

Contents

1	Creating a Tool Set for Optimizing Topology-Aware Node Mappings	1
	Martin Schulz, Abhinav Bhatele, Peer-Timo Bremer, Todd Gamblin, Katherine Isaacs, Joshua A. Levine, and Valerio Pascucci	
1.1	Motivation and Background	1
1.2	Gaining Insight from Multiple Perspectives	3
1.2.1	The <i>HAC</i> Model	3
1.2.2	Domains Relevant for Node Mappings	4
1.2.3	Mapping Communication to Hardware Data	4
1.3	Creating a Flexible Measurement Environment	5
1.3.1	Concurrent Measurements Using P^N MPI	5
1.3.2	Gathering Data in the Hardware Domain	6
1.3.3	Gathering Data in the Communication Domain	6
1.3.4	Phase Attribution	6
1.3.5	Data Storage in Structured YAML Files	7
1.3.6	Investigating Best Case Mappings	7
1.3.7	Approximating Collectives	8
1.4	Preliminary Results	8
1.5	Related Work	10
1.6	Conclusions and Future Work	10
	References	12
2	Using Sampling to Understand Parallel Program Performance	13
	Nathan R. Tallent and John Mellor-Crummey	
2.1	Introduction	13
2.2	Call Path Profiling	15
2.3	Pinpointing Scaling Bottlenecks	16

2.4	Blame Shifting	17
2.4.1	Parallel Idleness and Overhead in Work Stealing	18
2.4.2	Lock Contention	19
2.4.3	Load Imbalance.....	20
2.5	Call Path Tracing	21
2.6	Data-Centric Performance Analysis	22
2.7	Conclusions.....	22
	References.....	23
3	likwid-bench: An Extensible Microbenchmarking Platform for x86 Multicore Compute Nodes.....	27
	Jan Treibig, Georg Hager, and Gerhard Wellein	
3.1	Introduction.....	27
3.2	Related Work	28
3.3	Architecture.....	29
3.4	Benchmark .ptt File Format.....	30
3.5	Command Line Syntax.....	31
3.6	Examples	32
3.6.1	Identifying Bandwidth Bottlenecks.....	32
3.6.2	Characterizing ccNUMA Properties	33
3.7	Conclusion and Outlook	35
	References.....	35
4	An Open-Source Tool-Chain for Performance Analysis	37
	Kevin Coulomb, Augustin Degomme, Mathieu Faverge, and François Trahay	
4.1	Introduction.....	38
4.2	Related Work	38
4.3	Instrumenting Applications with EZTRACE	39
4.3.1	Tracing the Execution of an Application	39
4.3.2	Instrumenting an Application	40
4.4	Creating Trace Files with GTG	41
4.4.1	Overview of GTG	41
4.4.2	Interaction Between GTG and EZTRACE	42
4.5	Analyzing Trace Files with VITE	42
4.5.1	A Generic Trace Visualizer	43
4.5.2	Displaying Millions of Events	43
4.6	Evaluation	45
4.6.1	Overhead of Trace Collection	45
4.6.2	NAS Parallel Benchmarks.....	46
4.7	Conclusion and Future Work	47
	References.....	47

5	Debugging CUDA Accelerated Parallel Applications with TotalView	49
	Chris Gottbrath and Royd Lüttke	
5.1	Introduction.....	49
5.1.1	HPC Debugging Challenges	50
5.1.2	CUDA and Heterogeneous Acceleration Architectures	50
5.1.3	Challenges Introduced by CUDA	51
5.1.4	The TotalView Debugger	52
5.2	TotalView for CUDA.....	53
5.2.1	Previous Experience: Cell	53
5.2.2	The NVIDIA GPU Architecture and CUDA	54
5.2.3	The TotalView Model: Extended for CUDA	55
5.2.4	Challenges and Features.....	56
6	Advanced Memory Checking Frameworks for MPI Parallel Applications in Open MPI	63
	Shiqing Fan, Rainer Keller, and Michael Resch	
6.1	Introduction.....	64
6.2	Overview of Debugging Tools	65
6.2.1	Valgrind	65
6.2.2	Intel Pin	66
6.3	Design and Implementation.....	67
6.3.1	Valgrind Extensions	68
6.3.2	MemPin	69
6.4	Memory Checks in Parallel Application	70
6.4.1	Pre-communication Checks	70
6.4.2	Post-communication Checking	72
6.5	Performance Implications.....	73
6.6	Detectable Error Classes and Findings from Actual Applications	75
6.7	Conclusion.....	77
	References.....	77
7	Score-P: A Joint Performance Measurement Run-Time Infrastructure for Periscope, Scalasca, TAU, and Vampir	79
	Andreas Knüpfer, Christian Rössel, Dieter an Mey, Scott Biersdorff, Kai Diethelm, Dominic Eschweiler, Markus Geimer, Michael Gerndt, Daniel Lorenz, Allen Malony, Wolfgang E. Nagel, Yury Oleynik, Peter Philippen, Pavel Saviankou, Dirk Schmidl, Sameer Shende, Ronny Tschüter, Michael Wagner, Bert Wesarg, and Felix Wolf	
7.1	Introduction.....	79
7.1.1	Motivation for a Joint Measurement Infrastructure	80
7.2	The SILC and PRIMA Projects.....	81
7.3	Score-P.....	83

