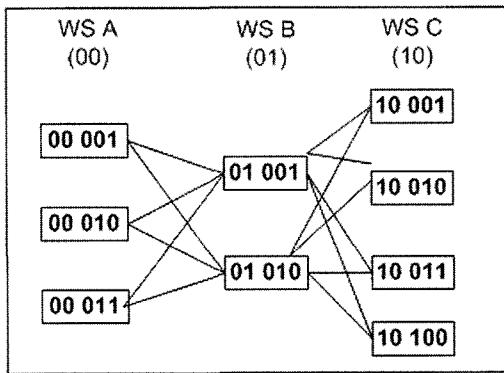
การกำหนดรูปแบบโครงໂຄຣໂໂສມนີ້ กระทำໄດ້ໂດຍການວິເຄາະທີ່ສາຍາການทำงานຂອງการປະກອບກັນຂອງເວັບເຊີຣີວິສ ທີ່ນັກອາແນບໄດ້ทำการອຳນວຍໃຫຍ່ໄວ້ ພ້ອມເກັບພິຈາລະນາສ່ວນທີ່ເກີຍກັບ instance ອີ່ຈຳນວນຂອງ instance ຂອງແຕລະເວັບເຊີຣີ



รูปที่ 2 ຕັວອຢ່າງສາຍານຂອງເວັບເຊີຣີທີ່ກຳນົດກັບກັນ

จากຕັວອຢ່າງດังກລ່າວ ສາຍານທີ່ເກີດຈາກ A-B-C ແນບຮະບຸ instance เชໍ່ນ A1-B2-C4 ຈະຖືກນຳມາເຂົ້າຮ້າທີ່ສເປັນເລີ່ມຈູາລວງເພື່ອແສດສັກນະໂຄຣໂໂສມທີ່ໃຊ້ໃນການກຳນົດກັບກັນແຕ່ທີ່ກຳນົດກັບກັນຈະປະກອບໄປດ້າວຍ 2 ສ່ວນເກີຍສ່ວນທີ່ອີບຍາເວັບເຊີຣີ ແລະ ສ່ວນທີ່ອີບຍາຄ່າ instance ຂອງເວັບເຊີຣີ ຈາກຕັວອຢ່າງທີ່ວ່າ ສາຍານ A1-B2-C4 ເມື່ອແປລັງເປັນເລີ່ມຈູາລວງແລ້ວ ກີ່ຄືການນຳຄ່າທີ່ເຂົ້າຮ້າສເປັນ

เลขฐานสองของห้าส่วนเว็บเซอร์วิสและส่วน instance ของเว็บเซอร์วิสนั้นมาจับคู่ต่อกัน ดังนั้นโครโน่ซึมของสายนี้คือ 000010101010100₂ ส่วนผลของการแปลงเป็นค่าเลขฐานสองห้าหมวดแสดงดังรูปด้านล่าง



รูปที่ 3 การเข้ารหัสเป็นเลขฐานสองของแต่ละเว็บเซอร์วิส

4.2 พังก์ชันความเหมาะสม

พังก์ชันความเหมาะสมของการประกอบกันของเว็บเซอร์วิสจะประกอบไปด้วย 2 ส่วนคือ (1) ส่วนของการหาค่าความสัมพันธ์ของค่า QoS ทั้ง 4 ค่า ซึ่งได้แก่ เวลาในการทำงานงานนั้นๆ (response time) โดยคิดเป็นวินทีที่อัตราของจำนวนเว็บเซอร์วิสที่สามารถทำงานแล้วเสร็จภายในช่วงเวลา (throughput) โดยคิดเป็นจำนวนการร้องขอ [8] จำนวนการร้องขอของบริการของเว็บเซอร์วิสในช่วงเวลาหนึ่งๆ (workload) โดยคิดเป็นจำนวนการร้องขอ และอัตราการประสบผลสำเร็จของการให้บริการของเว็บเซอร์วิส (service success rate) ในหนึ่งหน่วยเวลา โดยคิดเป็นจำนวนการร้องขอ และได้นำเอาค่าความต้องการ QoS ของผู้ร้องขอ (user preference) เพื่อใช้งานเว็บเซอร์วิสคิดร่วมด้วย โดยค่าดังกล่าวเป็นค่าที่กำหนดมาในตอนออกแบบสายการทำงานของการประกอบกันของเว็บเซอร์วิสซึ่งแต่ละค่าไม่เกิน 1 และ (2) ค่าความถูกต้องของค่าโครโน่ซึมซึ่งต้องเป็นค่าที่มีอยู่จริง โดยพังก์ชันความเหมาะสมของงานวิจัยนี้ได้แสดงดังสมการด้านล่าง

