Managing Python in HPC Environments

Daniel Gall, Engility

Frank Indiviglio, NOAA

Python in HPC 10 years ago

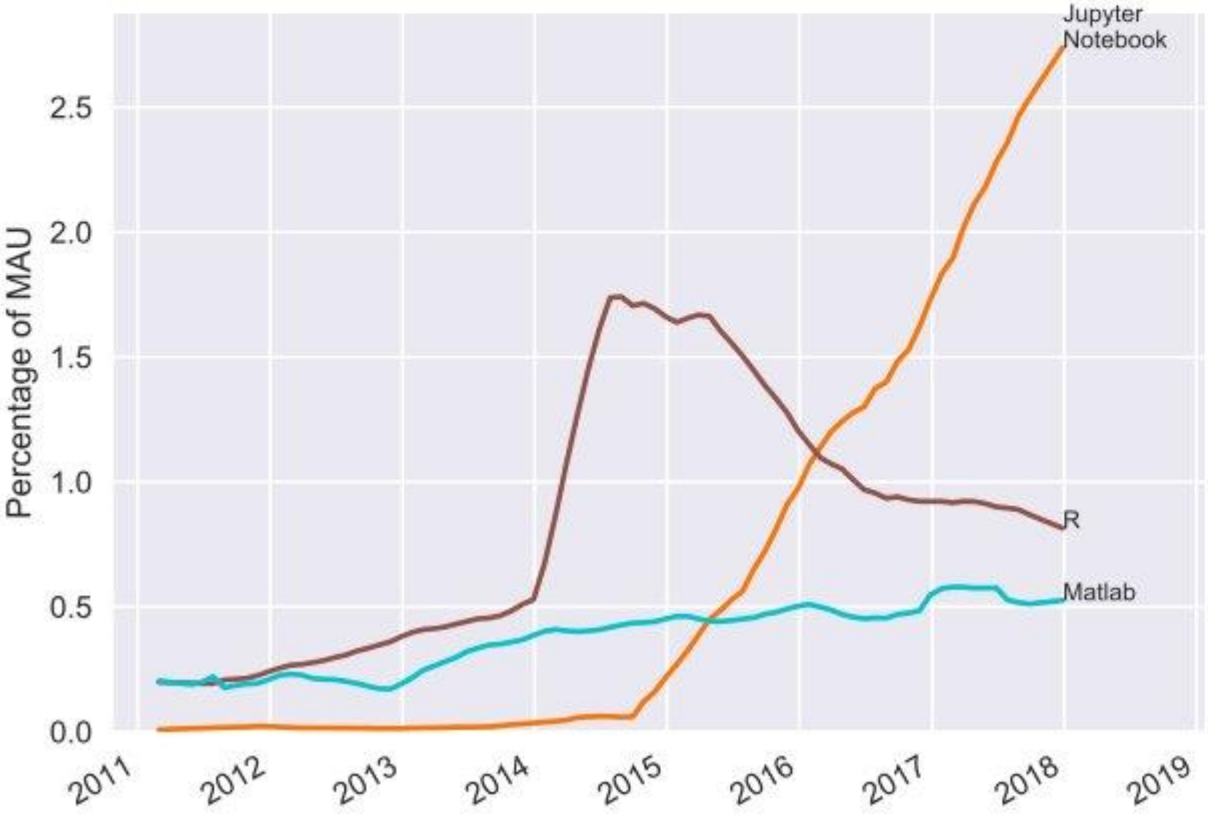
- Systems programming language
- Versions not as important
- 3 year old Python in enterprise Linux will be fine for the 5 year service life of an HPC asset
  - Corollary: Abrupt version transitions on HPC recapitalization drops are also fine
- Integrator reluctance to support different versions/packages

Disruption: scientists discover Python

- Versions become important
  - Python
  - Each package
  - Non-Python libraries depended on by Python packages
- Must keep old versions
- Environment modules not able to present the choices clearly or handle the dependency complexity

More disruption: scientists improve Python

- Jupyter notebooks
- Ease of installation/use on laptops
- Begin contributing to libraries
- Linux bugs crop up. What happened?
- Scientists develop mostly on Macs.
  - Linux testing suffers.



