Optimized and reproducible HPC Software deployment

with free software and GNU Guix

Pjotr Prins & Ludovic Courtès

FOSDEM

February 4th, 2017

UMC Utrecht/UTHSC GeneNetwork.org

GNU Guix

I am a software developer. What is it to me?

Holy grail

- Controlled software deployment
- Control the full dependency graph
- Write once and reproduce *any time*
- DRY

The stated problem

- HPC environments generally run old software (kernel, glibc, build tools)
- Newer software is installed through other routes
- Software targeting architectures such as Intel PHI require cross compilation

Supercomputer

- *Propriety* Intel/NVIDIA/Oracle software is up-to-date
- Python, R, gcc and llvm tend to be out of date
- Modern languages Dlang, Go, Julia are not supported
- Also Openblas and OpenCL libraries may not be supported
- True on most HPC systems

FOSS

- We are FOSS people - we want recent FOSS tools - how do we cope?
- Docker? Brew? Conda? EasyBuild? Static compiles from source?
- All suffer from bootstrapping and complexity resolving due to the underlying distribution
- And, yes, we want true reproducibility too

GNU Guix on HPC

- Guix installs in a root mounted /gnu/store ...
- Allow for running a privileged Guix daemon?
- Perhaps allow for a mounted /gnu/store? Build in other dir?
- If not, what to do?
- Tried PRoot live (nix-no-root, slow)
- Answer: relocatable Guix

`https://nixos.org/wiki/How_to_install_nix_in_home_(on_another_distribution)`

Insight 1

The first key insight: Guix store paths provide unique fingerprints (with Eelco Dolstra @FOSDEM):

```
/gnu/store/m9vxvhdj691bq1f85lpflvnhcvrdilih-glibc-2.23/lib/libc.so
```

Linked libraries

ldd ‘which ldc2‘

```
libconfig.so.9  /gnu/store/1v4anv1...-libconfig-1.5/lib/libconfig.so.9
librt.so.1      /gnu/store/m9vxvh...-glibc-2.23/lib/librt.so.1
libdl.so.2      /gnu/store/m9vxvh-glibc-2.23/lib/libdl.so.2
libpthread.so.0 /gnu/store/m9vxvh...-glibc-2.23/lib/libpthread.so.0
libz.so.1       /gnu/store/5992iq1...-zlib-1.2.8/lib/libz.so.1
libm.so.6       /gnu/store/m9vxvhd...-glibc-2.23/lib/libm.so.6
libstdc++.so.6  /gnu/store/9nifwk7...-gcc-4.9.3-lib/lib/libstdc++.so.6
libgcc_s.so.1   /gnu/store/9nifwk7...-gcc-4.9.3-lib/lib/libgcc_s.so.1
libc.so.6       /gnu/store/m9vxvh...-glibc-2.23/lib/libc.so.6
                /gnu/store/m9vxvh...-glibc-2.23/lib/ld-linux-x86-64.so.2
```

Relocate

- Guix binaries have unique fingerprints for PATHs
- Rewrite these with the target prefix (CONDA does this too)
- Only dependency left is the Linux kernel

After relocation

ldd ~/opt/ldc-test/ldc-1.1.0-pk9rkm4zvdp6pglam7s2/bin/ldc2

```
qlibconfig.so.9  ~/opt/ldc-test/libconfig-1.5-1v4anv1.../lib/libconfig.so.9
librt.so.1       ~/opt/ldc-test/glibc-2.23-m9vxvh.../lib/librt.so.1
libdl.so.2       ~/opt/ldc-test/glibc-2.23-m9vxvh.../lib/libdl.so.2
libpthread.so.0 ~/opt/ldc-test/glibc-2.23-m9vxvh.../lib/libpthread.so.0
libz.so.1        ~/opt/ldc-test/zlib-1.2.8-5992iq1.../lib/libz.so.1
libm.so.6        ~/opt/ldc-test/glibc-2.23-m9vxvh.../lib/libm.so.6
libstdc++.so.6   ~/opt/ldc-test/gcc-4.9.3-lib-9nifwk7.../lib/libstdc++.so.6
libgcc_s.so.1    ~/opt/ldc-test/gcc-4.9.3-lib-9nifwk7.../lib/libgcc_s.so.1
libc.so.6        ~/opt/ldc-test/glibc-2.23-m9vxvh.../lib/libc.so.6
                 ~/opt/ldc-test/glibc-2.23-m9vxvh.../lib/ld-linux-x86-64.so.2
```

What really happened here?

