GROMACS - Feature #2615

Switch to Python3

08/18/2018 03:34 AM - Pascal Merz

Status:	Closed
Priority:	Normal
Assignee:	
Category:	core library
Target version:	2020
Difficulty:	uncategorized

Description

Currently, all python scripts in GROMACS are written in Python2. Python2 support ends 01/01/2020 (https://www.python.org/dev/peps/pep-0373/#id4, https://pythonclock.org/), so a move to Python3 becomes necessary.

Python3 versions

https://devguide.python.org/#status-of-python-branches

Python versions generally have a 5 year lifetime after release. The oldest currently supported Python3 version is 3.4.

This probably excludes

- **Python <= 3.2** Compatibility with Python2.7 is difficult, and has been out-of-life since March 2016.
- **Python3.3** Compatibility with Python2.7 is easy, but has been out-of-life since September 2017. Also, recent Sphinx releases (>= 1.7.0) don't support Python3.3.

We could therefore decide to support 3.4+, or 3.3+ and move to 3.4+ once we require a newer Sphinx version (currently, we require 1.6.1).

Possible strategies

Python2/3 compatibility

Translate code to valid Python3, use `__future__` imports to guarantee compatibility to Python2.7. Accept possibly slightly lower performance in py2 when moving from `xrange` to `range`, `iterkeys()` to `keys()`, etc. Keep Python2.7 as required version in CMake (possibly first trying to find Python3.3/3.4+, and falling back to 2.7 if necessary - would need to check if that works reliably).

Move to Python3

Translate code to valid Python3 without keeping Python2.7 compatibility. Move required version to 3.3/3.4+. (Obviously, dropping Python2 could be done at a later stage.)

Required changes

The python scripts can be divided in different categories having different scope and different amount of required changes to be python3-compatible: (The assessment of what needs to be changed largely relies on help of tools like `futurize`, but with manual changes allowing for compatibility without the introduction of additional dependencies. Obviously, the scripts with "No changes needed" need to be thoroughly checked with Python3.)

- **Documentation scripts**
  - docs/*.py
  - docs/doxygen/*.py
  These scripts are needed to generate the documentation and have a fair amount of incompatibilities with Python 3. The main changes needed involve print statements, divisions, and some changes related to the move from lists (in py2) to iterators (in py3). Additionally, these scripts rely on the `cmp` function not present in python3. This needs to be replaced by rich comparisons.

- **Admin scripts**
  - admin/copyright.py
  - admin/builds/*.py

Probably no changes needed.

- **The nonbonded kernel scripts**
  - src/gromacs/mdlib/nbnxn_kernels/nbnxn_kernel_file_generator/make_verlet_simd_kernel_files.py
  Probably no changes needed.
  - src/gromacs/gmxlib/nonbonded/nb_kernel_*/*.py
  - src/gromacs/gmxlib/nonbonded/preprocessor/gmxpreprocess.py
  These scripts can probably be left untouched, as per Erik Lindahl's comment in https://gerrit.gromacs.org/#/c/6621/:

  > You can leave out the scripts in the nonbonded subdirectory. They are
  > only used to regenerate new group-style kernels, and I don't expect
  > we'll ever do that again since the present ones are scheduled to be
  > removed and fully replaced by the verlet-style kernels.

- **Physical validation scripts**
  - tests/physicalvalidation/ and subfolders
  The physical validation tests are already python2/3 compatible.

- **Misc**
  - src/gromacs/selection/tests/gensphere.py
  Has divisions, but none of them behaves differently under python3 -> no changes needed.
  - scripts/make_gromos_rtp.py
  This script seems very old, it is not even valid python 2.7. Probably best left unchanged or deleted.

- **CMake files**
  - cmake/FindPythonModule.cmake
  Small change is needed to ensure compatibility of the `find_python_module` function with python3.
  - cmake/FindPythonModule.cmake
  - docs/CMakeLists.txt
  These files explicitly require Python 2.7 via `find_package()` - needs to be changes once Python 3 is *required*.

Testing

How can we test that no errors are introduced? For the documentation scripts updated in https://gerrit.gromacs.org/#/c/6621/, the output of `make webpage` was compared. Is that sufficient? What about the other scripts?

Default *future* imports

As proposed by Eric Irrgang in https://gerrit.gromacs.org/#/c/6621/, if Python2 compatibility should be retained for a longer time, it might be worth to include a set of imports to all Python files to avoid future incompatibilities, e.g.

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
from __future__ import unicode_literals
```

Related issues:

Related to GROMACS - Task #2831: Bump required version numbers of infrastruct...	**Closed**

Associated revisions

Revision ad2ac649 - 04/24/2019 03:02 PM - Eric Irrgang

Add Flake8 linting to gmxapi tests.

Add the flake8 package to requirements-test.txt and a run_flake8
docker entry point. This change does not require Python source
checking, but facilitates it.

Refs: #2896
Refs: #2615

Gerrit patch set 10419/1

Revision 29616660 - 05/16/2019 04:39 AM - Pascal Merz

Use Python 3

This updates the Python scripts shipped with GROMACS to use Python 3.5+.
Most changes done are either due to `__cmp__` not being available in py3
(replaced by `functools.total_ordering`), py2-iterators being replaced by
py3-iterators, or print statements being replaced by functions.
Additionaly, some changes in the Cmake files were made to select the
different Python version.

All python scripts having an explicit `#!/usr/bin/env python2` as a first
line were checked and updated. Larger changes were only needed for the
documentation scripts, other scripts did not require any changes. Scripts
not explicitly using shebangs were also checked, but no changes were
needed. The following scripts were not touched:
- all scripts under tests/physicalvalidation/ (already py3 compatible)
- all scripts under python_packaging/
- scripts/make_gromos_rtp.py - this file is not valid py2.7

Refs #2615

Change-Id: I2edbe213bc6401563934ce56f5fd6a39886d3095

Revision b277931f - 06/12/2019 11:13 PM - Mark Abraham

Fix pygments detection for sphinx for docs build

Got broken in 29616660 but somehow wasn't a problem in the docs build
on the dedicated slave for that.

Refs #2615

Change-Id: I407383bd0342decae56e4c60202ffbf190e8bb1b

Revision a9ed4bbd - 07/02/2019 12:03 AM - Pascal Merz

Fix module detection for physical validation

In I2edbe213, an implicit include() command was deleted, leading to
the module detection for physical validation not working anymore.
This change fixes the problem.

Refs #2615

Change-Id: If865227c2050f5739cd855261f6229f902fa0346

History

#1 - 08/18/2018 03:35 AM - Gerrit Code Review Bot

Gerrit received a related patchset '8' for Issue #2615.
Uploader: Pascal Merz (pascal.merz@colorado.edu)
Change-Id: gromacs~master~I2edbe213bc6401563934ce56f5fd6a39886d3095
Gerrit URL: https://gerrit.gromacs.org/6621

#2 - 09/04/2018 12:04 AM - Roland Schulz

Do we add a dependency on either the future or six package? If not how would I do e.g. iteritems (http://python-future.org/compatible_idioms.html#iterating-through-dict-keys-values-items)? What do we plan to do with releng? Do we preserve 2 compatibility there or make it Python 3 only?

#3 - 09/04/2018 08:02 PM - Pascal Merz

Roland Schulz wrote:

> Do we add a dependency on either the future or six package? If not how would I do e.g. iteritems (http://python-future.org/compatible_idioms.html#iterating-through-dict-keys-values-items)?

I would suggest to avoid any new dependencies. There are ways to keep the code py2/py3 compatible without adding dependencies. For example, I would suggest to use items(), keys(), values(), range(), etc - with an explicit list(...) or iter(...) around if necessary. This reduces the efficiency using Python2, but I wouldn't expect this to be noticeable in our case based on some tests I ran, and the documentation generation is probably not the most performance-critical part of GROMACS. This is the approach I used in https://gerrit.gromacs.org/c/6621/.

> What do we plan to do with releng? Do we preserve 2 compatibility there or make it Python 3 only?

No thoughts / preferences on this, but having a quick look at the scripts, compatibility seems doable if needed.

#4 - 10/08/2018 06:48 PM - Mark Abraham

- Category set to core library

- Target version changed from 2019 to 2020

Paul had a look at whether we could e.g. build the docs or API support with v3. Currently releng is implemented in Python v2, and runs the build script from the repo in the same python process. But anything called via cmake --build could be arranged to start in a v3 interpreter. But since anyway we aren't targeting v3 for anything in GROMACS 2019, we can leave all of this on the table for 2019, and go all in on v3 (including for gmxapi) for 2020.

Pascal's work is still up in Gerrit when the time comes.

#5 - 04/01/2019 04:31 PM - Eric Irrgang

- Related to Task #2831: Bump required version numbers of infrastructure for 2020 added

#6 - 04/01/2019 04:33 PM - Eric Irrgang

Note: Python 3.4 has reached end-of-life. https://devguide.python.org/#status-of-python-branches

#7 - 04/03/2019 11:37 AM - Eric Irrgang

We should consider how best to apply linting / PEP-8 compliance checking during the submission / code-review process.

#8 - 05/08/2019 05:46 PM - Mark Abraham

We mooted in the dev telco that python 3.5 sounds like a good minimum to require for GROMACS 2020. We want to balance what is supported by python devs, available on LTS desktops that they'll have in the timeframe of the next release from master branch, and available on HPC resources.

Python 3.5 is maintained until sep 2020

Python3 is 3.5 in 16.04 and 3.6 in 18.04.

Of HPC sites, archer, beskow, piz daint, juwels, bsc-power9, marconi all had python 3.6 (among others)?

So 3.5 seems like a good minimum requirement. Moving fowrard we'll test in Jenkins the oldest supported and the latest (at least).

#9 - 05/16/2019 12:33 PM - Eric Irrgang

- Status changed from New to Resolved

#10 - 06/12/2019 05:34 PM - Mark Abraham

- Status changed from Resolved to Accepted

We will need to change our docs-build link checker, as the one we used to use requires python2, is unmaintained, and does not support python3. I suggest https://github.com/bartdag/pylinkvalidator

#11 - 06/20/2019 07:20 PM - Szilárd Páll

This has been showing up in my cmake configure runs recently:

