

Contents

Acknowledgments.....	xi
Preface.....	xiii

PART I CUDA FORTRAN PROGRAMMING

CHAPTER 1 Introduction	3
1.1 A Brief History of GPU Computing	3
1.2 Parallel Computation	5
1.3 Basic Concepts.....	5
1.3.1 A First CUDA Fortran Program.....	6
1.3.2 Extending to Larger Arrays.....	9
1.3.3 Multidimensional Arrays	12
1.4 Determining CUDA Hardware Features and Limits	13
1.4.1 Single and Double Precision	21
1.5 Error Handling	23
1.6 Compiling CUDA Fortran Code	24
1.6.1 Separate Compilation	27
 CHAPTER 2 Performance Measurement and Metrics	 31
2.1 Measuring Kernel Execution Time	31
2.1.1 Host-Device Synchronization and CPU Timers	32
2.1.2 Timing via CUDA Events	32
2.1.3 Command Line Profiler	34
2.1.4 The nvprof Profiling Tool.....	35
2.2 Instruction, Bandwidth, and Latency Bound Kernels.....	36
2.3 Memory Bandwidth.....	39
2.3.1 Theoretical Peak Bandwidth.....	39
2.3.2 Effective Bandwidth	41
2.3.3 Actual Data Throughput vs. Effective Bandwidth	42
 CHAPTER 3 Optimization.....	 43
3.1 Transfers between Host and Device	44
3.1.1 Pinned Memory.....	45
3.1.2 Batching Small Data Transfers	49
3.1.3 Asynchronous Data Transfers (<i>Advanced Topic</i>).....	52
3.2 Device Memory	61
3.2.1 Declaring Data in Device Code	62

3.2.2 Coalesced Access to Global Memory	63
3.2.3 Texture Memory	74
3.2.4 Local Memory	79
3.2.5 Constant Memory	82
3.3 On-Chip Memory	85
3.3.1 L1 Cache	85
3.3.2 Registers	86
3.3.3 Shared Memory	87
3.4 Memory Optimization Example: Matrix Transpose	93
3.4.1 Partition Camping (<i>Advanced Topic</i>)	99
3.5 Execution Configuration	102
3.5.1 Thread-Level Parallelism	102
3.5.2 Instruction-Level Parallelism	105
3.6 Instruction Optimization	107
3.6.1 Device Intrinsic	108
3.6.2 Compiler Options	108
3.6.3 Divergent Warps	109
3.7 Kernel Loop Directives	110
3.7.1 Reductions in CUF Kernels	113
3.7.2 Streams in CUF Kernels	113
3.7.3 Instruction-Level Parallelism in CUF Kernels	114
CHAPTER 4 Multi-GPU Programming	115
4.1 CUDA Multi-GPU Features	115
4.1.1 Peer-to-Peer Communication	117
4.1.2 Peer-to-Peer Direct Transfers	121
4.1.3 Peer-to-Peer Transpose	131
4.2 Multi-GPU Programming with MPI	140
4.2.1 Assigning Devices to MPI Ranks	141
4.2.2 MPI Transpose	147
4.2.3 GPU-Aware MPI Transpose	149
 PART II CASE STUDIES	
CHAPTER 5 Monte Carlo Method	155
5.1 CURAND	156
5.2 Computing π with CUF Kernels	161
5.2.1 IEEE-754 Precision (<i>Advanced Topic</i>)	164
5.3 Computing π with Reduction Kernels	168
5.3.1 Reductions with Atomic Locks (<i>Advanced Topic</i>)	173
5.4 Accuracy of Summation	174
5.5 Option Pricing	180

CHAPTER 6	Finite Difference Method	189
6.1	Nine-Point 1D Finite Difference Stencil	189
6.1.1	Data Reuse and Shared Memory	190
6.1.2	The x -Derivative Kernel	191
6.1.3	Derivatives in y and z	196
6.1.4	Nonuniform Grids	200
6.2	2D Laplace Equation	204
CHAPTER 7	Applications of Fast Fourier Transform	211
7.1	CUFFT	211
7.2	Spectral Derivatives	219
7.3	Convolution	222
7.4	Poisson Solver	229

PART III APPENDICES

APPENDIX A	Tesla Specifications	237
APPENDIX B	System and Environment Management	241
B.1	Environment Variables	241
B.1.1	General	241
B.1.2	Command Line Profiler	242
B.1.3	Just-in-Time Compilation	242
B.2	nvidia-smi System Management Interface	242
B.2.1	Enabling and Disabling ECC	243
B.2.2	Compute Mode	245
B.2.3	Persistence Mode	246
APPENDIX C	Calling CUDA C from CUDA Fortran	249
C.1	Calling CUDA C Libraries	249
C.2	Calling User-Written CUDA C Code	252
APPENDIX D	Source Code	255
D.1	Texture Memory	255
D.2	Matrix Transpose	259
D.3	Thread- and Instruction-Level Parallelism	267
D.4	Multi-GPU Programming	271
D.4.1	Peer-to-Peer Transpose	272
D.4.2	MPI Transpose with Host MPI Transfers	279
D.4.3	MPI Transpose with Device MPI Transfers	284

D.5 Finite Difference Code.....	289
D.6 Spectral Poisson Solver.....	310

References.....	317
-----------------	-----

Index.....	319
------------	-----