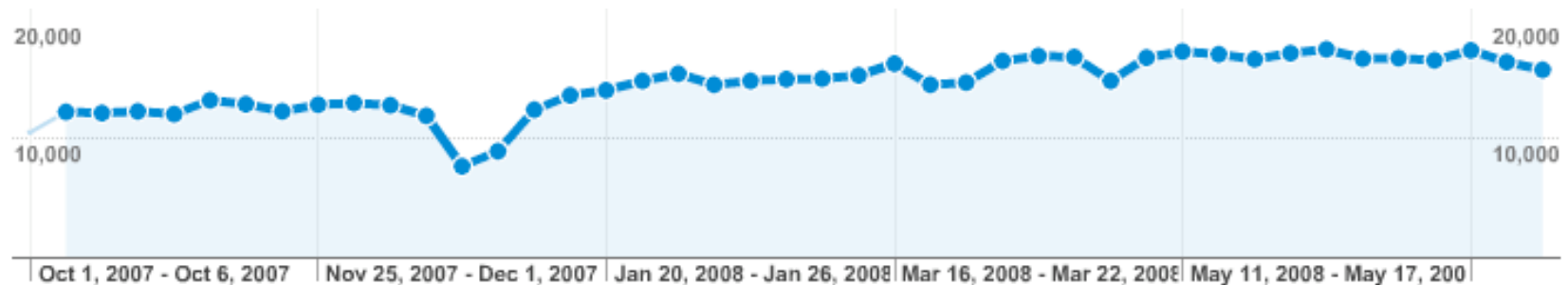# State of SciPy

Jarrod Millman
Travis N. Vaught

August 21, 2008

Overview

- Introduction (community statistics)
- Past: Where have we been
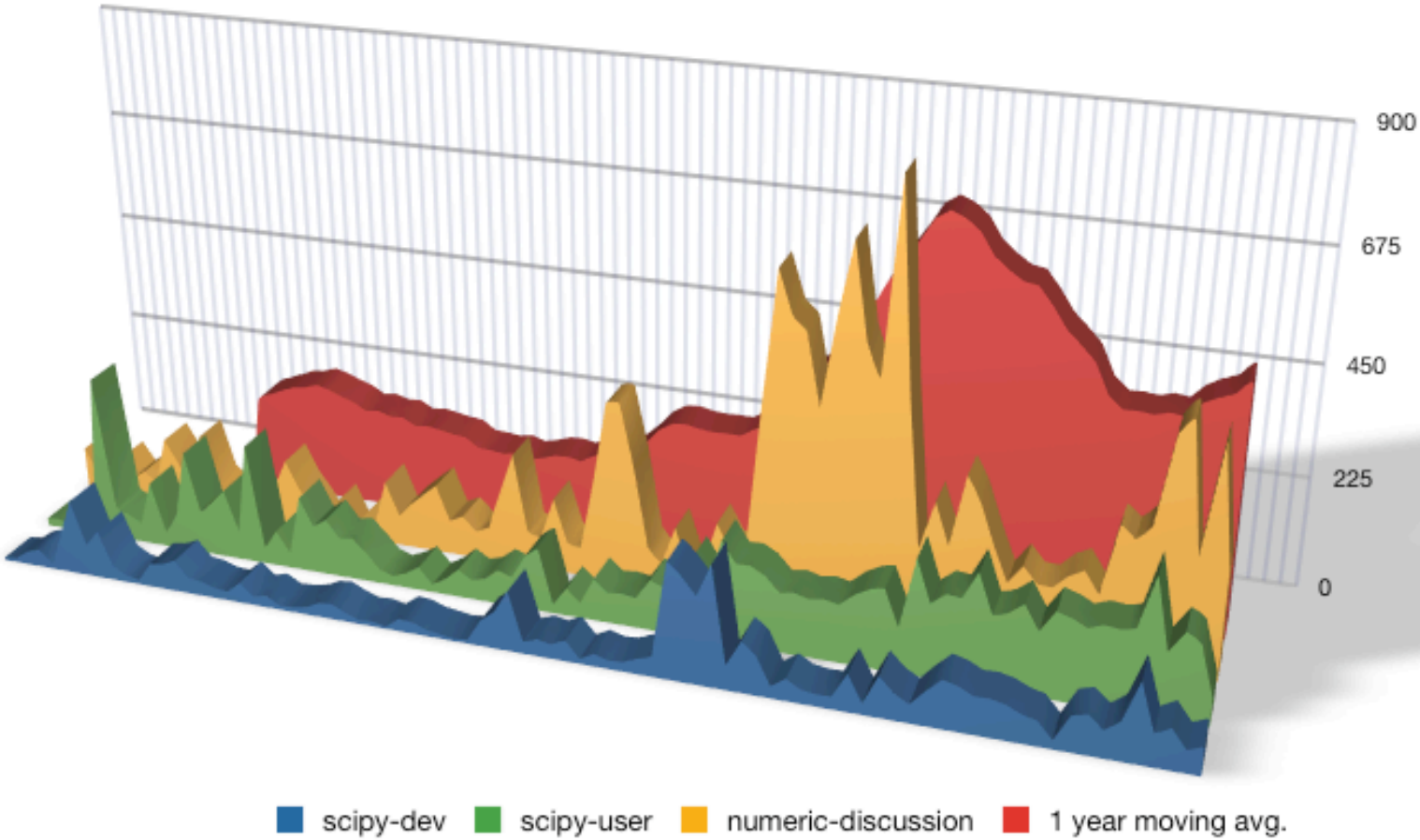- Present: Where are we
- Future: Where are we going?

