



---

# Contents

<b>Foreword</b>	<b>v</b>
<b>Preface</b>	<b>vii</b>
<b>Contributor List</b>	<b>ix</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xxi</b>
<b>Acknowledgments</b>	<b>xxxiii</b>
<b>1 Introduction</b>	<b>1</b>
<i>E. Wes Bethel</i>	
1.1 Historical Perspective . . . . .	1
1.2 Moore's Law and the Data Tsunami . . . . .	2
1.3 Focus of this Book . . . . .	3
1.4 Book Organization and Themes . . . . .	3
1.5 Conclusion . . . . .	6
<b>I Distributed Memory Parallel Concepts and Systems</b>	<b>7</b>
<b>2 Parallel Visualization Frameworks</b>	<b>9</b>
<i>Hank Childs</i>	
2.1 Introduction . . . . .	9
2.2 Background . . . . .	11
2.2.1 Parallel Computing . . . . .	11
2.2.2 Data Flow Networks . . . . .	12
2.3 Parallelization Strategy . . . . .	13
2.4 Usage . . . . .	16
2.5 Advanced Processing Techniques . . . . .	17
2.5.1 Contracts . . . . .	18
2.5.2 Data Subsetting . . . . .	19
2.5.3 Parallelization Artifacts . . . . .	19
2.5.4 Scheduling . . . . .	21
2.6 Conclusion . . . . .	22

<b>3</b>	<b>Remote and Distributed Visualization Architectures</b>	<b>25</b>
	<i>E. Wes Bethel and Mark Miller</i>	
3.1	Introduction . . . . .	26
3.2	Visualization Performance Fundamentals and Networks . . . . .	26
3.3	Send-Images Partitioning . . . . .	28
3.4	Send-Data Partitioning . . . . .	30
3.5	Send-Geometry Partitioning . . . . .	31
3.6	Hybrid and Adaptive Approaches . . . . .	32
3.7	Which Pipeline Partitioning Works the Best? . . . . .	33
3.8	Case Study: Visapult . . . . .	36
	3.8.1 Visapult Architecture: The Send-Geometry Partition . . . . .	37
	3.8.2 Visapult Architecture: The Send-Data Partition . . . . .	38
3.9	Case Study: Chromium Renderserver . . . . .	39
3.10	Case Study: VisIt and Dynamic Pipeline Reconfiguration . . . . .	42
	3.10.1 How VisIt Manages Pipeline Partitioning . . . . .	43
	3.10.2 Send-Geometry Partitioning . . . . .	43
	3.10.3 Send-Images Partitioning . . . . .	44
	3.10.4 Automatic Pipeline Partitioning Selection . . . . .	44
3.11	Conclusion . . . . .	45
<b>4</b>	<b>Rendering</b>	<b>49</b>
	<i>Charles Hansen, E. Wes Bethel, Thiago Ize, and Carson Brownlee</i>	
4.1	Introduction . . . . .	49
4.2	Rendering Taxonomy . . . . .	50
4.3	Rendering Geometry . . . . .	52
4.4	Volume Rendering . . . . .	53
4.5	Real-Time Ray Tracer for Visualization on a Cluster . . . . .	56
	4.5.1 Load Balancing . . . . .	57
	4.5.2 Display Process . . . . .	58
	4.5.3 Distributed Cache Ray Tracing . . . . .	58
	4.5.3.1 DC BVH . . . . .	59
	4.5.3.2 DC Primitives . . . . .	60
	4.5.4 Results . . . . .	61
	4.5.5 Maximum Frame Rate . . . . .	62
4.6	Conclusion . . . . .	66
<b>5</b>	<b>Parallel Image Compositing Methods</b>	<b>71</b>
	<i>Tom Peterka and Kwan-Liu Ma</i>	
5.1	Introduction . . . . .	72
5.2	Basic Concepts and Early Work in Compositing . . . . .	72
	5.2.1 Definition of Image Composition . . . . .	73
	5.2.2 Fundamental Image Composition Algorithms . . . . .	74
	5.2.3 Image Compositing Hardware . . . . .	77
5.3	Recent Advances . . . . .	77
	5.3.1 2-3 Swap . . . . .	77

