1.

Record Nr.	UNINA9910637747103321
Autore	Wang Liang
Titolo	Architecture of Advanced Numerical Analysis Systems : Designing a Scientific Computing System using OCaml / / by Liang Wang, Jianxin Zhao
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2023
ISBN	9781484288535 148428853X
Edizione	[1st ed. 2023.]
Descrizione fisica	1 electronic resource (472 p.)
Classificazione	COM000000COM004000COM014000COM031000
Altri autori (Persone)	ZhaoJianxin
Disciplina	005.13
Soggetti	Programming languages (Electronic computers) Computer science Artificial intelligence - Data processing Programming Language Computer Science Data Science Models of Computation
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Chapter 1: Introduction.-Chapter 2: Core Optimization -- Chapter 3: Algorithm Differentiation -- Chapter 4: Mathematical Optimization -- Chapter 5: Deep Neural Networks -- Chapter 6: Computation Graph -- Chapter 7: Performance Accelerators -- Chapter 8: Compiler Backends -- Chapter 9: Composition and Deployment -- Chapter 10: Distributed Computing -- Chapter 11: Testing Framework -- Appendix A: Basic Analytics Examples -- Appendix B: System Conventions -- Appendix C: Metric Systems and Constants -- Appendix D: AlgoDiff Module -- Appendix E: Neural Network Module -- Appendix F: Actor System for Distributed Computing -- Bibliography.
Sommario/riassunto	This unique open access book applies the functional OCaml programming language to numerical or computational weighted data science, engineering, and scientific applications. This book is based on the authors' first-hand experience building and maintaining Owl, an OCaml-based numerical computing library. You'll first learn the various

components in a modern numerical computation library. Then, you will learn how these components are designed and built up and how to optimize their performance. After reading and using this book, you'll have the knowledge required to design and build real-world complex systems that effectively leverage the advantages of the OCaml functional programming language. You will: Optimize core operations based on N-dimensional arrays Design and implement an industry-level algorithmic differentiation module Implement mathematical optimization, regression, and deep neural network functionalities based on algorithmic differentiation Design and optimize a computation graph module, and understand the benefits it brings to the numerical computing library Accommodate the growing number of hardware accelerators (e.g. GPU, TPU) and execution backends (e.g. web browser, unikernel) of numerical computation Use the Zoo system for efficient scripting, code sharing, service deployment, and composition Design and implement a distributed computing engine to work with a numerical computing library, providing convenient APIs and high performance.