openHPC
http://openhpc.community

Meeting of the Technical Steering Committee (TSC) Board

Tuesday, February 27th 2018
11:00am ET

Meeting Logistics

- https://www.uberconference.com/jeff_ef

- United States : +1 (510) 224-9559 (No PIN needed).

Antitrust Policy Notice

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

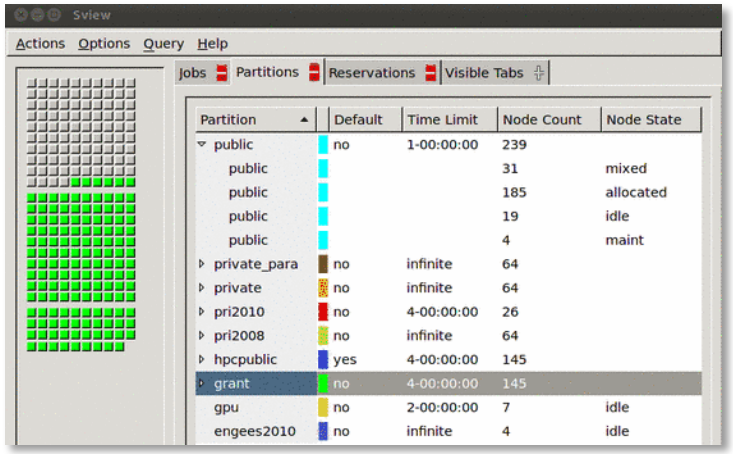
- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrust-policy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

THE LINUX FOUNDATION

Agenda

- TSC Teleconference update:
  - We will be switching to a new teleconferencing system (Zoom) beginning with the next meeting
  - Jeff will follow up with updated invite to include new meeting info
- Follow up on items from last time:
  - Update on submissions:
    - ✓PEARC'18 tutorial proposal submitted
    - ✓ISC'18 tutorial proposal submitted
  - Certificate of Origin is in place
    - ✓PR's are checked for "Signed-off" token
    - ✓PR guideline discussion added to wiki:
      - https://github.com/openhpc/ohpc/wiki/Contributions
- SLURM packaging update
- Python module follow up
- Component deprecation discussion

SLURM Packaging Update



- Had a request come up on the list asking about including "sview" in our SLURM build
- sview is a graphical user interface to view and modify SLURM state
  - it requires X (and is built with GTK)
- Canonical SLURM build will include "sview" if necessary GTK libs are detected at configure time
  - we could enable this addition by simply making GTK a build requirement (e.g. BuildRequires: gtk2-devel)
  - however, since this is a big dependency that would also get pulled in at install time, we wanted to be sensitive to those not wanting all of GTK
- Compromise:
  - include GTK build requirement, but isolate "sview" into a separate package -> slurm-sview-ohpc
  - users who want to use "sview" can install this optional package, but we would not include in default meta-package

Python discussion follow-up:

- From last time: OpenHPC will add python 3 versions (numpy, scipy, mpi4py, adios) along side updated python 2.7 versions for all OHPC releases through 2018

- Swappable modulefiles for python packages proved difficult, as compiler and mpi families must also be taken in to account.
  - We need a way to display multiple versions of module files at the same time
  - Happy to hear what other sites might do; proposing to expose both python2 and python3 packages within current hierarchy
    - two nomenclature proposals follow:

Python discussion follow-up: (cont)

