Numba: A Python Compiler

Stan Seibert
Continuum Analytics

2016-11-12
Numba: A JIT Compiler for Python Functions

- An open-source, function-at-a-time compiler library for Python
- Compiler toolbox for different targets and execution models:
  - single-threaded CPU, multi-threaded CPU, GPU
  - regular functions, “universal functions” (array functions), GPU kernels
- Speedup: 2x (compared to basic NumPy code) to 200x (compared to pure Python)
- Combine ease of writing Python with speeds approaching FORTRAN
- Empowers data scientists who make tools for themselves and other data scientists
How does Numba work?

@jit
def do_math(a, b):
    ... 

>>> do_math(x, y)
## Supported Platforms and Hardware

<table>
<thead>
<tr>
<th>OS</th>
<th>HW</th>
<th>SW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows (7 and later)</td>
<td>32 and 64-bit x86 CPUs</td>
<td>Python 2 and 3</td>
</tr>
<tr>
<td>OS X (10.9 and later)</td>
<td>CUDA &amp; HSA Capable GPUs</td>
<td>NumPy 1.7 through 1.11</td>
</tr>
<tr>
<td>Linux (RHEL 5 and later)</td>
<td>Experimental support for ARM, Xeon Phi, AMD Fiji GPUs</td>
<td></td>
</tr>
</tbody>
</table>
Tutorial Acknowledgements

- These Numba tutorial materials are adapted from the Numba Tutorial at SciPy 2016 by Gil Forsyth and Lorena Barba
- I’ve made some adjustments and additions, and also had to skip quite a bit of material for time.
Notebook 1: Numba Basics
Notebook 2: How Numba Works
Ex01: Intro to JIT
That’s it?

- Mostly, yes.
- The Secret of Numba is:
  - *If it doesn’t need to be fast, leave it alone. (See the profiler section of this tutorial.)*
  - *Stick to the well-worn path: Numba works best on loop-heavy numerical algorithms.*
  - *Choose the right data structures: Numba works best on NumPy arrays and scalars.*
Ex02: Direct Summation
There is more, though.

- Numba can compile other kinds of functions:
  - Universal function (ufuncs) apply a scalar function to elements of the input arrays according to the broadcast rules:

  ```
  numpy.add([1, 2, 3], 1) == [2, 3, 4]
  numpy.add([1, 2, 3], [10, 20, 30]) == [11, 12, 13]
  ```
Notebook 3: Ufuncs
More Advanced Topics

- Generalized ufuncs:
  - Instead of broadcasting all dimensions into a scalar function, you can control how input dimensions are broadcast.
  - Example: Writing a norm() function
    - http://numba.pydata.org/numba-doc/0.29.0/user/vectorize.html#the-guvectorize-decorator

- Calling external code:
  - Numba can call C code that has been wrapped with ctypes or CFFI
    - http://numba.pydata.org/numba-doc/0.29.0/reference/pysupported.html#ctypes
    - http://numba.pydata.org/numba-doc/0.29.0/reference/pysupported.html#cffi

- Ahead of time compilation: