Pulp - Issue #3149

pulpcore.plugin.download.asyncio has surprising timeout handling

11/27/2017 01:00 PM - gmbnomis

Status:	CLOSED - WONTFIX	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**Estimated time:**	0:00 hour
Category:			
Sprint/Milestone:	3.0		
Severity:	2. Medium	**Sprint Candidate:**	Yes
Version:	Master	**Tags:**	
Platform Release:		**QA Contact:**	
Blocks Release:		**Complexity:**	
OS:		**Smash Test:**	
Backwards Incompatible:	No	**Verified:**	No
Triaged:	Yes	**Verification Required:**	No
Groomed:	No	**Sprint:**	

Description

aiohttp defines two timeouts in the ClientSession read_timeout and conn_timeout. The read_timeout is cumulative for the entire connection, i.e. all data must be read within this timeout.

In case of longer connections used for downloads (i.e. connections using the streaming API to get the data in smaller chunks), this may lead to timeouts on a connection that is perfectly fine, as the timeout is global and not per read operation.

For example:

```
async def single_download():
    async with aiohttp.ClientSession(read_timeout=3, conn_timeout=3) as session:
        downloader = HttpDownloader('http://127.0.0.1:8000/large_file.txt', session=session)
        return await downloader.run()

loop = asyncio.get_event_loop()
result = loop.run_until_complete(single_download())
```

Results in:

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/lib64/python3.6/asyncio/base_events.py", line 467, in run_until_complete
    return future.result()
  File "<stdin>", line 4, in single_download
  File "/home/user/github/pulp-3-from-source/venv/src/pulpcore/plugin/pulpcore/plugin/download/asy
ncio/base.py", line 50, in wrapper
    raise error
  File "/home/user/github/pulp-3-from-source/venv/src/pulpcore/plugin/pulpcore/plugin/download/asy
ncio/base.py", line 47, in wrapper
    return await func(downloader)
  File "/home/user/github/pulp-3-from-source/venv/src/pulpcore/plugin/pulpcore/plugin/download/asy
ncio/http.py", line 130, in run
    to_return = await self._handle_response(response)
  File "/home/user/github/pulp-3-from-source/venv/src/pulpcore/plugin/pulpcore/plugin/download/asy
ncio/http.py", line 112, in _handle_response
    chunk = await response.content.read(1024 * 1024)
  File "/home/user/github/pulp-3-from-source/venv/lib64/python3.6/site-packages/aiohttp/streams.py
", line 607, in read
    return (yield from super().read(n))
  File "/home/user/github/pulp-3-from-source/venv/lib64/python3.6/site-packages/aiohttp/streams.py
", line 330, in read
    yield from self._wait('read')
  File "/home/user/github/pulp-3-from-source/venv/lib64/python3.6/site-packages/aiohttp/streams.py
", line 259, in _wait
    yield from waiter
  File "/home/user/github/pulp-3-from-source/venv/lib64/python3.6/site-packages/aiohttp/helpers.py
", line 727, in __exit__
    raise asyncio.TimeoutError from None
concurrent.futures._base.TimeoutError
```

after approx. 3 seconds.

One could use a high per connection timeout and an additional timeout for read() for long running connections. Something along these lines:

```
        async with client.get(download_url, timeout=long_timout) as resp:
            try:
...
                        new_data = await asyncio.wait_for(resp.content.read(self.CHUNK_SIZE),
                                                          per_read_timeout)
...
            except asyncio.TimeoutError:
...
```

Related issues:

Related to Pulp - Issue #3918: DeclarativeVersion cannot sync longer than 5 m...	**MODIFIED**

History

#1 - 11/28/2017 04:53 PM - dalley

- Sprint/Milestone set to 47

- Triaged changed from No to Yes

#2 - 11/30/2017 09:59 PM - rchan

- Sprint/Milestone changed from 47 to 48

#3 - 12/01/2017 09:41 PM - daviddavis

- Tags Pulp 3 MVP added

#4 - 12/19/2017 05:17 PM - rchan

- Sprint/Milestone changed from 48 to 52

#5 - 01/08/2018 10:26 PM - rchan

- Sprint/Milestone changed from 52 to 53

#6 - 01/30/2018 03:16 PM - jortel@redhat.com

- Sprint/Milestone changed from 53 to 54

#7 - 02/22/2018 05:22 PM - rchan

- Sprint/Milestone changed from 54 to 56

#8 - 03/09/2018 12:35 AM - bmbouter

- Sprint set to Sprint 33

#9 - 03/09/2018 12:36 AM - bmbouter

- Sprint/Milestone deleted (56)

#10 - 03/12/2018 01:56 PM - milan

- Assignee set to milan

#11 - 03/12/2018 02:57 PM - milan

- Assignee deleted (milan)

#12 - 03/14/2018 05:37 PM - jortel@redhat.com

- Sprint Candidate changed from No to Yes

#13 - 03/14/2018 05:38 PM - jortel@redhat.com

- Sprint deleted (Sprint 33)

#14 - 04/11/2018 05:45 PM - amacdona@redhat.com

- Tags deleted (Pulp 3 MVP)

#15 - 08/14/2018 07:23 PM - bmbouter

- Related to Issue #3918: DeclarativeVersion cannot sync longer than 5 minutes or a timeout error is raised added

#16 - 08/14/2018 07:36 PM - bmbouter

- Status changed from NEW to CLOSED - WONTFIX

FYI, In the aiohttp docs now (3.3.2) I see additional options named ['total', 'connect', 'sock_connect', 'sock_read']: https://aiohttp.readthedocs.io/en/stable/client_quickstart.html#timeouts

@gmbnomis I can see how the plugin writer can do things that make no sense with those types of options. I think they are aiohttp options, so there isn't much Pulp can do to resolve them. aiohttp could have some sanity validation added though, if it doesn't already do that. I recently filed an issue in upstream aiohttp and it was a good experience. Their tracker is here: https://github.com/aio-libs/aiohttp/issues/

In terms of what Pulp can do for this, I can only see closing this as WONTFIX since we aren't in control of those options. It's good to leave here as a reference though and WONTFIX will do that. @gmbnomis, if you have other ideas of what should happen please post. I just don't see another option.

#17 - 08/14/2018 10:47 PM - gmbnomis

@bmbouter This issue is not about preventing plugin writers from doing stupid things (life experience tells me that this would be futile...). It was intended as a kind of "heads up" that aiohttp timeouts seem not to be made for long downloads (at that time at least). And apparently that is what happened in #3918.

According to the documentation you linked, the sock_read timeout seems to implement the timeout implemented by the wait_for in the code I posted above. Thus, there should be no need for this anymore.

Nevertheless, I think Pulp should have sane default timeout definitions in the downloader for the typical use case (in order to ease the life of plugin writers). Probably, a very long or no total timeout. Connection and read timeouts should probably be in the range of seconds (or minutes?). Are there proven nectar default timeout settings you could reuse?

#18 - 04/25/2019 06:46 PM - daviddavis

- Sprint/Milestone set to 3.0

#19 - 04/26/2019 10:37 PM - bmbouter

- Tags deleted (Pulp 3)