



UvA-DARE (Digital Academic Repository)

Distributed Event-driven Simulation- Scheduling Strategies and Resource Management

Overeinder, B.J.

Publication date
2000

[Link to publication](#)

Citation for published version (APA):

Overeinder, B. J. (2000). *Distributed Event-driven Simulation- Scheduling Strategies and Resource Management*. [Thesis, fully internal, Universiteit van Amsterdam]. University of Amsterdam.

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Contents

1	Introduction	1
1.1	Rationale	1
1.2	Modeling and Simulation	3
1.2.1	Systems and System Environment	3
1.2.2	Components of a System	3
1.2.3	Model of a System	4
1.2.4	Experimentation and Simulation	4
1.2.5	A Closer Look at System Models	6
1.2.6	Model Execution: Time-Driven versus Event-Driven	7
1.2.7	World Views in Discrete Event Simulation	8
1.3	Parallel Computing	9
1.3.1	Parallel Architectures	9
1.3.2	Resource Management: Scheduling and Load Balancing	11
1.4	Problems and Challenges	13
1.5	Outline of Thesis	14
I	Scheduling Strategies	17
2	Issues in Parallel Discrete Event Simulation	19
2.1	Introduction	19
2.2	Basic Concepts	21
2.2.1	Need for Logical Processes	21
2.2.2	The Curse of Causality	22
2.3	Conservative Methods	24
2.3.1	Deadlock Avoidance	25
2.3.2	Deadlock Detection and Recovery	26
2.3.3	Performance of Conservative Methods	27
2.4	Optimistic Methods	28
2.4.1	Virtual Time	29
2.4.2	The Basic Time Warp Mechanism	30
2.4.3	Rollback Strategies	32
2.4.4	State Saving	34
2.4.5	Optimism Control	37
2.4.6	Global Virtual Time Algorithms	41
2.5	Summary and Discussion	44

3	The APSIS Time Warp Kernel	49
3.1	Introduction	49
3.2	Parallel Discrete Event Simulation Environments	50
3.2.1	Languages	51
3.2.2	Libraries	53
3.3	Design of the APSIS Environment	55
3.3.1	Requirements and Design Goals	55
3.3.2	Overview	56
3.3.3	The Application Programming Interface	57
3.3.4	The Software Architecture	62
3.4	Extensions to the Time Warp Kernel	65
3.4.1	Event Retraction	66
3.4.2	Incremental State Saving	67
3.5	Implementation Aspects of the Time Warp Simulation Kernel	68
3.5.1	Simulation Kernel and Data Structures	68
3.5.2	Synchronization	70
3.5.3	Fossil Collection and Irrevocable Events	72
3.5.4	The Global Virtual Time Computation	73
3.6	Summary and Discussion	74
4	APSE: Average Parallelism, Profile, and Shape Evaluation	77
4.1	Introduction	77
4.2	Characterization of Parallelism in Applications	78
4.2.1	The Average Parallelism Metric	79
4.2.2	The Space-Time Model	80
4.2.3	Critical Path Analysis	83
4.3	Design and Implementation of APSE	84
4.3.1	Conceptual Tool Structure	84
4.3.2	Overview of APSE	85
4.4	Experiments, Validation, and Assessment	91
4.4.1	Unidirectional Ring	91
4.4.2	Bidirectional Ring	94
4.5	Related Work	97
4.6	Summary and Discussion	99
5	Parallel Asynchronous Cellular Automata	103
5.1	Introduction	103
5.2	Asynchronous Cellular Automata	104
5.2.1	Cellular Automata	104
5.2.2	Asynchronous Cellular Automata	105
5.2.3	The Asynchronous Cellular Automata Model	106
5.2.4	Parallel Simulation of Cellular Automata Models	107
5.3	Ising Spin Systems	109
5.3.1	The Ising Spin Model	109
5.3.2	The Dynamics in the Ising Spin Model	111
5.4	Optimistic Simulation of Continuous-Time Ising Spin Systems	114

5.5	Parallel Performance and Scalability	118
5.5.1	Relative Parallel Performance and Scalability	118
5.5.2	Absolute Parallel Performance and Scalability	126
5.6	Summary and Discussion	130
6	Self-Organized Critical Behavior in Time Warp	133
6.1	Self-Organized Criticality	133
6.2	Self-Organized Criticality in Time Warp Dynamics	135
6.2.1	Slowly Driven, Interaction-Dominated Threshold Systems	135
6.2.2	Physical and Computational Critical Behavior	137
6.3	A First Indication of Self-Organized Criticality in Time Warp . .	138
6.4	Finite-Size Scaling Effects	142
6.4.1	Influence of lattice size	142
6.4.2	Varying the Number of Processors	143
6.4.3	Different Virtual Time Window Sizes	147
6.5	Summary and Discussion	149
II	Resource Management	151
7	Dynamic Load Balancing of Execution Threads	153
7.1	Introduction	153
7.2	Background and Design Aspects	155
7.2.1	Trends in Hardware	156
7.2.2	Trends in Software	157
7.3	The Polder Metacomputer Experimental Framework	159
7.3.1	Resource Management in the Polder Metacomputer	160
7.3.2	The Curse of Dynamics	161
7.4	Dynamite: Process Migration in Message Passing Environments	162
7.4.1	The PVM System	164
7.4.2	Design Aspects of Process Migration in Dynamite	165
7.5	Implementation Aspects of the Dynamite Environment	166
7.5.1	The Scheduler	166
7.5.2	Consistent Checkpointing Through Critical Sections . . .	168
7.5.3	The Migration Protocol	169
7.5.4	Packet Routing and Direct Connections	171
7.6	Performance Evaluation	172
7.6.1	Measuring DPVM Communication Overhead	173
7.6.2	Checkpoint and Migration Overhead	175
7.6.3	NAS Parallel Benchmarks	177
7.6.4	The GRAIL Finite-Element Model Simulation	180
7.7	Summary and Discussion	184
8	Summary and Conclusions	187
	Bibliography	191

Publications	209
Dutch Summary/Nederlandse Samenvatting	213
Nawoord	217
Index	219