HPC Cluster at CNAF

Antonio Falabella

INFN - CNAF (Bologna)

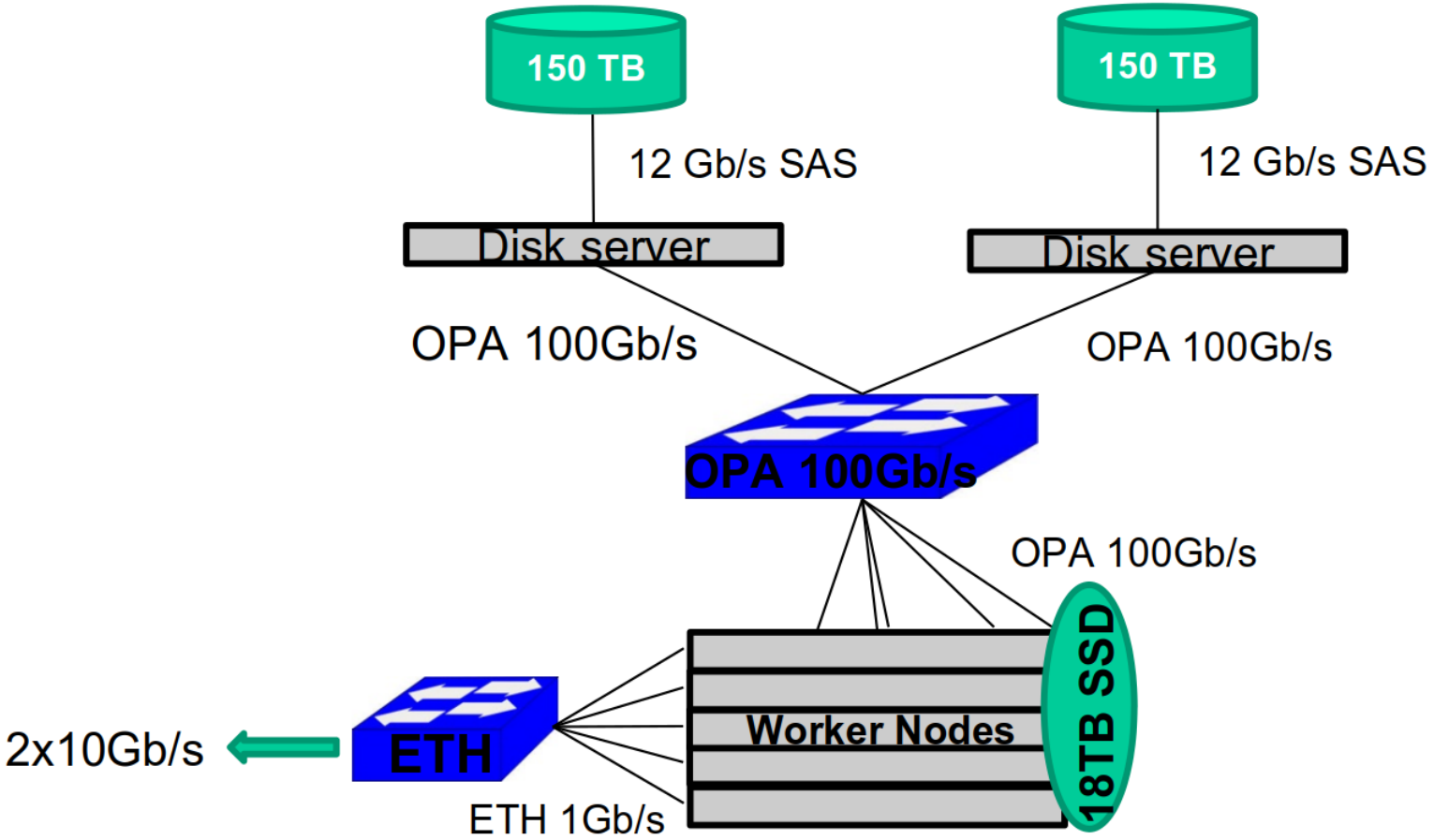
July 21, 2017- ABP-CWG meeting #18

Overview

HPC Cluster at CNAF

- Computing Units: 3 Chassis DELL PowerEdge C6300 (2U) each of them provides 4 computing nodes equipped with:
  - 384 physical CPU cores with HT
  - 2 x Cpu E5-2683v4 2.1Ghz 40MCache,8.0GT/s, 16C/32T (TDP 120W)
  - 8x32GB RDIMM, 2400MT/s Dual Rank,x4 Data Width (8GB/physical core)
- Local Storage:
  - 1 x 480GB Solid State Drive SATA Read Intensive MLC
  - 1 x 1.6TB Solid State Drive SATA Mix Use MLC 6Gbps
- Network connection:
  - 2x1Gbs eth ports
  - 1 x 100gbs OPA HFI
- Fabric Switch:
  - Switch Omni-PathH1048
  - Aggregate Bandwidth: 9.6Tbps
  - Port number: 48 QSFP28
  - Single port bandwidth: 100Gbits
- Shared Storage:
  - 2 x JBOD each installed with 24 HDDs 6TB SAS interface 12Gbs
  - 2 x Disk servers
  - Raw capacity of the shared filesystem is 288TB exported by a GPFS configured in replica 2 providing a net capacity of 144TB.
  - A second shared filesystem is configured combining the local storage of each computing node for a total of 18TB used to host the users home directories.

HPC Cluster at CNAF



Software Stack

- LSF 9.1 to access the cluster
- GPFS for the shared file systems
- gcc 4.8.5
- icc, ifort
- python 2.7.5 (different version of additional packages if requested)
- mpi 2.1 (OpenMPI, mvapich2 compiled with gcc and intel compiler)
  - You can list the installed implementation with:

```
mpi-selector --list
```

Cluster Access

- The INFN-CNAF user FAQs can be found here at https://www.cnaf.infn.it/en/users-faqs
