Plateforme CICLAD

Calcul Intensif pour le Climat, l'Atmosphère et la Dynamique

http://ciclad-web.ipsl.jussieu.fr

Administrateur
Philippe.Weill@latmos.ipsl.fr

12/12/2012

Accès et Documentations

- Documentation et demande de compte:
  http://ciclad-web.ipsl.jussieu.fr
- Accès:
  - ssh user@ciclad.ipsl.jussieu.fr
  - ssh user@ciclad1.ipsl.jussieu.fr
  - ssh user@ciclad2.ipsl.jussieu.fr
- en cas de problème ou bien demande de logiciels supplémentaires:
  svp-ciclad@ipsl.jussieu.fr
- Liste de diffusion :
  http://courriel.ipsl.jussieu.fr/mailman/listinfo/ciclad-infos
- /home 20Go par utilisateur
- Lecture écriture de données sur /data et JAMAIS SUR /home

Accès et Documentations

- Documentation et demande de compte:
  http://ciclad-web.ipsl.jussieu.fr
- Accès:
  - ssh user@ciclad.ipsl.jussieu.fr
  - ssh user@ciclad1.ipsl.jussieu.fr
  - ssh user@ciclad2.ipsl.jussieu.fr
- en cas de problème ou bien demande de logiciels supplémentaires:
  svp-ciclad@ipsl.jussieu.fr
- Liste de diffusion :
  http://courriel.ipsl.jussieu.fr/mailman/listinfo/ciclad-infos
- /home 20Go par utilisateur
- Lecture écriture de données JAMAIS SUR /home

  mais sur /data ou autres suivant vos projets

Sauvegardes

- L'équipe systeme ne fait aucune sauvegarde autre que les homes
- Pour /home, si vous effacez un fichier vous pouvez retrouver la version de la veille dans

  /etherfs/home_backup

CICLAD Historique

- Projet commencé en 2007
- Maquette Février 2008 (50K€)
  - 1 noeud d'entrée sortie 9To de disques
  - 3 noeuds de calcul
- Mise en production Septembre 2008 (180K€)
  - 4 serveurs de fichiers (70To)
  - 10 noeuds de calcul (80 Coeurs)
- Mise en production Phase II Septembre 2010
  - Transformation en plateforme de stockage,calcul et distribution de donnée (centre de donnée , pôle de modèlisation)
- Début de phase III 1er Semestre 2013
  - liaison dédiée haut débit vers climserv
  - Augmentation capcité disque et calcul
  - Changement de version de linux pour le calcul
    - RHEL like 6.x

CICLAD c'est quoi Aujourd'hui?

- Un gros volume de stockage (> 1peta octet)
- Une grappe de calcul (352 Coeurs AMD)
  - 19 noeuds
- Une plate-forme de virtualisation pour la redistributions des données
- Le tout étant inter-connecté par un réseau rapide (20 et 40Gb/s) à faible latence (InfiniBand)

Plateforme de stockage

- **15 serveurs de fichiers (interconnexion infiniband)**
- **14 baies de disques (plus de 700 disques durs)**
- **5 Systèmes de fichiers parallèles Lustre**
  - **Très rapide en écriture et lecture**
  - **En cas de problème sur le système de fichier les écritures ou lectures sont bloquées en attendant la réparation et les programmes repartent dans la grande majorité des cas ou ils en étaient**

    (pratique en batch , moins en interactif ;-)
  - **N'aime pas du tout les grands nombres de petits fichiers (< 1Mo) pensez y dans vos applications**

Calcul

- 19 serveurs (352 Coeurs AMD 64 bit 2,3 GHz)
  - Machines avec 8 , 12, 32 ou 64 Coeurs
  - 4go de Ram par Coeur
  - Inter-connexion infiniband
  - 2 machines d'accès (ciclad1 et ciclad2)
    - Merci d'utiliser qsub -VI plutôt que de lancer directement vos programmes sur ces machines
- Gestionnaire de batch (torque-maui)
- Compilateurs et Logiciels scientifiques nombreux

Plateforme de virtualisation

- Commune à tous les services de l'IPSL
- 5 serveurs (1 pour sauvegarde et contrôle , 4 pour héberger des machines virtuelles)
- Système de virtualisation VMWARE
  - Hébergement de sites web, distribution de données

Configuration Logicielle

- Centos 5.x sur toutes les machines: http://centos.org/
- système de fichier LUSTRE: http://lustre.org
- système de batch torque : scheduler maui http://www.clusterresources.com/pages/products/torque-resource-manager.php
- Idl (6.4 et 8.0.1 (idl80)) matlab (2010b)
- compilateur fortran commerciaux :pgf95(Portland) ifort(Intel) nagfor (NAG)
- compilateur fortran gratuit : gfortran g77 g95
- Parallelisme : openmpi
- Autres : ferret, gmt, netcdf, cdo, nco, ncl … (en général dans /opt)

