World-Wide Virtual Machine:

A Metacomputing Environment

Integrating World-Wide Web and High Performance

Computing and Communications Technologies

by

KIVANC DINCER

B.S., Middle East Technical University, Turkey, 1989
M.S., Iowa State University, 1991

DISSERTATION

Submitted in partial fulfillment of the requirements for the
degree of Doctor of Philosophy in Computer Science
in the Graduate School of Syracuse University

August 1997

Approved ______________________________
Professor Geoffrey C. Fox

Date ______________________________

© 1997 Kivanc Dincer
All Rights Reserved

Abstract

This thesis discusses the major issues in building a metacomputing environment based on World-Wide Web (WWW) and High Performance Computing and Communication (HPCC) technologies and describes the design and implementation of such an environment called World Wide Virtual Machine (WWVM). The presented work helps to carry much of the past decade's work in HPCC technologies to the larger WWW domain.

The WWVM exploits the open interfaces brought by Web servers. It extends the servers via Common Gateway Interface (CGI) extensions and uses PVM daemons and low-level protocols such as HTTP, TCP/IP, and UDP/IP in order to combine remote networked computers as a single machine. The WWVM can work in stand-alone, message-passing, and dataflow modes and provides the interoperability of many diverse software and hardware components. The stand-alone mode allows computations to be performed on a remote WWVM server or any other machine coordinated by a server. In the message-passing mode, the WWVM is capable of executing message-passing PVM and MPI programs, and High Performance Fortran (HPF) programs compiled by the Syracuse Fortran 90D/HPF compiler, as well as parallel programs using PCRC or Global Arrays runtime support libraries. In the dataflow mode, WWVM's coordination layer interprets a simple task flow (data-dependency) description language to deduct the dataflow patterns between different WWVM nodes.

The WWVM supplies an integrated, Web-based programming environment and gives pervasive access to remote WWVM facilities from any platform (Unix, PC, or Mac) using a standard Web browser. Client-side Web technologies such as HTML, JavaScript, plug-ins, and Java supply a platform-independent graphical user interface and visualization capabilities that

include analyzing data output from programs and performance information recorded in Pablo's SDDF format. The associated data wrapper libraries provide real-time, application-specific data

programs and client-side Java applets.

Table of Contents

LIST OF TABLES xii

LIST OF FIGURES xiii

ACKNOWLEDGEMENTS xvi

CHAPTER 1 – INTRODUCTION 18

1.1 HIGH PERFORMANCE COMPUTING AND COMMUNICATIONS (HPCC) INITIATIVE 19

1.2 WORLD-WIDE WEB TECHNOLOGY 20

1.3 PROBLEM STATEMENT 21

1.4 CONTRIBUTIONS OF THIS RESEARCH 23

1.5 RELATED WORK 25

1.5.1 Web-Based Metacomputing 28

1.6 ORGANIZATION OF THE THESIS 32

CHAPTER 2 – THE WORLD-WIDE VIRTUAL MACHINE 33

2.1 MOTIVATIONS 33

2.2 A TAXONOMY OF COMPUTING SYSTEMS 36

2.3 SYSTEM ARCHITECTURE 41

2.3.1 Web Interface Layer 41

2.3.2 Middle Layer 43

2.3.3 WWVM Virtual Machine Layer 46

2.3.3.1 Dataflow Model: Loosely Coupled Computational Resources 46

Visual Programming Tools 49

Implementation Of The Dataflow Computing Model 50

2.3.3.2 Message-Passing Model: Tightly Coupled Computational Resources...........51

The Rationale Behind Using PVM...51

PVM System Architecture 54

Configuration Of The Virtual Machine 56

Handling WWVM Configuration Changes...59

CHAPTER 3 – WEB-RELATED IMPLEMENTATION ISSUES.......................................61

3.1 USING WEB TECHNOLOGIES FOR GRAPHICAL USER INTERFACE DESIGN.........62

3.1.1 Using HTML Forms and Frames for GUI Development 62

3.1.2 Advantages of Using JavaScript...64

3.1.3 Advantages of Using Java..66

3.2 WEB-BASED PARALLEL PROGRAMMING ENVIRONMENTS..................................67

3.2.1 WPPE Architecture..69

3.2.2 Use of WPPEs in a Research and Education Setting 71

3.2.3 Completed Prototypes 73

3.2.3.1 HPF and Parallel C++ on the Web 74

3.2.3.2 The Web/HPF 74

3.2.3.3 The Virtual Programming Laboratory (VPL) 75

File Manager 75

Text Editor 77

HPF and MPI Programming Laboratories 78

3.2.4 Related Work..79

3.2.5 Comparison of Client Site vs. Server Site WPPEs: VPL vs. EPIC 80

3.3 PORTABILITY OF THE WPPE AND WWVM SOFTWARE 83

3.4 IMPLEMENTATION ISSUES...84

3.4.1 Providing User Accounts and Associated Server Configuration................................85

3.4.1.1 Real UNIX Accounts...86

3.4.1.2 Virtual Accounts..87

3.4.2 Using Server-Side or Client-Side Software...88

3.4.3 Running WWVM Web Servers on Parallel Platforms...91

3.5 SYSTEM SECURITY MEASURES...92

3.5.1 Protecting In-Transit Data...93

3.5.2 Protecting Data on Server..94

3.5.2.1 User Identification and authentication..94

3.5.2.2 Access Control 94

3.5.2.3 Auditing and Logging..95

3.5.2.4 Web (HTTP) Server Setup..95

3.5.2.5 Required Authorization Checks and Execution Rights..................................96

3.5.2.6 CGI Scripts 97

3.5.2.7 Filtering User Input for Metacharacters...98

3.5.2.8 Protection Against Other Users..98

3.5.2.9 Filtering Arbitrary User Programs...99

3.5.3 Client Protection...100

3.5.4 Implementation of User Identification and Access Control Mechanisms..................102

3.5.4.1 Basic Web Authentication Facility..102

3.5.4.2 CGI-Based Authentication and Access Control 104

Using HTML Forms and Hidden Fields 105

Using Persistent Cookies 105

3.5.4.3 Java-Based Authentication and Access Control 106

3.5.5 Future of Web Security 107

CHAPTER 4 – PORTING HPCC APPLICATIONS ONTO THE WORLD WIDE VIRTUAL MACHINE PLATFORM ……………………………………………………..109

4.1 PARALLEL PROGRAMMING MODELS 110

4.1.1 Library-Based Shared-Memory Implementations 111

4.1.2 Language-Based Shared-Memory Implementations 112

4.1.3 Evaluation of Message-Passing and Shared-Memory Models 114

