



UvA-DARE (Digital Academic Repository)

Distributed multiscale computing

Borgdorff, J.

Publication date
2014

[Link to publication](#)

Citation for published version (APA):
Borgdorff, J. (2014). *Distributed multiscale computing*.

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	I
1.1	Multiscale applications	3
1.2	Theory of multiscale modelling and computing	7
1.3	Multiscale computing	12
1.4	Distributed multiscale computing	16
1.5	Outline of this thesis	17
2	Theory of distributed multiscale computing	19
2.1	Introduction	20
2.1.1	Related work	22
2.2	Multiscale modelling formalisation	23
2.2.1	Process	25
2.2.2	Scale separation	28
2.2.3	Multiscale model and single scale models	30
2.2.4	Scale separation map	32
2.2.5	Submodel Execution Loop	33
2.2.6	Coupling templates	37
2.2.7	Coupling topology	39
2.3	Specifying a multiscale model	41
2.3.1	Computational elements in MML	42
2.3.2	Graphical MML	44
2.3.3	XML format of MML	46
2.4	Multiscale model analysis	49
2.4.1	Coupling topology: deduction and implications	50
2.4.2	Predicting runtime behaviour with a task graph	50
2.4.3	Task graph reduction	57
2.5	Distributed multiscale execution	60

2.6	Applying the multiscale concepts	61
2.7	Conclusions and further work	62
3	Distributed multiscale runtime environment	67
3.1	Introduction	68
3.2	Design	69
3.2.1	Conceptual background	71
3.2.2	Library	72
3.2.3	Configuration	73
3.2.4	Runtime environment	74
3.2.5	Cross-cluster computing	75
3.3	Performance	77
3.3.1	Overhead	77
3.3.2	Message speed	80
3.4	Use cases	84
3.4.1	Hydrology application	84
3.5	Conclusions	88
4	Applying distributed multiscale computing to in-stent restenosis	91
4.1	Introduction	92
4.2	Multiscale modelling language (MML)	94
4.3	Software	95
4.3.1	High level composition and execution tools	95
4.3.2	MUSCLE	96
4.3.3	Cross-cluster execution with QosCosGrid	98
4.4	A three-dimensional model of in-stent restenosis	99
4.5	Results	103
4.6	Conclusions and discussion	106
5	Performance of distributed multiscale computing	109
5.1	Introduction	110
5.2	Multiscale Modelling and Simulation Framework	111
5.3	Performance context	112
5.4	Results	114
5.4.1	Tied multiscale computing	115
5.4.2	Scalable multiscale computing	117
5.4.3	Skewed multiscale computing	119

5.4.4	Common benefits	121
5.5	Conclusions	122
6	Conclusions and discussion	125
6.1	Limitations	128
6.2	Directions for further research	130
	Appendices	151
A	An algorithm for a temporal conduit filter	152
B	Technical details of the MUSCLE 2 runtime environment	155
B.1	Implementation of the MUSCLE Transport Overlay (MTO)	156
B.2	QosCosGrid and MUSCLE 2 integration	156
B.3	Comparison between MUSCLE 1 and MUSCLE 2	157
C	Application and model details	158
C.1	Tokamak plasma (TTE, fusion)	158
C.2	Cerebrovascular blood flow (HemeLB, biomedicine)	159
C.3	Irrigation network (Canals, hydrology)	160
C.4	Reverse-engineering of gene-regulatory networks (MultiGrain, systems biology)	160
C.5	Clay-polymer nanocomposites (Nano, nanomaterials)	161
C.6	In-stent restenosis (ISR ₃ D, biomedicine)	162
D	Software resulting from this thesis	163
D.1	xMML	163
D.2	jMML	163
D.3	MUSCLE	165
D.4	ISR ₃ D	165
	Summary	166
	Samenvatting	167
	List of publications	169
	Acknowledgements	172