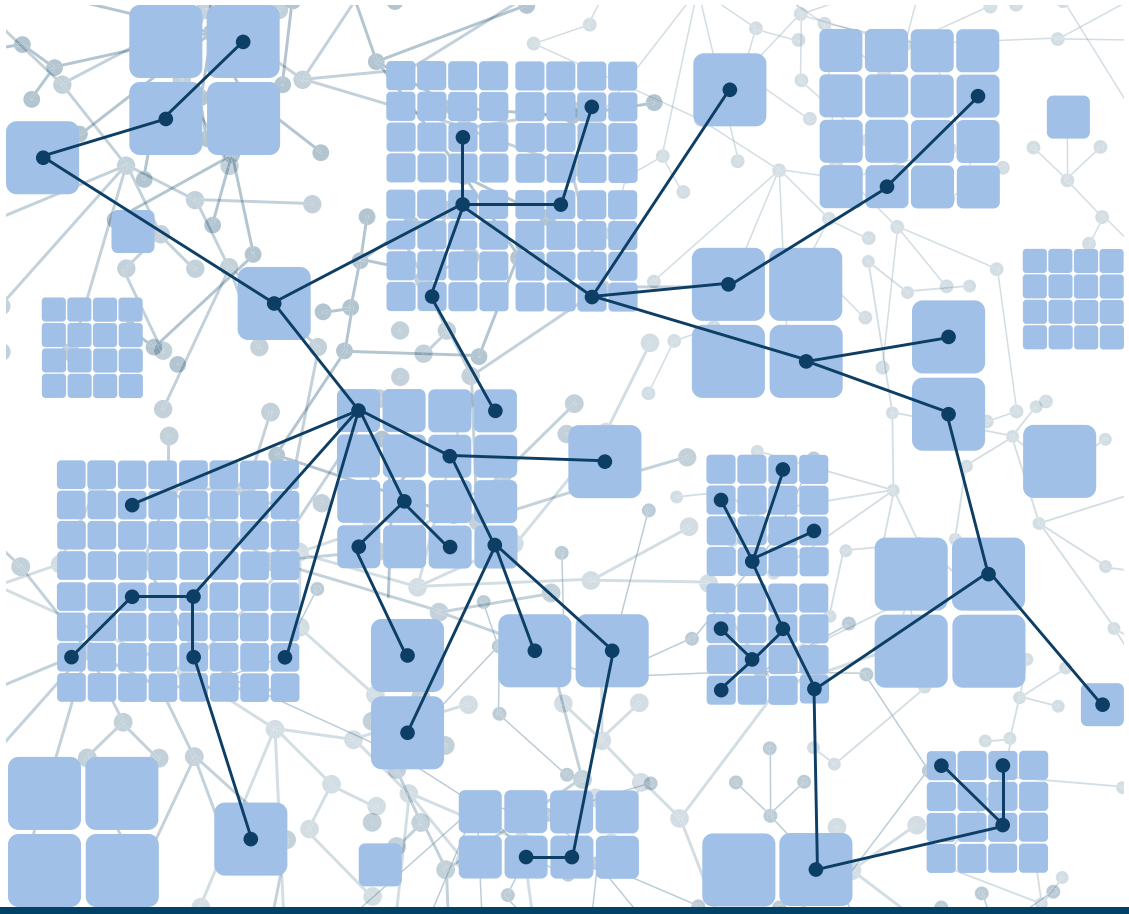

Eventify Meets Heterogeneity: Enabling Fine-Grained Task-Parallelism on GPUs

Laura Morgenstern

IAS Series
Band / Volume 63
ISBN 978-3-95806-765-3

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum

Forschungszentrum Jülich GmbH
Institute for Advanced Simulation (IAS)
Jülich Supercomputing Centre (JSC)

Eventify Meets Heterogeneity: Enabling Fine-Grained Task-Parallelism on GPUs

Laura Morgenstern

Schriften des Forschungszentrums Jülich
IAS Series

Band / Volume 63

ISSN 1868-8489

ISBN 978-3-95806-765-3

Contents

1 Introduction . . . 1
1.1 Motivation . . . 1
1.2 Research Objective . . . 2
1.2.1 Heterogeneous System . . . 3
1.2.2 Sustainability . . . 3
1.2.3 Scalability . . . 3
1.2.4 Event-Based Task-Parallelism . . . 3
1.2.5 Fast Multipole Method for Molecular Dynamics . . . 4
1.3 Solution Strategy . . . 4

2 Literature Review . . . 7
2.1 State of the Art . . . 7
2.2 Related Work . . . 7
2.2.1 Notions of Concurrency and Parallelism . . . 7
2.2.2 Classification of Parallel Architectures and Programming Models . . . 8
2.2.3 CPU-Managed Task-Parallelism on Heterogeneous Hardware . . . 8
2.2.4 GPU-Managed Task-Parallelism . . . 10
2.2.5 Parallel Fast Multipole Methods . . . 11

