An Early Experience on Job Checkpoint/Restart – Working with SGI Irix OS and the Portable Batch System (PBS)

Sherry Chang

schang@nas.nasa.gov
Scientific Consultant
NASA Advanced Supercomputing Division
NASA Ames Research Center
Moffett Field, CA

Helping Our Users

- Scientific Consultants at NAS help users to run their jobs successfully with little or no impact on other users

- This presentation originated from helping with a user's case

User plans to run a job that will need ~40 days to complete on O2K

  - Checkpoint/Restart within source code is limited
  - Job does not need many processors
  - Job does not warrant use of dedicated time on 64, 256, 512 cpus systems
  - Our batch queues on O2K allow maximum of 8 hours walltime

Outline

- Why, How, and Who ?
- Examples : a Gaussian job and an MPI job
- Introduction to SGI's cpr
- Four Methods for checkpoint/restart
  - using cpr interactively
  - using cpr within PBS script
  - using qhold and qrls of PBS
  - using qsub –c of PBS for automatic checkpointing periodically
- Success and Failures
- Future testing and wish list

Why Do Checkpointing?

- Halt and restart resource-intensive codes that take a long time to run
  - Prevent job loss if system fails
- Improve a system's load balancing and scheduling
- Replace hardware, maintenance

How to Do Checkpoint/Restart ?

- User code has its own checkpoint capability

Example:
many CFD codes
certain gaussian jobs

- OS has built-in checkpoint/restart utility

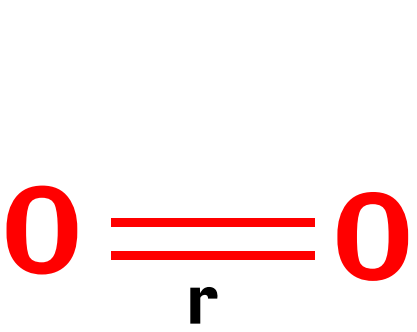
Example:
The Cray Unicos OS – chkpnt and restart
The SGI Irix OS – cpr implemented in 6.n releases

Batch systems NQE, LSF, PBS support checkpoint/restart

Who Can Checkpoint and Restart ?

- owner of process(es)
- superuser

Sample Gaussian Job



To run a Gaussian Job

% g98 o2.com o2.out &

Gaussian Script : o2.com

```
% nproc=2
% chk=o2.chk
#p CCSD/6-31g* OPT

O2 Geometry Optimization

0 1
O
O 1 r

r 1.500
```

Sample MPI Job

- Program pi.f : calculate the value of π

  % mpirun –np 3 ./pi > pi.out &

- Use –miser or –cpr option to allow checkpoint/restart

  % mpirun –miser –np 3 ./pi > pi.out &

SGI's CPR Commands

The cpr command provides a command-line interface for

- checkpoint

  % cpr -c statefile -p id:HID -k

  HID for process hierarchy (tree) rooted at that PID

- find information of an existing checkpoint statefile

  % cpr -i statefile

- restart

  % cpr -r statefile

- delete checkpoint statefile

  % cpr -D statefile

Using CPR Interactively

- Start a Gaussian Job

% g98 o2.com o2.out &
[1] 19432 (parent process ID)

% ps

PID	TTY	TIME	CMD	
19431	ttyq2	0:01	l101.exe	← child process
19432	ttyq2	0:00	g98	← parent process
19435	ttyq2	0:02	l101.exe	← child process

- Do the first checkpoint

% cpr -c chk1 -p 19432:HID -k
Checkpointing id 19432 (type HID) to directory chk1
Checkpoint done

Created in working directory and -rwx- by root only

Caveat: Multiple-processor Gaussian jobs do not automatically clear its 'shared memory segments' when the job is checkpointed.

Using CPR Interactively – continued

- Restart

```
% cpr –r chk1  &
[1] 19458
% Restarting processes from directory chk1
Process restarted successfully.

[1]    Done                  cpr –r chk1

% ps
 PID TTY     TIME CMD
19431 ttyq2   0:27 l114.exe
19432 ttyq2   0:00 g98
```

A child process of g98 is restarted

Failure – 1

- checkpoint stalled using cpr interactively
  - for mpi jobs

```
% mpirun –miser –np 3 ./pi > pi.out &
[1] 3527372
% cpr –c chk1 –p 3527372:HID –k (&)
[2] 3539292
(no progress at all)
```

Production systems : checkpoint stalled most of times
Test-bed systems : successful

Using CPR within PBS

- PBS script for first checkpoint

```
# PBS -l ncpus=2
# PBS -l mem=50mw
# PBS -l walltime=1:00:00

setenv g98root /usr/local/pkg
setenv $g98root/g98/bsd/g98.login

cd $PBS_O_WORKDIR

g98 o2.com o2.out &
sleep 20
cpr -c chk1 -p `ps -u schang | grep g98 | awk '{print $1}'`:HID -k
```

Find the PID of the parent process

- PBS script for subsequent cpr

```
# PBS -l ncpus=2
# PBS -l mem=50mw
# PBS -l walltime=1:00:00

setenv g98root /usr/local/pkg
setenv $g98root/g98/bsd/g98.login

cd $PBS_O_WORKDIR

cpr -r chk1  &
sleep 60
cpr -c chk2 -p 3971074:HID -k
```