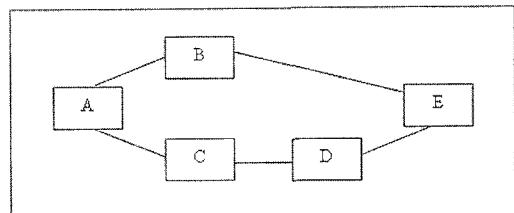
$$f(x) = \frac{up_r \sum_{i=1}^n T_{end-to-end} \max(up_w, up_{tp}) \max\left(\frac{W_i}{TP_i}\right)}{up_{ssr} \min(SSR_i)} \text{ (ChromValhd}(x)\text{)}$$

จากสมการข้างต้นจะอธิบายลักษณะที่ใช้ดังต่อไปนี้

1. x คือโครโน่ซึมที่ต้องการหาค่าความเหมาะสม
2. n คือจำนวน Web Services ทั้งหมดที่นำมา composition
3. $T_{end-to-end}$ คือค่า response time แบบ end-to-end โดยนำมาหารรวมแบบคิดเงื่อนไข
4. W คือค่า workload
5. TP คือค่า throughput
6. SSR คือค่า service success rate
7. up_r คือค่า user preference ของ QoS response time
8. up_w คือค่า user preference ของ QoS workload
9. up_{tp} คือค่า user preference ของ QoS throughput
10. up_{ssr} คือค่า user preference ของ QoS service success rate

11. ChromValhd(x) คือการหาค่าความถูกต้องของโครโน่ซึม x

การหาค่า response time แบบ end-to-end นั้นในกรณีที่มีเงื่อนไขดังตัวอย่างในรูปที่ 4 จะเห็นว่าเว็บเซอร์วิส A และ E มีโอกาสสูงทำงาน慢่อน ส่วน B และ C จะถูกทำงานก็ต่อเมื่อต้องพิจารณาเงื่อนไขเมื่อเว็บเซอร์วิส A ทำงานเสร็จ ส่วน D มีโอกาสทำงาน慢่อนแต่ C ต้องเป็นฝ่ายทำงาน ดังนั้นการคิดค่า response time ของการประกอบกัน ของเว็บเซอร์วิสแบบนี้



รูปที่ 4 ตัวอย่างการประกอบกันของเว็บเซอร์วิส 5 เว็บเซอร์วิส

จะคิดได้ดังสมการ

$$p_A T_A + \text{Max}(p_B T_B, p_C T_C + p_D T_D) + p_E T_E$$

จากสมการข้างต้น ค่า p คือค่าความน่าจะเป็นที่เว็บเซอร์วิสตัวนั้นจะถูกทำงาน ส่วนค่า T คือค่า response time ของแต่ละเว็บเซอร์วิส [9]