4.2 MPI–TO–PVM INTERFACE LAYER 115

4.2.1 Message-Passing Interface (MPI) 115

4.2.1.1 Safe Message Passing in MPI 118

4.2.2 Implementation of the MPI-to-PVM Interface Layer 119

4.2.2.1 Contexts, Process Groups, and Communicators 122

Communication Contexts 122

Process Groups 123

Communicator Objects 124

4.2.2.2 Application Topologies 124

4.2.2.3 Point-to-Point Communication 126

Implementation of Point-to-Point Operations 128

Communication Completion 129

Send-and-Receive Operations 130

4.2.2.4 Collective Communication Operations 130

General and User-Defined Data Types 131

4.3 HIGH PERFORMANCE FORTRAN (HPF) AND PCRC RUNTIME SUPPORT LIBRARIES 132

4.3.1 Fortran 90D/HPF Intrinsic Functions 136

4.3.2 Collective Communication Functions 138

4.3.3 Implementing Express Routines in Terms of MPI Routines 138

4.4 GLOBAL ARRAYS PROGRAMMING MODEL 139

4.4.1 A Brief Comparison of HPF and GA Libraries 142

4.4.2 Porting Global Arrays onto the WWVM platform 142

4.4.2.1 Point-to-Point Operations 144

4.4.2.2 Collective Communication Operations 145

4.4.2.3 GA Data Server 146

4.4.2.4 One-Sided Communication 147

4.4.2.5 Emulating the TCGMSG `NXTVAL()` Routine 147

4.5 SUMMARY: THE BIG PICTURE 148

CHAPTER 5 – WWVM DATA AND PERFORMANCE VISUALIZATION SYSTEMS 150

5.1 PERFORMANCE VISUALIZATION SYSTEM 152

5.1.1 Postmortem Analysis 153

5.1.2 Displays 155

5.1.2.1 Common Displays 156

Gantt Chart..156

Event Count Display...156

Animation...158

Concurrency Profile 158

Summary Display 158

Trace Display...159

Clock Display 159

Statistical Summary 159

5.1.2.2 Specific Displays 159

Kiviat Diagram 159

Communication Spacetime Diagram 160

Communication Matrix 160

5.1.2.3 Parameters...160

5.1.3 Interaction with Pablo 161

5.1.3.1 Pablo Self-Describing Trace Data Format 163

5.2 DATA VISUALIZATION SYSTEM 165

5.2.1 Customize Plot Menu...167

5.2.1.1 Data Format 167

5.2.1.2 Plot Styles 168

5.2.2 Customizing Graph Menu 168

5.3 DATA WRAPPERS FOR VISUALIZING PROGRAM DATA STRUCTURES 169

5.3.1 Implementation...169

5.3.2 Operation of Data Wrappers..172

5.3.3 Use of Data Wrappers for HPF Data and Performance Visualization 173

5.3.3.1 Related Work 174

CHAPTER 6 – CONCLUDING REMARKS AND DIRECTIONS FOR FUTURE RESEARCH 176

6.1 SUMMARY AND IMPLICATIONS OF THESIS 176

6.2 DIRECTIONS FOR FUTURE RESEARCH 182

REFERENCES 187

VITAE 206

List of Tables

TABLE 3–1 POTENTIAL WPPES AND THEIR ASSOCIATED CONSTRAINTS …………………….. 73

TABLE 3–2 APPROPRIATE SERVER ACCOUNT AND USER ID CHOICES FOR VARIOUS WPPES.…. 86

TABLE 4–1 A GENERAL COMPARISON OF FUNCTIONALITY PROVIDED BY PVM AND MPI.…... 120

TABLE 4–2 SEMANTICS OF PVM AND MPI POINT-TO-POINT PRIMITIVES ...………………….….. 127

TABLE 4–3 PVM MAPPING OF POINT-TO-POINT AND ENVIRONMENTAL MANAGEMENT PRIMITIVES OF MPI ………………………………………………………………………. 129

TABLE 4–4 COLLECTIVE COMMUNICATION ROUTINES OF PVM AND MPI ...………………… 130

TABLE 4–5 PVM MAPPING OF MPI COLLECTIVE COMMUNICATION ROUTINES ……………… 131

TABLE 4–6 EXPRESS FUNCTIONS USED IN THE IMPLEMENTATION OF FORTRAN 90D/HPF COMPILER INTRINSIC ROUTINES …………………………………………………………..…. 135

TABLE 4–7 EXPRESS FUNCTIONS USED IN THE IMPLEMENTATION OF FORTRAN 90D/HPF COMPILER COLLECTIVE COMMUNICATION ROUTINES …………………………………………. 136

TABLE 4–8 ONE-TO-ONE MAPPING OF EXPRESS AND MPI POINT-TO-POINT COMMUNICATION AND ENVIRONMENT MANAGEMENT ROUTINES …..………….…………… 137