http://www.benfrederickson.com/ranking-programming-languages-by-github-users/

Yet more disruption: Anaconda

- Easier to install – just works out of the box
  - Carries libraries with it
- Easier to distribute your packages
- Easier to distribute libraries and applications
- Becoming sole binary distribution channel for hard-to-build scientific packages
  - e.g. CDAT
  - But I have to build this with ICC/PGI/FotM compiler!
    - Higher HPC admin costs

Expectations higher
Robustness lower

- “I can just install this on my Mac/Ubuntu laptop. It shouldn’t be hard to get on HPC.”
- It shouldn’t be, but there are other pressures that aren’t going away.
  - Reliability
  - Reproducibility → persistence of old versions
  - Continuity of operations
  - Cost efficiency
- Answer:
  - Leave system Python alone
  - Deploy a baseline Python with packages all in one environment module
  - Support the power users with Anaconda/PyPI mirror

Baseline solution: Spack

- ORNL made a Spack deployment of Python
  - One environment module per version
  - All packages and interfacing libraries Spacked together
    - Actually it’s two Spacks: one for Python2 and one for Python3
    - Pip and pip3 don’t play nicely with one another
  - Pip to install most Python packages
  - Requirements file available via ‘module help PythonEnv-noaa’
  - New versions made out of cycle with hardware drops
    - Cost efficient, reproducible, reliable, easy to use
- Most users use this.

Supporting power users

- For users who want to run their own show we provide an Anaconda / PyPI mirror.
- We have to change the installers to make them “just work” with the mirror and meet higher expectations
- Want insights on what packages may need to be in the baseline
  - Enterprise-application logging of Python dependencies
  - PyRats = Snakefood + logging + sitecustomize.py

PyRats

- Snakefood dependency analyzer http://furius.ca/snakefood/
- Fixed bugs, made Python2-3 compatible, condensed into a single file
- Adds logging of dependencies to central log server – via UDP to get out of the way fast
- Preserves pristine user environment
- Skips itself if Python is run interactively
- Placed in $ANACONDA/lib/pythonX.Y/site-packages/sitecustomize.py
- One step away from a Python trojan
- https://github.com/daniel-gall/pyrats

