# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Series Foreword</td>
<td>xiii</td>
</tr>
<tr>
<td>Preface</td>
<td>xv</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Message Passing and Standardization</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Who Should Use This Standard?</td>
<td>4</td>
</tr>
<tr>
<td>1.3 What Platforms Are Targets for Implementation?</td>
<td>4</td>
</tr>
<tr>
<td>1.4 What Is Included in MPI?</td>
<td>5</td>
</tr>
<tr>
<td>1.5 Version of MPI</td>
<td>6</td>
</tr>
<tr>
<td>1.6 MPI Conventions and Design Choices</td>
<td>6</td>
</tr>
<tr>
<td>1.6.1 Document Notation</td>
<td>6</td>
</tr>
<tr>
<td>1.6.2 Naming Conventions</td>
<td>7</td>
</tr>
<tr>
<td>1.6.3 Procedure Specification</td>
<td>8</td>
</tr>
<tr>
<td>1.7 Semantic Terms</td>
<td>9</td>
</tr>
<tr>
<td>1.8 Function Argument Data Types</td>
<td>11</td>
</tr>
<tr>
<td>1.8.1 Opaque Objects</td>
<td>11</td>
</tr>
<tr>
<td>1.8.2 Array Arguments</td>
<td>13</td>
</tr>
<tr>
<td>1.8.3 State</td>
<td>13</td>
</tr>
<tr>
<td>1.8.4 Named Constants</td>
<td>13</td>
</tr>
<tr>
<td>1.8.5 Choice</td>
<td>14</td>
</tr>
<tr>
<td>1.8.6 Addresses</td>
<td>14</td>
</tr>
<tr>
<td>1.8.7 File Offsets</td>
<td>14</td>
</tr>
<tr>
<td>1.9 Language Binding</td>
<td>14</td>
</tr>
<tr>
<td>1.9.1 Deprecated Names and Functions</td>
<td>15</td>
</tr>
<tr>
<td>1.9.2 Fortran Binding Issues</td>
<td>15</td>
</tr>
<tr>
<td>1.9.3 C Binding Issues</td>
<td>18</td>
</tr>
<tr>
<td>1.9.4 C++ Binding Issues</td>
<td>19</td>
</tr>
<tr>
<td>1.10 Processes</td>
<td>23</td>
</tr>
<tr>
<td>1.11 Error Handling</td>
<td>24</td>
</tr>
<tr>
<td>1.12 Implementation Issues</td>
<td>26</td>
</tr>
<tr>
<td>1.12.1 Independence of Basic Runtime Routines</td>
<td>26</td>
</tr>
<tr>
<td>1.12.2 Interaction with Signals</td>
<td>27</td>
</tr>
</tbody>
</table>


2 Point-to-Point Communication

2.1 Introduction and Overview
2.2 Blocking Send and Receive Operations
  2.2.1 Blocking Send
  2.2.2 Send Buffer and Message Data
  2.2.3 Message Envelope
  2.2.4 Comments on Send
  2.2.5 Blocking Receive
  2.2.6 Receive Buffer
  2.2.7 Message Selection
  2.2.8 Return Status
  2.2.9 Comments on Receive
2.3 Datatype Matching and Data Conversion
  2.3.1 Type Matching Rules
  2.3.2 Data Conversion
  2.3.3 Comments on Data Conversion
2.4 Semantics of Blocking Point-to-Point
  2.4.1 Buffering and Safety
  2.4.2 Order
  2.4.3 Progress
  2.4.4 Fairness
2.5 Example—Jacobi Iteration
2.6 Send-Receive
2.7 Null Processes
2.8 Nonblocking Communication
  2.8.1 Request Objects
  2.8.2 Posting Operations
  2.8.3 Completion Operations
  2.8.4 Examples
  2.8.5 Freeing Requests
  2.8.6 Nondestructive Test of status
  2.8.7 Semantics of Nonblocking Communications
### Contents

2.8.8 Comments on Semantics of Nonblocking Communications

2.9 Multiple Completions

2.10 Probe and Cancel

2.11 Persistent Communication Requests

2.12 Communication-Complete Calls with Null Request Handles

2.13 Communication Modes
   - 2.13.1 Blocking Calls
   - 2.13.2 Nonblocking Calls
   - 2.13.3 Persistent Requests
   - 2.13.4 Buffer Allocation and Usage
   - 2.13.5 Model Implementation of Buffered Mode
   - 2.13.6 Comments on Communication Modes

