ULAP: ระบบเครือข่ายเพียร์ทูเพียร์แบบไร้โครงสร้างที่มีความรู้เกี่ยวกับตำแหน่ง
ULAP: An Unstructured Locality-Awareness Peer-to-Peer Network

ชูพันธุ์ รัตนโภคา
ภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์ วิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
1518 ถ.พิบูลสงคราม เขตบางซื่อ กรุงเทพฯ 10800 โทรศัพท์ : 0-2913-2500 ต่อ 6322
E-mail: choopanr@kmutnb.ac.th

บทคัดย่อ

ในปัจจุบันการประมวลผลผ่านเครือข่ายอินเทอร์เนตนั้นได้รับความสนใจอย่างมากทางการวิจัยด้านการประมวลผลประสิทธิภาพสูง เนื่องจากปัญหาทางด้านต่างๆ ไม่ว่าจะเป็นทางด้านวิทยาศาสตร์ หรือวิศวกรรมศาสตร์มีความซับซ้อนมากขึ้น ทำให้การประมวลผลโดยใช้เครื่องคอมพิวเตอร์ส่วนบุคคลเพียงเครื่องเดียวในการแก้ไขปัญหาต้องใช้เวลาประมวลผลที่นานมาก การประมวลผลผ่านเครือข่ายอินเทอร์เนตส่วนใหญ่ได้ประยุกต์นำเครือข่ายเพียร์ทูเพียร์มาใช้เนื่องจากคุณสมบัติของเครือข่ายที่สามารถรองรับการเปลี่ยนแปลงของเครื่องในระบบได้อย่างอัตโนมัติและรวดเร็ว อีกทั้งยังสามารถค้นหาเครื่องในระบบเครือข่ายได้อย่างง่ายดาย แต่อย่างไรก็ตามเครือข่ายเพียร์ทูเพียร์ส่วนใหญ่ยังไม่รองรับการเรียนรู้เกี่ยวกับตำแหน่ง ทำให้การค้นหาเครื่องคอมพิวเตอร์เพื่อใช้ในการประมวลผลผ่านเครือข่ายอินเทอร์เน็ตนั้นไม่มีประสิทธิภาพ บทความวิจัยนี้ต้องการนำเสนอการพัฒนาระบบเพียร์ทูเพียร์แบบไร้โครงสร้างที่สามารถเรียนรู้ตำแหน่ง(ULAP) รวมถึงทดลองและวิเคราะห์ผลการทดลองของระบบที่ได้พัฒนาขึ้นกับเครื่องคอมพิวเตอร์จำนวน 222 เครื่องในระบบกริดชื่อ Grid'5000 ณ ประเทศฝรั่งเศส

คำสำคัญ: เครือข่ายเพียร์ทูเพียร์, การประมวลผลประสิทธิภาพสูง, การเรียนรู้ตำแหน่ง

Abstract

Internet computing is currently an interesting topic in high performance computing research field, because, in many domains such as science domain or engineering domain, the problems become more complex. Thus, it takes a lot of time to solve the problem with a single desktop computer or even a single server machine. Internet computing platforms are mostly based on peer-to-peer network because it supports dynamic join/leave of nodes and it can easily discover resources (computer). However, most peer-to-peer network does not support locality-awareness. Thus, discovered resources on network for internet computing are not efficient. This article presents an implementation of unstructured peer-to-peer locality-awareness peer-to-peer network called ULAP. Also, the experiments of the implemented system are tested on 222 machines on a Grid platform called Grid'5000 in France.

Keywords: peer-to-peer network, high performance computing, locality-awareness

1. คำนำ

ระบบเครือข่ายเพียร์ทูเพียร์ได้รับความนิยมมากในปัจจุบัน ซึ่งส่วนใหญ่จะถูกนำไปใช้กับโปรแกรมแชร์แฟ้มข้อมูล เช่น Bittorrent [1], Gnutella [2] และอื่นๆ แต่ที่จริงแล้วเครือข่ายเพียร์ทูเพียร์สามารถนำไปประยุกต์กับการทำงานของโปรแกรมต่างๆได้อีกมากมาย เช่น โปรแกรมจำพวก instant message (IM) เช่น MSN Messenger [3] รวมถึงการนำไปใช้ในการประมวลผลประสิทธิภาพสูง ทั้งแบบกระจาย (Distributed Computing) และแบบขนาน (Parallel Computing) เช่น SETI@home [4], และ P2P-MPI [5] ตามลำดับ

