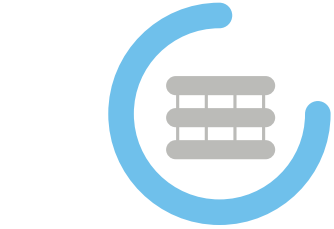


Meeting of the Technical Steering Committee (TSC) Board

Tuesday, April 17th 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

- Follow-ups:
  - pdtoolkit/gcc discussion
  - additional access statistics binning
- Review #6 Summary
- TSC administrative items
- Possible interim v1.3.5 release
- Use of OS distribution gcc

pdtoolkit – old gcc binary

- Recall GitHub ticket that was filed asking about the presence of old gcc binary in pdtoolkit packaging

  https://github.com/openhpc/ohpc/issues/687

  ```
  GNU C version 4.0.2 (University of Oregon, TAU, PDT (r128 - built Thu Jun 3 10:03:11 PDT
  2010)) (x86_64-unknown-linux-gnu)
  ```

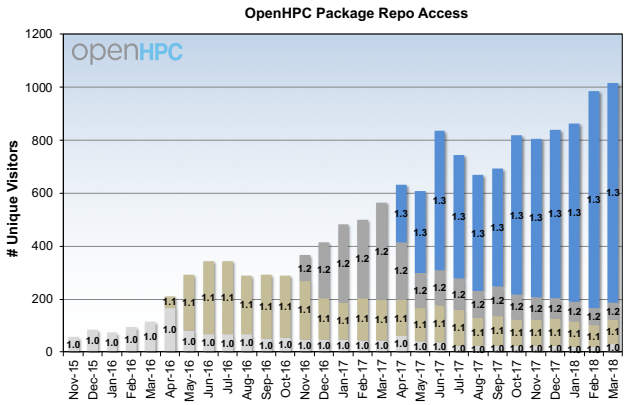
- We decided to reach out to upstream developer to get their take on if this. Here is what I received from Sameer (U. Oregon):

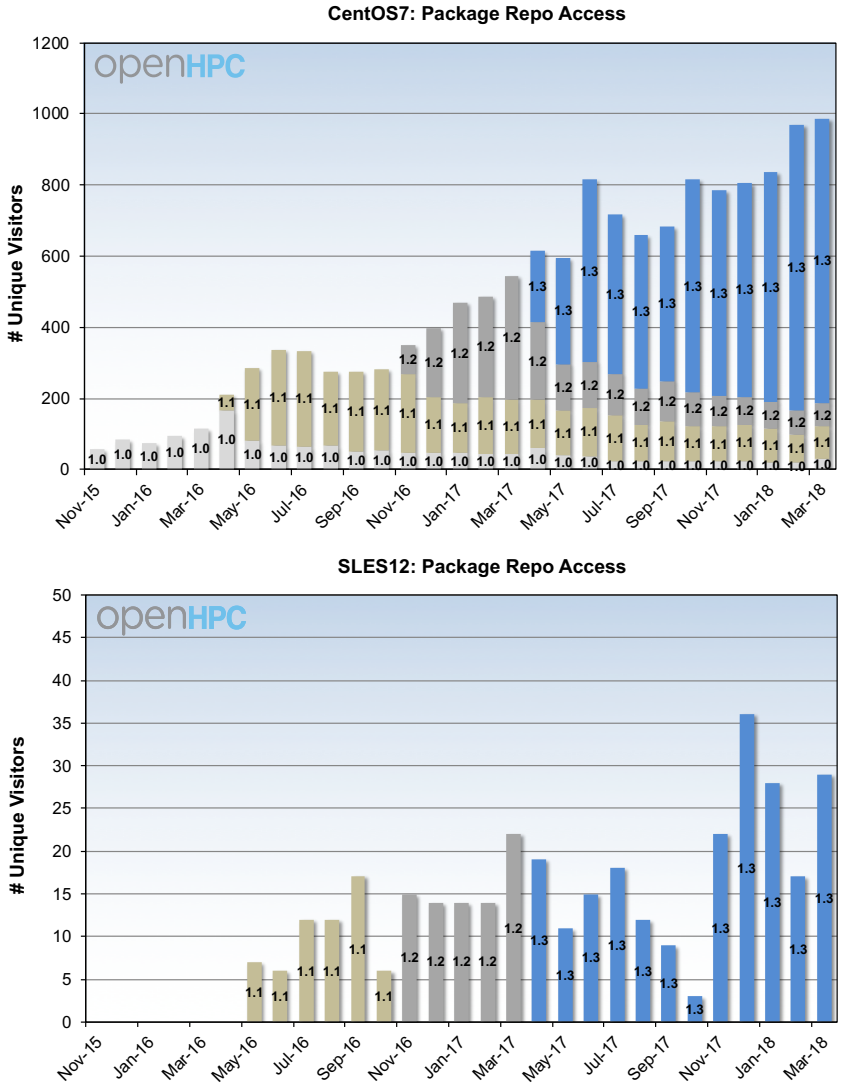
  "This is a patched gfortran compiler that doesn't really generate object files, but Program Database (PDB) files. These PDB files have information about location of functions and TAU uses this output to generate instrumented source code which is then compiled with the latest Intel or GCC system compilers. So, it poses minimal risks. Also, we have updated PDT to use gcc 4.8.5, so you may safely delete these binaries if you wish and TAU will pick up the 4.8.5 binaries by default."

