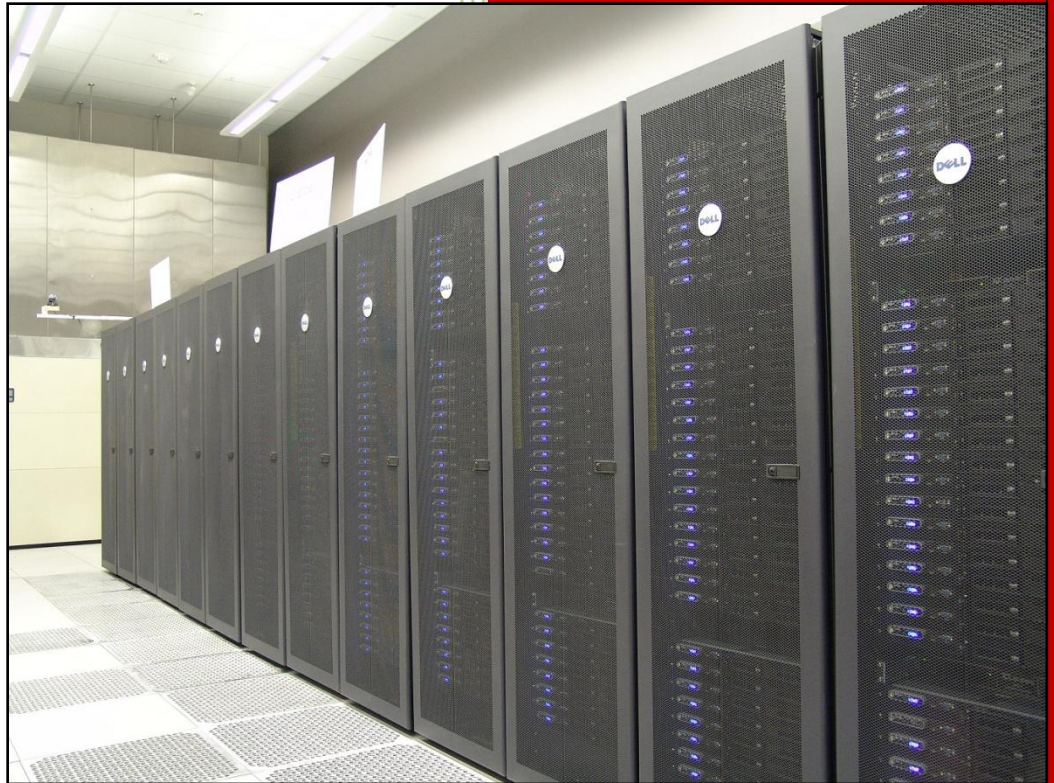


# HPCC - Hrothgar

A Tutorial of Checkpoint/Restart on Hrothgar



## Table of Contents

1. Testing Checkpoint/ restart with a sample code	3
1.1 Sample Code	3
1.2 Compile using icc or gcc	3
1.3 Submit the job	4
2. How to know if the checkpoint/restart works	4
3. Testing your own code	5
3.1 Running your own code	5

# User Guide

# A Tutorial of Checkpoint/Restart on Hrothgar

(This tutorial covers Serial Jobs)

## 1. Testing Checkpoint/ restart with a sample code

The following documentation provides a simple code that can be check pointed and shows step by step how to checkpoint it within Grid Engine.

### 1.1 Sample Code

You can test checkpoint/restart on Hrothgar by using the following simple C programme:

```
#include <stdio.h>
#include <time.h>
#define max_num 1200
int main(void)
{
    float s;
    int i, x, y, z;
    time_t pt;
    for (i=0;i<max_num;i++)
    {
        if(i%5 == 0)
        {
            time(&pt);
            printf("%si=%d\n",ctime(&pt),i);
        }
        sleep(1);
    }
    return 0;
}
```

User Guide

The above code increments the variable i each second and prints it out every 5 seconds. The maximum value is defined as 1200 which takes the job ~20min to finish. A different length can be picked by editing the definition.

### 1.2 Compile using icc or gcc

*gcc -o count count.c or  
icc -o count cont.c*

### 1.3 Submit the job

The checkpoint interval on Hrothgar is 15 min, so you can manually terminate your job by setting a soft runtime limit between the checkpoint time (00:15:00) and the finish time (00:20:00).

-l s\_rt=00:17:00. This sample program should be terminated in 17 min after starting, then restart (might be on another node) from the checkpoint of 15 min running until finish.

## 2. How to know if the checkpoint/restart works

You can check your job state by **qstat**. When your job achieves the s\_rt, it will go back to the queue waiting for a new job slot and restart to run after receiving a slot. So its state will change from **r** to **Rq** and to **Rr**.

job-ID prior name user state submit/start at queue  
slots ja-task-ID

```
-----
1986627 0.25000 test.sh saddepal Rr 02/09/2010 11:09:51 kharecc@compute-
19-7.local 1
```

You can also have a look the debug info in error file. You should see the following outputs:  
compute-19-5:\$ more checkpoint.out.1986627

```
=====
Do not find any context file, job 1986627 is a new job.
Init job 1986627 on [compute-19-5.local] from the starting point
program return value is 138
program is terminated by USR1, restart job
=====
restart job 1986627 on [compute-19-7.local] from checkpoint file:
/lustre/scratch/chkpoint/context_1986627.2
pid 12351 has not been taken on this node
```

```
...
program return value is 0
Job successfully completes
Job 1986627 cleaned up
job ends here
To Verify how many times it got checkpointed.
more test.sh.co1986664
```

```
-----
blcr_checkpoint.sh called at Tue Feb 9 12:40:09 CST 2010
called by: uid=20031(saddepal) gid=101(hpcc)
groups=101(hpcc),381(morales),602(amber),608(matlab)
```

```
with args: 1986664 11530 /lustre/scratch/chkpoint
on queue : kharecc
bash(11530)---a.out(11532)
Checkpoint command: cr_checkpoint -f context_1986664.2 --run 11532
-----
```

```
blcr_checkpoint.sh called at Tue Feb 9 12:57:24 CST 2010
called by: uid=20031(saddepal) gid=101(hpcc)
groups=101(hpcc),381(morales),602(amber),608(matlab)
with args: 1986664 15014 /lustre/scratch/chkpoint
on queue : kharecc
bash(15014)---a.out(15016)
Checkpoint command: cr_checkpoint -f context_1986664.2 --run 15016
```

### 3. Testing your own code

***cp /lustre/work/apps/examples/checkpoint.sh .***

It is suggested that you first test checkpoint/restart with the above sample code; you can see the complete output when it finishes successfully. You can also test with your own code. Although your code may takes days to finish, you can trust checkpoint/restart if your job's state is Rr after running half an hour, which will have needed to restart your job over 10 times. To achieve this you can submit your own job to the checkpoint testing nodes as follows

```
qsub -N checkpoint-test -cwd -ckpt BLCR -l s_rt=00:17:00 checkpoint.sh <full_path>/
<program>
```

#### 3.1 Running your own code

To use checkpointing you simply need to change the runtime limit to something more useful (i.e. 47 hours) and modify checkpoint.sh to appropriate queue.

```
qsub -N <job_name> -cwd -ckpt BLCR -l s_rt=<HH:MM:SS> checkpoint.sh
    <full_path>/<program>
```

**Last updated: 11/01/2010**

**For Additional Assistance Contact: [hpccsupport@ttu.edu](mailto:hpccsupport@ttu.edu)**

**For Comments/Suggestions on user guide [hpcc@ttu.edu](mailto:hpcc@ttu.edu)**

# User Guide