scipy.org Activity

- There has been marked growth in the community of users around scientific computing in Python in the past year.

Mailing List Activity

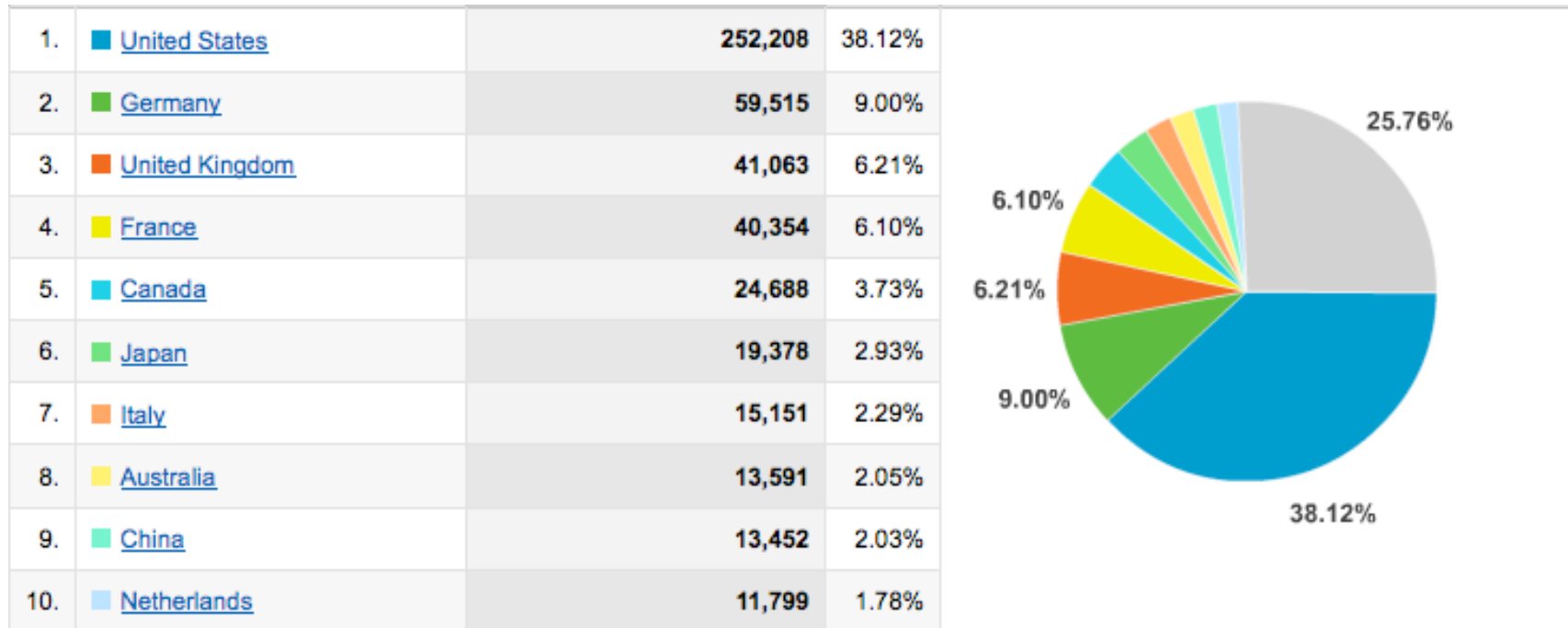Monthly Mailing List Traffic (KBytes) August 2001 - August 2008

