SPEED

... intro to Cython
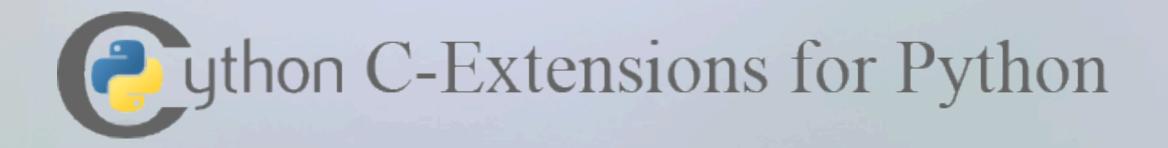
Aaron Defazio

My Background

Feedback Loop

ambiata
BIG DATA ANALYTICS

NICTA

Python C-Extensions for Python

Interacting with C/C++ libraries

Speeding up python code

General idea:

A python to
C compiler

Produces line by line translated C code

```
1111    */
1112     while (1) {
1113       __pyx_t_3 = (__pyx_v_childpos < __pyx_v_endpos);
1114       if (!__pyx_t_3) break;
1115
1116       /* "heap.pyx":21
1117    *          # Set childpos to index of the larger of the two children.
1118    *
1119    *          rightpos = childpos + 1                    # <<<<<<<<<<<<<<
1120    *          if rightpos < endpos and  heap[rightpos] > heap[childpos]:
1121    *              childpos = rightpos
1122    */
1123       __pyx_v_rightpos = (__pyx_v_childpos + 1);
1124
1125       /* "heap.pyx":22
1126    *
1127    *          rightpos = childpos + 1
1128    *          if rightpos < endpos and  heap[rightpos] > heap[childpos]:
1129    *              childpos = rightpos
1130    *
1131    */
1132       __pyx_t_3 = (__pyx_v_rightpos < __pyx_v_endpos);
1133       if (__pyx_t_3) {
```

Basic workflow

Rename file from .py to .pyx

Add types to your variables

Add some lines to setup.py

Compile & Run

DEMO:

Heap Operations

Speedup: **5.5x**

Excluding python calling code: **9x**

Caveats

Recompiling after changes

Harder to debug

Calling python methods with Cython can negate advantages

Thats all,
Thanks