- A specific wiki for this cluster is in progress
- To request and account at INFN-CNAF the first step is to read the AUP:
  - https://www.cnaf.infn.it/wp-content/uploads/2016/10/AUP_en.pdf
- Then you have to fill the form:
  - https://www.cnaf.infn.it/wp-content/uploads/2015/09/accesso-cnaf-EN.pdf
  - The reason for application : "Access to the HPC cluster"
  - Name of the INFN-CNAF contact person : "Daniele Cesini"
  - If you don't plan to come to CNAF if can leave the part "Request Access to the INFN-CNAF Network" blank
- If you are not an INFN associate you have to provide also a scan copy of a valid ID document
- Send the documents to **abp-cwg-admin@cern.ch**
- The mailing list for the technical support is **hpc-support@lists.cnaf.infn.it**

Use of the Cluster

- When you have your account you can access the cluster by logging into the CNAF bastion:

```
$> ssh <username>@bastion.cnaf.infn.it
```

- and then to the HPC user interface:

```
$> ssh <username>@ui-hpc2.cr.cnaf.infn.it
```

- **NOTE:** This is a virtual machine not being part of the cluster, so any CPU intensive session is discouraged
- The 12 worker nodes are:

```
hpc-201-11-01-a.cr.cnaf.infn.it
hpc-201-11-01-b.cr.cnaf.infn.it
hpc-201-11-02-a.cr.cnaf.infn.it
hpc-201-11-02-b.cr.cnaf.infn.it
hpc-201-11-03-a.cr.cnaf.infn.it
hpc-201-11-03-b.cr.cnaf.infn.it
```

```
hpc-201-11-04-a.cr.cnaf.infn.it
hpc-201-11-04-b.cr.cnaf.infn.it
hpc-201-11-05-a.cr.cnaf.infn.it
hpc-201-11-05-b.cr.cnaf.infn.it
hpc-201-11-06-a.cr.cnaf.infn.it
hpc-201-11-06-b.cr.cnaf.infn.it
```

- You can login into one of them for short prototyping sessions
- For job submission you should use the LSF batch system

