



GUIX AU CCIPL

Yann Dupont <Yann.Dupont@univ-nantes.fr>

CCIPL

Mésocentre des Pays de la Loire
Hébergé à l'Université de Nantes

(Centre de Calcul Intensif de Pays de la Loire)

→ GLICID en 2022

(ICI ECN + U. Angers + U. Le Mans + CCIPL)

CCIPL : CLUSTER *HÉTÉROGÈNE*

376 nœuds, 7196 cœurs, 3 réseaux, image unique

Loire (ex-PBS)	Xeon Westmere	IB 40
Waves v1 (Slurm)	Broadwell, Skylake, GPU, Cascade Lake	OPA 100
Waves v2.a	Cascade Lake	RoCE 2x25/100
Waves v2.b	AMD Epyc 7282	RoCE 2x25/100

UTILISATEURS & ÉQUIPE

306 utilisateurs des Pays de la Loire :

- Majorité non experts, quelques profils «pointus».
- Tous types de sciences = tous types de programmes
 - «Classiques» opensource, fortement optimisés
 - Outils «maisons» → (outils de développement)
 - Binaires divers et ... variés.
- ~ 1 ETP pour l'admin système, 0.5 pour les logiciels.

CALCULATEUR WAVES

Livré «clé en main» par l'intégrateur :

- Installation monolithique (2 serveurs / HA PCS)
- Déploiement par image (y compris frontale)
- Centos 7.2, Slurm 16.05, drivers OPA Intel
- Scratch BeeGFS
- Logiciels pré-installés (partage NFS)
Ensemble de compilateurs GNU, Intel et PGI
+ librairies et logiciels scientifiques pré-compilés.

ET... ?

Clé en main = confort initial de courte durée...

- Installation de nouveaux logiciels complexes
- Intégration des anciens nœuds
- Mise à jour annuelle délicate
 - Nœuds : Centos (7.2 → 7.3 ... 7.9)
 - Serveurs : Inextricable
- Autonomie nécessaire (pas/plus d'intégrateur...)



Pas de tout repos !

NŒUDS ET FRONTALE

- Upgrade glibc → Segfault de binaires générés par ICC (recompilation pas toujours possible)
- Incompatibilité de certains logiciels pré-installés

```
dupont-y@jaws ~]$ xmgrace  
--> Broken or incomplete installation - read the FAQ!
```

- des logiciels de la racine (mal) packagés !

```
[dupont-y@jaws ~]$ emacs  
emacs: error while loading shared libraries: libMagickWand.so.  
[dupont-y@jaws ~]$ ldapsearch  
LDAP vendor version mismatch: library 20444, header 20440
```

SATISFAIRE L'UTILISATEUR

- Enrichir et redéployer l'image ? **NON !**

```
yum install gedit evince
Installation 2 Paquets (+37 Paquets en dépendance)
Is this ok [y/d/N]:
```

- Demande de versions et dépendances précises (Ocaml,R, Python, Julia)...
- Optimisations/tunings spécifiques :
 - Nœuds Intel Waves : AVX 512, Loire : SSE4.1
 - Nœuds AMD Waves : AVX2

SOLUTION HISTORIQUE

Compilations «from scratch» puis modules.

Liens optionnels pour l'architecture créés au boot

Plusieurs versions concurrentes possibles

```
ls -al /opt/machine-dependant  
lrwx ... /opt/machine-dependant -> ../map/chezine003
```



Beaucoup de travail.

Dépendances très nombreuses.

Y compris à l'OS utilisé à la compilation.

CONTENEURS



Dans les cas où dépendances trop nombreuses.

- Docker ? **NON !**
Problèmes de sécurité importants
Pas d'accès simple aux GPU et réseaux OPA et IB
- Docker → Singularity
Davantage adapté au HPC, compatibilité
Problèmes pour créer l'image (non root)



6 mois après, recettes souvent non rejouables
images opaques, plus mises à jour...

MINICONDA (OU CONDA)

 CONDA

Utilisateur autonome !



place occupée sur le /home...

GESTIONNAIRES FONCTIONNELS



- Journées mésocentre 2017, Bruno présente Nix
- JCAD 2018, Ludovic présente Guix

Une façon de faire plus radicale ? 🤔

GUIX SUR TOUS LES NŒUDS

Tous les nœuds et frontales depuis fin 2018 :

```
controller.10g:/var/guix      2793287680    1331191808    1462095
controller.10g:/gnu         2793287680    1331191808    1462095
[root@cribbar012 ~]#
```

Source supplémentaire de logiciels,
Apporter de la reproductibilité (environnements...)
Faciliter la création de paquets «customisés»
Cf. suite des présentations (science reproductible)

GUIX

GUIX PRÉCHARGÉ POUR TOUS ?