TABLE 4–9 ONE-TO-ONE MAPPING OF EXPRESS AND MPI COLLECTIVE COMMUNICATION ROUTINES…………………………………………………………………..……………… 139

TABLE 4–10 TWO CODE SAMPLES WITH SIMILAR FUNCTIONALITY WRITTEN USING GA LIBRARY CALLS AND HPF ……………………………………….………..………………. 141

TABLE 4–11 ONE-TO-ONE MAPPING OF TCGMSG POINT-TO-POINT FUNCTIONS TO MPI FUNCTIONS ………………………………………………………………………….…….. 143

TABLE 4–12 ONE-TO-ONE MAPPING OF TCGMSG COLLECTIVE OPERATION FUNCTIONS TO MPI FUNCTIONS ……………………………………………………………………….…. 144

TABLE 4–13 ONE-TO-ONE MAPPING OF REMAINING TCGMSG FUNCTIONS TO MPI FUNCTIONS…………………………………………………………..…………….…. 145

List of Figures

FIGURE 2–1 OVERVIEW OF THE CLIENT-SERVER RELATIONSHIP IN THE WWVM ARCHITECTURE………………………………………………………………… 34

FIGURE 2–2 A TAXONOMY OF THE COMPUTING SYSTEMS ACCORDING TO EXECUTION MODE AND MACHINE MODEL.……………………………………………….…. 36

FIGURE 2–3 A COMPARISON OF THE (A) MPP AND (B) CLUSTER OF WORKSTATIONS ARCHITECTURES………………………………………………………...…… 38

FIGURE 2–4 SYSTEM ARCHITECTURE OF THE VIRTUAL MACHINE………………...….…. 40

FIGURE 2–5 A SNAPSHOT OF THE WEB FRONT-END FOR THE WWVM CONFIGURATION SERVICES……………………………………………………………………. 42

FIGURE 2–6 DATAFLOW ARCHITECTURE OF THE WWVM……………………………… 44

FIGURE 2–7 DESCRIPTION OF DATAFLOW TASKS…...…………………………………… 47

FIGURE 2–8 DEFINITION OF A PROBLEM AND ITS TASK FLOW GRAPH..…………….……. 48

FIGURE 2–9 LOGICAL VIEW OF A TYPICAL MESSAGE-PASSING APPLICATION ON THE WWVM………………………………………………………………... 55

FIGURE 2–10 STEPS FOR ADDING NEW HOSTS ACROSS THE NETWORK TO THE VIRTUAL MACHINE………………..…………………………………………………. 58

FIGURE 3–1 ILLUSTRATION OF THE WPPE ARCHITECTURE …………...………………… 68

FIGURE 3–2 SNAPSHOTS OF VPL V2.0'S WEB-BASED USER INTERFACE. ON THE RIGHT, HPF COMPILATION SCREEN AND HPF-EXTENDED TEXT EDITOR IS SHOWN………………………………………………………………………. 76

FIGURE 3–3 VPL USER INTERFACE ITEMS……………………………………………….. 77

FIGURE 3–4 VPL SERVER-SIDE COMPILATION ITEMS………………………………………. 78

FIGURE 3–5 RELATIONSHIP BETWEEN INTENDED USAGE AND CONFIGURATION OF SERVER ACCOUNT AND USER IDS……………………………………………. 84

FIGURE 4–1 MESSAGE-PASSING AND SHARED-MEMORY PROGRAMMING MODELS IMPLEMENTED ON TOP OF VARIOUS TYPES OF PARALLEL ARCHITECTURES, AND THEIR PROMINENT FEATURES…………………………………………...110

FIGURE 4–2 ILLUSTRATION OF THE INTENDED BEHAVIOR OF A MESSAGE-PASSING CODE WITH EMBEDDED LIBRARY CALLS …………………………………………... 116

FIGURE 4–3 ILLUSTRATION OF A POSSIBLE ANOMALY (I.E., DEADLOCK SITUATION) IN A MESSAGE-PASSING CODE WHEN CONTEXTS ARE NOT USED IN MESSAGE-PASSING CALLS ………………………………………………………………… 116

