Importance of NumPy in Advancing Scientific Discovery

INDIAN INSTITUTE OF SCIENCE EDUCATION AND RESEARCH • BHOPAL

विद्ययाऽमृतमश्नुते

Name: Sambhav Agrawal

Roll No. : 19264

Department: Data Science and Engineering

Python Libraries for Scientific Computation

- NumPy is not a part of Python’s standard library then also, it has a good relation with Python developers. SciPy and Matplotlib are coupled with NumPy in terms of history, development and uses.

- SciPy is a free and open-source Python library used for scientific and technical computing. It is a collection of mathematical algorithms and domain-specific toolboxes. It is built on the NumPy extension.

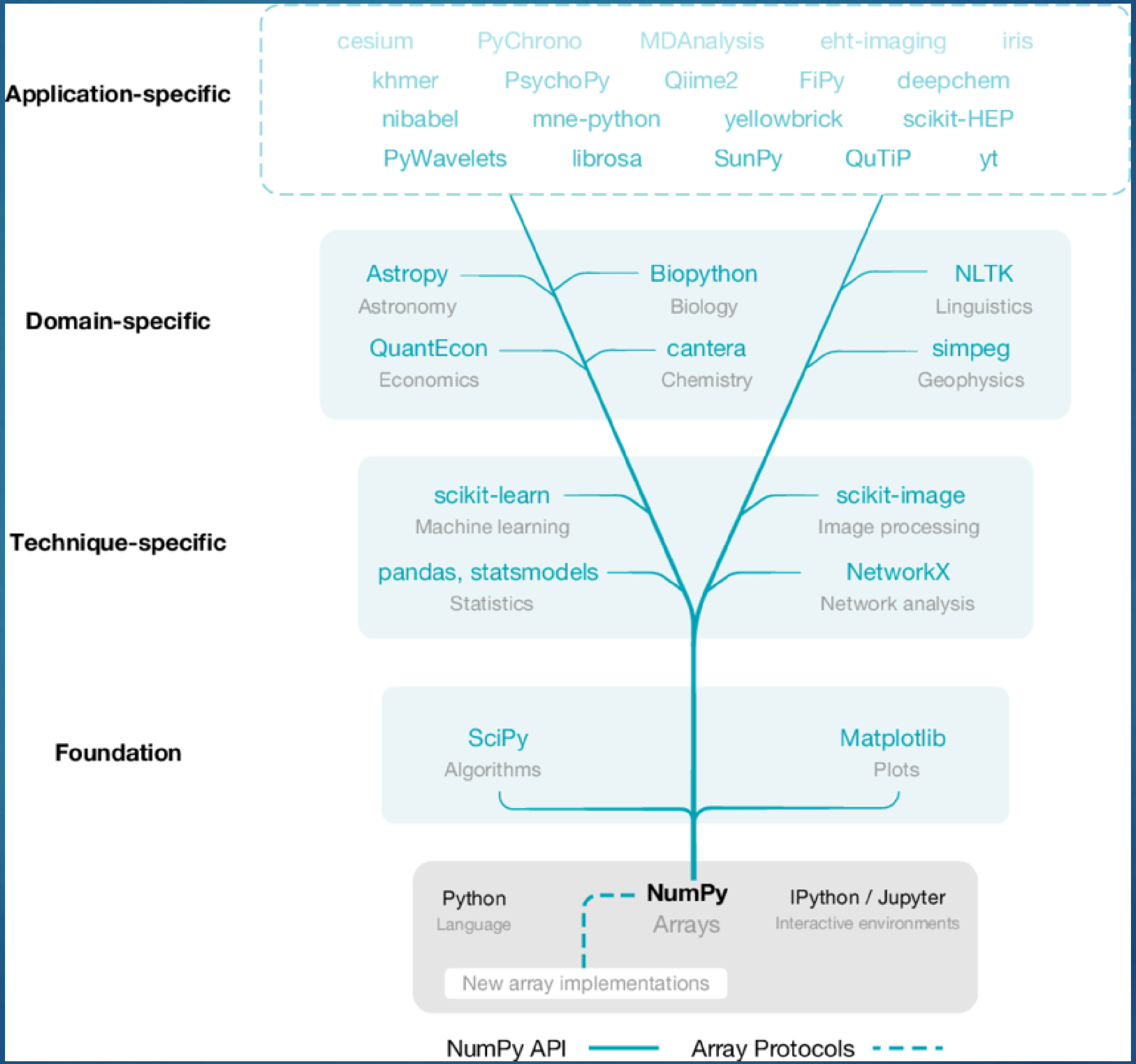
- Matplotlib is a visualization and plotting library for the Python programming language and NumPy. It is a multi-platform data visualization library built on NumPy arrays.

Introduction to NumPy

- Stands for Numerical Python, it is the fundamental package for scientific computing in Python, provides a multidimensional array object, along with routines for fast operations on arrays including mathematical, logical, shape manipulation, sorting, etc.