Oui, mais non, mais oui.

- Conflit possible avec des modules pré-chargés (certains utilisateurs en raffolent... 🤒)
- Autonomie utilisateurs ? → Formation nécessaire !
- Démarche volontaire : guix pas dans le path
- Une vieille procédure initiale spécifique installe `.config/guix/channels.scm` (non global)



Guix évolue plus vite que nous...

INSTALLATION GLOBALE

/usr/local/bin est commun à toutes les frontales
certains binaires pointent sur un profil guix spécial

```
ls -al /usr/local/bin  
lrwx.. /usr/local/bin -> ..ccipl/global-local/bin  
  
..global-local/bin/borg -> /home/global-guix-profile/.guix-prof
```

- ❗ Tous les utilisateurs accèdent à des logiciels guix sans le savoir
(et évitent ainsi bugs borg #5219 et #5220...)

GÉNÉRALISABLE ?



éviter les compilateurs

```
ldd bonjour
linux-vdso.so.1 => (0x00007ffc4b1ea000)
libc.so.6 => /lib64/libc.so.6
libgcc_s.so.1 => ..store/xq..1s-gcc-9.3.0-lib/lib/libgcc_s.so.1
```

et plus tard :

```
ldd bonjour
linux-vdso.so.1 => (0x00007ffc4b1ea000)
libc.so.6 => /lib64/libc.so.6
libgcc_s.so.1 => not found
```

Le GC est passé par là !

(Les paquets compilés ne sont pas connus du GC...)

PAQUETS SPÉCIFIQUES GLICID

Pour les «véritables» utilisateurs Guix :
des paquetages spécifiques GLICID sont maintenus :

- gfortran11 (intéressant pour AMD EPYC)
- hdf5, openblas compilés pour gfortran11,
- OpenMPI 4.1.1 (support gcc11, et mises à jour libfabric, rdmacore, ucx... pour RoCE)
- xmgrace (avec motif) : disponible globalement.
- ...

MIROIR LOCAL ENRICHIS DE GUIX

Enrichi par glicid.scm + (Précédemment) gcc.scm

GLICID-PUBLIC > guix-mirror > Dépôt

ccipl guix-mirror / Historique Rechercher un fichier

 Merge remote-tracking branch 'upstream/master' into ccipl
Yann DUPONT authored 23 hours ago

Nom	Dernier commit
 build-aux	Merge branch 'version-1.3.0'
 doc	doc: Show how to enable Guile support for G...
 etc	Merge branch 'version-1.3.0'



Probablement plus nécessaire
Guix évolue plus vite que nous...

DES PAQUETS DE RÉFÉRENCE

Les paquetages de Guix servent aussi de référence :

Lourd travail manuel de recompilations de GCC11 et OpenMPI (libfabric, rdamcore, ucx) en modules 🤔

Idem benchmarks :

```
(define-public glicid-specific-intel-mpi-benchmarks
(package
(inherit glicid-intel-mpi-benchmarks)
(name "glicid-specific-intel-mpi-benchmarks")
(inputs `(("openmpi", glicid-specific-openmpi-gcc-11)
,@(package-inputs glicid-intel-mpi-benchmarks)))
)
)
```

TAR.GZ PORTABLES

Versions courantes intégrées à Guix ou customisées pour dépanner des collègues... (DSI, ex: keepalived)

REFONTE DU CLUSTER

Guix pour générer le volume OS volatil des VM
+ 1 volume DATA persistant.

- <https://doc.ccipl.univ-nantes.fr> est déjà une VM Guix
- VM Zabbix + PostgreSQL bien avancées
- À venir : keepalived, openldap, slurm, mariadb...
- Troncs communs entre VM : include/modules (clés SSH des administrateurs, configs communes)

Déploiements s'appuyant sur Ceph.



GÉNÉRATION DE VM

- `build.sh` (`guix system image $NAME.scm`) construit directement sur Ceph (RBD créé à la volée) crée un snapshot «candidate» prêt à être utilisée.
- `promote_candidate.sh` bascule l'image courante vers la dernière générée. Il suffit de redémarrer l'instance KVM pour basculer.
- `prune.sh` supprime les volumes/snaps inutilisés.
La génération de VM et l'utilisation se font sur des machines séparées, grâce à Ceph.

POURQUOI PAS PLUS DE GUIX ?