- Option1: Prepend with short version tag (to indicate python 2/3)

```
------------------------------------------ /opt/ohpc/pub/moduledeps/gnu7 -------------------------------------------
 R/3.4.3          hdf5/1.10.1      mpich/3.2.1     ocr/1.0.1        openmpi3/3.0.0 (L)  plasma/2.8.0  superlu/5.2.1
 gsl/2.4          metis/5.1.0      mvapich2/2.2    openblas/0.2.20  pdtoolkit/3.25      scotch/6.0.4
 py2-numpy/1.14.1 py3-numpy/1.14.1

------------------------------------- /opt/ohpc/pub/moduledeps/gnu7-openmpi3 --------------------------------------
 adios/1.12.0     imb/2018.0        netcdf-cxx/4.3.0        phdf5/1.10.1       scalasca/2.3.1   slepc/3.8.2
 boost/1.65.1     mpi4py/2.0.0      netcdf-fortran/4.4.4    pnetcdf/1.8.1      scipy/0.19.1     superlu_dist/5.3.0
 fftw/3.3.6       mpiP/3.4.1        netcdf/4.5.0            ptscotch/6.0.4     scorep/3.1       tau/2.27
 hypre/2.11.2     mumps/5.1.2       petsc/3.8.3             scalapack/2.0.2    sionlib/1.7.1    trilinos/12.12.1
 py2-scipy/0.19.0 py3-scipy/0.19.0  py2-mpi4py/2.0.0        py3-mpi4py/2.0.0
```

Python discussion follow-up: (cont)

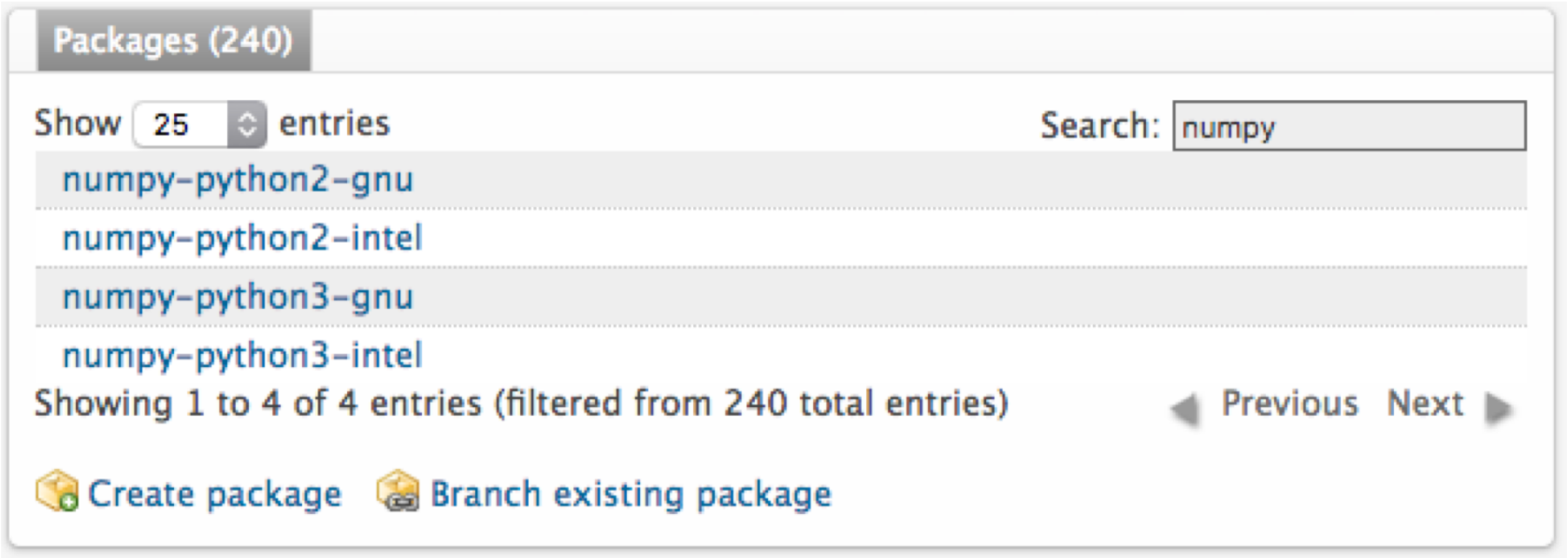
- Option2: Append a short version tag

```
------------------------------------------ /opt/ohpc/pub/moduledeps/gnu7 ------------------------------------------
 R/3.4.3          hdf5/1.10.1      mpich/3.2.1     ocr/1.0.1        openmpi3/3.0.0 (L)  plasma/2.8.0  superlu/5.2.1
 gsl/2.4          metis/5.1.0      mvapich2/2.2    openblas/0.2.20  pdtoolkit/3.25      scotch/6.0.4
 numpy-py2/1.14.1 numpy-py3/1.14.1

------------------------------------- /opt/ohpc/pub/moduledeps/gnu7-openmpi3 -------------------------------------
 adios/1.12.0     imb/2018.0        netcdf-cxx/4.3.0        phdf5/1.10.1       scalasca/2.3.1   slepc/3.8.2
 boost/1.65.1     mpi4py/2.0.0      netcdf-fortran/4.4.4    pnetcdf/1.8.1      scipy/0.19.1     superlu_dist/5.3.0
 fftw/3.3.6       mpiP/3.4.1        netcdf/4.5.0            ptscotch/6.0.4     scorep/3.1       tau/2.27
 hypre/2.11.2     mumps/5.1.2       petsc/3.8.3             scalapack/2.0.2    sionlib/1.7.1    trilinos/12.12.1
 scipy-py2/0.19.0 scipy-py3/0.19.0  mpi4py-py2/2.0.0        mpi4py-py3/2.0.0
```

