COMPUTACIÓ D'ALTES PRESTACIONS – PAC 1

Semestre Setembre 2017

- Presentació i objectius
- Enunciat de la pràctica
- Criteris d'avaluació
- Format de lliurament
- Data de lliurament

Presentació i objectius

Objectius

Aquesta pràctica té com a objectius introduir l'entorn de treball. A part de la logística per accedir al sistema, també es posarà en pràctica l'ús de sistemes de cues i l'execució sistemàtica de proves per a realitzar estudis paramètrics i/o de rendiment.

Presentació de la pràctica

Cal lliurar un document amb les respostes a les preguntes formulades, scripts/gràfics que es demanen i els comentaris que considereu..

Restriccions de l'entorn

Tot i que el desenvolupament de la pràctica és molt més interessant utilitzant dotzenes de computadors, s'assumeix que inicialment tindreu accés a un nombre força reduït de computadors amb diversos nuclis per node.

Material per a la realització de la pràctica

En els servidors de la UOC teniu el programari necessari per realitzar les execucions requerides. De totes maneres, és possible que tingueu que instal·lar software client per tal d'accedir als sistemes de la UOC.

Enunciat de la pràctica

Utilitzarem un entorn tipus clúster de computació basat en GNU/Linux per a la realització de les pràctiques. A l'aula us proporcionem un manual d'ús bàsic del sistema de cues, si us plau utilitzeu-ho com a primera referència. També podeu trobar més informació a la web, per exemple al següent link podeu trobar més detalls del sistema de cues SGE (Sun Grid Engine) que hi ha instal·lat al clúster de la UOC:

http://gridscheduler.sourceforge.net/htmlman/htmlman1/qsub.html

És important que sol·liciteu un compte d'usuari al clúster enviant un email a l'adreça de correu electrònic acasys@uoc.edu, el més aviat possible.

Al clúster us hi connectareu via connexió segura **ssh**. Si treballeu en entorns GNU/Linux o Mac OSX podreu establir la connexió ssh directament a través d'un terminal, en canvi, en el cas de Windows, haureu d'utilitzar alguna utilitat que us proporcioni aquest servei, com per exemple Putty. Us podeu descarregar Putty per exemple des del següent link:

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

També podeu trobar tutorials online com per exemple (no especial preferència, n'hi ha molts):

https://www.youtube.com/watch?v=p-cN0y1WFdA

L'accés a l'entorn de treball és mitjançant:

```
ssh nom_usuari@eimtarqso.uoc.edu
```

El nom_usuari es proporcionarà un cop sol·liciteu el vostre compte del sisema.

Si heu de transferir arxius també seria útil un client scp/sftp com winscp:

http://winscp.net/eng/docs/lang:es

En el cas de que observeu un funcionament incorrecte al cluster, o algun problema amb la vostre compte ho podreu reportar a l'anterior adreça de correu de suport.

Descripció del sistema:

Eimtarqso és el node principal d'un clúster de 10 nodes de còmput. Eimtarqso es l'únic node que heu d'accedir directament i a partir del qual podreu executar tasques a la resta dels nodes mitjançant el sistema de cues de SGE. **És molt important que respectiu aquesta directiva ja que els recursos s'han de compartir amb la resta de companys**.

Els detalls del sistema els podem observar a través de la següent comanda (el resultat es veu a continuació):

```
[ivan@eimtarqso ~]$ qstat -f
queuename                      qtype resv/used/tot. load_avg arch          states
---------------------------------------------------------------------------------
all.q@compute-0-0.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-1.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-2.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-3.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-4.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-5.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-6.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-7.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-8.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-9.local        BIP   0/0/4          0.00     linux-x64
```

La sortida mostra 10 nodes de còmput (compute-0-0.local ... compute-0.9.local), cadascun dels nodes amb 4 cores totals (en la columna resv/used/**tot**) i una càrrega mitja total de 0.00.

Nota: tot i que el sistema us ho permeti, no us connecteu a cap node de còmput de forma directa en cap cas.