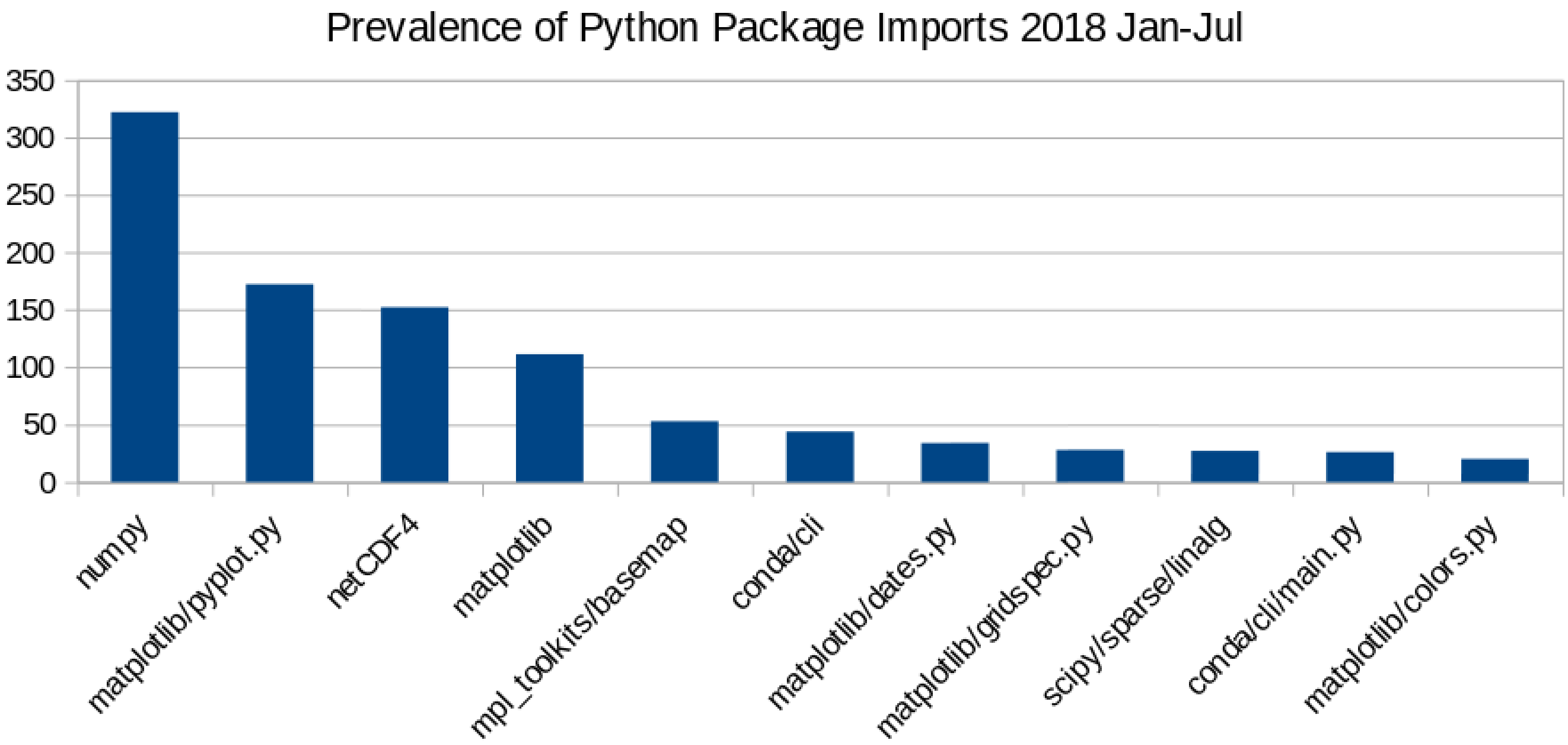
Conda create -n myenv

- Conda doesn’t propagate sitecustomize scripts to child environments
- We’re changing the Anaconda installers anyway...
- Change the conda it installs while we’re at it
  - Patch detects conda environment creation and copies any sitecustomize.py script to child environments
  - The patch needs to survive updating conda
    - We detect ‘conda update conda’ and perpetuate the patch by patching the new conda
    - One step away from a Python virus
- Made easier by Anaconda installers not checking individual package checksums, relies on whole installer MD5
  - Security risk for MitM chosen infix hash collision

