

ABSTRACT

The high performance computing systems or supercomputers nowadays are highly demanded especially for applications that requires high computation power in data processing such as problems involving quantum mechanical physics, financial analysis, weather forecasting, climate research, molecular modeling (computing the structures and properties of chemical compounds, biological macromolecules, polymers, and crystals), physical simulations (such as simulation of airplanes in wind tunnels and research into nuclear fusion), signal and image processing, cryptanalysis, and so on. But these kinds of machines is very costly and can't be available for every institute or firm, hence, this work is aiming to design and develop a low cost high performance computing system using computer cluster and applying the concepts of parallel computing. The intended cluster can be built using commodity of the shelf PCs and network hardware which will result in low cost supercomputer that can be implemented simply in any institute or university. In this research a deep study was done on parallel programming considerations and requirements as a foundation for applications development on the Beowulf cluster that was designed. Special emphasize was given to Message Passing Interface (MPI) which is a programming paradigm used widely on parallel computers, especially Scalable Parallel Computers with distributed memory, and on clusters of computers. During this study in order to verify the usage of parallel computing a numerical integration example was developed and the studied concepts were applied on it.

تجريد

إن الحوسبة عالية الأداء أو الحواسيب العملاقة Supercomputers صارت لا غنى عنها فى عصرنا هذا، خاصة للتطبيقات التى تحتاج معالجة سريعة ودقيقة لكم هائل من البيانات يقصُر عن أدائها الحاسوب الشخصى، مثل فيزياء ميكانيكا الكم، والتحليل المالى، وأبحاث المناخ، والنمذجة الجزيئية لدراسة خواص وبنية المركبات الكيميائية، والمحاكاة والتشبيه لإختبارات الطائرات فى أنفاق الهواء، ومحاكاة التفاعلات النووية، ومعالجة الإشارات والصور الرقمية وغيرها من التطبيقات التى تتطلب حواسيب ذات قدرات عالية لا توفرها إلا الحواسيب العملاقة. إلا إن مثل هذه الحواسيب تعتبر باهظة التكلفة ولا يمكن توفرها لكل الجامعات والهيئات خاصة فى بلادنا، لذا يهدف هذا البحث لدراسة وتصميم وتنفيذ بيئة حوسبة عالية الأداء بإستخدام عناقيد الحواسيب Computer Clusters وتطبيق مفاهيم المعالجة المتوازية. إن عنقود الحواسيب المستهدف بهذه الدراسة يمكن أن يُبنى بإستخدام حواسيب شخصية ومعدات شبكات كالتى تتوفر فى الأسواق المحلية مما يجعلها منخفضة الكلفة وسهلة التنفيذ وبمقدور أى جامعة أو هيئة أن تمتلكها، والجدير بالذكر أن عناقيد الحواسيب صارت مُهيمنة فى مجال الحواسيب العملاقة فى السنوات الأخيرة. فى هذا البحث دُرست بتعمق متطلبات البرمجة المتوازية Parallel Programming للتأسيس لتطوير البرامج والتطبيقات على عنقود الحواسيب المُصمم، وتم كذلك التركيز على Message Passing Interface (MPI) وهو نموذج وبيئة للبرمجة المتوازية يحدد مجموعة من المعايير والدوال والخصائص وتستخدم بكثرة فى الحوسبة المتوازية، ومن خلال هذه الدراسة وللتحقق من إستخدام الحوسبة المتوازية تم تطوير برنامج للتكامل العددي طُبقت عليه المفاهيم والمبادئ التى دُرست خلال البحث.

TABLE OF CONTENTS

ABSTRACT.....	I
تجريد.....	II
List of Tables.....	VII
List of Figures.....	VIII
1. Chapter 1: INTRODUCTION	
1.1. Background and Rationale.....	1
1.2. Supercomputers Common Uses and Applications.....	2
1.3. Supercomputer Architectures.....	2
1.3.1. Distributed Memory Multicomputer	2
1.3.2. Central Memory Symmetric Multiprocessors (SMPs) and Constellation of SMPs.....	3
1.3.3. Clusters	4
1.4. The Domination of Clustering.....	5
1.5. High Performance Computing (HPC) Software Models.....	6
1.5.1. Message Passing Interface MPI	6
1.5.2. OpenMP	6
1.5.3. Star-P.....	6
1.6. Problem Statement.....	6
1.7. Objectives.....	7
1.8. The Methodology.....	7
1.9. Thesis Outlines.....	7
2. Chapter 2: PARALLEL COMPUTER ARCHITECTURE	
2.1. Flynn's Taxonomy.....	9
2.2. Parallel Architectures.....	11
2.3. Memory Parallelism.....	11
2.4. Distributed Memory.....	12
2.5. Shared Memory.....	13
2.6. Parallel Random Access Memory.....	16