The SciPy Community is Global

scipy.org Usage by Country

- Visits to scipy.org since Sept. 2008

1.	United States		252,208	38.12%
2.	Germany		59,515	9.00%
3.	United Kingdom		41,063	6.21%
4.	France		40,354	6.10%
5.	Canada		24,688	3.73%
6.	Japan		19,378	2.93%
7.	Italy		15,151	2.29%
8.	Australia		13,591	2.05%
9.	China		13,452	2.03%
10.	Netherlands		11,799	1.78%

25.76%

6.10%

6.21%

9.00%

38.12%

Student Sponsorship

David Cournapeau	Japan
Stefan van der Walt	South Africa
Rahul Garg	Alberta
Seshadri Tirunillai	USC
Amir Khosrowshahi	Berkeley
Dav Clark	Berkeley
Gabriel Gellner	Toronto
Min Ragan-Kelley	Berkeley
Ryan May	OU
David Warde-Farley	Toronto

Many thanks to the sponsoring organizations!

python™   ENTHOUGHT
SCIENTIFIC COMPUTING SOLUTIONS

Short History

- 1994 — Python Matrix object (Jim Fulton)
- 1995 — Numeric born (Jim Hugunin, Konrad Hinsen, Paul Dubois, David Ascher, Jim Fulton)
- 2000 — Numeric moves to sourceforge (Project registered as numpy)
- 2001 — SciPy born (Pearu Peterson, Travis Oliphant, Eric Jones)
- 2001 — IPython born (Fernando Perez)
- 2002 — SciPy '02 - Python for Scientific Computing Workshop
- 2003 — matplotlib born (John Hunter)
- 2003 — Numarray (Perry Greenfield, J. Todd Miller, Rick White, Paul Barrett)
- 2006 — NumPy 1.0 Released

Code Swarm

- Now for something completely different

NumPy Book Announcement

Present: Where we are

- NumPy/SciPy packages
- Community involvement
- Larger ecosystem

2007-2008 NumPy/SciPy Releases

- SciPy 0.6.0 (September 2007)
- NumPy 1.0.4 (November 2007)
- NumPy 1.1.0 (May 2008)
- NumPy 1.1.1 (August 2008)

Major Changes

- New MaskedArrays
- IO code overhaul
- Improved test coverage
- Numscons
- Focus on coding style

Release Management

- Better packaging
- Release numbering
- Buildbot

Community Involvement

- Blogs (http://planet.scipy.org)
- IRC (#scipy on freenode)
- Mailing lists
- Sprints
- Conferences

List Traffic

- Average daily message count:
  - numeric-discussion, scipy-user, scipy-dev: 44 messages/day
  - Adding in ipython and matplotlib yields another 15 per day
  - If you monitor enthought-dev you'll be reading another 31 per day
- Total: 90 messages per day
- Yes, it's hard to keep up.

