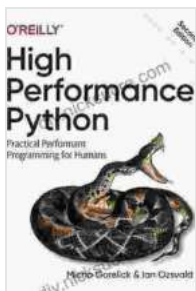


High Performance Python: Practical Performant Programming for Humans

Python is a powerful and versatile language that is used for a wide variety of applications. However, Python can also be slow, especially when compared to other languages such as C or C++. This is because Python is an interpreted language, which means that it is not compiled into machine code before it is run. As a result, Python code is typically slower than compiled code.



High Performance Python: Practical Performant Programming for Humans by Micha Gorelick

★★★★☆ 4.7 out of 5

Language : English
File size : 12246 KB
Text-to-Speech : Enabled
Screen Reader : Supported
Enhanced typesetting : Enabled
Print length : 470 pages



However, there are a number of things that you can do to improve the performance of your Python code. In this article, we will discuss some of the most common Python performance optimization techniques. We will start with some basic tips and then move on to more advanced strategies.

Basic Python Performance Optimization Techniques

Here are some basic tips for improving the performance of your Python code:

- Use the right data structures. The choice of data structure can have a significant impact on the performance of your code. For example, if you are storing a large number of items, you should use a list instead of a tuple. Tuples are immutable, which means that they cannot be changed once they have been created. This makes them faster than lists, but it also means that they cannot be used to store items that need to be changed.
- Avoid unnecessary loops. Loops can be slow, especially if they are nested. If you can, try to avoid using loops altogether. For example, instead of using a loop to iterate over a list, you can use the `sum()` function to add up all of the items in the list.
- Use the right algorithms. The choice of algorithm can also have a significant impact on the performance of your code. For example, if you are searching for an item in a list, you should use the `binary search` algorithm instead of the `linear search` algorithm. The binary search algorithm is much faster than the linear search algorithm, especially for large lists.
- Use a profiler. A profiler is a tool that can help you identify the parts of your code that are taking the most time. Once you know which parts of your code are slow, you can focus on optimizing those parts.

Advanced Python Performance Optimization Techniques

In addition to the basic tips above, there are a number of more advanced Python performance optimization techniques that you can use. These

techniques can be more difficult to implement, but they can also provide significant performance improvements.

Here are some of the most common advanced Python performance optimization techniques:

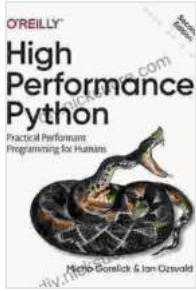
- Use a JIT compiler. A JIT compiler is a compiler that compiles your code into machine code at runtime. This can significantly improve the performance of your code, especially if you are using a lot of loops or recursion.
- Use a parallel programming framework. A parallel programming framework can help you to distribute your code across multiple CPUs. This can significantly improve the performance of your code, especially if you are performing a lot of data-intensive tasks.
- Use a database. A database can help you to store and retrieve data quickly and efficiently. This can significantly improve the performance of your code, especially if you are working with large amounts of data.
- Use a cache. A cache can help you to store frequently used data in memory. This can significantly improve the performance of your code, especially if you are accessing the same data multiple times.

Python is a powerful and versatile language, but it can also be slow. However, there are a number of things that you can do to improve the performance of your Python code. By following the tips in this article, you can write code that is both fast and efficient.

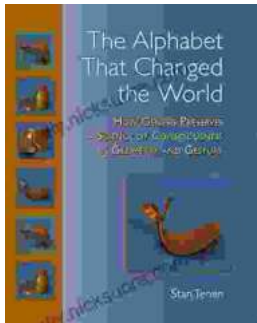
High Performance Python: Practical Performant Programming for Humans

by Micha Gorelick

★★★★☆ 4.7 out of 5



Language : English
File size : 12246 KB
Text-to-Speech : Enabled
Screen Reader : Supported
Enhanced typesetting : Enabled
Print length : 470 pages



How Genesis Preserves Science Of Consciousness In Geometry And Gesture

The book of Genesis is a foundational text for many religions, and it contains a wealth of information about the origins of the world and humankind. But...



At Day's Close, Night in Times Past

As the sun dips below the horizon, the world undergoes a remarkable transformation. The vibrant hues of day give way to the mysterious embrace of...