GSoC 2012 Application: Vladimir Perić: Python 3 preparation

Abstract

Python 3 is the future of Python. If Twisted is to see continued usage in the future, it will have to be ported, and rather sooner than later. As Twisted is a large and complicated code-base, this process needs to be done with care, ensuring that any code written remains compatible with the currently supported versions. The test-driven development methodology Twisted uses will ensure no regressions happen and will ease the maintenance of the code-base.

Synopsis

Porting a large project must be done in parts; additionally, care needs to be taken to avoid incompatibilities with older supported versions (though 2.5 support will be dropped after the next release, which makes porting easier). I plan to run Python with the “-3” flag to root out obvious issues, and then run 2to3 in a private branch and choose the right methods to handle any problems that arise. Twisted has a robust testing suite, and the small, self-contained tests will be of immense help, as they will allow me to quickly localize issues.

My development process will follow the standard Twisted development process (http://twistedmatrix.com/trac/wiki/ReviewProcess), under the guidance of my mentor. Features will be matched to small, self-contained tickets. For example, the replacement of a particular Python 3-incompatible idiom might be done as a ticket. Each ticket will be developed in a branch. When the code and matching unit tests are finished, they will go through code review by another Twisted developer to ensure it follows the coding standard, is well designed and sufficiently tested and documented. Once the issues from the code review have been addressed the branch will be merged into trunk and the ticket closed. Small, self-contained tickets, frequent feedback via code review, and the requirement for doing testing and documentation will ensure I am learning and improving throughout the semester, and will enhance my ability to get code merged into Twisted from the very beginning.

After the low-hanging fruit is taken care of, the right approach to porting needs to be chosen. Due to the size of Twisted, a single code-base must be used - using a classic dual code-base, with separate versions for Python 2 and 3, would only make development slower and spread the developers even thinner. This approach was attempted without much success. However, there are still two options: writing the code in a subset of Python 2 and 3, which will be runnable on both, or writing Python 2 code and relying on 2to3 to handle the translation at install-time. I have used the 2to3 tool before with success, but I plan to investigate both options to see which one is more suitable for Twisted. I will target Python 3.2, which is easier to support than 3.1; if possible,

code will then be expanded to cover 3.1 as well.

Although the ideas page specifically mentions the project is not about porting completely to Python 3, I will try to do this, or at least get as close as possible and provide a working prototype. In my experience, it is very hard to estimate the amount of work required for porting, so it just might be possible to finish it in the summer timeframe. If not, I intend to help after GSoC with whatever issues remain. If I manage to complete the project before the deadline, I will devote myself to improving PyPy support, which will help make the code more robust (by reducing or removing reliance on implementation details of the official interpreter).

Deliverables

- Minimal warnings when running the test suite with the -3 flag
- A working port of at least a minimal subset of Twisted to Python 3

Schedule

Unfortunately, the beginning of my coding period coincides with the start of all of my exams, so during the first two weeks I will not be able work a full 40 hour week. To compensate, I plan on spending more time coding during the community bonding period.

April 24 – May 21 (Community Bonding Period):

- Familiarize myself with the code in different parts of Twisted. Ascertain if the current testing framework is robust enough to handle porting to Python 3; if required, write additional tests in problematic areas.
- Create tickets for all issues found; this will make it easier later to concentrate on just coding and allow me to better prioritize my work.
- Communicate with current developers and investigate if there is code which can be deprecated, as deprecated code needn't be ported. This will also allow me to get a more in-depth knowledge of parts of Twisted.
- Bond with the community: get to know the developers responsible for the major parts of Twisted, so that I know who to turn to if problems arise; educate current developers on the problems arising from supporting multiple Python versions in a single code base.

May 21 – July 7 (Official coding period starts):

- Work on fixing the warnings produced when running with the -3 flag. The tool reports many warnings, some of which are irrelevant (eg. callable() is back in Python 3.2) and some for which it is impossible to estimate the amount of work required.
- Remove the usage of core modules not supported in Python 3 (many of these are caught by the -3 flag).
- Investigate which of the two approaches (shared codebase or using 2to3) is superior. Whichever approach is chosen, my plan would be to aggressively push any changes I

make upstream, to start the appropriate review processes. This will make working closely to master easier.
- As soon as a minimal version of Twisted can be run with Python 3, work on distutils/distribute supporting Python 3 and ensure it doesn't break anything.

MID TERM EVALUATION
At this point, I expect to have only a few warnings when running the test suite with the -3 flag. Ideally, I would also have a minimal version of Twisted working under Python 3; if this is not possible, then I hope the approach would at least allow me to notice errors not caught by the -3 flag.

July 14 – August 20:
- Port as large a subset as possible of Twisted to Python 3. If it turns out to be possible to port a significant part, I would help with making an "alpha" release, to get testing by users (which ought to reveal further bugs).
- Continue work on porting the remaining modules to Python 3. If this is completed before the end of the coding period, continue working on PyPy support.

About Me

Name: Vladimir Perić, student of Open Informatics at the Faculty of Electrical Engineering, Czech Technical University in Prague.

Email: vlada.peric@gmail.com / pericvla@fel.cvut.cz
IRC: vperic
GSoC blog: http://vperic.blogspot.com/

Coding platform: 64bit Linux

I have experience with several programming languages (notably Java, C, Python, Prolog), but I prefer using Python in most projects. In my AI and Game Theory classes, we usually had a choice between Python and Java, which is what finally convinced me that Python is an excellent language: compared to Java, I was able to develop much faster and produce cleaner code. Any performance advantage offered by Java was nullified by me being able to implement better algorithms. I also began taking a greater interest in Python, particularly the problematic of porting to Python 3. I've also had a class about networks, though I've never used Twisted before. I do feel I will be able to understand it well enough to finish my project, though.

Last year I was accepted as a GSoC student for SymPy to port it to Python 3; though a compatible version hasn't been released yet, my report and my blog have plenty of information about my experiences during the project. I have continued contributing to SymPy and now have push access to the main repository; all the code I contributed can be seen through the Github web interface here. As such, I feel I have a good knowledge basis and plenty of practical

experience with Python 3 porting and will be able to accomplish a large amount of work during in the summer. I am also proficient with git, having used it last summer on SymPy, and plan on using it this year too (via git-svn). I've found that the "cheap" branches offered by distributed version control systems greatly increase my productivity.

I have submitted a simple patch to Twisted which eliminates the deprecated dict.has_key(k) construct, though it hasn't been reviewed yet. The ticket is #4053.

Further reading

zzzeek's guide to Python 3 porting: A guide with a detailed description of the porting process, including the more practical questions, as well as links to further resources.
Official Porting to Python 3 guide: Further information related to supporting specific versions of Python, and other information.