Sprints

- August 2007 — SciPy 2007 sprint
- December 2007 — SciPy sprint at UCB
- February 2008 — Sage days 8 at Enthought
- March 2008 — NIPY/IPython sprint in Paris
- April 2008 — SciPy sprint at UCB
- July 2008 — MayaVi sprint at Enthought

Conferences

- PyCon 2008
- 2008 SIAM meeting
- EuroSciPy 2008
- SciPy 2008

The Larger Ecosystem

- Major releases
- Distribution
- New Tools

Major Releases

- matplotlib 0.98
- ETS 3
- MayaVi 2

Distribution

- Python Package Index
- Python(x,y)
- EPD
- Sage

Future: Where are we going?

- NumPy/SciPy packages
- The Larger Ecosystem
- Getting involved

NumPy/SciPy releases

- Upcoming releases
- Release management

NumPy 1.2

- Sphinx-based documentation!!
- Nose-based testing framework (GSoC project)
- Changes to histogram and median
- Arrayterators

NumPy 1.2 open issues

- ABI compatibility
- Generalized UFuncs
- C-API cleanups
- Release date

SciPy 0.7

- Sparse matrices improvements
- New functionality: Radial basis functions, hierarchical clustering, constants
- Nose-based testing framework
- Sandbox removed

Release Management

- Time-based releases
- Code Review
- Enhancement Proposals
- Distributed Version Control

Time- vs. Feauture-based Releases

Feature-based
A release cycle under this model is driven by deciding what features will be in the next release. Once all the features are complete, the code is stabilized and finally a release is made. Obviously this makes it relatively easy to predict what features will be in the next release, but extremely difficult to determine when the release will occur.

Time-based
A release cycle under this model is driven by deciding when the next release will be. This, of course, makes predicting when the release will be out extremely easy, but makes it difficult to know exactly what features will be included in the release.

Time-based Release Issues

- Branching
- Reviewing
- Testing
- Reverting
- Postponing
- Releasing

Time-based Releases: Branching

In order to be able to release on schedule requires that the mainline of development (the trunk) is extremely stable. This requires that a significant amount of work be conducted on branches.

Another important way to improve the quality of project and keep
the trunk in shape is to require peer code review and consensus among the core developers on which branches are ready to be merged.

Time-based Releases: Testing

A full test suite is also essential for being able to regularly release code.

Sticking to release schedule requires occasionally reverting commits.

Time-based Releases: Postponing

It also requires postponing branch merges until the branch is ready for release.

Time-based Releases: Releasing

Since time-based release management relies on a regular releases, the cost of making a release needs to be minimized. In particular, we need to make it much, much easier to create the packages, post the binaries, create the release notes, and send out the announcements.

Release Management Practices

- Code review
- Enhancement proposals
- Distributed version control

Code Review: Rietveld Tool

Rietveld Code Review Tool

Issues   Repositories

Recent Issues | Sign in with your Google Account to create issues and add comments

Issue 2946: Update NumPy API format to support updates that don't break binary compatibility

Can't Edit
Can't Publish+Mail
Start Review

Created:
5 days, 1 hour ago by Stefan

Modified:
1 day, 6 hours ago

Reviewers:
Andrew Straw

SVN Base:
http://svn.scipy.org/svn/numpy/trunk/

▼ **Description**

The current NumPy API number, stored as NPY_VERSION in the header files, needs
to be incremented every time the NumPy C-API changes. The counter tells
developers with exactly which revision of the API they are dealing. NumPy does
some checking to make sure that it does not run against an old version of the
API. Currently, we have no way of distinguishing between changes that break
binary compatibility and those that don't.

The proposed fix breaks the version number up into two counters -- one that gets
increased when binary compatibility is broken, and another when the API is
changed without breaking compatibility.

Backward compatibility with packages such as Matplotlib is maintained by
renaming NPY_VERSION to NPY_BINARY_VERSION.

