Community Experience Distilled

Parallel Programming with Python

Develop efficient parallel systems using the robust Python environment

Jan Palach

[PACKT] open source* PUBLISHING
community experience distilled

Parallel Programming with Python

Develop efficient parallel systems using the robust Python environment

Jan Palach

[PACKT] PUBLISHING open source community experience distilled

BIRMINGHAM - MUMBAI

Parallel Programming with Python

First published: June 2014

Production reference: 1180614

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78328-839-7

www.packtpub.com

Cover image by Lis Marie Martini (lismmartini@hotmail.com)

Credits

Author
Jan Palach

Reviewers
Cyrus Dasadia
Wei Di
Michael Galloy
Ludovic Gasc
Kamran Hussain
Bruno Torres

Commissioning Editor
Rebecca Youé

Acquisition Editor
Llewellyn Rozario

Content Development Editor
Sankalp Pawar

Technical Editors
Novina Kewalramani
Humera Shaikh

Copy Editors
Roshni Banerjee
Sarang Chari
Gladson Monteiro

Project Coordinator
Lima Danti

Proofreaders
Simran Bhogal
Maria Gould
Paul Hindle

Indexers
Mehreen Deshmukh
Rekha Nair
Tejal Soni
Priya Subramani

Graphics
Disha Haria
Abhinash Sahu

Production Coordinator
Saiprasad Kadam

Cover Work
Saiprasad Kadam

About the Author

Jan Palach has been a software developer for 13 years, having worked with scientific visualization and backend for private companies using C++, Java, and Python technologies. Jan has a degree in Information Systems from Estácio de Sá University, Rio de Janeiro, Brazil, and a postgraduate degree in Software Development from Paraná State Federal Technological University. Currently, he works as a senior system analyst at a private company within the telecommunication sector implementing C++ systems; however, he likes to have fun experimenting with Python and Erlang—his two technological passions. Naturally curious, he loves challenges and learning new technologies, meeting new people, and learning about different cultures.

Acknowledgments

I had no idea how hard it could be to write a book with such a tight deadline among so many other things taking place in my life. I had to fit the writing into my routine, taking care of my family, karate lessons, work, Diablo III, and so on. The task was not easy; however, I got to the end of it hoping that I have generated quality content to please most readers, considering that I have focused on the most important thing based on my experience.

The list of people I would like to acknowledge is so long that I would need a book only for this. So, I would like to thank some people I have constant contact with and who, in a direct or indirect way, helped me throughout this quest.

My wife Anicieli Valeska de Miranda Pertile, the woman I chose to share my love with and gather toothbrushes with to the end of this life, who allowed me to have the time to create this book and did not let me give up when I thought I could not make it. My family has always been important to me during my growth as a human being and taught me the path of goodness.

I would like to thank Fanthiane Ketrin Wentz, who beyond being my best friend is also guiding me through the ways of martial arts, teaching me the values I will carry during a lifetime—a role model for me. Lis Marie Martini, dear friend who provided the cover for this book, and who is an incredible photographer and animal lover.

Big thanks to my former English teacher, reviser, and proofreader, Marina Melo, who helped along the writing of this book. Thanks to the reviewers and personal friends, Vitor Mazzi and Bruno Torres, who contributed a lot to my professional growth and still do.

Special thanks to Rodrigo Cacilhas, Bruno Bemfica, Rodrigo Delduca, Luiz Shigunov, Bruno Almeida Santos, Paulo Tesch (corujito), Luciano Palma, Felipe Cruz, and other people with whom I often talk to about technology. A special thanks to Turma B.

Big thanks to Guido Van Rossum for creating Python, which transformed programming into something pleasant; we need more of this stuff and less set/get.

About the Reviewers

Cyrus Dasadia has worked as a Linux system administrator for over a decade for organizations such as AOL and InMobi. He is currently developing CitoEngine, an open source alert management service written entirely in Python.

Wei Di is a research scientist at eBay Research Labs, focusing on advanced computer vision, data mining, and information retrieval technologies for large-scale e-commerce applications. Her interest covers large-scale data mining, machine learning in merchandising, data quality for e-commerce, search relevance, and ranking and recommender systems. She also has years of research experience in pattern recognition and image processing. She received her PhD from Purdue University in 2011 with focuses on data mining and image classification.

Michael Galloy works as a research mathematician for Tech-X Corporation involved in scientific visualizations using IDL and Python. Before that, he worked for five years teaching all levels of IDL programming and consulting for Research Systems, Inc. (now Exelis Visual Information Solutions). He is the author of Modern IDL (`modernidl.idldev.com`) and is the creator/maintainer of several open source projects, including IDLdoc, mgunit, dist_tools, and cmdline_tools. He has written over 300 articles on IDL, scientific visualization, and high-performance computing for his website `michaelgalloy.com`. He is the principal investigator for NASA grants *Remote Data Exploration with IDL* for DAP bindings in IDL and *A Rapid Model Fitting Tool Suite* for accelerating curve fitting using modern graphic cards.

Ludovic Gasc is a senior software integration engineer at Eyepea, a highly renowned open source VoIP and unified communications company in Europe. Over the last five years, Ludovic has developed redundant distributed systems for Telecom based on Python (Twisted and now AsyncIO) and RabbitMQ.

He is also a contributor to several Python libraries. For more information and details on this, refer to `https://github.com/GMLudo`.

Kamran Husain has been in the computing industry for about 25 years, programming, designing, and developing software for the telecommunication and petroleum industry. He likes to dabble in cartooning in his free time.

