



ENEA-GRID: LSF

Monitoraggio LSF lato utente

Ing. Fiorenzo Ambrosino, PhD
UTICT-HPC C.R. Portici (NA)

**Corso ENEA-GRID –
Brindisi 25 giugno 2013**

- Si daranno alcune informazioni sul monitoraggio del servizio e delle caratteristiche di LSF soltanto dal lato utente e non amministrazione.
- Verranno illustrate solo alcune situazioni di problemi legati a malfunzionamenti del servizio (i casi reali possono essere moltissimi)
- Verranno illustrati alcuni comandi essenziali

Monitoraggio LSF: utenti

- Se LSF sta girando ed i comandi di interrogazione rispondono, non significa che non ci siano problemi.
- Ci sono molte situazioni “invisibili” di allarme che appaiono solo agli utenti:
 - File system non disponibile
 - Infiniband fuori uso
 - Problemi di sistema operativo
 - Errori sul caricamento del profile

Freecresco: verifica cpu disponibili

- E' stata sviluppata l'utility freecresco che permette di controllare gli slot liberi per sapere quanti processori si possono usare e per quanto tempo, al fine di non rimanere in pending.

Freecresco: verifica cpu disponibili

```
esempi@cresco1-fg4.portici.enea.it>freecresco
```

Queue name	Specific resource option	Available slots throughout queue runtime limit	Available slots for backfill runs	Backfill time window (minutes)
cresco_128h2	-	144(136,144,8)		
cresco_128h8	-	144(136,144,8)		
cresco_128h24	-	-	144(136,144,8)	1270
cresco_16h2	(-R type==low_parallel)	81(24,80,8)		
cresco_16h2	(-R type==high_parallel)	144(24,128,8)		
cresco_16h8	(-R type==low_parallel)	81(24,80,8)		
cresco_16h8	(-R type==high_parallel)	144(24,128,8)		
cresco_16h24	(-R type==low_parallel)	81(24,80,8)		
cresco_16h24	(-R type==high_parallel)	-	144(24,128,8)	1270
cresco_8h144	-	81(8,80,8)		
cresco_h2	-	81(2,16,1)		
cresco_h8	-	81(2,16,1)		
cresco_h24	-	81(2,16,1)		
cresco3_72h24	-	1176(72,1056,24)		
cresco3_h144	-	287(1,71,1)		
cresco_serh24	-	91 serial		
cresco_serh144	-	18 serial		

See "freecresco -h" for documentation and known bugs.

Lancio di un job parallelo

```
/* hello.c
 *
 * Simple "Hello World" program in MPI.
 * NOTA: Programma di test preso in prestito dal sito del CERN
 */

#include "mpi.h"
#include <stdio.h>
int main(int argc, char *argv[]) {
int numprocs; /* Number of processors */
int procnum; /* Processor number */
/* Initialize MPI */
MPI_Init(&argc, &argv);
/* Find this processor number */
MPI_Comm_rank(MPI_COMM_WORLD, &procnum);
/* Find the number of processors */
MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
printf ("Hello world! from processor %d out of %d\n", procnum, numprocs);
/* Shut down MPI */
MPI_Finalize();
return 0;
}
```

Lancio di un job parallelo: OpenMPI

- Compilazione:
 - mpicc hello.c
- Per lanciare il job, crearsi un semplice script di lancio (wrapper):

```
#!/bin/sh
PROCS=`cat $LSB_DJOB_HOSTFILE | wc -l`
mpirun -n $PROCS --mca pls_rsh_agent "blaunch.sh" -hostfile $LSB_DJOB_HOSTFILE
./a.out
```

- Lo script deve
 - Contare i processori assegnati da LSF
 - Sostituire “ssh” con “blaunch.sh”
- Lancio del programma provando freecresco:
`bsub -n 64 -q cresco_16h2-W 30 -o test.%J ./lancio.sh`

Se LSF non risponde

- Nel caso in cui il comando bsub rimanga “appeso” senza venir eseguito correttamente, significa che il demone mbatchd non risponde correttamente
 - Può essere in atto un problema di rete o filesystem che rallenta le operazioni di mbatchd
 - mbatchd può non essere attivo e necessita di un restart
 - mbatchd è diventato un processo zombi. In questo caso l'unico modo per ripristinare il cluster è il reboot del master; lo spegnimento del master lim potrebbe non essere sufficiente.
- In ogni caso aprire un ticket al sistema di GRID ticketing: <https://gridticket.enea.it/>

Se i job rimangono in pending

Se il job è lanciato correttamente ed è stato lanciato su un numero di slot coerente all'output di freecresco, esso deve andare in run entro pochi secondi.

Se ciò non capitasse (job in stato PEND):

- **IMPORTANTE:** Controllare l'opzione -R di bsub. Si suggerisce di usare sempre le code cresco ed evitare di specificare -R se non strettamente necessario.
- `bjobs -p` mostra i motivi per i quali LSF lascia il job in pending in modo tale da evidenziare la risorsa mancante (ad esempio memoria, licenze ecc)

Comandi per monitoring

Isload

Monitoring risorse dinamiche per host

Ishosts

Monitoring le risorse statiche per host

bhosts

Mostra il numero di job sui vari host

Isxload

Unisce le informazioni degli altri comandi (necessita di righe da 132 caratteri)

Grazie per la cortese attenzione!

Ing. Fiorenzo Ambrosino, Ph.D.

ENEA – UTICT-HPC

<http://www.afs.enea.it/ambrosino>

fiorenzo.ambrosino@enea.it