▼ **Patch Set 1**

Total comments: 4

Created: 5 days, 1 hour ago   Download raw patch set

Raw unified diffs	Stats		Side-by-side diffs with inline comments	Delta from patch set	Patch
numpy/core/code_generators/generate_numpy_api.py	1 chunk	20 lines	0 comments	2	Download
numpy/core/include/numpy/ndarrayobject.h	1 chunk	31 lines	4 comments	2	Download
numpy/core/src/multiarraymodule.c	1 chunk	13 lines	0 comments	2	Download

▼ **Patch Set 2 : Updated with Andrew's suggestions.**

Created: 4 days, 16 hours ago   Download raw patch set

Raw unified diffs	Stats	Side-by-side diffs with inline comments	Delta from patch set	Patch

Code Review: Rietveld Tool, cont.

Left column:

```
25
26 /* There are several places in the code where an array of dimensions is
27  * allocated statically.  This is the size of that static allocation.
28  *
29  * The array creation itself could have arbitrary dimensions but
30  * all the places where static allocation is used would need to
31  * be changed to dynamic (including inside of several structures)
32  */
33
34 #define NPY_MAXDIMS 32
35 #define NPY_MAXARGS 32
36
37 /* Used for Converter Functions "O&" code in ParseTuple */
38 #define NPY_FAIL 0
39 #define NPY_SUCCEED 1
40
41         /* Helpful to distinguish what is installed */
42 #define NPY_VERSION 0x0100000B
```

```
43
44         /* Some platforms don't define bool, long long, or long double.
45           Handle that here.
46         */
```

```
25
26 /* There are several places in the code where an array of dimensions is
27  * allocated statically.  This is the size of that static allocation.
28  *
29  * The array creation itself could have arbitrary dimensions but
30  * all the places where static allocation is used would need to
31  * be changed to dynamic (including inside of several structures)
32  */
33
34 #define NPY_MAXDIMS 32
35 #define NPY_MAXARGS 32
36
37 /* Used for Converter Functions "O&" code in ParseTuple */
38 #define NPY_FAIL 0
39 #define NPY_SUCCEED 1
40
41 /* Helpful to distinguish what is installed */
42 /* NPY_VERSION is no longer used -- see NPY_BINARY_VERSION below. */
43 #define NPY_VERSION 0x1000000B
```

Andrew Straw 2008/08/16 07:20:56 there's no need to change the actual value of NPY_

Stefan 2008/08/16 07:33:19 On 2008/08/16 07:20:56, Andrew Straw wrote: > ther

```
44
```

Andrew Straw 2008/08/16 07:20:56 Perhaps move NPY_VERSION below the NPY_BINARY_VERS