อย่างไรก็ตามเครือข่ายเพียร์ทูเพียร์โดยทั่วไป ที่ถูกนำมาใช้เพื่อการประมวลผลประสิทธิภาพสูงยังขาดความรู้เกี่ยวกับตำแหน่งของตัวเองในเครือข่าย ทำให้การค้นหาเครื่องในเครือข่ายเพื่อใช้ในการคำนวณนั้นเป็นไปอย่างไม่มีประสิทธิภาพ เนื่องจากโปรแกรมประเภทการประมวลผลประสิทธิภาพสูงจำเป็นต้องมีการ รับ-ส่ง ข้อความระหว่างกัน ถ้าเครื่องที่ใช้ในการประมวลผลประสิทธิภาพสูง อยู่ห่างกันคนละซีกโลก จะทำให้เวลาที่ใช้ในการประมวลผลส่วนใหญ่สูญเสียไปกับการรับ-ส่งข้อความ ดังนั้นสำหรับการประมวลผลประสิทธิภาพสูง ถ้าระบบเครือข่ายเพียร์ทูเพียร์มีความสามารถในการเรียนรู้ตำแหน่งของเครื่องในระบบได้ ระบบจะพยายามค้นหาเครื่องที่อยู่ใกล้กันที่สุดก่อน เพื่อเป็นการลดเวลาในการรับ-ส่งข้อความระหว่างเครื่องที่ใช้ประมวลผลของโปรแกรมประมวลผลประสิทธิภาพสูง

บทความวิจัยนี้แบ่งออกเป็นหัวข้อต่างๆ ดังนี้ หัวข้อที่ 2 อธิบายโครงสร้างระบบเครือข่ายแบบเพียร์ทูเพียร์ หัวข้อที่ 3 การออกแบบและการทำงานของ ULAP หัวข้อที่ 4 แสดงผลการทดลองและวิเคราะห์ผลการทดลอง และ หัวข้อสุดท้ายบทสรุป

CP1-3

2. โครงสร้างระบบเครือข่ายเพียร์ทูเพียร์

จาก [6] ระบบเครือข่ายแบบเพียร์ทูเพียร์นั้นสามารถแบ่งเป็น 2 ประเภทใหญ่ๆ คือ ระบบเครือข่ายแบบเพียร์ทูเพียร์แท้ (pure peer-to-peer system) และระบบเครือข่ายแบบไฮบริด (hybrid peer-to-peer system) ระบบเครือข่ายที่ได้รับความนิยมมากในหมู่นักวิจัยคือแบบเพียร์ทูเพียร์แท้ เนื่องจาก เป็นระบบที่มีความทนทานต่อการล้มเหลวของเครือข่าย (Fault tolerance) และไม่มีปัญหาของการล้มเหลวที่จุดเดียว (Single point of failure) ในระบบเครือข่ายแบบเพียร์ทูเพียร์แท้นั้นยังแบ่งออกได้อีก 2 ลักษณะคือ แบบไร้โครงสร้าง (Unstructured) และแบบมีโครงสร้าง (Structured)

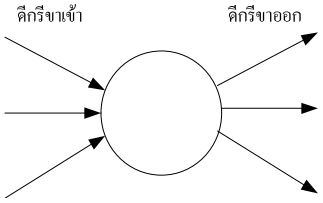
2.1 เครือข่ายเพียร์ทูเพียร์แบบไร้โครงสร้าง