5.3.2	Radix-k . . . . .	78
5.3.3	Optimizations . . . . .	80
5.4	Results . . . . .	81
5.5	Discussion and Conclusion . . . . .	82
5.5.1	Conclusion . . . . .	83
5.5.2	Directions for Future Research . . . . .	85
<b>6</b>	<b>Parallel Integral Curves</b>	<b>91</b>
	<i>David Pugmire, Tom Peterka, and Christoph Garth</i>	
6.1	Introduction . . . . .	92
6.2	Challenges to Parallelization . . . . .	94
6.2.1	Problem Classification . . . . .	94
6.3	Approaches to Parallelization . . . . .	95
6.3.1	Test Data . . . . .	96
6.3.2	Parallelization Over Seed Points . . . . .	98
6.3.3	Parallelization Over Data . . . . .	99
6.3.4	A Hybrid Approach to Parallelization . . . . .	99
6.3.5	Algorithm Analysis . . . . .	103
6.3.6	Hybrid Data Structure and Communication Algorithm	106
6.4	Conclusion . . . . .	111
<b>II</b>	<b>Advanced Processing Techniques</b>	<b>115</b>
<b>7</b>	<b>Query-Driven Visualization and Analysis</b>	<b>117</b>
	<i>Oliver Rübél, E. Wes Bethel, Prabhat, and Kesheng Wu</i>	
7.1	Introduction . . . . .	118
7.2	Data Subsetting and Performance . . . . .	119
7.2.1	Bitmap Indexing . . . . .	120
7.2.2	Data Interfaces . . . . .	122
7.3	Formulating Multivariate Queries . . . . .	124
7.3.1	Parallel Coordinates Multivariate Query Interface . . . . .	125
7.3.2	Segmenting Query Results . . . . .	127
7.4	Applications of Query-Driven Visualization . . . . .	129
7.4.1	Applications in Forensic Cybersecurity . . . . .	129
7.4.2	Applications in High Energy Physics . . . . .	132
7.4.2.1	Linear Particle Accelerator . . . . .	133
7.4.2.2	Laser Plasma Particle Accelerator . . . . .	135
7.5	Conclusion . . . . .	139
<b>8</b>	<b>Progressive Data Access for Regular Grids</b>	<b>145</b>
	<i>John Clyne</i>	
8.1	Introduction . . . . .	146
8.2	Preliminaries . . . . .	146
8.3	Z-Order Curves . . . . .	147
8.3.1	Constructing the Curve . . . . .	149

8.3.2	Progressive Access . . . . .	149
8.4	Wavelets . . . . .	151
8.4.1	Linear Decomposition . . . . .	153
8.4.2	Scaling and Wavelet Functions . . . . .	154
8.4.3	Wavelets and Filter Banks . . . . .	156
8.4.4	Compression . . . . .	158
8.4.5	Boundary Handling . . . . .	159
8.4.6	Multiple Dimensions . . . . .	164
8.4.7	Implementation Considerations . . . . .	164
8.4.7.1	Blocking . . . . .	165
8.4.7.2	Wavelet Choice . . . . .	165
8.4.7.3	Coefficient Addressing . . . . .	166
8.4.8	A Hybrid Approach . . . . .	166
8.4.9	Volume Rendering Example . . . . .	167
8.5	Further Reading . . . . .	167
<b>9</b>	<b><i>In Situ</i> Processing</b> . . . . .	<b>171</b>
	<i>Hank Childs, Kwan-Liu Ma, Hongfeng Yu, Brad Whitlock, Jeremy Meredith, Jean Favre, Scott Klasky, Norbert Podhorszki, Karsten Schwan, Matthew Wolf, Manish Parashar, and Fan Zhang</i>	
9.1	Introduction . . . . .	172
9.2	Tailored Co-Processing at High Concurrency . . . . .	174
9.3	Co-Processing With General Visualization Tools Via Adaptors . . . . .	175
9.3.1	Adaptor Design . . . . .	178
9.3.2	High Level Implementation Issues . . . . .	178
9.3.3	In Practice . . . . .	179
9.3.4	Co-Processing Performance . . . . .	181
9.4	Concurrent Processing . . . . .	183
9.4.1	Service Oriented Architecture for Data Management in HPC . . . . .	183
9.4.2	The ADaptable I/O System, ADIOS . . . . .	184
9.4.3	Data Staging for <i>In Situ</i> Processing . . . . .	185
9.4.4	Exploratory Visualization with VisIt and Paraview Using ADIOS . . . . .	186
9.5	<i>In Situ</i> Analytics Using Hybrid Staging . . . . .	187
9.6	Data Exploration and <i>In Situ</i> Processing . . . . .	190
9.6.1	<i>In Situ</i> Visualization by Proxy . . . . .	190
9.6.2	<i>In Situ</i> Data Triage . . . . .	191
9.7	Conclusion . . . . .	193
<b>10</b>	<b>Streaming and Out-of-Core Methods</b> . . . . .	<b>199</b>
	<i>David E. DeMarle, Berk Geveci, Jon Woodring, and Jim Ahrens</i>	
10.1	External Memory Algorithms . . . . .	200
10.2	Taxonomy of Streamed Visualization . . . . .	202
10.3	Streamed Visualization Concepts . . . . .	204

