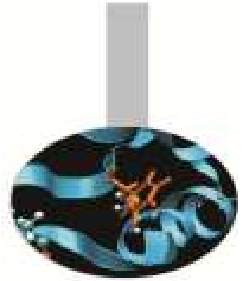


Production environment on FERMI

silvia.giuliani@cineca.it

i.baccarelli@cineca.it

USER programming space

- HOME

>cd $HOME

/fermi/home/userexternal/….

  - 50 GB **quota**

    >cindata (check your space usage)

  - **backup** active on $HOME

USER production space

- SCRATCH

>cd $CINECA_SCRATCH

/gpfs/scratch/userexternal/….

  - No **quota**

    >cindata (check your space usage)

  - No **backup**
  - **Cleaning** procedure (everyday the cleaning procedure deletes all files older than 30 days) → IMPLEMENTED on April 3rd, 2014

PROJECT production space

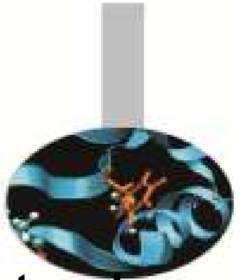
- WORK

```
>cd $WORK
/gpfs/scratch/userexternal/….
```

  - 1 TB **default quota**

    >cindata (check your space usage)
  - No **backup**
  - Data are **preserved up to** the end of the project
  - By **default, files are private**. The user can change the permission (chmod) and make files visible (R o R/W) to project collaborators

SuperComputing Applications and Innovation



MODULES

List of modules of available applications, compilers, tools and libraries

```
>module av
------------ /cineca/prod/modulefiles/base/libraries -------------
------------ /cineca/prod/modulefiles/base/compilers ------------
------------ /cineca/prod/modulefiles/base/tools ------------
------------ /cineca/prod/modulefiles/base/applications ------
abinit/6.12.3            crystal09/2.0.1(default)   octopus/4.1(default)
amber/12(default)        dl_poly/4.03               openfoam/2.1.1
bigdft/1.6.0             dl_poly/4.05(default)      qe/5.0.3b(default)
cp2k/2.3                 gromacs/4.5.5(default)     qe/5.0bgq
cp2k/2.4(default)        gromacs/4.6.1              siesta/3.1
cpmd/3.15.3_rev2606      lammps/20120816            siesta/3.1-TS
cpmd/3.17.1(default)     namd/2.9(default)          vasp/5.2.12
crystal09/1.01           nwchem/6.3(default)        vasp/5.3.3(default)
```

MODULES

Application module **HELP** (binaries compiled for back-end or front-end nodes, how to run them…)

```
>module help <module_name>
```

Application module **LOAD**

```
>module load <module_name>
```

Application variables **SHOW**

```
>module show <module_name>
```

PROFILES

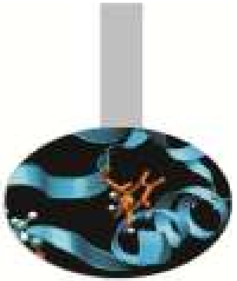
------------ /cineca/prod/modulefiles/**profiles** ------------

profile/advanced   profile/base(default) profile/front-end

- profile/**base (<u>default</u>)**. It contains modules compiled for back-end nodes
- profile/**front-end**. It contains modules compiled for front-end nodes (“front-end-” prefix)
- profile/**advanced**. Experimental profile. It contains also modules to be tested.

```
> module load profile/<profile_name>
> module av
```

EXECUTION



- On front-end and back-end nodes

- Via command line (on front-end only)

```
>./myexe
```

- Via batch

```
>llsubmit job.cmd
```

EXECUTION
Front End nodes

- **Pre and Post processing**
- **Data transfer**

- **Serial** execution (1 core)
- Executables compiled with serial **FE compilers**
  front-end-gnu/4.4.6 front-end-xl/1.0
- **Command line** execution (10 min)
- **Batch** execution (up to 6 h) -> queueing system

BATCH EXECUTION
Front End nodes

- USER'S EXECUTABLES

```
>edit job.cmd
```

- **Shell** interpreter invocation

  ```
  #!/bin/bash
  ```

- **Load Leveler (LL) Scheduler** Keywords

  ```
  # @
  # @
  # @
  …………..
  ```

- **Variables** inizialization

  ```
  export var1=
  ```

- **Execution** line

  ```
  ./myexe <options>
  ```

BATCH EXECUTION Front End nodes

- MODULE EXECUTABLE

>edit job.cmd

- **Shell** interpreter invocation
- #!/bin/bash
- **Load Leveler (LL) Scheduler** Keywords

  ```
  # @
  # @
  # @
  .............
  ```

