

Comment mieux utiliser le cluster URBM-SysDyn...

Nicolas Delsate

Projet de recherche : Gestion et développement d'un cluster de calcul
numérique intensif



/dev/null : le trou noir de Linux

```
[09:47:36]ndel@squeeze:~$ echo "Hello world"
Hello world
[09:47:45]ndel@squeeze:~$ echo "Hello world" > /dev/null
[09:47:50]ndel@squeeze:~$
```

ATTENTION : SGE redirige toute les sorties écrans dans les fichiers nameJob.oJobId

> /dev/null au bout de l'exécutable permet d'envoyer toute les sorties écrans (sauf les erreurs) dans le trou noir de Linux
ex. : ./monExecutable > /dev/null

/dev/null : Vos scripts

Avant

```
#!/bin/bash

#$ -S /bin/bash

cd
/home/math/ndelsate/TestScratch
#$ -cwd

echo "berk"
date
sleep 30s
date
echo "coucou"

./myprog unArgument

exit 0
```

Après

```
#!/bin/bash

#$ -S /bin/bash

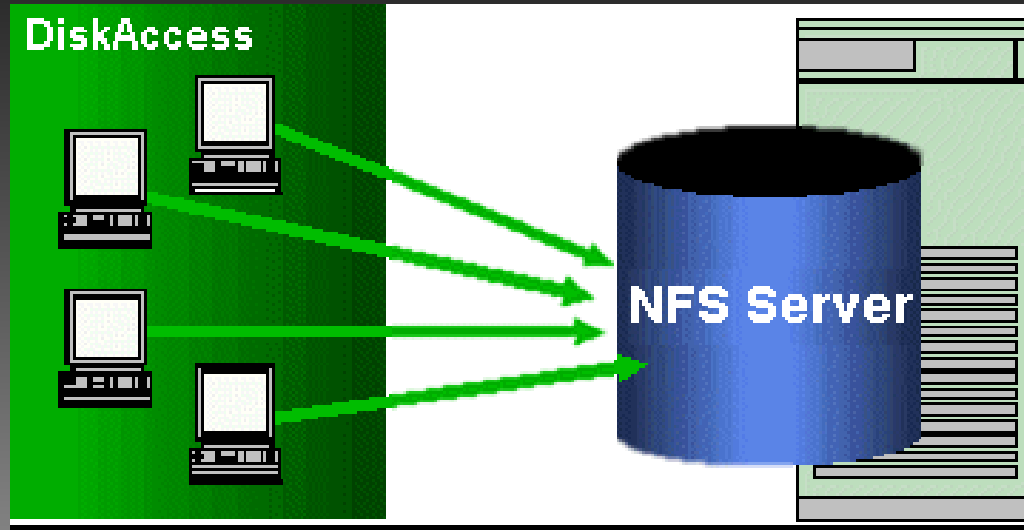
cd
/home/math/ndelsate/TestScratch
#$ -cwd

echo "berk"
date
sleep 30s
date
echo "coucou"

./myprog unArgument > /dev/null

exit 0
```

/scratch : les disques locaux



- Sur chaque noeud, le disque complet (6 To en 10000 t/min) de la frontale est monté en NFS

`/home/math/votreNom/`

Trop d'écriture/lecture peut charger le réseau et le disque

- Sur chaque noeud il existe un disque local (60 Go en 15000 t/min)

`/scratch`

Va accélérer le code et décharger le réseau

/scratch : Vos codes (après)

```
PROGRAM RecupVarEnviron
```

```
CHARACTER(LEN=200) :: FICHER, FICHERLOCAL, TMPDIR, cmde_cp
```

```
INTEGER :: Stat
```

Test si quelque chose est passé
en argument de l'exécutable

```
IF(iargc()==0) THEN
```

```
PRINT *, "VOUS n'avez rien passé! Donner le nom du fichier à créer" ; STOP
```

```
ELSE
```

```
CALL GETARG(1, FICHER)
```

```
END IF
```

