DEDICATIONS

To my family…

To my friends…

To my teachers…

ACKNOWLEDGMENTS

I owe my deepest gratitude to Mr. Ahmaed Abd Alhaleem from Petrodar Company who helped me to solve some problems in the implementation of the system. I would like to thank my colleagues at Sudan University of Science and Technology who helped me to complete this thesis: Ms. Amna Margani, Ms. Sara Tawfig, Mr. Abd Elrahman Mohamed Alamin and Mr. Almothana Alabid. A special thanks to my family, without the help and support of which the completion of this work wouldn't have been possible.

ABSTRACT

Grid computing aims at exploiting the available resources in computer networks for executing distributed and complex problems. The grid computing environment is an open and dynamic environment, where a new node can join the network at any time while other nodes may leave the network, thus resource and job management in this environment can be complex tasks.

This research aims at using agent technology to optimize the control and exploitation of available resources in the grid computing environment and facilitate monitoring of jobs that are executing on it.

A primary grid computing environment has been developed using agent technology. Since multi-agent systems are suitable for solving problems on open and dynamic environments, it is expected that the integration of grid computing technology and multi-agent technology will lead to a better management of the available resources in the environment. The proposed environment is comprised of three parts: construction of the network, establishing monitoring nodes, and executing the distributed task.

The proposed environment was implemented in two phases; the first phase was implemented on a virtual network; and the second was the implementation on a real network comprised of four hosts, with similar specifications connected in a local area network. Each of the hosts is using the Linux operating system. The Proposed environment was tested by running a parallel matrix multiplication application, using the MPI technology. Different dimensions of these matrices were used and the program was divided into a different number of processes for three different cases, namely: (1) Implementing the program in one host; (2) Implementing the program in four hosts using the traditional techniques and (3) Implementing the program in four hosts using the proposed environment.

Results obtained indicate that better resource management has been achieved using the developed environment. Selection of nodes to execute a job is done automatically at execution time by checking the current status of the node. Better job management has been achieved also. If there is not enough resources to execute a job it is scheduled until resources are found, the client is notified of such an action.

It is recommended that multi-agent negotiation techniques be used to achieve better control over distributed resources and jobs. Development of a security model for new environment is also needed.

المستخلص

تهدف الحوسبة الشبكية إلى إستغلال الموارد المتاحة في شبكات الحاسوب لتنفيذ البرامج الموزعة و المعقدة. وبيئة الحوسبة الشبكية هي بيئة مفتوحة ومتغيرة باستمرار، حيث تنضم عقد جديدة إلى الشبكة من حين إلى آخر في حين تخرج عقد أخرى نسبة لتغير ظروفها. مما جعل من عملية ادارة المصادر المتاحة و تنفيذ البرامج في هذه البيئة أمراً غاية في التعقيد.

يهدف هذا البحث إلى استخدام تقانة الوكلاء لزيادة التحكم في ادارة واستغلال الموارد المتاحة و تسهيل متابعة تنفيذ البرامج في بيئة الحوسبة الشبكية.

تم في هذا البحث تطوير بيئة ابتدائية للحوسبة الشبكية باستخدام تقانة الوكلاء، و ذلك بالنظر إلى أن أنظمة الوكلاء المتعددين تناسب البيئات المفتوحة والمتغيرة. وتتكون البيئة المقترحة من ثلاثة أجزاء رئيسية هي انشاء الشبكة، و مراقبة العقد الموجودة بالشبكة، و تنفيذ المهام الموزعة بها.

تم تنفيذ البيئة المقترحة على مرحلتين: الاولى في شبكة افتراضية، اما الثانية فقد تم التنفيذ في شبكة حقيقية مكونة من أربعة عقد ذات مواصفات متشابهة موصلة في شبكة محلية و بها نظام التشغيل لينكس. تم اختبار البيئة بإستخدام برنامج ضرب مصفوفات بتقنية واجهة تمرير الرسائل (MPI). حيث تم استخدام أبعاد مختلفة لهذه المصفوفات. تم تقسيم البرنامج إلى عدد مختلف من العمليات لثلاث حالات مختلفة هي تنفيذ البرنامج في جهاز واحد، و أربعة اجهزة بإستخدام الطرق التقليدية، و أربعة اجهزة بإستخدام البيئة المقترحة.