LSF batch scheduling

- The batch system installed is IBM Platform LSF Standard 9.1.3.0
- The link to documentation is :
- https://www.ibm.com/support/knowledgecenter/en/SSETD4_9.1.3/lsf_welcome.html
- For user submission and management the box "Working with jobs" contains all the relevant informations
- The first command is "bhosts" to check the host and the jobs they are running

```
$> bhosts
HOST_NAME          STATUS       JL/U    MAX  NJOBS    RUN  SSUSP  USUSP  RSV
hpc-201-11-01-a    closed_Full  -        64     64     64      0      0    0
hpc-201-11-01-b    closed_Admin -        64      0      0      0      0    0
hpc-201-11-02-a    ok           -        64     48     48      0      0    0
hpc-201-11-02-b    ok           -        64     48     48      0      0    0
hpc-201-11-03-a    ok           -        64     48     48      0      0    0
hpc-201-11-03-b    ok           -        64     48     48      0      0    0
hpc-201-11-04-a    ok           -        64     32     32      0      0    0
hpc-201-11-04-b    ok           -        64     32     32      0      0    0
hpc-201-11-05-a    ok           -        64     48     48      0      0    0
hpc-201-11-05-b    ok           -        64     48     48      0      0    0
hpc-201-11-06-a    ok           -        64     48     48      0      0    0
hpc-201-11-06-b    ok           -        64     48     48      0      0    0
```

- "closed_Full" means no more jobs will be scheduled, "closed_Admin" means closed by LSF daemons of admins, "ok" will accept new jobs

LSF batch scheduling

- LSF is configured in submission queues
- The queue for your group is "hpc_acc"

```
$>bqueues
QUEUE_NAME  PRIO  STATUS          MAX  JL/U  JL/P  JL/H  NJOBS  PEND   RUN  SUSP
hpc_acc        1  Open:Active       -     -     -    64    512     0   512     0
```

- The scheduling policy is "fairshare" which means that the priority of a job is recomputed for waiting jobs. The more a job will wait the more priority it will gain
- If the job remains in a pending state indefinitely it's better to investigate (see next slides) or contact the support

LSF job submit

- The command to submit a job is "bsub"

```
$>bsub "echo Hello"
Job <81922> is submitted to default queue <hpc_short>.
```

- If you don't specify a queue the default is "hpc_short"
- The default queue is configured with a very short run limit meaning that if you submit a job that lasts more than 6 hours it will be killed by the batch system

```
bsub -q hpc_acc "echo Hello"
Job <81923> is submitted to queue <hpc_acc>.
bjobs
JOBID USER    STAT  QUEUE    FROM_HOST EXEC_HOST JOB_NAME
81924 falabel PEND  hpc_acc  ui-hpc2             echo Hello
```

LSF MPI job submit

- To submit an MPI job you first need to choose the version you want to use:

```
$> mpi-selector --list
mvapich2_gcc-2.1
mvapich2_gcc_hfi-2.1
mvapich2_intel_hfi-2.1
openmpi_gcc-1.10.4
openmpi_gcc_hfi-1.10.4
openmpi_intel_hfi-1.10.4
```

- In this example I chose the OpenMPI implementation version compiled with gcc using the OmniPath drivers (hfi)

```
$>mpi-selector --set openmpi_gcc_hfi-1.10.4
Defaults already exist; overwrite them? (y/N) y
$> which mpirun
/usr/mpi/gcc/openmpi-1.10.4-hfi/bin/mpirun
```

LSF MPI job submit

- Prepare a machine file containing a subset of available hosts:

```
hpc-201-11-01-a.cr.cnaf.infn.it
hpc-201-11-01-b.cr.cnaf.infn.it
hpc-201-11-02-a.cr.cnaf.infn.it
hpc-201-11-02-b.cr.cnaf.infn.it
hpc-201-11-03-a.cr.cnaf.infn.it
hpc-201-11-03-b.cr.cnaf.infn.it
hpc-201-11-04-a.cr.cnaf.infn.it
hpc-201-11-04-b.cr.cnaf.infn.it
hpc-201-11-05-a.cr.cnaf.infn.it
hpc-201-11-05-b.cr.cnaf.infn.it
hpc-201-11-06-a.cr.cnaf.infn.it
hpc-201-11-06-b.cr.cnaf.infn.it
```

```
bsub -q hpc_acc  -a openmpi -m 'hpc-201-11-01-a hpc-201-11-01-b hpc
     -201-11-02-a hpc-201-11-02-b' -n 8  -R span[ptile=1] -o testmpi.
     out -e testmpi.err
'/usr/mpi/gcc/openmpi-1.10.4-hfi/bin/mpirun -np 8 -machinefile
     machinefile.xt /home/HPC/falabellahpc/reduce_pi'
```

- NOTE: You have to specify the list of machines to pass to the "-m" option even if you provide the machinefile.txt