- Based on this, would propose we continue to enable TAU with pdtoolkit tracing support, look into removing the older 4.0.x gcc binaries included with pdtoolkit next build cycle

Updated Usage/Access Stats (thru March 2018)

There was a request last time to look at the repo access stats by OS variant:





Updated Usage/Access Stats (thru March 2018)

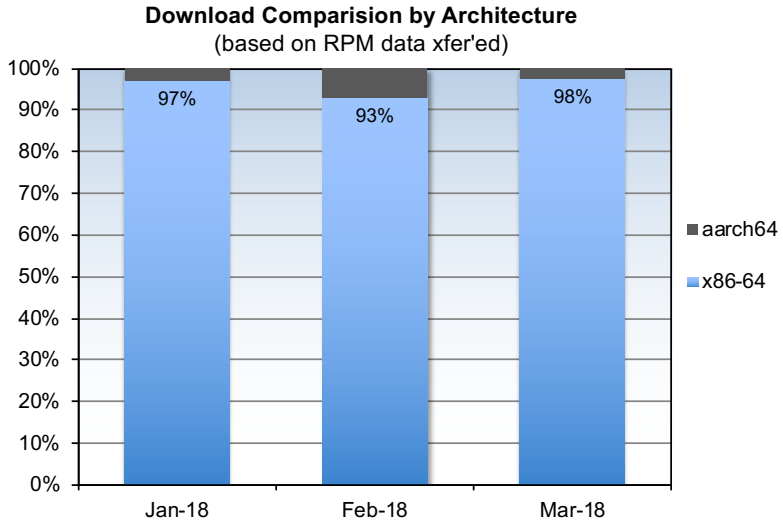
There was a related request last time to also look at the repo access stats by architecture:

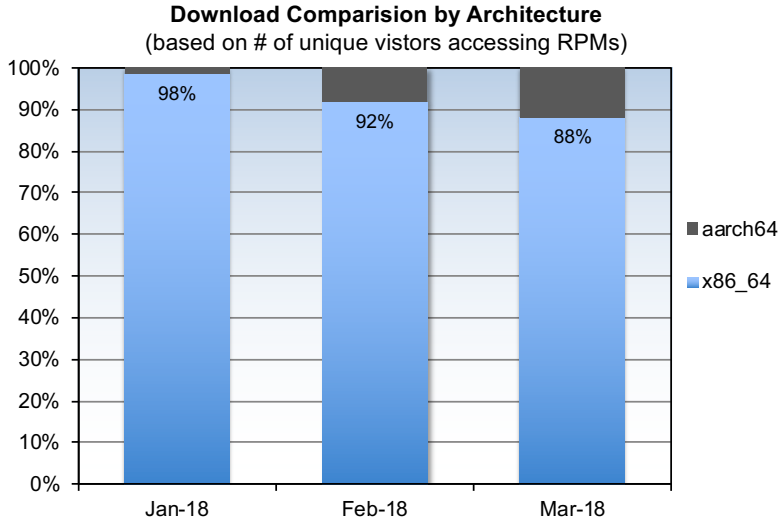
- a bit trickier since we publish a single repo with multiple architectures
- ie. there is a single repomd.xml file per distro

To provide some characterization, we have scraped the access logs to analyze two architecture specific file types:

- `(aarch64|x86_64).rpm`
- `(aarch64|x86_64).tar`

Look at amount of data xfer'ed and # of unique vistors accessing the (aarch64|x86_64) files and compute percentages in the plots shown here





Review cycle #6

Component Name	# of Reviewers	# of accepts	# of rejects	Avg. Priority
Extrae	6	6 (1*)	0	6.5
Paraver	6	6 (1*)	0	6.3
Dimemas	6	6 (1*)	0	6.3
OpenCoarrays	5	5	0	6.2
Charliecloud	5	3 (1*)	2 (1*)	4.2