تشير النتائج التي تم التوصل إليها إلى أنه أمكن تحقيق تحكم أفضل بالموارد المتاحة باستخدام البيئة التي تم تطويرها، حيث أن اختيار العقد لتنفيذ برنامج معين يتم أثناء تنفيذ البرنامج عن طريق فحص الحالة الراهنة للعقدة. كما أنه إذا حدث توقف لعقدة معينة فإنه يتم تغيير حالة هذه العقدة إلى الحالة "غيرمتصل. لوحظ ايضاً أنه إذا لم توجد موارد كافية لتنفيذ برنامج معين تتم جدولة هذا البرنامج للتنفيذ لاحقاً و يتم إعلام العميل بذلك.

لزيادة التحكم بالموراد و البرامج الموزعة نوصي باستخدام تقنية التفاوض المستخدمة في انظمة الوكلاء المتعددين. كما نوصي بتطوير نظام حماية للبيئة الجديدة.

LIST OF FIGURES

Figure 2-1 Structure of multi agent system. 7
Figure 2-2 Itinerary Pattern 8
Figure 2-3 Star-Shape Pattern 9
Figure 2-4 Branching Pattern 10
Figure 2-5 Message Transport Reference Model 14
Figure 2-6 Virtual Resource of Grid Computing 17
Figure 2-7 Jobs are migrated to less busy parts of the grid to balance loads 18
Figure 2-8 Administrators can adjust policies to better allocate resources 19
Figure 2-9 Grid Computing Architecture 20
Figure 2-10 Federated DBMS Architecture 22
Figure 2-11 Intragrids, extragrids, and intergrids 24
Figure 2-12 Agent framework with Grid protocol. 27
Figure 2-13 Framework for Grid Computing. 28
Figure 2-14 Relationship between the Modules and the Container. 29
Figure 2-15 Architecture of S.M.A.R.T-2 Grid Container. 29
Figure 2-16 Agent Interactions within Grid Container 30
Figure 4-1 Use Case of overall System 36
Figure 4-2 Use Case of Construction Grid Network 37
Figure 4-3 Execute Task Use Case 37
Figure 4-4 Monitoring Grid Network Use Case 38
Figure 4-5 System Class Diagram 40
Figure 4-6 Relationship between places, Agent and classes 41
Figure 4-7 Execute Task State Chart Diagram 42
Figure 4-8 Manage Task State Chart Diagram 43
Figure 4-9 Execute Task on Grid Network State Chart Diagram 44
Figure 4-10 Monitoring Node State Chart Diagram 44
Figure 4-11 Monitoring Grid Network State Chart Diagram 45
Figure 4-12 Monitoring Grid System Diagram 45
Figure 4-13 Neighborhoods Diagram 48
Figure 4-14 Construct Grid Network Interaction Diagram 48
Figure 4-15 Monitoring Grid Network Interaction Diagram 49
Figure 4-16 Execute Task Interaction Diagram 49
Figure 5-1 Grid System Architecture 52
Figure 5-2 Execution of Receive File Program 55
Figure 5-3 Execution of Receive Message Program 56
Figure 5-4 Execution of Monitoring Grid Network Program 57
Figure 5-5 Execution of Monitor Node Agent Program 57
Figure 5-6 Execution of Manage Task Agent program 58
Figure 5-7 Execution of Execute Task Agent program 58
Figure 6-1 Execution Time in seconds for different matrix 60

dimensions - 4 processes on one node

Figure 6-2 Execution Time in seconds for different matrix dimensions - 4 processes on 4 nodes 61

Figure 6-3 Execution Time in seconds for different matrix dimensions - 10 processes on one node 62

Figure 6-4 Execution Time in seconds for different matrix dimensions - 10 processes on 4 node 63

Figure 6-5 Execution Time in seconds for different matrix dimensions - 4 processes on one node using developed middleware 64

