

Contents

Series Foreword	xv
Preface	xvii
1 Introduction	1
1.1 Background	1
1.1.1 Ancient History	1
1.1.2 The MPI Forum	2
1.1.3 The MPI-2 Forum	3
1.2 What's New in MPI-2?	4
1.2.1 Parallel I/O	5
1.2.2 Remote Memory Operations	6
1.2.3 Dynamic Process Management	7
1.2.4 Odds and Ends	7
1.3 Reading This Book	9
2 Getting Started with MPI-2	11
2.1 Portable Process Startup	11
2.2 Parallel I/O	12
2.2.1 Non-Parallel I/O from an MPI Program	13
2.2.2 Non-MPI Parallel I/O from an MPI Program	15
2.2.3 MPI I/O to Separate Files	16
2.2.4 Parallel MPI I/O to a Single File	19
2.2.5 Fortran 90 Version	21
2.2.6 Reading the File with a Different Number of Processes	22
2.2.7 C++ Version	24
2.2.8 Other Ways to Write to a Shared File	28
2.3 Remote Memory Access	29
2.3.1 The Basic Idea: Memory Windows	30
2.3.2 RMA Version of <code>cp</code>	30
2.4 Dynamic Process Management	36
2.4.1 Spawning Processes	37
2.4.2 Parallel <code>cp</code> : A Simple System Utility	38
2.5 More Info on <code>Info</code>	47

2.5.1	Motivation, Description, and Rationale	47
2.5.2	An Example from Parallel I/O	47
2.5.3	An Example from Dynamic Process Management	48
2.6	Summary	50
3	Parallel I/O	51
3.1	Introduction	51
3.2	Using MPI for Simple I/O	51
3.2.1	Using Individual File Pointers	52
3.2.2	Using Explicit Offsets	55
3.2.3	Writing to a File	59
3.3	Noncontiguous Accesses and Collective I/O	59
3.3.1	Noncontiguous Accesses	60
3.3.2	Collective I/O	64
3.4	Accessing Arrays Stored in Files	67
3.4.1	Distributed Arrays	68
3.4.2	A Word of Warning about Darray	71
3.4.3	Subarray Datatype Constructor	72
3.4.4	Local Array with Ghost Area	74
3.4.5	Irregularly Distributed Arrays	78
3.5	Nonblocking I/O and Split Collective I/O	81
3.6	Shared File Pointers	83
3.7	Passing Hints to the Implementation	85
3.8	Consistency Semantics	89
3.8.1	Simple Cases	89
3.8.2	Accessing a Common File Opened with MPI_COMM_WORLD	91
3.8.3	Accessing a Common File Opened with MPI_COMM_SELF	94
3.8.4	General Recommendation	95
3.9	File Interoperability	95
3.9.1	File Structure	96
3.9.2	File Data Representation	97
3.9.3	Use of Datatypes for Portability	98

3.9.4	User-Defined Data Representations	100
3.10	Achieving High I/O Performance with MPI	101
3.10.1	The Four “Levels” of Access	101
3.10.2	Performance Results	105
3.10.3	Upshot Graphs	106
3.11	An Astrophysics Example	112
3.11.1	ASTRO3D I/O Requirements	112
3.11.2	Implementing the I/O with MPI	114
3.11.3	Header Issues	116
3.12	Summary	118
4	Understanding Synchronization	119
4.1	Introduction	119
4.2	Synchronization in Message Passing	119
4.3	Comparison with Shared Memory	127
4.3.1	Volatile Variables	129
4.3.2	Write Ordering	130
4.3.3	Comments	131
5	Introduction to Remote Memory Operations	133
5.1	Introduction	135
5.2	Contrast with Message Passing	136
5.3	Memory Windows	139
5.3.1	Hints on Choosing Window Parameters	141
5.3.2	Relationship to Other Approaches	142
5.4	Moving Data	142
5.4.1	Reasons for Using Displacement Units	146
5.4.2	Cautions in Using Displacement Units	147
5.4.3	Displacement Sizes in Fortran	148
5.5	Completing Data Transfers	148
5.6	Examples of RMA Operations	150
5.6.1	Mesh Ghost Cell Communication	150