ระบบเพียร์ทูเพียร์แบบไร้โครงสร้างได้ถูกพัฒนาขึ้นหลังจาก Napster [7] ระบบแชร์แฟ้มข้อมูลประเภท MP3 ที่มีลักษณะในการใช้เครื่องแม่ข่ายกลางทำหน้าที่ค้นหาเครื่องลูกข่ายที่มีแฟ้มข้อมูลนั้นๆ อยู่ ได้ถูกปิดตัวลงเนื่องจาก การละเมิดลิขสิทธิ์ของเพลง โดยระบบที่ถูกพัฒนาเป็นระบบแรกๆ คือ Gnutella ที่ไม่จำเป็นต้องมีเครื่องแม่ข่ายกลาง การค้นหาแฟ้มข้อมูล ทำโดยวิธีการฟลัด (flood) ชื่อแฟ้มข้อมูลที่ต้องการจะค้นหา โดยเครื่องที่ได้รับข้อความฟลัด ถ้ามีแฟ้มข้อมูลนั้นๆ อยู่เครื่องก็จะตอบสนองกับผู้ที่ทำการฟลัด ระบบแบบนี้มีข้อดีที่ระบบไม่มีทางที่จะถูกปิดตัวลงได้ เนื่องจากไม่มีเครื่องแม่ข่ายกลาง มีการเชื่อมต่อที่ป้องกันการล้มเหลวของระบบได้ค่อนข้างดี แต่ข้อเสียของระบบเครือข่ายประเภทนี้คือ การฟลัด สามารถทำให้เครือข่ายเกิดการอิ่มตัวได้ (saturate)

2.2 เครือข่ายเพียร์ทูเพียร์แบบมีโครงสร้าง

ระบบเพียร์ทูเพียร์แบบมีโครงสร้าง ได้รับความนิยมมากขึ้นเนื่องจากได้ทำการแก้ไข ปัญหาในการฟลัด ของเครือข่ายเพียร์ทูเพียร์แบบไร้โครงสร้าง โดยการแก้ไขนั้นได้ประยุกต์ก็ใช้งานของ DHT (Distributed Hash Table) ซึ่งจะเป็นการจับคู่ของค่าประจำตัวสิ่งของ (object identification, objectID) และ ค่าประจำตัวของเพียร์ (node identification, nodeID) กล่าวคือ เมื่อเพียร์หนึ่งต้องการที่จะกระจายข้อมูลเพียร์นั้นจะทำการคำนวณค่า objectID จากนั้นจะส่งต่อข้อมูลต่อไปเรื่อยๆ จนถึงเพียร์ปลายทางที่มีค่า nodeID ใกล้เคียงกับ objectID มากที่สุด เมื่อเพียร์ต้องการจะค้นหาข้อมูลก็จะทำงานในลักษณะเดียวกัน ซึ่งจะเป็นการกระจายภาระการทำงานของเพียร์ในระบบ และการส่งต่อข้อมูล มีแนวทางที่แน่นอนตามอัลกอริธึมต่างๆ ที่ถูกใช้ในแต่ละระบบโครงงานที่พัฒนาระบบเครือข่ายเพียร์ทูเพียร์แบบมีโครงสร้างนั้น เช่น Chord [8], Pastry [9], Tapestry [10] แต่อย่างไรก็ตามระบบเครือข่ายประเภทนี้มีปัญหาการดูแลรักษา(maintenance)สภาพของเครือข่าย และไม่สามารถรองรับกับระบบที่มีการเปลี่ยนแปลงอย่างรวดเร็ว

การทำงานของระบบเครือข่ายประเภทนี้เหมาะสำหรับงานที่ต้องการกระจายภาระให้เท่าๆกันทุกเพียร์ แต่ไม่เหมาะสมที่จะมาใช้ในการเรียนรู้ตำแหน่งของเพียร์รวมทั้งใช้ในการค้นหาเพียร์เพื่อการประมวลผลประสิทธิภาพสูง