- **Variables** inizialization
- **Modules** load

  ```
  module load profile/front-end
  module load <module_name>
  ```

- **Execution** line

  ```
  exe <options>
  ```

LL KEYWORDS Front End nodes

```
# @ job_name = serial.$(jobid)
# @ output = $(job_name).out
# @ error = $(job_name).err
# @ wall_clock_limit = 00:00:00 # hh:mm:ss
execution time up to 6 hours
# @ class = serial
# @ resources = ConsumableMemory (count units)
# from 2 GB (default) to 4 GB
# @ account_no = <budget_name> # saldo –b
# @ queue # end
```

EXECUTION
Back End nodes

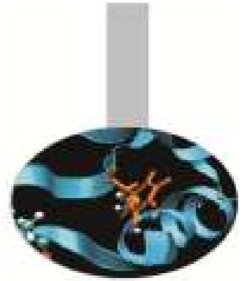
- **Parallel** execution (serial is possible too, but you always allocate 64 nodes)
- Executable compiled with serial and parallel **BE compilers**

```
>bgq-gnu/4.4.6       bgq-xl/1.0
```

- NO **command line** execution
- **Batch** execution (from 64 compute nodes up to 2048 compute nodes, wall clock time up to 24 h)
- **Runjob** command

```
>runjob <options>
>man runjob
```

BATCH EXECUTION Back End nodes



- USER'S EXECUTABLE
  - **Shell** interpreter path

```
#!/bin/bash
```

  - **Load Leveler Scheduler** Keywords

```
# @
# @
# @
…………..
```

  - **Variables** inizialization
  - **Execution** line

```
runjob <runjob_options> : ./myexe
  <myexe_options>
```

BATCH EXECUTION Back End nodes

- MODULE EXECUTABLE

  - **Shell** interpreter path

    #!/bin/bash

  - **Load Leveler Scheduler** Keywords

    # @
    # @
    # @
    ....

  - **Variables** inizialization

    module load <module_name>

  - **Execution** line

    runjob <runjob_options> : $MODULE_HOME/bin/exe <exe_options>
    #check the path of executable by “module show <module_name>”

General LL KEYWORDS Back End nodes

```
# @ job_name = test.$(jobid)
# @ output = $(job_name).out
# @ error = $(job_name).err
# @ environment = COPY_ALL #export all variables from your submission shell
# @ job_type = bluegene
# @ wall_clock_limit = 00:00:00 # hh:hm:ss execution time up to 24 hours
# @ bg_size =  64 # compute nodes number from 64 up to 2048 (64, 128, 256, 512, 1024, 1536, 2048)
# @ notification = always|never|start|complete|error
# @ notify_user = <email_address>
# @ account_no = <budget_name> #saldo –b
# @ queue #end
```

RUNJOB OPTIONS Back End nodes

--exe Path name for the executable to run

```
runjob --exe <exe_name>
```

--args Arguments for the executable specified by --exe

```
runjob --exe <exe_name> --args <option1> --args <option2>
```

RUNJOB OPTIONS Back End nodes

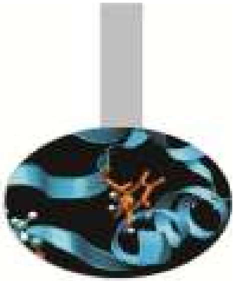
--ranks-per-node Number of ranks (MPI task) per compute node. Valid values are 1 (default), 2, 4, 8, 16, 32 and 64 → SMT

```
bg_size = 64
runjob --ranks-per-node 1 : ./exe <options>
```

-n Number of ranks (MPI task) in the entire job

```
bg_size = 64
runjob -n 64 --ranks-per-node 1: ./exe <options>
#serial job:
runjob -n 1 --ranks-per-node 1: ./exe <options>
```

RUNJOB OPTIONS Back End nodes



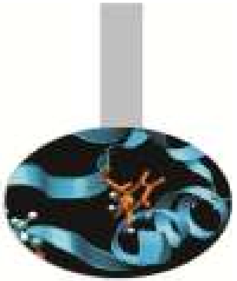
--envs Sets the environment variable to export from the current environment to the compute nodes

```
  bg_size = 64
#MPI/OpenMP job (foreach MPI task 16 threads)
  runjob -n 128 --ranks-per-node 2 --envs
OMP_NUM_THREADS = 16 : ./exe <options>
```