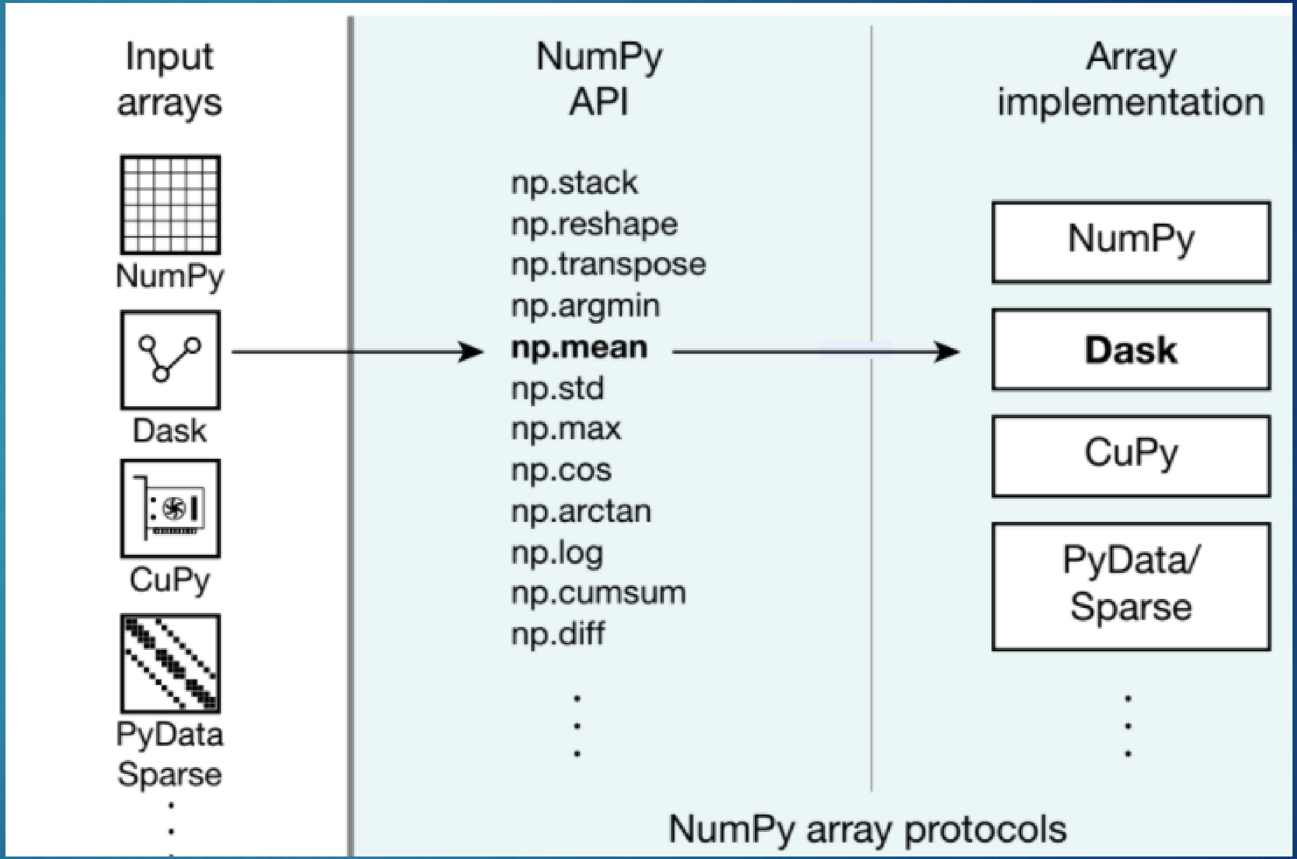
- Primary array programming library for Python and has an important role in research analysis in physics, chemistry, astronomy, etc. In astronomy, it was an important part of software stack used in discovery of gravitational waves and first imaging of a black hole.

NumPy: Base for scientific Python ecosystem



NumPy Arrays

- Central data structures of NumPy library, it is a grid of values and contain information about raw data.
- The elements are all of same type called as array dtype.
- Can be indexed by tuple of nonnegative integers, by another array, etc.
- Shape of array is tuple of integers giving size of array along each dimension.



Structure of NumPy Arrays

- Data Pointer is the memory address of first byte in array and data type description is kind of elements in array.

- Shape of array like (8,8) for a eight-by-eight array or (6,6,6) for a block of data describing grid of x, y, z coordinates.

- Strides are the number of bytes to skip in memory to proceed to the next element. For (8,8) array of bytes, strides may be (8,2) i.e. proceed 2 bytes to go to next column and 8 bytes to next row.

