Contents

Customer Support ix

1 About Version 2.1 1

1.1 Release Highlights 1

1.2 CM Platforms and Execution Models 2

1.3 New Features by Execution Model 3

1.4 Documentation Corrections 6

1.4.1 Fortran **READ/WRITE** and the **SO** Utilities 6

1.4.2 Passing an Alias Descriptor to a Subprogram 6

1.5 Notice of Possible Future Changes 7

1.5.1 Phasing Out Support for Foreign Compilers 7

1.5.2 Phasing Out Linking with Sun **f77** Libraries 7

1.5.3 Axis Ordering in Memory 7

1.5.4 Vector-Length Padding 8

1.5.5 Compatibility 8

2 Optimizations on Serial Dimensions **9**

2.1 Recognizing Serial Dimensions 10

2.2 No-Motion Detection 11

2.3 Nested and Flattened Loops 13

2.4 Better Handling of Context 14

2.5 Contiguous Serial Sections Passed in Place 15

2.6 Simple Dependence Analysis 18

2.7 Gathers/Scatters with **FORALL** 19

2.8 Serial Spreads with **FORALL** or **SPREAD** 19

3 User Control over Array Layout **20**

3.1 Controllable Axis Sequence: **-[no]axisreorder** 20

3.2 Controllable Vector-Length Padding: **-[no]padding** 23

Copyright © 1993 Thinking Machines Corporation

3.3 Detailed **LAYOUT** Directive 26
3.3.1 Detailed **LAYOUT** Directive Syntax 26
3.3.2 Restrictions on Axis Specifications 27
3.3.3 Examples of Detailed **LAYOUT** Directives 27
3.3.4 Restrictions on **LAYOUT** with Deferred-Shape Arrays 28
3.3.5 Restrictions on Array Size with Padding Enabled 28
3.3.6 Note on CM-5 **:SEND** Ordering 29

4 Array Aliasing Utilities 30
4.1 Using the Alias Package 31
4.1.1 Constructing Aliases 31
4.1.2 Operating upon Aliases 31
4.1.3 Array Descriptor Inquiries 32
4.1.4 Execution Environment 32
4.2 Restrictions and Caveats 33
4.3 Procedure Descriptions 34
CMF_ALIAS_DETAILED 35
CMF_ALIAS_TYPE 38
CMF_ARRAY_IS_ALIAS 41
CMF_ARRAY_SIZE_QUANTUM 42
CMF_GET_ALIAS_SAFETY_LEVEL 43
CMF_GET_AXIS_ORDERS 45
CMF_GET_GLOBAL_LOCATION 47
CMF_GET_LOCAL_EXTENTS 49
CMF_GET_LOCAL_STRIDES 51
CMF_GET_PHYSICAL_LOCATION 53
CMF_GET_PROCESSOR_MASKS 55
CMF_REMOVE_ALIAS 57
CMF_SET_ALIAS_LOCATION 58
CMF_SET_ALIAS_SAFETY_LEVEL 60
4.4 Alias Safety Checking: Error Codes 62
4.5 Examples of Detailed-Alias Construction 65

5 FORALL Optimizations 78
5.1 Operations Now Optimized 78
5.1.1 Gathers/Scatters on Serial Dimensions 78
5.1.2 Spread Operations on Serial Dimensions 79

Version 2.1, November 1993
Copyright © 1993 Thinking Machines Corporation

5.1.3 Scan Operations 80
5.1.4 Comparison with Utility Library **CMF_SCAN_** 81
5.1.5 Send-with-Combiner Operations 83
5.2 Expressions Now Parallelized 84
5.3 Temporary Restrictions 85
5.4 Permanent Restrictions 86
5.5 Using Alternative Forms of **FORALL** 87

6 Other Optimizations 88
6.1 Global Optimizer Improvements 88
6.1.1 Dead-Code Elimination 88
6.1.2 Copy Propagation 89
6.1.3 Second, or "Post," Run of Optimizer 89
6.1.4 Selectively Disabling New Optimizer Features 90
6.2 Alignment of Dynamic Arrays 91
6.2.1 The Compiler's View of Dynamic Arrays 91
6.2.2 **ALIGN** Directives and Dynamic Arrays 92
6.2.3 Using **ALIGN** Effectively with Dynamic Arrays 94
6.3 Compile-Time Shape Checking 96
6.3.1 Run-Time Geometry Checking 97
6.4 Single-Precision Memory Accesses in Vector Unit Code 98
6.5 A Change: Geometries of Array Arguments 99
6.6 Linking with CMSSL to Enhance Performance 100
6.6.1 Linking with CMSSL on the CM-5 100
6.6.2 Linking with CMSSL on the CM-2/200 100

7 Global/Local Programming 101
7.1 Flow of Control 104
7.2 Global/Local Data 105
7.3 Arrays 105
7.3.1 Parallel Arrays 106
7.3.2 Serial Arrays 108
7.4 Restrictions 110
7.5 Global/Local Library Functions 110
CMGL_BROADCAST_SERIAL_ARRAY 111

Copyright © 1993 Thinking Machines Corporation

CMGL_local_to_global 113
CMGL_global_to_local 116
7.6 Program Construction 118
7.6.1 How to Write the CM Fortran Global Program 118
7.6.2 How to Call a Local Routine 119
7.6.3 How to Write a Local Routine in CM Fortran 119
7.6.4 How to Write a Local Routine in C 121
7.6.5 The Prototype File 123
7.6.6 Compiling and Linking 124
7.6.7 Debugging 126
7.6.8 Profiling 127

8 I/O from CM Fortran 128
8.1 Parallel I/O with **READ** and **WRITE** 128
8.1.1 Specifying a Target File 128
8.1.2 Examples 130
8.1.3 When is I/O Parallel? 132
8.2 Enhancement of Other I/O Statements 134
8.2.1 **PHYSICAL** Keyword for **OPEN** 134
8.2.2 New Specifiers in **OPEN** Statement 134
8.2.3 New Specifiers in **INQUIRE** Statement 136
8.2.4 New Format for **NAMELIST** 137
8.2.5 **BACKSPACE** and **REWIND** on Redirected Streams 138
8.2.6 Printing First Character in Output Streams 138
8.3 Utility Library Procedures 139
CMF_FILE_LLSEEK 141
CMF_FILE_LTRUNCATE 144
CMF_FILE_OPEN_READONLY 146
8.4 Note on I/O Unit Numbers 147
8.5 Note on Back-Compatibility of Utility I/O 147

9 Array Pointers 148
9.1 Declaring Array Pointers 148
9.2 Allocating Pointer “Target” Arrays 149
9.3 Using Array Pointers 150
9.4 Summary of Syntax 152

Copyright © 1993 Thinking Machines Corporation

9.4.1 **POINTER** Attribute 152
9.4.2 **POINTER** Statement 153
9.4.3 **ALLOCATE** Statement 153
9.4.4 **DEALLOCATE** Statement 154
9.4.5 **ASSOCIATED** Function 154

10 64-Bit Integers and the Kind Mechanism 155
10.1 Type Specifications 155
10.1.1 **KIND** Keyword 156
10.1.2 Typed Constants 156
10.2 Enhanced Intrinsic Functions 157
10.3 Utility Library Support 158
10.3.1 64-Bit Send Addresses 158
10.4 Restrictions on 64-Bit Integers 159

Sample Program 161

Copyright © 1993 Thinking Machines Corporation