

Table of Contents

Introduction	1
The Data Parallel Programming Model: A Semantic Perspective	
Luc Bougé	4
1. Introduction: Execution Model vs. Programming Model	4
2. From Control Parallelism to Data Parallelism	7
3. Data Parallelism as a Programming Model	10
3.1 Macroscopic and Microscopic Viewpoints	11
3.2 Automatic Parallelization	14
3.3 CSP Communication-Closed Layers	16
4. Basic Semantic Concepts for Data Parallelism	17
4.1 Data Parallel Assignment	18
4.2 General Data Parallel Communication	18
4.3 Synchronous Parallel Control	21
4.4 Bounded Asynchrony	21
5. Conclusion	22
An Introduction to HPF	
Robert S. Schreiber	27
1. Data Parallel Programming in High Performance Fortran	27
1.1 Data Mapping is Advice and Assertion	29
1.2 HPF's Data Mapping Language	29
1.3 Mapping and Communication in HPF	31
1.4 Parallelism	32
1.5 Extended Data Parallelism	33
1.6 Intrinsics and the HPF Library	33
1.7 Program Structure	34
1.8 Extrinsic Subprograms	35
2. HPF in the Future	35
2.1 A Richer Data Mapping Language	37
2.2 Independent Loops with Reductions	38
2.3 Mapping Computation	39

VIII Table of Contents

2.4 Spawning, Synchronizing, and Killing HPF Threads	41
A Data Parallel Scientific Computing Introduction	
Serge G. Petiton and Nahid Emad	45
1. Introduction	45
2. The Data Parallel Programming Model	46
3. Data Parallel Algorithm Examples	47
3.1 AP λ A: Matrix + (Scalar $\times$ Matrix)	47
3.2 AXPX: (Matrix $\times$ Vector) + Vector	47
3.3 Gaussian Elimination	48
3.4 Back Substitution	49
3.5 Gauss-Jordan Diagonalization	50
3.6 The QR Method	51
3.7 Discussions	53
4. First Data Parallel Programming Rules	55
4.1 Linear Algebra Methods	55
4.2 Very Large Sparse Matrix Case	56
5. Data Parallel Sparse Matrix Vector Multiplication	56
5.1 Data Parallel Mappings and Alignments	56
5.2 A Data Parallel Algorithm	60
6. Preconditioned Conjugate Gradient Method	61
6.1 The Numerical Method	61
6.2 Polynomial Preconditioner	61
6.3 Data Mapping and Data Parallel Algorithm	62
7. Conclusion	62
Current Status of the SUIF Research Project	
Monica S. Lam	65
1. Introduction	65
2. The SUIF Compiler Infrastructure	66
3. Pointer Alias Analysis	67
3.1 Background	67
3.2 Algorithms and Experience	68
4. Interprocedural Parallelization	69
4.1 Problem Formulation	69
4.2 Algorithms and Experience	70
5. Locality and Parallelism	71
5.1 Problem Formulation	72
5.2 Algorithms and Experience	73
6. Concluding Remarks	75

Automatic Parallelization in the Polytope Model

Paul Feautrier	79
1. The Geometry of Programs	79
2. Geometrical Semantics	81
2.1 Programs as Orders	81
2.2 Static Control Programs	83
2.3 The Dataflow Calculation	85
3. Basic Tools for Handling Polyhedra and $\mathbb{Z}$ -Polyhedra	86
3.1 Polyhedra and Polytopes	86
3.2 $\mathbb{Z}$ -Modules	87
3.3 $\mathbb{Z}$ -Polyhedra	88
3.4 Parametric Problems	89
4. Program Transformations	90
4.1 Reordering Transformations	91
4.2 Storage Management	98
5. Loop Rewriting and Code Generation	98
5.1 The Case of a Perfect Loop Nest	98
5.2 The Case of a Complete Program	99
5.3 Communication Code	100
6. Conclusion: The Limits of the Polytope Model	100

State of the Art in Compiling HPF

Fabien Coelho, Cécile Germain, and Jean-Louis Pazat	104
1. Introduction	104
2. HPF Compilation Issues	105
2.1 Application	105
2.2 Language	107
2.3 Target	108
2.4 Outline	108
3. Compilation Overview	108
3.1 Compiling for the SMPD Model	109
3.2 Owner Computes Rule	109
3.3 Run-Time Resolution	110
3.4 Index and Communication Sets	111
3.5 Code Generation	112
3.6 Optimizations	113
3.7 Affine Framework	114
3.8 The Alignment/Distribution Problem	115
4. Regular References	116
4.1 Closed Forms	116
4.2 Finite State Machines	118
4.3 Polyhedra and Lattices	119
5. Irregular Problems	122
5.1 The Parti+ Library	123

5.2	Generated Code	124
5.3	The ESD and Schedule Reuse	126
6.	Memory Allocation and Addressing	126
6.1	Issues	126
6.2	Techniques	127
7.	Conclusion	130