FIGURE 4–4 AN MPI COMMUNICATOR CONSISTS OF A GROUP, CONTEXT AND SOME OTHER INTERNAL INFORMATION. GROUP IDS ARE MATCHED TO PVM TASK IDS ……………………………………………………………………... 121

FIGURE 4–5 THE LIFE CYCLE OF A HPF PROGRAM FROM COMPILATION TO EXECUTION ON THE WWVM ……………………………………………………………….. 133

FIGURE 4–6 A LOCAL ARRAY SECTION AND ITS CORRESPONDING DAD IN FORTRAN 90D/HPF RTS LIBRARIES. THIS IS USED TO PASS ARRAY SPECIFICATIONS TO THE INTRINSIC ROUTINES………………………………………………… 134

FIGURE 4–7 ASYNCHRONOUS ONE-SIDED GET() AND PUT() ACCESSES TO ARBITRARY SECTIONS OF A DISTRIBUTED GA MATRIX…………………………………... 140

FIGURE 4–8 ORDER OF OPERATIONS DURING THE INTERPRETATION OF A REMOTE GET() OPERATION ON THE WWVM ARCHITECTURE………………………………. 146

FIGURE 4–9 SHARED-MEMORY MODELS AND MESSAGE-PASSING LIBRARIES SUPPORTED BY WWVM…………………………………………………….……………. 149

FIGURE 5–1 PREPARING VISUALIZATION TRACES FOR AN MPI-BASED MESSAGE-PASSING PROGRAM WRITTEN IN C…………………………………………………….. 151

FIGURE 5–2 A SNAPSHOT OF A SAMPLE PERFORMANCE VISUALIZATION SESSION….……157

FIGURE 5–3 A SAMPLE SDDF FILE IN ASCII FORMAT…………………………………….….. 165

FIGURE 5–4 A SNAPSHOT OF A SAMPLE 2-D DATA VISUALIZATION SESSION…………… 166

FIGURE 5–5 OPERATION OF DATA WRAPPERS FOR VISUALIZATION AND INTERACTIVE COMPUTATION STEERING……………………………………………………. 170

Acknowledgements

I appreciate my advisor, Professor Geoffrey C. Fox, for his guidance throughout my research work and for his wise and acute observations on how to improve my work. I gratefully acknowledge Dr. Anne Trefethen from Cornell Theory Center, Dr. Simon Catterall, Dr. Wojtek Furmanski, Dr. Xiaoming Li, and Dr. Stephen Taylor for serving on my defense committee.

I would like to extend my deepest thanks to Dr. Alok Choudhary, Dr. Tomasz Haupt, Dr. Sanjay Ranka, Dr. David Bernholdt, and Mr. Donald Leskiw with whom I worked and learned from during the last four years.

Many thanks are also due Dr. Salim Hariri, Dr. Xiaoming Li's students from China, and a number of anonymous reviewers who read earlier technical paper versions of this manuscript and provided useful comments. Their excellent suggestions greatly helped me in preparing the final revision.

My sincere thanks go to Ms. Kathy Barbieri and Ms. Caroline Hecht from Cornell Theory Center for providing valuable suggestions for my work.

I am especially grateful to Mrs. Elaine Weinman for her help in English and for her patience in proofreading endless revisions of this manuscript.

I would like to express my appreciation to many friends who contributed to my work with their encouragement. Among them, Mr. Haluk Topcuoglu deserves special thanks. His moral support and suggestions were of great value.

I would also like to thank the Turkish Scientific and Research Council for providing support for part of my graduate studies through a North Atlantic Treaty Organization (NATO) Ph.D. fellowship.

In particular, I would like to thank my wife, my parents, and my wife’s parents for their constant support and encouragement throughout my graduate career. They showed tolerance beyond the bounds of reasonable expectations. I could not have completed my degree without their support.