

Running jobs on the CS research cluster

“How to get results while web surfing!”

by
Değer **Cenk** Erdil

October 10, 2007

Outline

- Motivation
- Overview of CS cluster(s)
- Secure remote access (ssh+VNC)
- Monitoring multiple jobs
- Getting and processing results fast
- Find the least-loaded node
- Utilizing nodes remotely
- Summary and next

Motivation

- Why do you care?
 - Have fun
 - Fair-share
 - Keep admin@cs happy
 - Keep your advisor happy
 - Graduate faster!

CS Cluster

- Nodes that you can run jobs at:
 - **Cars**: Old(!) desktop machines
 - **Generals**: rommel, nimitz, etc.
 - **Players**: pele, ziko, etc. (beware!)
 - Newer ones:
 - Dual/Dual (64 ct.): node00 – node3f
 - Dual/Quad (32 ct.): node40 – node5f
- A complete list is available!

Secure Remote Access

- VNC for graphical interface
- How make it secure on linux/unix:
 - `ssh yournode`
 - `vncserver :81`
 - `ssh -f -N -T -l erdil`
 `-L5966:localhost:5981 yournode`
 - `vncviewer localhost:66`
- VNC server will run all the time
 - accessible from anywhere!

Secure Remote Access

- VNC for graphical interface
- How make it secure on wintel+putty:
 - `ssh yournode`
 - `vncserver :81`
 - create a tunnel with putty
 - ssh into yournode using putty
 - `vncviewer localhost:66`
- VNC server will run all the time
 - accessible from anywhere!

Monitoring Multiple Jobs

- Better alternatives:
 - Condor, Sun Grid Engine, Globus(?), etc.
 - may not always be available!
- Home cooked solution
 - Keep It Simple, Stupid
 - basic unix tools + bash commands
 - you can carry your scripts with you!

Monitoring Multiple Jobs

- Simple solution:
 - do
 - run program(s)
 - get results
 - parse results
 - while (!satisfied)
- Requires interaction
- Slow

Monitoring Multiple Jobs

- Let's try to automate things:

```
for param in `cat parameters.txt`  
do  
    ./mytest $param  
done
```

- Errors and output?

Monitoring Multiple Jobs

- Errors and output:

```
for param in `cat parameters.txt`  
do  
    echo "input1 yes" >input.txt  
    echo "input2 no" >>input.txt  
    ./mytest $param <input.txt >output.txt 2>&1 &  
done
```

- Separate output files?

Monitoring Multiple Jobs

- Separate output files:

```
for param in `cat parameters.txt`  
do  
    echo "input1 yes" >input${param}.txt  
    echo "input2 no" >>input${param}.txt  
    ./mytest $param <input${param}.txt  
    >output${param}.txt 2>&1 &  
done
```

- Separate directories?

Monitoring Multiple Jobs

- Separate directories:

```
for param in `cat parameters.txt`; do
  for i in `seq 2 2 10`; do
    mkdir result_${i}_${param}
    cd result_${i}_${param}
    ./mytest $param $i &
  cd ..
done
done
```

- Pause between sequences?

Monitoring Multiple Jobs

- Pause between sequences?

```
for param in `cat parameters.txt`; do [...] done
PS_OUT=`ps -ef | grep mytest | grep erdil | grep
specialinputfilename | grep [...] | wc -l`
((SECS=0))
while [ $PS_OUT -gt 0 ]; do
    echo -n $PS_OUT
    sleep 300
    ((SECS+=300))
    PS_OUT=`ps -ef | grep [...]`
done
for param in `cat parameters.txt`; do [...] done
```

Getting results fast

- Home directories are on NFS!
 - creating a network access at each read/write
- Run tests in local directories!

```
$ df -lh
```

```
/dev/sda5    38G 4.4G  34G 12% /var/work
```

- Parse results there
 - Move clean and smaller result set to NFS
- **Please clean your files afterwards!**
 - **Bug those who don't (including me!)**

Passwordless login

```
$ ssh-keygen -t rsa
```

```
$ ls .ssh/id_rsa*
```

```
id_rsa id_rsa.pub
```

```
$ cat .ssh/id_rsa.pub >>.ssh/authorized_keys2
```

```
$ ssh yournode
```

```
yournode$ logout
```

```
$ ssh yournode uname -n
```

```
yournode
```

```
$
```

Find the Least-Loaded Node

- Most nodes are heavily utilized

- Some don't:

```
for gnrl in `cat generals.txt`; do
```

```
    ssh $gnrl uptime
```

```
    ssh $gnrl free | grep Mem| awk '{print $3}'
```

```
done
```

- Pick one!
- Can we automate this?

Find me 10 nodes

- Can we automate this?

```
for node in `cat generals.txt`; do
```

```
    ./testMachine $node &
```

```
done
```

```
# wait until all nodes are added to list
```

```
[...]
```

```
# sort nodes in the order of current CPU usage
```

```
cat workingNodes | sort -k 2 -g | awk '{print $1}'
```

```
# strip the excess
```

```
cat workingNodes | [...] | head -10
```

- Utilize remote nodes?

Utilizing nodes remotely

```
# testsToRun: a list of test scripts
./getWorkingNodes ${TC} >workingNodes
WN=`wc -l workingNodes`
if [ "${WN}" -lt "${TC}" ]; then
    echo "not enough nodes"; exit
fi
diff -y -W 300 testsToRun workingNodes | awk
' {print $4 " " $2 " " $1}' >matched.lst
exec <matched.lst
while read line; do
    [...]
done
```

Utilize nodes remotely

[...]

```
PWD=`pwd`
```

```
exec <matched.lst
```

```
while read line; do
```

```
    GNRL=`echo $line | awk '{print $1}'`
```

```
    TSTNM=`echo $line | awk '{print $2}'`
```

```
    TSTNR=`echo $line | awk '{print $3}'`
```

```
    ssh -X ${GNRL} "cd /var/work/erdil;
```

```
    /home/erdil/local/scripts/do_multiple_tests.sh ${TSTNM}
```

```
    ${RUNCNT} ${PWD} ${TSTNR}" >run_${GNRL}.out
```

```
    2>&1 &
```

```
done
```

- Now you can do this from NFS!

Collect results

```
exec <finishedTests
while read line; do
    TESTNUMBER=`echo $line |awk '{print $1}'`
    GENERAL=`echo $line |awk '{print $2}'`
    TESTDIR=`echo $line |awk '{print $3}'`
    DU=`echo $line |awk '{print $4}'`
    echo "ssh $GENERAL mv $TESTDIR $PWD ;" >>moveScript
done
chmod 755 moveScript
./moveScript
echo "Done."
```

Collect results as tarballs

```
exec <finishedTests
while read line; do
    TESTNUMBER=`echo $line |awk '{print $1}'`
    GENERAL=`echo $line |awk '{print $2}'`
    TESTDIR=`echo $line |awk '{print $3}'`
    DU=`echo $line |awk '{print $4}'`
    echo "ssh $GENERAL mv
        /var/work/erdil/runtime_logs_test${TESTNUMBER}.tgz $PWD;"
        >>moveAsTGZScript
done
chmod 755 moveAsTGZScript
./moveAsTGZScript
echo "Done."
```

Summary and next

- Access the cluster securely
- Have a “remote terminal” there
- Find free nodes
 - Run jobs on them
 - output to local directories
 - Collect results
- Next:
 - Parse output via matlab
 - Use HTML templates and sed
 - to create graphs on the fly

Questions and Comments

Thank You!

Değer **Cenk** Erdil
erdil@cs.binghamton.edu