Python discussion follow-up: (cont)

- Though not swappable, modules will not allow concurrent loading
- Additional macros added to OHPC_macros file to allow build of all permutations from a single spec



Component deprecation discussion (cont.)

- From last time, seems that general consensus was that we should define policy by which we can deprecate components
- Policy should focus on ability to build and use specific component(s): either in isolation or in combination with other ohpc components
  - note: decision criteria for deprecation would not include any notion of "usage" by the broader ohpc community

Component deprecation (cont.)

Starting Proposal Draft for Component Deprecation (1 of 2)

Purpose:
OpenHPC relies on a growing number of open-source projects to provide a variety of pre-built binaries and libraries common in HPC environments for multiple Linux distributions. While OpenHPC strives to maintain the entire set of selected components for each release, the purpose of this policy is to cover situations where a particular component cannot be maintained and outlines the OpenHPC deprecation process.

Outline:
Existing components within OpenHPC may be flagged for deprecation consideration for the following general reasons:

- build failures encountered using current OpenHPC development toolchain(s)
- runtime test failures encountered in OpenHPC integration test suite
- incompatibility with other component changes (e.g. API changes, etc)
- component functionality superseded by newer development project
- availability of component in binary form from other community repositories that are sufficient for use with OpenHPC
- incompatible license change
- unresolved security issues
- introduction of incompatible dependency requirements
- upstream source removal/deprecation
- miscellaneous issues that prevent component from being used as desired in OpenHPC environment

Component deprecation (cont.)

Starting Proposal Draft for Component Deprecation (2 of 2)

If any of the issues outlined are encountered, OpenHPC maintainers will first try to resolve the problems directly through creation of patches and interaction with the relevant upstream development parties. Generic patches devised for use within the OpenHPC build process will be submitted upstream for consideration. In cases where a particular problem cannot be resolved in time for the next planned OpenHPC release (N), the following process will be triggered:

- relevant component(s) will be demarcated with an "under deprecation consideration?|orphaned|stalled|stale|retired" flag in the Release Notes with a brief summary of the issue(s) encountered
- relevant component(s) will not be included in the update release (N)

Deprecation:

Components flagged as "under deprecation consideration" during the Nth release will continue to be analyzed during the development cycle for the subsequent OpenHPC release (N+1). If a component issue cannot be satisfactorily resolved via changes by upstream community or OpenHPC maintainers in time for the (N+1) release, the relevant component(s) will be deprecated.

Once deprecated, the relevant component(s) will not be included for any future releases unless it is selected for re-inclusion via the OpenHPC component submission process.

Component deprecation (cont.)

- Questions, thoughts, comments?:
  - do we like this 2-stage process proposal?
    - reasonable to keep component in "limbo" for 1 release cycle?
  - good identifiers/recommendations for "under deprecation consideration" and "deprecated" flags?
  - any exception policies or the like?

- If folks are generally favorable of this direction, I'll place in google doc and send to TSC for direct edits/comments