Système de batch (les commandes)

- Soumettre un job : qsub
- Combien de jobs dans les queues : qstat
- Détruire un job : qdel
- État des Noeuds : pbsnodes

  check-cluster
- Modifier un job : qalter

Les différentes queues d'exécution

- qstat -q pour les connaître
- 7 queues short std day days3 week weeks2 infini
  - short (default) --> 1Heure CPU
  - std --> 6Heures CPU
  - day --> 24Heures CPU
  - days3 --> 72Heures CPU
  - week --> 168Heures CPU
  - weeks2 --> 340Heures CPU
  - infini --> 840Heures CPU
- Plus la queue à un temps CPU long plus la priorité du job est faible sur la machine

Les limites utilisateur

- Toutes ces limites seront susceptibles d'évoluer en fonction des besoins et de l'utilisation du cluster au 01/09/2011 les limites sont :
- 1 utilisateur ne peut pas avoir :
  - plus de 24 Jobs actifs
  - Utiliser plus de 24 CPU
  - + de 3 jobs actifs dans la queue infini
  - + de 4 jobs actifs dans la queue weeks2
  - + de 8 jobs actifs dans la queue week
  - + de 16 jobs actifs dans autres queues
  - Par défaut la mémoire est limitée à **3go par process**
- *Pour éviter de trop charger le noeud interactif ,vous pouvez très facilement faire de l'interactif avec le système de batch:*
- qsub -VI [-q short|std|h12|day|days3|week|weeks2|infini]

Commande pbsnodes et check-cluster

- pbsnodes -l : liste les noeuds qui ne sont pas en fonctionnement
- pbsnodes -a : liste l'état de tous les noeuds
- Les autres options présentés dans le manuel sont réservé à l'administration
- Commande locale check-cluster (utilise pbsnodes)

```
[weill@ciclad1 ~]$ check-cluster
NODE            STATE                FREE-CPU
ciclad10        free                 2/8
ciclad9         job-exclusive        0/8
ciclad8         free                 1/8
ciclad7         free                 1/8
ciclad6         job-exclusive        0/8
ciclad5         free                 3/8
ciclad4         free                 8/8
ciclad3         job-exclusive        0/8
ciclad2         free                 4/6

    29 Active Jobs      51 of   70 Processors Active (72.86%)
                         8 of    9 Nodes Active      (88.89%)
Total Jobs: 44   Active Jobs: 29   Idle Jobs: 0   Blocked Jobs: 15
```

Commandes qstat

- qstat (sans options) : liste tous les jobs
- qstat -q : donnes les informations sur les queues
- qstat -a : donnes des infos étendues
- qstat -n : donnes les infos des noeuds sur lesquels tournent les jobs
- qstat -f “num_job”: donne des infos très étendues sur les jobs
- Un utilisateur peut voir les jobs de tout le monde dans la queue

Commande qdel

- Qdel numero_du_job (on récupere le numéro du job par qstat

Commande qsub (1)

- La premiere chose à savoir est que l'on ne peut soumettre que des shells scripts (pas des binaires)
- Le shell script minimum c'est

```
#!/bin/sh

mon_binaire
```

- il doit être exécutable et au lancement par défaut vous êtes dans votre répertoire d'accueuil

```
chmod +x mon_shell_script

qsub mon_shell_script
```

- En sortie vous récupérerez deux fichiers une fois le job terminé:
  - `“nom_du_script”.o”num_jobs”` : il contient la sortie standard de votre script + les informations de durée et de mémoire du jobs
  - `“nom_du_script”.e”num_jobs”` : il contient la sortie erreur de votre script

Commande qsub (2) Options

- qsub [-a date_time] [-A account_string] [-c interval] [-C directive_prefix] [-e path] [-h] [-I] [-j join] [-k keep] [-l resource_list][-m mail_options] [-M user_list] [-N name] [-o path] [-p priority] [-qdestination] [-r c] [-S path_list] [-u user_list] [-v variable_list][-V] [-W additional_attributes] [-z] [script]

Commandes qsub (3)

- qsub -q«nom_de_la_queue» mon_script
- qsub -VI
  - job interactif vous avez la main sur une machine (à condition qu'il y ai un processeur libre)
- qsub -j eo
  - Joindre les fichiers erreur er sortie en un seul
- qsub -k eo
  - Les fichiers d'entrée et de sortie sont créés dès le début du job par contre uniquement dans la home
- qsub -l nodes=ciclad3
  - Choisit le noeud (fortement déconseillé)

Commandes qsub(4)

- qsub -v variable_name=”variable_valeur” permet de passer des variables aux scripts car les arguments d'un script ne sont pas pris en compte sur la ligne de commande

  dans le script on utilise $variable_name

- Il est possible de mettre les options de qsub directement dans le script
  - Avec des directives :

  - #PBS -qweek
  - #PBS -k eo
  - #PBS -j eo

Commandes qsub(5)

