CICLAD cluster

Calcul Intensif for Climat, Atmospher and Dynamic

http://ciclad-web.ipsl.jussieu.fr

Sysadmin
Philippe.Weill@latmos.ipsl.fr

05/01/2015

Documentations and Access

- Documentation and account: http://ciclad-web.ipsl.jussieu.fr
- Access:
  - ssh user@ciclad.ipsl.jussieu.fr
  - ssh user@ciclad2.ipsl.jussieu.fr
  - ssh user@ciclad1.ipsl.jussieu.fr(old config centos 5)
- For problems and software update: svp-ciclad@ipsl.jussieu.fr
- Mailing list : http://courriel.ipsl.jussieu.fr/mailman/listinfo/ciclad-infos
- /home 20Go for each user with Backup
- In your Jobs don't use /home for input data and computation results (performance problems)

Backup

- Only /home has backup
- for /home, if you remove files, you could find a version from last night in

  /backup/home_backup

CICLAD Story

- Project start in last 2007
- first test in february 2008 (50K€)
- Production in September 2008 (180K€)
  - 4 I/O nodes (70To) and 10 computation nodes (80 Cores)
- Production Phase II September 2010
  - Now ciclad is a storage , calculation and data distribution plateform on IPSL
- Start of CICLAD III (Summer 2014 -)
  - **1Gb/s dedicated link** to **IPSL CLIMSERV** cluster at polytechnique school cross **read-only** filesystems mounting beetween the 2 cluster (Operationnal summer 2014)
  - Increase computation nodes (128 more core since december 1st)
  - Changing linux distribution on all compute node (ciclad and climserv) (End 2014)
    - From Centos 5.x to Scientific Linux 6.x
  - Changing Lustre Version 1.8.9wc1 --> 2.5.x and increasing disk space (beginning of 2015)

What's CICLAD NOW?

- Big Storage Space (> 1 Peta Byte)
- Cluster (640 AMD core 2.4GHz) 24 nodes
- One virtualization Infrastructure for data distribution
- High speed network with low latency between nodes (40Gb/s InfiniBand)

Storage Infrastructure

- **15 I/O nodes (Infiniband interconnexion)**
- **14 disk rack (more than 700 Hard disks)**
- **5 Lustre parallel file system**
  - **More than 2Gb/s write for /data filesystem**
  - **If we have some problems on filesystems, I/O are blocked and jobs continue to run after the filesystem is restarted.**

    This is a good feature in batch mode , less in interactive mode ;-)
  - **Don't work with many small files (< 1Mo) Think about this in your applications**

Compute nodes

- 24 Nodes (640 AMD core 64 bit 2,4 GHz)
  - Computer with 8 , 12, 32 or 64 Core
  - 4GB Ram memory by Core on any node
  - Infiniband Inter-connexion between node
  - 3 external access nodes by ssh with key

    (ciclad.ipsl.jussieu.fr (ciclad-ng), ciclad2 and ciclad1)
    - Please don't use ciclad-ng, ciclad1 and ciclad2 for computation, you could do qsub -IV for interactive jobs
- batch system (torque-maui)
- Many Scientific software and compiler

Virtualization Infrastructure

- Shared with all IPSL services at UPMC
- 6 nodes (1 for Backup ,4 to run Virtual machines, 1 for control and monitoring)
- Virtualization System: VMWARE
  - Website , data distribution ....

Software Configuration

- Scientific Linux 6.6 on all nodes: https://www.scientificlinux.org
  - Scientific software is using now module command
- LUSTRE filesystem: http://opensfs.org
- Batch System torque with maui scheduler http://www.clusterresources.com/pages/products/torque-resource-manager.php
- Idl (6.4 et 8.0.1 (idl80)) matlab (2010b) scilab
- Commercial fortran compiler :pgf95(Portland) ifort(Intel) nagfor (NAG)
- Free fortran compiler : gfortran
- Parallelism : openmpi
- Others : ferret, gmt, netcdf, cdo, nco, ncl, epd python … (module avail to see)

Module command

- Now ipsl mesocenter (ciclad and climserv) is using module command for software initialization
- Syntax:
  - module (avail [product] | load product[/version] | list | switch product/version1 product/version2 | display product[/version] …)
    - "avail" : list all known product and versions
    - "load" : initialize a product in his default version if no version is specified
    - "list" : list already loaded product and version
    - "switch" : changing the version of already loaded product
    - "purge" : unload product
    - "display" : show the module file
- Product could be a compiler, a library or a software

