

Contents

Part I Debugging

Debugging at Scale with Allinea DDT	3
David Lecomber and Patrick Wohlschlegel	
1 Why Scalability Matters for Debugging	3
2 The Ability to Debug at-Scale Changes Everything	4
3 How Allinea DDT Helps to Fix Bugs.....	5
4 Understanding Multiple Processes	5
5 Simple and Effective Process Control	6
6 Smart Highlighting and Sparklines	7
7 Searching Data Sets.....	8
8 Visualizing Large Data Sets in Real-Time.....	9
9 Deadlocks	10
10 Memory Debugging	11
11 Summary	11
12 Further Readings	12
Task Debugging with TEMANEJO	13
Steffen Brinkmann, José Gracia, and Christoph Niethammer	
1 Introduction	13
2 What Debugging Means in the Context of Task-Based Parallelism	15
3 The Debugging Process.....	16
3.1 Communication	16
3.2 Graph Display.....	17
3.3 Execution Control	20
4 Conclusion	20
References	21

Part II Automatic Error Detection

MPI Runtime Error Detection with MUST: Advanced Error Reports 25

Joachim Protze, Tobias Hilbrich, Bronis R. de Supinski,
Martin Schulz, Matthias S. Müller, and Wolfgang E. Nagel

1	Introduction	25
2	MUST	27
3	Shortcoming of Current Error Views	29
3.1	Example 1: Pinpointing Deadlocks	29
3.2	Example 2: Viewing Datatype Related Problems	31
4	Deadlock View in MUST	31
5	Type Tree View	33
6	Related Work	35
7	Conclusion	36
	References	37

Advanced Memory Checking for MPI Parallel Applications

Using MemPin 39

Shiqing Fan, Rainer Keller, and Michael Resch

1	Introduction	40
2	Overview of Intel Pin	41
3	Design and Implementation	42
3.1	MemPin	43
3.2	Integration of MemPin with Open MPI	44
4	Memory Checks in Parallel Application	46
4.1	Pre-communication Checks	46
4.2	Post-communication Checking	47
5	Performance Comparison	48
6	2D Heat Conduction Program with MemPin	49
7	Conclusion	51
	References	52

Part III Performance Analysis and Optimization

Generic Support for Remote Memory Access Operations in Score-P and OTF2..... 57

Andreas Knüpfer, Robert Dietrich, Jens Doleschal,
Markus Geimer, Marc-André Hermanns, Christian Rössel,
Ronny Tschüter, Bert Wesarg, and Felix Wolf

1	Introduction	57
2	Overview of Score-P and OTF2	58
2.1	The Score-P Instrumentation and Measurement System	59
2.2	The Open Trace Format 2	59
2.3	Existing Event Record Types	60
2.4	Current and Future Directions	60

3	RMA in HPC Parallel Programming	61
3.1	RMA Operations in HPC Parallelization Libraries	61
3.2	Concepts in RMA Parallelization Models	62
4	Generic RMA Event Types	64
4.1	RMA Window Handling	64
4.2	Specification of the Passive Side	65
4.3	Get and Put	65
4.4	Atomic RMA Operations	66
4.5	Completion Records	67
4.6	Notification via RMA	67
4.7	Synchronization	68
4.8	Collective Operations and Synchronization	69
4.9	Locking of Resources	69
5	Example Cases with RMA Event Types	70
6	Conclusions and Outlook	72
	References	73

Cache-Related Performance Analysis Using Rogue Wave

Software's ThreadSpotter	75
---------------------------------------	-----------

Royd Lüdtké and Chris Gottbrath

1	Introduction	75
2	Basic Overview on Caching	76
2.1	Motivation for Caching	76
2.2	Cache Architectures	76
2.3	Cache Organization	77
2.4	Prefetching	77
2.5	Eviction of Cache Lines (Replacement Policies)	77
2.6	Complexity Added by Coherence	78
2.7	Important Statistics	78
2.8	Optimal Cache Utilization	79
2.9	What Performance Improvement Is Possible When Optimizing an Application's Cache Utilization?	79
3	ThreadSpotter: A Statistical Approach for Cache-Related Profiling	80
3.1	Different Approaches of Cache Related Performance Analysis	80
3.2	What Kind of Data Is ThreadSpotter Looking at in Order to Create a Report?	81
3.3	Sampling an Application	81
3.4	Report Generation	83
3.5	Presenting Optimization Opportunities That ThreadSpotter Discovers	85
4	Types of Performance Optimization Opportunities Discovered by ThreadSpotter	89
4.1	What Kind of Cache-Related Opportunities for Performance Optimization Can Be Discovered by ThreadSpotter's Statistical Approach?	89
4.2	Reuse	91

4.3	Non-temporal Data	92
4.4	Cache Hot Spots	92
5	Conclusion	93
	References	93
Custom Hot Spot Analysis of HPC Software with the Vampir Performance Tool Suite		95
Holger Brunst and Matthias Weber		
1	Introduction	95
2	Heat Map Overlay for the Master Timeline	97
3	Customization of Performance Metrics	100
3.1	Delayed MPI Communication	101
3.2	Conditional Floating Point Performance	101
4	Refinement of Invocation Graph	102
4.1	Rules	104
4.2	Examples	105
5	Comparison of Multiple Program Runs	110
5.1	Multiple Program Executions at a Glance	111
5.2	Alignment of Multiple Trace Files	112
6	Conclusion	114
	References	114
Extending Scalasca's Analysis Features		115
Daniel Lorenz, David Böhme, Bernd Mohr, Alexandre Strube, and Zoltán Szebenyi		
1	Introduction	115
2	Root Cause Analysis	117
3	Critical Path Analysis	118
4	Time-Series Profiling	120
5	Topologies	122
5.1	Hardware Topologies	123
5.2	Processes X Threads	124
5.3	Runtime Mapping	124
5.4	Algorithm Domain	124
6	Future Work	125
	References	125
The HOPSA Workflow and Tools		127
Bernd Mohr, Vladimir Voevodin, Judit Giménez, Erik Hagersten, Andreas Knüpfer, Dmitry A. Nikitenko, Mats Nilsson, Harald Servat, Aamer Shah, Frank Winkler, Felix Wolf, and Ilya Zhukov		
1	Introduction	128
2	The HOPSA Workflow	129
2.1	Overview	129
2.2	Performance Screening	130

2.3	Performance Diagnosis	132
2.4	The HOPSA Performance Tools	136
2.5	Integration Among Performance Analysis Tools.....	141
2.6	Integration of System Data and Performance Analysis Tools	142
2.7	Opportunities for System Tuning	144
3	Conclusion	145
	References	145

Part IV Performance Data Visualization

	Visualizing More Performance Data Than What Fits on Your Screen	149
	Lucas M. Schnorr and Arnaud Legrand	
1	Introduction	149
2	Motivation and Discussion	151
3	Multi-scale Trace Aggregation for Visualization	153
4	Visualization Techniques	155
4.1	Squarified Treemap View	155
4.2	Hierarchical Graph View	157
5	The Viva Visualization Tool	159
6	Conclusion	159
	References	160

Tools for High Performance Computing 2012

Cheptsov, A.; Brinkmann, S.; Gracia, J.; Resch, M.;
Nagel, W.E. (Eds.)

2013, XI, 162 p. 70 illus. in color., Hardcover

ISBN: 978-3-642-37348-0