

Contents

1	PERISCOPE: An Online-Based Distributed Performance Analysis Tool	1
	Shajulin Benedict, Ventsislav Petkov, and Michael Gerndt	
1.1	Introduction	1
1.2	Related Work	3
1.3	Periscope Design	4
1.3.1	Periscope Architecture	5
1.3.2	Agent's Data Capture Mechanism	6
1.3.3	Search Strategies	7
1.3.4	User Interface	9
1.4	Periscope Usage Model	10
1.4.1	Preparing Analysis Run	10
1.4.2	Starting an Analysis Run	11
1.5	Distributed Performance Analysis	11
1.5.1	Found Properties	12
1.5.2	Scalability of the Analysis	13
1.6	Conclusion and Outlook	14
	References	15
2	Comprehensive Performance Tracking with Vampir 7	17
	Holger Brunst, Daniel Hackenberg, Guido Juckeland, and Heide Rohling	
2.1	Introduction	17
2.2	Overview	18
2.3	Recent Developments in the Graphical User Interface	18
2.3.1	Custom Side by Side Chart Arrangement	18
2.3.2	Counter Data Timeline	20
2.3.3	Performance Markers	21
2.3.4	Clustering of Performance Data	22
2.4	New Performance Data Sources	24

2.4.1	Accelerators	24
2.4.2	Energy Consumption	27
2.5	Summary and Outlook	29
	References	29
3	Performance Analysis and Workload Characterization with IPM ...	31
	Karl F�rlinger, Nicholas J. Wright, and David Skinner	
3.1	Introduction	31
3.2	Overview	32
3.3	Performance Analysis with IPM	33
3.4	Example Scaling Study with IPM	35
3.5	Related Work	37
3.6	Conclusion and Outlook	37
	References	37
4	Recent Developments in the Scalasca Toolset.	39
	Markus Geimer, Felix Wolf, Brian J. N. Wylie, Daniel Becker, David B�hme, Wolfgang Frings, Marc-Andr� Hermanns, Bernd Mohr, and Zolt�n Szebenyi	
4.1	Introduction	40
4.2	Scalasca Overview	41
4.3	Analysis of Hybrid MPI/OpenMP Codes	41
4.4	Scalable Wait-State Analysis	44
4.4.1	Scalability	44
4.4.2	Improvement of Trace-Data I/O	45
4.4.3	Analysis of MPI-2 Remote Memory Access Operations ..	46
4.4.4	Delay Analysis	46
4.4.5	Evaluation of Optimization Hypotheses	46
4.4.6	Configurable Source-Code Instrumentation	47
4.5	Analysis of Time-Dependent Behavior	47
4.5.1	Observing Individual Iterations	47
4.5.2	Space-Efficient Time-Series Call-Path Profiling	49
4.6	Outlook	49
	References	50
5	MUST: A Scalable Approach to Runtime Error Detection in MPI Programs	53
	Tobias Hilbrich, Martin Schulz, Bronis R. de Supinski, and Matthias S. M�ller	
5.1	Introduction	53
5.2	Experiences from Marmot and Umpire	54
5.2.1	Marmot	55
5.2.2	Umpire	56
5.3	Introduction to MUST	57
5.4	MUST Design	59
5.4.1	Offloading of Checks	59

