Checkpoint e restore di job LSF

Workshop Congiunto INFN CCR e GRID

Acireale, 17-21 Maggio 2010

INFN - Sezione di Trieste - C. Strizzolo - L. Strizzolo

La farm di calcolo INFN-Trieste

- Farm di calcolo “condivisa”
- Possibilità di utilizzo con sottomissione di job in locale e via grid
- Condivisione completa di risorse di calcolo finanziate dalla sezione, dagli esperimenti locali e da grid, senza assegnazione esclusiva delle risorse di calcolo
- Utilizzo prioritario delle proprie risorse di calcolo da parte degli esperimenti locali, condivisione delle risorse in caso di non utilizzo
- Utilizzo di tutte le risorse di calcolo disponibili in caso di necessità evitando situazioni di scarso utilizzo delle risorse

Perchè un checkpoint/restore

- Necessità di sospendere l'esecuzione di job sottomessi su alcune code:
  - coda locale non privilegiata per collaborazione con enti diversi
  - coda per l'esecuzione di job in determinate fasce orarie
  - code non di esperimento su nodi di esperimento
- Possibilità di sospendere l'esecuzione di job su nodi che esauriscono la RAM

LSF – Sospensione e checkpoint

- La sospensione dei job con il sistema di preemption di LSF o mediante l'utilizzo del comando bstop di LSF libera la CPU ma non libera la RAM
- Il checkpoint fornito da LSF non è direttamente utilizzabile dai job degli utenti (ricompilazione)

Berkeley Lab Checkpoint/Restart (BLCR) 1/2

- https://ftg.lbl.gov/CheckpointRestart/CheckpointRestart.shtml
- Richiede kernel 2.6.x (x86 e x86_64)
- Lavora a livello di kernel (moduli), permettendo il restore dei PID, processi e sottoprocessi, pipe di connessione fra i processi
- Funziona con programmi singlethread e multithread (pthreads)

Berkeley Lab Checkpoint/ Restart (BLCR) 2/2

- Non esegue il checkpoint/restore dei socket aperti e degli oggetti SysV IPC
- Non esegue il restart di eventuali processi “zombie” presenti al momento del chekpoint
- Esiste un post su checkpoint@lbl.gov list per configurare LSF per l’utilizzo di BLCR, ma non è sufficiente...

Configurazione di LSF - bqueues

```
JOB_STARTER  =  /usr/bin/cr_run
   /lsf/scripts/jobstarter_ckpt_lsf_lcg.sh '%USRCMD`

CHKPNT       = /gpfs/common/lsf-ckpnt

POST_EXEC    = /bin/rm -Rf $LSB_CHKPNT_DIR

JOB_CONTROLS  = SUSPEND[$LSF_SERVERDIR/echkpnt.blcr -k]
   RESUME[$LSF_SERVERDIR/erestart.blcr]
```

Configurazione di GRID

Grazie a F. Prelz

Sul CE viene modificato il file:

/opt/globus/lib/perl/Globus/GRAM/submit-helper.pl

Se il job di LSF è stato lanciato con il jobstarter indicato viene commentata la riga:

trap 'fatal_error "Job has been terminated (got SIGTERM)" "OSB"' TERM

nel file in cui il CE incapsula l'effettivo job da eseguire.

Il funzionamento Jobstarter

LSF esegue il jobstarter caricando la libreria di checkpoint con “cr_run”

```
JOB_STARTER=/usr/bin/cr_run /lsf/scripts/jobstarter_ckpt_lsf_lcg.sh '%USRCMD'
```

Il jobstarter crea e lancia una shell script (chkpnt_script.sh) con cui viene eseguito il job inviato dall’utente (%USRCMD) e rimane in attesa della conclusione della script.

Il funzionamento 2/6
Sospensione del job

Un’eventuale richiesta di sospensione del job (comando bstop di LSF) viene eseguita con il comando echkpnt.blcr

```
JOB_CONTROLS  = SUSPEND[$LSF_SERVERDIR/echkpnt.blcr -k]
  RESUME[$LSF_SERVERDIR/erestart.blcr]
```

che esegue il checkpoint su disco (viene creato un file) della shell script e dei sottoprocessi (e non del jobstarter) con il comando cr_checkpoint ed invia SIGTERM alla shell script e a tutti i sottoprocessi.

Il funzionamento 3/6
Sospensione del job – fallimento

Prima di effettuare l'operazione di checkpoint il comando echkpnt.blcr controlla che per i sottoprocessi della shell script non risultino aperti dei socket. Nel caso in cui ci siano socket aperti l'operazione di checkpoint non viene eseguita.

In questo caso, e nel caso in cui l'operazione di checkpoint fallisca per qualche motivo (exit code ≠ 0), il job viene sospeso inviando SIGSTOP ai processi.

Il funzionamento 4/6
Controllo dello stato dei processi

E’ necessario eseguire il checkpoint della shell script e non del jobstarter perché l’operazione di checkpoint invia un SIGTERM ai processi ed LSF interpreterebbe la fine del jobstarter come la fine del job.

Il funzionamento 5/6
Ripresa del job

La richiesta di ripresa del job (comando bresume di LSF) viene eseguita con il comando erestart.blcr

```
JOB_CONTROLS  = SUSPEND[$LSF_SERVERDIR/echkpnt.blcr -k]
  RESUME[$LSF_SERVERDIR/erestart.blcr]
```

che esegue il ripristino dei processi, con il comando cr_restart, a partire dal file di checkpoint creato.

Il funzionamento 6/6
Conclusione del job

Una volta terminata l'esecuzione del job inviato dall'utente (%USRCMD) la shell script termina segnalando al jobstarter l'avvenuta conclusione del job in modo da portare a termine anche l'esecuzione del jobstarter, che corrisponde alla conclusione del job LSF.

I risultati

Utenti locali	Esito del checkpoint/restore
alice, compass, pamela, cms	OK
utenti esterni	OK
GRID VO	
atlas, cms, euindia, glast, compchem	OK
biomed	NO – socket aperti
esr, theophys	NO – checkpoint fallito (code 52)
lhcb	NO – problemi in restore (processi con PPID=1)