```
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.6.8", minimum required is "3.5")
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ModuleNotFoundError: No module named 'pygments'
```

AFACIT 29616660 of the follow-up might be causing it.

#12 - 06/21/2019 10:44 AM - Eric Irrgang

Szilárd Páll wrote:

> This has been showing up in my cmake configure runs recently:
> [...]
> AFACIT 29616660 of the follow-up might be causing it.

In a default CMake run, CMake checks whether the docs can be built, which includes checking for Python, Sphinx, and Pygments. Previously, *all* output regarding Pygments detection was suppressed. The new behavior is intended to be that output related to sphinx and pygments detection is produced on the first occasion of CMake configure, but not on subsequent configures. The behavior was not correct in the change you cite, but should have been corrected in a later change by Mark. If this behavior is not present in https://gerrit.gromacs.org/c/gromacs/+/11081 or if this behavior should be changed, then please comment in https://gerrit.gromacs.org/c/gromacs/+/11081

#13 - 07/04/2019 07:57 PM - Eric Irrgang

Szilárd Páll wrote:

> This has been showing up in my cmake configure runs recently:
> [...]
> AFACIT 29616660 of the follow-up might be causing it.

I recalled seeing this comment when I opened issue #3024 just now. I see now that I misremembered your observation, but this case is at least similar, I think.

#14 - 07/05/2019 02:57 PM - Mark Abraham

I have seen similar issues, and different ones when the machine didn't have python3 at all. Perhaps that was related to the way I use cmake -DGMX_DEVELOPER_BUILD=on routinely. We do expect cmake code to keep cmake quiet when optional things aren't found during subsequent runs.

#15 - 12/20/2019 11:56 AM - Paul Bauer

We now have Python3 working as the main Python version, with testing and docs building done on Gitlab.
What else is needed to mark the issue as resolved?

#16 - 12/27/2019 03:52 PM - Paul Bauer

- Status changed from Accepted to Resolved

Mark Abraham wrote:

> We will need to change our docs-build link checker, as the one we used to use requires python2, is unmaintained, and does not support python3. I suggest https://github.com/bartdag/pylinkvalidator

I have the same linkchecker still for the Gitlab builds and don't have any issues with it, so I'm not sure what you are referring to

#17 - 12/29/2019 10:33 AM - Paul Bauer

- Status changed from Resolved to Closed