Improving NumPy for Better Data Science
Progress Report (II)

Stéfan J. van der Walt, Sebastian Berg, Jarrod Millman,
Fernando Pérez, Matti Picus, Tyler Reddy, Warren Weckesser

Berkeley Institute for Data Science (BIDS)
University of California, Berkeley

October 2018–October 2019

Alfred P. Sloan Foundation Grant G-2017-9960
https://sloan.org/grant-detail/8222

Gordon and Betty Moore Foundation Grant GBMF5447
https://www.moore.org/grant-detail?grantId=GBMF5447

Principal Investigator: Stéfan J. van der Walt

These grants support improvements to NumPy, a library for the Python scientific programming language. NumPy provides support for numerical programming in Python, which is critical for data-driven research.

Contents

Introduction 3
Personnel 3
Technical 3
Random Number Generator Policy 3
New Data Types 4
Test Infrastructure 4
Removal of Financial Functions 5
Recommend Python and Numpy version support 5
Process Improvements 5
External distribution modifications 5
Social 6
Community Building 6
Outreachy Internship 6
Events 6
SciPy Paper 6
Documentation 6
Conferences 7
Contribution Statistics 7

Introduction

This report marks the beginning of the final year of the grant, which ends October 2020. Below, we report that we are well on our way to satisfying all technical aims, and have made progress on each social aim.

This cycle also sees a change in approach: we focus on a few select larger scope items, the kind that would be harder to track as a volunteer contributor with fractured time commitment, and otherwise engage mainly in activities that support the community and frees up core developer time: review, testing, cleanup, and coordination.

Social Aims

S1 Improve Community Engagement

S2 Grow Core Team, Add Contributors

S3 Diversify Contributors

Technical Aims

T1 More Flexible & Sustainable Code

T2 Frequent & Consistent Releases

T3 Improve Data Type System

T4 New Array Protocol

Personnel

We appointed two new developers: Sebastian Berg (May 2019) and Warren Weckesser (August 2019), both of whom had a long history of contributing to NumPy and/or SciPy before joining the project. After a year-long sabbatical at BIDS, Tyler Reddy returned to Los Alamos National Laboratory.

Technical

Random Number Generator Policy

After a NumPy Enhancement Proposal (NEP) sprint hosted at Berkeley[1] in March 2018, Robert Kern formulated a plan for a new random number generator policy[2] as NEP 19, inspired by the prototype implementation[3] of Kevin Sheppard. The NEP was accepted in July 2018, after which we facilitated conversations with Robert and Kevin to clarify implementation details. We then integrated Kevin's implementation into NumPy[4,5].

The long-overdue revamp of this system was complicated by stringent backward-compatibility guarantees offered by NumPy on random bitstreams. The new implementation is modular and allows both for customizable bit generators[6], as well as for upgrades to underlying random distribution algorithms.

As an interesting aside, the fascinating discussion[7] around choosing a default bit generator for NumPy inspired[8] the PCG author to design a new member of the PCG family.

It also led to NumPy including a sophisticated seed generator, that in turn simplifies the reproducible spawning of several independent

[1] https://scisprints.github.io

[2] https://numpy.org/neps/nep-0019-rng-policy.html

[3] https://bashtage.github.io/randomgen/

[4] github:numpy/numpy#13164

[5] github:numpy/numpy#13163

[6] https://github.com/numpy/bitgenerators

[7] github:numpy/numpy#13635

[8] http://bit.ly/new-pcg-bitgen

bitstreams[9].

New Data Types

We are busy overhauling the datatype system to make it consistent (aligned with expectations), to simplify the creation of custom dtypes, and to enable much requested features such as units. The new implementation should be simpler and more cleanly implemented.

A design document[10] is under development in consultation with the community. In November, we are hosting another developer meeting at which the proposed design will be reviewed.

The data type implementation will require extensive modifications to NumPy. A challenge with such large-scale changes is that they are hard to review, but we have general agreement from the community that the modifications can be made as long as they satisfy backward compatibility, the existing test suite, and a full test suite for proposed changes. All planning and code development happens openly, and we frequently consult with various core developers to identify any glaring design flaws. Ultimately, of course, all code will still be submitted for peer review.

Some smaller changes are also being made in anticipation of the larger refactor, including the introduction of three new operators for the `timedelta64` dtype: `divmod`[11], `floordiv`[12], and `modulus`[13]. These operators were well received by `pandas`[14], since they allowed for the removal of substantial portions of code.

Test Infrastructure

NumPy's test infrastructure was improved to better support non-x86 chipsets such as ARMv8[15] and, with help from IBM, POWER8[16,17]. IBM also now maintains our code for their architectures. We also matched builds and tests more closely with one of our primary distribution mechanisms[18].

Previously, code from docstrings (code appearing inline in function documentation) was untested, but we now test such code using using the same infrastructure as SciPy[19].

Reference counts (number of pointers held to Python objects) are now tracked to ensure that our C API allocates and de-allocates objects correctly. This, in turn, makes it easier to review the addition of large pieces of C code.