รูปที่ 1 เพียร์ใน ULAP ที่มี degree (d) เท่ากับ 3

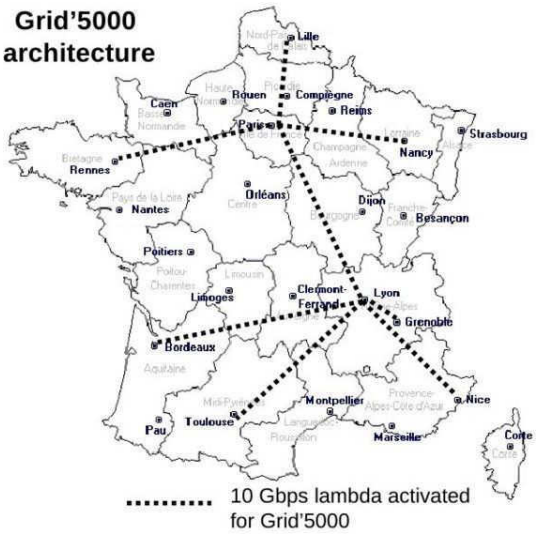
3. การออกแบบและการทำงานของ ULAP

ULAP (Unstructured Locality-Awareness Peer-to-Peer System) เป็นระบบที่ถูกพัฒนาขึ้นโดยใช้หลักการของระบบเครือข่ายเพียร์ทูเพียร์แบบไร้โครงสร้าง ซึ่งโดยการทำงานนั้นเพียร์จะทำการติดต่อกับเครื่อง bootstrap ซึ่งตัว bootstrap นี้จะทำหน้าที่ให้ IP address เริ่มต้นกับเพียร์ที่ร้องขอเพื่อเข้าร่วมระบบเครือข่าย bootstrap นี้ไม่ได้ทำงานในลักษณะรวมศูนย์ (Centrailized) เนื่องจาก เพียร์ทุกเพียร์หลังจากติดต่อกับ bootstrap เป็นครั้งแรก ก็จะเก็บ IP Address ของเพียร์อื่นไว้ใน แคช เพื่อที่จะสามารถติดต่อในครั้งต่อไปได้เอง โดยไม่ผ่าน bootstrap

เพียร์ใน ULAP จะมีการกำหนดจำนวนลิงค์ที่เชื่อมต่อกับเพียร์อื่นๆ ซึ่งเรียกว่า ดีกรี (degree, d) โดยจะมี ดีกรีขาเข้า (incoming) และขาออก (outgoing) เท่ากัน เพื่อป้องกันการเกิดพาร์ทิชั่นขึ้นในระบบเครือ ข่าย รูปที่ 1 แสดงตัวอย่างของเพียร์ใน ULAP ที่มีดีกรี(d) เท่ากับ 3 ซึ่งจะหมายความว่า เพียร์นี้มี ดีกรีขาเข้า และดีกรีขาออกเท่ากับ 3 ใน ULAP ผู้ใช้สามารถกำหนด ดีกรี(d) ได้เอง ถ้าไม่มีการกำหนด d จะมีค่าเป็น 5

การเพิ่มสามารถให้เพียร์ใน ULAP เรียนรู้ตำแหน่งนั้นเกิดจากการส่งข้อความ Ping กันในระดับของแอฟพลิเคชั่นเพื่อเป็นการตรวจสอบ สภาพความพร้อมของเพียร์ปลายทางไปในตัว โดยเพียร์ทุกเพียร์ (P) จะทำถามเพียร์เพื่อนบ้าน(Neighbor) ทั้งหมดที่เชื่อมต่ออยู่ (N) ถึงหมายเลข IP ทั้งหมดของเพียร์ที่เพื่อนบ้านนั้นๆได้เชื่อมต่ออยู่ (NN) จากนั้นเพียร์จะทำการส่งข้อความ Ping ไปหาทุกเครื่องในเซ็ต NN เพื่อวัดความห่างระหว่างกัน ถ้าผลที่ได้จากการ Ping พบว่า เวลาที่ส่งข้อมูลไป-กลับของข้อมูล Ping ระหว่าง P กับ เครื่องใน NN น้อยกว่าเวลาที่ส่งข้อมูลไป-กลับระหว่าง P กับ เครื่องใน N จะเป็นผลทำให้เพียร์ P ทำการยกเลิกการเชื่อมต่อกับ N แล้วไปเชื่อมต่อกับ NN แทน โดยเมื่อปล่อยเวลาซักระยะเพียร์ทุกเพียร์ที่อยู่ใกล้กันจะเริ่มมีการเชื่อมต่อเข้าสู่กัน

ทางด้านการค้นหาเครื่องเพื่อใช้ในการประมวลผลประสิทธิภาพสูงนั้น ULAP ใช้เทคนิคการการฟลัด โดยแต่ละเพียร์จะตรวจสอบข้อความที่ได้รับจากการฟลัด ซึ่งถ้าเคยได้รับมาแล้วข้อความ



รูปที่ 2 การเชื่อมต่อของ 9 ไซต์ ในระบบ Grid'5000

นั้นจะถูกโยนทิ้ง ไม่เช่นนั้นจะทำการส่งต่อข้อความ เพื่อลดปัญหาระบบเครือข่ายอิ่มตัว

