

การทดสอบประสิทธิภาพของ 64-bit Linux Cluster

บทคัดย่อ

ซูเปอร์คอมพิวเตอร์ หรือเมนเฟรมคอมพิวเตอร์ เป็นคอมพิวเตอร์สมรรถนะสูง ซึ่งถูกออกแบบมาให้มีความสามารถในการคำนวณ และทำงานได้อย่างมีประสิทธิภาพดีกว่าคอมพิวเตอร์ส่วนบุคคลที่ใช้งานทั่วไป อย่างไรก็ตาม ต้นทุนในการผลิตประกอบกับการใช้อุปกรณ์ที่ไม่สามารถหาซื้อได้ตามท้องตลาด ทำให้คอมพิวเตอร์ประเภทนี้มีราคาสูงและเป็นอุปสรรคต่อการจัดหาเพื่อนำมาใช้งานในองค์กร โดยเฉพาะสถาบันการศึกษา และวิจัย คอมพิวเตอร์สมรรถนะสูงประเภทคลัสเตอร์ที่ประกอบขึ้นจากคอมพิวเตอร์ส่วนบุคคล จึงได้รับความนิยมให้เป็นที่เลือกในการทดแทนความต้องการในการคำนวณด้วยการใช้เทคนิคการประมวลผลแบบขนาน เอกสารฉบับนี้รายงานความคืบหน้าในการดำเนินโครงการ “การทดสอบประสิทธิภาพคลัสเตอร์”¹ ซึ่งจะเป็นคลัสเตอร์คอมพิวเตอร์สมรรถนะสูงเครื่องแรกของ

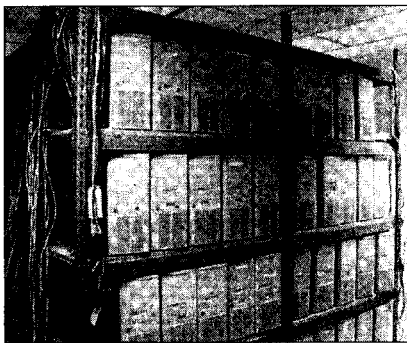
โรงเรียนนายร้อยพระจุลจอมเกล้าที่สามารถใช้งานร่วมกันได้โดยเฉพาะอย่างยิ่งกองวิชาในสาขาวิศวกรรมศาสตร์ และวิทยาการคอมพิวเตอร์ โดยรายงานนี้จะกล่าวถึงขั้นตอนและวิธีการทดสอบประสิทธิภาพของคลัสเตอร์ ด้วยการใช้โปรแกรมวัดประสิทธิภาพมาตรฐาน และโปรแกรมแบบขนานที่พัฒนาขึ้นเอง รวมถึงการประยุกต์ใช้งานคลัสเตอร์ในรูปแบบต่างๆ

1. บทนำ

เทคโนโลยีในการผลิตหน่วยประมวลผลกลาง หรือซีพียู สามารถเพิ่มความเร็วของสัญญาณนาฬิกาให้สูงขึ้นตามลำดับ ในปัจจุบันซีพียูที่มีความเร็วสูงสุดจะใช้สัญญาณนาฬิกาที่ความเร็ว 3 GHz หรือมากกว่า อย่างไรก็ตาม จำกัดัดในเรื่องการระบายความร้อนของสัญญาณ ขนาดและจำนวนของ

¹ โครงการนี้ได้รับการสนับสนุนเงินทุนวิจัยจาก NECTEC และเป็นส่วนหนึ่งของ “โครงการความร่วมมือทางวิชาการระหว่าง สวทช. และ รร.จปร.”

ทรานซิสเตอร์ที่บรรจุในซีพียูทำให้ยังไม่สามารถพัฒนาให้มีความเร็วที่มากไปกว่านี้ ความต้องการการคำนวณที่ต้องใช้คอมพิวเตอร์สมรรถนะสูงนั้นเป็นสิ่งผลักดันให้เกิดแนวคิดในการประมวลผลแบบขนาน (parallel processing) ซึ่งอาศัยหลักการทำงานหลาย ๆ อย่างพร้อมๆ กัน เช่น นำซีพียูหลายตัวมาเชื่อมต่อกัน เป็นระบบคอมพิวเตอร์แบบขนาน ด้วยโครงสร้างการเชื่อมต่อที่มีรูปแบบต่างๆ กัน เช่น แบบอาร์เรย์ วงแหวน ตาข่าย หรือไฮเปอร์คิวบ์ เป็นต้น พัฒนาการการประมวลผลแบบขนานเริ่มมีมาพร้อมๆ กับพัฒนาการของระบบคอมพิวเตอร์และบริษัทคอมพิวเตอร์ต่างๆ เช่น ไอบีเอ็ม ก็ได้มีการสร้างคอมพิวเตอร์แบบขนานที่มีประสิทธิภาพในการคำนวณ และสามารถรองรับการใช้งานของนักวิทยาศาสตร์ได้เป็นอย่างดี



