

INSTALL AND MANAGE GUIX ON A CLUSTER

Bruno Bzeznik - GRICAD

Guix Workshop - Montpellier - 2023-11-10



GUIX AT GRICAD

GRICAD

- GRICAD hosts the HPC center of "Université Grenoble Alpes"
- 7000 cpu-core (10000 more in 2024)
- 48 GPUS (+ ~40 H100 in 2024)
- About 100 nodes

GUIX AT GRICAD

- We are using **NIX** since 2015
- No more **environment modules** since 2017
- We've set up **GUIX** as an alternative more recently, but not to replace NIX (our users have the choice!)
- Other way to deploy applications: **Singularity/Apptainer** but not recommended by default

GUIX / NIX PACKAGING AT GRICAD

- Main packages maintainers and user support at GRICAD:
 - GUIX: *Pierre-Antoine Bouttier*
 - NIX: *Bruno Bzeznik (me)*

SETTING UP GUIX : THE MASTER NODE

MASTER NODE CONFIGURATION

- Prefer a dedicated, bare-metal host
- Storage
 - A RAID volume with at least a few TB
 - NFS exports
 - Avoid distributed storage due to *stats storm* sensibility (see later)
- Building (guix-daemon)
 - Enough CPU for parallel building
- Garbage collector

MASTER NODE SETUP: LOCAL ACCOUNTS

Check that you have installed guix and that the local unix accounts for the building processes have been created:

```
root@guix:~# grep _guixbuilder /etc/passwd
_guixbuilder0:x:996:996:Guix build user
0:/var/empty:/usr/sbin/nologin
_guixbuilder1:x:995:995:Guix build user
1:/var/empty:/usr/sbin/nologin
_guixbuilder2:x:994:994:Guix build user
2:/var/empty:/usr/sbin/nologin
_guixbuilder3:x:993:993:Guix build user
3:/var/empty:/usr/sbin/nologin
_guixbuilder4:x:992:992:Guix build user
4:/var/empty:/usr/sbin/nologin
_guixbuilder5:x:991:991:Guix build user
5:/var/empty:/usr/sbin/nologin
_guixbuilder6:x:990:990:Guix build user
6:/var/empty:/usr/sbin/nologin
_guixbuilder7:x:989:989:Guix build user
7:/var/empty:/usr/sbin/nologin
_guixbuilder8:x:988:988:Guix build user
8:/var/empty:/usr/sbin/nologin
_guixbuilder9:x:987:987:Guix build user
9:/var/empty:/usr/sbin/nologin
```


https://guix.gnu.org/manual/en/html_node/Installation.html

MASTER NODE SETUP: GUIX-DAEMON

Edit `/etc/systemd/system/guix-daemon.service`:

```
1 ExecStart=/var/guix/profiles/per-user/root/current-  
  guix/bin/guix-daemon \  
2     --build-users-group=guixbuild \  
3     --listen=/var/guix/daemon-socket/socket \  
4     --listen=0.0.0.0
```

The `--listen=0.0.0.0` option will make the daemon listen to all your networks (you may restrict to some local network interface)

MASTER NODE SETUP: NFS EXPORTS

- `/gnu/store *(ro)`
Big storage **read-only**
- `/var/guix *(rw, async)`
Users profiles -> **r/w**
- `/var/log/guix *(ro)`
Logs from the daemon

MASTER NODE: GARBAGE COLLECTOR

The garbage collector will parse users profile to know which packages are not used anymore.

Run it manually or automatically with cron:

```
0 5 * * 1 /usr/local/bin/guix gc -F10G
```

Warning: if users are using some applications directly from the store without using a profile, it may be a problem!

Warning: home directories must be mounted on the master node, for the GC to not delete apps used by active profiles!

