

Checkpointing automatique avec CRIU et OAR

appliqué au downscaling de
variables climatiques

Bruno Bzeznik (GRICAD)
Maya Gueguen (LECA)

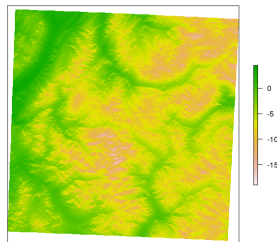
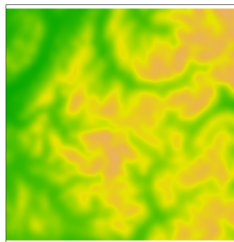
5 novembre 2019



Passer d'une carte satellitaire de température d'une résolution de 1km à un résolution de 25m, à l'échelle des Alpes, en prenant en compte :

- ▶ la topographie ;
- ▶ les changements de température selon la hauteur dans l'atmosphère,
- ▶ les radiations solaires (fonction de la topographie), corrigées par la couverture nuageuse.

Température
moyenne en
janvier dans le
Parc National
des Ecrins





Calcul des radiations solaires :

- ▶ quelle 'quantité' du ciel est visible depuis chaque position sans couverture nuageuse
- ▶ MAIS nécessité de le faire à un faible pas de temps pour diminuer les artefacts (ombres) $\Rightarrow \frac{1}{2}$ heure de pas de temps sur un mois ($\simeq 1500$ steps pour une donnée mensuelle par pixel)

Objectif : downscaling à 25m sur l'ensemble des Alpes
($\simeq 2,5$ milliards de pixels)

Moyens : **programmation en R couplé à SAGA** (SAGA=logiciel type ArcGIS pour faire du traitement d'objet cartographiques)

Ressources : parallélisation automatique sur un même noeud, mémoire minimale requise $\simeq 100$ Go,
 $15 \leq$ temps estimé ≤ 30 jours selon le mois
(janvier ou juin...)

Contrainte : **walltime max 3 jours!**

- ▶ Checkpoint/Restore In Userspace (pronounced kree-oo)
- ▶ Projet qui fait partie d'OpenVZ, démarré en 2011
- ▶ Bien intégré au kernel linux depuis le 3.11
- ▶ **Permet de sauver sur disque l'état d'un arbre de process pour le restaurer ultérieurement**
- ▶ Utilisation très simple en ligne de commande (root only)
- ▶ Au moment de la mise en place, impossible de faire un "self-dump":
https://criu.org/Self_dump
- ▶ Ne permet pas le checkpoint de jobs parallèles (avec communications entre plusieurs noeuds)

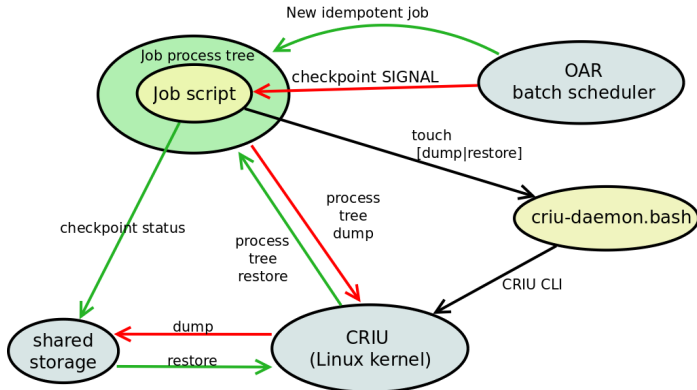


- ▶ OAR est un gestionnaire de jobs et de ressources, versatile et bien adapté aux "petites" configs, développé à Grenoble, avec des collaborations entre le LIG et GRICAD
- ▶ Il offre des hooks simples pour le checkpoint: un job peut recevoir un signal lui demandant de se checkpointer et de se relancer de manière idempotente



criu-daemon.bash

Afin de permettre à un job de faire un self-dump, un daemon très simple est mis en place sur les noeuds de calcul pour faire le passage de privilège entre le job OAR et la ligne de commande de CRIU.



Exemple de script OAR

1/3



```
#!/bin/bash
#OAR -n test_script
#OAR -t idempotent
#OAR -l /nodes=1/core=1,walltime=00:15:00
#OAR -O test.%jobid%.stdout
#OAR -E test.%jobid%.stderr
#OAR --project admin
#OAR --checkpoint 60
#OAR --notify mail:Bruno.Beznik@univ-grenoble-alpes.fr

# Handler for checkpointing signal sent by OAR
handler() { echo "Caught checkpoint signal at: `date`"
            echo "Checkpointing..."
            echo -e "$PROG_PID\n$(pwd)" > /var/lib/checkpoints/$OAR_JOB_ID.checkpoint
            }
trap handler SIGUSR2

[...]
```



[...]

```
# Load environment
source /applis/site/nix.sh

# A checkpoint exists, resuming it
if [ -e checkpoint_ok ]
then
  rm -f checkpoint/pidfile
  echo -e "$(pwd)" > /var/lib/checkpoints/$OAR_JOB_ID.resume
  # Wait for the restore (for pidfile to be created)
  while [ \! -e checkpoint/pidfile ]
  do
    sleep 5
  done
  PROG_PID=$(cat checkpoint/pidfile)

# No checkpoint, starting the program
else
  nohup ./boucle.sh > output_data &
  PROG_PID=$!
fi
```

[...]



```
[...]
# Wait for $PROG_PID ('wait' does not work in all cases, and
# bash kills the script when a trap occurs within a wait)
while [ -e /proc/$PROG_PID ]
do
    sleep 1
done

# Idempotent job exits with 99 code to be automatically re-submitted
exit 99
```



- ▶ Machine utilisée: DAHU (GRICAD). Chaque job occupant 1 noeud entier (32 coeurs, 192Go de RAM)
- ▶ Pour cette expérimentation, 3 gros jobs d'environ 15 jours ont pu ainsi être automatiquement découpés en une vingtaine de jobs de 48 heures
- ▶ Les jobs checkpointés ont tourné sur des noeuds différents, quasiment à chaque re-soumission
- ▶ Quelques bugs au début, relance de jobs "à la main", mais le dernier run OK, et dans tous les cas, on repartait à partir d'un checkpoint
- ▶ Resultats scientifiques: obtention de cartes de radiations solaires, puis de températures mensuelles à une résolution de 25m à l'échelle des Alpes

- ▶ Très utile pour des jobs longs, pour lesquels on ne maîtrise pas forcément la possibilité de sauver l'état
- ▶ Sources: <https://github.com/oar-team/oar/tree/2.5/misc/checkpointing/criu>
- ▶ Perspectives:
 - ▶ essayer les nouvelles fonctionnalités de self-dumping, implémentées en RPC: <https://criu.org/RPC>
 - ▶ tester DMTCP : <http://dmtcp.sourceforge.net>



Merci !

questions? ...

