

## FICHE TECHNIQUE CTIG

### DGA20

Diffusée le :

Destinataires :

#### OBJET – CHAMP D'APPLICATION

Mode d'emploi destiné aux utilisateurs de DGA20 mis à disposition par le CTIG afin d'effectuer des tâches qui nécessitent de faire tourner x jobs en simultané.

#### CONDITIONS REQUISES

Avoir un compte sur dga20.

Avoir un compte sur dga-cluster si besoin de faire tourner des jobs via SGE ou pour compiler avec Intel Fortran

#### DEFINITIONS ET ABBREVIATIONS

## SOMMAIRE

<b>1</b>	<b>PRESENTATION DE DGA20</b> .....	<b>3</b>
<b>2</b>	<b>CONNEXION A DGA20</b> .....	<b>3</b>
	2.1.1 <i>Ligne de commande (SSH)</i> .....	3
	2.1.2 <i>Connexion graphique (NX)</i> .....	3
<b>3</b>	<b>ORGANISATION ESPACE DISQUE</b> .....	<b>3</b>
<b>4</b>	<b>L'ORDONNANCEUR SGE (TRAVAUX BATCH)</b> .....	<b>3</b>
	4.1.1 <i>Présentation de SGE</i> .....	3
	4.1.2 <i>Files d'attentes</i> .....	4
	4.1.3 <i>Limites par nœud pour SGE</i> .....	4
	4.1.4 <i>Soumettre un job</i> .....	5
	4.1.5 <i>Suivre l'exécution d'un job</i> .....	5
	4.1.6 <i>Obtenir de l'information sur un job terminé</i> .....	6
	4.1.7 <i>Tuer un job</i> .....	6
<b>5</b>	<b>CONTACTS</b> .....	<b>6</b>

## 1 Présentation de dga20

---

dga20 est un serveur SuperMicro équipé de 2 processeurs Xeon 16 Core E5-2698v3 et de 512 Go de RAM. L'OS installé est RedHat Enterprise Linux 6.

### **dga20 est dédié à l'interactivité.**

Les utilisateurs peuvent lancer des travaux, mais si ceux-ci demandent beaucoup de ressources (mémoire, processeurs), il est préférable d'utiliser l'ordonnanceur de travaux batch (SGE).

### **Le compilateur Intel Fortran n'est pas disponible sur dga20.**

Pour compiler un programme avec ce compilateur, il faut aller sur **dga-cluster** (ssh user@dga-cluster) ou utiliser la classe SGE **ifort**.

**Attention : les jobs s'exécutant sur dga20 pourront être supprimés sans préavis s'ils pénalisent les autres utilisateurs.**

## 2 Connexion à dga20

---

### 2.1.1 Ligne de commande (SSH)

Pour se connecter utiliser un client SSH tel que PuTTY sous Windows, ou SSH sous linux :

```
ssh user@dga20.jouy.inra.fr
```

Votre mot de passe vous est demandé. Vous arrivez ensuite dans votre répertoire personnel : /home/user.

### 2.1.2 Connexion graphique (NX)

Pour se connecter au bureau graphique GNOME il faut utiliser un client NX comme x2Go. Voir « Comment accéder à nos serveurs ? » dans la FAQ du Wiki du projet « CTIG Environnement » sur la forge du dga ( <https://forge-dga.jouy.inra.fr/projects/ctig-env-base> ).

## 3 Organisation espace disque

---

A la connexion vous vous retrouvez dans votre espace personnel sous /home. Cet espace à une taille totale de 600 Go, mais vous êtes limités à 4 Go par utilisateur. Cet espace est sauvegardé tous les jours.

Sauvegarde (classe MCAIX1) : tant que le fichier existe sur disque, la dernière version de sauvegarde de celui-ci est conservée. Les deux versions précédentes du fichier sont conservées 90 jours. Lorsque le fichier est supprimé du disque, la dernière version est conservée 40 jours.

## 4 L'ordonnanceur SGE (travaux batch)

---

### 4.1.1 Présentation de SGE

L'ordonnanceur utilisé pour soumettre des jobs de calcul sur **dga-cluster** est SGE (Son of Grid Engine), c'est un logiciel open source dont la documentation complète est disponible librement à l'adresse : <https://arc.liv.ac.uk/trac/SGE>

La différence principale par rapport à l'ancien serveur dga12 est qu'il n'y a plus de jobs batch qui s'exécutent sur dga20. Maintenant, les jobs soumis par SGE s'exécutent exclusivement sur

le cluster. **Avoir un compte sur dga-cluster** est donc nécessaire pour pouvoir soumettre des jobs via l'ordonnanceur

Il a été mis en place plusieurs files d'attente (queues) suivant entre autres la durée maximum d'exécution du job.