Module installed

```
[root@ciclad-ng ~]# module avail

------------------------------------------------- /usr/share/Modules/modulefiles -------------------------------------------------
dot         module-git  module-info modules     null        use.own

------------------------------------------------- /etc/modulefiles/Compilers -------------------------------------------------
gnu/4.4.7(default)        intel/12.1.3.293(default) nagfor/5.3(default)       pgi/2011                pgi/2013(default)

------------------------------------------------- /etc/modulefiles/Libraries -------------------------------------------------
hdf5/1.8.10.patch1-gfortran  netcdf4/4.2.1.1-ifort          openmpi/1.4.5-pgf2011       openmpi/1.6.5-ifort
hdf5/1.8.10.patch1-ifort     netcdf4/4.2.1.1-pgf2011        openmpi/1.4.5-pgf2011gcc    openmpi/1.6.5-pgf2011
hdf5/1.8.10.patch1-pgf2011   netcdf4/4.2.1.1-pgf2013        openmpi/1.4.5-pgf2013       openmpi/1.6.5-pgf2013
hdf5/1.8.10.patch1-pgf2013   openmpi/1.4.5-gfortran         openmpi/1.4.5-pgfgcc        openmpi/1.6.5-pgfgcc
netcdf4/4.2.1.1-gfortran     openmpi/1.4.5-ifort            openmpi/1.6.5-gfortran

------------------------------------------------- /etc/modulefiles/Products -------------------------------------------------
cdo/1.6.3(default)                idl/8.2                   python/2.7.3-epd7
ferret/6.7.2(default)             matlab/2010b.sp2           python/2.7.6-canopy-1.3(default)
ferret/6.9                        matlab/2013b(default)      python/2.7-anaconda
gmt/4.5.11(default)               ncl/6.1.2(default)         python/3.4.1-anaconda3
grads/2.0.2(default)              nco/4.3.7(default)         R/3.1.1
grads/2.0.2.oga.2-opengrads       nco/4.4.2                  scilab/5.4.1(default)
idl/6.4(default)                  pyferret/1.1.0             scilab/5.5.1
```

Module command Sample

```
[weill@ciclad-ng ~]$ type pgf90
pgf90 est /opt/pgi-2013/linux86-64/2013/bin/pgf90
[weill@ciclad-ng ~]$ module load pgi/2011
[weill@ciclad-ng ~]$ type pgf90
pgf90 est /opt/pgi-2011/linux86-64/2011/bin/pgf90
[weill@ciclad-ng ~]$ type matlab
matlab est /opt/matlab-2013b/matlab
[weill@ciclad-ng ~]$ module load matlab/2010b.sp2
[weill@ciclad-ng ~]$ type matlab
matlab est /opt/matlab-2010b.sp2/matlab
[weill@ciclad-ng ~]$ module switch matlab/2013b
[weill@ciclad-ng ~]$ type matlab
matlab est /opt/matlab-2013b/matlab
[weill@ciclad-ng ~]$ module list
Currently Loaded Modulefiles:
  1) pgi/2011       2) matlab/2013b
[weill@ciclad-ng ~]$ module purge
[weill@ciclad-ng ~]$ type pgf90
-bash: type: pgf90 : non trouvé
[weill@ciclad-ng ~]$ module list
No Modulefiles Currently Loaded.
```

Batch System (commands)

- Launch a job : qsub
- How many jobs in queue : qstat or showq
- To kill a job : qdel
- Nodes state : check-cluster
  pbsnodes
- To modify a job : qalter

Execution Queues

- qstat -q to show all queues definitions
- 8 queues: short(Default) std h12 day days3 week weeks2 infini
- WALLTime are in Hour CPUTIME is unlimited

```
Queue            Memory CPU Time Walltime Node  Run Que Lm  State
---------------- ------ -------- -------- ----  --- --- --  -----
short            --     --       02:00:00 --    0   0 --   E R
default          --     --          --    --    0   0 --   E R
std              --     --       06:00:00 --    1   0 --   E R
h12              --     --       12:00:00 --    0   0 --   E R
day              --     --       24:00:00 --    0   0 --   E R
days3            --     --       72:00:00 --    0   0 --   E R
week             --     --       168:00:0 --    0   0 --   E R
weeks2           --     --       340:00:0 --    0   0 --   E R
infini           --     --       840:00:0 --    0   0 --   E R
```