Tools for High Performance Fortran: A Survey

Jean-Louis Pazat	134
1. Introduction	134
2. Restructuring Tools	134
2.1 FORGExplorer (APR)	135
2.2 FORESYS (SIMULOG)	135
2.3 PIPS (Ecole des mines de Paris)	136
2.4 VAST/77toHPF (Pacific-Sierra Research Corp.)	137
2.5 Analyst (PPPE)	138
3. Data Distribution Visualization Tools	139
3.1 Annai/DDV (CSCS / NEC)	139
3.2 GDDT (Johannes Kepler University Linz)	140
3.3 HPF-Builder (LIFL)	141
3.4 DAQV (Ptools)	142
4. Performance Evaluation and Measurement Tools	143
4.1 Annai/PMA (CSCS / NEC)	143
4.2 dPablo (University of Illinois & Rice University)	144
4.3 Evaluation Module (TUM München)	145
4.4 EPPP Performance Debugger (CRIM Montréal)	146
4.5 ESP (University of Texas at Austin & Syracuse University) ...	147
4.6 PSE Profiler (Digital Equipment Corporation)	148
4.7 pgprof (Portland Group Inc.)	148
4.8 P3T (University of Vienna)	149
4.9 VFPMS (University of Vienna)	149
5. Debugging Tools	150
5.1 Annai/PDT (CSCS / NEC)	150
5.2 Source Level Debugger, MDB (N.A. Software)	151
5.3 PSE Debugger (Digital Equipment Corporation)	151
6. Runtime Systems	152
6.1 Annai Run-Time (CSCS / NEC)	152
6.2 Distributed Array Library - DALIB (GMD SCAI)	154
6.3 PSE (Digital Equipment Corporation)	154
6.4 HPFC (Ecole des mines Paris)	155
6.5 Pandore (IRISA)	155
7. Other HPF Tools	156
7.1 HPF Binding for MPI (Argonne Nat. Lab.)	156
8. Compilers' Home Pages	157

Implementing Data Parallel Programs on Commodity Clusters

P. J. Hatcher, R. D. Russell, S. Kumaran, and M. J. Quinn	159
1. Introduction	159
2. Efficient Communication in Clusters	161
3. Data Parallel Programming in Clusters	162
3.1 UNH Net* and UNH C*	162
3.2 Preliminary Performance Results	164
3.3 Open Questions	165
4. Clusters of Multiprocessors	169
4.1 Language Design Issues	169
4.2 Generating Code for a Multiprocessor Node	170
4.3 Structuring Communication	171
5. Conclusions	176

Task Parallelism and High-Performance Languages

Ian Foster	179
1. Introduction	179
2. High Performance Fortran	180
3. The Role of Task Parallelism	181
3.1 Software Engineering	182
3.2 Locality	183
3.3 Scheduling	184
3.4 Summary	186
4. Implicit Parallelism	187
5. Language-Based Task Parallelism	188
5.1 Experiences	190
5.2 Determinism	190
6. Library Based Approaches	191
7. Conclusions	192

Supporting Irregular and Dynamic Computations in Data Parallel Languages

Jean-Luc Dekeyser and Philippe Marquet	197
1. Irregularity and Data Parallelism	197
2. Data Parallel Language Classification	199
3. Irregularity and Dynamicity in Data Parallel Languages	201
4. Irregular and Dynamic Constructions in Data Parallel Languages ..	202
4.1 Dynamic Instantiation Machines	202
4.2 Irregular Instantiation Machines	204
4.3 Dynamic and Irregular Parallel Objects	206
4.4 Irregular Distributions	210
5. Idole	214

6. Conclusion	217
---------------------	-----

Data Parallelism and Functional Programming

Björn Lisper	220
1. Introduction	220
2. Data-Parallelism	221
2.1 Elementwise Applied Operations	222
2.2 Communication Operations	223
2.3 Replication	225
2.4 Restriction	225
2.5 Reductions	226
2.6 Index Sets	226
3. Functional Languages	227
4. Data-Parallel Functional Languages	230
4.1 Formalisms	231
5. An Abstract Formalism for Aggregate Data	232
5.1 Data Fields	233
5.2 A Formal Language Scheme	234
5.3 Explicit Restriction	235
5.4 Data-Parallel Operations	235
5.5 Algebraic Laws	238
6. Towards Implementations	240
6.1 C.P.O.'s with Explicitly Undefined Values	240
6.2 Parallel Evaluators	241
6.3 Loosely Specified Reductions	243
6.4 Extent Analysis	244
6.5 Non-Consistent Extensions	245
6.6 Implementations of $*$	245
7. Conclusions and Further Research	246

Formal Validation of Data Parallel Programs: Introducing the Assertional Approach

L. Bougé, D. Cachera, Y. Le Guyadec, G. Utard, and B. Viot	252
1. Introduction	252
2. A Simple Data Parallel Language and its Semantics	253
2.1 The $\mathcal{L}$ Language	253
2.2 A Natural Semantics for $\mathcal{L}$	256
3. A Two-Component Assertional Proof System for $\mathcal{L}$ Programs	258
3.1 Two-Component Assertion Language	258
3.2 Proof System	259
3.3 Example	262
4. Weakest Preconditions of $\mathcal{L}$ Programs	264
4.1 The Definability Property	264

4.2 Definability of the Weakest Strict Preconditions of Linear Programs	266
4.3 Definability of Weakest Liberal Preconditions	270
4.4 Discussion: Why are Weakest Preconditions Not Definable in General?	273
5. Completeness of the Proof System	274
5.1 Extending the Completeness Proof to Non-Regular, Linear Programs	276
5.2 Extending the Completeness Proof to Non-Plain Specifications .	278
6. Conclusion and Related Work	279
General Index	282