Rust in Python

SPEEDING UP PYTHON CODE WITH MATURIN AND PYO3

Why not Python?

Python is an interpreted, high-level, general-purpose programming language. It is dynamically typed and garbage-collected ~Wikipedia

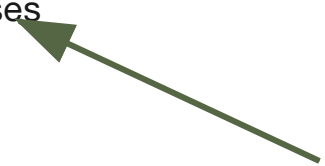
🐌 Its slow

🐌 Hard to do things in paralell because of GIL - only one thread hold the control of the Python interpreter

How we usually overcome this?

🚀 By importing already compiled functions and classes

🚀 By forking - running code on workers



π estimation in Python

```python
def estimate_pi(n: int) -> float:
    """
    Estimate pi using Leibniz's formula:
    X = 4 - 4/3 + 4/5 - 4/7 + 4/9 ...
    """
    denominator = 1
    pi = 0
    sign = 1

    for _ in range(n):
        pi += sign * 4/denominator
        denominator += 2
        sign = -sign

    return pi
```

π estimation in Rust

```rust
/// Estimate pi using Leibniz's formula:
/// X = 4 - 4/3 + 4/5 - 4/7 + 4/9 ...
pub fn estimate_pi(n: i32) -> f64 {
    let mut denominator = 1.0;
    let mut pi = 0.0;
    let mut sign = 1.0;

    for _ in 0..n {
        pi += sign * 4.0 / denominator;
        denominator += 2.0;
        sign = -sign;
    }

    pi
}
```

Execution time comparison

python3 main.py --pi 100000000 --py

```
Running π estimation in python
estimated: π=3.141592643589
target:    π=3.141592653589
Estimation took 8.95 seconds
```

python3 main.py --pi 100000000 --rust

```
Running π estimation in rust
estimated: π=3.141592643589
target:    π=3.141592653589
Estimation took 0.15 seconds
```

Pyo3 and maturin

- Pyo3 https://pyo3.rs/v0.19.1
- Maturin https://pypi.org/project/maturin

Lets dig into the code …

Packet decoding in Python

```
/// Decode packets from given pcap file.
/// Store each packet data in PktInfo struct
/// Return list of PktInfo structs
```

python3 main.py --pcap-info some_pcap.pcap --py

```
Decoding packets in python
Decoded pcaps: 100000
Pcap decoding took 29.87 seconds
```

python3 main.py --pcap-info some_pcap.pcap --rust

```
Decoding packets in rust
Decoded pcaps: 100000
Pcap decoding took 0.04 seconds
```

Using Rust structs in Python

- Getting fields

```
#[pyo3(get, set)]
pub src_port: u16,

#[getter(src_ip)]
fn src_ip(&self) -> pyo3::PyResult<String> {
    Ok(self.src_ip.to_string())
}
```

- Creating new instances

```
#[new]
fn new(src_ip: &str, dst_ip: &str, src_port: u16, dst_port: u16) -> Result<Self, AddrParseError> {
    ...
}
```

- Printing

```
fn __repr__(&self) -> String {
    ...
}
```

Using Python structs in Rust

- GIL

```rust
pub fn get_filtered_pkt_infos(filename: &str, pkt_filter: &PyAny) -> Vec<PktInfo>
```

- Accessing fields and methods

```rust
fn call_method(&self, name: &str, args: (u16, u16), kwargs: Option<&PyDict>) -> Result<&PyAny, PyErr>
```

- NO (less) GIL

```rust
#[derive(pyo3::FromPyObject)]
pub struct PktFilter {
    #[pyo3(item)]
    pub ports: Vec<u16>,
    #[pyo3(item)]
    pub ips: Vec<String>,
}
```

Thank you

HTTPS://PYO3.RS/MAIN

HTTPS://GITHUB.COM/PYO3/MATURIN