Figure 6-6 Execution Time in seconds for different matrix dimensions - 10 processes on 4 node using developed middleware 65

Figure 6-7 Comparison of Execution Time for all Cases 67

LIST OF TABLES

Table 2-1	Agent proprieties	6
Table 2-2	FIPA ACL Message Parameters	13
Table 2-3	Grid Middleware Systems	25
Table 5-1	Hardware properties of hosts	51
Table 5-2	Names and IP Address of hosts	51
Table 5-3	Software on hosts	52
Table 6-1	Execution Time in seconds for different matrix dimensions - 4 processes on one node	60
Table 6-2	Execution Time in seconds for different matrix dimensions - 4 processes on 4 nodes	61
Table 6-3	Execution Time in seconds for different matrix dimensions - 10 processes on one node	62
Table 6-4	Execution Time in seconds for different matrix dimensions - 10 processes on 4 nodes	63
Table 6-5	Execution Time in seconds for different matrix dimensions - 4 processes on 4 nodes using developed middleware	64
Table 6-6	Execution Time in seconds for different matrix dimensions - 10 processes on 4 nodes using developed middleware	65
Table 6-7	Execution Time in seconds for all cases	66

ABBREVIATIONS

#	Abbreviation	Meaning
1	ACL	Agent Communication Language
2	AUML	Agent Unified Modeling Language
3	DBMS	Data Base Management System
4	FIPA	Foundation of Intelligent Physical Agent
5	IP	Internet Protocol
6	MPI	Message Passing Interface
7	SSH	Secure Shell
8	VM	Virtual Machine

