Empaquetage et déploiement d'AVBP avec Guix

Assemblée générale ExaDI
14 mars 2025

Romain Garbage
romain.garbage@inria.fr

Introduction

Déploiement d'AVBP sur une machine Guix

Déploiement sur les supercalculateurs nationaux

Adastra : binaires repositionnables (relocatable binaries)

Jean-Zay : Singularity

Irene : PCOCC

Conclusion et perspectives

À propos

- CDD ingénieur de recherche au SED du centre INRIA de Bordeaux depuis le 1er décembre 2023
- Pas d'expérience de Guix avant cette date
- Utilisateur de logiciels libres depuis le début des années 2000
- Utilisateur d'une distribution Linux déclarative depuis fin 2022
- Première expérience professionnelle dans le secteur
- Pas de diplôme lié à l'informatique

Pourquoi empaqueter AVBP ?

- Logiciel classique du HPC, déployé sur de nombreux calculateurs dans le monde
- Grand challenge AMD Rome 2022 (Joliot-Curie)
- Permet de cibler les supercalculateurs nationaux (CINES, TGCC. . .)
- Argument en faveur de l'installation de Guix sur ces machines

Paquet AVBP dans Guix HPC

- définition de la pile logicielle et des options de compilation
- inclus dans le canal Guix HPC non-free (`https://gitlab.inria.fr/guix-hpc/guix-hpc-non-free`)
- permet de construire et de déployer en une seule commande un environnement de calcul (isolé ou non) sur une machine Guix
- permet de générer en unse seule commande des archives déployables sur les supercalculateurs sans Guix
- voir le fichier avbp.scm sur le dépôt Guix HPC non-free

Caractéristiques du paquet `avbp`

- version 7.15.4
- support CPU uniquement
- pile logicielle :
  - OpenMPI 4.1.6
  - GCC 11.4.0
  - Metis/Parmetis
  - OpenBLAS
- compilation avec CMake
- test COVO (inclus dans le source) avec un seul processus MPI après la compilation

Tutoriel

- `https://numpex-pc5.gitlabpages.inria.fr/tutorials/application-specific/avbp/index.html`
- comment utiliser le paquet `avbp`
  - sur une machine avec Guix
  - sur une machine sans Guix

AVBP sur supercalculateurs nationaux

- machines n'ayant pas Guix actuellement
- utilisation des définitions de paquets pour générer une archive (`guix pack`)
- déploiement sur Adastra et Jean-Zay
  - mononoeud/monocoeur : test `COVO` et suite de tests
  - tests multinoeuds sur Adastra avec Explo60M
    - simulations en mode batch et interactif, de 64 à 1600 processus MPI
    - extrait des résultats obtenus

N° de processus MPI	Temps d'exécution (s) : AVBP	boucle temporelle (s)
64	2302.20	2252.88
640	239.24	232.05
1280	130.86	119.88
1600	118.86	98.35

  - COVO/Explo60M sur Jean-Zay avec Singularity
    - simulations en mode batch et interactif, de 20 à 256 processus
  - COVO/Explo60M sur Irene avec PCOCC
    - simulations en mode batch et interactif, de 256 à 3480 processus

Introduction

Déploiement d'AVBP sur une machine Guix

Déploiement sur les supercalculateurs nationaux

Adastra : binaires repositionnables (relocatable binaries)

Jean-Zay : Singularity

Irene : PCOCC

Conclusion et perspectives

Sur une machine Guix

Positionner deux variables d'environnement :

- `AVBP_GIT_REPO`: chemin vers le clone local du dépôt git contenant les sources d'AVBP
- `AVBP_LIBSUP`: chemin vers le répertoire contenant le fichier licence d'AVBP

Instancier l'environnement logiciel et lancer la simulation

```
cd /chemin/vers/simulation/RUN # répertoire contenant run.params
AVBP_GIT_REPO=... AVBP_LIBSUP=... guix shell avbp -- avbp
# avec OpenMPI
AVBP_GIT_REPO=... AVBP_LIBSUP=... guix shell avbp openmpi@4 \
                                   -- mpirun --np 12 avbp
```

Introduction

Déploiement d'AVBP sur une machine Guix

Déploiement sur les supercalculateurs nationaux

Adastra : binaires repositionnables (relocatable binaries)

Jean-Zay : Singularity

Irene : PCOCC

Conclusion et perspectives

Méthodologie générale

- positionner les variables d'environnement `AVBP_GIT_REPO` et `AVBP_LIBSUP` sur la machine avec Guix
- générer une archive en utilisant la commande `guix pack`
- déployer l'archive sur la machine cible

Introduction

Déploiement d'AVBP sur une machine Guix

Déploiement sur les supercalculateurs nationaux

Adastra : binaires repositionnables (relocatable binaries)