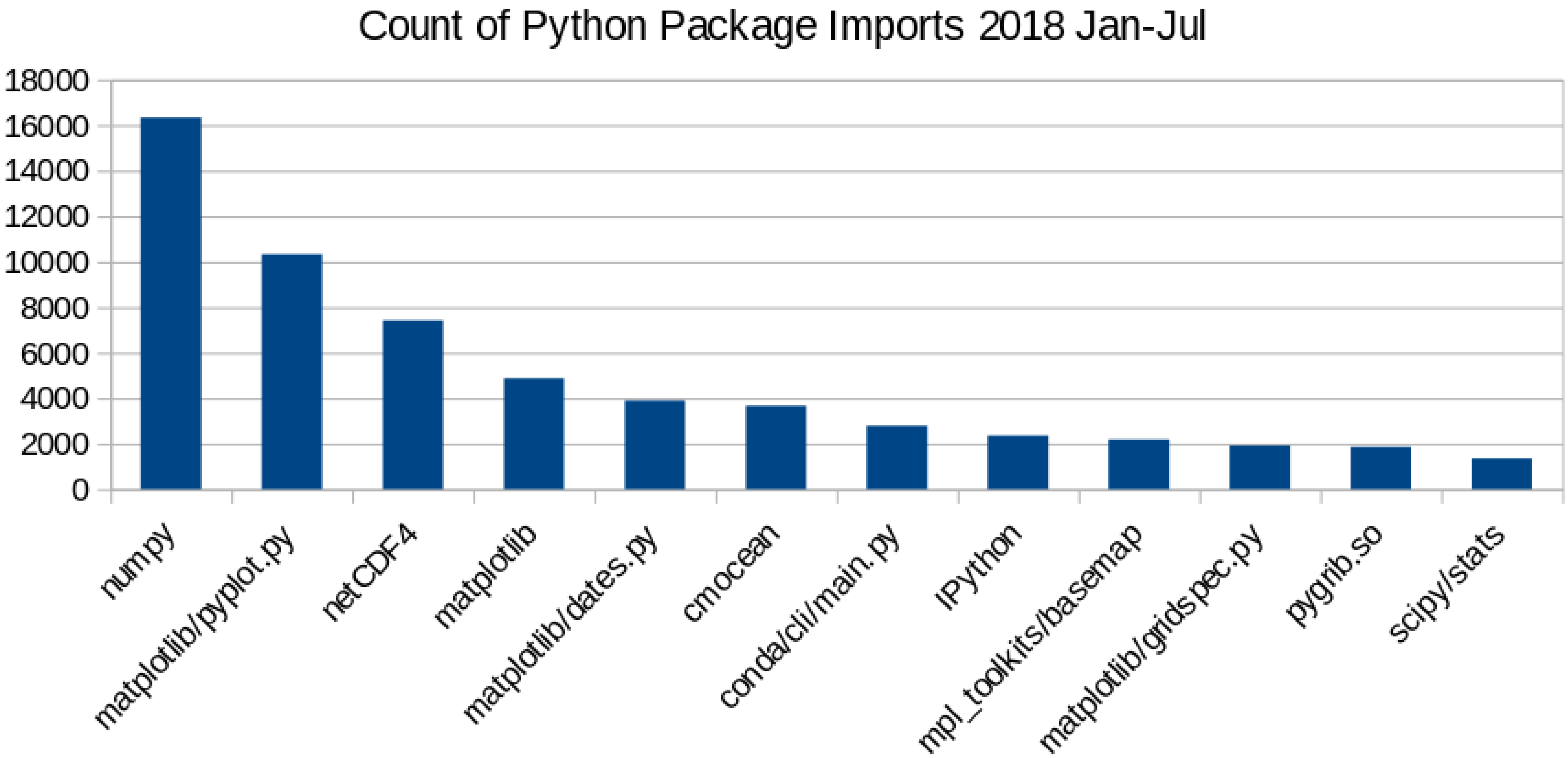
Log example

-
  ```
  Nov  2 16:06:46 node_name Python: ts=2018-11-02 20:06:46,642;loggerName=PyRats;pathName=/contrib/anaconda/anaconda2/4.4.0/lib/python2.7/site-packages/sitecustomize.py;python=/contrib/anaconda/anaconda2/4.4.0/bin/python;logRecordCreationTime=1541189206.642521;levelNo=20;levelName=INFO;message=((('/path/to/the/thing', 'plotting_scripts/plot_grid2obs_conus_sfc_tsmean.py'), ('/path/to/the/dependency', 'matplotlib'))),((('/path/to/the/thing', 'plotting_scripts/plot_grid2obs_conus_sfc_tsmean.py'), ('/path/to/the/dependency', 'matplotlib/pyplot.py'))),((('/path/to/the/thing', 'plotting_scripts/plot_grid2obs_conus_sfc_tsmean.py'), ('/path/to/the/dependency', 'numpy'))),((('/path/to/the/thing', 'plotting_scripts/plot_grid2obs_conus_sfc_tsmean.py'), ('/path/to/the/dependency', 'pandas'))),((('/path/to/the/thing', 'plotting_scripts/plot_grid2obs_conus_sfc_tsmean.py'), ('/path/to/the/dependency', 'plotting_scripts/plot_defs.py')))
  ```

What do we get? What packages get used by the most users?