รูปที่ 1: คลัสเตอร์ที่ประกอบขึ้นจากคอมพิวเตอร์ส่วนบุคคลทั่วไป

องค์การนาซ่า (NASA: National Aeronautics and Space Administration) แห่งประเทศสหรัฐอเมริกาเป็นผู้บุกเบิกในการสร้างคอมพิวเตอร์สมรรถนะสูงด้วยการนำคอมพิวเตอร์ส่วนบุคคลมาเชื่อมต่อกันและสามารถทำงานได้ใกล้เคียงหรือบางครั้งดีกว่าซูเปอร์คอมพิวเตอร์ที่มีราคาแพง นาซ่ามีความต้องการในการใช้งานคอมพิวเตอร์สมรรถนะสูงเป็นอย่างมาก ทั้งนี้เพื่อสนับสนุนการคำนวณทางวิทยาศาสตร์และการจำลองบนคอมพิวเตอร์ในรูปแบบต่างๆ ที่คอมพิวเตอร์ส่วนบุคคลไม่สามารถทำได้เนื่องจากข้อจำกัดในด้านประสิทธิภาพของซีพียู

ขนาดของหน่วยความจำ และความจุของดิสก์ การใช้งานระบบคอมพิวเตอร์สมรรถนะสูงยังมีข้อจำกัดในเรื่องการจัดการการใช้งาน เนื่องจากมีนักวิทยาศาสตร์ที่ต้องการใช้ค่อนข้างมาก และไม่พอเพียงต่อความต้องการ จึงเป็นแรงผลักดันให้เกิดโครงการเบอวูล์ฟ (Beowulf Project) เพื่อสร้างคอมพิวเตอร์สมรรถนะสูง ที่สามารถใช้เทคโนโลยีการประมวลผลแบบขนาน โดยใช้อุปกรณ์ที่ใช้ในคอมพิวเตอร์ส่วนบุคคลทั่วไปมาประกอบเข้าด้วยกัน และเชื่อมโยงด้วยระบบเครือข่ายแบบ LAN ที่มีราคาถูก และใช้ซอฟต์แวร์แบบรหัสเปิด (Open-source Software) ในการประมวลผลทำให้ลดค่าใช้จ่ายในด้านซอฟต์แวร์ลงไปด้วย [10]

การทดสอบประสิทธิภาพของระบบคลัสเตอร์มีความสำคัญอย่างยิ่งในการที่จะค้นหาวิธีการปรับแต่งโปรแกรม และตัวเลือกของคอมไพเลอร์ เพื่อให้โปรแกรมแบบขนานที่จะใช้งานบนคลัสเตอร์มีประสิทธิภาพมากที่สุด โดยทั่วไปจะทำการวัดค่าประสิทธิภาพหรือขีดความสามารถในการคำนวณของซีพียู ขีดความสามารถในการโอนย้ายข้อมูลระหว่างหน่วยความจำกับดิสก์และอุปกรณ์เชื่อมต่อ ขีดความสามารถในการรับส่งข้อมูลระหว่างซีพียูผ่านระบบเครือข่าย หรือขีดความสามารถโดยรวมในการประมวลผลแบบขนาน เมื่อได้ค่าประสิทธิภาพและทราบวิธีการที่จะใช้งานเครื่องคลัสเตอร์อย่างเต็มที่แล้วก็นำข้อมูลที่ได้ไปทำการปรับแต่งการคอมไพล์โปรแกรมประยุกต์ ด้วยค่าตัวเลือกต่างๆ ที่เหมาะสมกับระบบคลัสเตอร์ ก่อนที่จะนำมาใช้งานบนคลัสเตอร์ต่อไป การนำโปรแกรมประยุกต์มาใช้งานโดยไม่ทำการปรับแต่งมีผลทำให้ไม่สามารถใช้งานคลัสเตอร์หรือซูเปอร์คอมพิวเตอร์ได้อย่างเต็มประสิทธิภาพ

เอกสารฉบับนี้รายงานความคืบหน้าในการดำเนินโครงการทดสอบสมรรถนะของคลัสเตอร์คอมพิวเตอร์ และข้อมูลเบื้องต้นในการดำเนินโครงการ โดยในขั้นต้นจะกล่าวถึงงานวิจัยที่เกี่ยวข้อง ขั้นตอนในการทดสอบประสิทธิภาพของคลัสเตอร์ด้วย

โปรแกรมทดสอบมาตรฐาน และโปรแกรมที่จะพัฒนาขึ้นเองตลอดจนผลที่คาดว่าจะได้รับการดำเนินการ

2. งานวิจัยที่เกี่ยวข้อง

2.1 เบวูล์ฟคลัสเตอร์ (Beowulf Cluster)

