Analysis of Numpy Computing Efficiency

Wei Xiao*

May 3, 2018

Abstract

The NumPy system is an open source, high-performance numerical computing extension package for the Python language. Its performance usually affects the efficiency of the entire program. We analyze the memory management rules of Numpy on a Python virtual machine, the Python virtual machine on the instruction set, the operation to find the optimal calculation efficiency. We use the Python built-in tools such as cProfile, %timeit, and memory_profiler to analyze the performance of the function. We measure the different computational efficiency of different function statements. Finally based on the above analysis gives a part of the recommendations to improve the efficiency of the implementation[1]

*I am grateful to the PLPP teacher for supervise me during this course.

[1]This isn't the last title.With more study in high performance of Python.I may change to another title.

1 Introduction

The introduction section mainly describes the python language, the main problem of low efficiency, the cost of the type judgment, memory operation, etc.

2 Numpy System

2.1 Environment

2.1.1 Python virtual machine

Introducting the basic concepts of Python virtual machine[1]...

2.1.2 Numpy background

The Numpy is developed by C,Fortran,shell...

2.2 Memory Rule

2.2.1 List

Application and release rules of memory in the Numpy list[2]

2.2.2 tuple

Rules for the management of elements in Numpy tuples.

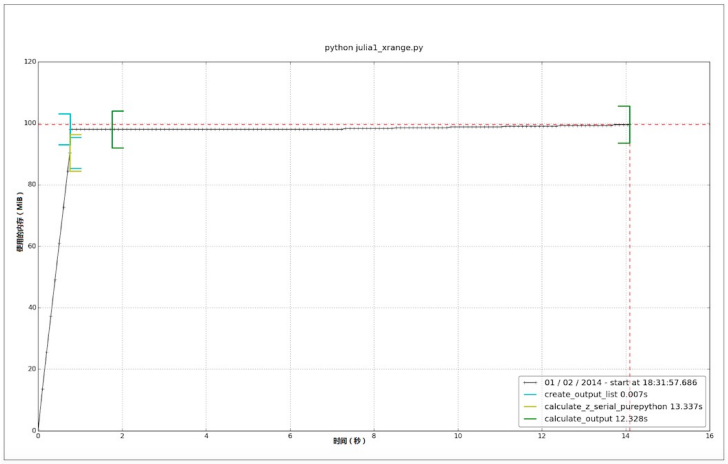

Figure 1: The difference of range and xrange

3 Optimization of the status

3.1 Computing optimization

there are another reason of using CPU

$$E = mc^2 \tag{1}$$

to be honest, python have many failure.in the next code[1, 2]

```
    class Solution:
def findMedianSortedArrays(self, nums1, nums2):
    """
    :type nums1: List[int]
    :type nums2: List[int]
    :rtype: float
    """
    sl=len(nums1)+len(nums2)
    for i in range(sl):
        if len
        if(nums1[0]<nums2[0])
```

3.2 Memory Optimization

balabala...

3.3 Algorithm Optimization

balabala...

4 a more useful Optimization

4.1 conclusion

.....

4.2 Result analysis

.....

References

[1] High Performance Python by Micha Gorelick and Ian Ozsvald (OReilly). Copyright 2014 Micha Gorelick and Ian Ozsvald, 978-1-449-36159-4.

[2] The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science and Engg. 13, 2 (March 2011), 2230

[3] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y., 2010. Theano: a CPU and GPU math expression compiler. In: Proceedings of the Python for Scientific Computing Conference (SciPy), June 30July 3, Austin, TX.