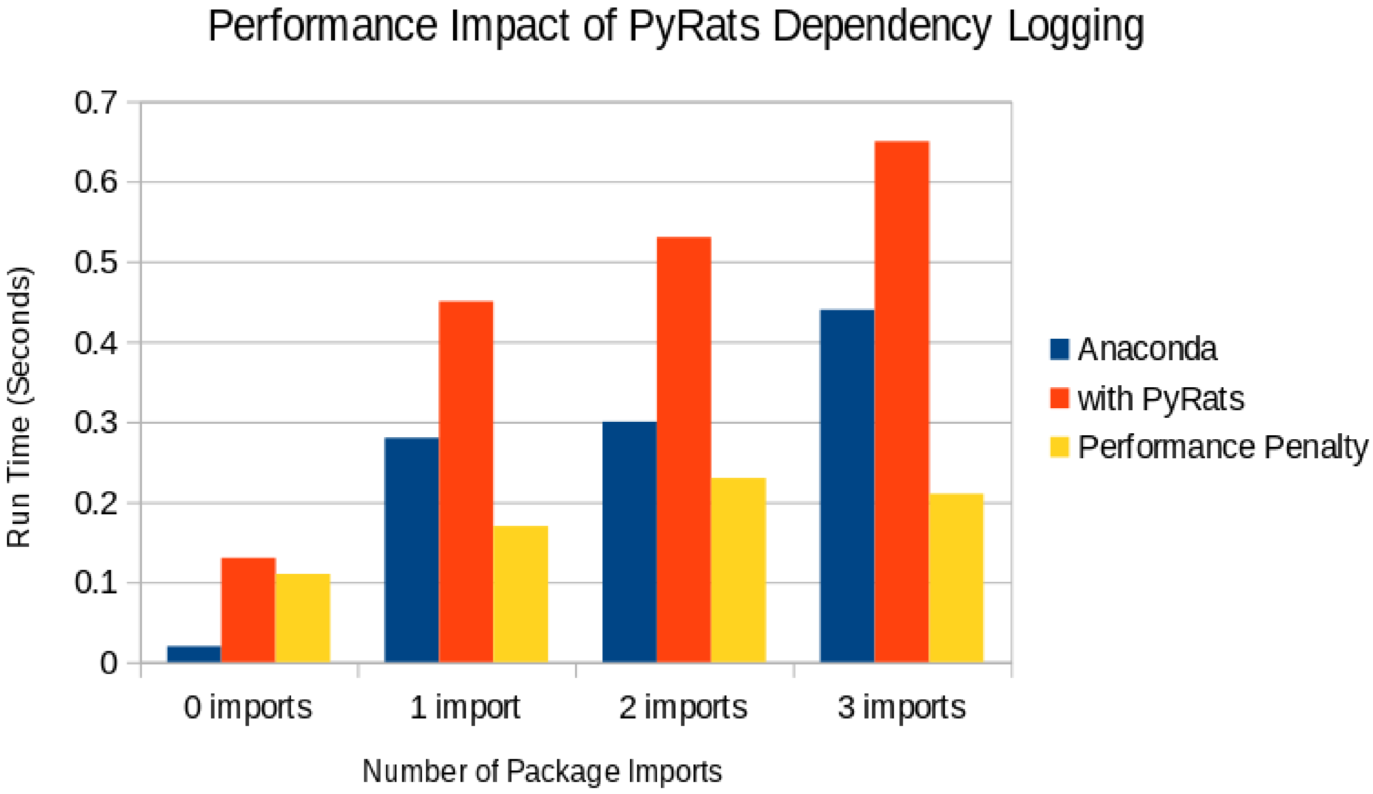


What is run the most?

