Pip download prefers newer package version even when local package exists #5500

New issue

Open **bendikro** opened this issue Jun 13, 2018 · 8 comments

bendikro commented Jun 13, 2018

Environment

- pip version: 10.0.1/master
- Python version: Tested on python 3.6
- OS: Linux

Description

`pip download` does not prefer package found locally even if it satisfies the requirements when there is a newer available at the remote package index

Expected behavior

Prefer the already existing package as long as long as it satisfies the dependency requirements

How to Reproduce

1. Create directory `pkg_cache`
2. Run `pip3 download --dest pkg_cache/ --find-links pkg_cache/ setuptools==39.0.1 && pip3 download --dest pkg_cache/ --find-links pkg_cache/ setuptools`

Output

```
pip3 download --dest pkg_cache/  --find-links pkg_cache/ setuptools==39.0.1 && pip3
download --dest pkg_cache/  --find-links pkg_cache/ setuptools
Looking in links: pkg_cache/
Collecting setuptools==39.0.1
  Using cached
https://files.pythonhosted.org/packages/20/d7/04a0b689d3035143e2ff288f4b9ee4bf6ed80585cc121
39.0.1-py2.py3-none-any.whl
  Saved ./pkg_cache/setuptools-39.0.1-py2.py3-none-any.whl
Successfully downloaded setuptools
Looking in links: pkg_cache/
Collecting setuptools
  Using cached
https://files.pythonhosted.org/packages/7f/e1/820d941153923aac1d49d7fc37e17b6e73bfbd2904959
39.2.0-py2.py3-none-any.whl
  Saved ./pkg_cache/setuptools-39.2.0-py2.py3-none-any.whl
Successfully downloaded setuptools
```

pfmoore commented Jun 13, 2018 Member

That behaviour is by design. Pip will always prefer the latest available version, it takes no account of where a package comes from.

bendikro added a commit to bendikro/pip that referenced this issue Jun 13, 2018

```
Fix pypa#5500: Make pip download prefer local package over remote ···
1a297cd
```

bendikro added a commit to bendikro/pip that referenced this issue Jun 13, 2018

```
Fix #5500: Make pip download prefer local package over remote ···
3b96381
```

pfmoore referenced this issue Jun 13, 2018

Fix #5500: Make pip download prefer local package over remote #5501 Open

bendikro commented Jun 13, 2018

@pfmoore
I see. We have multiple requirement files, and since pip does not handle double requirements it is necessary to do multiple calls to `pip download`, one for each requirements file. With the current behavior of pip, where one file has `setuptools` and another has `setuptools==39.0.1`, both `39.0.1` and `39.2.0` will be downloaded.

pfmoore commented Jun 13, 2018 Member

So? That's the point of pip download. I don't know if I'm missing something here but I can't see what the problem is. What exactly do you use the files downloaded via `pip download` for? As per the docs the intention is that you use `pip download` to populate a directory from which you can later use `pip install --find-links` to do an install while offline. The `pip install` command is perfectly capable of handling a `--find-links` directory with multiple versions of the same package in it, so why are you bothered that this is happening?

1

pradyunsg added C: download S: awaiting response labels Jun 14, 2018

bendikro commented Aug 21, 2018

The point is that consistency is useful. Things that behave differently all the time is less useful than things that do the same thing every time. Had pip supported handling multiple requirement files and dealt properly with the dependencies, this wouldn't be a problem though.

With two requirement files, as explained earlier, you never actually know exactly what package versions will be downloaded.

`The pip install command is perfectly capable of handling a --find-links directory with multiple versions of the same package in it, so why are you bothered that this is happening?`
Depending on the order of the requirement files you provide, different package versions are installed. Consistency is key.

Second reason is speed. By looking locally and finding a package that satisfies the dependencies, there is no need to check remotely. Therefore, a call to `pip download` would be blazing fast if the packages are already downloaded. Currently it's very slow.

pfmoore commented Aug 21, 2018 Member

I'm not sure I follow. Pip's current behaviour is perfectly consistent - I described it above:

> Pip will always prefer the latest available version, it takes no account of where a package comes from.

In fact, if we preferred local files, we'd be *harming* consistency, because you'd get something different installed depending on what was present locally.

I don't see anything actionable here. Pip's current behaviour is by design, if you want to propose a change, you'll need to provide details of what you propose, and you'll probably need more persuasive arguments than you've currently offered.

bendikro commented Aug 21, 2018

`I'm not sure I follow. Pip's current behaviour is perfectly consistent - I described it above:`
True. It's consistent in that you never know which version it will download in the scenario I describe.

`In fact, if we preferred local files, we'd be harming consistency, because you'd get something different installed depending on what was present locally.`
The whole point is to know exactly what will be installed based on the local files. But having pip download the same package versions each time is not possible with multiple requirement files as I describe.

I agree that the current default behaviour shouldn't be changed, but an option to be able to prefer local packages over checking remotely would still be useful.

What I propose is to have an option that makes pip check locally if a package that satisfies the given dependency already exists locally, and if so, do not check remotely.

pfmoore commented Aug 21, 2018 Member

OK, so what you're suggesting is an option to `pip download` that says "for each requirement, if it can already be satisfied from the destination directory, skip it, otherwise download the requirement as normal and store the downloaded file in the destination directory.

I can see the logic in that. If you wanted to create a PR implementing it, I'm not going to object. I can't say that I find your justification for the behaviour compelling, but that's something that can be debated later, when there's a PR to review.

3

pfmoore added the R: deferred till PR label Aug 21, 2018

mboisson commented Aug 30, 2018

This would also be very useful for HPC clusters on which the staff may build python wheels that are optimized for their CPU architecture. The current behavior requires HPC staff to always be recompiling new versions as soon as they are out, or risk users using dramatically slower python packages in some situations. Being able to tell pip to favor a local wheelhouse over some minor version increase found online would be very useful to us.

Assignees
No one assigned

Labels
C: download
R: deferred till PR
S: awaiting response

Projects
None yet

Milestone
No milestone

4 participants