--exp-env Exports an environment variable from the current environment to the compute nodes

```
bg_size = 64
export OMP_NUM_THREADS = 16
  runjob -n 64 --ranks-per-node 1   --exp-env
OMP_NUM_THREADS : ./exe <options>
```

Blue Gene LL KEYWORDS Back End nodes



#@ bg_size = number of compute nodes

# @ bg_shape = MP(A)xMP(B)xMP(C)xMP(D)#midplanes number in the A,B,C,D dimensions

# @ bg_rotate = true|false

# @ bg_connectivity = torus|mesh #type of connectivity

Bg size and connectivity

# @ bg_size = number of compute nodes

# @ bg_connectivity = Mesh #default

- **for requests <= 1midplane** (512 compute nodes)

  bg_size = 64| 128| 256| 512

- **for requests > 1midplane**

  bg_size = (512)*2 | (512)*3 | (512)*4

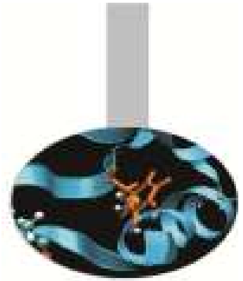
Bg size and connectivity

# @ bg_size = number of compute nodes

# @ bg_connectivity = Torus

- **for requests >= 1MP**

  bg_size = 512 | (512)*2 | (512)*4



Shape

# @ bg_shape = distribution of midplanes on A, B, C, D directions

MP(**A**)*MP(**B**)*MP(**C**)*MP(**D**)

Fermi machine (20 midplanes): **1X5X2X2**

– **for requests >= 1MP**

The values of A, B, C, and D must not be greater than the corresponding A, B, C, and D sizes of the FERMI machine, otherwise, the job will never be able to start

Shape and Connectivity

# @ bg_connectivity = Mesh #default

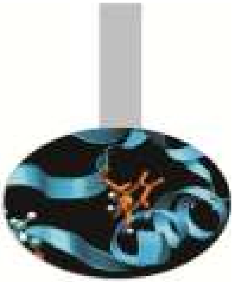
bg_size	→	bg_shape
512		**1x1x1x1**
512*2		**1x1x1x2**
		1x1x2x1
		1x2x1x1
512*3		**1x3x1x1**
512*4		**1x1x2x2**
		1x2x1x2
		1x2x2x1
		1X4X1X1

Shape and Connectivity

# @ bg_connectivity = Torus

bg_size	→	bg_shape
512		1x1x1x1
512*2		1x1x1x2
		1x1x2x1
		~~1x2x1x1~~ #No torus
512*4		1x1x2x2
		~~1x2x1x2~~ #No torus
		~~1x2x2x1~~ #No torus
		~~1X4X1X1~~#No torus

Blue Gene LL KEYWORDS Examples



EXAMPLE

4 midplanes #@bg_size = 2048

#@connectivity = Mesh

⇩

1X2X2X1

1X2X1X2

1X1X2X2

Blue Gene LL KEYWORDS Examples

4 midplanes

#@bg_size = 2048

#@connectivity = Torus

⇩

1X1X2X2

Blue Gene LL KEYWORDS Examples

4 midplanes # @bg_size = 2048

# @ bg_connectivity = Mesh

# @ bg_shape = 1X1X2X2

By default # @ bg_rotate = true. The scheduler should consider all possible rotations of the given shape

⇩

1X1X2X2

1X2X1X2

1X2X2X1

LL COMMANDS

llsubmit

llsubmit job.cmd

llq

llq **-u** $USER

```
[sgiulian@fen07 ~]$ llq -u amarani0

Id                    Owner      Submitted   ST PRI Class    Running On
------------------------ ---------- ----------- -- --- ------------ -----
fen04.7334.0  amarani0   9/21 15:11  I    50  parallel

1 job step(s) in query, 1 waiting, 0 pending, 0 running, 0 held, 0 preempted
```

llq **-s** <job_id>

Provides information on why a selected list of jobs remain in the NotQueued, Idle, or Deferred state.