Bruno Torres has worked for more than a decade, solving a variety of computing problems in a number of areas, touching a mix of client-side and server-side applications. Bruno has a degree in Computer Science from Universidade Federal Fluminense, Rio de Janeiro, Brazil.

Having worked with data processing, telecommunications systems, as well as app development and media streaming, he developed many different skills starting from Java and C++ data processing systems, coming through solving scalability problems in the telecommunications industry and simplifying large applications customization using Lua, to developing apps for mobile devices and supporting systems.

Currently he works at a large media company, developing a number of solutions for delivering videos through the Internet for both desktop browsers and mobile devices.

He has a passion for learning different technologies and languages, meeting people, and loves the challenges of solving computing problems.

www.PacktPub.com

Support files, eBooks, discount offers, and more

You might want to visit `www.PacktPub.com` for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.PacktPub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `service@packtpub.com` for more details.

At `www.PacktPub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

`http://PacktLib.PacktPub.com`

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

Free access for Packt account holders

If you have an account with Packt at `www.PacktPub.com`, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

I dedicate this book in the loving memory of Carlos Farias Ouro de Carvalho Neto.

–Jan Palach

Preface

Months ago, in 2013, I was contacted by Packt Publishing professionals with the mission of writing a book about parallel programming using the Python language. I had never thought of writing a book before and had no idea of the work that was about to come; how complex it would be to conceive this piece of work and how it would feel to fit it into my work schedule within my current job. Although I thought about the idea for over a couple of days, I ended up accepting the mission and said to myself that it will be a great deal of personal learning and a perfect chance to disseminate my knowledge of Python to a worldwide audience, and thus, hopefully leave a worthy legacy along my journey in this life.

The first part of this work is to outline its topics. It is not easy to please everybody; however, I believe I have achieved a good balance in the topics proposed in this mini book, in which I intended to introduce Python parallel programming combining theory and practice. I have taken a risk in this work. I have used a new format to show how problems can be solved, in which examples are defined in the first chapters and then solved by using the tools presented along the length of the book. I think this is an interesting format as it allows the reader to analyze and question the different modules that Python offers.

All chapters combine a bit of theory, thereby building the context that will provide you with some basic knowledge to follow the practical bits of the text. I truly hope this book will be useful for those adventuring into the world of Python parallel programming, for I have tried to focus on quality writing.

What this book covers

Chapter 1, Contextualizing Parallel, Concurrent, and Distributed Programming, covers the concepts, advantages, disadvantages, and implications of parallel programming models. In addition, this chapter exposes some Python libraries to implement parallel solutions.

Chapter 2, Designing Parallel Algorithms, introduces a discussion about some techniques to design parallel algorithms.

Chapter 3, Identifying a Parallelizable Problem, introduces some examples of problems, and analyzes if these problems can be divided into parallel pieces.

Chapter 4, Using the threading and concurrent.futures Modules, explains how to implement each problem presented in *Chapter 3, Identifying a Parallelizable Problem*, using the threading and concurrent.futures modules.

Chapter 5, Using Multiprocessing and ProcessPoolExecutor, covers how to implement each problem presented in *Chapter 3, Identifying a Parallelizable Problem*, using multiprocessing and ProcessPoolExecutor.

Chapter 6, Utilizing Parallel Python, covers how to implement each problem presented in *Chapter 3, Identifying a Parallelizable Problem*, using the parallel Python module.

Chapter 7, Distributing Tasks with Celery, explains how to implement each problem presented in *Chapter 3, Identifying a Parallelizable Problem*, using the Celery distributed task queue.

Chapter 8, Doing Things Asynchronously, explains how to use the asyncio module and concepts about asynchronous programming.

What you need for this book

Previous knowledge of Python programming is necessary as a Python tutorial will not be included in this book. Knowledge of concurrence and parallel programming is welcome since this book is designed for developers who are getting started in this category of software development. In regards to software, it is necessary to obtain the following:

- Python 3.3 and Python 3.4 (still under development) are required for *Chapter 8, Doing Things Asynchronously*
- Any code editor of the reader's choice is required
- Parallel Python module 1.6.4 should be installed

- Celery framework 3.1 is required for *Chapter 5, Using Multiprocessing and ProcessPoolExecutor*
- Any operating system of the reader's choice is required

Who this book is for

This book is a compact discussion about parallel programming using Python. It provides tools for beginner and intermediate Python developers. This book is for those who are willing to get a general view of developing parallel/concurrent software using Python, and to learn different Python alternatives. By the end of this book, you will have enlarged your toolbox with the information presented in the chapters.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "In order to exemplify the use of the `multiprocessing.Pipe` object, we will implement a Python program that creates two processes, A and B."

A block of code is set as follows:

```
def producer_task(conn):
    value = random.randint(1, 10)
    conn.send(value)
    print('Value [%d] sent by PID [%d]' % (value, os.getpid()))
    conn.close()
```

Any command-line input or output is written as follows:

```
$celery -A tasks -Q sqrt_queue,fibo_queue,webcrawler_queue worker
--loglevel=info
```

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to `feedback@packtpub.com`, and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at `http://www.packtpub.com`. If you purchased this book elsewhere, you can visit `http://www.packtpub.com/support` and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from `http://www.packtpub.com/support`.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at `questions@packtpub.com` if you are having a problem with any aspect of the book, and we will do our best to address it.

Table of Contents