



Grid Engine best practice

Nadia LAJILI

Tutorial GE@CC 22/11/2012

- Précautions préalables
- Optimisation de la production
- Règles de bonne conduite

▶ Précautions préalables



- Valider le code par un job interactif
 - compilations, tests
- Choisir le profil du job adéquat
 - séquentiel, tableau, parallèle, multicore
- Spécifier un projet et une priorité à la soumission
 - <qsub -P>
- Vérifier les permissions sur les queues privilégiées
 - pa*,mc*,huge,verylong,longlasting
 - <qconf -su longlastingqueue>

▶ Précautions à la soumission



- Demander les ressources appropriées
 - Complexes (HPSS, SPS, Xrootd..)
 - Physiques (mémoire, cpu, espace disque)
- Pour les jobs parallèles et multicore :
 - Spécifier la queue
 - Spécifier explicitement l'objet parallèle, nb cores
- Demander des ressources existantes
 - > Sinon le job reste indéfiniment en queue
 - nb slots max pour le mc* < 16
 - nb slots max pour le mc-interactif < 8
 - nb tâches max array job < 256

▶ Précautions à la soumission



- Demander la queue batch appropriée
 - s'assurer de l'adéquation ressource/queue
- Demander les ressources minimum
 - 64 MB pour la mémoire
 - 64 Mo pour l'espace disque
- Stockage de l'output
 - prévoir le système en fonction de sa taille :
 - * AFS, HPSS (> 1GB), SPS, Xrootd
 - prévoir la taille de l'espace

Optimisation de la production



- Examiner régulièrement l'efficacité des jobs
 - Éviter les jobs courts
 - Éviter les jobs lents
 - Suivre le statut des jobs
 - Jobs en erreur

<http://cc.in2p3.fr/docenligne/969#JobCheckStatus>
 - Jobs bloqués par le système / HOLD
- Demander les ressources physiques appropriées
 - S'assurer de l'adéquation besoins du job/consommations réelles

- Soumettre des jobs régulièrement
 - Proportionnellement à l'objectif
- Réguler la soumission
 - Job démon
- Éviter les accès intensifs aux systèmes de stockage
 - Copie des données dans \$TMPDIR
- Bien définir les priorités au sein du groupe
 - Définition des « share » pour les sous-projets

▶ Critères de mise en exécution des jobs



- Objectif (share) du groupe

- Etat global de la ferme

http://cctools.in2p3.fr/mrtguser/mrtguser/global/sge_global_slots_rm.html

- Répartition du share au sein du groupe (sous-projets)

- État d'utilisation des complexes requis par le job

http://cctools.in2p3.fr/mrtguser/info_sge_complex.php

- Ressources physiques demandées par le job
- mémoire, core, temps CPU

▶ Critères de mise en exécution



- Historique de la consommation du groupe
http://cctools.in2p3.fr/mrtguser/manips/objectives_attr.php
- Jobs en exécution en cours au sein du groupe
 - Slots dédiés au groupe déjà occupés
- Priorité du job au sein du groupe
 - Possibilité de fixer des priorités < 0
- Maintien d'un minimum de jobs par utilisateurs
 - Augmentation automatique de la priorité

▶ Règles de bonne conduite



- A proscrire
 - les « qstat » intensifs dans les jobs
 - les « qsub -ma »
- Éviter les soumissions massives
 - < 800 jobs
- Être réactifs aux mails d'alertes
- Privilégier l'interactif pour les tests
- Si on spécifie la queue uniquement:
 - valeur max par défaut sont attribuées

▶ Règles de bonne conduite



- Si on spécifie la queue + ressources
 - être sûr de l'adéquation queue/ressource
- Vérifier régulièrement l'efficacité des jobs
- Utiliser AFS dans des cas appropriés
 - Pas d'accès intensifs
 - Pas de stockage de masse
- Consulter les outils de monitoring pour suivre une production en cours
- Pour tout problème/question solliciter le support
 - <https://cc-usersupport.in2p3.fr/>

Liens utiles



- Monitoring global
<http://monicc.in2p3.fr/monitoring/>
- MRTG : Etat des ressources/groupe
http://cctools.in2p3.fr/mrtguser/info_sge_complex.php
- MRTG : Descriptif des queues GE
http://cctools.in2p3.fr/mrtguser/info_sge_queue.php
- GE : User guide
<http://docs.oracle.com/cd/E1908001/n1.grid.eng6/817-6117/>