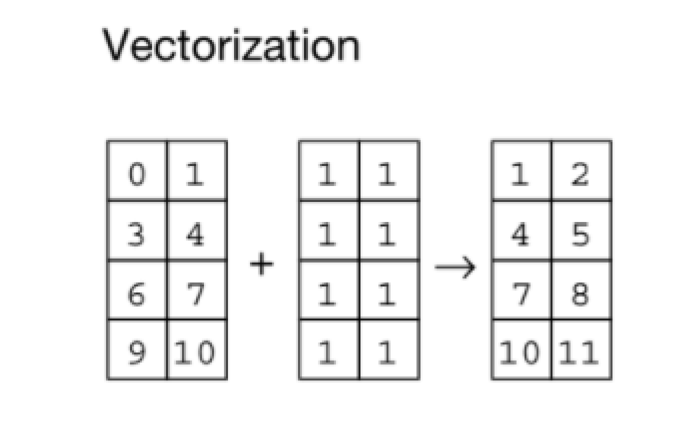
- Flag defines whether we are allowed to modify array whether memory layout is C or Fortran-contiguous.

Vectorization

- Grouping these element-wise operations together to prevent excessive use of for loops to avoid poor performance is called Vectorization.

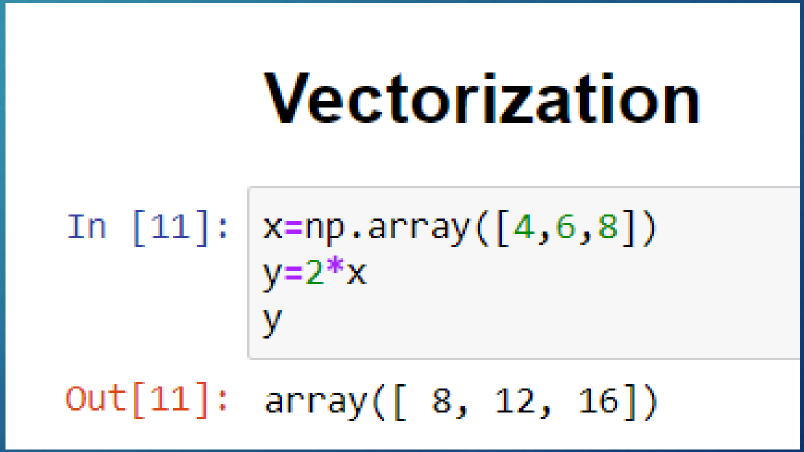
- Allows NumPy to perform such computations much more rapidly.

- This results in concise code and frees users to focus on the details of their analysis, while NumPy handles looping over array elements near-optimally.



Vectorization

$$\begin{bmatrix}0&1\\3&4\\6&7\\9&10\end{bmatrix} + \begin{bmatrix}1&1\\1&1\\1&1\\1&1\end{bmatrix} \rightarrow \begin{bmatrix}1&2\\4&5\\7&8\\10&11\end{bmatrix}$$

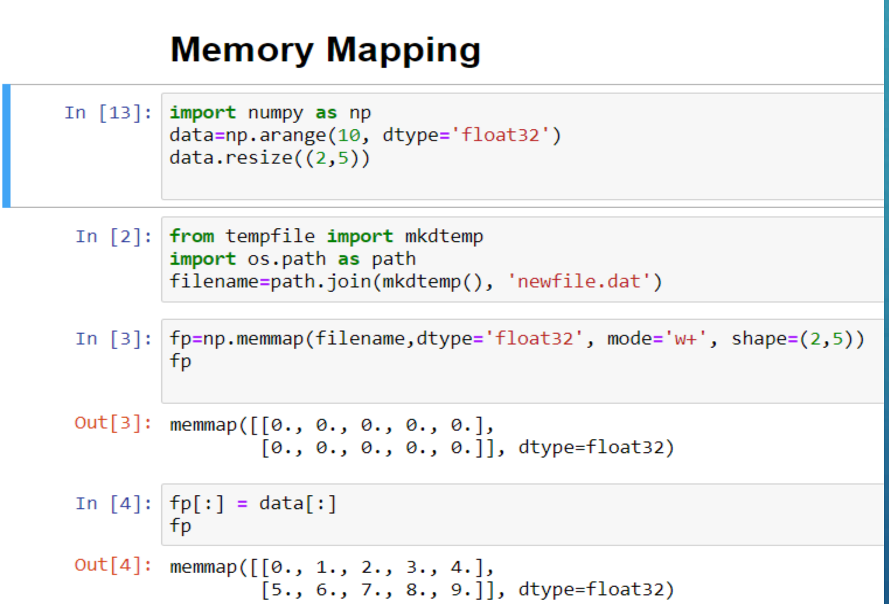


Vectorization

```
In [11]: x=np.array([4,6,8])
         y=2*x
         y

Out[11]: array([ 8, 12, 16])
```

