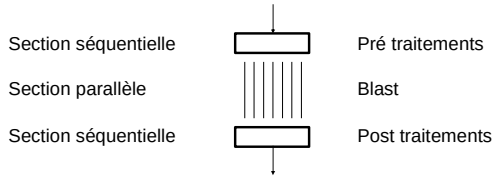
Paraloop: une interface d'accès générique aux ressources de calcul

Emmanuel Courcelle et Jérôme Gouzy
Laboratoire des Interactions Plantes Micro-organismes,
UMR CNRS-INRA 2594-441, B.P. 52627, 31326 CASTANET-TOLOSANE Cedex, France
Emmanuel.courcelle@toulouse.inra.fr, jerome.gouzy@toulouse.inra.fr

http://lipm-bioinfo.toulouse.inra.fr/paraloop

Le problème

Un exemple classique d'une section de pipe-line de traitement



Le prétraitement et le postraitement sont séquentiels
Le blast fonctionne plus rapidement sur une machine parallèle
Quelle machine parallèle ?

- Un cluster de calcul avec système de queue ?
- Une machine multiprocesseurs ?
- Un cluster de calcul sans système de queue ?

Vous souhaitez développer un pipeline qui puisse s'exécuter indifféremment sur l'une ou l'autre de ces trois architectures

La solution

La section parallèle s'écrit:

```
paraloop --program Blast \
         --input fichier.fasta \
         --db swissprot \
         --output blastoutput \
         --ncpus 10 \
         --wait
```

Le programme paraloop:

- Découpe le fichier d'entréee en 10 parties égales
- Envoie un job sur chaque processeur
- Ecrit des informations dans 10 fichiers Log
- Génère 10 fichiers de sortie
- Attend que tous les jobs soient terminés

De plus:

- Vous pouvez facilement interrompre et reprendre le travail
- Vous pouvez gérer la taille maximale des fichiers de sortie
- Lorsqu'un certain temps s'est écoulé, paraloop s'interrompt lui-même *entre deux séquences*, s'insère à nouveau dans la queue afin de laisser le processeur aux autres utilisateurs.

Votre programme

Les plugins

Paraloop est basé sur des "Plugins", c'est-à-dire des *objets spécialisés* dont l'interface est *standardisée.*

L'un d'entre eux permet de faire des **`Blast`**, mais il y a aussi des plugins généralistes:

- Le plugin **`Bioperl`** appelle un script externe pour chaque enregistrement d'un fichier lu par bioperl. Vous n'avez plus qu'à écrire le script externe.
- Le plugin **`Shell`** exécute une ligne de commande sur chaque processeur.

Pour des traitements plus spécialisés, mais répétitifs, vous pouvez écrire votre propre plugin, par exemple si vous souhaitez extraire à la volée les données de votre base de données relationnelle.

Votre machine

Les Schedulers

- Le mécanisme permettant de masquer l'architecture des machines se trouve encapsulé dans un objet appelé *Scheduler*
- La distribution contient des "schedulers" pour les architectures SMP (fork) et de calcul distribué utilisant rsh, ssh et PBS pro (qsub)

Si le type de machine dont vous disposez n'est pas actuellement supporté par paraloop vous pouvez écrire votre propre scheduler.

Un outil pour les centres de calcul ?

Bien sûr mais pas seulement

Nous avons tous des machines bi-processeurs, voire quadri-processeurs.
Nous avons (presque) tous des clusters de calculs
Nous avons souvent des comptes à l'IDRIS ou au CINES...

- **paraloop peut être installé par un simple utilisateur, il n'est pas nécessaire d'avoir les droits administrateur.**
- **paraloop permet de conserver le même environnement, quelque soit la machine sur laquelle vous calculez et ainsi d'assurer la portabilité de vos pipelines de calcul distribués.**

Téléchargement

- paraloop est un logiciel libre (perl >=5.6.1), il est couvert par la license CeCILL version 2 (adaptation de la GPL au droit français)
- Si vous avez développé un nouveau plugin ou scheduler, merci de nous l'envoyer, nous serons heureux de l'iintégrer à la distribution standard.
- Vous pouvez télécharger le programme à partir de ftp://ftp.toulouse.inra.fr/pub/paraloop

Sur le site web

- La présentation succincte de Paraloop
- La documentation complète
- Une Foire Aux Questions avec des forums publics