Preface

Stéfan van der Walt, Jarrod Millman

Introduction

Welcome to the fourth publication of the SciPy conference proceedings. This year marks the 10th Python in Science Conference (SciPy2011), held in Austin, Texas from July 11th to 16th where more than 190 participants, in a truly global attendance from North and South America, Europe, Africa, and Asia, represented both academia and industry.

Ten years and counting ...

The first "Python for Scientific Computing Workshop" was held at Caltech in 2002—a single track, 2-day workshop with about 70 attendees. Since then, the number of attendees nearly tripled; the 2-days of talks are now preceded by 2 days of tutorials and followed by 2 days of developer sprints. Multiple conference tracks accommodate the increasing number of high-quality talk submissions, and satellite conferences are held annually in Europe and India. We give a brief overview of these developments:

In 2004 the workshop was renamed the SciPy conference and extended by 2 days of developer sprints; the number of attendees increased to about 90. The first keynote was given by Jim Hugunin, creator of numeric (the precursor to NumPy), Jython, IronPython, and co-designer of AspectJ.

In 2005, the community was split between two different core numerical libraries—numeric and numarray, making code exchange and shared development hard. Travis Oliphant, who had been working on a successor to both libraries, opened the conference with a presentation of the new core package now known as NumPy. In order to move the new library forward, the number and length of talks were limited, and additional break sessions were held for informal discussions.

By 2006, NumPy approached a 1.0 release, with support from a reunited community. Guido van Rossum, the creator of Python, delivered the keynote, and tutorials appeared for the first time. The number of attendees grew to 138.

The 2007 conference included the first open call for student sponsorship, which has continued and expanded every year since. Ivan Krstić, then the director of security architecture at One Laptop per Child, gave the keynote.

The 2008 keynote was delivered by Alex Martelli, "Über Tech Lead" for Google, perhaps better known in the Python community as the MartelliBot. A peer-reviewed conference proceedings appeared, and EuroSciPy was held for the first time.

In 2009, keynotes were delivered by Peter Norvig, Director of Research at Google, and Jon Guyer, Materials Scientist at NIST. In addition to the second European SciPy conference, the 1st SciPy India was held.

The 2010 conference moved to Austin, TX, and included multiple tracks and 187 participants. Keynotes were delivered by David Beazley (the author of the Python Essential Reference and creator of SWIG) and Travis Oliphant (then President of Enthought, Inc. and one of the original co-authors of SciPy).

2011 Conference

This year's conference consisted of several events: tutorials, followed by the main conference and sprints, with some birds-of-a-feather sessions sprinkled throughout. Aric Hagberg and the NetworkX team also held a satellite workshop preceding the main conference.

Tutorials

Tutorials were divided into an introductory and an advanced track. This year's introductory track started with Jonathan Rocher's overview of NumPy, IPython and matplotlib, followed by an introduction to SciPy by Anthony Scopatz. Mateusz Paprocki and Aaron Muller, both from SymPy, showed how to use their project to do symbolic computation, and Corran Webster from Enthought illustrated the power of matplotlib, Traits and Chaco. Traits, of course, being Enthought's popular GUI building tool.

On the advanced end we had Chris Fonnesbeck and Abie Flaxman discuss PyMonteCarlo for statistical modeling, followed by Gaël Varoquaux's overview of scikits.learn, a collection of machine learning algorithms. On the parallel computing front Jeff Daily spoke about the Global Arrays Toolkit.

The last tutorial was given by Min Ragan-Kelley on using IPython to perform interactive parallel computing.

Keynote speakers

The first keynote address was delivered by Eric Jones, a founder of Enthought and one of the first SciPy authors. In his talk, titled "What matters in Scientific Software Projects: 10 Years of Success and Failures Distilled", Eric touched on some of the lessons he has gleaned over the past decade. For example, he mentioned the rare skills intersection that we currently have between scientists or engineers and computer

Stéfan van der Walt and Jarrod Millman are with the University of California, Berkeley. E-mail: stefan@sun.ac.za.

scientists, and how valuable those skills are in today's world. He also emphasized the importance of engaged participation with your customer or users, in order to develop practical and useful software specification as well as the trust required to build such relationships. Finally, he warned against overemphasizing development process, which he argued often has less of an impact on success than frequently believed: Process cannot substitute for intelligent reasoning brought to a project by talented people.

The second Monday plenary, "How the Digital Age Affects Research" was delivered by Kaitlin Thanay. Kaitlin has an interesting background as manager of the science division of Creative Commons and is now with Digital Science, a company spun out of Nature. She started her talk by introducing what some think to be the first computation machine: the Antikythera mechanism. Her talk then followed the interaction of science and technology as both developed, and how this influences the research cycle.

On Wednesday morning, Hilary Mason from bit.ly spoke about "Science in the Service of Awesome". bit.ly generates about 1% of all new URLs that appear on the internet per day, and hosts more than 10**9 unique URLs. Interestingly, they do quite a bit of data science and generate statistics on things like click-throughs etc.

The last plenary was given by Perry Greenfield, who is a familiar face at SciPy conferences. Perry leads the Science Software Branch of the Space Telescope Science Institute, and is one of the pioneers of Python in astronomy. His talk, "How Python Slithered into Astronomy", a follow-up of his talk delivered at SciPy India last year, looked at the history of Python in this very active field. The astronomy community has been and continues to be central to the development of tools such as NumPy and matplotlib.

Talks

The conference had three sessions this year, and at any one time two ran in parallel. Anthony Scopatz chaired the Python and Core Technologies session, which was an effort to bring science related talks submitted to PyCon to this conference. Peter Wang, author of Chaco, hosted the session on data science.

Highlighting the growing role of Python in statistical computing was Wes McKinney's talk titled "Time Series Analysis in Python with *statsmodels*". Wes, Skipper Seabold, Josef Perktold, Jonathan Taylor, and others have made huge improvements in the statistics tools available for Python, as witnessed by the growing number of statistical talks. In addition, there was also a birds-of-a-feather session, following up from last year's on DatArray, discussing the best Python data structures for representing statistical data.

Scott Determan's Vision Spreadsheet was particularly enjoyable to watch in action, which can be found here:

```
http://visionspreadsheet.com
```

After a couple of years' silence from the IPython team, Fernando Perez's report on their recent progress was well received. He introduced several new features in the recently released IPython 0.11, such as the new Qt and web-based front-ends as well as the parallel computation machinery backed by 0MQ.

Two other talks from the Python Core Technologies talks that deserve special mention were "Twiggy: A Pythonic Logger" by Pete Fein, and "Lessons Learned Building a Scalable Distributed Storage System in Python" by Chuck Thier. The first, because of long-standing issues with the Python standard library's logging module, and the second because it is great to know that you can implement things like Distributed Storage Systems in Python.

Gaël Varoquaux gave a great talk on Python for Brain Mining. Of particular interest to the larger community was Joblib—a lightweight system for building scientific pipelines. In the quest for reproducibility, this is a great find!

Then, on the business side of things, Josh Heman from Sports Authority spoke about the challenges of getting Python into a multi-billion dollar enterprise.

As usual, we also had the very entertaining lightning talks session on the last day of the conference. Travis Oliphant's talk on "What I would like to see in NumPy" should probably have made its way into the main conference, so find his slides online if you'd like to have a look at some of the ideas that may well form part of NumPy 2.0.

The slides and videos for all these talks and many others are available from the conference website:

```
http://conference.scipy.org/scipy2011
```

Sprints

With open source development, bringing developers to the same physical location can be challenging. The sprints provide an ideal opportunity to overcome this difficulty, during which developers can have face-to-face conversations and directly work together to solve long-standing problems or implement exciting new features. It is interesting that, in today's connected world, these real-world conversations still have such a large impact.

Special issue: IEEE Computing in Science and Engineering

Attendees this year were given copies of the March/April 2011 issue of Computing in Science and Engineering [Millman]. This was a special issue on Python for Scientists and Engineers Computing in Science & Engineering (CiSE). CiSE is a peer-reviewed technical magazine jointly published by the American Institute of Physics and the IEEE Computer Society.