Recupère un argument passé à l'exécutable,
en l'occurrence, ici, le nom du fichier à créer

```
write(*,*) 'Recuperation de la variable TMPDIR'
```

```
CALL GET_ENVIRONMENT_VARIABLE(NAME="TMPDIR",VALUE=TMPDIR,STATUS=Stat)
```

Récupère la variable d'environnement TMPDIR qui contient le répertoire du disque local

```
IF (Stat/=0) THEN
```

```
PRINT *, "Environment variable 'TMPDIR' not set!" ; STOP
```

```
ELSE
```

```
FICHERLOCAL=TRIM(TMPDIR)//'/'//TRIM(FICHER)
```

Cré le nom du fichier local

```
write(*,*) 'TMPDIR=',TRIM(TMPDIR),'Fichier local au noeud=',TRIM(FICHERLOCAL)
```

```
END IF
```

```
OPEN(UNIT=20,FILE=TRIM(FICHERLOCAL),ACTION="WRITE")
```

```
WRITE(20,*) "Le fichier local était, ",TRIM(FICHERLOCAL) ; CLOSE(20)
```

Ecris qq
chose dans
le fichier et
à l'écran

```
write(*,*) 'Copie du disque local au disque global'
```

```
cmde_cp= 'cp '//TRIM(FICHERLOCAL)//' '//TRIM(FICHER)
```

```
CALL system(TRIM(cmde_cp))
```

Appelle la commande
system.

Cré la commande
qui copie les fichiers

```
END PROGRAM RecupVarEnviron
```

Donc copie du fichier local vers le fichier global

Tests

- **Test 1 : pas de /dev/null** `./myprog TestAndrew.txt`

```
>> more coucou.sh.o32283
berk
coucou
Recuperation de la variable TMPDIR
TMPDIR=/scratch/32283.1.all.q Fichier local au noeud=/scratch/32283.1.all.q/TestAndrew.txt
Copie du disque local au disque global

>> more coucou.sh.e32283
id: cannot find name for group ID 1000

>> more TestAndrew.txt
Le fichier local était, /scratch/32283.1.all.q/TestAndrew.txt
```

- **Test 2 : /dev/null/** `./myprog TestAndrew.txt`

```
>> more coucou.sh.o32283
berk
coucou
```

```
>> more TestAndrew.txt
Le fichier local était, /scratch/32283.1.all.q/TestAndrew.txt
```

```
>> more coucou.sh.e32283
RIEN DU TOUT ICI
```

En C

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(int argc, char *argv[])
{
```

```
    char * nomFich = NULL, * nomLocal = NULL;
```

```
    char cmdeCP[200];
```

```
    FILE *file = NULL;
```

Récupère la variable d'environnement
TMPDIR (qui contient le répertoire du
disque local) et la stocke dans nomLocal

```
    nomLocal = getenv("TMPDIR");
```

```
    nomFich = argv[1];
```

```
    printf("Valeur de nomFich= %s\n",nomFich);
```

Récupère l'argument du pgm qui
contient le nom du fichier à créer

```
    strcat(nomLocal,"/");
```

```
    strcat(nomLocal,nomFich);
```

```
    file = fopen(nomLocal, "w");
```

```
    fprintf (file, "Nom du fichier local:%s\n",nomLocal);
```

```
    fprintf (file, "Nom du fichier final %s\n",nomFich);
```

```
    fclose(file);
```

Cré la variable "nomLocal" qui contient le
nom du fichier sur le disque local

```
    strcpy(cmdeCP,"cp ");
```

```
    strcat(cmdeCP,nomLocal);
```

```
    strcat(cmdeCP," ");
```

```
    strcat(cmdeCP,nomFich);
```

Cré la commande
qui copie les fichiers

```
    system(cmdeCP);
```

Appel la

commande system.

Donc copie du fichier local vers le fichier global

```
}
```