5.6.2	Combining Communication and Computation	164
5.7	Pitfalls in Accessing Memory	169
5.7.1	Atomicity of Memory Operations	169
5.7.2	Memory Coherency	171
5.7.3	Some Simple Rules for RMA	171
5.7.4	Overlapping Windows	173
5.7.5	Compiler Optimizations	173
5.8	Performance Tuning for RMA Operations	175
5.8.1	Options for <code>MPI_Win_create</code>	175
5.8.2	Options for <code>MPI_Win_fence</code>	177
6	Advanced Remote Memory Access	181
6.1	Introduction	181
6.2	Lock and Unlock	181
6.2.1	Implementing Blocking, Independent RMA Operations	183
6.3	Allocating Memory for MPI Windows	184
6.3.1	Using <code>MPI_Alloc_mem</code> from C/C++	184
6.3.2	Using <code>MPI_Alloc_mem</code> from Fortran	185
6.4	Global Arrays	185
6.4.1	Create and Free	188
6.4.2	Put and Get	192
6.4.3	Accumulate	194
6.5	Another Version of <code>NXTVAL</code>	194
6.5.1	The Nonblocking Lock	197
6.5.2	A Nonscalable Implementation of <code>NXTVAL</code>	197
6.5.3	Window Attributes	201
6.5.4	A Scalable Implementation of <code>NXTVAL</code>	204
6.6	An RMA Mutex	208
6.7	The Rest of Global Arrays	210
6.7.1	Read and Increment	210
6.7.2	Mutual Exclusion for Global Arrays	210
6.7.3	Comments on the MPI Version of Global Arrays	212

6.8	Differences between RMA and Shared Memory	212
6.9	Managing a Distributed Data Structure	215
6.9.1	A Shared-Memory Distributed List Implementation	215
6.9.2	An MPI Implementation of a Distributed List	216
6.9.3	Handling Dynamically Changing Distributed Data Structures	220
6.9.4	An MPI Implementation of a Dynamic Distributed List	224
6.10	Compiler Optimization and Passive Targets	225
6.11	Scalable Synchronization	228
6.11.1	Exposure Epochs	229
6.11.2	The Ghost-Point Exchange Revisited	229
6.11.3	Performance Optimizations for Scalable Synchronization	231
6.12	Summary	232
7	Dynamic Process Management	233
7.1	Introduction	233
7.2	Creating New MPI Processes	233
7.2.1	Intercommunicators	234
7.2.2	Matrix-Vector Multiplication Example	235
7.2.3	Intercommunicator Collective Operations	238
7.2.4	Intercommunicator Point-to-Point Communication	239
7.2.5	Finding the Number of Available Processes	242
7.2.6	Passing Command-Line Arguments to Spawned Programs	245
7.3	Connecting MPI Processes	245
7.3.1	Visualizing the Computation in an MPI Program	247
7.3.2	Accepting Connections from Other Programs	249
7.3.3	Comparison with Sockets	251
7.3.4	Moving Data between Groups of Processes	253
7.3.5	Name Publishing	254
7.4	Design of the MPI Dynamic Process Routines	258
7.4.1	Goals for MPI Dynamic Process Management	258

7.4.2	What MPI Did Not Standardize	260
8	Using MPI with Threads	261
8.1	Thread Basics and Issues	261
8.1.1	Thread Safety	262
8.1.2	Threads and Processes	263
8.2	MPI and Threads	263
8.3	Yet Another Version of <code>NXTVAL</code>	266
8.4	Implementing Nonblocking Collective Operations	268
8.5	Mixed-Model Programming: MPI for SMP Clusters	269
9	Advanced Features	273
9.1	Defining New File Data Representations	273
9.2	External Interface Functions	275
9.2.1	Decoding Datatypes	277
9.2.2	Generalized Requests	279
9.2.3	Adding New Error Codes and Classes	285
9.3	Mixed-Language Programming	289
9.4	Attribute Caching	292
9.5	Error Handling	295
9.5.1	Error Handlers	295
9.5.2	Error Codes and Classes	297
9.6	Topics Not Covered in This Book	298
10	Conclusions	301
10.1	New Classes of Parallel Programs	301
10.2	MPI-2 Implementation Status	301
10.2.1	Vendor Implementations	301
10.2.2	Free, Portable Implementations	302
10.2.3	Layering	302
10.3	Where Does MPI Go from Here?	302
10.3.1	More Remote Memory Operations	303

10.3.2	More on Threads	303
10.3.3	More Language Bindings	304
10.3.4	Interoperability of MPI Implementations	304
10.3.5	Real-Time MPI	304
10.4	Final Words	304
A	Summary of MPI-2 Routines and Their Arguments	307
B	MPI Resources on the World Wide Web	355
C	Surprises, Questions, and Problems in MPI	357
D	Standardizing External Startup with <code>mpiexec</code>	361
	References	365
	Subject Index	373
	Function and Term Index	379