This issue was a follow-up to May/June 2007 special issue, "Python: Batteries Included" [Dubois]. The 2007 issue had a couple articles presenting the core Python libraries for scientific computing: NumPy, SciPy, IPython, and matplotlib as well as a series of shorter pieces presenting specific scientific, engineering, and educational applications. This year's special issue had fewer, but longer articles focusing on some of the more advanced features of the core stack of scientific tools for Python [Perez], [VanderWalt], [Behnel], [Ramachandran].

Proceedings

Each of the submitted abstracts was reviewed by both the program chairs and two additional members of the program committee. The committee consisted of members from several countries, and represented both industry and academia.

Abstracts were evaluated according to the following criteria: applicability, novelty, and general impression.

At the end of the conference, we invited the presenters to submit their work for publication in the proceedings. Each reviewer provided an overall score for each reviewed paper, based on the quality of approach and writing. Reviewers were also asked to provide more specific feedback according to the questionnaire shown in the appendix.

The tools used to produce this document are made available under an open source license, and may be obtained from the code repository at https://github.com/scipy/scipy-proceedings.

After several years of following this review process, we've found that this review process was too onerous for the volunteers helping make these proceedings possible. In following years, we will be implementing a more streamlined process that leverages GitHub's code review machinery. Hopefully, this will enable us to produce the proceedings in a more timely manner.

Acknowledgments

A conference the size of SciPy is only possible through the hard work and dedication of a large number of volunteers. Once again Enthought provided significant administrative support. We also thank the numerous sponsors (listed on the conference website).

These proceedings are the result of many hours of work by authors and reviewers alike. We thank them for their significant investment in these manuscripts. The names of all contributers are listed in the "Organization" section, which forms part of the cover material.

Appendix: Reviewer Questionnaire

- Is the code made publicly available and does the article sufficiently describe how to access it? We aim not to publish papers that essentially advertise proprietary software. Therefore, if the code is not publicly available, please provide a one- to two- sentence response to each of the following questions:
  - Does the article focus on a topic other than the features of the software itself?
  - Can the majority of statements made be externally validated (i.e., without the use of the software)?
  - Is the information presented of interest to readers other than those at whom the software is aimed?
  - Is there any other aspect of the article that would justify including it despite the fact that the code isn't available?
  - Does the article discuss the reasons the software is closed?
- Does the article present the problem in an appropriate context? Specifically, does it:
  - explain why the problem is important,
  - describe in which situations it arises,
  - outline relevant previous work,
  - provide background information for non-experts
- Is the content of the paper accessible to a computational scientist with no specific knowledge in the given field?
- Does the paper describe a well-formulated scientific or technical achievement?
- Are the technical and scientific decisions well-motivated and clearly explained?
- Are the code examples (if any) sound, clear, and well-written?
- Is the paper factually correct?
- Is the language and grammar of sufficient quality?
- Are the conclusions justified?
- Is prior work properly and fully cited?
- Should any part of the article be shortened or expanded?
- In your view, is the paper fit for publication in the conference proceedings? Please suggest specific improvements and indicate whether you think the article needs a significant rewrite (rather than a minor revision).

References

[Dubois] Dubois, Paul F. "Guest editor's introduction: Python: batteries included." Computing in Science & Engineering 9.3 (2007): 7-9.

[Millman] Millman, K. Jarrod, and Michael Aivazis. "Python for scientists and engineers." Computing in Science & Engineering 13.2 (2011): 9-12.

[Perez] Pérez, Fernando, Brian E. Granger, and John D. Hunter. "Python: an ecosystem for scientific computing." Computing in Science & Engineering 13.2 (2011): 13-21.

[VanderWalt] Van der Walt, Stéfan, S Chris Colbert, and Gaël Varoquaux. "The NumPy array: a structure for efficient numerical computation." Computing in Science & Engineering 13.2 (2011): 22-30.

[Behnel] Behnel, Stefan, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn, and Kurt Smith. "Cython: The best of both worlds." Computing in Science & Engineering 13.2 (2011): 31-39.

[Ramachandran] Ramachandran, Prabhu, and Gaël Varoquaux. "Mayavi: 3D visualization of scientific data." Computing in Science & Engineering 13.2 (2011): 40-51.