User Limits

- All those limits could change to reflect need and usage of the CICLAD cluster :
- Default limits for standard user are on december 2012:
  - No more than 24 actives Jobs or 24 cpu core
  - No more than 3 active jobs in infini queue
  - No more than 4 active jobs in weeks2 queue
  - No more than 8 active jobs in week queue
  - No more than 16 active jobs in days3 queue
  - No more than 24 active jobs in other queue
  - By default memory is limited to 4GB by process
- ***Please don't stress interactive nodes*** *you could do interactive work with batch system:*
- qsub -IV [-q short|std|h12]

pbsnodes and check-cluster commands

- pbsnodes -l : list out of order nodes or reserved node for sysadmin
- pbsnodes -a : list all nodes state
- check-cluster (local command)

```
[weill@ciclad1 ~]$ check-cluster
NODE         STATE        FREE-CPU    FREE-MEM    LOAD  PROPERTY   PARTITION
ciclad5      Running         1/8      10/31 Gb    7.08  [std]      std
ciclad10     Idle            8/8      31/31 Gb    0.24  [std]      std
ciclad18     Busy           0/32     26/126 Gb   32.13  [std]      std
ciclad19     Running       63/64    249/252 Gb    0.62  [heppi]    std
ciclad20     Idle          64/64    252/252 Gb    0.06  [heppi]    std
ciclad21     Idle          64/64    252/252 Gb    0.04  [heppi]    std
ciclad22     Running       32/64    152/252 Gb   32.05  [heppi]    std

 10 Active Jobs      72 of  304 Processors Active (23.68%)
                     4 of    7 Nodes Active      (57.14%)
Total Jobs: 12  Active Jobs: 10   Idle Jobs: 0   Blocked Jobs: 2
```

qstat command

- qstat (without option) : list all jobs (not only yours)
- qstat -q : give queue definition information
- qstat -a : list all jobs with more info (should be a good default)
- qstat -n : list also node used by jobs
- qstat -f “num_job”: give all info on one job

showq command

- Could be complementary with qstat
- 3 states for a job
  - Running
  - Idle (no resources for running now but should start when there is free resource
  - Blocked (you're over your limits or asking impossible resources)
  - showq
  - showq -i to see idle jobs
  - showq -b to see why jobs are blocked

Qdel job_num

- Qdel job_num (you can find job number by qstat or showq)

qsub command (1)

- First thing to know is that only ShellScript (no binary files) could be submited to batch system and by default script start in your HOME so you have to go in the good directory in your job
- Minimum shell script is :

```
#!/bin/bash

my_binary
```

- Shell script must be runable so you have to do

```
chmod +x my_shell_script

qsub my_shell_script
```

- At the end of your jobs you have two files :
  - `“my_shell_script”.o”jobs_number”` : Standard output of your job and node computation information (node name running , cputime walltime and memory used by your job
  - `“my_shell_script”.e”jobs_number”` : output error of your job in case of problems

qsub command options (2)

- qsub [-a date_time] [-A account_string] [-c interval] [-C directive_prefix] [-e path] [-h] [-I] [-j join] [-k keep] [-l resource_list][-m mail_options] [-M user_list] [-N name] [-o path] [-p priority] [-qdestination] [-r c] [-S path_list] [-u user_list] [-v variable_list][-V] [-W additional_attributes] [-z] [script]
- You could see also on ciclad

  man qsub

qsub command (3)

- qsub -q«queue_name» my_shell_script
- qsub -IV (warning -VI doesn't work anymore)
  - To launch interactive jobs
- qsub -j oe
  - Joining standard and error output in one file (.o”job_number”)
- qsub -k oe
  - Keeping Standard and error output : could be see during the execution of your job but in this case those files are in the roots of your home (warn for quota if you have debug output in your job)

Qsub commands (advanced) (4)

- qsub -v variable_name=”variable_value” permit to pass variables to shell script (argument on command line are not supported by torque)

  in the script you can use $variable_name

- You could put job directives directly in the script using line begining by #PBS
  - Like this :

    #PBS -qweek

    #PBS -k oe

    #PBS -j oe

qsub commands (5)

- qsub -j oe -k oe job.sh : with this you could view in realtime output of your job in your home

  tail -f job.sh.exxx

- You can use idl,matlab,scilab in batch jobs only if there is no X11 graphic in your job :
  - Idl
  - Matlab (with -nodisplay option)

  graphic should be produced directely in ps,eps ,pdf or image

qsub command(6)

- Default limit for jobs memory is :
  - 4GB of virtual memory per Job
  - 3GB of real memory per Job
- How to launch job with more memory ?
    - You could do that in qsub
    - mem : memory per job
    - vmem : virtual memory per job
  - qsub -l mem=10gb -l vmem=12gb