SETTING UP GUIX: THE COMPUTING NODES

NODES: MOUNTING GUIX DIRECTORIES

```
head-node:/gnu/store      /gnu/store      nfs
defaults,_netdev,vers=3  0 0
head-node:/var/guix       /var/guix       nfs
defaults,_netdev,vers=3  0 0
head-node:/var/log/guix   /var/log/guix   nfs
defaults,_netdev,vers=3  0 0
```

NODES: PROVIDE A MINIMAL GUIX ENVIRONMENT

```
$ source /applis/site/guix.sh
```

```
export GUIX_PROFILE=$HOME/.guix-profile/  
export GUIX_USER_PROFILE_DIR=/var/guix/profiles/per-user/$USER  
export GUIX_DAEMON_SOCKET="guix://head-node"  
export PATH=/var/guix/profiles/per-user/root/current-  
guix/bin:$PATH  
export GUIX_LOCPATH=/var/guix/profiles/per-user/root/guix-  
profile/lib/locale  
export USERGUIXPATH=$HOME/.config/guix/current  
export INFOPATH="$USERGUIXPATH/share/info:$INFOPATH"  
source $USERGUIXPATH/etc/profile  
source $USERGUIXPATH/etc/bash_completion.d/guix
```

The very minimal should be:

- guix command in the PATH of the users
- GUIX_DAEMON_SOCKET set to the master host (guix-daemon)

OTHER CONCERNS

NETWORKING

- The master node acts as a *proxy* for the computing nodes
- The master node should have at least access to `https://ci.guix.gnu.org`
- The master node should also have access to the source files of the packages that are not pre-built into `ci.guix.gnu.org`, but it is possible to do *offline* packages with some effort (see https://guix.gnu.org/cookbook/fr/html_node/Acces-reseau-de-la-grappe.html)

DISK USAGE

If disk usage is a concern, you should be aware that even if you run the garbage collector on a regular basis, this may not free as much space as you think, because of the rollback capability of Guix that allows the users to revert to an old generation of a given profile. To gain GC efficiency, you should have to ask to your users to clean their *old* generations:

```
guix package --delete-generations=2m
```

(this will clean generations older than two monthes)

You also should ask to the users to to upgrade their profiles frequently:

```
guix pull  
guix upgrade
```

SECURITY

To track compromised libraries, for example a specific build of `glibc@2.25` before correction:

```
guix gc --referrers /gnu/store/...-glibc-2.25
```

ABOUT STAT STORMS

- A "Stat storm" may occur when you load an application having a lot of dynamic deps as every potential path for every library is checked
- With up to 200 nodes (maybe much more depending on the master node configuration), no problem with a centralized NFS server thanks to de facto centralized caching
- With jobs involving a lot of nodes and applications loading frequently (for example a sequential python script launched on every core of every nodes...), you may have storage overloading issues, especially if the store is on a distributed storage, because there's no centralized caching. Even with only 10 nodes...
- Some solutions have been proposed by the *store* FS community (NIX/GUIX/SPACK)

ABOUT *STAT STORMS*(REFS)

- <https://guix.gnu.org/en/blog/2021/taming-the-stat-storm-with-a-loader-cache/>
- <https://discourse.nixos.org/t/reducing-stat-calls-for-library-loading-during-application-startup/24358/7>
- https://archive.fosdem.org/2023/schedule/event/spack_stat_storm/
- <https://inria.hal.science/hal-04197724/file/RapportStageINFO4-2023-BRUN-Samuel.pdf>

QUESTIONS ?

QUESTION TO THE AUDIENCE

Have you already deployed Guix/Nix on your computing clusters?

QUESTION TO THE AUDIENCE

Did you experiment any issues with deploying / using Guix/Nix on your computing clusters ?

QUESTION TO THE AUDIENCE

What are for you, the barriers, if any, to provide Guix to your HPC users on your computing clusters?

CONTACT / LINK

- [Bruno.Bzezni@univ-grenoble-alpes.fr](mailto: Bruno.Bzezni@univ-grenoble-alpes.fr)
- https://ciment-grid.univ-grenoble-alpes.fr/guix_install/

Speaker notes