Managing 80 Open Source Packages and 5 Million Lines of Code in Sage

http://www.sagemath.org

Michael Abshoff[1]

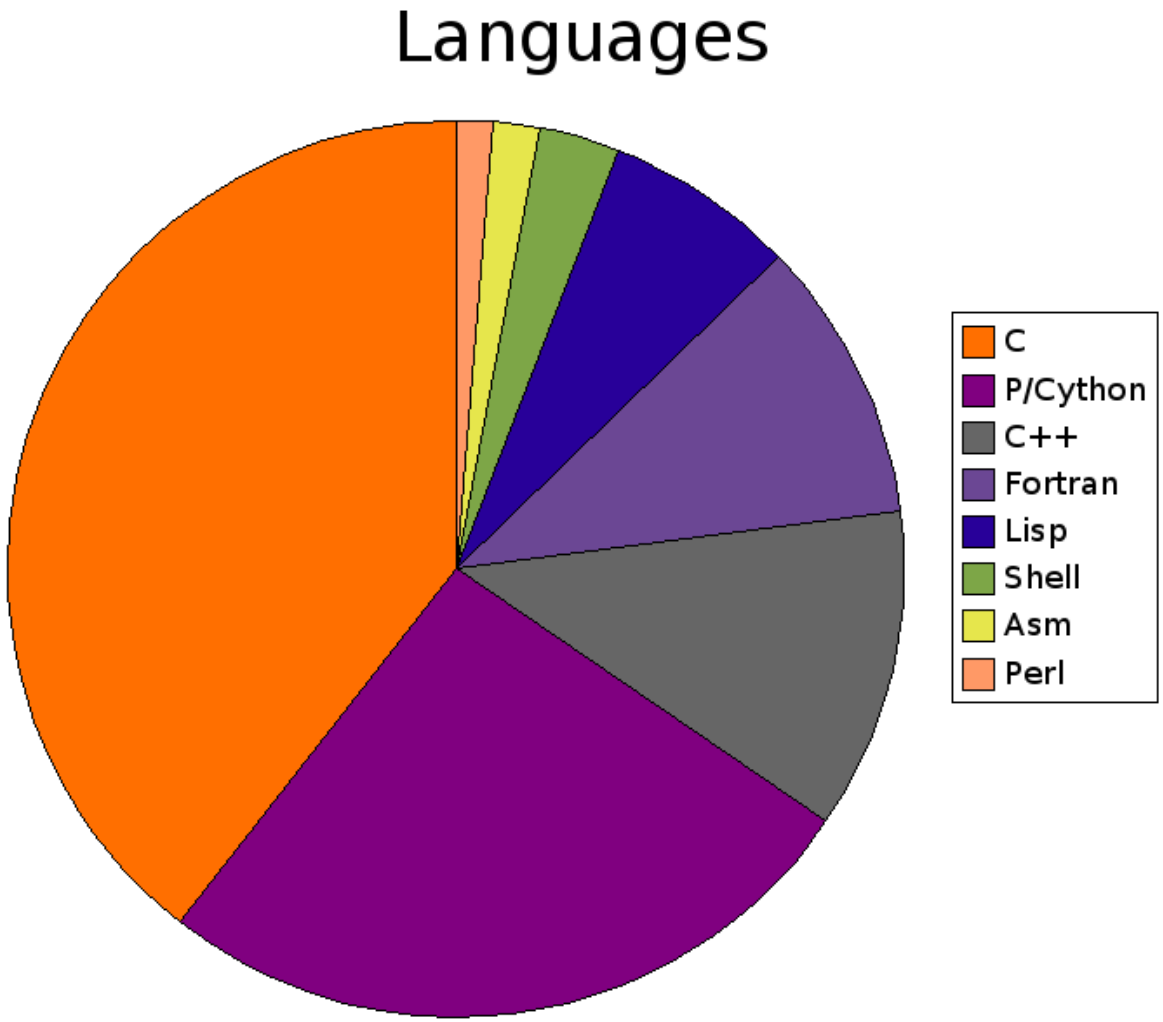
[1]Department of Mathematics
Technical University of Dortmund, Germany

