

TELECOM PARISTECH
ÉCOLE DOCTORALE STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE

pour obtenir le titre de

Docteur en Sciences

de TELECOM ParisTech

Mention Informatique

présentée et soutenue par

Bilel BEN ROMDHANNE

Simulation des Réseaux à grande Échelle sur les architectures de calculs hétérogènes

Thèse dirigée par Navid NIKAEIN et Christian BONNET

Laboratoire EURECOM, Sophia Antipolis

soutenue le 16 Decembre 2013, devant le jury composé de:

<i>Président du Jury</i>		, France
<i>Rapporteurs</i>	--	, France
	--	, France
<i>Examineurs</i>	--	, France
	--	, France
	--	
<i>Directeur de thèse</i>	Navid NIKAEIN	Maître de Conférences, EURECOM Antipolis, France

© 2013
Ben Romdhanne Bilel
ALL RIGHTS RESERVED

Contents

1 Introduction	1
1.1 Motivation and Objectives	1
1.2 Contributions Storyline	2
1.3 Thesis Structure	4
1.4 Publications	5
1.5 Project Deliverables	6
1 Background	7
2 Network Experimentation	9
2.1 Introduction	9
2.2 Network Experimentation Tools	9
2.2.1 Real world Field	9
2.2.2 Real world Testbed	10
2.2.3 Emulation Testbed	11
2.2.4 Simulation Testbed	11
2.2.5 Hybrid Testbed	12
2.3 Network Experimentation Aspects	13
3 Discrete Event Simulation	17
3.1 Introduction	17
3.2 Discrete-Event Simulation	17
3.2.1 Terminology and Components	17
3.2.2 The Principle	18
3.3 Parallel Discrete Event Simulation	19
3.3.1 Discrete Event Simulation Limits	19
3.3.2 Principals of Parallel Discrete Event Simulation	19
3.3.3 Parallel Simulation Model and Algorithms	20
4 Hardware Trends	23
4.1 Introduction	23
4.2 Evolution of Computing Chips	23
4.2.1 CPU: Historical Evolution and Trends	23
4.2.2 GPU: Historical Evolution and Trends	24
4.2.3 Emerging Solutions	25
4.2.4 Multi-Core accelerator	26
4.3 Parallel Programming: Models and API [70]	26
4.3.1 Pthreads	26
4.3.2 OpenMP	26

4.3.3 MPI	27
4.3.4 CUDA	27
4.3.5 OpenCL	27
5 Related Work	29
5.1 Introduction	29
5.2 Large Scale Simulation	29
5.3 PDES Issues	31
5.3.1 Data Representation	31
5.3.2 Event Scheduling	32
5.4 Heterogeneous Computing Considerations	33
II Contributions	37
6 Cunetsim: An Experimentation Framework to Discover Scalability	39
Horizons	39
6.1 Introduction	39
6.2 Fundamental Concepts	40
6.2.1 The Worker Pool	41
6.2.2 Separation Between the Event and Its Description	41
6.2.3 Massive Parallel Events Generation	41
6.3 Cunetsim Software Architectures	42
6.3.1 The Worker Design	42
6.3.2 The Master Design	44
6.3.3 Legacy Architecture for Multi-Core CPU	45
6.4 Comparative Performances Results	46
6.4.1 Simulation Runtime	48
6.5 Technical Challenges of GPU-based Simulation	51
6.5.1 Synchronization Challenge	51
6.5.2 Memory management Challenge	53
6.5.3 Precision Issue	54
6.6 Configuration Issues of GPU-Oriented Simulation	54
6.6.1 Space Representation and partitioning	54
6.6.2 Tuning Parameters: Block Size as a Study Case	54
6.7 Conclusion	56
7 Hybrid Events Scheduler	57
7.1 Introduction	57
7.2 The Hybrid Scheduler	59
7.2.1 Model and Components	59
7.2.2 Scheduling Algorithms	60
7.3 Performance Evaluation	65
7.3.1 Scenario & Setup	66
7.3.2 Comparative Evaluation	67

