



Architecture of Advanced Numerical Analysis Systems

Designing a Scientific Computing System
using OCaml

—
Liang Wang
Jianxin Zhao

Apress
open

Architecture of Advanced Numerical Analysis Systems

Designing a Scientific Computing
System using OCaml

Liang Wang
Jianxin Zhao

apress
open

Architecture of Advanced Numerical Analysis Systems: Designing a Scientific Computing System using OCaml

Liang Wang
Helsinki, Finland

Jianxin Zhao
Beijing, China

ISBN-13 (pbk): 978-1-4842-8852-8
<https://doi.org/10.1007/978-1-4842-8853-5>

ISBN-13 (electronic): 978-1-4842-8853-5

Copyright © 2023 by Liang Wang, Jianxin Zhao

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.



Open Access This book is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this book are included in the book's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the book's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Steve Anglin
Development Editor: James Markham
Coordinating Editor: Mark Powers

Cover designed by eStudioCalamar

Cover image by Lukasz Niescioruk on Unsplash (www.unsplash.com)

Distributed to the book trade worldwide by Apress Media, LLC, 1 New York Plaza, New York, NY 10004, U.S.A. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub (<https://github.com/Apress>). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*To my wife Maria, our daughters Matilda and Sofia,
and my beloved family.*

—Liang

*To my parents and sister; to their unyielding love and support.
To all those who are fighting for freedom and righteousness against the
unleashing evil from hell.*

*“Namárië, ar nai aistalë Eldar ar Atani ar ilyë Lérialieron hilya le. Eleni
sílar antalyannar.”*

—Jianxin

Table of Contents

About the Authors	xi
Acknowledgments	xiii
Chapter 1: Introduction	1
1.1 Numerical Computing in OCaml	1
1.2 Architecture.....	3
Basic Computing and Analytics with Owl	4
Advanced Design in Owl.....	5
Hardware and Deployment.....	6
Research on Owl.....	7
1.3 Summary.....	8
Chapter 2: Core Optimizations	9
2.1 N-Dimensional Array in Owl.....	9
2.2 Interface OCaml to C	12
2.3 Core Optimization of Computation	15
CPU Support of Parallel Computing	17
Vectorization.....	17
Memory	21
2.4 Optimization Techniques	26
Hardware Parallelization	26
Cache Optimization.....	27
Other Techniques.....	31
2.5 Example: Convolution.....	32
2.6 Automated Empirical Optimization of Software	42
2.7 Summary.....	47

TABLE OF CONTENTS

- Chapter 3: Algorithmic Differentiation..... 49**
 - 3.1 Introduction..... 49
 - Three Ways of Differentiating 50
 - Architecture of Algodiff Module 51
 - 3.2 Types 53
 - Forward and Reverse Modes 54
 - Data Types 58
 - Operations on AD Type..... 61
 - 3.3 Operators 63
 - Calculation Rules 63
 - Generalize Rules into Builder Template 66
 - 3.4 API 70
 - Low-Level APIs 70
 - High-Level APIs..... 73
 - 3.5 More Implementation Details 76
 - Perturbation Confusion and Tag 76
 - Lazy Evaluation..... 77
 - Extending AD 79
 - Graph Utility 80
 - 3.6 How AD Is Built upon Ndarray 81
 - 3.7 Summary..... 85
- Chapter 4: Mathematical Optimization..... 87**
 - 4.1 Gradient Descent..... 88
 - 4.2 Components 89
 - Learning Rate 90
 - Momentum 98
 - Batch 100
 - Checkpoint..... 101
 - 4.3 Gradient Descent Implementation..... 106

4.4 Regression	111
Linear Regression.....	112
Loss	112
Implementation of Linear Regression.....	113
Other Types of Regression	116
Regularization.....	117
4.5 Summary.....	119
Chapter 5: Deep Neural Networks	121
5.1 Module Architecture.....	121
5.2 Neurons.....	123
Core Functions.....	124
Activation Module.....	126
5.3 Networks.....	127
5.4 Training	130
Forward and Backward Pass.....	131
5.5 Neural Network Compiler.....	133
5.6 Case Study: Object Detection.....	138
Object Detection Network Architectures	140
Implementation of Mask R-CNN	142
5.7 Summary.....	147
Chapter 6: Computation Graph	149
6.1 The Definition of Computation Graph	149
Dynamic Graph and Static Graph.....	150
Significance in Computing.....	152
6.2 Applications Inside the Computing System.....	153
Basic Numerical Operations	153
Algorithmic Differentiation with CGraph.....	155
Deep Neural Network	158
6.3 Design of Computation Graph Module	160
Computing Device	163

TABLE OF CONTENTS

- Types of Operation 166
- Shape Inference 167
- Creating and Linking Nodes 169
- Optimization of Graph Structure 173
- Computation Engine 180
- 6.4 Optimizing Memory Usage in Computation Graph 183
- 6.5 Summary 188
- Chapter 7: Performance Accelerators 191**
- 7.1 Hardware Accelerators 191
 - Utilizing Accelerators 193
- 7.2 Design 194
 - Core Abstraction 195
 - Engines 202
- 7.3 ONNX Engine 202
 - Example 1: Basic Operations 204
 - Example 2: Variable Initialization 205
 - Example 3: Neural Network 206
- 7.4 LaTeX Engine 208
- 7.5 Owl Engine 210
- 7.6 Summary 212
- Chapter 8: Compiler Backends 215**
- 8.1 Base Library 215
- 8.2 Backend: JavaScript 217
 - Native OCaml 218
 - Facebook Reason 220
- 8.3 Backend: MirageOS 221
 - Example: Gradient Descent 221
 - Example: Neural Network 224
- 8.4 Evaluation 225
- 8.5 Summary 228

Chapter 9: Composition and Deployment	229
9.1 Script Sharing with Zoo	229
Example.....	229
Version Control	231
9.2 Service Deployment and Composition	233
9.3 System Design	235
Service	236
Type Checking	236
DSL.....	237
Service Discovery.....	238
9.4 Use Case	238
9.5 Discussion.....	239
9.6 Summary.....	240
Chapter 10: Distributed Computing	243
10.1 Distributed Machine Learning.....	243
10.2 The Actor Distributed Engine.....	245
Map-Reduce Engine	245
Parameter Server Engine	246
Compose Actor with Owl	249
10.3 Synchronization: Barrier Control Methods	252
10.4 System Design Space and Parameters.....	256
Compatibility.....	260
10.5 Convergence Analysis	264
How Effective Is Sampling.....	266
Implementation Technique	268
10.6 Evaluation	269
Experiment Setup	269
System Progress	272
Sampling Settings	275
10.7 Summary.....	278

TABLE OF CONTENTS

Chapter 11: Testing Framework 281

 11.1 Unit Test 281

 11.2 Example 282

 11.3 What Could Go Wrong..... 287

 Corner Cases 287

 Test Coverage..... 288

 11.4 Use Functor..... 288

 11.5 Performance Tests 289

 11.6 Summary..... 291

Appendix A: Basic Analytics Examples 293

Appendix B: System Conventions 303

Appendix C: Metric Systems and Constants..... 311

Appendix D: Algodiff Module 325

Appendix E: Neural Network Module 363

Appendix F: Actor System for Distributed Computing..... 423

Bibliography 457

Index..... 465