

# Contents

<b>1</b>	<b>Introduction</b> .....	1
1.1	Challenges of Grid Programming .....	2
1.2	The Role of Middleware for the Grid .....	5
1.3	Communication Technologies for Distributed Computing .....	7
1.3.1	Java Remote Method Invocation (RMI) .....	7
1.3.2	Common Object Request Broker Architecture (CORBA) ...	9
1.3.3	Containers for Components & Services .....	10
1.4	Shortcomings in State-Of-The-Art Grid Middleware .....	16
1.4.1	Responsibilities of the Middleware User .....	16
1.4.2	Requirements for the Software Components .....	17
<b>2</b>	<b>HOCs: Software Components for Grid Programming</b> .....	19
2.1	Higher-Order Components (HOCs) .....	20
2.1.1	Motivation for HOCs .....	21
2.1.2	Grid Programming Using HOCs .....	22
2.1.3	Introducing Code Mobility to the Middleware .....	24
2.1.4	Polymorphism and Type Checking for Code Parameters ...	26
2.1.5	First Application Case Study: Julia Sets .....	30
2.2	HOCs and Grid Middleware .....	31
2.2.1	An Analysis of the Requirements of the Grid Platform without Components .....	31
2.2.2	Bridging Middleware and Application with HOCs .....	36
2.2.3	Case Study Revisited: Using the Farm-HOC .....	38
2.2.4	Performance Experiments on a Wide-Area Testbed .....	40
2.2.5	HOCs and Hand-Written Code: A Performance Comparison	41
2.3	APIs for Grid Application Programming with HOCs .....	42
2.4	Adaptability of HOCs .....	51
2.4.1	Code Parameters for Adaptation .....	52
2.4.2	Case Study: From Farm to Wavefront .....	53
2.5	Discussion: Adaptation vs. AOP .....	60

<b>3</b>	<b>Higher-Order Component Service Architecture (HOC-SA)</b>	63
3.1	Service-Oriented Grid Programming Using the HOC-SA	64
3.1.1	How Code Mobility Works: HOC-SA Code Service & Remote Code Loader	66
3.1.2	Parameter Databases in the HOC-SA	69
3.2	HOCs and Web Services	71
3.2.1	Web Services	71
3.2.2	Components and Resources	73
3.2.3	The HOC-SA Component Repository	74
3.2.4	The HOC-SA Portal	75
3.3	A Comparison of the HOC-SA and Globus WS-GRAM	78
3.3.1	Grid Programming with WS-GRAM and the HOC-SA	79
3.3.2	Application Types for HOC-SA and WS-GRAM	80
3.3.3	Response Times: HOC-SA vs. WS-GRAM	84
3.4	MPI, Skeletons and Web Services: Integrating Grid Technologies	87
3.4.1	A Gateway for Bridging between Web Services and MPI	88
3.4.2	Example: Discrete Wavelet Transform (DWT)	90
3.4.3	Wavelet Transform in General	90
3.4.4	DWT for Image Processing	92
3.4.5	DWT on the Grid Using the Lifting-HOC	93
3.4.6	Portable Parameters for the Lifting-HOC	96
3.4.7	An Adaptation of the Lifting-HOC	97
3.4.8	Experimental Performance Evaluation	99
3.4.9	Discussion: Interoperability and Portable Code	100
3.5	A HOC-SA Based Map/Reduce Implementation	103
3.5.1	Cloud Computing Technologies for the HOC-SA	103
3.5.2	MapReduce and Hadoop	105
3.5.3	HOC-SA Features for Map/Reduce on the Grid	105
3.6	Summary of HOC-SA Features	107
<b>4</b>	<b>Applications of Higher-Order Components</b>	109
4.1	Clayworks: A Collaborative Simulation Environment	110
4.1.1	The 3-tier Architecture of Clayworks	112
4.1.2	The Deformation-HOC for Parallel Simulations	115
4.2	Protein Sequence Analysis with HOCs	121
4.2.1	The Alignment Problem in Bioinformatics	121
4.2.2	Circular Permutations of DNA	122
4.2.3	The Alignment-HOC and its Code Parameters	123
4.2.4	Using an Alternative Traceback	126
4.2.5	Optimizations of the Alignment-HOC	126
4.2.6	Experiments with the Alignment-HOC	128
4.3	Conclusions from Using HOCs in Large-Scale applications	129

<b>5</b>	<b>HOCs with Embedded Scheduling and Loop Parallelization</b>	131
5.1	User-Transparent Grid Scheduling	132
5.1.1	The KOALA Grid Scheduling Infrastructure	133
5.1.2	Extensions of KOALA for User-Transparent Scheduling	135
5.1.3	Integrating KOALA & HOC-SA	136
5.1.4	A HOC-Aware Scheduling Algorithm	138
5.1.5	HOC Scheduling Cost-Functions	138
5.1.6	Scheduling Large-Scale Applications	140
5.1.7	Experiments with HOCs and KOALA	141
5.1.8	Conclusions from the Scheduling Experiments	144
5.2	Parallelization of Code Parameters in HOCs	144
5.2.1	The Internal Compute Farm of the LooPo-HOC	145
5.2.2	Transforming Loop Nests into Task Graphs	146
5.2.3	Integrating Loop Parallelization with the Grid	148
5.2.4	Case Study: The SOR Equation System Solver	150
5.2.5	Experiments	152
5.3	Combining HOCs with Related technologies: ProActive, SOFA and the GCM	153
5.3.1	Combining HOCs with ProActive, the GCM and SOFA	155
5.3.2	Creation of Web Services Using ProActive	158
5.4	Discussion: HOCs and Different Tools for Distributed Computing	159
<b>6</b>	<b>Conclusions and Related Work</b>	161
6.1	New Contributions	161
6.2	Related Work	162
6.2.1	The Real-Time Framework (RTF)	163
6.2.2	A Survey of Related Component Models	164
6.2.3	The Skeleton Model	164
6.2.4	CCA: The Common Component Architecture	165
6.2.5	CCM: The CORBA Component Model	166
6.2.6	Java Servlets and JSPs	166
6.2.7	Enterprise Java Beans and .NET Components	166
6.2.8	The Web 2.0	168
6.2.9	The Semantic Web	168
6.3	Future Work	169
	<b>Bibliography</b>	171
	References	171
	(a) Textbooks	171
	(b) Research Papers	172
	(c) Online References	177
	<b>Index</b>	181