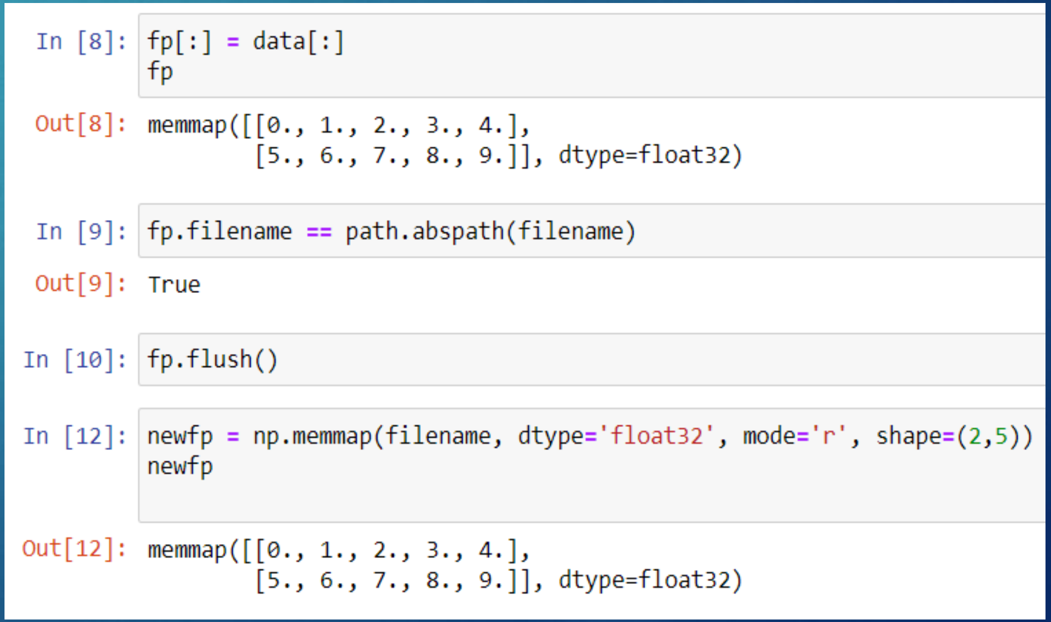
Memory Mapping

- Addressing an array on a disk without copying it to memory in its entirety is called Memory Mapping.
- Useful for addressing small portion of very large array.
- Supports memory mapped arrays with same interface as with other NumPy array.



Memory Mapping

```
In [13]: import numpy as np
         data=np.arange(10, dtype='float32')
         data.resize((2,5))

In [2]:  from tempfile import mkdtemp
         import os.path as path
         filename=path.join(mkdtemp(), 'newfile.dat')

In [3]:  fp=np.memmap(filename,dtype='float32', mode='w+', shape=(2,5))
         fp

Out[3]:  memmap([[0., 0., 0., 0., 0.],
                 [0., 0., 0., 0., 0.]], dtype=float32)

In [4]:  fp[:] = data[:]
         fp

Out[4]:  memmap([[0., 1., 2., 3., 4.],
                 [5., 6., 7., 8., 9.]], dtype=float32)
```



```
In [8]:   fp[:] = data[:]
          fp

Out[8]:   memmap([[0., 1., 2., 3., 4.],
                  [5., 6., 7., 8., 9.]], dtype=float32)

In [9]:   fp.filename == path.abspath(filename)

Out[9]:   True

In [10]:  fp.flush()

In [12]:  newfp = np.memmap(filename, dtype='float32', mode='r', shape=(2,5))
          newfp

Out[12]:  memmap([[0., 1., 2., 3., 4.],
                  [5., 6., 7., 8., 9.]], dtype=float32)
```

Array Aware Functions

- Used for creating, reshaping, concatenating and padding arrays; searching, sorting and counting data and reading and writing files.

