

Οδηγίες χρήσης του SGE στον Πυθαγόρα

Παναγιώτης Νάστου (pnastou@aegean.gr)
Παναγιώτης Τούρλας (mathm17027@aegean.gr)
Γιώργος Πετρούδης (mathm17016@aegean.gr)

Ο SGE αποτελεί έναν scheduler που αναλαμβάνει τη δρομολόγηση των εργασιών που αποστέλλονται στον Πυθαγόρα από χρήστες. Οι εργασίες αυτές αποστέλλονται από τον frontend σε κάποιον (ή κάποιους) από τους υπολογιστικούς κόμβους βάσει διαθεσιμότητας πόρων, γίνεται η εκτέλεσή τους στους κόμβους που δεσμεύτηκαν και στη συνέχεια επιστρέφεται ένα output file και ένα error log.

1 Προβολή πληροφοριών συστήματος

Πριν την υποβολή κάποιας εργασίας προς εκτέλεση, ο χρήστης μπορεί να δει μια λίστα με τα χαρακτηριστικά όλων των κόμβων μέσω της εντολής **qhost**. Με την εκτέλεσή της, εμφανίζεται μια λίστα με τα εξής πεδία:

1. **HOSTNAME**: Όνομα υπολογιστικού κόμβου.
2. **ARCH**: Αρχιτεκτονική-λειτουργικό.
3. **NCPU**: Πλήθος επεξεργαστών κόμβου.
4. **LOAD**: Φορτίο κόμβου τη στιγμή εκτέλεσης της qhost.
5. **MEMTOT**: Συνολική μνήμη κόμβου.
6. **MEMUSE**: Χρησιμοποιούμενη μνήμη τη στιγμή εκτέλεσης της qhost.
7. **SWAPTO**: Συνολική μνήμη swap.
8. **SWAPUS**: Χρησιμοποιούμενη μνήμη swap.

Περισσότερες πληροφορίες σχετικά με τη χρήση της εντολής **qhost** παρέχονται μέσω της **man qhost**.

2 Προβολή κατάστασης SGE

Μέσω της εντολής **qstat** ο χρήστης μπορεί να δει τις εργασίες προς εκτέλεση που έχουν αποσταλλεί στον SGE από όλους τους χρήστες. Σε περίπτωση που η ουρά αναμονής είναι άδεια από εργασίες, η εκτέλεση της **qstat** δεν επιστρέφει κάποιο αποτέλεσμα. Σε αντίθετη περίπτωση θα εμφανιστεί μια λίστα με τα εξής πεδία:

1. **job-ID**: Αναγνωριστικός αριθμός της υποβληθείσας εργασίας.
2. **prior**: Προτεραιότητα εργασίας, δυναμικά μεταβαλλόμενη.
3. **name**: Όνομα της υποβληθείσας εργασίας.
4. **user**: Όνομα του χρήστη που υπέβαλλε την εργασία.
5. **state**: Κατάσταση της εργασίας.
 - (d)ele tion
 - (E)rror
 - (h)old
 - (r)unning
 - (R)estarted
 - (s)uspended
 - (S)uspended
 - (t)ransfering
 - (T)hreshold
 - (w)aiting
6. **submit/start at**: Ημερομηνία και ώρα υποβολής της εργασίας.
7. **queue**: Η ουρά στην οποία έχει αποσταλλεί η εργασία, ισχύει για εργασίες στην κατάσταση **r** ή **s/S**.
8. **slots**: Το πλήθος των job slots που δεσμεύτηκαν.
9. **ja-task-ID**: Αναγνωριστικός αριθμός του job array που έχει υποβληθεί. Το πεδίο αυτό είναι άδειο όταν έχει υποβληθεί μια μόνο εργασία.

Περισσότερες πληροφορίες σχετικά με τη χρήση της εντολής **qstat** παρέχονται μέσω της **man qstat**.

3 Υποβολή εργασιών στον SGE

Η εντολή **qsub** αποτελεί τη βασική διεπαφή για την υποβολή εργασιών και χρησιμοποιείται ως εξής:

qsub (shell script)

Στο preamble του shell script δηλώνονται τα options που αφορούν την εκτέλεση και στη συνέχεια η πλήρης τοποθεσία του εκτελέσιμου και τα input arguments που χρειάζεται για να τρέξει. Για παράδειγμα, αν θέλουμε να τρέξουμε το εκτελέσιμο **prog** στην τοποθεσία **/home/username/code** με input arguments **3** και **5**, δίνουμε την εντολή:

qsub prog.qsub

Η απλούστερη μορφή που μπορεί να πάρει το prog.qsub είναι:

```
#!/bin/bash
```

```
/home/username/code/proc 3 5
```