7.3.3 Performance Analysis	68
7.4 Related Work	72
7.5 Discussion	73
7.6 Conclusion	74
8 General Purpose Coordinator-Master-Worker Model	77
8.1 Introduction	77
8.2 The General Purpose Coordinator-Master-Worker Mode	79
8.2.1 Events Management: Description, Scheduling and Execution .	81
8.2.2 The Synchronization Mechanism of the GP-CMW Mode	86
8.2.3 GP-CMW Communication Mode	87
8.3 Comparative Evaluation	89
8.3.1 Comparative Performance Evaluation	90
8.3.2 Inherent Performance Evaluation	92
8.4 Related Work	100
8.5 Conclusion	102
9 Study Case of PADS Methodology Deployment: NS-3	105
9.1 Introduction	105
9.2 Overview	106
9.3 Events scheduling on NS-3	107
9.4 NS-3 Events scheduler extensions	109
9.4.1 The Explicit CPU Parallelism	110
9.4.2 The Implicit CPU Parallelism	111
9.4.3 The GPU Offloading	112
9.4.4 The Co-scheduler Approach	113
9.5 Comparative evaluation	115
9.5.1 Medium Load	115
9.5.2 High Load	116
9.6 Conclusion	118
III Conclusion	121
10 Conclusion	123
A Experimentation Methodology	125
A.1 Introduction	125
A.2 Scientific Experimentation	126
A.3 OpenAirInterface Experimentation methodology	126
A.3.1 OpenAirInterface Formal Experimentation Methodology	127
A.3.2 Methodology Implementation	128
A.4 Conclusion	129
Bibliography	131

List of Figures

2.1 A survey of the most popular experimentation frameworks	15
6.1 Mobility Events pattern	43
6.2 Simplified master/workers model that targets GPU execution.	45
6.3 Simplified master/workers model that targets CPU execution.	46
6.4 Simple Grid Topology	47
6.5 End-to-End Packet Loss.	48
6.6 Simulation runtime of the static network.	49
6.7 Simulation runtime of the mobile network.	50
6.8 Block size impact	55
7.1 Events reordering and storage: top figure schematizes an events de- pendency diagram while bottom figure schematizes how they will be reordered and stored. Events dependency is transformed to interval.	60
7.2 Event scheduler model	61
7.3 Statistical table used for recalibration	65
7.4 Topology of the benchmarking scenario	70
7.5 Normalized speedup with respect to the sequential runtime	70
7.6 The hardware usage rate.	70
7.7 The scheduling cost.	71
7.8 Variation of the input rate vs Time	71
7.9 Output event rate of different algorithms.	71
7.10 Average decision path length during the simulation.	71
7.11 Average interval length during the simulation. It closely reflects the events dependency.	72
8.1 The concept of common addressing space:	80
8.2 The GP-CMW blocks diagram:	81
8.3 Events Life Cycle	83
8.4 Examples of events grouping:	84
8.5 The simulation runtime of the studied simulators:	91
8.6 The synchronization delay as a function of the number of LPs	92
8.7 The impact of the PAL and HAL on the simulation runtime:	95
8.8 The synchronization delay as a function of the scenario and the con- figuration	96
8.9 The average external communication latency:	97
8.10 The average internal communication latency:	98
8.11 The average CPU usage rate:	99
8.12 The average GPU usage rate:	99
8.13 The average RAM usage rate:	99

8.14 The average GRAM usage rate:	100
9.1 Default NS-3events scheduling approach:	107
9.2 Distributed event scheduling approach in NS3:	108
9.3 Scheduling Cost as a function of nodes number	110
9.4 Explicit CPU parallelism:	111
9.5 Implicit CPU parallelism:	112
9.6 GPU offloading:	113
9.7 CO-scheduling approach:	114
9.8 default one	117
9.9 default one	119
A.1 Experimentation Workflow	127
A.2 The user experimentation workflow:	129