Seattle, 2008-06-12

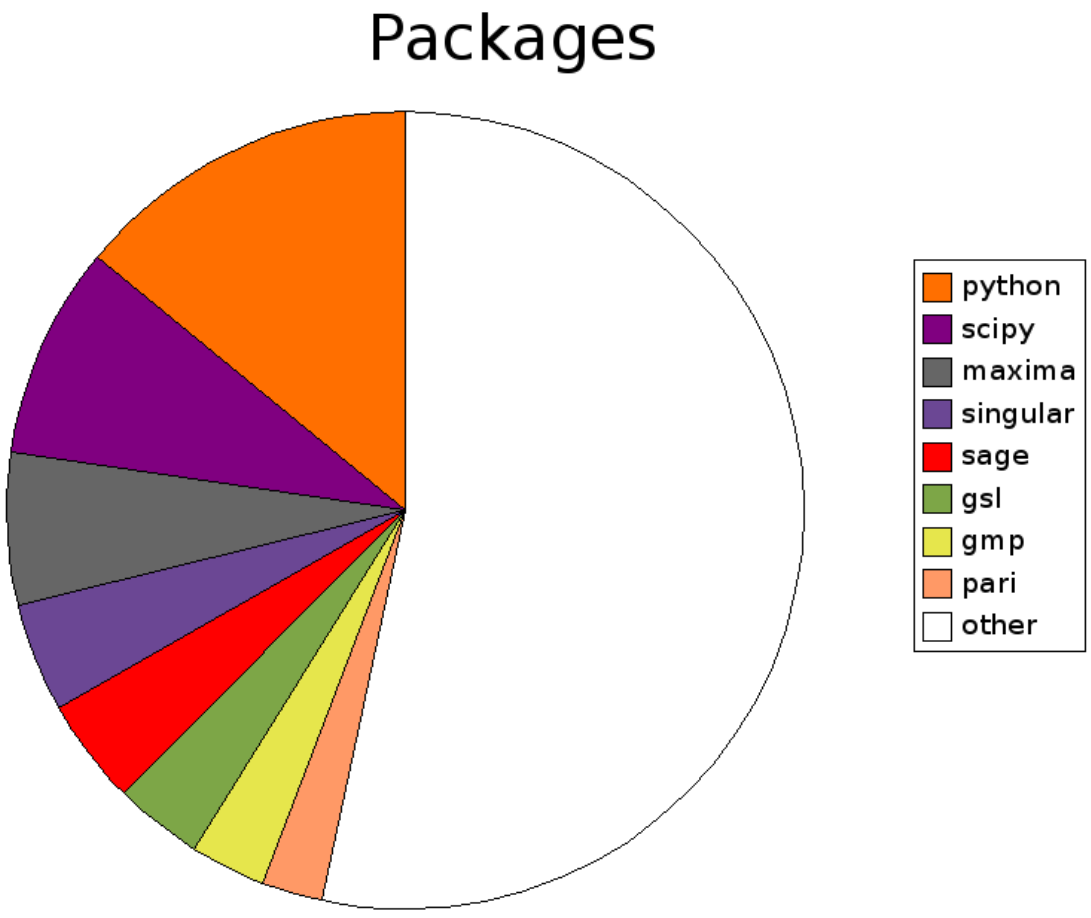
Outline

What Is Sage?

- 5,000,000+ lines of code
- 80 different units
- languages: mainly C, C++, Python, Cython, Fortran, Lisp



A total of nearly 5 million lines of source code (several hundred person-years).


Packages
python
scipy
maxima
singular
sage
gsl
gmp
pari
other

Outline

Sage: How and Why?

- KISS
- a release about every two weeks
- “Stone soup“ development model
- 45,000+ test cases run after each patch merged - we will hit 100,000+ tests hopefully by the end of the year
- no need for a separate development version since all sources are included
- The Sage library is under revision control and changes can be made and checked in without ever leaving Sage

Outline

Sage: Deployment and Integration

- batteries included, i.e. no need to get the sysadmin to install any packages
- requirements to build Sage on Debian, Ubuntu: “apt-get install build-essential“, on OSX: Install a current XCode release
- to build: execute “make“ and come back after a while
- easily extendable via optional spkgs installed from a central (in house) server
- optimize for your CPU locally or class of workstations/nodes in a cluster

Outline

Sage: Playing Well With Others

- no outside dependencies for binary besides the usual suspects, i.e. libc, libm, libstdc++ ...
- no file outside the build directory and $DOT_SAGE is written to or read from
- many Sage releases can be installed in parallel without affecting each other
- package up the exact build with your changes in a binary and deploy it to a bunch of machines or throughout the department

Outline

Sage: Contributing To Sage

- Twisted rule: “Don’t work on anything unless there is a trac ticket for it“
- mandatory patch review
- mandatory 100% test coverage
- must pass build testing on all supported platforms

Outline

Sage: Community Resources

Communicate with the people who wrote the code and/or know it really well via:

- Google Groups
- Email
- Trac
- IRC