10.3.1	Data Structures . . . . .	204
10.3.2	Repetition . . . . .	205
10.3.3	Algorithms . . . . .	205
10.3.4	Sparse Traversal . . . . .	207
10.4	Survey of Current State of the Art . . . . .	209
10.4.1	Rendering . . . . .	209
10.4.2	Streamed Processing of Unstructured Data . . . . .	210
10.4.3	General Purpose Systems . . . . .	211
10.4.4	Asynchronous Systems . . . . .	212
10.4.5	Lazy Evaluation . . . . .	214
10.5	Conclusion . . . . .	215
<b>III Advanced Architectural Challenges and Solutions</b>		<b>221</b>
<b>11 GPU-Accelerated Visualization</b>		<b>223</b>
	<i>Marco Ament, Steffen Frey, Christoph Müller, Sebastian Grottel, Thomas Ertl, and Daniel Weiskopf</i>	
11.1	Introduction . . . . .	224
11.2	Programmable Graphics Hardware . . . . .	225
11.2.1	High-Level Shader Languages . . . . .	226
11.2.2	General Purpose Computing on GPUs . . . . .	227
11.2.3	GPGPU Programming Languages . . . . .	228
11.3	GPU-Accelerated Volume Rendering . . . . .	229
11.3.1	Basic GPU Techniques . . . . .	229
11.3.1.1	2D Texture-Based Rendering . . . . .	229
11.3.1.2	3D Texture-Based Rendering . . . . .	230
11.3.1.3	Ray Casting . . . . .	231
11.3.2	Advanced GPU Algorithms . . . . .	231
11.3.3	Scalable Volume Rendering on GPU-Clusters . . . . .	233
11.3.3.1	Sort-Last Volume Rendering . . . . .	233
11.3.3.2	Sort-First Volume Rendering . . . . .	234
11.4	Particle-Based Rendering . . . . .	235
11.4.1	GPU-Based Glyph Rendering . . . . .	236
11.4.2	Large Molecular Dynamics Visualization . . . . .	238
11.4.3	Iterative Surface Ray Casting . . . . .	239
11.5	GPGPU High Performance Environments . . . . .	240
11.5.1	New Challenges in GPGPU Environments . . . . .	240
11.5.2	Distributed GPU Computing . . . . .	241
11.5.3	Distributed Heterogeneous Computing . . . . .	242
11.6	Large Display Visualization . . . . .	243
11.6.1	Flat Panel-Based Systems . . . . .	243
11.6.2	Projection-Based Systems . . . . .	244
11.6.3	Rendering for Large Displays . . . . .	246

<b>12 Hybrid Parallelism</b>	<b>261</b>
<i>E. Wes Bethel, David Camp, Hank Childs, Christoph Garth, Mark Howison, Kenneth I. Joy, and David Pugmire</i>	
12.1 Introduction . . . . .	262
12.2 Hybrid Parallelism and Volume Rendering . . . . .	264
12.2.1 Background and Previous Work . . . . .	264
12.2.2 Implementation . . . . .	265
12.2.2.1 Shared-Memory Parallel Ray Casting . . . . .	266
12.2.2.2 Parallel Compositing . . . . .	266
12.2.3 Experiment Methodology . . . . .	267
12.2.4 Results . . . . .	268
12.2.4.1 Initialization . . . . .	268
12.2.4.2 Ghost Data/Halo Exchange . . . . .	269
12.2.4.3 Ray Casting . . . . .	269
12.2.4.4 Compositing . . . . .	272
12.2.4.5 Overall Performance . . . . .	272
12.3 Hybrid Parallelism and Integral Curve Calculation . . . . .	275
12.3.1 Background and Context . . . . .	275
12.3.2 Design and Implementation . . . . .	276
12.3.2.1 Parallelize Over Seeds . . . . .	276
12.3.2.2 Parallelize Over Blocks . . . . .	277
12.3.3 Experiment Methodology . . . . .	278
12.3.3.1 Factors Influencing Parallelization Strategy . . . . .	278
12.3.3.2 Test Cases . . . . .	279
12.3.3.3 Runtime Environment . . . . .	279
12.3.3.4 Measurements . . . . .	280
12.3.4 Results . . . . .	280
12.3.4.1 Parallelization Over Seeds . . . . .	280
12.3.4.2 Parallelization Over Blocks . . . . .	282
12.4 Conclusion and Future Work . . . . .	283
<b>13 Visualization at Extreme Scale Concurrency</b>	<b>291</b>
<i>Hank Childs, David Pugmire, Sean Ahern, Brad Whitlock, Mark Howison, Prabhat, Gunther Weber, and E. Wes Bethel</i>	
13.1 Overview—Pure Parallelism . . . . .	292
13.2 Massive Data Experiments . . . . .	293
13.2.1 Varying over Supercomputing Environment . . . . .	296
13.2.2 Varying over I/O Pattern . . . . .	297
13.2.3 Varying over Data Generation . . . . .	298
13.3 Scaling Experiments . . . . .	299
13.3.1 Study Overview . . . . .	299
13.3.2 Results . . . . .	300
13.4 Pitfalls at Scale . . . . .	301
13.4.1 Volume Rendering . . . . .	301
13.4.2 All-to-One Communication . . . . .	303