Mem: 32442980k total, 21798348k used, 10644632k free, 47112k buffers
Swap: 67111528k total, 2499232k used, 64612296k free, 14130248k cached

PID	USER	PR	NI	***VIRT***	***RES***	SHR	S	%CPU	%MEM	TIME+	COMMAND
12355	weill	25	0	***9164m***	***8.1g***	8.1g	S	44	26.3	1:08.55	MATLAB

- Warning don't put those values to high without good reason

qsub advanced(7)

- Starting a job at specific time :
  - qsub -a “time :

  qsub -a 2010 # launch job 20H10

  qsub -a $(date -d '+2min' +%H%M) # launch this le job 2minutes after the submit): use this when you have a job that submit another job at the end
- Specifying walltime cputime :

  qsub -l walltime=20:00:00,cput=80:00:00
  - 20H of walltime, 80H of cputime
- jobs dependancy : running a job after the successful run of another job[s]

  qsub -W depend=afterok:jobid[:jobid...]

Parallelism(1)

- submission of parallels jobs :
  - qsub -l “nodes=n:ppn=m”
    - I want n nodes with m core by nodes
    - Samples : qsub -l “nodes=1:ppn=8”
    - qsub -l “nodes=2:ppn=4”
    - qsub -l “nodes=4:ppn=1”
    - Special request: qsub -l “nodes=1:ppn=2+nodes=3:ppn=8”
    - One node with 2 core and 3 nodes with eight core
- Job mpi :
  - mpirun program
  - Batch system pass the number of cpu to mpi program

    (-np is not necessary except for special purpose)

Parallelism (2)

- At the start of a parallel job the shell variable PBS_NODEFILE contain the filename of the list of node dedicated by batch system

  sample

  - more foo.sh
    cat $PBS_NODEFILE
  - qsub -l “nodes=4:ppn:1” foo.sh
  - cat foo.sh.o56517
    [...]
    ciclad2.ipsl.jussieu.fr
    ciclad10.ipsl.jussieu.fr
    ciclad8.ipsl.jussieu.fr
    ciclad7.ipsl.jussieu.fr
    Epilogue Args:
    [...]
    Resource List: cput=01:00:00,neednodes=4:ppn=1,nodes=4:ppn=1
    Resources Used: cput=00:00:00,mem=4948kb,vmem=260216kb,walltime=00:00:14
    Queue Name: short
    Running Host: ciclad2.ipsl.jussieu.fr

Mpi library

- There is many existing mpi implementation
  - Only openmpi is installed on ciclad (no mpitch)
  - for openmpi 1.4.x and 1.6.x are compiled with infiniband supports
- [root@ciclad-ng ~]# module avail openmpi

  ---------- /etc/modulefiles/Libraries ----------

  openmpi/1.4.5-gfortran openmpi/1.4.5-pgf2013 openmpi/1.6.5-pgf2011

  openmpi/1.4.5-ifort openmpi/1.4.5-pgfgcc openmpi/1.6.5-pgf2013

  openmpi/1.4.5-pgf2011 openmpi/1.6.5-gfortran openmpi/1.6.5-pgfgcc

  openmpi/1.4.5-pgf2011gcc openmpi/1.6.5-ifort

- Example : using pgi fortran 2011 compiler and openmpi 1.6.5

  module load pgi/2011 openmpi/1.6.5-pgf2011

FAQ

- I can't connect to ciclad.ipsl.jussieu.fr

  1 if it's first time perhaps ssh problem or files protection on your home

  2 try to log on ciclad2.ipsl.jussieu.fr

  3 network problem , firewall

  4 problem on ciclad (electricity , climatisation, other)

- I can't save my file on /home

  1 save it in another filesystem, place (/tmp ...)

  2 look on your quota using the quota command

- I submit a job and it's stay in queue ?
  - showq -b can help you
- By error I have removed files in /data
  - **SORRY NO BACKUP**
- Same jobs works sometime and sometime not
  - look on output file to see on which node they run when they works and which node when it's not working (could be a problem on one node , hardware , filesystem full or missing library
  - In case of submission of the problem to svp-ciclad@ipsl.jussieu.fr thanks to give us job number , place of script launched and also place of output of your jobs (without this, we can't do something)

Ciclad evolution ...

- Limits :
  - No more than 32 nodes
    - network infrastructure
    - Human administration
  - Bigger data space
    - budget dependant ...
- Perhaps your project could help us