Μπορούν όμως να χρησιμοποιηθούν και επιπλέον επιλογές στο preamble, με τις βασικές να αφορούν τα εξής:

- **-N (name)**: Το όνομα με το οποίο θα υποβληθεί η εργασία.
- **-i (pathname)**: Η πλήρης τοποθεσία του input stream που ενδέχεται να χρειάζεται το εκτελέσιμο.
- **-o (pathname)**: Η πλήρης τοποθεσία στην οποία θα επιστραφεί το output file με το πέρας της εκτέλεσης.
- **-e (pathname)**: Η πλήρης τοποθεσία στην οποία θα επιστραφεί το error log με το πέρας της εκτέλεσης.
- **-cwd**: Χρήση του current working directory. Αυτή η επιλογή επιτρέπει τη χρήση σχετικών αναφορών αντί για absolute pathnames.
- **-pe (args)**: Χρήση παράλληλου περιβάλλοντος, όπως το PVM (parallel virtual machine) ή ORTE (open run time environment).
(Σημείωση: Στην παρούσα φάση ο Πυθαγόρας υποστηρίζει μόνο το ORTE ενώ στο μέλλον θα υποστηρίξει και τα περιβάλλοντα PVM και MPICH2.)

Ένα πιο εξειδικευμένο παράδειγμα τέτοιου shell script είναι το εξής:

```
#!/bin/bash
# Request for 6 parallel processes
# under the parallel environment ORTE.
#$ -pe orte 6
# Put stderr in /tmp/foo
#$ -e /tmp/foo
# Put stdout in /tmp/foo
#$ -o /tmp/foo
# Use input stream input.txt in /tmp/foo
#$ -i /tmp/foo/input.txt
# The job is located in the current
# working directory.
#$ -cwd
proc 3 5
```

Υπόψιν ότι τα options βρίσκονται μετά από `#$`, ενώ τα σχόλια μετά από `#`. Εφόσον η εργασία υποβληθεί και εκτελεστεί, το output file και το error log θα αποσταλλούν στις επιλεγμένες τοποθεσίες ή την τοποθεσία από όπου έγινε η υποβολή της εργασίας (αν δεν έχει επιλεγθεί τοποθεσία).

Αν η υποβολή είναι επιτυχής, θα εμφανιστεί το μήνυμα:

Your job xx ("prog.qsub") has been submitted.

Περισσότερες πληροφορίες σχετικά με τη χρήση της εντολής **qsub** παρέχονται μέσω της **man qsub**.

4 Διαγραφή εργασιών

Σε περίπτωση που ο χρήστης επιθυμεί την διαγραφή κάποιας εργασίας που υπέβαλλε, γίνεται χρήση της εντολής:

qdel (job-ID)

Η επιλογή **-f** για την **qdel** οδηγεί σε forced deletion της εργασίας και χρησιμοποιείται στην περίπτωση που η απλή **qdel** δεν λειτούργησε. Σε περίπτωση που ο SGE δεν ανταποκρίνεται, είναι αναγκαία η επικοινωνία με τον διαχειριστή.

5 Οργάνωση αρχείων χρήστη

Η οργάνωση των αρχείων σε ξεχωριστά directories βάσει τύπου είναι αναγκαία για τη διευκόλυνση των χρηστών και την αντιμετώπιση τυχόν προβλημάτων κατά την υποβολή εργασιών στον SGE. Προτείνεται ο διαχωρισμός των αρχείων σε 3 directories:

1. **source:** Η τοποθεσία των αρχείων πηγαίου κώδικα του χρήστη.
2. **bin:** Η τοποθεσία των εκτελέσιμων που προκύπτουν από αρχεία του προηγούμενου directory. Κάθε shell script που υποβάλλεται στον SGE είτε έχει το full path name ενός εκτελέσιμου που βρίσκεται εδώ, είτε απλά το όνομα του εκτελέσιμου και την επιλογή **-cwd** στο preamble του script. Στη δεύτερη περίπτωση ο χρήστης θα πρέπει να βρίσκεται στην τοποθεσία **bin** όταν υποβάλλει το script.
3. **data:** Η τοποθεσία των shell scripts μαζί με τα output files και τα error logs. Τα ονόματα των αρχείων αυτών στη γενική μορφή είναι
 - (job-name).qsub
 - (job-name).o(job-ID)
 - (job-name).e(job-ID)

Για παράδειγμα, αν ο χρήστης υπέβαλλε το script **sgc-job.qsub** με job-ID 19, στο φάκελο **data** επιστρέφονται τα αρχεία **sgc-job.o19** και **sgc-job.e19**.