Monitor your jobs

- To check the status of your jobs you use the command "bjobs":

```
bjobs -w -u falabellahpc
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME
    SUBMIT_TIME
81933   falabellahpc PEND  hpc_acc    ui-hpc2       -        /usr/
    mpi/gcc/openmpi-1.10.4-hfi/bin/mpirun -np 1 -machinefile
    machinefile.xt /home/HPC/falabellahpc/reduce_pi Jul 18 15:46
81934   falabellahpc PEND  hpc_acc    ui-hpc2       -        /usr/
    mpi/gcc/openmpi-1.10.4-hfi/bin/mpirun -np 1 -machinefile
    machinefile.xt /home/HPC/falabellahpc/reduce_pi Jul 18 15:46
81935   falabellahpc PEND  hpc_acc    ui-hpc2       -        /usr/
    mpi/gcc/openmpi-1.10.4-hfi/bin/mpirun -np 1 -machinefile
    machinefile.xt /home/HPC/falabellahpc/reduce_pi Jul 18 15:47
81936   falabellahpc PEND  hpc_acc    ui-hpc2       -        /usr/
    mpi/gcc/openmpi-1.10.4-hfi/bin/mpirun -np 8 -machinefile
    machinefile.xt /home/HPC/falabellahpc/reduce_pi Jul 18 15:47
```

- By default only PENDING jobs are showed
- With the option "-p" you can see why the jobs are in pending state
- To have information of already finished jobs you can use the command "bhist"

Monitor your jobs

- To check the status of all your jobs you use the command "bjobs -a":

```
bjobs -a -u falabellahpc
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME
     SUBMIT_TIME
81933   falabel PEND  hpc_acc    ui-hpc2                 *reduce_pi
     Jul 18 15:46
81934   falabel PEND  hpc_acc    ui-hpc2                 *reduce_pi
     Jul 18 15:46
81946   falabel PEND  hpc_acc    ui-hpc2                 *reduce_pi
     Jul 18 16:52
81922   falabel DONE  hpc_short  ui-hpc2     hpc-200-06- echo Hello
     Jul 18 14:45
81923   falabel DONE  hpc_acc    ui-hpc2     hpc-201-11- echo Hello
     Jul 18 14:47
81924   falabel DONE  hpc_acc    ui-hpc2     hpc-201-11- echo Hello
     Jul 18 14:48
81928   falabel EXIT  hpc_acc    ui-hpc2        -        *reduce_pi
     Jul 18 15:12
81947   falabel DONE  hpc_acc    ui-hpc2     4*hpc-201-1 *reduce_pi
     Jul 18 16:53
                                              4*hpc-201-11-02-b
```

Job history

```
bhist -l 81884

Job <81884>, Job Name <int_0030>, User <giadarol>, Project <default>,
     Command <
        #BSUB -J int_0030;#BSUB -o %J.out;#BSUB -e %J.err;#BSUB -N
       ;#BSUB -B;#BSUB -q hpc_acc;#B -a openmpi;#BSUB -n 16;#BSUB
        -R span[ptile=16]; source setup_env_cnaf; CURRDIR=/home/H
        PC/giadarol/sim_workspace_PyECPyHT/BG017_HLLHC_450GeV_dipO
        FF_quadON_Qpscan_oct-2_damperON_ppbscan/simulations/ch20.0
        _1.6e11ppb_ecdipOFF_ecquadON;cd $CURRDIR;pwd; stdbuf -oL p
        ython ../../../PyPARIS/multiprocexec.py -n 8 sim_class=Sim
        ulation_with_eclouds.Simulation >> opic.txt 2>> epic.txt>
Tue Jul 18 11:41:54: Submitted from host <hpc-201-11-02-b>, to Queue <
     hpc_acc>,
        CWD <$HOME/sim_workspace_PyECPyHT/BG017_HLLHC_450GeV_dipO
        FF_quadON_Qpscan_oct-2_damperON_ppbscan/simulations/ch20.0
        _1.6e11ppb_ecdipOFF_ecquadON>, Output File <%J.out>, Error
        File <%J.err>, Notify when job begins/ends, 16 Processors
        Requested, Requested Resources <span[ptile=16]>;
Tue Jul 18 11:41:55: Dispatched to 16 Hosts/Processors <16*hpc-201-11-05-b
     >, Ef
        fective RES_REQ <select[(type == any )] order[r15s:pg] spa
        n[ptile=16] >;
Tue Jul 18 11:41:55: Starting (Pid 102443);
Tue Jul 18 11:41:55: Running with execution home </home/HPC/giadarol>,
     Executio
        n CWD </home/HPC/giadarol/sim_workspace_PyECPyHT/BG017_HLL
        HC_450GeV_dipOFF_quadON_Qpscan_oct-2_damperON_ppbscan/simu
        lations/ch20.0_1.6e11ppb_ecdipOFF_ecquadON>, Execution Pid
        <102443>;
Tue Jul 18 16:12:38: Done successfully. The CPU time used is 111421.2
     seconds;
Tue Jul 18 16:12:44: Post job process failed;
```

