

1. Record Nr.	UNISA990003619660203316
Autore	D'ALBERTI, Marco
Titolo	L'effettività e il diritto amministrativo / Marco D'Alberti
Pubbl/distr/stampa	Napoli : Editoriale scientifica, 2011
ISBN	978-88-6342-221-4
Descrizione fisica	66 p. ; 21 cm
Collana	Lezioni magistrali / Università degli studi Suor Orsola Benincasa, Facoltà di Giurisprudenza ; 39
Disciplina	346.45066
Soggetti	Diritti soggettivi - Tutela - Diritto amministrativo
Collocazione	XXIV.3.L. 392
Lingua di pubblicazione	Italiano
Formato	Materiale a stampa
Livello bibliografico	Monografia

2. Record Nr.	UNINA9910816885903321
Autore	Zaccone Giancarlo
Titolo	Python parallel programming cookbook : master efficient parallel programming to build powerful applications using Python / / Giancarlo Zaccone
Pubbl/distr/stampa	Birmingham, England ; ; Mumbai, India : , : Packt Publishing, , 2015 ©2015
ISBN	1-78528-672-2
Edizione	[1st edition]
Descrizione fisica	1 online resource (286 p.)
Collana	Quick answers to common problems
Disciplina	005.275
Soggetti	Parallel programming (Computer science) Python (Computer program language)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	<p> ""Cover""; ""Copyright""; ""Credits""; ""About the Author""; ""About the Reviewers""; ""www.PacktPub.com""; ""Table of Contents""; ""Preface""; ""Chapter 1: Getting Started with Parallel Computing and Python""; ""Introduction""; ""The parallel computing memory architecture""; ""Memory organization""; ""Parallel programming models""; ""How to design a parallel program""; ""How to evaluate the performance of a parallel program""; ""Introducing Python""; ""Python in a parallel world""; ""Introducing processes and threads""; ""Start working with processes in Python"" </p> <p> ""Start working with threads in Python""""Chapter 2: Thread-based Parallelism""; ""Introduction""; ""Using the Python threading module""; ""How to define a thread""; ""How to determine the current thread""; ""How to use a thread in a subclass""; ""Thread synchronization with Lock and RLock""; ""Thread synchronization with RLock""; ""Thread synchronization with semaphores""; ""Thread synchronization with a condition""; ""Threads synchronization with an event""; ""Using the with statement""; ""Thread communication using a queue""; ""Evaluating the performance of multithread applications"" </p> <p> ""Chapter 3: Process-based Parallelism""""Introduction""; ""How to spawn a process""; ""How to name a process""; ""How to run a process in the background""; ""How to kill a process""; ""How to use process in a </p>

```

subclass"; "How to exchange objects between processes"; "How to
synchronize processes"; "How to manage a state between processes";
"How to use a process pool"; "Using the mpi4py Python module";
"Point-to-point communication"; "Avoiding deadlock problems";
"Collective communication using broadcast"; "Collective
communication using scatter"
"Collective communication using gather""Collective communication
using Alltoall"; "The reduction operation"; "How to optimize the
communication"; "Chapter 4: Asynchronous Programming";
"Introduction"; "Using the concurrent.futures Python modules";
"Event loop management with Asyncio"; "Handling coroutines with
Asyncio"; "Task manipulation with Asyncio"; "Dealing with Asyncio
and Futures"; "Chapter 5: Distributed Python"; "Introduction";
"Using Celery to distribute tasks"; "How to create a task with Celery";
"Scientific computing with SCOOP"
"Handling map functions with SCOOP""Remote Method Invocation
with Pyro4"; "Chaining objects with Pyro4"; "Developing a client-
server application with Pyro4"; "Communicating sequential processes
with PyCSP"; "Using MapReduce with Disco"; "A remote procedure
call with RPyC"; "Chapter 6: GPU Programming with Python";
"Introduction"; "Using the PyCUDA module"; "How to build a
PyCUDA application"; "Understanding the PyCuda memory model with
matrix manipulation"; "Kernel invocations with GPUArray";
"Evaluating element-wise expressions with PyCUDA"
"The MapReduce operation with PyCuda"

```

Sommario/riassunto

Master efficient parallel programming to build powerful applications using Python About This Book Design and implement efficient parallel software Master new programming techniques to address and solve complex programming problems Explore the world of parallel programming with this book, which is a go-to resource for different kinds of parallel computing tasks in Python, using examples and topics covered in great depth Who This Book Is For Python Parallel Programming Cookbook is intended for software developers who are well versed with Python and want to use parallel programming techniques to write powerful and efficient code. This book will help you master the basics and the advanced of parallel computing. What You Will Learn Synchronize multiple threads and processes to manage parallel tasks Implement message passing communication between processes to build parallel applications Program your own GPU cards to address complex problems Manage computing entities to execute distributed computational tasks Write efficient programs by adopting the event-driven programming model Explore the cloud technology with Django and Google App Engine Apply parallel programming techniques that can lead to performance improvements In Detail Parallel programming techniques are required for a developer to get the best use of all the computational resources available today and to build efficient software systems. From multi-core to GPU systems up to the distributed architectures, the high computation of programs throughout requires the use of programming tools and software libraries. Because of this, it is becoming increasingly important to know what the parallel programming techniques are. Python is commonly used as even non-experts can easily deal with its concepts. This book will teach you parallel programming techniques using examples in Python and will help you explore the many ways in which you can write code that allows more than one process to happen at once. Starting with introducing you to the world of parallel computing, it moves on to cover the fundamentals in Python. This is followed by exploring the thread-based parallelism model using the Python threading module by

synchronizing threads and using locks, mutex, semaphores queues, GIL, and the thread pool. Next you will be taught about process-based parallelism where you will synchronize processes using message passing along with learning about the performance of MPI Python Modules. You will then go o...