2.7. Interconnects.....	17
2.8. CPU Parallelism.....	19
2.9. Superscalar Processing.....	19
2.10. Explicitly Parallel Instructions.....	21
2.11. SIMD and Vectors.....	21
2.12. Multithreading.....	22
2.13. Beowulf Cluster Design.....	23
2.13.1. The Node.....	24
2.13.2. Network Hardware.....	25
3. Chapter 3: PARALLEL COMPUTER PROGRAMMING CONSIDERATIONS	
3.1. Architectural Considerations.....	28
3.2. Shared Memory.....	28
3.3. Distributed Memory.....	29
3.4. Hybrid Systems.....	30
3.5. Memory Hierarchy.....	31
3.6. Decomposing Programs for Parallelism.....	33
3.6.1. Identification of Parallelism.....	33
3.6.2. Decomposition Strategy.....	34
3.6.3. Programming Models.....	35
3.6.4. Implementation Styles.....	36
3.6.5. Parallel Loop Programming.....	36
3.6.6. SPMD Programming.....	38
3.6.7. Recursive Task Programming.....	39
3.7. Message Passing Interface (MPI).....	39
3.7.1. The Goals of MPI.....	42
3.7.2. Platforms Targets.....	43
4. Chapter 4: CLUSTER INSTALLATION AND CONFIGURATION	
4.1. Cluster Installation and Configuration Using Oscar.....	45
4.2. Overview of System Installation Suite (SIS)	45
4.3. Supported Distributions.....	46
4.4. Minimum System Requirements.....	47

4.5. General Installation Notes.....	48
4.6. Networking Notes.....	48
4.6.1. SELinux Conflict.....	49
4.6.2. pfilter Notes.....	49
4.7. OSCAR Distribution.....	50
4.8. Disk Space and Directory Considerations.....	51
4.9. Preparing the OSCAR Repositories.....	51
4.10. Configure the Ethernet Adapter for the Cluster.....	52
4.11. Cluster Installation Procedure.....	54
4.11.1. Launching the OSCAR Installer.....	54
4.11.2. Selecting Packages to Install.....	56
4.11.3. Install OSCAR Server Packages.....	57
4.11.4. Build OSCAR Client Image.....	57
4.11.5. Define OSCAR Clients.....	59
4.11.6. Setup Networking.....	61
4.11.7. Collect Client Node MAC Addresses.....	61
4.11.8. Select Installation Mode.....	62
4.11.9. Setup Boot Environment.....	63
4.11.10. Client Installations.....	63
4.11.11. Complete the Cluster Setup.....	64
4.11.12. Test Cluster Setup.....	64
5. Chapter 5: CLUSTER PROGRAMMING	
5.1. Message Passing Interface (MPI)	66
5.2. Core Message Passing Interface	66
5.2.1. Transferring Data.....	71
5.2.2. Input and Output with MPI.....	72
5.3. Designing Parallel Programs.....	79
5.4. Problem Decomposition.....	80
5.4.1. Decomposition Strategies.....	83
5.4.2. Data Decomposition.....	83
5.4.3. Control Decomposition.....	83
5.5. Mapping Tasks to Processors.....	85
5.6. Communication Overhead.....	86
5.7. Load Balancing.....	88

6. Chapter 6: RESULTS AND DISCUSSION	
6.1. Virtualization.....	93
6.1.1. Types of Virtualization.....	93
6.2. A Simple Problem.....	94
6.2.1. Single-Processor Program.....	95
6.2.2. An MPI Solution.....	97
6.3. Results Analysis and Performance Metrics	100
7. Chapter 7: CONCLUSION AND FUTURE WORK	
7.1. Conclusion.....	102
7.2. Future Work	103
7.2.1. Physical Deployment.....	103
7.2.2. Performance Measurements.....	104
7.2.3. Implementation of Heterogeneous Clusters.....	104
REFERENCES.....	105

List of Tables

Table No.	Table Name	Page
1-1	Architecture share for June 2009	5
4-1	OSCAR supported distributions	47
4-2	OSCAR distribution files	50

List of Figures

Figure No.	Figure Name	Page
1-1	(1) A computer. (2) A fancier computer. (3) A multicomputer	3
1-2	A four processor SMP	4
2-1	SISD architecture	10
2-2	SIMD architecture	10
2-3	MIMD architecture	11
2-4	Schematic parallel computer organizations	12
2-5	Generic CPU diagram	20
2-6	Beowulf Cluster Design	23
3-1	Uniform-access shared-memory architecture	29
3-2	Distributed-memory architecture	30
3-3	Standard uniprocessor memory hierarchy	32
4-1	OSCAR Wizard	55
4-2	OSCAR package selection	57
4-3	The image building	59
4-4	The clients definition	60
4-5	Collect client MAC addresses	62
4-6	Cluster setup testing	65
5-1	Data flow for numerical integration	82
5-2	Ideal process overlap	84
5-3	Process overlap with idle time	84
6-1	Virtualization.	92
6-2	Area under x^2 from 2 to 5 with three rectangles	95
6-3	Single processor program flowchart	96
6-4	Single Processor Program	97
6-5	Parallel Program	100