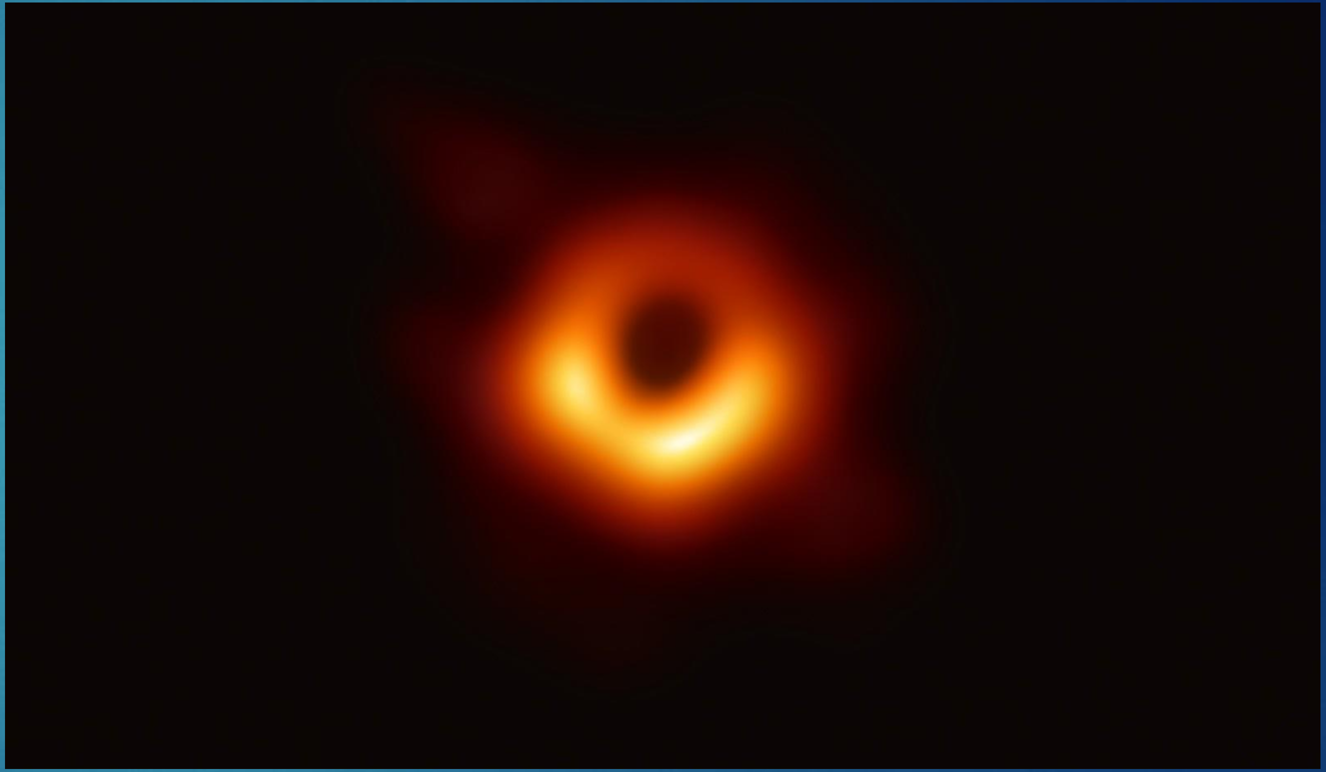
- Provides extensive support for generating pseudorandom numbers, including an assortment of probability distributions.

- Performs accelerated linear algebra using one of several backends such as Open BLAS or Intel MKL .

First Image of a Black Hole

- The Event Horizon Telescope (EHT) is an array of eight ground-based radio telescopes forming a computational telescope the size of the Earth.

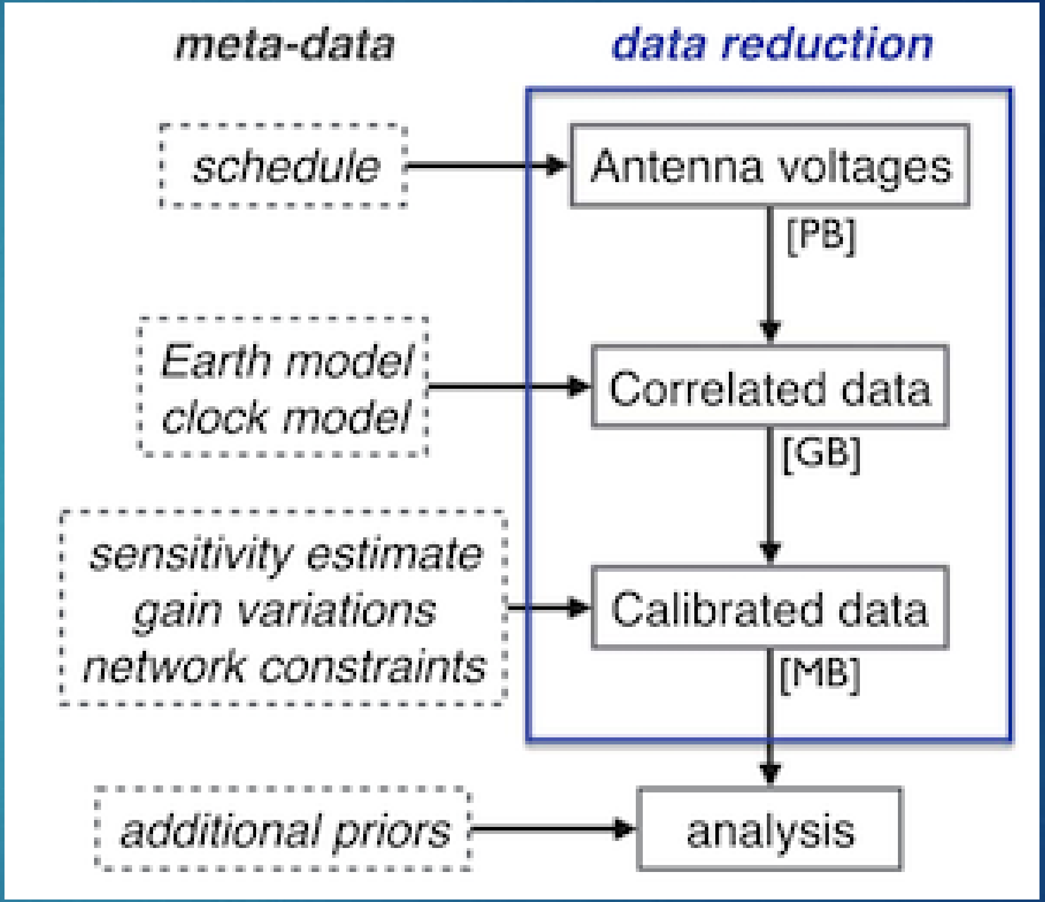
- It uses technique called as Very Long Baseline Interferometry (VLBI) and has an angular resolution of 20 micro-arcseconds.



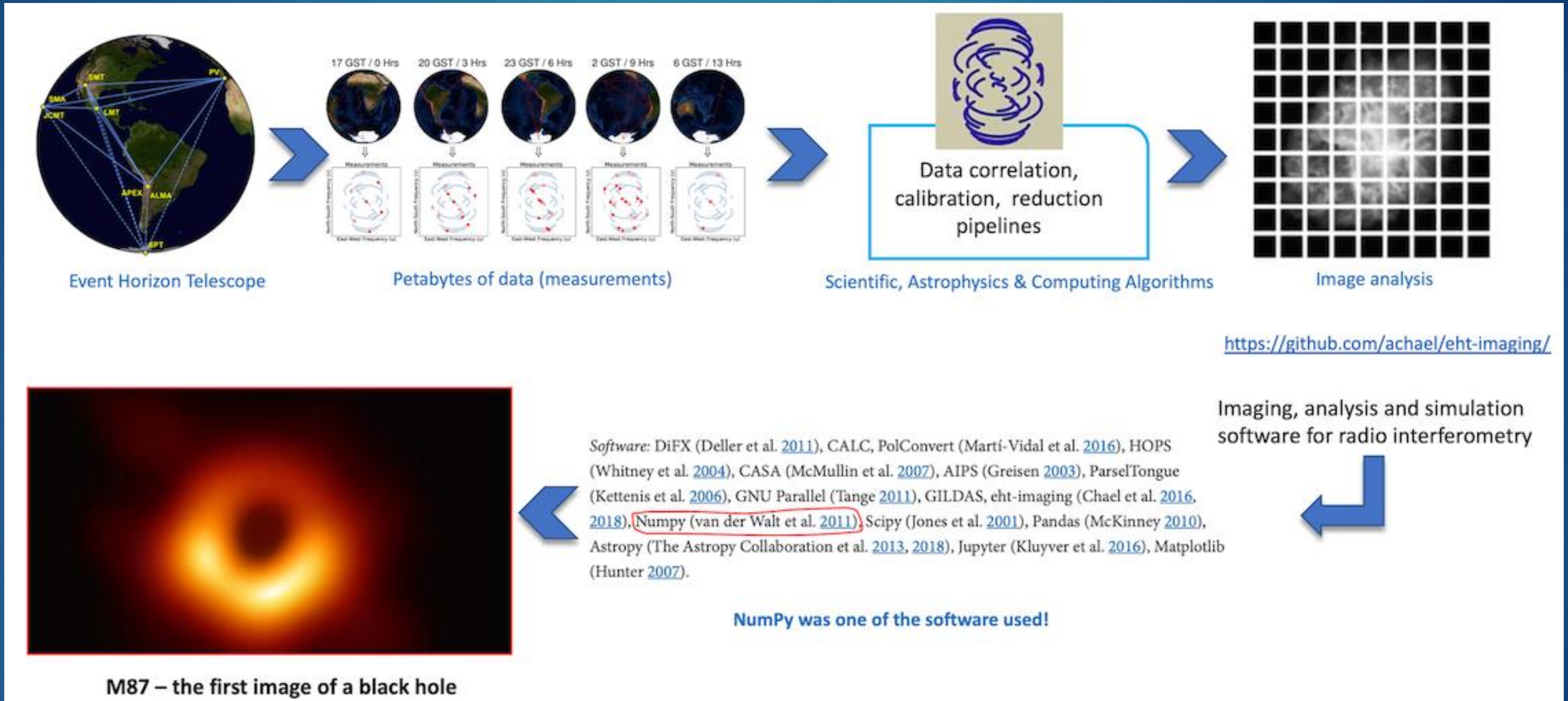
Challenges

- EHT poses massive data processing challenges such as atmospheric phase fluctuations, large recording bandwidth and much more.

- EHT generates over 350 terabytes of observations and stored on Helium-filled hard drives. Reducing the volume and the complexity of this data is very difficult.



