NIX sur machines de calcul

B. Bzeznik
Journées mesocentres 2016
11/10/2016
Heterogeneous systems

Platforms of the HPC center of the University of Grenoble

**HPC platform**
- Froggy
- 3200 Xeon E5 cores @2.6Ghz + 18 GPUS K20m
- High performance distributed storage (Lustre): 90 TB
- Infiniband FDR network
- Remote visu nodes

**Data processing platform**
- Luke
- ~400 cores - heterogeneous systems and continuously evolving
- Local scratches on nodes 450 TB
- 10 Gbe network
- Remote visu nodes

**Other thematic platforms**
- ~3000 cores heterogeneous systems federated from 10 clusters of member laboratories
- NFS filesystems: a few TB per cluster
- Infiniband QDR network

**Common distributed storage (iRODS) 1Po**
CIMENT computing GRID

CIGRI RESTful API
- CIGRI server
  - OAR RESTful API
    - OAR Cluster A
    - OAR Cluster B
    - OAR Cluster C

IRODS storage
- Seen as a unique namespace

Jobs

Initial put of:
- input data
- code of application

Getting results

Results data

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

- **Easybuild**  
  [https://hpcugent.github.io/easybuild](https://hpcugent.github.io/easybuild)  
  *User level automatic building*

- **Spack**  
  [https://github.com/LLNL/spack](https://github.com/LLNL/spack)  
  *User level automatic building*

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

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

- **Container based solutions** (*Shifter, Singularity,...*)  
  *Light virtualization → maintaining system images*
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 (that can make containers pretty useless...).
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_USER_PROFILE_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
NIX packages: example

development/libraries/nco/default.nix

{ stdenv, fetchurl, netcdf, netcdfcxx4, gsl, udunits, antlr, which, curl }:

stdenv.mkDerivation rec {
  version = "4.5.5";
  name = "nco";

  buildInputs = [ netcdf netcdfcxx4 gsl udunits antlr which curl ];

  src = fetchurl {
    url = "https://github.com/nco/nco/archive/${version}.tar.gz";
    sha256 = "bc6f5b976fdfebdec51f2ebefa158fa54672442c2fd5f042ba884f9f32c2ad666";
  };

  meta = {
    description = "The NCO (netCDF Operator) toolkit manipulates and analyzes data stored in netCDF-accessible formats, including DAP, HDF4, and HDF5";
    homepage = http://nco.sourceforge.net/;
    license = stdenv.lib.licenses.gpl3;
    maintainers = [ stdenv.lib.maintainers.bzizou ];
    platforms = stdenv.lib.platforms.linux;
  };
}
NIX packages: example