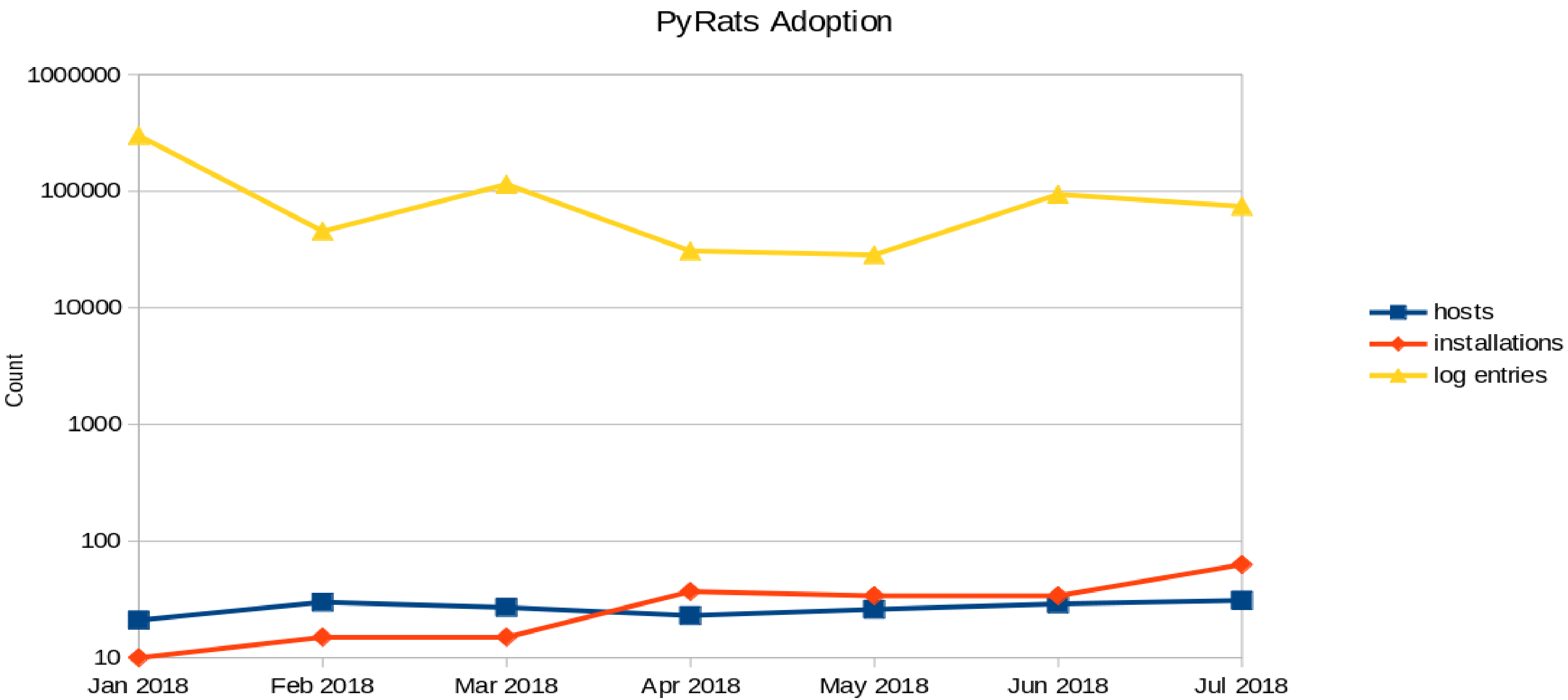


Addressing concerns

- Privacy – Federal system: no privacy
- Performance ~0.2s penalty
- For collegiality and in case of performance issues we inform users of what we're doing and how to remove it