13.4.3	Shared Libraries and Start-up Time . . . . .	304
13.5	Conclusion . . . . .	305
<b>14</b>	<b>Performance Optimization and Auto-Tuning</b>	<b>307</b>
	<i>E. Wes Bethel and Mark Howison</i>	
14.1	Introduction . . . . .	308
14.2	Optimizing Performance of a 3D Stencil Operator on the GPU	310
14.2.1	Introduction and Related Work . . . . .	310
14.2.2	Design and Methodology . . . . .	312
14.2.3	Results . . . . .	313
14.2.3.1	Algorithmic Design Option: Width-, Height-, and Depth-Row Kernels . . . . .	313
14.2.3.2	Device-Specific Feature: Constant Versus Global Memory for Filter Weights . . . . .	314
14.2.3.3	Tunable Algorithmic Parameter: Thread Block Size . . . . .	314
14.2.4	Lessons Learned . . . . .	317
14.3	Optimizing Ray Casting Volume Rendering on Multi-Core GPUs and Many-Core GPUs . . . . .	317
14.3.1	Introduction and Related Work . . . . .	317
14.3.2	Design and Methodology . . . . .	319
14.3.3	Results . . . . .	320
14.3.3.1	Tunable Parameter: Image Tile Size/CUDA Block Size . . . . .	320
14.3.3.2	Algorithmic Optimization: Early Ray Termi- nation . . . . .	323
14.3.3.3	Algorithmic Optimization: Z-Ordered Mem- ory . . . . .	324
14.3.4	Lessons Learned . . . . .	325
14.4	Conclusion . . . . .	326
<b>15</b>	<b>The Path to Exascale</b>	<b>331</b>
	<i>Sean Ahern</i>	
15.1	Introduction . . . . .	332
15.2	Future System Architectures . . . . .	332
15.3	Science Understanding Needs at the Exascale . . . . .	335
15.4	Research Directions . . . . .	338
15.4.1	Data Processing Modes . . . . .	338
15.4.1.1	<i>In Situ</i> Processing . . . . .	338
15.4.1.2	Post-Processing Data Analysis . . . . .	339
15.4.2	Visualization and Analysis Methods . . . . .	341
15.4.2.1	Support for Data Processing Modes . . . . .	341
15.4.2.2	Topological Methods . . . . .	342
15.4.2.3	Statistical Methods . . . . .	343
15.4.2.4	Adapting to Increased Data Complexity . . . . .	343



- 15.4.3 I/O and Storage Systems . . . . . 344
  - 15.4.3.1 Storage Technologies for the Exascale . . . . . 345
  - 15.4.3.2 I/O Middleware Platforms . . . . . 346
- 15.5 Conclusion and the Path Forward . . . . . 347

**IV High Performance Visualization Implementations 355**

**16 VisIt: An End-User Tool for Visualizing and Analyzing Very Large Data 357**

*Hank Childs, Eric Brugger, Brad Whitlock, Jeremy Meredith, Sean Ahern, David Pugmire, Kathleen Biagas, Mark Miller, Cyrus Harrison, Gunther H. Weber, Hari Krishnan, Thomas Fogal, Allen Sanderson, Christoph Garth, E. Wes Bethel, David Camp, Oliver Rübel, and Marc Durant, Jean M. Favre, Paul Navrátil*