ในปี ค.ศ. 1993 นาซ่าได้เริ่มโครงการเบวูล์ฟซึ่งดำเนินการโดย CESDIS (Center of Excellence in Space Data and Information Sciences) ซึ่งเป็นส่วนหนึ่งของ USRA (University Space Research Association) CESDIS ตั้งอยู่ ณ Goddard Space Flight Center มลรัฐแมริแลนด์ และได้รับการสนับสนุนทุนวิจัยบางส่วนจากโครงการ NASA Earth and Space Sciences (ESS) โดยมีจุดมุ่งหมายเพื่อที่จะแก้ไขปัญหาการใช้งานคอมพิวเตอร์ประเภทซูเปอร์คอมพิวเตอร์ขององค์การนาซ่าเอง เบวูล์ฟคลัสเตอร์เครื่องแรกถูกสร้างขึ้นจากแผงวงจรหลักของคอมพิวเตอร์ส่วนบุคคลที่ใช้ซีพียูของอินเทล 486 และใช้ระบบเชื่อมต่อแบบ Ethernet ความเร็ว 10 Mbps สร้างขึ้นมาเพื่อช่วยนักวิทยาศาสตร์ในการคำนวณข้อมูลที่มีขนาดใหญ่ [10] โดยมีขอบเขตในการสร้างคลัสเตอร์ โดยสรุปดังนี้

- เป็นเครื่องคอมพิวเตอร์แบบขนานที่มีราคาถูก (เมื่อเทียบกับคอมพิวเตอร์แบบขนานที่อยู่ในกลุ่มซูเปอร์คอมพิวเตอร์) สามารถประมวลผลโปรแกรมและข้อมูลแบบขนานได้
- ใช้อุปกรณ์ที่ทำได้ในท้องตลาด โดยเรียกหลักการนี้ว่า COTS (Commodity Off-The-Shelf) เช่นไมโครโพรเซสเซอร์ แผงวงจรแม่ หน่วยความจำ ดิสก์ และระบบเครือข่าย เช่นเดียวกับที่ใช้ในเครื่องคอมพิวเตอร์ส่วนบุคคล
- ใช้ซอฟต์แวร์ประเภทรหัสเปิด หรือโอเพ่นซอร์ส (Opensource) ที่ไม่ต้องเสียค่าใช้จ่าย ในเรื่องลิขสิทธิ์ เช่นระบบปฏิบัติการลินุกซ์ (Linux) หรือ MPI [9]
- ผู้ใช้เป็นนักวิทยาศาสตร์ หรือนักวิทยาการ

คอมพิวเตอร์ที่มีประสิทธิภาพในการใช้งานคอมพิวเตอร์ สามารถดัดแปลงโปรแกรมแบบขนานได้เอง และสามารถบริหารจัดการคลัสเตอร์ได้ด้วยตนเอง

หลังจากที่โครงการเบวูล์ฟได้ประสบความสำเร็จในการพิสูจน์เชิงหลักการและสามารถใช้งานจนเป็นที่ยอมรับในหมู่ผู้ใช้งานในวงการวิจัย จากนั้นรูปแบบในการสร้างคลัสเตอร์โดยอาศัยหลักการออกแบบของเบวูล์ฟคลัสเตอร์ได้รับความนิยมแพร่หลายในมหาวิทยาลัยชั้นนำที่มีการศึกษาในสาขาวิทยาการคอมพิวเตอร์และสาขาอื่นที่เกี่ยวข้อง รวมถึงหน่วยงานวิจัยในสหรัฐอเมริกาและต่อมาได้ขยายการสร้างคลัสเตอร์แบบเบวูล์ฟไปทั่วโลก [6][10][12]

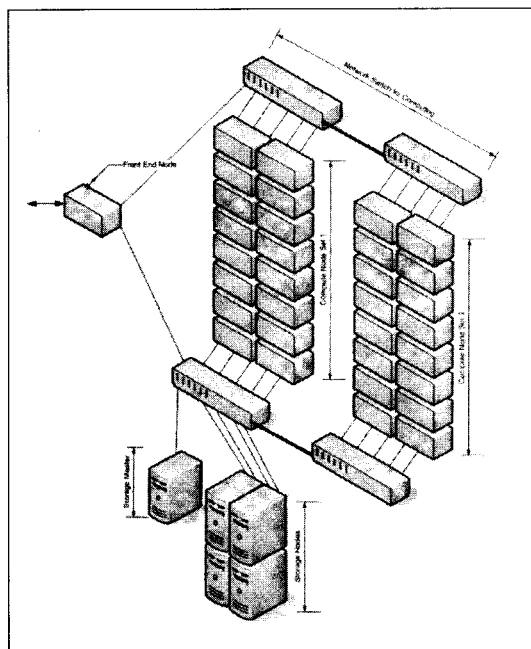
2.2 คลัสเตอร์ต่างๆ ในประเทศไทย

ในประเทศไทยเบวูล์ฟคลัสเตอร์ชุดแรกได้ถูกสร้างขึ้นมาที่คณะวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์ [11] ซึ่งปัจจุบันมหาวิทยาลัยชั้นนำในประเทศไทย ต่างก็สร้างคลัสเตอร์ของตนเอง เช่น จุฬาลงกรณ์มหาวิทยาลัย มหาวิทยาลัยเชียงใหม่ มหาวิทยาลัยวลัยลักษณ์ สถาบันพระจอมเกล้าพระนครเหนือ มหาวิทยาลัยเทคโนโลยีสุรนารี และสถาบันเอไอที (Asian Institute of Technology) รวมถึงศูนย์เทคโนโลยีอิเล็กทรอนิกส์ และคอมพิวเตอร์แห่งชาติ หรือ NECTEC