Stefan 2008/08/16 07:33:19 On 2008/08/16 07:20:56, Andrew Straw wrote: > Perh

```
45 /* Whenever (NPY_BINARY_VERSION & NPY_BINARY_COMPATIBLE) changes,
46     external libraries that depend on NumPy need to be recompiled.
47
48    In other words, NPY_BINARY_VERSION consists of two parts:
49
50    0xMMMMKKKK
51
52    where M gets increased whenever binary compatibility is broken,
53    and K gets increased whenever the API is changed but binary
54    compatibility is maintained.
55 */
56 #define NPY_BINARY_VERSION 0x00010001
57 #define NPY_BINARY_COMPATIBLE 0xFFFF0000
58
59         /* Some platforms don't define bool, long long, or long double.
60           Handle that here.
61         */
```
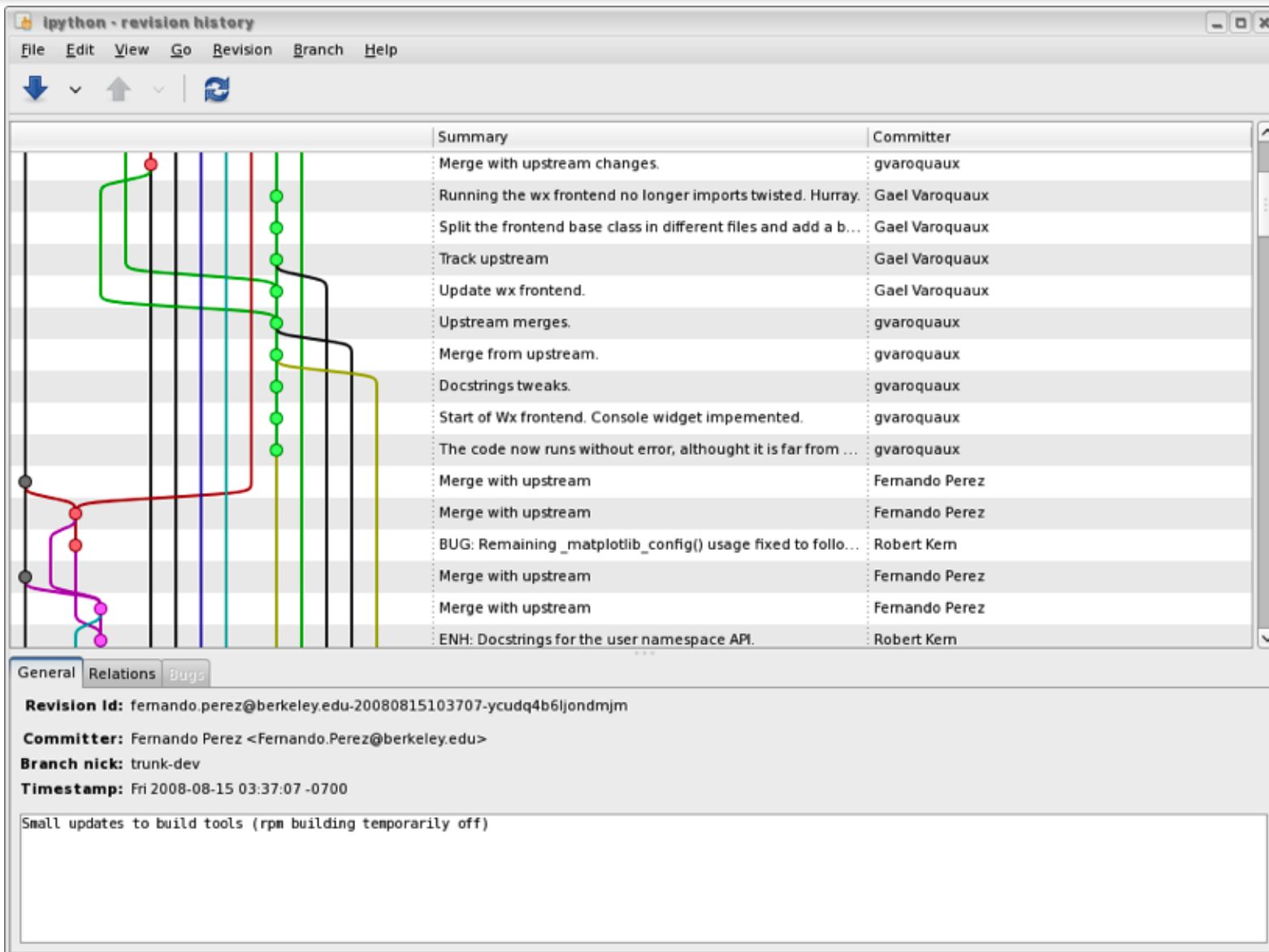
Enhancement Proposals

- A Simple File Format for NumPy Arrays
- Implementing date/time types in NumPy
- Matrix Indexing
- Runtime Optimization
- Solvers Proposal