Execució tradicional vs. en clúster:

Per exemplificar l'ús del sistema de cues (SGE) farem servir dues comandes de sistema que teniu per defecte al vostre path:

- `hostname`, proporciona el nom del servidor/node
- `sleep`, fa que el procés entri en espera durant el nombre de segons indicat

Típicament la forma de consultar el nom del servidor és el següent:

```
[ivan@eimtarqso ~]$ hostname
eimtarqso.uoc.edu
```

Podem executar aquesta comanda en altres nodes del clúster mitjançant un script de SGE. A continuació us proporcionem un exemple bàsic:

```
[ivan@eimtarqso ~]$ cat hostname_example.sge
#!/bin/bash
#$ -cwd
#$ -S /bin/bash
#$ -N hostname_jobname
#$ -o hostname.out
#$ -e hostname.err
hostname
```

La última línia de l'script fa la crida a la comanda que volem executar. La següent línia seria equivalent:

`echo ``hostname``` (noteu que les cometes son cap a l'esquerra...)

Per executar l'script haurem de fer servir qsub. Un exemple es mostra a continuació:

```
[ivan@eimtarqso ~]$ qsub hostname_example.sge
Your job 263534 ("hostname_jobname") has been submitted
```

PREGUNTES:

1. Quin és el resultat de la comanda hostname? On es pot trobar? Comproveu si s'han creat fitxers nous.

2. Executeu l'script d'exemple varies vegades (per exemple 3-4). Quin són els resultats obtinguts? És sempre el mateix? Per què?

3. Si executeu l'script d'exemple (el mateix fitxer) múltiples vegades a la vegada, teniu problemes per conservar el resultat de la sortida de cadascuna de les execucions? Com ho podeu fer per guardar el resultat de totes les execucions?

Un cop heu enviat un treball (job) a la cua mitjançant qsub, podeu cancel·lar-lo si és necessari (per exemple, si us heu adonat que hi ha un error o voleu fer canvis). Això es pot fer mitjançant la comanda SGE **qdel**.

Per a demostrar la seva utilització farem servir un altre script de prova que faci un sleep de varis segons i ens permeti veure que està a la cua, en execució i finalment cancel·lar-ho. L'script proposat és el següent:

```
[ivan@eimtarqso ~]$ cat sleep.sge
#!/bin/bash
#$ -cwd
#$ -S /bin/bash
#$ -N sleep_job
#$ -o sleep.out

sleep 100
```

A continuació podeu veure la seqüència de comandes i resultats per il·lustrar el funcionament de qdel (hi ha comentaris intercalats en vermell):

```
[ivan@eimtarqso ~]$ qsub sleep.sge
Your job 121598 ("sleep_job") has been submitted
[enviem el treball a la cua]

[ivan@eimtarqso ~]$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
 121598 0.00000 sleep_job  ivan         qw    09/21/2015 01:37:34                                    1
[amb qstat veiem que el treball enviat està a la cua esperant per ser assignat a un node i ser executat]


[ivan@eimtarqso ~]$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
 121598 0.55500 sleep_job  ivan         r     09/21/2015 01:37:37 all.q@compute-0-2.local            1
[al cap d'uns segons veiem que el treball està en estat "r"-running, per tant en execució]

[ivan@eimtarqso ~]$ qstat -f
queuename                      qtype resv/used/tot. load_avg arch          states
---------------------------------------------------------------------------------
all.q@compute-0-0.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-1.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-2.local        BIP   0/1/4          0.00     linux-x64
 121598 0.55500 sleep_job  ivan         r     09/21/2015 01:37:37     1
---------------------------------------------------------------------------------
all.q@compute-0-3.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-4.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-5.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-6.local        BIP   0/0/4          0.00     linux-x64
[amb qstat –f veiem que el nostre treball està executant-se al node compute-0-2.local]

[ivan@eimtarqso ~]$ qdel 121598
ivan has registered the job 121598 for deletion
[amb qdel i l'identificar del treball demanem cancelar el treball]

[ivan@eimtarqso ~]$ qstat
[un cop cancel.lat, el treball ja no apareix en la llista dels nostre treballs]

[ivan@eimtarqso ~]$ qstat -f
queuename                      qtype resv/used/tot. load_avg arch          states
---------------------------------------------------------------------------------
all.q@compute-0-0.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-1.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-2.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-3.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-4.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-5.local        BIP   0/0/4          0.00     linux-x64
---------------------------------------------------------------------------------
all.q@compute-0-6.local        BIP   0/0/4          0.00     linux-x64
[ivan@eimtarqso ~]$
```

Tasques (jobs) grans vs. múltiples tasques petites:

A vegades hem de fer estudis paramètrics o estadístics que impliquen l'execució d'un programa múltiples vegades. En aquest escenari tenim dues opcions bàsiques:

1. Utilitzar un script SGE que realitzi les diferents execucions com a part d'un únic treball
2. Enviar múltiples treballs per a cadascuna de les execucions.