NumPy’s Role

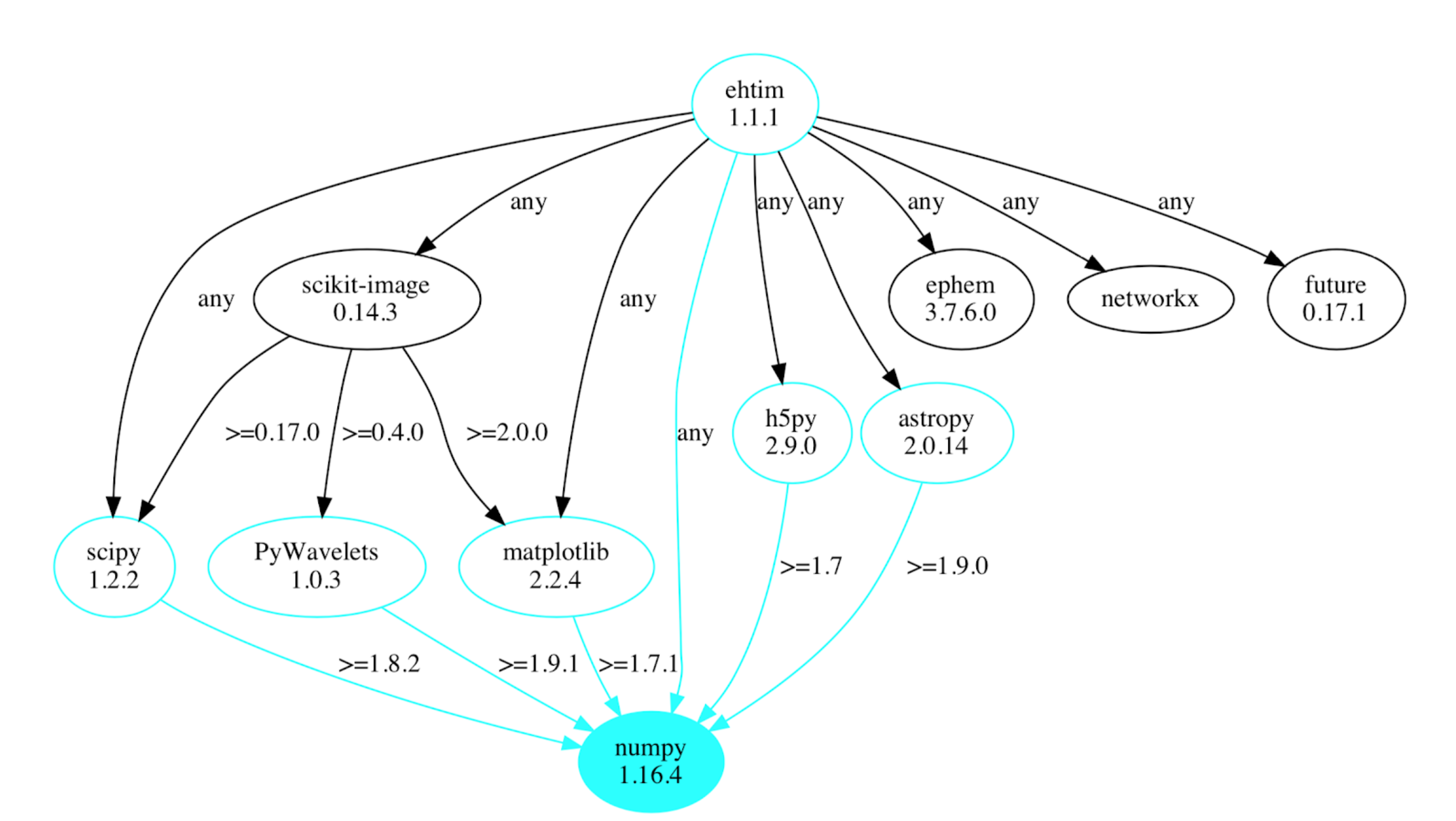

SMT
PV
SMA
JCMT
LMT
APEX
ALMA
SPT
Event Horizon Telescope
17 GST / 0 Hrs
20 GST / 3 Hrs
23 GST / 6 Hrs
2 GST / 9 Hrs
6 GST / 13 Hrs
Petabytes of data (measurements)
Data correlation, calibration, reduction pipelines
Scientific, Astrophysics & Computing Algorithms
Image analysis
https://github.com/achael/eht-imaging/
Imaging, analysis and simulation software for radio interferometry

Software: DiFX (Deller et al. 2011), CALC, PolConvert (Martí-Vidal et al. 2016), HOPS (Whitney et al. 2004), CASA (McMullin et al. 2007), AIPS (Greisen 2003), ParselTongue (Kettenis et al. 2006), GNU Parallel (Tange 2011), GILDAS, eht-imaging (Chael et al. 2016, 2018), Numpy (van der Walt et al. 2011), Scipy (Jones et al. 2001), Pandas (McKinney 2010), Astropy (The Astropy Collaboration et al. 2013, 2018), Jupyter (Kluyver et al. 2016), Matplotlib (Hunter 2007).

