Pybots – testing Python projects in real time

Grig Gheorghiu
RIS Technology
grig@ristech.net

PyCon 2007, Addison, TX

Executive Summary

- the goal of the Pybots project (www.pybots.org) is to allow people to run automated tests for their Python projects "in real time"
- testing is done every time a commit is made to the Python's subversion repository (trunk and 2.5 branch currently), with Python binaries built from the latest source

Presentation outline

- setup of the Pybots buildbot farm
- issues that the Pybots farm has helped uncover
- lessons learned
- future work
- lessons learned in building, sustaining and growing an open-source community project.

Genesis of the Pybots project

- message from Glyph to python-dev
- help from PSF in getting buildmaster up
  - buildbot master running on python.org server
- projects are being tested every time a commit is made into the Python trunk (2.6) and the 2.5 branch
- idea can be applied to any project -- got on the radar screen of the Mozilla QA team (Dave Liebreich)

Pybots setup

- 11 buildbot slave machines contributed by volunteers; 20+ projects under test
- x86 (Debian, Ubuntu, Gentoo, FC6, RH9, OS X, Windows 2003), AMD64 (Ubuntu), SPARC (Solaris 10), Mac G5
- each buildslave can test one or more projects
  - examples of projects under test: Twisted, Trac, Genshi, Django, docutils, roundup, Kid, parsedatetime, vobject, zanshin, MySQLdb, pysqlite, feedvalidator, CherryPy, lxml, Bazaar, Cheesecake, twill, py library, nose

How it all works

- svn commit notifications on Python trunk and 2.5 branch trigger new builds on each Pybot buildslave
- each buildslave builds Python binaries and libraries, runs Python unit tests
- each buildslave runs project-specific automated tests using the freshly built Python binaries/libraries
- notifications sent via email and RSS feeds (for all buildslaves or per slave)

Issues uncovered (part 1)

- functional testing of Python installation via installation of pre-requisite packages
  - when version was changed to 2.6, nothing could be installed anymore
- issues with new reserved keywords: 'as' and 'with'
  - examples of projects affected: zope.interface, roundup, pydoctor
- test suite issues
  - CherryPy automated tests required input from user (hit enter -- for Windows-specific DOS prompt)
  - some CherryPy unit tests required user input

Issues uncovered (part 2)

- platform-specific issues
  - OS X test_itertools issue: test_count
    - self.assertEqual(repr(c), 'count(-9)') AssertionError: 'count(4294967287)' != 'count(-9)'
    - Fix: "Fix %zd string formatting on Mac OS X so it prints negative numbers"
  - Bazaar issues on Windows: path tests, TCP client tests
  - Twisted issues on RH9: multicast-related tests
- pre-requisite issues
  - neo_cgi.so backtrace in Trac tests
  - Pyrex refcount issue in lxml tests

Issues uncovered (part 3)

- Sam Ruby's feedvalidator unit test suite was composed of several test suites; Sam accepted patches from Manuzhai to make it easier to run the tests in one aggregated suite
- Python extensions modules sometimes didn't get rebuilt after changes to configure/configure.in
  - setup/py distutils not doing full rebuild when header files change

Lessons learned (part 1)

- VERY IMPORTANT
- unit testing is not enough!
  - Python unit tests all passing, while installation of any Python package was failing
- need functional testing which exercises applications from the outside in
- need installability tests
  - example: installing projects/utilities used by the main projects under test

Lessons learned (part 2)

- important to get tests to pass
  - if buildbot step is red all the time, people tend to ignore the alerts ('broken window' syndrome)
- somebody needs to stay on top of current status, alert people when their slaves are down
- buildbot scripts and configuration files checked in into Google Code project help other people get up to speed quicker
- os.system very unreliable for return codes; use subprocess

Future work

- more projects
- more buildslaves on more platforms
- automatic testing of PyPI packages on virtual machines
- Jython
- Py3K

How to build an open source community

- blog, blog, blog
- send messages to mailing lists related to the area of your project
- write extensive documentation, make it easy for people to join
- create a project repository (Google Code)
- get help from early adopters, involve them in the project
  - lieutenant role very important

How to sustain and grow an open source community

- blog, blog, blog
- send messages to individuals who might be interested in contributing
- acknowledge contributions
- respond quickly to issues on mailing list
- demonstrate usefulness of the project
  - to contributors, organizations involved in project (PSF)
- market/promote/evangelize the project tirelessly
  - "Fearless change: Patterns for introducing new ideas", by Mary Lynn Manns and Linda Rising
  - wikipatterns.com