- 16.1 Introduction . . . . . 358
- 16.2 Focal Points . . . . . 359
  - 16.2.1 Enable Data Understanding . . . . . 359
  - 16.2.2 Support for Large Data . . . . . 360
  - 16.2.3 Provide a Robust and Usable Product for End Users . 360
- 16.3 Design . . . . . 361
  - 16.3.1 Architecture . . . . . 361
  - 16.3.2 Parallelism . . . . . 362
  - 16.3.3 User Interface Concepts and Extensibility . . . . . 363
  - 16.3.4 The Size and Breadth of VisIt . . . . . 364
- 16.4 Successes . . . . . 364
  - 16.4.1 Scalability Successes . . . . . 365
  - 16.4.2 A Repository for Large Data Algorithms . . . . . 365
  - 16.4.3 Supercomputing Research Performed with VisIt . . . . 366
  - 16.4.4 User Successes . . . . . 366
- 16.5 Future Challenges . . . . . 368
- 16.6 Conclusion . . . . . 368

**17 IceT 373**

*Kenneth Moreland*

- 17.1 Introduction . . . . . 373
- 17.2 Motivation . . . . . 374
- 17.3 Implementation . . . . . 374
  - 17.3.1 Theoretical Limitations ... and How to Break Them . 375
  - 17.3.2 Pixel Reduction Techniques . . . . . 376
  - 17.3.3 Tricks to Boost the Frame Rate . . . . . 377
- 17.4 Application Programming Interface . . . . . 378
  - 17.4.1 Image Generation . . . . . 378
  - 17.4.2 Opaque versus Transparent Rendering . . . . . 379
- 17.5 Conclusion . . . . . 379

<b>18 The ParaView Visualization Application</b>	<b>383</b>
<i>Utkarsh Ayachit, Berk Geveci, Kenneth Moreland, and John Patchett, Jim Ahrens</i>	
18.1 Introduction . . . . .	384
18.2 Understanding the Need . . . . .	384
18.3 The ParaView Framework . . . . .	386
18.3.1 Configurations . . . . .	387
18.4 Parallel Data Processing . . . . .	387
18.5 The ParaView Application . . . . .	390
18.5.1 Graphical User Interface . . . . .	390
18.5.2 Scripting with Python . . . . .	391
18.6 Customizing with Plug-ins and Custom Applications . . . . .	391
18.7 Co-Processing: <i>In Situ</i> Visualization and Data Analysis . . . . .	392
18.8 ParaViewWeb: Interactive Visualization for the Web . . . . .	393
18.9 ParaView In Use . . . . .	394
18.9.1 Identifying and Validating Fragmentation in Shock Physics Simulation . . . . .	394
18.9.2 ParaView at the Los Alamos National Laboratory . . . . .	396
18.9.3 Analyzing Simulations of the Earth's Magnetosphere . . . . .	397
18.10 Conclusion . . . . .	398
<b>19 The ViSUS Visualization Framework</b>	<b>401</b>
<i>Valerio Pascucci, Giorgio Scorzelli, Brian Summa, Peer-Timo Bremer, Attila Gyulassy, Cameron Christensen, Sujin Philip, and Sidharth Kumar</i>	
19.1 Introduction . . . . .	402
19.2 ViSUS Software Architecture . . . . .	402
19.3 Applications . . . . .	408
<b>20 The VAPOR Visualization Application</b>	<b>415</b>
<i>Alan Norton and John Clyne</i>	
20.1 Introduction . . . . .	415
20.1.1 Features . . . . .	416
20.1.2 Limitations . . . . .	417
20.2 Progressive Data Access . . . . .	417
20.2.1 VAPOR Data Collection . . . . .	418
20.2.2 Multiresolution . . . . .	419
20.3 Visualization-Guided Analysis . . . . .	420
20.4 Progressive Access Examination . . . . .	422
20.4.1 Discussion . . . . .	423
20.5 Conclusion . . . . .	424
<b>21 The EnSight Visualization Application</b>	<b>429</b>
<i>Randall Frank and Michael F. Krogh</i>	
21.1 Introduction . . . . .	429
21.2 EnSight Architectural Overview . . . . .	430

21.3	Cluster Abstraction: CEIShell . . . . .	432
21.3.1	Virtual Clustering Via CEIShell Roles . . . . .	433
21.3.2	Application Invocation . . . . .	434
21.3.3	CEIShell Extensibility . . . . .	434
21.4	Advanced Rendering . . . . .	435
21.4.1	Customized Fragment Rendering . . . . .	435
21.4.2	Image Composition System . . . . .	438
21.5	Conclusion . . . . .	440
<b>Index</b>		<b>443</b>