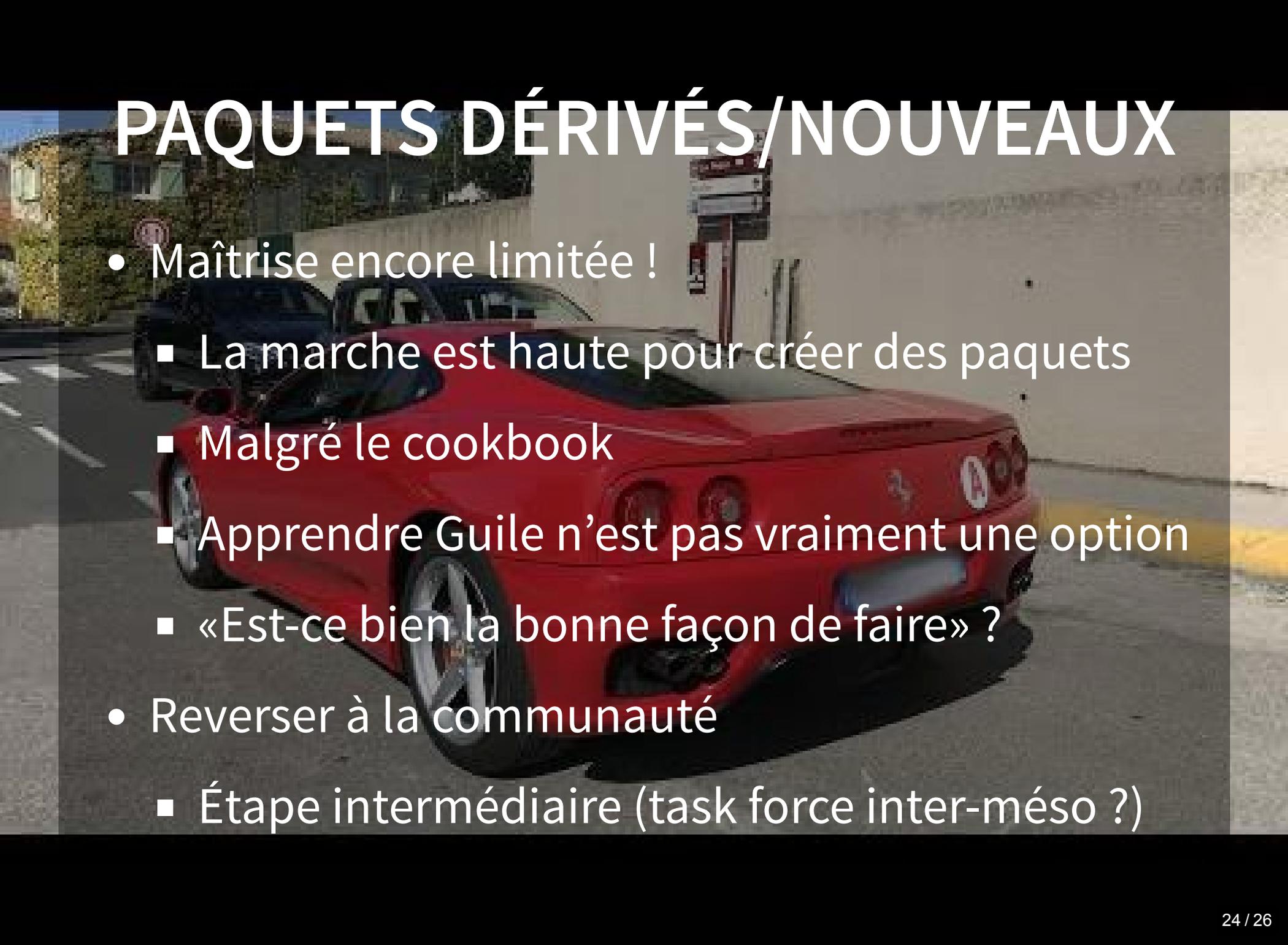
Question de priorité... et RH.

- Changer les habitudes des utilisateurs
 - Doc minimale mais de qualité
 - Pédagogie, formations assorties



Guix évolue plus vite que nous...

PAQUETS DÉRIVÉS/NOUVEAUX



- Maîtrise encore limitée !
 - La marche est haute pour créer des paquets
 - Malgré le cookbook
 - Apprendre Guile n'est pas vraiment une option
 - «Est-ce bien la bonne façon de faire» ?
- Reverser à la communauté
 - Étape intermédiaire (task force inter-méso ?)

LA SUITE

- CRIU, DMTCP
- Jupyter couplé au cluster (+ hub ?)
- VM «Guix» : récupération passwords et configs (bitwarden, gitlab)
 - S'inspirer de l'article de blog sur Ganeti ?
- IaaS OpenNebula / addon-context-linux (idem #154)
- Populariser
- Contribuer !