2.3 คลัสเตอร์ 64-bit Itanium2 ของ NECTEC

คลัสเตอร์เครื่องล่าสุดของ NECTEC (รูปที่ 2) เป็นคลัสเตอร์ที่ประกอบไปด้วยเครื่องคอมพิวเตอร์ที่เป็น Front-end หนึ่งเครื่องและมี 32 โหนดโดยที่แต่ละโหนดประกอบไปด้วยสองซีพียูของอินเทลแบบ Itanium2 และมีฮาร์ดดิสก์ แบบ SCSI ขนาด 36 Gbytes มีหน่วยความจำ 4 Gbytes และมีการเชื่อมต่อแบบ Gigabit Ethernet สองพอร์ต คลัสเตอร์นี้รันโปรแกรม Rock Cluster 3.3 ที่ใช้เคอร์เนลของลินุกซ์ 2.4.21-20.E.L ในส่วนที่เป็นหน่วยความจำสำรองประกอบไปด้วยเครื่องแม่ข่ายที่ใช้ซีพียู Xeon จำนวน 5 เครื่อง แต่ละเครื่อง มีฮาร์ดดิสก์แบบ SCSI

ขนาด 36 GBytes จำนวน 2 ตัว และใช้ระบบปฏิบัติการ Linux ของ RedHat เวอร์ชัน 9 ปัจจุบันเป็นเครื่องคลัสเตอร์ที่ทันสมัยที่สุดในประเทศไทย รายงานวิจัยฉบับนี้จะเป็นส่วนย่อยหนึ่งของโครงการ Itanium 2 คลัสเตอร์ของ NECTEC ที่ จะ ทำ การทดสอบประสิทธิภาพของซีพียูรุ่นใหม่ของอินเทลในการคำนวณเชิงขนาน



รูปที่ 2: โครงสร้างการจัดของ Itanium2 คลัสเตอร์ของ NECTEC

2.4 การประยุกต์ใช้งานคลัสเตอร์

การประยุกต์ใช้งานเบเวฟคลัสเตอร์นั้น มักนำมาใช้ในการประมวลผลข้อมูลขนาดใหญ่ หรือใช้ในการประมวลผล ที่ต้องใช้สมรรถนะของหน่วยประมวลผลที่เกินขีดความสามารถของคอมพิวเตอร์ส่วนบุคคล ตัวอย่างโครงการที่ใช้การคำนวณแบบขนานบนเครื่องคลัสเตอร์ในประเทศไทยได้แก่ งานวิจัยระบบสารสนเทศภูมิศาสตร์และฐานข้อมูลเชิงพื้นที่ [13] งานวิจัยวิทยาการเชิงคอมพิวเตอร์ [14] มีโครงการที่ใช้การคำนวณบนคลัสเตอร์ของ NECTEC

เช่น การจำลองแบบ H-mode scenarios ด้วยการใช้ integrated modeling code, การวิเคราะห์ 3D-QSAR ของ flavones โดย HIV-1 reverse transcriptase inhibitor การจำลองโมเลกุลของ chromone derivatives และ phthalimide derivatives เพื่อให้เป็น anti HIV - 1 drugs, การศึกษาการยึดจับระหว่างยาด้านมาลาเรีย 1,2,4-ไดรอกเซนกับฮีม โดยใช้วิธีการคำนวณต็อกกิ่ง, การค้นหาและออกแบบโมเลกุล (Drug Discovery and Molecular Design), การสังเคราะห์, ประเมินผล, และจำลอง Chromone Derivatives as HIV-1 Protease Inhibitors และการจำลอง Phthalimide Derivatives as HIV - 1 Reverse Transcriptase Inhibitors เป็นต้น

นอกจากนี้ยังมีการนำคลัสเตอร์หลายๆ คลัสเตอร์มาจัดการให้สามารถคำนวณร่วมกัน ผ่านเครือข่ายอินเทอร์เน็ต โดยมีชื่อโครงการว่า “โครงการ ThaiGrid” [15] [16] มีการประยุกต์ ใช้งานในการประมวลผลแบบขนาน เช่น โครงการ Efficient and Scalable Message Passing Model and Algorithms for Grid Systems, โครงการ Resources Management Algorithms for Scalable Grid Systems, โครงการ Drug Design using Quantum Chemistry, โครงการ 3D Animation Rendering Portal, โครงการ Clean Room Simulation using CFD, และโครงการ Remote Sensing Application for Crop Modeling on Grid เป็นต้น