ส่วนการหาค่าความถูกต้องของโครโน่ช็อมเป็นการนำค่าความถูกต้องของค่าที่อยู่แต่ละแห่งของโครโน่ช็อมมาคิดด้วยโดยถ้าหากห้องนัด ซึ่งก็คือเป็นค่าที่มีอยู่จริงจะมีค่าเป็น 1 แต่ถ้าไม่ถูกเลยจะมีค่าเป็น 0 และในกรณีที่ถูกบังแจ้งว่าค่ามากกว่า 1 และมีค่าเท่ากับผลรวมของจำนวนชุดของค่าเว็บเซอร์วิสและค่า instance หากด้วยจำนวนชุดที่ถูกต้อง

ในการหาค่าตอบที่ดีที่สุดของแต่ละโครโน่ช็อม ค่าตอบที่ได้ต้องเป็นค่าที่ห้อยที่สุดและต้องมีค่ามากกว่า 0 และไม่เกิน 1 ซึ่งถ้าเกิน 1 หรือเป็น 0 แล้วโครโน่ช็อมนั้นย่อมจะมีค่าความถูกต้องน้อย หมายความว่าอาจมีบางค่าที่แทนเว็บเซอร์วิสหรือ instance ที่ไม่มีอยู่จริง

5. การทดสอบ

การทดสอบในงานวิจัยนี้เป็นการทดสอบประสิทธิภาพของเว็บเซอร์วิสที่เกิดจากการประกอบกันโดยใช้เจนเนติกอัลกอริทึมเบรียบเทียบกับเว็บเซอร์วิสที่เกิดจากการประกอบกันโดยใช้ช่วงอันมีรายละเอียดการทดสอบดังต่อไปนี้

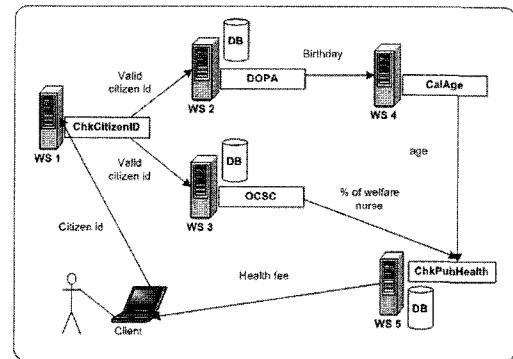
5.1 สภาพแวดล้อมที่ใช้ในการทดสอบ

1. เครื่องคอมพิวเตอร์ CPU: P4 2.4 GHz. และ Ram: 256 MB จำนวน 25 เครื่อง
2. Network: Fast Ethernet (100 Mbps)
3. ระบบปฏิบัติการ: Microsoft Windows XP SP2
4. Java Run-time 1.5.0
5. ชุดไลบรารี Apache Axis 1.2 RC3
6. ฐานข้อมูล MySQL
7. Apache Tomcat 5.5
8. โปรแกรมในการเรียกใช้บริการเว็บเซอร์วิสที่เกิดจากการประกอบกัน โดยโปรแกรมนี้ทำงานบน เครื่องคอมพิวเตอร์

CPU: Celeron 1 GHz. และ Ram: 256 MB นอกจากรันนิ่งส่วนแวดล้อมเดียวกันตั้งแต่ข้อ 2 ถึง 7

5.2 เว็บเซอร์วิสที่ใช้ในการทดสอบ

เว็บเซอร์วิสที่ใช้ทั้งหมด 5 ตัว นำมาเพื่อประกอบกัน เป็นเว็บเซอร์วิสระบบการตรวจสอบสิทธิ์ในการรักษาพยาบาลของคนไทย ซึ่งมีการทำงานดังรูป



รูปที่ 5 เว็บเซอร์วิสระบบการตรวจสอบสิทธิ์ในการรักษาพยาบาลของคนไทย

1. ChkCitizenID ทำหน้าที่ตรวจสอบว่าเลขที่บัตรประชาชนที่รับมาเป็นเลขที่บัตรจริง

2. DOPA เป็นเว็บเซอร์วิสของกรมการปกครอง (Department Of Provincial Administration) ทำหน้าที่ตรวจสอบบัณฑิตของประเทศไทยที่บัตรประชาชน