- Decent agreement for first four submissions above
- Note: some discussion regarding license of dependency for paravar -> wxwidget
  - wxwidget has modified LGPL (they state it is is essentially the L-GPL with an exception stating that derived works in binary form may be distributed on the user's own terms)
  - the wxWindows Library Licence has been approved by the Open Source Initiative
  - initial CentOS build leverages wxGTK3-devel that is available in EPEL
- Assuming no objections, propose we accept first four above and add to backlog
- Charliecloud: less consistent agreement here. Per submission, to use Charliecloud container requires either (1) a kernel with user namespaces enabled or (2) willingness to install a setuid-root binary.
  - 2 accepts with fairly high priority (comments: keen on including HPC-focused container runtimes in OpenHPC so users have a choice, should be easy to deploy another (unprivileged) tool
  - 1 accept w/ changes: focus on packaging (or testing) to use w/ kernels that have user namespaces enabled (not using setuid binary)
  - 1 reject, 1 reject until changes; some comments:
    - does not seem to be widely used
    - application performance benchmarking not done (listed as future work)
    - new to HPC space, required sudo to run tests post-build
    - since we offer support to Singularity I believe that offering support to Charliecloud is not needed

TSC Administrative Items

TSC meeting schedule change?

- with the project evolving to more steady-state operations, we have not had an occasion to meet as frequently
- would folks be amenable to updating TSC meeting schedule to be bi-weekly until the summer? (frequency can be revisited when new TSC term starts)

TSC elections for 2018-2019 term:

- recall that TSC terms are for 1 year with the next term to start on July 1, 2018 (so election needs to happen in June)
- Jeff has setup list for nominations that is open for business: tsc-nominations@lists.openhpc.community
- can self-nominate or be nominated by others
- would like to send out call for nominations to the lists in May
- proposed election process dates:
  - call for nominations: on or before Friday, May 18, 2018
  - nomination deadline: Friday June 15th, 2018
  - electronic election opens (votes by current TSC members): Monday June 18, 2018
  - electronic election closes: Saturday June 30, 2018 (immediately following ISC)
  - results available first week of July
    - new TSC selects project leader in July 2018

Possible interim v1.3.5 release?

- There were a couple of reports on user list of stateful provisioning not cooperating with v1.3.4 release
  - https://github.com/openhpc/ohpc/issues/700
  - patch order fixed, updated builds available in OpenHPC:Update5:Factory
- Had some requests to enable two plugins for use with SLURM
  - jobcomp_elasticsearch and job_submit_lua
  - slurm rebuild to include the above also available in OpenHPC:Update5:Factory
- More difficult issue: a user reported an error using standard package manager (yum) upgrades after v1.3.4 release
  - based on the logs, user was starting from packages in the v1.3.2 release
  - error occurred when updating TAU, related to updated papi
  - recall that we introduced updated dependency conventions in v1.3.4 and rebuilt all recent packages that installed into /opt/ohpc/pub
    - for packages with compiler/MPI variants, we rebuilt all of the packages with the latest variants (e.g. gnu7, openmpi3, etc)
    - however, we did not rebuild the packages with older variants (e.g. gnu5 and openmpi1)
    - since this user had older variants installed, this is the reason for the error. Was able to work around by highlighting how to remove the old variants, and then opt-in for the new ones

Possible interim v1.3.5 release?

- Thoughts on whether folks would like to rebuild components that were available in the v1.3.x tree that used older compiler/MPI variants (this applies to gcc 5.x builds)?
  - rebuild using updated dependency analysis would allow users who installed ohpc earlier on in the v1.3. release cycle to do standard upgrades
  - not clear how many folks necessarily want the older variants if new ones are available, but current upgrade path is not exactly as intended (ie. yum/zipper upgrade)
- Decision 1: do we want to rebuild these older variants or not?
- Decision 2: if yes to #1, should we go for a quick interim v1.3.5 release (would allow for WW fix and minor SLURM updates) target around the end of this month?

Use of OS distribution gcc

- Adrian posted a question to TSC list about possibly using distro gcc variants to do all ohpc builds (versus distributing ohpc gcc builds)
  - newer gcc available under CentOS with devtoolset-7
  - gcc7 looks to be available under separate toolchain module with SLES
- So, while newer variants are available with distro add-ons, the question is would we want to use these as opposed to current approach that is fairly prevalent in HPC centers?
  - Renato points out this updated approach would lead to inconsistency in versions across distros (rhel/sles may be different)
  - can gain fixes and leverage testing by distro providers, but also limits ability for community to use newer/multiple versions and apply separate patches if needed
  - another issue I see is the expectation from HPC admins and users to have total control over versioning, ie, the reason we use modules in the first place:
    - very common to retain multiple versions of compiler toolchains over the lifespan of a system
    - users need to be able access multiple versions easily; convenient if we supply the compiler toolchain as well to enable modules and allow multi-version installs for admins who want it
    - can also avoid things like having to specify version with binary, e.g. from SLES12

      After installation, use gcc-7 or g++-7 instead of gcc/g++ to compile your program.
      ```
      $ gcc-7 hello.c -o hello
      ```
- Thoughts/discussion?
- If we end up doing a survey with Hyperion research, we could perhaps craft a question around this issue