งานวิจัยของผู้เขียนเองอยู่ในโครงการ High Performance Automatic Image Registration² มีวัตถุประสงค์เพื่อออกแบบโปรแกรมแบบขนานให้สามารถประมวลผลการทำ Image Registration [7] ซึ่งเป็นการค้นหาตำแหน่งในรูปแบบของ transformation ของภาพถ่ายดาวเทียมที่ได้มาใหม่จากภาพเก่าที่ได้ทำการลงทะเบียนไว้แล้ว โดยทำการคำนวณบนเครื่องคลัสเตอร์ของนาซ่าชื่อ egthow มีซีพียูจำนวน 128 ซีพียู จากนั้นวัดผลประสิทธิภาพการทำงาน

² ได้รับการสนับสนุนทุนวิจัยจากUSRA NASA Goddard Space Flight Center ผ่าน CESDIS และ Distributed Active Archive Center (DAAC)

และนำมาเปรียบเทียบการทำงานกับเครื่องซูเปอร์คอมพิวเตอร์ Cray T3D และ T3E ที่มี 256 ซีพียู โดยใช้โปรแกรมและข้อมูลเดียวกันมีผลงานตีพิมพ์ตามรายละเอียด (1)-(8)

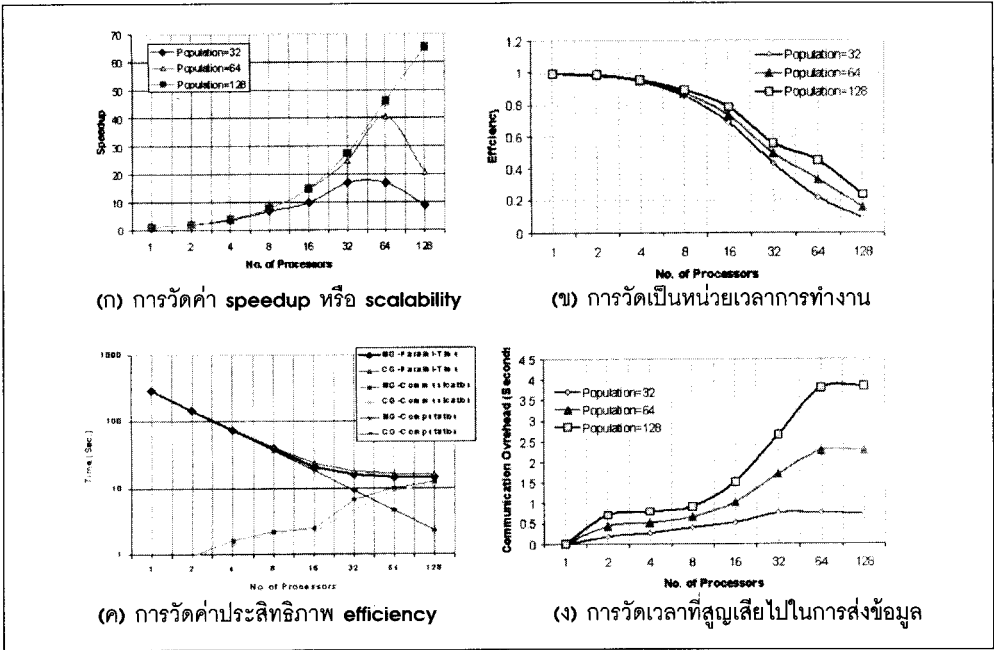
นอกจากนั้นยังมีโปรแกรมแบบขนานที่สามารถนำมาประยุกต์ใช้ในการออกแบบเชิงวิศวกรรม เช่น ซอฟต์แวร์ที่เกี่ยวข้องกับ Finite element: GeoFEM (<http://geofem.tokyo.rist.or.jp/>), Molecular dynamics package: Amber (<http://amber.scripps.edu/>), และ Electronic structure calculation program: Ab-init (<http://www.abinit.org/>) เป็นต้น

นอกจากโครงการทดสอบประสิทธิภาพเครื่องคลัสเตอร์ Itanium2 ซึ่งจะมีการสร้างไว้ ณ โรงเรียนนายร้อยพระจุลจอมเกล้าแล้ว ยังมีโครงการ CRMA Cluster ซึ่งเป็นโครงการงาน (Senior Project) ของนักเรียนนายร้อยในสาขาวิศวกรรมคอมพิวเตอร์และไฟฟ้าสื่อสารเพื่อสร้างคลัสเตอร์ขนาดเล็ก ในงบประมาณที่จำกัด โดยคลัสเตอร์ที่จะสร้างขึ้นนี้สามารถใช้เป็นเครื่องมือในการทดลองออกแบบโปรแกรมแบบขนานก่อนที่จะนำโปรแกรมนั้นไปใช้งานจริงบน