3. OCSC เป็นเว็บเซอร์วิสของสำนักงานคณะกรรมการข้าราชการพลเรือนหรือ กพ. (Office of the Civil Service Commission) ทำหน้าที่ตรวจสอบสิทธิ์และอัตราในการเบิกค่ารักษาพยาบาลจากเลขที่บัตรประชาชน

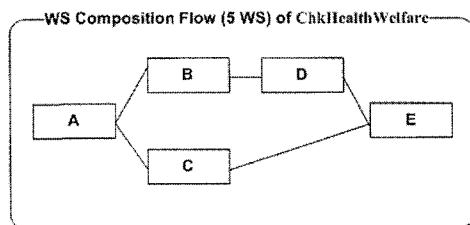
4. CalAge ทำหน้าที่คำนวณอายุจากวันเดือนปีเกิด

5. ChkPubHealth เป็นเว็บเซอร์วิสของกระทรวงสาธารณสุข (Ministry of Public Health) ทำหน้าที่ตรวจสอบอัตราการจ่ายค่ารักษาพยาบาลจากอายุและสิทธิการเบิกอื่นๆ

โดยแต่ละเว็บเซอร์วิสมีเครื่องให้บริการทั้งหมดชนิดละ 5 เครื่อง ในส่วนเว็บเซอร์วิสที่ 2, 3 และ 5 นั้นเป็นเว็บเซอร์วิสที่มี

การติดต่อฐานข้อมูล ซึ่งข้อมูลที่เก็บอยู่คุณลักษณะฐานข้อมูลกัน โดยทั้งเว็บเซอร์วิส 2 และ 3 มีข้อมูลที่เก็บไว้อย่างละ 50 record

เว็บเซอร์วิสที่เกิดจากการประกอบกันของเว็บเซอร์วิสทั้ง 5 ตัวคือ ChkHealthWelfare ที่สร้างโดยใช้ Axis เข้าเดียวกับ เว็บเซอร์วิสวัวเน่ฯ โดยจะทำหน้าที่ตั้งแต่รับค่าเลขที่บัตรประชาชนมาตรวจสอบว่าจริงหรือไม่ ถ้าจริงส่งไปทำงานยัง 2 เว็บเซอร์วิสคือการตรวจสอบวันเดือนปีเกิดจากกรมการปกครอง และตรวจสอบสิทธิ์ในการเบิกค่ารักษาพยาบาลของสำนักงานคณะกรรมการชาร์จการผลเรือน แล้วให้วันเดือนปีเกิดมาแล้วก็ ส่งไปคำนวนอายุปัจจุบันเพื่อส่งให้ตัวสุดท้ายพร้อมกับสิทธิ์การเบิกเพื่อตรวจสอบว่าจะต้องจ่ายค่ารักษาพยาบาลหรือไม่ ถ้าจ่าย จ่ายเป็นจำนวนเท่าไร โดยลักษณะการประกอบกันของเว็บเซอร์วิสนั้นแสดงดังรูป



รูปที่ 6 ลักษณะการประกอบกันของเว็บเซอร์วิสตรวจสอบสิทธิ์ การรักษาพยาบาล

5.3 วิธีการทดสอบ

การทดสอบเพื่อวัดประสิทธิภาพเว็บเซอร์วิสที่เกิดจากการประกอบกัน โดยการเลือก instance นั้นมีการเลือก 3 วิธีคือ

1. กระบวนการเจนเนติกอัลกอริทึมของงานวิจัยนี้
2. แบบคิริเอ็กอก่อนที่ได้รับบริการก่อน (FCFS)
3. การสุ่ม (random)