- All Guix packages are isolated in the store
- Path is a fingerprint, e.g. /gnu/store/m9vxvhdj691bq1f85lpflvnhcvrdilih-glibc-2.23
- Scan all files and replace fingerprints with relative path ∼/opt/ldc-test/glibc-2.23-m9vxvhdj691bq1f85lpf
- First attempt by using Eelco Dolstra's Patchelf tool worked for shared libs by rewriting RPATH in binaries

`http://nixos.org/patchelf.html`

Other files

- Text files that reference the store can be rewritten (Ruby, Perl, bash scripts)
- Some formats are not zero-terminated (compiled Python and JVM files)
- Also in ELF files there are references that are not zero-terminated
- Solution: keep the file path at exactly the same length and patch all

Insight 2

The second key insight: if a PATH gets rewritten with the exact same size string it will always work (unless there is encryption or some CRC checking)

```
/gnu/store/m9vxvhdj691bq1f85lpflvnhcvrdilih-glibc-2.23/lib/libc.so
/home/user/opt/ldc-test/glibc-2.23-m9vxvhdj691bq1f851p/lib/libc.so
```

Same size patching

1. start from store path, e.g.
   /gnu/store/m9vxvhdj691bq1f85lpflvnhcvrdilih-glibc-2.23
2. reverse the contents
   /gnu/store/glibc-2.23-m9vxvhdj691bq1f85lpflvnhcvrdilih
3. overwrite with prefix and shorten the HASH value to match the same size
   /home/user/opt/ldc-test/glibc-2.23-m9vxvhdj691bq1f851p

`https://github.com/pjotrp/guix-relocate`

Example

So, store path

```
/gnu/store/m9vxvhdj691bq1f85lpflvnhcvrdilih-glibc-2.23/bin/ldc2

prefix /home/usr/opt/ldc-test/ becomes
/home/user/opt/ldc-test/glibc-2.23-m9vxvhdj691bq1f851p/bin/ldc2

prefix /usr/local/share/ldc-1.1.0/ becomes
/usr/local/share/ldc-1.1.0/glibc-2.23-m9vxvhdj691bq1f8/bin/ldc2
```

Note: prefix can be up to ~ 40 letters long (by default)

Example

Download (42Mb), unpack (137Mb), install (3 sec.) and run Ldc the latest LLVM based D compiler 1.1.0:

```
wget http://biogems.info/contrib/genenetwork/ \
    pk9rkm4zvdp6pglam7s280x1x8y5rvbz-ldc-1.1.0-x86_64.tar.bz2
  (md5sum fe2508135eadc87fcc31027524c11ec5)
time ./install.sh TARGETDIR
TARGETDIR/ldc*/bin/ldc2 --version
```

https://forum.dlang.org/post/ckoiqzoatxyjlpakufsa@forum.dlang.org

So far...

Successfully built on Guix and packaged as relocatable binary:

- ldc2 1.1.0: the LLVM D compiler
- ruby 2.3.0: with ssl and nokogiri
- sambamba: tool used in many sequencing HPCs around the world (found sporadic segfault)
- more to come, including OpenCL, R, Python and Julia

`https://github.com/lomereiter/sambamba/issues/219#issuecomment-275838455`
`https://github.com/pjotrp/guix-relocatable-binary-packages`

Cross compile

- Install compilers that can cross compile binaries
- LLVM can output C code
- Provide GNU Guix packages for Intel PHI and NVIDIA TESLA
- GNU Guix has elegant support for different targets, including a build farm for Intel, ARM, MIPS, . . .

```
guix build --target=mips64el-linux-gnu guile
```

http://savannah.gnu.org/forum/forum.php?forum_id=7634

Conclusion

- GNU Guix provides a rolling distribution of Linux software packages with full control over the dependency graph (more tomorrow!)
- Now we have tested relocatable binary packages that do not require administrative privileges to install
- Two simple ideas can go a long way

See

`https://www.gnu.org/software/guix/packages/`

Future

Implications carry beyond HPC

- Automated builds with testing for different architectures
- Repository of binary packages
- One-click installs - download and run install.sh script
- Ship software easily
- Talking about a holy grail. . .

`https://github.com/pjotrp/gnu-install-bin`

Acknowledgements

- Roel Janssen (@roelj), Dennis Mungai (@Brainiarc7), and Frederick Muriithi (@fredmanglis) for helping with packaging D compilers, sambamba, Ruby packages, OpenCL etc.
- Eelco Dolstra for creating Nix, patchelf and discussions
- GNU Guix project leaders Ludovic Courtès and Ricardo Wurmus
- The GNU and GNU Guix communities (many, many talented individuals)
- Prof. Robert W. Williams and the GeneNetwork.org project