

# Soumission et Récupération de travaux

---

- ❑ Intrants et extrants
- ❑ Livraison des extrants(listes)
- ❑ Soumet

# Intrant

---

- Un travail ( script pour Korn shell ou Posix shell ) a exécuter sur une machine particuliere, un « cluster » , ou une machine parmi un ensemble de machines.

# Extrant

---

- Une liste ( « standard output » + « standard error » en un ou plusieurs morceaux mis bout à bout ) qui est le résultat de l'exécution du travail (script) grâce à un système de traitement par lots (batch).

# Livraison des listes

---

- ❑ Endroit: un répertoire ( \$HOME/listings par défaut )
- ❑ Avant: sur la machine sur laquelle le travail a été exécuté ou sur une autre, au choix de l'utilisateur (avec quelques restrictions).
- ❑ Maintenant: **OBLIGATOIREMENT** sur la machine sur laquelle le travail a été exécuté.

# Nom du fichier liste

---

- Dans le répertoire de livraison. Le nom du fichier liste sera:  
nom\_du\_travail.ppid.seqno  
voir l'option -jn pour le nom du travail, de même que les options -ppid et -seqno.

# Les sous-systèmes « batch »

---

- ❑ NQS : SX5 kaze + yonaka, SX6 yata (disparition sous peu)
- ❑ SGE ( Sun Grid Engine) : SX6 yata, pollux, « cluster graphique » (nouveau)
- ❑ PBS : les machines Linux de la recherche (5eme étage)
- ❑ LoadLeveler : la nouvelle machine IBM
- ❑ BATCH : soumission avec la commande batch sous Linux

# Soumet pas à pas

---

- Appel :  
**soumet** *nom\_du\_script* **OPTIONS**
- *nom\_du\_script* : nom du fichier contenant les commandes à exécuter (max 14 caractères sinon il faut donner un nom au travail avec **-jn** *nom\_du\_travail*)  
si – est utilisé comme *nom\_du\_script*, les commandes a exécuter seron lues de « stdin ».

# Soumet

---

- ❑ Le script soumet prépare un travail selon les options spécifiées en insérant certaines commandes Korn shell ou Posix shell avant et après les commandes qui constituent le travail de l'utilisateur. Ces commandes varient selon les options, le système de traitement par lots utilisé, et la machine d'exécution.



# **-mach [batch@]machine**

---

- ❑ Sur quelle machine, « cluster », ou groupe de machines le travail sera-t-il exécuté ? Quel système de traitement par lots ( batch) sera utilisé ? Par défaut, la soumission se fera sur la machine locale. batch peut prendre les valeurs nqs, pbs, sge, llv, bat. Si batch n'est pas spécifié, soumet essaie de déterminer le système à utiliser pour la machine demandée.

## **-t nsec**

---

- Temp CPU nécessaire à l'exécution du travail. Pour un travail MPI sous SGE, temps CPU nécessaire à l'exécution de chaque tâche MPI. La valeur par défaut est de 20 secondes CPU.

## **-cm nkB**

---

- Quantité de mémoire nécessaire à l'exécution du travail. La valeur par défaut est de 20MO.

## **-cpus MxN**

---

- Configuration CPU nécessaire. Dans le cas d'un travail MPI ou OpenMP pur, on utilise `-cpus N`. Dans le cas où on se sert de OpenMP dans les processus MPI, on utilise `-cpus MxN` où M indique le nombre de processus MPI et N le nombre de tâches OpenMP des processus MPI.

# -mpi

---

- Soumission d'un travail MPI (pur ou combiné avec l'usage de OpenMP). (voir -cpus et -nodes)

# **-jn nom\_du\_travail**

---

- Paramètre optionnel indiquant le nom du travail (max 14 caractères) si celui-ci est différent de *nom\_du\_script* ou si ce dernier est plus long que 14 caractères.

# -ppid n

---

- Seconde partie du nom du fichier de liste. Ce paramètre est optionnel et prend comme valeur par défaut \$\$, numéro de processus du « shell » qui exécute la commande soumet. Cette portion du nom du fichier de liste sert d'identificateur unique.

## **-seqno n**

---

- ❑ Troisième partie du nom du fichier de liste. Il s'agit du numéro de séquence du travail, le paramètre est optionnel et sa valeur par défaut est 1. Cette portion du nom du fichier de liste permet de différencier les membres d'une séquence de travaux clonés.



# -clone

---

- ❑ Préparer un clone identique du travail. Il suffit d'exécuter la commande  
    `export SOUMET=qsub`  
dans le travail pour provoquer la soumission d'un clone (copie identique) du travail courant à la fin de l'exécution de celui-ci.

# **-listing répertoire\_liste**

---

- Répertoire dans lequel le fichier de liste sera écrit. Si la variable d'environnement CRAYOUT est définie, ce sera la valeur par défaut pour ce paramètre, sinon la valeur par défaut sera \$HOME/tmp. Ce répertoire DOIT EXISTER sur la machine où le travail sera exécuté.

# -nosubmit

---

- ❑ Ne pas soumettre le travail mais produire un fichier tar appelé lajob.tar et qui permettra de procéder plus tard à la soumission au moyen de la commande *soumet\_lajob lajob.tar*  
lajob.tar renferme un fichier contenant le travail à soumettre et un fichier de commandes permettant de procéder à la soumission.

# -cl classe

---

- Information supplémentaire demandant d'utiliser une « classe d'exécution » spécifique pour le travail (utilisé uniquement pour des cas très particuliers)

# **-nec -noqsub -newtmp**

---

- ❑ Paramètres désuets ne devant plus être utilisés et qui seront supprimés dans les prochaines versions.

# Exemple 1

---

- ❑ soumet majob -mach yata -t 3600  
-cm 2G -jn yoyodyne-mpi -cpus 4
- ❑ Soumission du fichier majob au SX6 (yata). Ce travail utilisant MPI nécessite 2 GO de mémoire. 3600 secondes de temps CPU sont nécessaires pour chacune des 4 tâches. La liste reviendra sous le nom yoyodyne.\$\$1 dans le répertoire contenant les listes.

## Exemple 2

---

- ❑ `soumet ceci -mach pbs@turing -t 1800 -cm 130000`
- ❑ Soumission du fichier ceci à la machine turing en se servant du système de traitement par lots PBS. Ce travail a besoin de 1800 secondes de CPU et de 130MO de mémoire.

## Exemple 3

---

- ❑ `soumet majob -mach llv@ibm-conv -t 1200 -cpus 4`
- ❑ Soumission d'un travail OpenMP demandant 4 processeurs et 1200 secondes de temp CPU. Ce travail sera soumis à la machine `ibm-conv` (système de conversion IBM) en se servant du système de traitement par lots `LoadLeveler`.