- Caveat : Restart will fail if PBS stdout/stderr not present
- Alternative : start job and do first checkpoint interactively

Failure – 2

- restart failed using cpr in PBS script

– both for mpi and gaussian jobs

– checkpoint/restart successful for a few cycles, restart failed in a later cycle

```
cpr –c chk1 –p:xxxx:HID –k &
cpr –r chk1
cpr –c chk2 –p:xxxx:HID –k &
cpr –r chk2
....
cpr –c chkn –p:xxxx:HID –k &
cpr –r chkn
```

} successful

} failed

Error Messages:

CPR Error: Failed to place mld 0 (Invalid argument)
CPR Error: Unexpected status EOF
CPR Error: Cleaning up the failed restart

Failure – 3

- restart failed from a checkpoint state which was once successfully restarted
  - both for mpi and gaussian jobs

```
cpr –c chk1 –p:xxxx:HID –k &
cpr –r chk1
cpr –c chk2 –p:xxxx:HID –k &
cpr –r chk2
....
cpr –c chkn –p:xxxx:HID –k &
cpr –r chkn
```

} successful

} failed

```
cpr –r chk2 &
```

} failed

Error Message :
same as previous page

Using qhold and qrls of PBS

```
% qsub o2.script
1121.evelyn.nas.nasa.gov

% qhold 1121
% ls -l /PBS/checkpoint
drwxr-xr-x   3 root    root       28 May  1 12:20  1121.evelyn.CK

% qstat -a
Job ID                  S
1121.evelyn             H

% qrls 1121
Job ID                  S
1121.evelyn             R

% qhold
% qrls
```

Create checkpoint directory in /PBS/... when qhold

Job held

Job restarted

Checkpoint directory updated for subsequent qhold

PBS script: o2.script

```
# PBS -l ncpus=2
# PBS -l mem=50mw
# PBS -l walltime=1:00:00

setenv g98root /usr/local/pkg
setenv $g98root/g98/bsd/g98.login

cd $PBS_O_WORKDIR

g98 o2.com o2.out
```

Failure – 4 (hopper)

Turing % qsub mpi.pbs
8128.fermi.nas.nasa.gov

Turing % qhold 8128 Job held, processes stopped, 8128.fermi..CK created

Turing % qrls 8128 Job restarted successfully, processes running

Turing % qhold 8128 Job held again, 8128.fermi..CK updated

Turing % qrls 8128 Restart failed

Turing % qstat –a

qstat says job is running.
But, "ps" or "top" shows no processes running

Job ID	S
8128.fermi	R

Turing % qdel 8128

Job can not be deleted by qdel
PBS needs serious clean–up

Job ID	S
8128.fermi	R

Failure –5 (t3)

t3% qsub mpi.pbs
148.t3.nas.nasa.gov

33 sec t3 % qhold 148 — Job held, processes stopped, 148.t3.nas..CK created

72 sec t3 % qrls 148 — Job restarted successfully, processes running

Job ran for ~40 seconds and then got killed

t3 % qstat –a

Job ID	S
148.t3	R

qstat says job is running.
But, “ps” or “top” shows no processes running

t3 % qdel 148

Job ID	S
148.t3	R

Job can not be deleted by qdel
PBS needs serious clean–up

Using qsub –c of PBS for Automatic Checkpointing Periodically

% qsub o2.script
or
% qsub –c c=3 o2.script
1120.evelyn.nas.nasa.gov

PBS script : o2.script

```
# PBS –l ncpus=2
# PBS –l mem=50mw
# PBS –l walltime=1:00:00
# PBS –c c=3

setenv g98root /usr/local/pkg
setenv $g98root/g98/bsd/g98.login

cd $PBS_O_WORKDIR

g98 o2.com o2.out
```

% ls –l /PBS/checkpoint

drwxr-xr-x 3 root root 28 May 1 12:05 1120.evelyn.CK

Checkpoint directory
Updated every 3 minutes

If PBS mom or system crashes :

PBS should automatically restart a job that has a checkpoint directory associated with it after the system is back

Future Testing and Wish List

Future Testing :

- A wide variety of user applications – OpenMP, pvm, mpi
- Large parallel jobs
- System-wide checkpoint/restart
- System crash simulation
- Efficiency

Ultimate Goal :

Make sure checkpoint/restart is reliable
in a real production environment

IRIX/CPR – A Popular Topic

Recent email-exchanges on this topic, sgi-tech@cug.org

Barry Sharp – Boeing

Paul White – CSC

Miroslaw Kupczyk – Poznan Supercomputing and Network Center

Torgny Faxen – National Supercomputing Center, Sweden

- **Irix OS**
- **NQE, LSF**
- **MPI, OpenMP, Gaussian – no pvm yet**
- **Irix vs Unicos**

SGI- supportfolio bug report provides limited information

would like to see more exchange on the details of the success and failure cases

Acknowledgement

- Ed Hook – SciCon, PBS expert
- Lorraine Freeman – sysadmin
- Bron Nelson – SGI on-site analyst
- Chuck Niggley – SciCon Group Lead
- NASA Advanced Supercomputing Division
- CUG