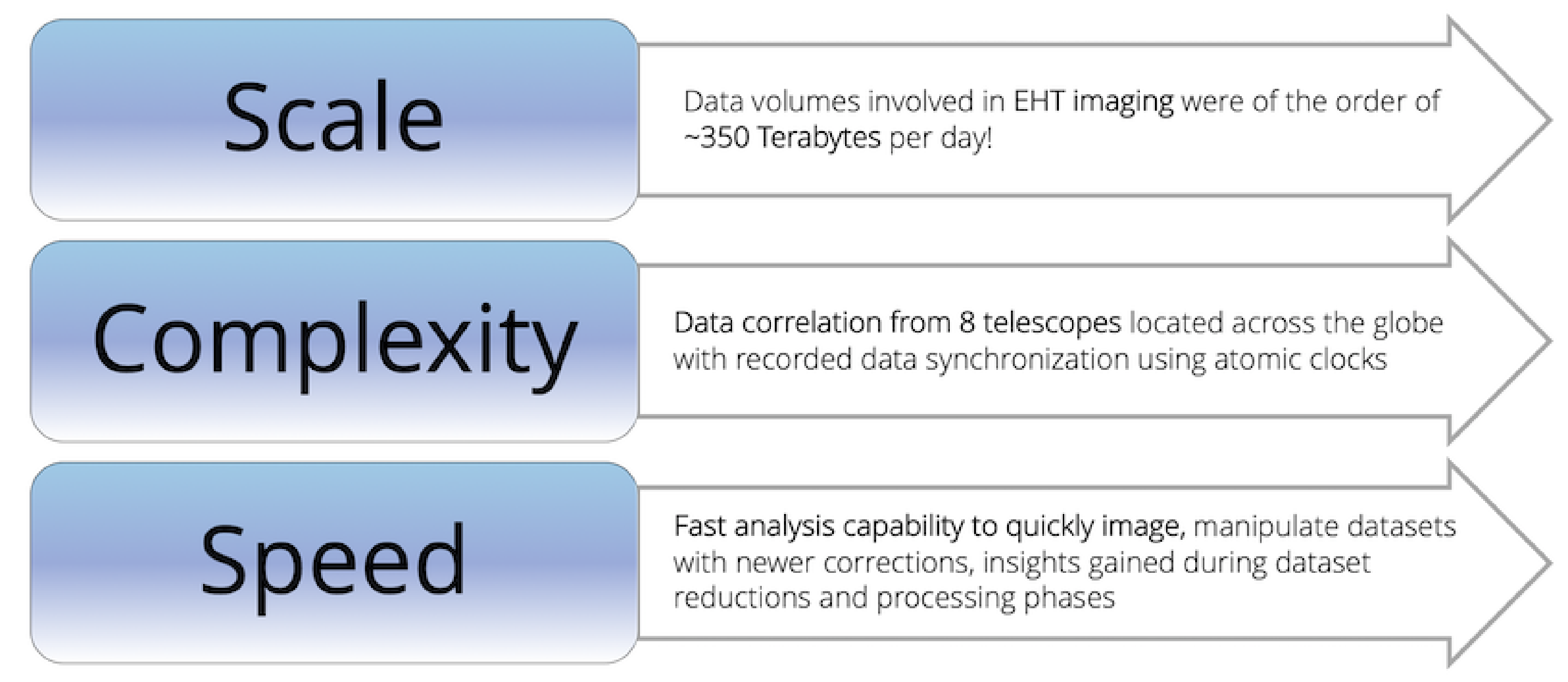
NumPy was one of the software used!

M87 – the first image of a black hole

Software Dependency Chart of ehtim package



NumPy Capabilities



Scale

Data volumes involved in EHT imaging were of the order of ~350 Terabytes per day!

Complexity

Data correlation from 8 telescopes located across the globe with recorded data synchronization using atomic clocks

Speed

Fast analysis capability to quickly image, manipulate datasets with newer corrections, insights gained during dataset reductions and processing phases

Array Proliferation

- Excessive large size of the scientific datasets require them to be stored on multiple machines or in the cloud.
- Recent need to accelerate deep learning and AI applications has resulted in using more accelerator hardware like GPUs, TPUs and FPGAs.
- Inability of NumPy to directly utilize the storage and specialized hardware has resulted in proliferation of new array implementations.
- Each DL framework now has its own arrays like PyTorch, TensorFlow, Apache, MXnet, JAX arrays.

Array Interoperability

- Operating on specialized arrays using NumPy functions and semantics.
- Users would only need to write code once and then would benefit from switching between NumPy arrays, GPU arrays, distributed arrays as suitable.
- To facilitate this interoperability, NumPy provides protocols, implemented by libraries like Dask, CuPy, xarray, PyData.
- Users can scale their computations on distributed, multi-GPU systems using NumPy's high level API's.

Discussion and Future Prospects

- NumPy combines power of array, performance of C and versatility of Python.
- NumPy is the standard API for tensor computation and central coordinating mechanism between array types and technologies in Python.
- In the coming decade, NumPy developers will face challenges and amount of data and scale will increase.
- New generation languages like Rust, Julia and LLVM will create new concepts and data structures.

References

- https://numpy.org/doc/stable/index.html
- https://www.datacamp.com/community/tutorials/python-numpy-tutorial
- https://www.nature.com/articles/s41586-020-2649-2
- https://arxiv.org/abs/2006.10256
- https://ieeexplore.ieee.org/document/5725236
- https://numpy.org/case-studies/blackhole-image/