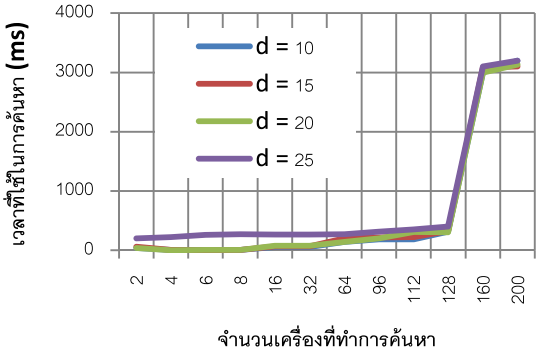
4. ผลการทดลอง

ULAP ได้ถูกทดลองบนระบบกริดของประเทศฝรั่งเศส ชื่อ Grid'5000 [11] ซึ่งเป็นระบบกริดที่เกิดจากการร่วมมือกันของสถาบันการ ศึกษาในฝรั่งเศส ระบบกริดแบ่งออกเป็น 9 ไซต์ ดังรูปที่ 2 คือ Rennes, Paris, Lille, Nancy, Lyon, Bordeaux, Toulouse, Grenoble และ Nice โดยที่แต่ละไซต์จะเชื่อมกันด้วยเครือข่ายความเร็วสูง 10 Gbps lambda ในปัจจุบันมีจำนวนเครื่องทั้งหมดในระบบ 3202 เครื่อง โดยมีจำนวนคอร์(core) ทั้งสิ้น 5714 คอร์

การทดลองในครั้งนี้ได้ทำการทดลอง บนเครื่องคอมพิวเตอร์จำนวนทั้งหมดจำนวน 222 เครื่อง ท่ามกลาง 4 ไซต์ คือ Nancy (Intel Xeon 5110 จำนวน 64 เครื่อง), Rennes (Intel Xeon L5420 จำนวน 64 เครื่อง), Nice (AMD Opteron 2218 จำนวน 64 เครื่อง) และ Lille (AMD Opteron 248 จำนวน 30 เครื่อง) โดยทุกเครื่องมี RAM อย่างน้อย 2 GB การเชื่อมต่อระหว่างไซต์ เวลาที่ใช้ในการรับส่งข้อความผ่านการ Ping แสดงในตารางที่ 1

	Nancy	Nice	Rennes	Lille
Nancy	-	16.3	11.3	9.7
Nice	16.3	-	18	15.9
Rennes	11.3	18	-	10.8
Lille	9.7	15.9	10.8	-

ตารางที่ 1 เวลาที่ใช้ในการ Ping ระหว่างไซต์ (ms)

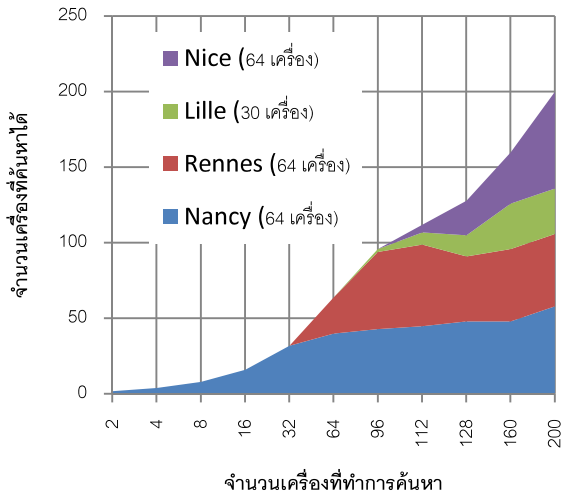


รูปที่ 3 เวลาที่ใช้ในการค้นหาเครื่องในเครือข่าย ULAP

การทดลองเริ่มต้นด้วยการสั่งหนึ่งเพียร์ ให้ค้นหาเครื่องในเครือข่าย โดยเพียร์ที่ถูกสั่งให้ค้นหาเครื่องอยู่ที่ไซต์ Nancy ในการทดลองได้ทำการวัดเวลาที่ใช้ในการค้นหาเครื่อง และตรวจสอบเครื่องที่ค้นหาพบ ว่ามีความใกล้ไกลต่างกันเพียงใด โดยการสั่งให้ค้นหาเครื่องในการทดลองจะสั่งให้ค้นหาเครื่องจำนวน 2, 4, 6, 8, 16, 32, 64, 96, 112, 128, 160 และ 200 เครื่องตามลำดับ แต่ละครั้งจะทำการทดลอง 5 ครั้ง เพื่อหาเวลาเฉลี่ยในการค้นหาเครื่อง

