172290

หัวข้อวิทยานิพนธ์	การคูณเมตริกซ์แบบขนานบนระบบพีซีคลัสเตอร์
นักศึกษา	นายไพโรจน์ สมุทรักษ์
รหัสประจำตัว	46063611
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์
พ.ศ.	2548
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร.จีรพร ศรีสวัสดิ์

บทคัดย่อ

การหาผลคูณของเมตริกซ์เป็นการประมวลผลขั้นพื้นฐานในการหาคำตอบทางวิทยาศาสตร์และวิศวกรรมศาสตร์หลายด้าน เช่น การหาระยะทางที่สั้นที่สุดโดยใช้เมตริกซ์ การวิเคราะห์ออกแบบโครงสร้างเหล็ก และงานทางด้านกราฟิค เป็นต้น ปกติในการหาผลคูณเมตริกซ์แต่ละครั้งต้องใช้ระยะเวลาในการคำนวณเป็นเวลานาน โดยเฉพาะสำหรับเมตริกซ์ขนาดใหญ่ๆ (nxn) เนื่องจากมีความซับซ้อนด้านเวลาเท่ากับ $O(n^3)$ งานวิจัยนี้จึงเสนอการหาผลคูณเมตริกซ์แบบขนานบนระบบพีซีคลัสเตอร์โดยใช้มาตรฐานภาษาเอ็มพีไอ (MPI Standard) เพื่อช่วยลดระยะเวลาในการคำนวณดังกล่าวให้น้อยลง การหาผลคูณเมตริกซ์แบบขนานที่เสนอมี 3 วิธีซึ่งต่างกันตามลักษณะของการจัดแบ่งข้อมูลเพื่อการประมวลผลแบบขนานและการติดต่อสื่อสารที่เหมาะสมรวมถึงนโยบายการเก็บข้อมูลที่ซ้ำซ้อนกันในแต่ละหน่วยประมวลผล คือ 1) การหาผลคูณเมตริกซ์แบบขนานด้วยวิธี rbmm_w (Row-Block Partitioning Matrix Multiplication with replicated data) 2) การหาผลคูณเมตริกซ์แบบขนานด้วยวิธี rbmm_wo (Row-Block Partitioning Matrix Multiplication without replicated data) และ 3) การหาผลคูณเมตริกซ์แบบขนานด้วยวิธี cbmm_w (Checkerboard-Block Partitioning Matrix Multiplication with replicated data) โดยวิธีแรกเป็นวิธีที่มีผู้เสนอไว้แล้วบนระบบพีซีคลัสเตอร์ ส่วนสองวิธีหลังเป็นวิธีที่เสนอขึ้นในวิทยานิพนธ์ วิธีสุดท้ายเป็นการนำข้อดีของสองวิธีแรกมารวมกัน การวัดประสิทธิภาพของวิธีการคูณเมตริกซ์แบบขนานวิธีต่างๆ ที่เสนอรวมถึงการคูณเมตริกซ์แบบอนุกรม (Sequential Matrix Multiplication) ทำโดยการนำค่าเวลา ที่ใช้ในการประมวลผล (Response Time) อัตราการเพิ่มขึ้นของความเร็ว (Speedup) และ ประสิทธิภาพ (Efficiency) ของทุกวิธีมาเปรียบเทียบกัน จากผลการทดลองบนระบบพีซีคลัสเตอร์ที่มีขนาด 1 ถึง 9 หน่วยประมวลผล พบว่าวิธีใหม่ที่เสนอการหาผลคูณเมตริกซ์แบบขนานด้วยวิธี cbmm_w มีค่าอัตราการเพิ่มขึ้นของความเร็วเข้าใกล้กรณีอุดมคติมาก คือ 3.63 ในกรณีที่จำนวนหน่วยประมวลผลเท่ากับ 4 หน่วยและมีประสิทธิภาพดีกว่า 12% และ 37% เมื่อเปรียบเทียบกับวิธีการหาผลคูณเมตริกซ์แบบขนานด้วยวิธี rbmm_w และวิธี rbmm_wo ตามลำดับ

172290

Thesis	Parallel Matrix Multiplication on a Cluster of Pcs
Student	Mr. Phairoj Samutrak
Student ID	46063611
Degree	Master of Science
Programme	Computer Science
Year	2005
Thesis Advisor	Asst. Prof. Dr. Jeeraporn Srisawat

ABSTRACT

The matrix multiplication is one of the most common operation and solution of science and engineering applications i.e. an approach of all-pair shortest path with the modified matrix multiplication, steel-structure design, graphic design and computing, etc. Time complexity of the sequential matrix multiplication is $O(n^3)$ for any matrix size nxn. However when the applications need very large matrix size, the sequential matrix multiplication cannot finish in reasonable time and hence parallel matrix multiplication is introduced to reduce the sequential computing time. This study proposes and implements the practical coarse-grained parallel matrix multiplication on the cluster of PCs using MPI (Message Passing Interface) standard. In particular, three parallel matrix multiplication methods are presented, according to data partitioning schemes for decomposing matrix A and matrix B, data communication and replicated data policy namely: 1) rbmm_w (Row-Block Partitioning Matrix Multiplication with replicated data), 2) rbmm_wo (Row-Block Partitioning Matrix Multiplication without replicated data) and 3) cbmm_w (Checkerboard-Block Partitioning Matrix Multiplication with replicated data). The first method is the existing one on the cluster of PCs, while the last two methods are proposed in this thesis. Specially the last one is introduced by combining the advantages of the first two methods. Finally, the system performance of sequential and parallel processing of the matrix multiplication have been compared and evaluated in terms of response time, speedup and efficiency. On the cluster system of size up to 9 processors, the experimental results showed that the introduced cbmm_w strategy yielded nearly ideal speedup, which was 3.63 for P=4. In addition, the cbmm_w method performed the improved performance upto 12% over that of the existing rbmm_w method and 37% over that of the rbmm_wo method.