คลัสเตอร์ขนาดใหญ่ต่อไป

3. วิธีการวัดประสิทธิภาพของ Itanium 2 คลัสเตอร์

หลังจากที่ได้ทำการติดตั้ง Itanium 2 คลัสเตอร์ ณ โรงเรียนนายร้อยพระจุลจอมเกล้า เรียบร้อยแล้ว จะทำการวัดค่าประสิทธิภาพของคลัสเตอร์ โดยทั่วไป จะทำการวัดประสิทธิภาพของหน่วยประมวลผล และประสิทธิภาพของระบบเครือข่าย เปรียบเทียบ ค่าเวลาที่ใช้งานโดยเปลี่ยนจำนวนโพรเซสเซอร์ (ซีพียู) ไปทางขวาของกราฟ [2][5] ผลการวัดประสิทธิภาพแสดงใน รูปที่ 3 ซึ่งผู้ใช้งานมักมีความต้องการที่จะทราบ ค่า Speedup หรือ Scalability ดังแสดงในรูปที่ 3 (ก) เพื่อดูขีดความสามารถในการเพิ่มขยายระบบ หรืออาจวัดเป็นค่าของหน่วยเวลาโดยตรงเปรียบเทียบการทำงานเมื่อใช้จำนวนหน่วยประมวลผลที่ต่างกันไป เช่นในรูปที่ 3 (ข) กราฟของประสิทธิภาพในการคำนวณเช่นในรูปที่ 3 (ค) สามารถแสดงให้เห็นความคุ้มค่าในการเพิ่มจำนวนโหนดของคลัสเตอร์ เปรียบเทียบกับสมรรถนะที่จะได้รับ หรืออาจวัดผลในเชิงลบเช่น เวลาที่สูญเสียไปในการส่งข้อมูลไปมาระหว่างโหนดดังแสดงในรูปที่ 3 (ง)



รูปที่ 3 การวัดค่าประสิทธิภาพของคลัสเตอร์ [2] [5]

3.1 ซอฟต์แวร์ที่เป็นเครื่องมือสำหรับวัดประสิทธิภาพ (Benchmark Software Tools)

ในการกล่าวถึงประสิทธิภาพของการประมวลผลแบบขนานมักกล่าวถึงค่าที่ประสิทธิภาพสูงสุดทางทฤษฎี (Theoretical Peak Performance) หรือค่าประสิทธิภาพสูงสุดที่สามารถวัดได้จากการใช้โปรแกรมแบบขนานที่เป็นมาตรฐาน โครงการนี้ใช้ซอฟต์แวร์วัดประสิทธิภาพแบบมาตรฐาน เพื่อวัดสมรรถนะของระบบคลัสเตอร์โดยรวม โดยใช้โปรแกรมทดสอบมาตรฐาน เช่น High Performance LINPACK (HPL) [20] หรือ วัดความเร็วในการคำนวณเป็น FLOPS³ การใช้ library มาตรฐาน เช่น BLAS [17] LAPACK [18] และ SCALAPACK [19] หรือสามารถวัดความเร็วในการสื่อสารระหว่างเครื่องลูกข่ายในคลัสเตอร์ โดยใช้ไลบรารีของ MPI (Message Passing Interface) (9)

3.2 โปรแกรม MPI ที่พัฒนาขึ้นเอง

โปรแกรมที่พัฒนาขึ้นโดยผู้เขียน จะทำการคำนวณค่าแบบสุ่มและรับส่งข้อมูลพร้อมทั้งค้นหาค่าประสิทธิภาพที่ดีของระบบที่สามารถจะทำได้ด้วยการใช้ประยุกต์ใช้งานการหาค่าตัวแปรในการปรับที่ดีที่สุด (parameter optimization) จากการคำนวณด้วยทฤษฎีทางพันธุกรรมแบบขนาน หรือ Parallel Genetic Algorithms (2) โดยอัลกอริทึมที่นำเสนอ จะพัฒนาด้วยภาษาซีและไลบรารีของ MPI และใช้คอมไพเลอร์แบบ 64 บิต ซึ่งจะสามารถหาค่าการปรับแต่ง เช่น ตัวแปรต่างๆสำหรับคอมพิวเตอร์ ความยาวที่เหมาะสมที่สุดของข้อมูล (packet size) สำหรับคลัสเตอร์ โดยจะทำการวิเคราะห์และค้นหาค่าแบบอัตโนมัติ และนำผลที่ได้จากการ

วิเคราะห์ ไปทำการปรับแต่งการคอมไพล์โปรแกรมประยุกต์ เช่น GeoFEM, Molecular dynamics package (Amber), และ โปรแกรม Electronic structure calculation (Ab-init) เป็นต้น หลังจากที่ได้ทำการปรับแต่งและคอมไพล์โปรแกรมประยุกต์แล้ว จะทำการทดสอบประสิทธิภาพในระบบปิดและโดยผู้ใช้งานจริงต่อไป