Distributed Version Control

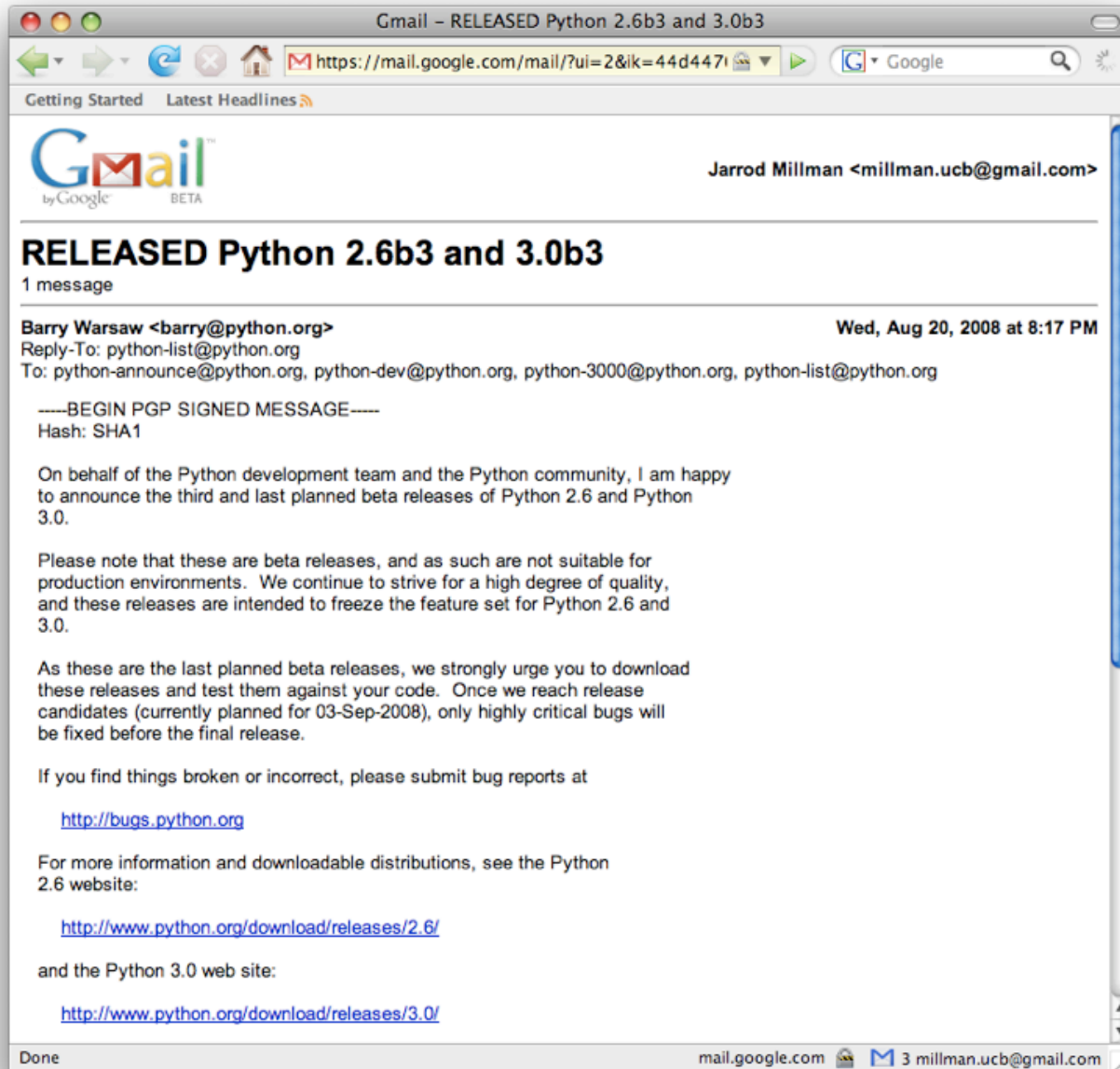# Launchpad

Launchpad Code Review

Launchpad

The Larger Ecosystem

- IPython
- Python

IPython 0.9

- IPython1 in trunk
- Graphical frontends
- More...

Python 2.6 and 3.0

NumPy/SciPy and Python 2.6/3.0

- Improve test coverage
- Port to Python 2.6
- Improve test coverage
- Run 2to3 tool
- Fix issues
- ISSUE: No 2to3 tool for the C-API!

Getting Involved

- Documentation
- Bug-fixes
- Testing
- Code (review, cleanup, contributions)
- Mailing list participation
- Start a local SciPy group
- Code sprints
- Documentation/Bug-fix Days
- Web design

Thank You