En aquest curs esperem que utilitzeu la modalitat 2 ja que fa que l'accés al recursos sigui més justa entre tots els estudiants. Per exemple, si uns pocs estudiants realitzen execucions de gran durada en tots els recursos poden fer que la resta hagin d'esperar durant hores o fins i tot dies.

Avaluació de rendiment: estudi paramètric d'un codi d'exemple:

A continuació es presenten els passos necessaris per compilar i executar un programa per a la multiplicació de matrius (sense optimitzacions) que us proporcionem en aquesta pràctica:

- Compilar el programa `mm.c` amb:
  Sense optimitzacions: `gcc -O0 mm.c -o mm`
  Amb optimitzacions (compilador): `gcc -O3 mm.c -o mm`

- Executar el programa mitjançant el sistema de cues SGE. Noteu que el tamany de la matriu se li passa al programa mitjançant un paràmetre.
  IMPORTANT: no feu execucions directament a l'intèrpret de comandes ja que podeu saturar el node d'accés i les mesures que prengueu poden no ser vàlides per haver de compartir els recursos amb altres usuaris al mateix temps.

PREGUNTES:

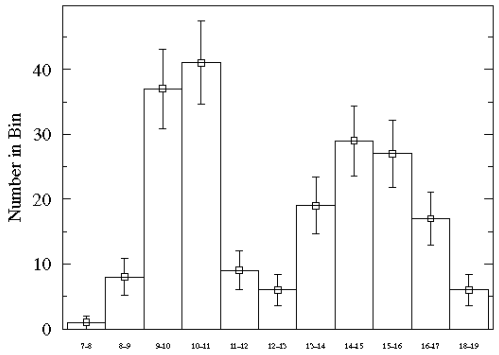
4. Com ho heu fet per copiar et fitxer `mm.c` des del vostre PC fins al servidor `eimtarqso.uoc.edu`?

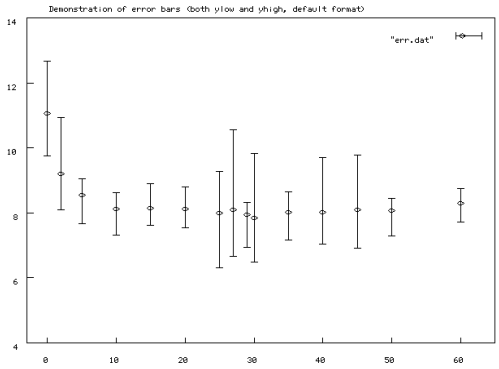
5. Proporcioneu el script SGE que heu fet servir per executar el programa

A continuació es demana fer un estudi paramètric de la multiplicació de matrius. El que es demana és que proporcioneu el temps d'execució del programa segons mida del problema (mida de les matrius, en aquest cas quadrades).

Quan es realitzen estudis de rendiment pot haver-hi certa variabilitat entre execucions degut a qüestions relacionades amb la contenció de memòria per la compartició de recursos amb altres processos del sistema, per efectes de cache, aleatorietat, etc.

Per tal de mitigar la variabilitat en les execucions, cal fer un estudi de caràcter estadístic on es realitzen varies execucions i es proporcionen mètriques estadístiques com ara la mitjana aritmètica, i els quartils. Podeu veure un parell d'exemple de gràfics utilitzat gnuplot (eina per fer gràfiques de codi obert amb "error bars", però podeu utilitzar la eina que més us agradi.





Com que haureu de realitzar vàries execucions (per exemple 4) per a cada configuració, és convenient que realitzeu aquestes execucions de forma sistemàtica i programada. Us proposem utilitzar scripts per realitzar aquesta tasca (per exemple shell scripts que cridin a qsub per a cadascuna de les configuracions).

