Durant la demo es va demostrar la forma d'instal·lar Putty i Xming (per suport gràfic) en entorns Windows i com accedir al clúster de la UOC a través d'aquesta eina i entorns Linux/OSX.

Es van comentar els diferents passos de la PAC amb les següents observacions principals:

- Es va comentar com especificar en SGE fitxers de sortida dependents de l'execució concreta a través de la variable de SGE amb l'ID del treball. Això soluciona la possible re-escriptura del fitxer de sortida. L'exemple mostrar és a continuació:

```
#!/bin/bash
#$ -cwd
#$ -S /bin/bash
#$ -N hostname
#$ -o hostname.out.$JOB_ID
#$ -e hostname.err.$JOB_ID

hostname
```

- Es va demostrar com obtenir el temps d'execució a través de la comanda del sistema "time". L'exemple de script corresponent és a continuació:

```
#!/bin/bash
#$ -cwd
#$ -S /bin/bash
#$ -N mm
#$ -o mm.out.$JOB_ID
#$ -e mm.err.$JOB_ID

echo "Size 500"
time ./mm 500
```

- Es va comentar que time mostra el temps d'execució per defecte a la sortida estàndard d'errors i es pot obtenir en el mateix fitxer de sortida simplement modificant la següent línia:

```
#$ -e mm.out.$JOB_ID
```

per escriure la sortida d'errors en el mateix fitxer que la sortida estàndard.

- Es va comentar la problemàtica de fer execucions molt llargues en un únic treball MPI. Això fa que la utilització dels recursos no sigui justa i que un usuari potencialment fa esperar a la resta durant molt de temps. Es va demanar que féssiu execucions que per treball trigués com a molt en l'ordre de minuts. Un exemple de com fer l'execució de diferents configuracions dins del script SGE es mostra a continuació:

```
#!/bin/bash
#$ -cwd
#$ -S /bin/bash
#$ -N mm
#$ -o mm.out.$JOB_ID
#$ -e mm.out.$JOB_ID

for i in {500,1000,1500}; do echo "Size $i; ./mm $i; done;
```

en la realització de la PAC cal fer varies execucions per tal d'obtenir conclusions evitant outliers que poden ser deguts per factors aliens a l'aplicació (per exemple manteniment al sistema). Es va comentar que tot i procurar fer scripts de no massa duració, tampoc s'espera que encueu centenars de scripts de molt curta duració - podeu fer varies combinacions dins d'un únic script SGE mentre el temps d'execució total no sigui massa elevat (s'ha de buscar un balanç).

- Es va comentar com realitzar l'execució de varies combinacions a través de varis treballs (un per execució). L'script SGE es mostra a continuació:

```
#!/bin/bash
#$ -cwd
#$ -S /bin/bash
#$ -N mm
#$ -o mm.out.$JOB_ID
#$ -e mm.out.$JOB_ID

echo "Size ${size}”
time ./mm ${size}
```

i des del command line es podria realitzar un bucle que fes les crides qsub, un exemple és a continuació:

```
for i in {500,1000,1500}; do qsub -v size=''$i'' h.sge; done;
```

Noteu que -v variable=‘VALOR’, assigna el valor indicat a VALOR en la variable ${variable} en l’script SGE.

- Finalment es va comentar una altra possible forma alternativa que, tot i ser menys eficient, no utilitza el mecanisme anterior de la variable de SGE. A continuació es mostra un exemple tot i que es pot realitzar de moltes formes diferents.

En aquest cas utilitzem la cadena KEY en un script template que copiarem per a cadascuna de les combinacions i substituirem amb el paràmetre adequat.

```
#!/bin/bash
#$ -cwd
#$ -S /bin/bash
#$ -N mmkey
#$ -o mmkey.out.$JOB_ID
#$ -e mmkey.out.$JOB_ID

echo "Size KEY"
time ./mm KEY
```

i des del command line es podria realitzar el següent bucle, tenint en compte que el script template anterior és “mm.sge”.

```
for i in {500,1000,1500}; do cp mm.sge mm$i.sge; sed -i 's/KEY/'$i'/g' mm$i.sge;
qsub mm$i.sge; done;
```