L'objectif des files est de répondre à la majeure partie des besoins tout en optimisant l'utilisation des ressources de la machine.

### 4.1.2 Files d'attentes

Nom de la file	Durée maximum du job*	Nombre de cœurs disponibles	Max jobs (2) / utilisateur
bigmem	Infinie	160	120
copyq	Infinie	16	12
ifort	infinie	1	1
longq	48h	154	110
sas	infinie	16	12
sasindex(1)	infinie	16	12
unlimitq	infinie	30	21
workq	02h	1001	640

(1) réservées aux utilisateurs inscrits dans le groupe index-sge.

(2) le nombre maximum de jobs actifs pour un utilisateur est de 890 (toutes classes confondues)

\*Durée maximum d'utilisation du processeur

Les jobs en trop sont mis en attente.

### 4.1.3 Limites par nœud pour SGE

Ressources utilisables par nœud sur	72 Nœuds à 128 Go de mémoire	8 Nœuds à 256 Go de mémoire	2 Nœuds à 512 Go de mémoire
Mémoire	125 Go	250 Go	500 Go
CPU (nombre de cœurs)	16	16	32

## 4.1.4 Soumettre un job

En ligne de commande :

```
qsub - submit a batch job to Sun Grid Engine.
qsh - submit an interactive X-windows session to Sun Grid Engine.
qlogin - submit an interactive login session to Sun Grid Engine.
qrsh - submit an interactive rsh session to Sun Grid Engine.
qalter - modify a pending batch job of Sun Grid Engine.
qresub - submit a copy of an existing Sun Grid Engine job.
```

Exemple pour un job simple :

- 1 - Il faut préparer un fichier (script) contenant la (ou les) ligne(s) de commande
- 2 - Vos fichiers de sortie doivent être impérativement dirigés vers votre espace de travail
- 3 - Il faut **impérativement** préciser la quantité de mémoire à utiliser avec `-l h_vmem=` sinon il vous sera alloué 4Go maximum par job.
- 4 - Soumettre le job avec la commande de soumission (`qsub`)

Voici un exemple de script de soumission (il est aussi possible de lancer des jobs en interactif):  
`monscript.sh`

```
#!/bin/sh
#$ -o /work/.../output.txt
#$ -e /work/.../error.txt
#$ -q long
#$ -M mon_email@inra.fr
#$ -m bea
#$ -l h_vmem=3G
# Mon programme commence ici :
blastall -d swissprot -p blastx -i /work/.../z72882.fa
```

Toute ligne commençant par `#$` indique une option à exécuter par `sgc` (e.g. : `-q long` indique à SGE d'exécuter le job sur la queue longue.)

Pour soumettre :  
`qsub monscript.sh`

*Ne pas mettre le droit d'exécution au fichier `monscript.sh` car la commande `qsub` renverra le message d'erreur « `qsub: Unknown option` ».*

Alternativement, vous pouvez ne faire apparaître que les lignes de commandes dans le script et indiquer les options à l'appel de `qsub`.

Soit `qsub -l h_vmem=3G -q long -M mon\_email@inra.fr -m bea Monscript.sh`

Avec l'interface graphique ce sont les mêmes commandes. Pour lancer l'interface taper « `qmon` ».

## 4.1.5 Suivre l'exécution d'un job

Utiliser la commande `qstat` dont voici quelques options :

```
# qstat : liste les jobs de tous les utilisateurs en cours
# qstat -u user : donne les informations uniquement sur l'utilisateur
# qstat -j job_id : détail sur un job en particulier (numéro id attribué par SGE)
# qstat -s r : donne uniquement les jobs avec le status r(unning)
# qstat -f : visualise les jobs en cours par file et par nœud
```

La commande `qmon` permet de donner le même type d'information via une interface graphique.

### 4.1.6 Obtenir de l'information sur un job terminé

Utiliser la commande qacct :

```
#qacct -j job_id : donne les informations sur un job en particulier.
```

### 4.1.7 Tuer un job

```
#qdel -j job_id : tue un job en particulier
```

```
#qdel -u user : tue l'ensemble de mes jobs
```

On ne peut pas tuer les jobs d'un autre utilisateur.

## 5 Contacts

---

Pour toute demande, question ou problème, contacter l'équipe système du CTIG à l'adresse [ctig.systeme@dga.jouy.inra.fr](mailto:ctig.systeme@dga.jouy.inra.fr)