3 **User-Defined Datatypes and Packing**

3.1 Introduction

3.2 Introduction to User-Defined Datatypes

3.3 Datatype Accessors

3.4 Datatype Constructors
   - 3.4.1 Dup
   - 3.4.2 Contiguous
   - 3.4.3 Vector
   - 3.4.4 Hvector
   - 3.4.5 Indexed
   - 3.4.6 Block Indexed
   - 3.4.7 Hindexed
   - 3.4.8 Struct

3.5 Use of Derived Datatypes
   - 3.5.1 Commit
   - 3.5.2 Deallocation
   - 3.5.3 Relation to count
   - 3.5.4 Type Matching
   - 3.5.5 Message Length
3.6 Address Function 152
3.7 Datatype Resizing 154
  3.7.1 True Extent of Datatypes 159
3.8 Absolute Addresses 160
3.9 Array Datatype Constructors 161
3.10 Portability of Datatypes 170
3.11 Deprecated Functions 171
3.12 Pack and Unpack 175
  3.12.1 Canonical MPI_PACK and MPI_UNPACK 184
  3.12.2 Derived Datatypes versus Pack/Unpack 188

4 Collective Communications 191
4.1 Introduction and Overview 191
4.2 Operational Details 193
4.3 Communicator Argument 194
4.4 Barrier Synchronization 195
4.5 Global Communication Functions 195
4.6 Broadcast 197
  4.6.1 An Example Using MPI_BCAST 198
4.7 Gather 199
  4.7.1 Examples Using MPI_GATHER 200
  4.7.2 Gather, Vector Variant 202
  4.7.3 Examples Using MPI_GATHERV 203
4.8 Scatter 210
  4.8.1 An Example Using MPI_SCATTER 211
  4.8.2 Scatter: Vector Variant 212
  4.8.3 Examples Using MPI_SCATTERV 213
4.9 Gather to All 215
  4.9.1 An Example Using MPI_ALLGATHER 216
  4.9.2 Gather to All: Vector Variant 217
4.10 All to All Scatter/Gather 218
4.10.1 All to All: Vector Variant 219
4.10.2 All to All: Generalized Function 221
4.10.3 An Example Using MPI_ALLTOALLW 222

4.11 Global Reduction Operations 225
4.11.1 Reduce 225
4.11.2 Predefined Reduce Operations 229
4.11.3 MINLOC and MAXLOC 232
4.11.4 All Reduce 238
4.11.5 Reduce-Scatter 240

4.12 Scan Operations 242
4.12.1 Inclusive Scan 242
4.12.2 Exclusive Scan 243

4.13 User-Defined Operations for Reduce and Scan 244

4.14 The Semantics of Collective Communications 250

5 Communicators 255

5.1 Introduction 255
5.1.1 Division of Processes 255
5.1.2 Avoiding Message Conflicts between Modules 256
5.1.3 Extensibility by Users 256
5.1.4 Safety 256

5.2 Overview 257
5.2.1 Groups 257
5.2.2 Communicator 258
5.2.3 Communication Domains 259
5.2.4 Compatibility with Previous Practice 261