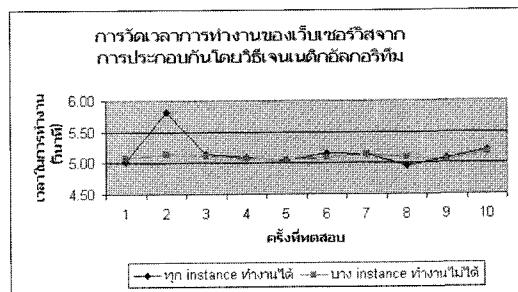
สำหรับการประกอบกันของเว็บเซอร์วิสแบบเจนเนติก อัลกอริทึมนี้มีการทดสอบเพื่อวัดเวลา response time ของแต่ละเว็บเซอร์วิสแต่ละ instance เพื่อนำผลที่ได้ไปเป็นข้อมูลนำเข้า QoS ของกระบวนการเจนเนติกอัลกอริทึมของ service broker หลังจากนั้นทำการทดสอบ service broker ด้วยข้อมูลของการประกอบกันของเว็บเซอร์วิสซึ่งได้กำหนดให้ข้อมูลของค่า

workload throughput และ service success rate ของทุก instance ของทุกเว็บเซอร์วิสค่าเท่ากันยกเว้นค่า response time ที่ได้มาจากการทดลองก่อนหน้า เพื่อหาค่าที่ instance ของแต่ละเว็บเซอร์วิสที่เหมาะสมที่สุด สำหรับการทดสอบในส่วนของวิธีเจนเนติกอัลกอริทึม โดยกำหนดค่า population และ generation เป็น 50

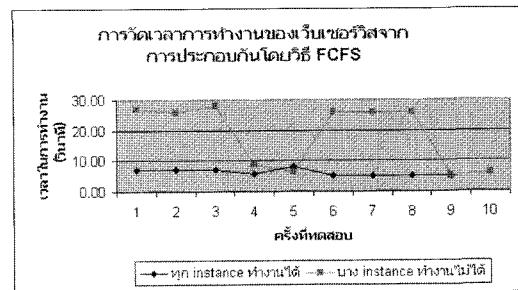
ทำการสร้างเว็บเซอร์วิสที่เกิดจากการประกอบกันโดยสร้างจาก 3 วิธีข้างต้น แล้วทดสอบจะมีอยู่ 2 ประเภทการทดสอบที่ทั้ง 3 วิธีต้องทดสอบเหมือนกันคือประเภทที่ instance บางตัวทำงานไม่ได้ กับประเภทที่ instance ทุกตัวทำงานได้หมด และนำมาเปรียบเทียบกัน

5.4 ผลการทดสอบ

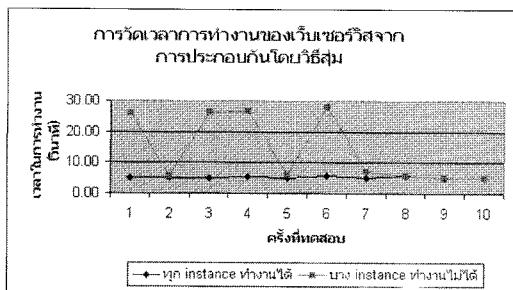
ผลการทดสอบประสิทธิภาพในงานวิจัยที่จะเสนอต่อไปเป็นผลลัพธ์ที่ได้จากการทดสอบทั้งหมดโดยนำเสนอในรูปแบบของแผนภูมิแสดงค่าเวลาในการทำงานของเว็บเซอร์วิสที่เกิดจากการประกอบกันซึ่งเปรียบเทียบกันตามประเภทการทดสอบโดยแยกเป็น 3 วิธี ซึ่งมีรายละเอียดดังต่อไปนี้