Jean-Zay : Singularity

Irene : PCOCC

Conclusion et perspectives

Préparation de l'archive sur la machine Guix

Génération de l'archive

```
AVBP_GIT_REPO=... AVBP_LIBSUP=... guix pack -R -S /bin=bin -C zstd avbp
/gnu/store/xxxxxxxxxxxxxxx-avbp-tarball-pack.tar.zst
```

Copie de l'archive sur Adastra

```
scp /gnu/store/xxxxxxxxxxxxxxx-avbp-tarball-pack.tar.zst \
    user@adastra.cines.fr:/path/to/$CCFRWORK/avbp-pack.tar.zst
```

Déploiement sur Adastra

Extraction de l'archive dans $CCFRWORK

```
cd $CCFRWORK &&  mkdir avbp-pack && tar xf avbp-pack.tar.zst -C avbp-pack
```

Nettoyage de l'environnement

```
unset LD_LIBRARY_PATH
```

Lancement de la simulation

```
cd /path/to/simulation/run
srun -A user --time=0:20:00 --constraint=GENOA --nodes=2 \
     --ntasks-per-node=96 --cpus-per-task=1 --threads-per-core=1 \
     --mpi=pmi2 $CCFRWORK/avbp-pack/bin/avbp
```

Introduction

Déploiement d'AVBP sur une machine Guix

Déploiement sur les supercalculateurs nationaux

Adastra : binaires repositionnables (relocatable binaries)

Jean-Zay : Singularity

Irene : PCOCC

Conclusion et perspectives

Préparation de l'image Singularity sur la machine Guix

Génération de l'image

```
AVBP_GIT_REPO=... AVBP_LIBSUP=... guix pack -f squashfs \
                                            --entry-point=/bin/bash \
                                            -S /bin=bin avbp coreutils bash
[...]
/gnu/store/xxxxxxxxxxxxxxx-avbp-coreutils-bash-squashfs-pack.gz.squashfs
```

- Note : Singularity nécessite le paquet `bash` dans l'image.

Copie de l'image dans l'espace `$WORK`

```
scp /gnu/store/xxxxxxxxxxxxxxx-avbp-coreutils-bash-squashfs-pack.gz.squashfs \
    user@jean-zay.idris.fr:/path/to/$WORK/avbp.sif
```

- Note : l'image Singularity doit avoir une extension `.sif`.

Déploiement sur Jean-Zay

Copie de l'image dans l'espace autorisé

```
idrcontmgr cp $WORK/avbp.sif
```

- Note: sur Jean-Zay, il faut copier l'image Singularity dans l'espace autorisé (voir la documentation)

Activation de la commande singularity dans l'environnement.

```
module load singularity
```

Lancement de la simulation

```
srun -A user@cpu --nodes=1 --ntasks-per-node=40 --cpus-per-task=1 \
     --time=01:00:00 --hint=nomultithread --mpi=pmi2 singularity exec \
     --bind $WORK:/work $SINGULARITY_ALLOWED_DIR/avbp-bash.sif \
     bash -c 'cd /work/chemin/vers/la/simulation/run && avbp'
```

Introduction

Déploiement d'AVBP sur une machine Guix

Déploiement sur les supercalculateurs nationaux

Adastra : binaires repositionnables (relocatable binaries)

Jean-Zay : Singularity

Irene : PCOCC

Conclusion et perspectives

Préparation de l'image Docker sur la machine Guix

Génération de l'archive

```
AVBP_GIT_REPO=... AVBP_LIBSUP=... guix pack -f docker bash coreutils avbp
```

Copie de l'archive sur Irene

```
scp /gnu/store/xxxxxxxxxxxxxxx-bash-coreutils-avbp-docker-pack.tar.gz \
    user@irene-fr.ccc.cea.fr:/path/to/$CCFRWORK/avbp.tar.gz
```

Déploiement sur Irene

Import de l'archive avec PCOCC

```
pcocc-rs image import docker-archive:$CCFRWORK/avbp.tar.gz avbp
```

Lancement de la simulation

```
cd /path/to/simulation/RUN
ccc_mprun -p rome -N 2 -n 256 -c 1 -E '--mpi=pmi2' -m work,scratch \
          -A project_id -T 600 -C avbp -- avbp
```

Introduction

Déploiement d'AVBP sur une machine Guix

Déploiement sur les supercalculateurs nationaux

Adastra : binaires repositionnables (relocatable binaries)

Jean-Zay : Singularity

Irene : PCOCC

Conclusion et perspectives

Résultats obtenus

- Création d'un paquet Guix qui permet de générer la pile logicielle complète en une seule commande
- Déploiement sur les 3 supercalculateurs nationaux
- Validation d'une simulation lançant jusqu'à 3480 processus MPI

Perspectives

- Comparaison des performances entre un déploiement Guix et un déploiement manuel
- Ajout du support GPU (OpenACC)