

How to checkpoint application in cluster!

Kernel Level Checkpointing for SGI Altix

+

TORQUE Resource Manager

1. Get the packages

- Kernel level checkpointer

<http://checkpointing.psnc.pl/SGIGrid/>

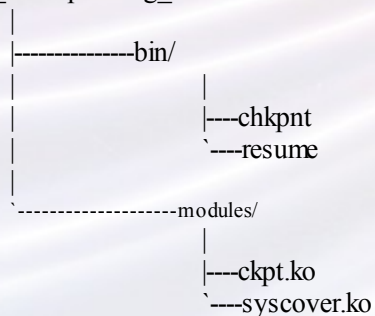
- Resource Manager

<http://www.clusterresources.com/products/torque>

2. Install checkpointer

- Go to the directory where you have the downloaded package PACKAGE-NAME.tar.gz
- Unpack and extract files from the package:
> gzip -d PACKAGE-NAME.tar.gz
> tar -xvf PACKAGE-NAME.tar
- The files are extracted to the directory named like a package PACKAGE-NAME
- The content of the directory looks as follows:

SGI_Altix_checkpointing_0.71/



- Go to the directory PACKAGE-NAME
- Install modules in the running kernel (obey the order of typing the commands):
> insmod modules/syscover.ko
> insmod modules/ckpt.ko
- To do the checkpoint or restart a program, use user level tools provided with the package. It is good idea to copy the tools to the directory that is enumerated by the PATH environment variable. Usage of these tools is described in the next section.

2. Torque Installation

- **Download the TORQUE distribution file from:**
<http://clusterresources.com/downloads/torque>
- **Extract the distribution**
 - > tar -xzvf torque.tar.gz
 - > cd torque
- **Enable checkpointing mechanism**
Change value of variable **MOM_CHECKPOINT** to 1 in file `src/include/pbs_config.h`

```
#define MOM_CHECKPOINT 1
```
- **Build and install the distribution**
 - > ./configure
 - > make
 - > make install

3. Torque Configuratin

- Configure **pbs_server**, **pbs_mom** and **pbs_sched** changing **appropriat files** (see documentation)
 - server_priv/nodes
 - mom_priv/config
 - sched_priv/sched_config
- **IMPORTANT!**

You must set the **\$checkpoint_script** variable ine the **mom_priv/config**

```
$checkpoint_script    /usr/local/sbin/mom-checkpoint.sh
```
- Finally define the queue (see documentation)

4. Running Torque services

- Prepare simple script which will start all services eg. /etc/init.d/torque
Remember to set proper access flags: `chmod 544 /etc/init.d/torque`

```
#!/bin/sh

case "$1" in
    start)
        /usr/local/sbin/pbs_sched
        /usr/local/sbin/pbs_server
        /usr/local/sbin/pbs_mom
        ;;
    stop)
        /usr/local/bin/qterm -t quick
        /usr/local/sbin/momctl -s
        rm -rf /var/spool/torque/mom_priv/mom.lock
        ;;
    status)
        PBS_MOM_PID=`cat /var/spool/torque/mom_priv/mom.lock`
        echo $PBS_MOM_PID
        ;;
    *)
        echo "Usage: $0 {start|stop|status}"
        exit 1
        ;;
esac
```

- Run the services
> /etc/init.d/torque start

5. Preparing checkpointing script

- Torque executes the script defined in the `mom_priv/config` file to perform the checkpoint of the job
- The script (e.g. `/usr/local/sbin/mom-checkpoint.sh`) at last should look like this example:

```
#!/bin/sh

job_pid=$1
job_id=$2
exec_pid=`/bin/ps -ef | /usr/bin/grep $1 | /usr/bin/awk '{ if ($3 == PATTERN)
    print $2; }' PATTERN=$job_pid`
OUTPUT_DIR=/var/spool/checkpoint/$exec_pid.job_id

/usr/local/bin/chkpnt -q -p $exec_pid -od $OUTPUT_DIR
```

6. Submitting job

- To submit a job to the queue you must prepare the shell script like this simple one called **run_job.sh**:

```
#!/bin/sh
```

```
my_application.exe
```

- If you want to submit job to the queue which will be checkpointable use the `qsub` command with two additional switches `-c c=period`
`>qsub -c c=1 run_job.sh`
- The application will be checkpointed automatically in 1 minute period and the image will be stored in the directories defined in the script **`/usr/local/sbin/mom-checkpoint.sh`**
- The variable `OUTPUT_DIR` defines that the directory with the image is placed in `/var/spool/checkpoint` and additionally depends on two additional variables connected with the running job `$exec_pid` `$job_id`

7. Restarting the job

- To restart job you must learn where the image was stored - simply check the directory `/var/spool/checkpoint` and prepare the script called e.g. **resume_job.sh**:

(you must change the `DIRECTORY_WITH_IMAGE` with proper directory!)

```
#!/bin/sh
```

```
/usr/local/bin/resume -drp -cwd $PBS_O_WORKDIR -od DIRECTORY_WITH_IMAGE
```

- Submit script to restart the job from image

```
> qsub resume_job.sh
```