UNIVERSITY OF ADELAIDE
10.11.99
LIBRARY

SCHEDULING IN METACOMPUTING SYSTEMS

By
Heath A. James, B.Sc.(Ma. & Comp. Sc.)(Hons)
July 1999

A THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ADELAIDE

Contents

Abstract ix

Declaration xi

Acknowledgements xiii

1 Introduction 1
- 1.1 Project Motivation 2
- 1.2 Why Metacomputing? 4
- 1.3 Focus and Contributions of this Thesis 5
- 1.4 Organisation of this Thesis 6

2 Cluster Computing 9
- 2.1 Resource Managers 10
  - 2.1.1 Batch Systems 12
  - 2.1.2 Extended Batch Systems 16
- 2.2 Middleware and Metacomputing Systems 19
  - 2.2.1 Middleware Tools 22
  - 2.2.2 Globus 26
  - 2.2.3 Harness 28
  - 2.2.4 Infospheres 30
  - 2.2.5 Legion 31
  - 2.2.6 DOCT 33
  - 2.2.7 WebFlow 34
- 2.3 DISCWorld 34
- 2.4 Discussion and Comparison of Metacomputing Systems 37
- 2.5 Conclusion 39

3 A Review of Scheduling 43
3.1 Scheduling Models . . . 46
3.1.1 Static Scheduling . . . 47
3.1.2 Dynamic Scheduling . . . 48
3.1.3 Hybrid Static-Dynamic Scheduling . . . 49
3.2 Independent Tasks . . . 50
3.3 Dependent Tasks . . . 51
3.3.1 List Schedules . . . 51
3.3.2 Clustering Algorithms . . . 53
3.4 System State Information . . . 53
3.5 Historical Review of Scheduling . . . 57
3.6 Discussion . . . 63
3.7 Conclusion . . . 64

4 Scheduling Independent Tasks 67
4.1 Introduction . . . 67
4.2 Scheduling Independent Jobs . . . 68
4.3 Scheduling Algorithms . . . 70
4.4 Algorithms Test Results . . . 75
4.5 Discussion and Conclusion . . . 81

5 Modeling Scheduling and Placement 85
5.1 Terminology . . . 86
5.2 Features and Constraints of Model . . . 86
5.2.1 Distributed Scheduling with Partial System State . . . 87
5.2.2 Model Inputs . . . 88
5.2.3 Characteristics of System Components . . . 91
5.2.4 Restricted Placement . . . 96
5.2.5 Heuristic Optimal Schedules . . . 97
5.2.6 Duplication of Services and Data . . . 98
5.2.7 Clustering of Services . . . 98
5.2.8 Non-Preemptive Scheduling . . . 99
5.3 A Distributed Job Placement Language . . . 99
5.4 Scheduling Model . . . 104
5.4.1 Cost-Minimisation Function . . . 105

5.4.2 Schedule Creation and Processor Assignment 109
5.5 Discussion 112
5.6 Conclusion 113

6 Implementing Scheduling in DISCWorld 115
6.1 Schedule Execution Models 116
6.2 DISCWorld Global Naming Strategy 118
6.3 DISCWorld Remote Access Mechanism 120
6.3.1 Operations on DRAMs 124
6.4 DRAM Futures 129
6.5 Execution in DISCWorld 130
6.5.1 Schedule Creation 132
6.5.2 Global Execution 133
6.5.3 Localised Execution 148
6.6 Summary 156

7 DISCWorld Performance Analysis 159
7.1 Example Services 160
7.2 A Detailed Example 165
7.3 Performance Considerations 170
7.4 Conclusion 172

8 Conclusions and Future Work 175
8.1 Conclusions 175
8.2 Future Work 178

A Distributed Geographic Information Systems 181
A.1 Introduction 181
A.2 Online Geospatial Data and Services 183
A.2.1 Standards and APIs 184
A.2.2 Implementation of a Standardised Geospatial Data Archive . . 185
A.3 Distributed Systems for Decision Support 187
A.4 Decision Support Applications requiring Distributed GIS 193
A.4.1 Land Management and Crop Yield Forecasting 193
A.4.2 Defence and C^{3}I 193

A.4.3 Emergency Services 194
A.5 Conclusions 195

B ERIC 201
B.1 Introduction 201
B.2 Repository Design 202
B.3 Imagery Operations 203
B.4 Eric Architecture 205
B.5 Performance Issues 208
B.6 Discussion and Conclusions 209

Bibliography 211

Abstract

The problem of scheduling in a distributed heterogeneous environment is that of deciding where to place programs, and when to start execution of the programs. The general problem of scheduling is investigated, with focus on jobs consisting of both independent and dependent programs. We consider program execution within the context of metacomputing environments, and the benefits of being able to make predictions on the performance of programs. Using the constraint of restricted placement of programs, we present a scheduling system that produces heuristically good execution schedules in the absence of complete global system state information. The scheduling system is reliant on a processor-independent global naming strategy and a single-assignment restriction for data.

Cluster computing is an abstraction that treats a collection of interconnected parallel or distributed computers as a single resource. This abstraction is commonly used to refer to the scope of resource managers, most often in the context of queueing systems. To express greater complexity in a cluster computing environment, and the programs that are run on the environment, the term *metacomputing* [74] is now being widely adopted. This may be defined as a collection of possibly heterogeneous computational nodes which have the appearance of a single, virtual computer for the purposes of resource management and remote execution.

We review current technologies for cluster computing and metacomputing, with focus on their resource management and scheduling capabilities. We critically review these technologies against our Distributed Information Systems Control World (DISCWorld).

We develop novel mechanisms that enable and support scheduling and program placement in the DISCWorld prototype. We also discuss a mechanism by which a high-level job's internal structure can be represented, and processing requests controlled. We formulate this using extended markup language (XML).

To enable processing requests, which consist of a directed acyclic graph of programs, we develop a mechanism for their composition and scheduling placement. This rich data pointer and a complimentary futures mechanism are implemented as part of the DISCWorld remote access mechanism (DRAM). They also form the basis for a model of wide-area computation independent of DISCWorld.

We have implemented geospatial imagery applications using both simple RMI and common gateway interface, and the novel mechanisms developed in this thesis. After analysing and measuring our system, we find significant performance and scalability benefits.