3 Concepts of Concurrency . . . 15
3.1 Parallelism and Concurrency . . . 15
3.2 General Steps of Parallelization . . . 16
3.2.1 Decomposition . . . 17
3.2.2 Assignment . . . 18
3.2.3 Orchestration . . . 18
3.2.4 Mapping . . . 19
3.3 Stream Interaction Model . . . 19

4 Computer Architectures . . . 23
4.1 Architecture Models . . . 23
4.1.1 SISD . . . 25
4.1.2 SIMD . . . 26
4.1.3 MISD . . . 26
4.1.4 MIMD . . . 26
4.2 CPU Architecture . . . 28
4.3 GPU Architecture . . . 28
4.4 CPU vs. GPU Architecture . . . 30
4.4.1 Optimization Goals . . . 31
4.4.2 SIMD Capabilities . . . 31
4.4.3 MISD Capabilities . . . 32
4.4.4 MIMD Capabilities . . . 32
4.5 Quantitative Architecture Trends . . . 33
4.5.1 Methods . . . 33
4.5.2 Clock Frequency Trend . . . 33
4.5.3 Compute Unit Trend . . . 34
4.5.4 Processing Element Trend . . . 34

5 Parallel Programming Models . . . 37
5.1 Classification . . . 37

5.2 OpenMP . . . 38
5.2.1 Architecture Model . . . 38
5.2.2 Memory Model . . . 38
5.2.3 Execution Model . . . 38
5.2.4 Partitioning Models . . . 39
5.3 Eventify . . . 39
5.3.1 Architecture Model . . . 39
5.3.2 Memory Model . . . 40
5.3.3 Execution Model . . . 40
5.3.4 Partitioning Models . . . 40
5.4 CUDA . . . 42
5.4.1 Architecture Model . . . 42
5.4.2 Memory Model . . . 43
5.4.3 Execution Model . . . 43
5.4.4 Partitioning Models . . . 44
5.5 OpenACC . . . 45
5.5.1 Architecture Model . . . 45
5.5.2 Memory Model . . . 45
5.5.3 Execution Model . . . 46
5.5.4 Partitioning Models . . . 46
5.6 Programming Model Trends . . . 47
5.6.1 Enhancing Flexibility . . . 47
5.6.2 Enhancing Uniformity . . . 48
5.6.3 Conclusion . . . 48

6 Event-based Task Parallelism on GPUs . . . 49
6.1 Uniform Architecture Model . . . 49
6.2 Uniform Execution Model . . . 50
6.2.1 Persistent Threads . . . 51
6.2.2 Thread Safety . . . 51
6.3 Eventify Execution Model on GPUs . . . 53
6.3.1 Comparison of CPU and GPU Queueing Principles . . . 54
6.3.2 Data Structure . . . 54
6.3.3 Taxonomy . . . 54
6.3.4 Queueing Schemes . . . 55
6.4 Implementation . . . 60
6.4.1 Software Architecture . . . 60
6.4.2 Workflow . . . 60

7 Use Case: Fast Multipole Method . . . 63
7.1 *N*-Body Problem of Electrostatics . . . 63
7.2 Sequential Fast Multipole Method . . . 64
7.2.1 Input Parameters . . . 64
7.2.2 Hierarchical Space Decomposition . . . 64
7.2.3 Workflow . . . 64
7.2.4 Assumptions . . . 65
7.3 Parallel Fast Multipole Method . . . 65
7.3.1 Data Dependency Graph . . . 66
7.3.2 Data-Parallel FMM . . . 67
7.3.3 Task-Parallel FMM . . . 68
7.4 FMSolvr . . . 68
7.4.1 Data-Parallel Implementations . . . 69
7.4.2 Task-Parallel Implementation . . . 73

8 Evaluation . . . 77
8.1 Metrics . . . 77
8.1.1 Scalability . . . 77
8.1.2 Sustainability . . . 78
8.2 Methodology . . . 78
8.2.1 Hardware . . . 78
8.2.2 CPU Runtime Measurements . . . 79
8.2.3 GPU Runtime Measurements . . . 80
8.3 Performance Analysis on CPUs: Data-Parallelism vs. Task-Parallelism . . . 80
8.4 Performance Analysis on GPUs: Data-Parallelism vs. Task-Parallelism . . . 81
8.4.1 OpenACC . . . 81
8.4.2 Eventify . . . 81
8.4.3 Sustainability . . . 86
8.5 Threats to Validity . . . 87

9 Conclusion . . . 89
9.1 Summary . . . 89
9.1.1 Hardware Architecture Trends . . . 89
9.1.2 Parallel Programming Model Trends . . . 89
9.1.3 Task-Parallelism vs. Data-Parallelism on CPUs . . . 89
9.1.4 Event-Based Task-Parallelism on GPUs . . . 89
9.1.5 Task-Parallelism vs. Data-Parallelism on GPUs . . . 90
9.2 Future Work . . . 90
9.2.1 Stream Interaction Model . . . 90
9.2.2 Eventify . . . 90

10 Theses . . . 93

A GPU Terminology Mapping . . . 97

B Evaluation of Additional Task Graph Sizes . . . 99

Bibliography . . . 103

IAS Series
Band / Volume 63
ISBN 978-3-95806-765-3

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum