Rozszerzanie Pythona modułami w C++

Karol Horosin

8 marca 2018

Co omówimy?

Jak zidentyfikować wolne części naszego programu w Pythonie i przyspieszyć je za pomocą C++.

Plan

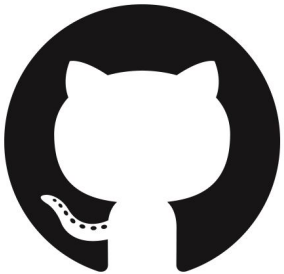
1. Jak i dlaczego?
2. Profilowanie kodu
3. Przyspieszamy
4. Tips & tricks
5. Q&A

ardigen

Artificial Intelligence & Bioinformatics
for Precision Medicine

CODE AGAINST CANCER

karol@horosin.com



github.com/horosin/4dev-cpp

Nie róbcie tego.

Chyba, że potrzebujecie

C++

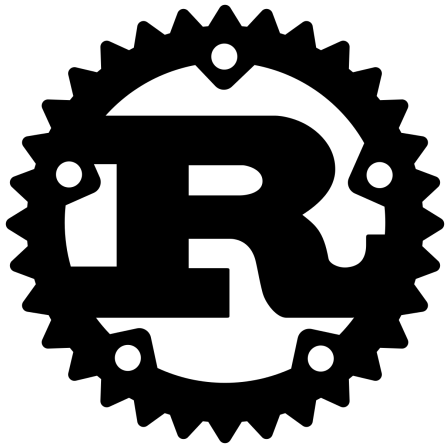
Python C API - ...

pybind11

Alternatywy

1. Przepisanie na numpy
2. CFFI/Ctypes
3. Rust modules
4. PyPy
5. Cython

Rust & PyO3



https://pyo3.rs
https://www.benfrederickson.com/writing-python-extensions-in-rust-using-pyo3/

The program

Profiling

```
python3 -m cProfile tests/program.py
```

or save to file

```
python3 -m cProfile -o stats tests/program.py
```

```
MacBook-Pro-Karol:python_example karol$ python3 -m cProfile tests/program.py
         5006129 function calls (5006086 primitive calls) in 1.850 seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        9    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:103(release)
        9    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:143(__init__)
        9    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:147(__enter__)
        9    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:151(__exit__)
        9    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:157(_get_module_lock)
        9    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:176(cb)
     15/1    0.000    0.000    0.011    0.011 <frozen importlib._bootstrap>:211(_call_with_frames_removed)
      165    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:222(_verbose_message)
        9    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:307(__init__)
        9    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:311(__enter__)
        9    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:318(__exit__)
       36    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:321(<genexpr>)
        3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:35(_new_module)
        9    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:369(__init__)
       12    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:403(cached)
        9    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:416(parent)
        9    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:424(has_location)
        9    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:504(_init_module_attrs)
        9    0.000    0.000    0.008    0.001 <frozen importlib._bootstrap>:564(module_from_spec)
        9    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:58(__init__)
      9/1    0.000    0.000    0.012    0.012 <frozen importlib._bootstrap>:651(_load_unlocked)
```

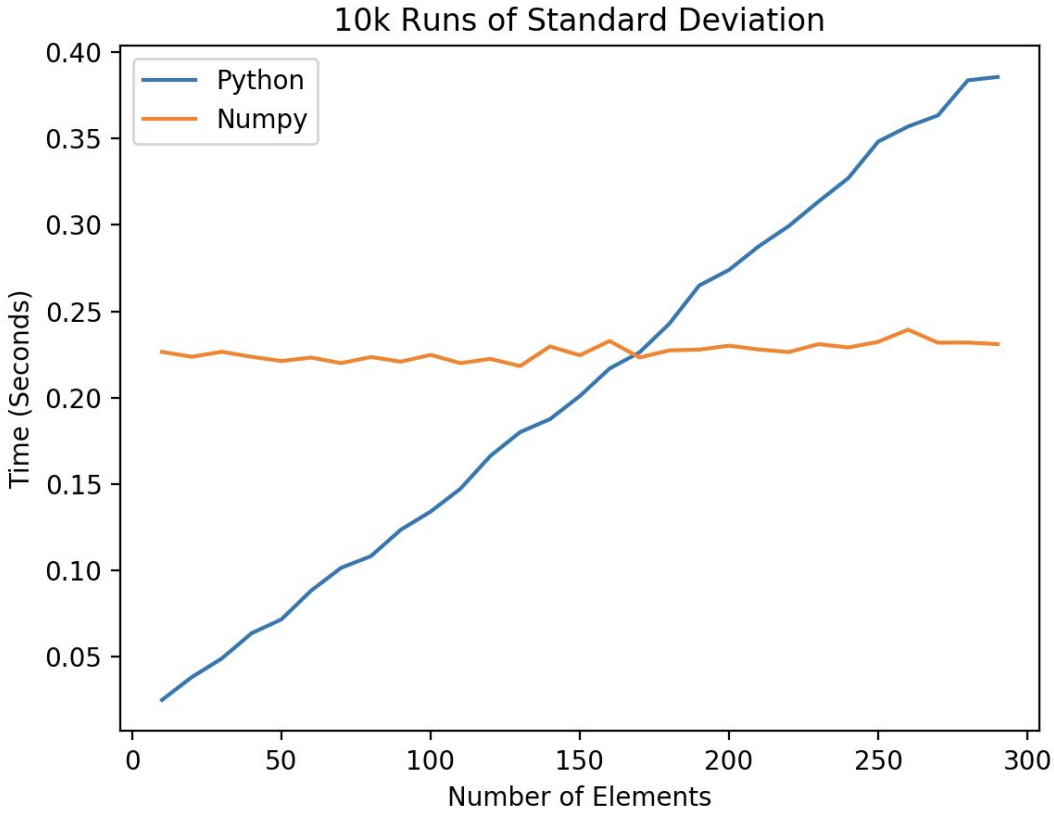
```
pip install pyinstrument
python -m pyinstrument tests/program.py
```

```
MacBook-Pro-Karol:python_example karol$ python3 -m pyinstrument tests/program.py
1.549 main  program.py:17
├─ 1.200 standard_deviation  program.py:10
│  ├─ 1.150 <listcomp>  program.py:13
│  └─ 0.022 mean  program.py:5
└─ 0.314 <listcomp>  program.py:23
```

Przyspieszamy (numpy)

```
np.std(arr)
```

Benchmarking


10k Runs of Standard Deviation
Python
Numpy
Time (Seconds)
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
50
100
150
200
250
300
Number of Elements

Przyspieszamy (C++)

Wymagania

1. Python
2. Starter repo
    a. Simplified:
    github.com/horosin/python-cpp-starter
    b. Official (with conda support, etc.)
    https://github.com/pybind/python_example

Mały test

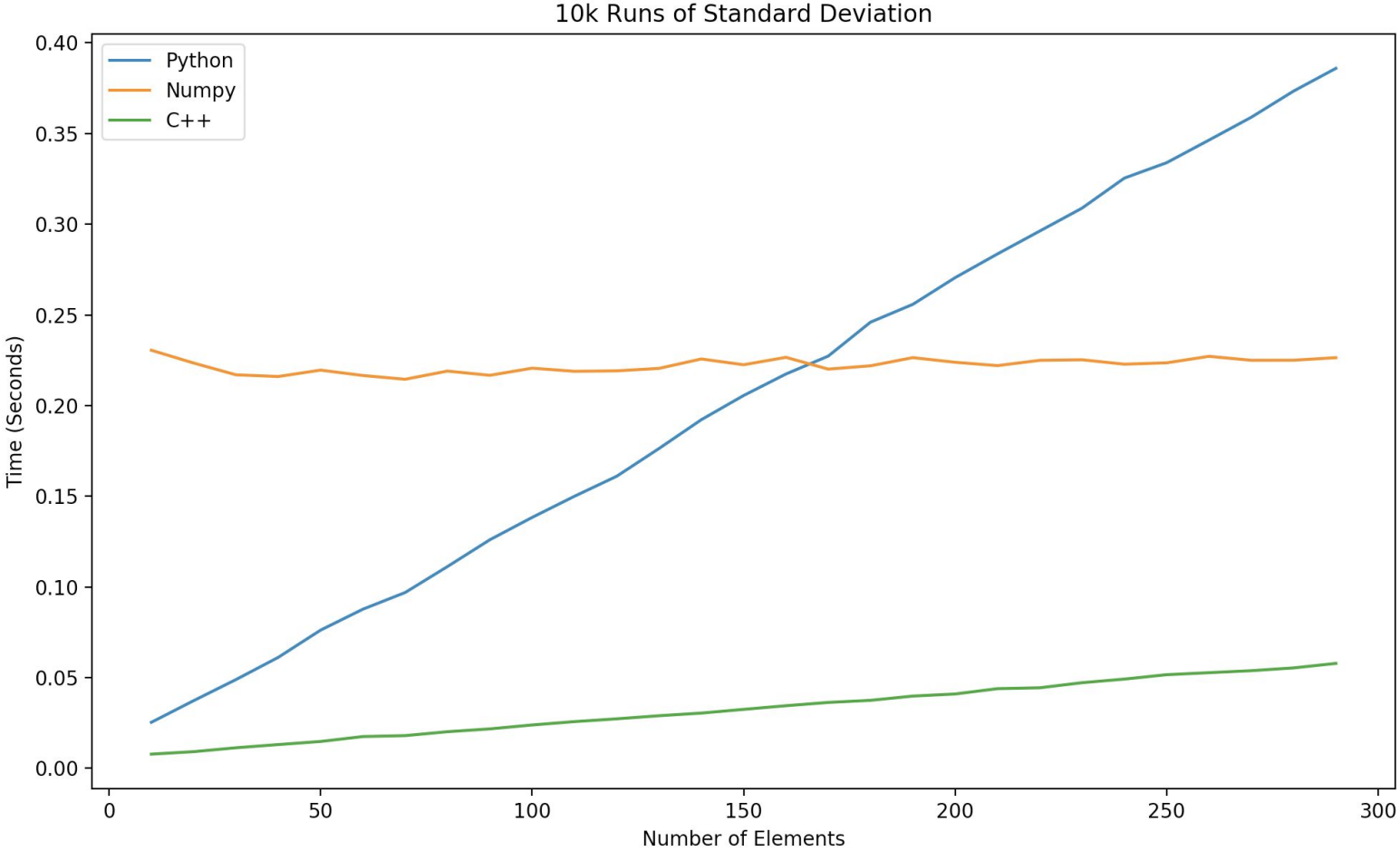
```
git clone https://github.com/horosin/python-cpp-starter

cd python-cpp-starter

pip install .

python3 -c "import python_example;
print(python_example.add(1, 2))"
```

Implementacja


10k Runs of Standard Deviation
Python
Numpy
C++
Time (Seconds)
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
50
100
150
200
250
300
Number of Elements

Tips & tricks

```
template <typename T>
T add(T i, T j) {
    return i + j;
}

PYBIND11_MODULE(python_example, m) {
    m.def("add", add<int>, R"pbdoc(Add two ints);
    m.def("add", add<double>, R"pbdoc(Add two doubles);
}
```

```
namespace py = pybind11;

...

py::gil_scoped_release release;

heavy_computation();

py::gil_scoped_acquire acquire;
```

notebook


All aboard the
HOT MESS
EXPRESS

podsumowując


eventory

Q&A

Materiały

Really good talk, going more in-depth with features:
https://www.youtube.com/watch?v=jQedHfF1Jfw

A lot of examples
https://github.com/tdegeus/pybind11_examples

General C Ext advice:
https://www.youtube.com/watch?v=bJq1n4gQFfw

https://medium.com/coding-with-clarity/speeding-up-python-and-numpy-c-ing-the-way-3b9658ed78f4

Gentle intro to cpp for high level language users

https://itnext.io/c-for-javascript-developers-program-compilation-source-vs-header-files-1829a69a0a56