รูปที่ 7 เปรียบเทียบเวลาของวิธีเจนเนติกอัลกอริทึม



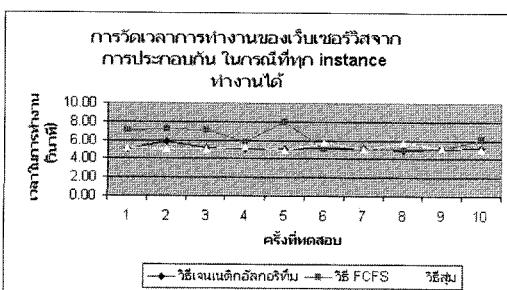
รูปที่ 8 เมริบเทียบเวลาของวิธี FCFS



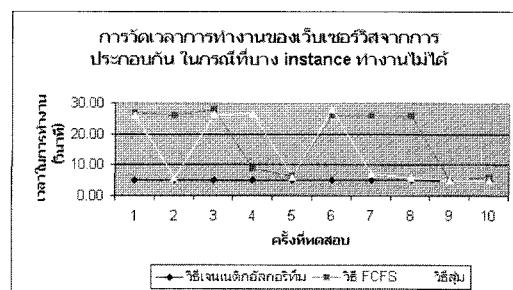
รูปที่ 9 เปรียบเทียบเวลาของวิธีสู่

จากรูปที่ 7 8 และ 9 เป็นการแสดงเวลาในการทำงานโดยเปรียบเทียบ 2 ประเภทคือประเภทที่ทุก instance ทำงานได้ ห้ามดักบัง instance ทำงานไม่ได้ ของทั้ง 3 วิธี จะเห็นว่าวิธี Jenen เนติกอัลกอริทึมมีเวลาใกล้เคียงกันมากทั้ง 2 ประเภท ซึ่งแตกต่างกันทั้ง 2 วิธี ทั้งนี้เนื่องจากการประกอบกันของเว็บเซอร์วิสโดยใช้เจนเนติกอัลกอริทึมนั้นได้มีการเลือก instance ที่มีค่าเหมาะสมไว้แล้วซึ่งสามารถรับประทานได้กว่าในการทำงานของเว็บเซอร์วิสที่ประกอบด้วยวิธีนี้มีประสิทธิภาพเพิ่มขึ้นอย่างมาก ซึ่งต่างจากอีก 2 วิธีที่ไม่สามารถคาดการณ์ได้กว่าในการเข้าไปเรียกใช้งานในแต่ละครั้งนั้นจะได้ instance ใดมาใช้ทำให้มีความสามารถรับประทานประสิทธิภาพของการให้บริการได้

นอกจากการวิเคราะห์ข้างต้น จากการทดสอบยังพบว่าในกรณีที่ทุก instance นัดทำงานได้หมด เวลาของการทำงานของเว็บเซอร์วิสที่เกิดจากการประกอบกันของทั้งวิธี Jenen เนติก อัลกอริทึมและวิธีสู่มีเวลาใกล้เคียงกันมาก ซึ่งแตกต่างกับในกรณีที่บัง instance ทำงานไม่ได้ โดยจะเห็นว่าวิธี Jenen เนติก อัลกอริทึมทำงานได้ดีกว่าทุกวิธี ดังรูป



รูปที่ 10 เปรียบเทียบเวลาของ 3 วิธีในการนัดทุก instance ทำงานได้



รูปที่ 11 เปรียบเทียบเวลาของ 3 วิธีในการนัดทุก instance ทำงานได้

ผลที่ได้ที่สุดคือค่าเวลาการทำงานที่น้อยที่สุดของแต่ละวิธี และลดระยะเวลาการทดลอง

6. บทสรุป

บทความนี้ได้นำเสนองานวิจัยการนำเจนเนติกอัลกอริทึมมาใช้ในการคัดเลือก instance ของเว็บเซอร์วิสมากทำการประกอบกันโดยพิจารณา QoS 4 คุณสมบัติคือ (1) response time (2) throughput (3) workload และ (4) service success rate ซึ่งในการพัฒนามีการออกแบบสถาบันเบ็ดกรุณาวงระบบการประกอบกันซึ่งประกอบไปด้วย (1) service composition designer (2) service discovery manager (3) service broker และ (4) service composer

จากการทดลองจะเห็นว่าเวลาในการทำงานของเว็บเซอร์วิสที่เกิดจากการประกอบกันโดยใช้วิธี Jenen เนติก อัลกอริทึมและวิธีสู่มีเวลาใกล้เคียงกันมาก ซึ่งแตกต่างกับในกรณีที่บัง instance ทำงานไม่ได้ โดยจะเห็นว่าจะเลือกตัวนั้นหรือไม่ สามารถทำงานได้ และสูงไปด้วยกว่าการใช้เจนเนติกอัลกอริทึม สำหรับการประกอบกันของเว็บเซอร์วิสหนึ่งเหมาะสมอย่างยิ่งสำหรับกรณีที่บัง instance ทำงานไม่ได้ เพราะชั้นตอนเจนเนติก อัลกอริทึมจะไม่ไปเลือก instance ตัวนั้นมาทำงาน ซึ่งต่างจากอีก 2 วิธีที่ไม่สามารถรับประทานได้กว่าจะเลือกตัวนั้นหรือไม่

7. กิตติกรรมประกาศ

ขอขอบพระคุณ พ.ท.ปรัชญา เศลิมวัฒน์ กองวิชา วิศวกรรมไฟฟ้า ส่วนการศึกษา โรงเรียนนายร้อยพระ

จุลจอมเกล้า ในการใช้คำปรึกษาทางด้านเงินเนติกอัลกอริทึม และขอขอบพระคุณมหาวิทยาลัยราชภัฏสวนดุสิต ศูนย์ พนิชยการสยาม ในกรณีอนุญาตให้ใช้ห้องปฏิบัติการคอมพิวเตอร์ สำหรับการทดลองงานนี้ด้วย

8. เอกสารอ้างอิง

- [1] ประจักษ์ ภูลามโนชวงศ์, เว็บเซอร์วิส เครื่องมือสำหรับ ธุรกิจยุคใหม่, บ. BEA Systems (ประเทศไทย) จำกัด, <http://www.mcot.net>.
- [2] L. Zeng, B. Benatallah, Anne H.H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, Qos-Aware Middleware for Web Services Composition, IEEE Transactions on Software Engineering, Vol. 30, No. 5, May 2004.
- [3] บำรเมศ ชูติมา และ จงกล เอี่ยมมิ, การประยุกต์ใช้ เงินเนติกอัลกอริทึมในการจัดสรรสมดุลสายการประกอบแบบ ผสานกันทั้งผสาน, วิทยานิพนธ์ปริญญาโท, จุฬาลงกรณ์ มหาวิทยาลัย, กรุงเทพฯ, 2544.
- [4] ภาณุจัน วงศ์วิภาวด, การจัดตารางสอนของโรงเรียนแบบ อัตโนมัติโดยจีเนติกอัลกอริทึม, วิทยานิพนธ์ปริญญาโท, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพฯ, 2532.
- [5] D. Chakraborty and A. Joshi, Dynamic Service Composition: State-of-the-Art and Research Directions, Technical Report, University of Maryland, Baltimore County, Baltimore, December 2001.
- [6] S. Chandrasekaran, G. Silver, John A. Miller, J. Cardoso and Amit P. Sheth, Web Service Technologies and Their Synergy with Simulation, University of Georgia, Proceedings of the 2002 Winter Simulation Conference.
- [7] D. Chakraborty, F. Perich, A. Joshi, T. Finin, Y. Yesha, A Reactive Service Composition Architecture for Pervasive Computing Environments, In 7th Personal Wireless Communications Conference (PWC 2002), Singapore, October 2002.
- [8] D.A. Menasce, QoS Issues in Web Services, IEEE Internet Computing, Nov. – Dec. 2002, pp. 72-75.
- [9] D.A. Menasce, Composing Web Services: A QoS View, IEEE Internet Computing, Nov. – Dec. 2004, pp. 88-90.