4. สรุป

โครงการ 64 บิตลินุกซ์คลัสเตอร์ของ รร.จปร. ที่นำเสนออยู่ในระหว่างการจัดซื้ออุปกรณ์ และคาดว่าจะติดตั้งและเริ่มการทดลองได้หลังจากที่ NECTEC ได้ดำเนินการส่งมอบอุปกรณ์ซึ่งจะเป็นคลัสเตอร์แบบ 64 บิต ที่ใช้ซีพียู Itanium2 โดยจะเป็นคลัสเตอร์ที่มีจำนวน 8 โหนด เชื่อมต่อเข้ากับระบบเครือข่ายภายใน แต่จะไม่เปิดให้ใช้งานเป็นส่วนรวมจนกว่าการทดสอบในระบบปิดจะเสร็จสิ้น เพื่อป้องกันการรบกวนจากปัจจัยภายนอก หลังจากที่ได้ทำการวัดประสิทธิภาพและปรับแต่งค่าต่างๆ ในการคอมไพล์โปรแกรมประยุกต์แล้วจะทำการเปิดระบบ ซึ่งจะทำให้สามารถใช้งานร่วมกันระหว่างผู้ใช้ในกองวิชาวิศวกรรม วิทยาการคอมพิวเตอร์และผู้ใช้ในสาขาอื่น โดยจะสามารถรองรับความต้องการในการคำนวณเชิงวิศวกรรม หรือการจำลองแบบต่างๆ ด้วยระบบคอมพิวเตอร์ โดยเฉพาะอย่างยิ่ง การศึกษาในสาขาวิทยาการคอมพิวเตอร์ วิศวกรรมคอมพิวเตอร์ และเทคโนโลยีสารสนเทศของโรงเรียนนายร้อยพระจุลจอมเกล้า รายละเอียดเพิ่มเติมตลอดจนความคืบหน้าในการดำเนินโครงการสามารถดูได้ที่เว็บไซต์ของโครงการ <http://beowulf.crma.ac.th> 

³Floating Point Operations per Second เป็นค่ามาตรฐานในการเปรียบเทียบสมรรถนะในการคำนวณของซูเปอร์คอมพิวเตอร์

- [1] P. Chalermwat, T. El-Ghazawi, and J. LeMoigne, "2-Phase GA-based Image Registration on Parallel Clusters," *International Journal of Future Generation Computing Systems*, Volume 17 , Issue 4 January 2001.
- [2] P. Chalermwat, "High Performance Image Registration for Remote Sensing," *Doctoral Dissertation*, Fall 1999, Director: T. El-Ghazawi, Institute for Computational Science and Informatics, George Mason University.
- [3] P. Chalermwat, T. El-Chazawi and J. LeMoigne, "GA-based Parallel Image Registration on Parallel Clusters," *Lecture Notes in Computer Science*, Vol. 1586, p. 257-262, 1999.
- [4] P. Chalermwat and T. El-Ghazawi, "Multi-Resolution Image Registration Using Genetics," *IEEE International Conference on Image Processing (ICIP99)*, Japan, October 1999.
- [5] P. Chalermwat, T. El-Ghazawi, and J. LeMoigne, "2-Phase GA-based Image Registration on Parallel Clusters," *in proceedings of BioSP3 Workshop*, in conjunction with *IEEE International Parallel Processing Symposium (IPPS99)*, San Juan, Puerto Rico, April 1999.
- [6] T. El-Ghazawi, P. Chalermwat, P. Piamsa-nga, A. Ozkaya, N. Speciale, D. Wilson, "PACET: A PC-Parallel Architecture for Cost-Efficient Telemetry," *in proceedings of IEEE Aerospace Conference*, Snowmass at Aspen, Colorado, March 1998.
- [7] P. Chalermwat, T. El-Ghazawi, and J. LeMoigne, "Image Registration by Parts," *in proceedings of Image Registration Workshop (IRW97)*, NASA Goddard Space Flight Center, MD, November 1997.
- [8] T. El-Ghazawi, P. Chalermwat, and J. LeMoigne, "Parallel Image Registration on High Performance Computers," *in proceedings of Super Computing (SC97)*, San Jose, CA, November 1997.
- [9] Pacheco, Peter S., *Parallel Programming with MPI*, Morgan Kaufman Publishers, 2001, ISBN 1-55860-339-5.
- [10] <http://www.beowulf.org/overview/history.html>
- [11] <http://hpcnc.cpe.ku.ac.th/research/>
- [12] Mike Berry and Tarek El-Ghazawi. "Parallel Input/Output Characteristics of NASA Science Applications". *Proceedings of the International Parallel Processing Symposium (IPPS'96)*, IEEE Computer Society Press. Honolulu, April 1996.
- [13] URL http://www.hpcc.nectec.or.th/C1/c1_project.shtml
- [14] URL <http://www.hpcc.nectec.or.th/>
- [15] URL <http://www.thaigrid.net/projects/>
- [16] Ian Foster and Carl Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, 1998, ISBN 1-55860-475-8.
- [17] BLAS reference page, URL <http://www.netlib.org/blas/>
- [18] LAPACK reference page, URL <http://www.netlib.org/lapack/>
- [19] SCALAPACK reference page, URL <http://www.netlib.org/scalapack/>
- [20] HP LINPACK reference page, URL <http://www.netlib.org/benchmark/hpl/>