รูปที่ 3 แสดงเวลาที่ใช้ในการค้นหาเครื่องโดยเฉลี่ย โดยมีการปรับขนาดของเพียร์ดีกรี(d) ที่ 10, 15, 20 และ 25 ตามลำดับ จากการทดลองพบว่า เมื่อเพียร์มีดีกรีต่ำจะใช้เวลาในการค้นหาเครื่องที่น้อยกว่า ถ้าจำนวนของเครื่องที่ต้องการจะค้นหามีจำนวนน้อย เนื่องจากเมื่อเพียร์มีจำนวนดีกรีที่มากขึ้นจะเสียเวลาไปกับการฟลัด อีกทั้งเป็นการเพิ่มข้อความที่ไหลเวียนในเครือข่ายให้มากขึ้น ทำให้ระบบตอบสนองการทำงานช้าลง จะเห็นว่าการตอบสนองของ ULAP ในการค้นหาเครื่องในเครือข่ายนั้นค่อนข้างจะรวดเร็ว สามารถหาเครื่อง 128 เครื่องได้ภายในเวลา 0.5 วินาที อย่างไรก็ตามเมื่อจำนวนเครื่องที่ต้องการค้นหาเพิ่มขึ้นเป็น 160 และ 200 เครื่อง ULAP จำเป็นต้องใช้เวลาถึงประมาณ 3 วินาที ซึ่งเป็นเพราะมีข้อความไหลเวียนในระบบเป็นจำนวนมากจากการฟลัดทำให้เครือข่ายเริ่มอิ่มตัว (saturated) และตอบสนองช้าลง

ในการค้นหาเครื่องเพื่อใช้ในการคำนวณนั้น ด้วยคุณสมบัติของ ULAP ที่มีความสามารถในการพยายามหาเครื่องคอมพิวเตอร์ที่อยู่ใกล้กับตัวเองก่อน รูปที่ 4 แสดงผลการกระจายของเครื่องที่ค้นหาพบ จากรูป จะเห็นได้ว่าเมื่อค้นหาเครื่องตั้งแต่ 2 เครื่อง ถึง 32 เครื่อง ULAP ได้พบเครื่องที่อยู่ในไซต์ Nancy เท่านั้น จากนั้นเมื่อมีการค้นหาเครื่อง 64 เครื่อง ULAP ได้ค้นพบเครื่องที่อยู่ที่ Rennes จะได้ว่าจำนวนเครื่องที่ Nancy นั้นอันที่จริงแล้วเพียงพอสำหรับ การค้นหา 64 เครื่อง แต่เนื่องด้วยการเชื่อมต่อของเพียร์ใน ULAP นั้น มีการกำหนดจำนวนดีกรี (d) เพื่อให้การเชื่อมต่อมีประสิทธิภาพมากที่สุด เพียร์ใน ULAP พยายามจะหาเพียร์อื่นในการเชื่อมต่อจนเต็มจำนวนดีกรีที่กำหนด ทำให้บางเพียร์ที่ Nancy จะมีการเชื่อมต่อเข้ากับเพียร์ที่ Rennes และ Lille เนื่องด้วย Latency ของ Rennes และ Lille มีค่าใกล้เคียงกันมากทำระบบไม่สามารถ



รูปที่ 4 การกระจายของเครื่องที่ค้นหาได้ตามไซต์ต่าง ๆ

แยกแยะความเร็วในการเชื่อมต่อได้อย่างชัดเจน และในการค้นหา เครื่องบน ULAP นั้นจะนำเครื่องที่ตอบกลับเร็วที่สุดขึ้นคืนให้กับผู้ที่ทำการค้นหา จะเห็นได้ว่าเมื่อค้นหาเครื่องจำนวน 96 เครื่อง เพียร์ที่ Lille เริ่มตอบสนองต่อการค้นหา และ เพียร์ที่ Nice เริ่มตอบสนองเมื่อทำการค้นหาเครื่องตั้งแต่ 112 เครื่องขึ้นไป

