Lisa User Day 2011

Lisa architecture

John Donners
John.Donners@sara.nl

What's in this presentation?

- Overview of all nodes in lisa
- How to specify jobs for particular purposes:
- -a quick turnaround
- -highest core performance
-

Overview

Nodes	Type	Clock	Memory	Cache	Cores	Scratch	InfiniBand
32	L5420	2.5GHz	16GB FSB	12MB	8	85GB	Qlogic
128	L5520	2.26GHz	24GB QPI	8MB	8	85GB	-
256	L5520	2.26GHz	24GB QPI	8MB	8	85GB	Qlogic
32	L5640	2.26GHz	24GB QPI	12MB	12	220GB	-
64	L5640	2.26GHz	24GB QPI	12MB	12	220GB	Qlogic

What is the fastest core?

Nodes	Type	Clock freq.	Memory	Cache	Cores	Scratch	InfiniBand
32	L5420	2.5GHz	16GB FSB	12MB	8	85GB	Qlogic
128	L5520	2.26GHz	24GB QPI	8MB	8	85GB	-
256	L5520	2.26GHz	24GB QPI	8MB	8	85GB	Qlogic
32	L5640	2.26GHz	24GB QPI	12MB	12	220GB	-
64	L5640	2.26GHz	24GB QPI	12MB	12	220GB	Qlogic

=> Newer processor types are faster, especially due to QPI.
=> Minimal difference between processors with QPI.
Clock frequency is poor indicator of performance!

What is the best node type for my application?

- data-intensive: e.g. finite-difference model => nodes with QPI, 8 cores (so more bandwidth per core)
- compute-intensive: e.g. matrix operations => nodes with 12 cores, bandwidth is less important.
- memory-intensive: nodes with 24GB of memory

I want my single-node job to start asap!

- Specify as little as possible, so your job can start when any type of node becomes available.
- To find the number of cores in the node (at this moment either 8 or 12):

  $ module load saranodes

  $ wc -l < $SARA_NODEFILE

- To find the type of processor:

  $ fgrep -m1 "model name" /proc/cpuinfo

  model name : Intel(R) Xeon(R) CPU L5420 @ 2.50GHz

- To find the available memory:

  $ fgrep MemTotal /proc/meminfo

  MemTotal: 16469668 kB

I need more memory!

- use the keyword 'mem24gb':

```
#PBS -lmem24gb
```

- or, if 24 GB of memory is not enough, your application needs to be parallellized to use 'distributed memory', i.e. running on multiple nodes.
- SARA can help to parallellize your application, just ask for more details.

I need more scratch space

- Most nodes have 85GB of scratch space.
- Unfortunately, there is no separate keyword to ask for more scratch space.
- But the 12-core nodes have a scratch space of 220GB.
- so, for now, you can get more scratch space with the keyword 'cores12'.
- However, note that this situation could change in the future.
- You can see the total and available amount of scratch space with the command:

```
$ df -h /scratch
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda5             228G  342M  228G   1% /scratch
```

I need more scratch space (2)

- For multi-node jobs, there's also the possibility for a global scratch space, using GlusterFS.
- The scratch disks of the first two nodes are combined into one filesystem of 170-440GB.
- That filesystem is globally visible on all nodes of your job as the directory /**global**.

- Note that the variable $TMPDIR still points to the local scratch directory (which is still usable)
- Eplicitly use /**global** in your job to access the global scratch space.

- Interested? Send an email to **hic@sara.nl** with a request to use GlusterFS.

I need a fast turnaround for tests

- please use a wall clock time of 5 minutes:

```
#PBS -lwalltime=5:00
```

- Your job is automatically placed in the express queue
- Some nodes are especially reserved for express jobs.
- So it should start as soon as possible.

- Do NOT use the interactive nodes for tests!

interactive access to batch nodes

- Use the option -I:

```
donners@login3:~$ qsub -I -lwalltime=5:00
qsub: waiting for job 5635559.batch1.irc.sara.nl to start
qsub: job 5635559.batch1.irc.sara.nl ready

donners@gb-r7n4:~$ echo $TMPDIR
/scratch/5635559.batch1.irc.sara.nl
donners@gb-r7n4:~$ exit
logout

qsub: job 5635559.batch1.irc.sara.nl completed
donners@login3:~$
```

- CPU time is still budgetted
- Do NOT use the login nodes for tests!

“Job rejected by all possible ..”

- Usually a typo in the qsub or PBS commands:

- donners@login3:~$ qsub -lnodes=1:cores9 job
  qsub: Job rejected by all possible destinations

- donners@login3:~$ qsub -lnode=1:cores8 job
- donners@login3:~$ qsub -lcores8:nodes=1 job
- donners@login3:~$ qsub -lmem32gb job

- #PBS -lnodes=1:cores2:ppn=1
- #PBS -lnodes=1:cores=12

- $ qsub -lnodes=1:cores8 job
- #PBS -lnodes=1:cores12

How can I run a multi-node MPI job?

Nodes	Type	Clock	Memory	Cache	Cores	Scratch	InfiniBand
32	L5420	2.5GHz	16GB FSB	12MB	8	85GB	Qlogic
128	L5520	2.26GHz	24GB QPI	8MB	8	85GB	-
256	L5520	2.26GHz	24GB QPI	8MB	8	85GB	Qlogic
32	L5640	2.26GHz	24GB QPI	12MB	12	220GB	-
64	L5640	2.26GHz	24GB QPI	12MB	12	220GB	Qlogic

- It is best to choose a homogeneous set of nodes: then all cores are equally fast and processes are not waiting for each other.
- Choose:
  - **-lnodes=3:cores8:mem24gb**
  - **-lnodes=2:cores12**
  - **-lnodes=3:cores8:mem16gb**

Can I use 8- and 12-core nodes together?

- YES!
-
- But please specify 'infiniband':
- **-lnodes=3:mem24gb:infiniband** (recommended, only nodes with QPI)
- **-lnodes=3:infiniband** (possibly with older 8-core nodes using FSB)

- But a mix of nodes is only useful if your MPI-application does some dynamic load balancing, so it adjusts the workload to the performance of the processor.

Can I use 8- and 12-core nodes together?

- NO!
- Or, at least: not efficiently or reliably.
- There are some unexplained problems with a mix of 8- and 12-core nodes.

Interactive jobs

- It is possible to start interactive jobs:

```
login4:~$ qsub -I -lwalltime=30:00 -lnodes=1:cores12
qsub: waiting for job 5629382.batch1.irc.sara.nl to start
qsub: job 5629382.batch1.irc.sara.nl ready
gb-r36n24:~$
```

- Useful for testing purposes.
- Jobs would queue like any other job, so a possibly long waiting time.

What nodes do I get?

Specification	# of nodes	type of nodes	#MPI/ node	total # MPI processes
-lnodes=3:cores8:ppn=8	3	8-core	8	24
-lnodes=3:cores12:ppn=12	3	12-core	12	36
-lnodes=3:ppn=4	3	8- or 12-core	4	12
-lnodes=3:cores8	3	8-core	1	3

- cores8 or cores12: When neither cores8 or cores12 is specified, the system will first try to allocate 8-core nodes. If there are no 8-core nodes available anymore, 12-core nodes will be allocated, possibly resulting in a mix of 8- and 12-core nodes.
- a single-node job is first allocated to nodes without InfiniBand
- a multi-node job is only allocated to nodes with InfiniBand.

Thank you for your attention!

Any questions?