TABLE OF CONTENT

Dedications i

Acknowledgments ii

Abstract iii

المستخلص iv

List of Figures v

List of Tables vii

Abbreviations viii

Table of Content ix

Chapter One Introduction 1

1.1 Problem Statement 2

1.2 Objectives 3

1.3 Scope of research 3

Chapter Two 4

2.1 Multi Agent Systems 5

2.1.1 Agent Definition 5

2.1.2 Agent Properties 6

2.1.3 Multi Agents Definition 7

2.1.4 Mobile Agents 7

2.1.4.1 Mobile Agent Definition 7

2.1.4.2 Advantages of Mobile Agents 7

2.1.4.3 Disadvantages of Mobile Agents 8

2.1.4.4 Mobile Agent Design Pattern 8

2.1.4.5 Mobile Agent Security 11

2.1.4.5.1 Security Threats 11

2.1.4.5.2 Security Requirements 11

2.1.5 Foundation for Intelligent Physical Agents (FIPA) 12

2.1.5.1 FIPA ACL Message Structure 12

2.1.5.2 Agent Message Transport Reference Model 13

2.2 Grid Computing 15

2.2.1 Gird Computing Definition 15

2.2.2 What Grid Computing can do 15

2.2.2.1 Exploiting underutilized resources 15

2.2.2.2 Parallel CPU capacity 16

2.2.2.3 Applications 16

2.2.2.4 Virtual resources and virtual organizations for collaboration 16

2.2.2.5 Access to additional resources 17

2.2.2.6 Resource balancing 18

2.2.2.7 Reliability 18

2.2.2.8 Management 19

2.2.3 Grid Computing Architecture 20

2.2.3.1 Grid Fabric level 20

2.2.3.2 Core Grid middleware 20
2.2.3.3 User-level Grid middleware 21
2.2.3.4 Grid applications and portals 21
2.2.4 Grid Architecture Models 21
2.2.4.1 Computational grid 21
2.2.4.2 Data grid 22
2.2.5 Grid Topologies 22
2.2.5.1 Intragrids 23
2.2.5.2 Extragrids 23
2.2.5.3 Intergrids 23
2.2.6 Grid Computing Middleware 24
2.2.7 What the Grid Computing Cannot Do 25
2.3 Previous Studies 27
2.3.1 Agent-Based Framework with Grid Protocol 27
2.3.2 Agent Grid: Coordinating Multiple Agent 27
2.3.3 Agent-Based Peer-to-Peer Grid Architecture 28
2.3.3.1 Overall Architecture and Core Components 29
Chapter Three Tools 31
3.1 VMware Workstation 32
3.1.1 Benefits of Vmware 32
3.1.1.1 Workstation streamlines software development and testing 32
3.1.1.2 Workstation enhances productivity of IT professionals 32
1.1.3 Workstation facilitates computer-based training and software demos 32
3.2 Agent Unified Modelling Languages (AUML) 33
3.2.1 Analysis 33
3.2.2 Design 33
3.3 MYSQL Database 34
Chapter Four System Analysis and Design 35
4.1 Analysis 36
4.1.1 Define Task of the System 36
4.1.1.1 Grid Network Construction 36
4.1.1.2 Job Management 37
4.1.1.3 Network Monitoring 37
4.1.2 Define Places of Agents 38
4.1.2.1 Server 38
4.1.2.2 Nodes 38
4.1.2.3 Client 38
4.1.3 Identify Needed Agent 38
4.1.3.1 Monitor Node Agent 38
4.1.3.2 Monitor Grid Network Agent 39
4.1.3.3 Execute Task Agent 39
4.1.3.4 Manage Task Agent 39
4.1.3.5 Constrict grid network agent 39

4.1.4 Identify Classes 39
4.1.4.1 Agent class 39
4.1.4.2 ACLMessage class 39
4.1.4.3 Receive message class 39
4.1.4.4 Receive file class 40
4.1.4.5 Sqlcommand class 40
4.1.5 Model Relationships between Agents, Classes and Places 40
2.2 Design The System 42
4.2.1 Define Authorization Rights 42
4.2.2 Define State Charts for Agents, Classes and Places 42
4.2.2.1 Execute Task 42
4.2.2.2 Manage Task 43
4.2.2.3 Monitoring Node 44
4.2.2.4 Monitoring Grid 45
4.2.3 Live Cycle Model 46
4.2.3.1 Construct Grid Network Agent Live Cycle 46
4.2.3.2 Monitoring Node Agent Live Cycle 46
4.2.3.3 Monitoring Grid Network Agent Live Cycle 46
4.2.3.4 Execute Task Agent Live Cycle 46
4.2.3.5 Manage Task Agent Live Cycle 47
4.2.4 Define Subsystems and Neighbourhoods 47
4.2.4.1 Subsystems 47
4.2.4.1.1 Construct Grid Network 47
4.2.4.1.2 Monitoring Grid 47
4.2.4.1.3 Execute Tasks 47
4.2.4.2 Neighbourhoods: 48
4.2.5 Describe the Dynamic Behaviors of the System 48
4.2.5.1 Interaction Diagram for construct Grid Network 48
4.2.5.2 Interaction Diagram for Monitoring Grid Network 48
4.2.5.3 Interaction Diagram for Execute Task 49
Chapter Five Implementation 50
5.1 System Component 51
5.1.1 Server 52
5.1.1.1 Programs Run on the Server 52
5.1.2 Node 53
5.1.2.1 Programs Run on the Nodes 53
5.1.3 Client 53
5.1.3.1 Programs Run on the Client 53
5.2 Steps of Works 53
5.2.1 Step One: Prepare hosts 54
5.2.2 Step Two: Stetting Network and Firewall 54
5.2.3 Step Three: Create New User and Use SSH without Password 54
5.2.4 Step Four: Create Bata Base and table 54
5.2.5 Step Five: Run File Transfer server program 54

5.2.6 Step Six: Run Receive Message Program 55
5.2.7 Step Seven: Run Monitor Grid Network Agent 56
5.2.8 Step Eight: Run Monitor Node Agent 57
5.2.9 Step Nine Run Manage Task Agent 58
5.2.10 Step Ten: Run Execute Task Agent 58
Chapter Six Result & Discussion 59
Chapter seven Conclusion Recommendations 69
References 72
Appendices 74
Appendix I Network Configuration 75
Appendix II SSH Less Password 78
Appendix III Data Base 79
Appendix IV Life Cycle Model 81