- qsub -j eo -k eo job.sh : cela vous permet de suivre le déroulement des sorties dans votre home job.sh.exxx
- Il est possible d'utiliser en batch des programmes comme :
  - Idl
  - Matlab (avec l'option -nodisplay)
  - À condition bien sûr qu'il ne fasse pas de graphique X11

Commande qsub(6)

- Limite par défaut des jobs à 4 Go de mémoire virtuelle et 3 Go de memoire vive par défaut :
  - Comment lancer des jobs utilisant plus de 4gb?
    - Donner la taille mémoire voulu dans qsub:
    - pmem : mémoire par process
    - pvmem : mémoire virtuelle par process
  - qsub -l pmem=10gb -l pvmem=12gb

Mem: 32442980k total, 21798348k used, 10644632k free, 47112k buffers
Swap: 67111528k total, 2499232k used, 64612296k free, 14130248k cached

PID	USER	PR	NI	***VIRT***	***RES***	SHR	S	%CPU	%MEM	TIME+	COMMAND
12355	weill	25	0	***9164m***	***8.1g***	8.1g	S	44	26.3	1:08.55	MATLAB

qsub (7)

- Lancer un job à une heure donnée :
  - qsub -a heure :

  qsub -a 2010 # lance le job à 20H10

  qsub -a $(date -d '+2min' +%H%M) # lance le job 2minutes après la date courante)
- Spécifier le walltime ou le cputime :

  qsub -l walltime=20:00:00,cput=80:00:00 # 20H de walltime, 80H de cpu
- Dépendance de jobs : lancer un job après l'exécution réussi d'un ou plusieurs autres

  qsub -W depend=afterok:jobid[:jobid...]

Parallelisme

- soumission de jobs parallèles :
  - qsub -l “nodes=n:ppn=m”
    - ceci signifie : je veux n noeuds avec m processeurs par noeuds
    - exemples: qsub -l “nodes=1:ppn=8”
    - qsub -l “nodes=2:ppn=4”
    - qsub -l “nodes=4:ppn=1”
- Job mpi :
  - mpirun program
  - il est inutile de donner à mpirun le nombre de CPU (-np)
  - le système de batch s'en charge
- Au démarrage d'un job parallèle la variable shell PBS_NODEFILE contient le nom du fichier listant les noeuds affectés :
  - Exemples: [weill@ciclad1 ~]$ more foo.sh

    cat $PBS_NODEFILE

Parallelisme 2

- Au démarrage d'un job parallèle la variable shell PBS_NODEFILE
  contient le nom du fichier listant les noeuds affectés :
  - more foo.sh
    cat $PBS_NODEFILE
  - qsub -l “nodes=4:ppn:1” foo.sh
  - cat foo.sh.o56517
    [...]
    ciclad2.ipsl.jussieu.fr
    ciclad10.ipsl.jussieu.fr
    ciclad8.ipsl.jussieu.fr
    ciclad7.ipsl.jussieu.fr
    Epilogue Args:
    [...]
    Resource List: cput=01:00:00,neednodes=4:ppn=1,nodes=4:ppn=1
    Resources Used: cput=00:00:00,mem=4948kb,vmem=260216kb,walltime=00:00:14
    Queue Name: short
    Running Host: ciclad2.ipsl.jussieu.fr

Librairies mpi

- Il existes plusieurs implémentations de mpi
  - Openmpi, lammpi et mpich (non installé sur ciclad)
  - Pour openmpi préferer désormais les version 1.4.x qui utilise l'infiniband
- La commande **mpi-selector** permet à l'utilisateur de choisir son implémentation

  mpi-selector --query

  default:openmpi-1.4.3-ifort-x86_64

  level:user

  mpi-selector --list

  lam-x86_64

  openmpi-1.2.8-pgf-x86_64

  openmpi-1.4.2-g95-x86_64

  openmpi-1.4.2-gfortran-x86_64

  openmpi-1.4.2-pgf-x86_64

  openmpi-1.4.3-ifort-x86_64

  openmpi-1.4.3-pgf-x86_64

  openmpi-1.4.3-pgfgcc-x86_64

FAQ

- J'arrive pas à me connecter à ciclad.ipsl.jussieu.fr
  - essayer sur ciclad2.ipsl.jussieu.fr
- J'ai lancé un job il reste en queue alors qu'il y a de la place ?
  - showq -b peut vous donner la raison
- J'ai effacé par erreur un fichier dans /data
  - **NO BACKUP**
- J'ai un job qui se plante aléatoirement
  - Merci de fournir à svp-ciclad@ipsl.jussieu.fr l'emplacement des fichiers de sortie du job , vérifier aussi si le plantage à toujours lieu sur le même noeud

Avenir ...

- Limites :
  - Pas plus de 32 noeuds
  - Augmentation des filesystems
    - Si les budget suivent ...