Volem realitzar els següents estudis:

- Temps d'execució per a diferents mides de problema sense optimitzacions (-O0) i amb optimitzacions (-O3)
  - Feu proves amb mides (paràmetre d'entrada): 100, 500, 1000 i 1500.

PREGUNTES:

6. Com ho heu fet per obtenir el temps d'execució de les diverses proves?

7. Proporcioneu els scripts (programa o metodologia) fets servir per a realitzar les proves de forma sistemàtica

8. Proporcioneu un gràfic de temps d'execució amb diferents mides, amb i sense optimitzacions, i compareu els resultats

Sistemes de cues i planificació: PREGUNTES:

9. Enumereu quatre sistemes de cues per a sistemes d'altes prestacions (clústers). Proporcioneu un script d'exemple per a cadascun d'ells i comenteu breument les principals diferències i similituds.

10. Com ho fa el clúster per a poder accedir as fitxers del vostre $HOME des de qualsevol node de còmput del clúster? Quin sistema de fitxer penseu que fa servir el clúster de la UOC?

11. Proporcioneu la planificació dels treballs indicats a la següent taula (assumint un sistema amb 10 CPUs) utilitzant les polítiques:

(a) FCFS

(b) EASY-backfilling (de tal forma que backfill no permeti endarrerir un treball que estava davant en la cua)

Job #	Arrival Time	Runtime	#CPUs
1	1	8	8
2	2	2	2
3	2	8	1
4	3	4	4
5	6	2	1
6	8	3	9
7	8	4	1
8	9	6	4
9	9	4	4
10	10	2	10
11	11	4	2
12	12	4	2
13	12	4	1
14	16	3	1
15	16	5	1
16	20	2	5
17	24	3	8

També proporcioneu:

(c) Utilització dels recursos (%)

(d) Average queue time

(e) Average slowdown

La utilització dels recursos es defineix com:

Utilització = recursos utilitzats (en el nostre cas CPUs)/total dels recursos disponibles

Average queue time és la mitja del temps que els treballs estan a la cua abans de ser planificats (més info: http://web.mit.edu/sgraves/www/papers/Little%27s%20Law-Published.pdf)

Slowdown (SLD) es defineix com:

SLD = (waiting time + runtime) / runtime

Un exemple del que es demana a escala més petita (6 CPUs) i utilitzant la política FCFS es mostra a continuació:

Job #	Arrival Time	Runtime	#CPUs
1	2	2	4
2	2	1	1
3	3	1	6
4	6	2	4

Planificació:

time / CPU#	1	2	3	4	5	6	7
CPU 1		J1	J1	J3		J4	J4
CPU 2		J1	J1	J3		J4	J4
CPU 3		J1	J1	J3		J4	J4
CPU 4		J1	J1	J3		J4	J4
CPU 5		J2		J3			
CPU 6				J3			

Util = 54.76%

AVG queue time = 0.25s

AVG SLD = 1.25

Criteris d'avaluació

Es valorarà especialment la utilització dels scripts per a l'execució sistemàtica i que proporcioneu gràfics amb valors estadístics.

Format de lliurament

Es crearà un fitxer en format PDF amb tota la informació.

Si els scripts són llarg per afegir-los al document de l'entrega (no s'esperen que siguin massa sofisticats), podeu adjuntar-los amb la comanda següent:

```
$ tar cvf tot.tar fitxer1 fitxer2 ...
```

es crearà el fitxer "tot.tar" on s'hauran emmagatzemat els fitxers "fitxer1", fitxer2" i ...

Per llistar la informació d'un fitxer `tar` es pot utilitzar la comanda següent:

```
$ tar tvf tot.tar
```

Per extraure la informació d'un fitxer `tar` es pot utilitzar:

```
$ tar xvf tot.tar
```

El nom del fitxer tindrà el format següent: "Cognom1Cognom2PAC2.pdf" (o *.tar). Els cognoms s'escriuran sense accents. Per exemple, l'estudiant Marta Vallès i Marfany utilitzarà el nom de fitxer següent: `VallesMarfanyPAC2.pdf (o *.tar)`

Data de lliurament

Diumenge 15 d'Octubre de 2017.