5. สรุป

บทความวิจัยที่ได้นำเสนอการพัฒนาระบบเครือข่าย ULAP ที่เป็นระบบเพียร์ทูเพียร์แบบไร้โครงสร้าง และ มีความสามารถที่จะเรียนรู้ตำแหน่งของเพียร์ในระบบ จากการทดลองพบว่า ระบบ ULAP สามารถค้นหาเครื่องคอมพิวเตอร์ในระบบได้อย่างรวดเร็ว เมื่อจำนวนเครื่องที่ต้องการค้นหามีจำนวนอยู่ที่ประมาณ 128 เครื่อง แต่ ULAP สูญเสียประสิทธิภาพในการค้นหาเครื่องในระบบ เมื่อจำนวนเครื่องที่ต้องการค้นหามีจำนวนมากขึ้น เพราะเทคนิคการฟลัดเพื่อค้นหาเครื่องจำนวนมาก อาจทำให้เครือข่ายอิ่มตัว แต่อย่างไรก็ตามการค้นหาเครื่องคอมพิวเตอร์ในระบบแสดงให้เห็นว่า ULAP มีการเรียนรู้เกี่ยวกับตำแหน่งที่ค่อนข้างดี และ พยายามที่จะค้นหาเครื่องที่อยู่ใกล้ที่สุดก่อนเพื่อที่จะทำให้ โปรแกรมการประมวลผลประสิทธิภาพสูง ที่พัฒนาบน ULAP มีประสิทธิภาพมากขึ้น

สิ่งที่ควรจะพัฒนาต่อไปใน ULAP คือการลดจำนวนข้อความที่เกิดจากการฟลัด และ พัฒนาเทคนิคในการวัดความเร็วในการเชื่อมต่อระหว่างเพียร์ให้มีประสิทธิภาพมากขึ้น

6. กิตติกรรมประกาศ

ขอขอบคุณวิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือที่ให้ทุนในการวิจัย และระบบ Grid’5000 ประเทศฝรั่งเศสที่ให้พื้นที่ในการทดลอง ในการทดลองครั้งนี้

เอกสารอ้างอิง

[1] **Cohen, Bram.** Incentives Build Robustness in BitTorrent. [Online] พฤษภาคม 22, 2003. http://www.bittorrent.org/bittorrentecon.pdf.

[2] **Oram, Andy.** Gnutella. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies.* s.l. : O'reilly, 2001.

[3] *MSN Web Messager.* [Online] http://webmessenger.msn.com.

[4] *SETI@home.* [Online] http://setiathome.berkeley.edu.

[5] *P2P-MPI: A Peer-to-Peer Framework for Robust Execution of Message Passing Parallel Programs on Grids.* **S. Genaud, C. Rattanapoka.** 1, s.l. : Springer, 2007, Journal of Grid Computing, Vol. 5, pp. 27-42.

[6] **Schollmeier, R.** A Definition of Peer-to-Peer Networking for Classification of Peer-to-Peer Architectures. *Proceedings of the First International Conference on Peer-to-Peer Computing.* 2001.

[7] **Shirky, C.** Listening to Napster. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies.* s.l. : O'Reilly, 2001.

[8] **I. Stoica, et al.** Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. *Proceedings of the 2001 ACM SIGCOMM Conference.* 2001.

[9] **A. Rowstron, P. Druschel.** Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. *Lecture Notes in Computer Science.* 2001.

[10] *Tapestry: An Infrastructure for Fault-tolerent Wide-area Location and Routing.* **B. Y. Zhao, J. D. Kubiatowicz, A. D. Joseph.** s.l. : UC Berkeley, 2001.

[11] **F. Cappello, et al.** Grid'5000: A Large Scale, Reconfigurable, Controlable and Monitorable Grid Platform. *Grid'2005 Workshop.* 2005.

ชูพันธุ์ รัตนโภคา สำเร็จการศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมคอมพิวเตอร์ จากมหาวิทยาลัยเกษตรศาสตร์ จากนั้นสำเร็จการศึกษาระดับปริญญาโท (DEA d’informatique) และ ปริญญาเอก (Doctorat en Informatique) จาก Strasbourg University ประเทศฝรั่งเศส ขณะนี้เป็นอาจารย์ประจำภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์ วิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ มีความสนใจในการวิจัยในหัวข้อการประมวลผลแบบกระจาย และ การประมวลผลแบบขนาน เทคโนโลยีระบบเครือข่ายเพียร์ทูเพียร์ ระบบฐานข้อมูล และ เทคโนโลยีสารสนเทศ