My operating system is old but I don't care :)

I'm using NIX!

B. Bzeznik
BUX meeting, Vilnius
22/03/2016
CIMENT is the computing center of the University of Grenoble
CIMENT computing platforms

132Tflops (88 BullX)

<table>
<thead>
<tr>
<th>HPC platform</th>
<th>Data processing platform</th>
<th>Other thematic platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Froggy</td>
<td>Luke</td>
<td>r2d2, foehn, ceciccluster, digitalis,...</td>
</tr>
<tr>
<td>3200 Xeon E5 cores @2.6Ghz +18 GPUS K20m</td>
<td>~400 cores – heterogeneous systems and continuously evolving</td>
<td>~3000 cores heterogeneous systems federated from 10 clusters of member laboratories</td>
</tr>
<tr>
<td>BullX DLC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCS4 AE R3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High performance distributed storage (Lustre): 90 TB</td>
<td>Local scratches on nodes: 450 TB</td>
<td>NFS filesystems: a few TB per cluster</td>
</tr>
<tr>
<td>Infiniband FDR network</td>
<td>10 GE network</td>
<td>Infiniband QDR networks</td>
</tr>
</tbody>
</table>

Common storage (IRODS): 1 PB
Like everybody in the HPC world, we use environment modules.

Each computing cluster has its “site” modules that we compile “by hand”.

For the grid, we created an environment which holds its own glibc to have a uniform set of modules on every cluster.
CIMENT libraries repository

• PROBLEMS:
  - Hard to maintain
  - Not very easy to link against our libraries
  - A lot of dependencies, more and more complicated to build as the operating system becomes old
  - Recompilation at system change (or not, but...)
  - Jobs are not reproducible in the “sites” environments as soon as we upgrade the system
  - A feeling of doing something that could be more effective if we share our work
Solutions

- lmod
  https://www.tacc.utexas.edu/research-development/tacc-projects/lmod
  *Lua replacement for “modules”, with hierarchical support*

- Easybuild
  https://hpcugent.github.io/easybuild
  *User level automatic building*

- Spack
  https://github.com/LLNL/spack
  *User level automatic building*

- Nix
  https://nixos.org/nix/
  *A packaging system that allows user-level installs*

- Guix
  https://www.gnu.org/software/guix
  *A packaging system that allows user-level installs (the GNU one)*
NIX packaging system

- Nix is a free packaging system
- Packages are described with a functional language (the Nix language) → *derivations*
- Nix packages can be installed at the user level, into a shared /nix store
- Each package version is stored into a unique directory of /nix/store, starting by a hash
- Glibc is embedded, so Nix can run on top of almost every Linux flavour
NIX profiles

- Each user can have many profiles, allowing installation of different versions of a given package
- Rollback at a given version of a profile is very easy
- Administrator can set up system wide default profiles
- A profile is a set of symbolic links into ~/.nix-profile
- The PATH of the user contains ~/.nix-profile/bin
- switch profile:
  
  $ nix-env -switch-profile $NIX_USERPROFILE_DIR/my_test_profile
NIX profiles

(abstract from the Nix manual: https://nixos.org/nix/manual/)
NIX packages: nixpkgs

- Nixpkgs is a set of more than 10k packages
- To get all the latest derivations at once:
  
  ```
  $ git clone
  git://github.com/NixOS/nixpkgs.git
  ```

- install a package:
  
  ```
  $ nix-env -i -A gromacsMPI
  ```
  It will install the MPI variant of the gromacs package. If not already in the binary cache, it will be automatically compiled and be available for the other users directly as a binary

- remove a package:
  
  ```
  $ nix-env -e gromacs
Why is this a good solution?

- Focuses on reproducibility
- Offers an isolated development environment (nix-shell)
- Already +10k packages maintained by a strong community
- Optimized to share binaries and packages definitions among the users (multiuser mode + binary caches)
- Ease of use
- Ease to contribute (github pull requests)
- Ease of hacking and sharing derivations (a package description that allows to rebuild the package, eventually after some modifications)
- Users can install the same environment on their workstation
What do you need to make it available for your users?

- a shared `/nix` mount on all the nodes
- `nix-daemon` on one of your head node (`+socat` if you have several head nodes)
- a local repository (web server) if you want to setup a custom channel
  - to hold packages of non-free applications
  - to hold packages variants you've contributed to but that are not already in the official distribution
- NIXOS: the NIX operating system (nix + nixpkgs)
CIMENT contributions

- openib support into openmpi
- mpi support into Gromacs
- netcdf support into gdal
- mlx4 support into libibverbs (not yet merged)
- new packages: libmatheval, scotch, nco, libdap,...
- Non public: Intel 2016 compilers packaging
- A lot more to come!

You can also contribute!

A BullX Nix channel with bullxmpi one day? :-)

Try it! → $ curl https://nixos.org/nix/install | sh
Thank you!

https://nixos.org/nix/
Thank you!

Grenoble