A Model for Parallelizing Evolutionary Algorithms

ABSTRACT

This paper presents a new parallelization model for evolutionary algorithms called the AMD model. It is an extension of a coarse-grain model and optimized for implementation on the web. Unique features of the model includes usage of mobile software agents for task distribution, usage of a migrant buffer to neutralize differences in speed of hosts, usage of a hierarchy of distributed objects to allow system scalability, and usage of “on-the-fly” control parameter optimization strategy. Test results on thirty numerical, combinatorial, and machine learning problems shows good speedup and solution quality on majority of the test problems.

1. Introduction

The World Wide Web has progressed from presenting the user with passive hypertext documents to executing remote applets. The next natural extension would involve running compute-intensive distributed applications. Evolutionary algorithms are

good candidates for running on the web because they are naturally parallelizable. Existing parallelization techniques such as the global, coarse-grain and fine-grain models are not readily suitable for implementation on the web because of the low bandwidth/high latency characteristic of the web.

Section 2 presents the new model. Section 3 describes a set of experiments that were performed on the model. The results are reported in section 4, and concluding remarks are found in section 5.

2. The AMD Model

Our AMD model allows a massive number of computers on the Internet to efficiently share the same evolutionary algorithm task, and therefore speed up the computation. It is an extension of the coarse-grain parallelization model [Tanese 89b] and performs task decomposition at the deme level. The architecture of the system is shown in Figure 1.

เอกสารอ้างอิง

- [1] P. Chalermwat, T. El-Ghazawi, and J. LeMoigne, "2-Phase GA-based Image Registration on Parallel Clusters," *International Journal of Future Generation Computing Systems*, Volume 17, Issue 4 January 2001.
- [2] P. Chalermwat, "High Performance Image Registration for Remote Sensing," *Doctoral Dissertation*, Fall 1999, Director: T. El-Ghazawi, Institute for Computational Science and Informatics, George Mason University.
- [3] P. Chalermwat, T. El-Ghazawi and J. LeMoigne, "GA-based Parallel Image Registration on Parallel Clusters," *Lecture Notes in Computer Science*, Vol. 1586, p. 257-262, 1999.
- [4] P. Chalermwat and T. El-Ghazawi, "Multi-Resolution Image Registration Using Genetics," *IEEE International Conference on Image Processing (ICIP99)*, Japan, October 1999.
- [5] P. Chalermwat, T. El-Ghazawi, and J. LeMoigne, "2-Phase GA-based Image Registration on Parallel Clusters," *in proceedings of BioSP3 Workshop*, in conjunction with *IEEE International Parallel Processing Symposium (IPPS99)*, San Juan, Puerto Rico, April 1999.
- [6] T. El-Ghazawi, P. Chalermwat, P. Piamsa-nga, A. Ozkaya, N. Speciale, D. Wilson, "PACET: A PC-Parallel Architecture for Cost-Efficient Telemetry," *in proceedings of IEEE Aerospace Conference*, Snowmass at Aspen, Colorado, March 1998.
- [7] P. Chalermwat, T. El-Ghazawi, and J. LeMoigne, "Image Registration by Parts," *in proceedings of Image Registration Workshop (IRW97)*, NASA Goddard Space Flight Center, MD, November 1997.
- [8] T. El-Ghazawi, P. Chalermwat, and J. LeMoigne, "Parallel Image Registration on High Performance Computers," *in proceedings of Super Computing (SC97)*, San Jose, CA, November 1997.
- [9] Pacheco, Peter S., *Parallel Programming with MPI*, Morgan Kaufman Publishers, 2001, ISBN 1-55860-339-5.
- [10] <http://www.beowulf.org/overview/history.html>
- [11] <http://hpcnc.cpe.ku.ac.th/research/>
- [12] Mike Berry and Tarek El-Ghazawi. "Parallel Input/Output Characteristics of NASA Science Applications". *Proceedings of the International Parallel Processing Symposium (IPPS'96)*, IEEE Computer Society Press. Honolulu, April 1996.
- [13] URL http://www.hpcc.nectec.or.th/C1/c1_project.shtml
- [14] URL <http://www.hpcc.nectec.or.th/>
- [15] URL <http://www.thaigrid.net/projects/>
- [16] Ian Foster and Carl Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, 1998, ISBN 1-55860-475-8.
- [17] BLAS reference page, URL <http://www.netlib.org/blas/>
- [18] LAPACK reference page, URL <http://www.netlib.org/lapack/>
- [19] SCALAPACK reference page, URL <http://www.netlib.org/scalapack/>
- [20] HP LINPACK reference page, URL <http://www.netlib.org/benchmark/hpl/>