5.4.2	Major Components	60
5.4.3	Trace Communication System	61
5.5	Initial Experiments	62
5.6	Related Work	65
5.7	Conclusions	65
	References	66
6	HPC Profiling with the Sun Studio™ Performance Tools	67
	Marty Itzkowitz and Yukon Maruyama	
6.1	Introduction	67
6.1.1	The Sun Studio Performance Tools	67
6.1.2	The Sun Studio Performance Tools Usage Model	68
6.1.3	The Sun Studio Performance Tools Features	68
6.1.4	Diagnosing Performance Problems	69
6.2	Single-Threaded Application Performance Issues	69
6.2.1	Algorithmic Inefficiency	70
6.2.2	Memory Subsystem Performance Issues	73
6.3	Multi-threading Performance Issues	76
6.3.1	Lock Contention	76
6.3.2	False Sharing of Cache Lines	79
6.4	OpenMP Performance Issues	79
6.4.1	Excess Parallel Overhead	80
6.4.2	Insufficient Parallelism	81
6.4.3	Lock Contention	81
6.4.4	Load Imbalance	82
6.5	MPI Performance Issues	84
6.5.1	Computation Issues in MPI Programs	84
6.5.2	Parallelization Issues in MPI Programs	85
6.6	Conclusions	91
	References	92
7	Performance Tuning of x86 OpenMP Codes with MAQAO	95
	Denis Barthou, Andres Charif Rubial, William Jalby, Souad Kolaij, and Cédric Valensi	
7.1	Introduction	95
7.2	Static Performance Evaluation	96
7.2.1	Code Restructuring	97
7.2.2	Performance Model	99
7.2.3	Applying MAQAO to Real-World Applications	102
7.3	Memory Traces for OpenMP Codes	105
7.3.1	Static Binary Instrumentation	106
7.3.2	Memory Traces	107
7.3.3	Using Traces for OpenMP Performance Issues	108
7.4	Related Work	110
7.5	Conclusions and Future Work	111
	References	112

8	Scalable Parallel Debugging with g-Eclipse	115
	Thomas Köckerbauer, Christof Klausecker, and Dieter Kranzlmüller	
8.1	Introduction	115
8.2	Related Work	116
8.3	Debugging Parallel Programs Using g-Eclipse	117
8.4	Reduction of the Trace Complexity	118
8.5	Pattern Matching	120
8.6	Summary and Conclusions	122
	References	122
9	New Analysis Techniques in the CEPBA-Tools Environment	125
	Jesus Labarta	
9.1	Introduction	126
9.2	The CEPBA-Tools Environment	126
9.3	Spectral Analysis	128
9.4	Clustering Techniques	130
	9.4.1 Clustering Algorithms	130
	9.4.2 Application of Clustering Techniques	131
	9.4.3 Quantification of the Clustering Quality	133
9.5	Sampling and Mixed Instrumentation	136
	9.5.1 High Frequency Sampling	136
	9.5.2 Hybrid Instrumentation and Sampling	138
9.6	On-line Techniques	139
9.7	Conclusion	142
	References	142
10	The Importance of Run-Time Error Detection	145
	Glenn R. Luecke, James Coyle, James Hoekstra, Marina Kraeva, Ying Xu, Mi-Young Park, Elizabeth Kleiman, Olga Weiss, Andre Wehe, and Melissa Yahya	
10.1	Introduction	145
10.2	Background	147
10.3	Methodology	147
10.4	Results	149
10.5	Recommendations	153
10.6	Conclusions	153
	References	154
11	Collecting Performance Data with PAPI-C	157
	Dan Terpstra, Heike Jagode, Haihang You, and Jack Dongarra	
11.1	Introduction	157
11.2	Extending PAPI to Multiple Measurement Components	159
	11.2.1 Preset Events	160
	11.2.2 Native Events	161
	11.2.3 API Changes	162
	11.2.4 The CPU Component	163

11.2.5	Accessing the CPU Hardware Counters	164
11.2.6	The ACPI and MX Components	165
11.2.7	The LM-SENSORS Component	166
11.3	Multi-core Performance Measurement	167
11.3.1	Various Multi-core Designs	169
11.4	Future Directions	170
11.5	Conclusion	171
	References	172
12	ISP Tool Update: Scalable MPI Verification	175
	Anh Vo, Sarvani Vakkalanka, and Ganesh Gopalakrishnan	
12.1	Introduction	175
12.1.1	Determining and Enforcing Relevant Schedules	176
12.2	ISP Overview	177
12.3	GEM	180
12.4	DMA - A Distributed MPI Analyzer	181
12.4.1	Limitations of ISP	181
12.4.2	DMA	182
12.5	Conclusions	183
	References	184