Ο φάκελος **data** είναι αυτός με τις μεγαλύτερες απαιτήσεις σε οργάνωση, καθώς εκεί αποθηκεύονται shell scripts τα οποία μπορεί να αφορούν το ίδιο εκτελέσιμο. Επιπλέον το κάθε script μπορεί να έχει διαφορετικά options στο preamble, διαφορετικά input arguments και διαφορετικά job names, ενώ χρησιμοποιείται το ίδιο εκτελέσιμο. Είναι αναγκαίο λοιπόν ο χρήστης να μπορεί να ξεχωρίσει για κάθε script, ποιά είναι τα input arguments που αντιστοιχούν στο output file που έλαβε. Μια ενδεδειγμένη τακτική για αυτό είναι η προσαρμογή του κώδικα, ώστε κατά την εκτέλεση να εκτυπώνει και όσα έλαβε ως input. Ένα παράδειγμα-εφαρμογή των παραπάνω είναι το εξής:

Ο χρήστης user15 έχει τα πηγαία αρχεία main1.c και main2.cpp από τα οποία προέκυψαν τα εκτελέσιμα exec1 και exec2 αντίστοιχα. Στο φάκελο data υπάρχουν όμως τα εξής scripts (με τα αντίστοιχα output - error logs):

- Το script exec1.args1.qsub, όπου χρησιμοποιείται το εκτελέσιμο exec1 με το argument set 1 (preamble options, input arguments).

- Το script `exec1.args2.qsub`, όπου χρησιμοποιείται το εκτελέσιμο `exec1` με το argument set 2.
- Το script `exec2.args1.qsub`, όπου χρησιμοποιείται το εκτελέσιμο `exec2` με το argument set 1.

Μια καλή τακτική για τα argument sets είναι να κρατούνται σε ξεχωριστά αρχεία κειμένου. Αυτή η διαδικασία οργάνωσης μπορεί να γίνει εύκολα αντιληπτή στο σχήμα που ακολουθεί:

6 Μεταγλώττιση και εκτέλεση παράλληλου κώδικα

Αρχικά προϋποθέτουμε ότι ο χρήστης είναι εξοικειωμένος με τον vim editor, προκειμένου να γράψει τον προς εκτέλεση κώδικα. Εναλλακτικά, μπορεί να στείλει τα αρχεία μέσω sftp στον Πυθαγόρα. Την παρούσα στιγμή χρησιμοποιείται η βιβλιοθήκη openmpi για παράλληλες εφαρμογές.

6.1 Μεταγλώττιση

Η μεταγλώττιση γίνεται με την εντολή **mpicc** (MPI C compiler) ή **mpic++** (MPI C++ compiler) για C και C++ αντίστοιχα. Οι εντολές αυτές έχουν την ίδια συμπεριφορά με τους απλούς C/C++ compilers.

Περισσότερες πληροφορίες σχετικά με τη χρήση των εντολών παρέχονται μέσω της **man mpicc**.

6.2 Εκτέλεση

Η εκτέλεση γίνεται με την εντολή **mpirun** ακολουθούμενη από το εκτελέσιμο που παράχθηκε στο προηγούμενο βήμα. Εδώ πρέπει να τονιστεί ότι η εντολή αυτή ΔΕΝ χρησιμοποιείται απευθείας αλλά εισάγεται στο **qsub script** μαζί με το **-pe orte** option ακολουθούμενο από το πλήθος των παράλληλων διεργασιών που επιθυμεί ο χρήστης (βλ. προηγούμενα παραδείγματα).

Αν ο χρήστης δεν προβεί στην παραπάνω ενέργεια, τότε ο κώδικας τρέχει τοπικά επιβαρύνοντας μόνο τον frontend αντί να διαμοιραστεί σε όλο το cluster μέσω του SGE.

6.3 Ένα απλό παράδειγμα

Έστω **parallel.c** το αρχείο πηγαίου κώδικα του χρήστη. Η μεταγλώττιση γίνεται ως εξής:

```
mpicc -o parallel_code parallel.c
```

Στη συνέχεια ο χρήστης συντάσσει ένα qsub script σαν το παρακάτω:

```
#!/bin/bash
#
#$ -pe orte 13
#$ -cwd
```

```
mpirun parallel_code 3 5
```

όπου το 3 και 5 είναι παράμετροι του προγράμματος. Έχοντας ετοιμάσει το script (έστω ότι ονομάστηκε mytest.qsub), δίνει την εντολή:

```
qsub mytest.qsub
```

Εφόσον το αρχείο υποβλήθηκε επιτυχώς, ο χρήστης μπορεί να παρακολουθεί την διαδικασία εκτέλεσης με την **qstat**. Υπόψιν ότι αν η **qstat** δεν εμφανίζει κάτι στην οθόνη τότε είτε το πρόγραμμα εκτελέστηκε επιτυχώς ή υπήρξε πρόβλημα κατά την υποβολή του στον scheduler.