NumPy: What has changed and what is going to change?

Matti Picus (mattip)

BIDs staff and full-time NumPy developer

BIDS
BERKELEY INSTITUTE
FOR DATA SCIENCE

Berkeley
UNIVERSITY OF CALIFORNIA

Disclaimer: all ideas are mine, I do not represent BIDS

Who am I?

- Kibbutznik at heart, believer in community
- PyPy core developer: numpypy and cpyext
- Halfway through a two-year contract to work on NumPy

Wait a minute, I’m getting paid to work on Open Source?

Yes!!! Not only that, there are non-tangible benefits:

- Makes my work more varied
- Gives back to the community
- Makes me a better programmer (this is a whole other talk)

How can you do this too?

- Join a group / become a Fellow: OpenCollectives, GSOC, GSOD, Django Fellows, NumPy ...
- Work for a company that expressly supports OpenSource: QuanSight, Anaconda, RedHat ...
- Negotiate with your employer for time to contribute
- Individual paths: Patreon, Twitch (much harder)

Back to NumPy

What is the grant sponsoring?

Technology

- Triaging
- matmul `@`
- Protocols like `__array_function__`
- random module
- dtypes

What is the grant sponsoring?

Community

- Vision for the future
- Mentoring
- Speaking

Vision

NEPS

Roadmap

Wishlist

Governance model

Governance

- Steering council
- SciPy meetings, meetups at BIDS
- Open to all
- Code of conduct

Mentoring to improve diversity

- Outreachy
- Google Season of Documentation
- Teaching, providing reviewed resources for learning

Technology

Triaging

- 50 - 75% of my time
- We process about 120 PRs and 240 issues a month
- Started with 280 PRs, 1800 issues
- Now 210 PRs, 1700 issues 🤪

Merged over the past year

- matmul (a @ a)
- random
- `__array_function__`

Random (Kevin Sheppard and others)

```
np.random.random(10)  # old
np.Generator(Xoshiro256()).random(10)
```

- Faster than the old Mercienne Twister
- Able to jump
- It’s a bird, it’s a plane, it’s **randomgen**

NumPy is great but ...

- Dask
- CuPy
- xtensor
- xdarray
- TensorFlow, Pytorch, Jax
- arrow
- Others?

How to preserve the API but not be tied to an implementation?

Protocols!

__array__

__array_wrap__

__array_ufunc__

__array_function__ (Stefan Hoyer and others)

```
import numpy as np
import cupy

def funky_function(a):
    # Uses the numpy API directly by calling np.*
    return np.diagonal(a).reshape(10, -1)

c = cupy.empty_like(20, dtype='float32')
d = funky_function(c)
```

__array_function__

```
class cparray(np.ndarray):
    def __array_function__(...):
        ...
```

Jax implementation

Dtype

Use cases

- units
- quaternions, “tensors”
- categorical data
- ints-with-overflow-warnings

Dtype changes – what are the implications?

- Ability to subclass np.dtype
- “Pythonify” the class
- Enhancing ufuncs to know about dtypes
- Make the PyArray_Descr more opaque, will allow future changes more easily

Takeaways I

- There are ways to get paid to work on Open Source
- Involves "people" skills and a thick skin
- Makes you a better programmer

Takeaways II

- NumPy is like a good wine: getting better with age
- Strives to be a good community member, plays well with others via protocols and PEP 3118
- Needs your help. If not via code
  - Support NumFocus, TideLift
  - Tell us your pain points