Job Accounting

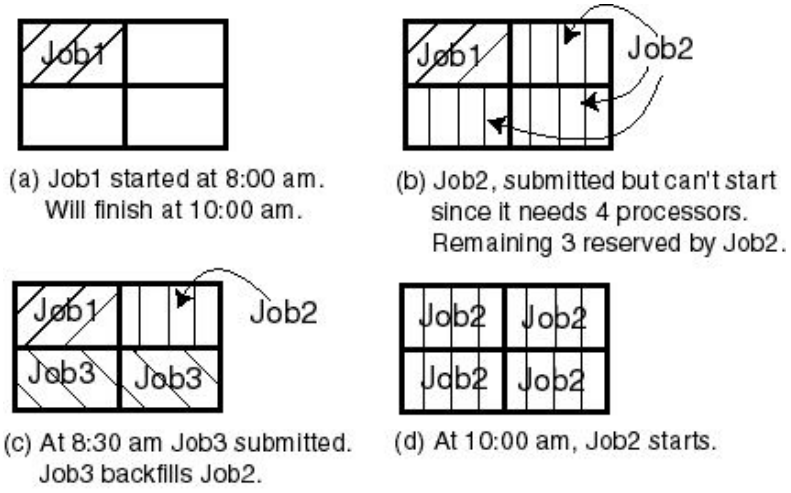
```
bacct -l 81884

Accounting information about this job:
     Share group charged </giadarol>
     CPU_T     WAIT     TURNAROUND   STATUS     HOG_FACTOR     MEM      SWAP
 111421.18        1          16244     done          6.8592    2.6G        0M
------------------------------------------------------------------------------


SUMMARY:      ( time unit: second )
 Total number of done jobs:       1      Total number of exited jobs:
     0
 Total CPU time consumed:   111421.2      Average CPU time consumed:
     111421.2
 Maximum CPU time of a job: 111421.2      Minimum CPU time of a job:
     111421.2
 Total wait time in queues:      1.0
 Average wait time in queue:     1.0
 Maximum wait time in queue:     1.0      Minimum wait time in queue:
     1.0
 Average turnaround time:      16244 (seconds/job)
 Maximum turnaround time:      16244      Minimum turnaround time:
     16244
 Average hog factor of a job:  6.86 ( cpu time / turnaround time )
 Maximum hog factor of a job:  6.86      Minimum hog factor of a job:
     6.86
```

Processor reservation

- If you submit a parallel job concurrently to sequential jobs it may remain in pending status indefinitely
- To overcome this we can enable processor reservation
- If this is enabled the system keep the job slots empty to fulfil the parallel job requirements
- To avoid having several empty slots we can enable back filling as well



(a) Job1 started at 8:00 am. Will finish at 10:00 am.

(b) Job2, submitted but can't start since it needs 4 processors. Remaining 3 reserved by Job2.

(c) At 8:30 am Job3 submitted. Job3 backfills Job2.

(d) At 10:00 am, Job2 starts.

- To submit jobs to idle job slots with backfilling you have to specify the run limit of the job with "bsub -W [hour:]minute"
- **NOTE:** LSF uses that value as a hard limit and terminates jobs that exceed the specified duration

Thank you!

Any Questions?

Additional settings

- For "hpc_acc" queue the run and cpu limits are the default ones (21 days)
- the run limit should be configured according to the group needs, to avoid the load of the cluster by a single user
- Configuring additional queues: e.g. short high priority queues
- Specific software installation
- The cluster will be extended by the end of the with additional worker nodes equipped with GPGPUs