NumPy depends on OpenBLAS to provide its underlying accelerated linear algebra implementation. Prompted by a related failure[20],

[9] http://bit.ly/np-seedsequence

[10] http://bit.ly/dtype-NEP-draft

[11] github:numpy/numpy#12683

[12] github:numpy/numpy#12308

[13] github:numpy/numpy#12120

[14] http://bit.ly/pandas-on-operators

[15] github:numpy/numpy#13270

[16] github:numpy/numpy#13264

[17] github:numpy/numpy#12709

[18] github:xianyi/OpenBLAS#2124

[19] github:numpy/numpy#12253

[20] github:numpy/numpy#13401

we assisted OpenBLAS—who has only one active maintainer—to improve its testing infrastructure. Specifically, we added emulation of a modern CPU architecture[21] and native testing on a new architecture[22].

Removal of Financial Functions

Warren wrote up a proposal to remove the financial functions from NumPy (NEP 32). The proposal has been accepted. This change is notable, because NumPy has almost never in its past removed functions from its public API.

Recommend Python and Numpy version support

In NEP 29[23], we recommend, along with leaders from various other projects, that all projects across the Scientific Python ecosystem adopt a common "time window-based" policy for support of Python and NumPy versions. This standard will simplify downstream project and release planning.

Process Improvements

Previously, release notes were compiled by hand. We configured Towncrier to automatically gather snippets with detail on upcoming changes[24] into release notes[25].

Previously, changes to test dependencies could break the test. We now pin those packages and receive a weekly update Pull Request via DependencyBot[26] that can easily be checked for regressions.

External distribution modifications

The NumPy project releases binary packages, but the source code is also used by the Intel Distribution for Python[27], Anaconda[28], and conda-forge[29].

A bug report to NumPy revealed[30] that the version of the library shipped with Intel's distribution had a different API—i.e., contained newly added functions. This prompted us to start a conversation with Intel. While the issue of API modification has not resolved, Intel is now working with NumPy to reduce duplicated work. They have published their version[31] on GitHub.com to simplify tracking differences and filing issues. As of this writing, 33 issues have been filed on NumPy with the Intel/Anaconda label[32].

[21] github:xianyi/OpenBLAS#2134

[22] github:xianyi/OpenBLAS#2121

[23] https://numpy.org/neps/nep-0029-deprecation_policy.html

[24] http://bit.ly/np-upcoming-changes

[25] https://numpy.org/devdocs/release/1.18.0-notes.html

[26] https://github.com/marketplace/dependabot-preview

[27] https://software.intel.com/en-us/distribution-for-python

[28] https://www.anaconda.com/

[29] https://conda-forge.org

[30] github:numpy/numpy#12512

[31] https://github.com/IntelPython/numpy

[32] http://bit.ly/np-intel-anaconda

Social

Community Building

In 2018, we started weekly calls to update the community on our progress. These meetings have now become *community calls* that are well attended by core developers, community members, and industry. Meeting notes are published on GitHub[33].

Ralf Gommers started a new position as the director of QuanSight Labs this year, and focuses on many of the same issues we do for this grant. As such, we have been meeting regularly to coordinate efforts.

Together, we presented a talk at SciPy2019 titled "Inside NumPy: Preparing for the Next Decade"[34], in which we examine the impact funding has had on the project, as well as challenges that remain.

Outreachy Internship

Outreachy is an internship program that supports diversity in free and open source software. This year, we sponsored Kriti Singh through their program, and provided mentorship for her work on technical documentation during the summer. Kriti wrote a blog post about her experience attending EuroSciPy2019[35] as part of her internship.

Events

We hosted two developer sprints this past year[36,37] and plan a "Tensor Developer Summit" for Spring 2020.

SciPy Paper

We helped the SciPy community to prepare their manuscript *SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python*[38]. The paper contains a background section that also covers some of the origins of NumPy.

Documentation

The documentation build system was cleaned up to run without warnings, and the continuous integration (CI) system now flags problematic contributions. Documentation, which was previously hosted at https://docs.scipy.org/doc (sponsored by Enthought), was moved to https://numpy.org/doc, hosted by GitHub and cached by CloudFlare.

[33] https://github.com/numpy/archive/tree/master/status_meetings

[34] https://youtu.be/dBTJD_FDVjU

[35] http://bit.ly/kriti-euroscipy

[36] http://bit.ly/numpy-2018-12

[37] http://bit.ly/np-2018-05

[38] https://arxiv.org/abs/1907.10121

Ralf Gommers and Inessa Pawson are leading an effort to redesign `numpy.org`. To assist, we hired a contractor to do theme design and content development.

Conferences

We attended, gave talks and tutorials, and led sprints at numerous conferences this year, including PyCon USA (Picus), Pycon Israel (Picus[39]), SciPy Austin (Gommers, Picus, Reddy, Van der Walt[40]), and EuroSciPy (Picus[41,42]).

Contribution Statistics

Of 913 Pull Requests (PRs) opened and merged since October 2018, the team at BIDS created 243, and merged another 321. Of the remaining PRs, BIDS made at least half the non-author comments on 63, and commented on another 78. In total, the team therefore was involved in about 705 of the 913 PRs.

[39] https://youtu.be/YFLVQFjRmPY

[40] https://youtu.be/dBTJD_FDVjU

[41] https://pretalx.com/euroscipy-2019/talk/NQMWSX/

[42] https://pretalx.com/euroscipy-2019/talk/R3TJLP/