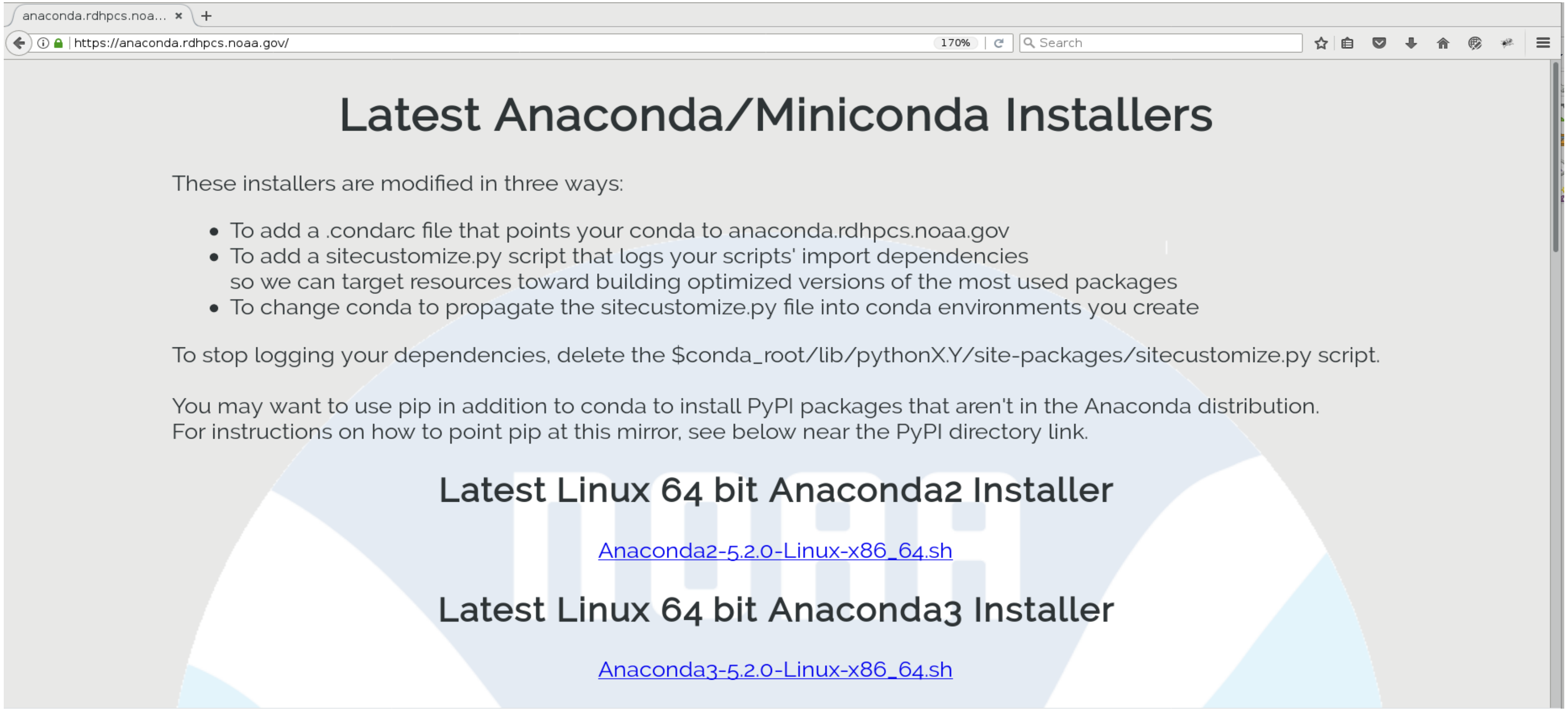
PyRats adoption rate



Mirror

- Full Anaconda main channel mirror
- Partial mirroring of other conda channels
- Partial PyPI mirror
- Package lists + wget with pattern matching
- One user request results in all past and future versions of a package being mirrored
- Namespaces don’t collide – can use one mirror VM
- About 220GiB for main channel + our current package lists
- Automatically generate an index page that
  - documents the changes made to the installers
  - provides links to the latest installers
  - and documents how to make a ~/.pip/pip.conf

Mirror



Latest Anaconda/Miniconda Installers

These installers are modified in three ways:

- To add a .condarc file that points your conda to anaconda.rdhpcs.noaa.gov
- To add a sitecustomize.py script that logs your scripts' import dependencies so we can target resources toward building optimized versions of the most used packages
- To change conda to propagate the sitecustomize.py file into conda environments you create

To stop logging your dependencies, delete the $conda_root/lib/pythonX.Y/site-packages/sitecustomize.py script.

You may want to use pip in addition to conda to install PyPI packages that aren't in the Anaconda distribution.
For instructions on how to point pip at this mirror, see below near the PyPI directory link.

Latest Linux 64 bit Anaconda2 Installer

Anaconda2-5.2.0-Linux-x86_64.sh

Latest Linux 64 bit Anaconda3 Installer

Anaconda3-5.2.0-Linux-x86_64.sh

Questions?

- Questions?
- Thank you for coming to our talk.