5.3 Group Management 262
5.3.1 Group Accessors 262
5.3.2 Group Constructors 264
5.3.3 Group Destructors 271

5.4 Communicator Management 272
5.4.1 Communicator Accessors 272
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4.2</td>
<td>Communicator Constructors</td>
<td>274</td>
</tr>
<tr>
<td>5.4.3</td>
<td>Communicator Destructor</td>
<td>279</td>
</tr>
<tr>
<td>5.5</td>
<td>Safe Parallel Libraries</td>
<td>280</td>
</tr>
<tr>
<td>5.6</td>
<td>Caching</td>
<td>287</td>
</tr>
<tr>
<td>5.6.1</td>
<td>Introduction</td>
<td>287</td>
</tr>
<tr>
<td>5.6.2</td>
<td>Caching Functions</td>
<td>290</td>
</tr>
<tr>
<td>5.6.3</td>
<td>Deprecated Functions</td>
<td>301</td>
</tr>
<tr>
<td>5.7</td>
<td>Intercommunication</td>
<td>304</td>
</tr>
<tr>
<td>5.7.1</td>
<td>Introduction</td>
<td>304</td>
</tr>
<tr>
<td>5.7.2</td>
<td>Intercommunicator Accessors</td>
<td>308</td>
</tr>
<tr>
<td>5.7.3</td>
<td>Intercommunicator Constructors and Destructors</td>
<td>310</td>
</tr>
<tr>
<td>5.7.4</td>
<td>Examples</td>
<td>314</td>
</tr>
<tr>
<td>6</td>
<td>Process Topologies</td>
<td>319</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>319</td>
</tr>
<tr>
<td>6.2</td>
<td>Virtual Topologies</td>
<td>320</td>
</tr>
<tr>
<td>6.3</td>
<td>Overlapping Topologies</td>
<td>321</td>
</tr>
<tr>
<td>6.4</td>
<td>Embedding in MPI</td>
<td>322</td>
</tr>
<tr>
<td>6.5</td>
<td>Cartesian Topology Functions</td>
<td>323</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Cartesian Constructor Function</td>
<td>323</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Cartesian Convenience Function: MPI.DIMS.Create</td>
<td>324</td>
</tr>
<tr>
<td>6.5.3</td>
<td>Cartesian Inquiry Functions</td>
<td>325</td>
</tr>
<tr>
<td>6.5.4</td>
<td>Cartesian Translator Functions</td>
<td>327</td>
</tr>
<tr>
<td>6.5.5</td>
<td>Cartesian Shift Function</td>
<td>328</td>
</tr>
<tr>
<td>6.5.6</td>
<td>Cartesian Partition Function</td>
<td>331</td>
</tr>
<tr>
<td>6.5.7</td>
<td>Cartesian Low-Level Functions</td>
<td>332</td>
</tr>
<tr>
<td>6.6</td>
<td>Graph Topology Functions</td>
<td>333</td>
</tr>
<tr>
<td>6.6.1</td>
<td>Graph Constructor Function</td>
<td>334</td>
</tr>
<tr>
<td>6.6.2</td>
<td>Graph Inquiry Functions</td>
<td>336</td>
</tr>
<tr>
<td>6.6.3</td>
<td>Graph Information Functions</td>
<td>337</td>
</tr>
<tr>
<td>6.6.4</td>
<td>Low-Level Graph Functions</td>
<td>339</td>
</tr>
<tr>
<td>6.7</td>
<td>Topology Inquiry Functions</td>
<td>340</td>
</tr>
<tr>
<td>6.8</td>
<td>An Application Example</td>
<td>340</td>
</tr>
</tbody>
</table>
7 Environmental Management
7.1 MPI Process Startup 353
7.2 Initialization and Exit 357
7.3 Implementation Information 364
  7.3.1 MPI Version Number 364
  7.3.2 Environmental Inquiries 365
7.4 Timers and Synchronization 368
7.5 Error Handling 369
  7.5.1 Error Handlers 371
  7.5.2 Deprecated Functions 379
  7.5.3 Error Codes 380
7.6 Interaction with Executing Environment 384
  7.6.1 Independence of Basic Runtime Routines 384
  7.6.2 Interaction with Signals in POSIX 385

8 The MPI Profiling Interface
8.1 Requirements 387
8.2 Discussion 388
8.3 Logic of the Design 388
  8.3.1 Miscellaneous Control of Profiling 389
8.4 Examples 390
  8.4.1 Profiler Implementation 390
  8.4.2 MPI Library Implementation 391
  8.4.3 Complications 393
8.5 Multiple Levels of Interception 394

9 Conclusions
9.1 Design Issues 395
  9.1.1 Why Is MPI So Big? 395
  9.1.2 Should We Be Concerned about the Size of MPI? 395
  9.1.3 Why Does MPI Not Guarantee Buffering? 396
9.2 Portable Programming with MPI 397
9.2.1 Dependency on Buffering 398
9.2.2 Collective Communication and Synchronization 404
9.2.3 Ambiguous Communications and Portability 404

9.3 Heterogeneous Computing with MPI 406

9.4 MPI Implementations 407

References 409
Constants Index 411
Function Index 415
Index 421