7.4	The Open Trace Format Version 2	84
7.4.1	The SIONlib OTF2 Substrate	84
7.5	The CUBE4 Format and GUI	85
7.6	The OPARI2 Instrumentor for OpenMP	85
7.7	The On-Line Access Interface	86
7.8	Early Evaluation	87
7.8.1	Run-Time Measurement Overhead	87
7.8.2	Trace Format Memory Consumption	88
7.9	Conclusion and Outlook	89
	References	90
8	Trace-Based Performance Analysis for Hardware Accelerators	93
	Guido Juckeland	
8.1	Introduction	93
8.2	Related Work	94
8.3	Gathering Accelerator Related Performance Information	94
8.3.1	Accelerator APIs	95
8.3.2	Tracking Accelerator Events	97
8.3.3	Including Accelerator Specific Data	99
8.4	Example: Integration of CUDA and OpenCL into VampirTrace/Vampir	100
8.5	Summary and Future Work	103
	References	103
9	Folding: Detailed Analysis with Coarse Sampling	105
	Harald Servat, Germán Llort, Judit Giménez, Kevin Huck, and Jesús Labarta	
9.1	Introduction	105
9.2	Folding: Instrumentation and Sampling	106
9.2.1	The Hardware Counters	108
9.2.2	The Callstack	109
9.3	Example of Usage	110
9.4	Validation of the Results	113
9.5	Related Work	115
9.6	Conclusions and Future Directions	116
	References	117
10	Advances in the TAU Performance System	119
	Allen Malony, Sameer Shende, Wyatt Spear, Chee Wai Lee, and Scott Biersdorff	
10.1	Introduction	119
10.2	Instrumentation Of GPU Accelerated Code	120
10.2.1	Synchronous Method	120
10.2.2	Event Queue Method	121
10.2.3	Callback Method	122
10.2.4	TAU Performance System Implementation	122

10.3	Event Based Sampling (EBS).....	123
10.4	Automatic Wrapper Library Generation	124
10.5	3D Visualization.....	125
10.6	Eclipse Integration	127
10.7	Conclusion.....	129
	References.....	130
11	Temanejo: Debugging of Thread-Based Task-Parallel Programs in StarSS	131
	Rainer Keller, Steffen Brinkmann, José Gracia, and Christoph Niethammer	
11.1	Introduction.....	131
11.2	Implementation	133
11.3	Debugging Capabilities	134
11.4	Related Work	136
11.5	Outlook	136
	References.....	137
12	HiFlow³: A Hardware-Aware Parallel Finite Element Package	139
	H. Anzt, W. Augustin, M. Baumann, T. Gengenbach, T. Hahn, A. Helfrich-Schkarbanenko, V. Heuveline, E. Ketelaer, D. Lukarski, A. Nestler, S. Ritterbusch, S. Ronnas, M. Schick, M. Schmidtobreick, C. Subramanian, J.-P. Weiss, F. Wilhelm, and M. Wlotzka	
12.1	Introduction.....	139
12.2	Motivation, Concepts and Structure of HiFlow ³	140
12.2.1	Fields of Application	140
12.2.2	Flexibility	140
12.2.3	Performance, Parallelism, Emerging Technologies	141
12.2.4	Hardware-Aware Computing.....	141
12.3	Mesh Module	142
12.4	DoF/FEM Module.....	143
12.4.1	Submodules	143
12.4.2	Partitioning	145
12.5	Linear Algebra Module	145
12.5.1	Linear Solvers and Preconditioners.....	146
12.6	Example: Advection-Diffusion Equation	147
12.6.1	Numerical Results and Benchmarks	148
12.7	Conclusion.....	150
	References.....	150
	Author Index.....	153