[bzizou@bart:~]$ ldd /nix/store/gp50cqa35frra2zs3hnmgm7h8zvk32zlj-nco-4.5.5/bin/ncrename
linux-vdso.so.1 (0x00007ffd64f57000)
libnco-4.5.5.so => /nix/store/gp50cqa35frra2zs3hnmgm7h8zvk32zlj-nco-4.5.5/lib/libnco-4.5.5.so (0x00007f42d83c2000)
libnetcdf.so.7 => /nix/store/5ypb3jwflgsdkq52hi92n3jx5f1xwjg0-netcdf-4.3.3.1/lib/libnetcdf.so.7 (0x00007f42d506d000)
libcurl.so.4 => /nix/store/bjvwriaz0dp82bdy00sljxfqvm94pqps-curl-7.50.1/lib/libcurl.so.4 (0x00007f42d4dfa000)
libgsl.so.19 => /nix/store/fbh3zdyc51lga8qc25ddws70fk157sna-gsl-2.2/lib/libgsl.so.19 (0x00007f42d4999000)
libgslcblas.so.0 => /nix/store/fbh3zdyc51lga8qc25ddws70fk157sna-gsl-2.2/lib/libgslcblas.so.0 (0x00007f42d475d000)
libm.so.6 => /nix/store/6fix3zqpnahym18zpf2sxi2rwan55rgb8-glibc-2.24/lib/libm.so.6 (0x00007f42d423b000)
libgomp.so.1 => /nix/store/ly5dbisg2h0k3xnfdbk955m3pc4knvjk-gcc-5.4.0/lib/libgomp.so.1 (0x00007f42d401b000)
libpthread.so.0 => /nix/store/6fix3zqpnahym18zpf2sxi2rwan55rgb8-glibc-2.24/lib/libpthread.so.0 (0x00007f42d3dfe000)
libc.so.6 => /nix/store/6fix3zqpnahym18zpf2sxi2rwan55rgb8-glibc-2.24/lib/libc.so.6 (0x00007f42d3a60000)
libstdc++.so.6 => /nix/store/ly5dbisg2h0k3xnfdbk955m3pc4knvjk-gcc-5.4.0/lib/../lib64/libstdc++.so.6 (0x00007f42d36e8000)
libgcc_s.so.1 => /nix/store/ly5dbisg2h0k3xnfdbk955m3pc4knvjk-gcc-5.4.0/lib/../lib64/libgcc_s.so.1 (0x00007f42d34d1000)
libhdf5_hl.so.10 => /nix/store/allph90xffl82c17n50ivrn74n43ka74-hdf5-1.8.16/lib/libhdf5_hl.so.10 (0x00007f42d32b1000)
libhdf5.so.10 => /nix/store/allph90xffl82c17n50ivrn74n43ka74-hdf5-1.8.16/lib/libhdf5.so.10 (0x00007f42d2b24000)
libssh2.so.14 => /nix/store/clym2g8fd7r2ys5jfr05f18cf3dlqv-nghttp2-1.10.0/lib/libssh2.so.14 (0x00007f42d29e9000)
libssl.so.1.0.0 => /nix/store/55azyw1bcrzn8q5ganaav0cnqs2viwdn-openssl-1.0.2i/lib/libssl.so.1.0.0 (0x00007f42d254d000)
libcrypto.so.1.0.0 => /nix/store/55azyw1bcrzn8q5ganaav0cnqs2viwdn-openssl-1.0.2i/lib/libcrypto.so.1.0.0 (0x00007f42d2111000)
libexpat.so.1 => /nix/store/b5mwbrx8cldkchigggwkagaqw91xfjr89-zlib-1.2.8/lib/libz.so.1 (0x00007f42d16d2000)
libexpat.so.1 => /nix/store/6fix3zqpnahym18zpf2sxi2rwan55rgb8-glibc-2.24/lib/1d-linux-x86-64.so.2 (0x00007f42d868d000)
libexpat.so.1 => /nix/store/nq9bc7x8r8xh40yprwdrkxbxhmigawzw0-expat-2.2.0/lib/libexpat.so.1 (0x00007f42d82b1000)
libexpat.so.1 => /nix/store/nq9bc7x8r8xh40yprwdrkxbxhmigawzw0-expat-2.2.0/lib/libexpat.so.1 (0x00007f42d7f7b000)
libexpat.so.1 => /nix/store/nq9bc7x8r8xh40yprwdrkxbxhmigawzw0-expat-2.2.0/lib/libexpat.so.1 (0x00007f42d7e46000)
libexpat.so.1 => /nix/store/nq9bc7x8r8xh40yprwdrkxbxhmigawzw0-expat-2.2.0/lib/libexpat.so.1 (0x00007f42d7d1b000)
libexpat.so.1 => /nix/store/nq9bc7x8r8xh40yprwdrkxbxhmigawzw0-expat-2.2.0/lib/libexpat.so.1 (0x00007f42d7b86000)
libexpat.so.1 => /nix/store/nq9bc7x8r8xh40yprwdrkxbxhmigawzw0-expat-2.2.0/lib/libexpat.so.1 (0x00007f42d7a5b000)
libexpat.so.1 => /nix/store/nq9bc7x8r8xh40yprwdrkxbxhmigawzw0-expat-2.2.0/lib/libexpat.so.1 (0x00007f42d792c000)
libexpat.so.1 => /nix/store/nq9bc7x8r8xh40yprwdrkxbxhmigawzw0-expat-2.2.0/lib/libexpat.so.1 (0x00007f42d779c000)
libexpat.so.1 => /nix/store/nq9bc7x8r8xh40yprwdrkxbxhmigawzw0-expat-2.2.0/lib/libexpat.so.1 (0x00007f42d7667000)
libexpat.so.1 => /nix/store/nq9bc7x8r8xh40yprwdrkxbxhmigawzw0-expat-2.2.0/lib/libexpat.so.1 (0x00007f42d7537000)
libexpat.so.1 => /nix/store/nq9bc7x8r8xh40yprwdrkxbxhmigawzw0-expat-2.2.0/lib/libexpat.so.1 (0x00007f42d73a2000)
libexpat.so.1 => /nix/store/nq9bc7x8r8xh40yprwdrkxbxhmigawzw0-expat-2.2.0/lib/libexpat.so.1 (0x00007f42d7277000)
NIX packages: example

**development/libraries/nco/default.nix**

```nix
{ stdenv, fetchurl, netcdf, netcdfcxx4, gsl, udunits, antlr, which, curl }:

stdenv.mkDerivation rec {
    version = "4.5.5";
    name = "nco";

    buildInputs = [ netcdf netcdfcxx4 gsl udunits antlr which curl ];

    src = fetchurl {
        url = "https://github.com/nco/nco/archive/${version}.tar.gz";
        sha256 = "bc6f5b976fdffde5c1f2ebefa158fa54672442c2fd5f042ba884f9f32c2ad666";
    };

    meta = {
        description = "The NCO (netCDF Operator) toolkit manipulates and analyzes data stored in netCDF-accessible formats, including DAP, HDF4, and HDF5";
        homepage = http://nco.sourceforge.net/;
        license = stdenv.lib.licenses.gpl3;
        maintainers = [ stdenv.lib.maintainers.bzizou ];
        platforms = stdenv.lib.platforms.linux;
    };
}
```
NIX

Why is this a good solution?

- Focuses on reproducibility: everything is described into nix derivations (a kind of recipes for creating packages)
- No side effects: if I change a package, it's a new package and it does nothing to other packages depending on the old one (which is kept until no more packages depend on it)
- 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
- 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

- NIXOS: the NIX operating system (nix + nixpkgs)
- An OS that natively allows users to install/hack packages of their choice
- All the system configuration in a file: configuration.nix
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!
- Try it! → $ curl https://nixos.org/nix/install | sh
About containers

- Not at the same level: we can have NIXOS images
- Containers still allow you to create black boxes, with no easy reproducibility: ok to re-use an image, but what about upgrades or modifying the image 10 years layer?
- Do we need NIXOS inside a container..?
- ... or just a cluster under NIXOS?
Thank you!

https://nixos.org/nix/