InterPlanetary Wheels

A resilient approach to distributing software

Ngô Ngọc Đức Huy

2022-02-05

Problems

- Problem domain: Python packages
- Current main method: Installing via PyPI
- Others: Distros packages, Anaconda (non-free), ...

PyPI issues

Warehouse in nature:

- Single point of failure: CDN outage?
- Checksum and URL from same origin?*
- Expensive yet not future-proof dependency resolution
- Uncurated: Typosquatting? Malware?†

*See Linux Mint 17.3 incident

†As previously seen on npm

Outages

Major Outage across all Infrastructure

Resolved This incident has been resolved.
Posted 5 months ago. Jun 08, 2021 - 16:04 UTC

Monitoring Our provider reports a fix is applied. We are monitoring.
Posted 5 months ago. Jun 08, 2021 - 11:33 UTC

Update This outage is now impacting many of our CDN fronted services. We will continue to follow along with our providers updates.
Posted 5 months ago. Jun 08, 2021 - 10:09 UTC

Identified Our CDN provider has opened an incident and we are following along with their updates.
Posted 5 months ago. Jun 08, 2021 - 10:01 UTC

Investigating We are currently investigating an issue with our CDN causing PyPI to be unavailable. We've established that our backends are online and functioning.
Posted 5 months ago. Jun 08, 2021 - 09:58 UTC

This incident affected: PyPI (pypi.org - CDN), python.org (python.org - CDN), docs.python.org (docs.python.org - CDN), and us.pycon.org.

2021-06-08T09:58Z/16:04Z

Resolved This incident has been resolved.
Posted 7 months ago. Apr 05, 2021 - 17:08 UTC

Monitoring Backend services have been restored. We are monitoring for stability and preparing notes for an incident report.
Posted 7 months ago. Apr 05, 2021 - 16:06 UTC

Update Our core backend service is back online and stable. We are slowly bringing the backing services back online followed by the PyPI applications.
Posted 7 months ago. Apr 05, 2021 - 16:01 UTC

Update We are continuing to try to bring our backend systems back online.
Posted 7 months ago. Apr 05, 2021 - 15:39 UTC

Identified An internal certificate in our deployment infrastructure has expired and we are working to roll out a new certificate and restart services.
Posted 7 months ago. Apr 05, 2021 - 14:54 UTC

Investigating All Web UI and Uploads impacted.
Posted 7 months ago. Apr 05, 2021 - 14:30 UTC

This incident affected: PyPI (pypi.org - CDN, pypi.org - Backends, files.pythonhosted.org - Redirects, files.pythonhosted.org - Redirects Backends).

2021-04-05T14:30Z/17:08Z

Workarounds

- Pin dependencies
- Replicate checksum hashes
- Run own mirror, e.g. devpi

Approach

- Cherry-pick packages
- Decentralize

Python simple repository API

- A.k.a. PEP 503
- E.g. https://pypi.org/simple
- HTML view of directory-like tree

```
/
|
+- pip/
|   |
|   +- pip-21.3.1-py3-none-any.whl
|
+- setuptools/
    |
    +- setuptools-58.5.3-py3-none-any.whl
```

InterPlanetary File System

- Peer-to-peer network
  $\longrightarrow$ no single point of failure
- Content-addressing in global namespace
  $\longrightarrow$ clients are organic mirrors
- Merkle DAG (like Git but for directories)
  $\longrightarrow$ parent ID ensures child contents
- HTTP gateway is directory tree in HTML

Relevant works

- OpenBSD mirror via IPFS
- Pacman mirror via IPFS

Floating cheeses

- Often found in *the* flying circus
- A.k.a. InterPlanetary Wheels (IPWHL)
- Simple API (PEP 503) on IPFS
- Platform-unique & singly-versioned wheels
- Downstream repository
- Declarative index, i.e. dumb down nixpkgs

Package declaration format

```
source = 'https://[...]/mypy-0.910-py3-none-any.whl'
content-id = 'Qm[...]'
requires-python = '>=3.5'
extras = [ 'dmypy', 'python2' ]
dependencies = [
  'typing-extensions>=3.7.4',
  'mypy-extensions<0.5.0,>=0.4.3',
  'toml',
  'typed-ast<1.5.0,>=1.4.0; python_version < "3.8"',
  'psutil>=4.0; extra == "dmypy"',
  'typed-ast<1.5.0,>=1.4.0; extra == "python2"',
]
```

Checks

- No wheel pair in same project installable on same platform
- Verifiable wheel source
- Matching metadata (CID & requirements)
- All dependencies satisfied

Publishing releases

- From declarations
- (Optional) on top of previous version
- To one single CID, cryptographically signed
- Same wheels shared between releases

Using IPWHL

1. Get signed repo `$cid` from git.sr.ht/~cnx/ipwhl-data/refs
2. Set up IPFS gateway at `$url`
3. `pip config set site.index-url $url/ipfs/$cid`
4. ...
5. Profit!!!

The good

Efficiency:

- No client-side dependency resolution
- Local and regional network caching

Security & reproducibility:

- No untrusted code execution (sdists)
- Human verified packages
- One CID vs many versions & hashes

Reuse & collaboration:

- Promise packages integration
- Encourage upstream contribution
- Promote collaborative self-hosting

The bad and the ugly

- Lack of downstream maintainers
- Lack of reviews and testings
- Small number of packages (about 600)
- No new and shiny packages (about 100 outdated)
- Limited support matrix (so far, only x86-64)
- Lack of mirrors (chicken & egg)

Future plans

- Running packages' test suite
- Reproducible wheels
- Environment self-markers[‡]
- Horizontal scaling of human effort

[‡]Wheel tags on steroids

Getting involved

- Run a mirror: `ipfs pin add $cid`
- Contribute: https://sr.ht/~cnx/ipwhl

Questions and feedbacks?

- Mailing list: ~cnx/ipwhl-discuss@lists.sr.ht
- Matrix: #ipwhl:halogen.city
- IRC: #ipwhl at hackint.org

Copying