LL COMMANDS
“llq –s” output

```
–   [sgiulian@fen07 ~]$ llq -s fen04.7334.0

–   ===== EVALUATIONS FOR JOB STEP fen04.fermi.cineca.it.7334.0 =====

–   Step state                    : Idle
–   Considered for scheduling at    : Mon 24 Sep 2012 10:31:45 AM CEST
–   Top dog estimated start time    : Tue 25 Sep 2012 08:48:07 AM CEST

–   Minimum initiators needed: 1 per machine, 1 total.
–   8 machines can run at least 1 tasks per machine, 128 tasks total.
–   Not enough resources to start now.
–   Shape 1x1x1x4 does not fit machine 1x5x2x2.
–   Shape 1x1x4x1 does not fit machine 1x5x2x2.
–   Shape 4x1x1x1 does not fit machine 1x5x2x2.
–   Shape 2x1x1x2 does not fit machine 1x5x2x2.
–   Shape 2x1x2x1 does not fit machine 1x5x2x2.
–   Shape 2x2x1x1 does not fit machine 1x5x2x2.
–   MP "R00-M0" is busy.
–   MP "R00-M1" is busy.
–   MP "R01-M0" is busy.
–   MP "R01-M1" is busy.
–   MP "R20-M0" is busy.
–   MP "R20-M1" is busy.
–   MP "R21-M0" is busy.
–   MP "R21-M1" is busy.
–   MP "R40-M0" is busy.
–   MP "R30-M0" is busy.
–   MP "R10-M0" is busy.
–   MP "R41-M0" is busy.
–   MP "R31-M0" cannot be used by job class.
–   MP "R40-M1" is busy.
–   MP "R30-M1" is busy.
–   This step is a top-dog.
```

BG_SIZE = 2048 # 4 MD
BG_CONNECTIVITY = MESH

The job is a top dog.

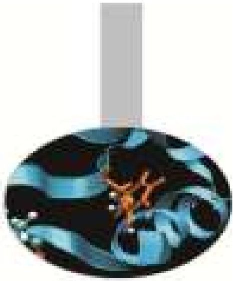
LL COMMANDS
“llq –s” output

```
[sgiulian@fen07 proveMPI]$ llq -s fen03.7942.0

===== EVALUATIONS FOR JOB STEP fen03.fermi.cineca.it.7942.0 =====

Step state                    : Idle
Considered for scheduling at  : Tue 25 Sep 2012 09:52:23 AM CEST

Minimum initiators needed: 1 per machine, 1 total.
8 machines can run at least 1 tasks per machine, 128 tasks total.
Not enough resources to start now.
Shape 2x1x1x1 does not fit machine 1x5x2x2.
MP "R00-M0" is busy.
MP "R01-M0" is busy.
MP "R20-M0" is busy.
MP "R21-M0" is on drain list.
MP "R40-M0" is not AVAILABLE (state="LoadLeveler Drained").
MP "R41-M0" is busy.
MP "R30-M0" is not AVAILABLE (state="LoadLeveler Drained").
MP "R31-M0" cannot be used by job class.
MP "R10-M0" is busy.
MP "R11-M0" cannot be used by job class.
MP "R00-M1" is busy.
MP "R21-M1" is on drain list.
MP "R40-M1" is not AVAILABLE (state="LoadLeveler Drained").
MP "R30-M1" is not AVAILABLE (state="LoadLeveler Drained").
MP "R10-M1" is busy.
MP "R01-M1" is busy.
MP "R41-M1" is busy.
MP "R31-M1" cannot be used by job class.
Not enough resources for this step to be backfilled.
This step can not become a top-dog. Global MAX_TOP_DOGS limit of 1 reached.
```

BG_SIZE =1024 # 2 MD
BG_CONNECTIVITY = MESH

The job is not a top dog and it can not be backfilled.

LL COMMANDS
“llq –s” output

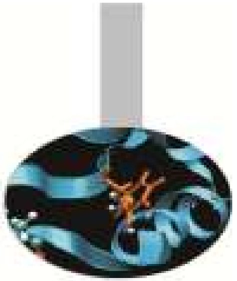


```
•   [sgiulian@fen07 proveMPI]$ llq -s fen04.7546.0

•   ===== EVALUATIONS FOR JOB STEP fen04.fermi.cineca.it.7546.0 =====

•   Step state                    : Idle
•   Considered for scheduling at     : Mon 24 Sep 2012 01:56:00 PM CEST

•   Minimum initiators needed: 1 per machine, 1 total.
•   8 machines can run at least 1 tasks per machine, 128 tasks total.
•   Not enough resources to start now.
•   Shape 1x1x1x3 does not fit machine 1x5x2x2.
•   Shape 1x1x3x1 does not fit machine 1x5x2x2.
•   Shape 3x1x1x1 does not fit machine 1x5x2x2.
•   MP "R00-M0" is busy.
•   MP "R00-M1" is busy.
•   MP "R01-M0" is busy.
•   MP "R01-M1" is busy.
•   MP "R20-M0" is busy.
•   MP "R20-M1" is busy.
•   MP "R21-M0" is busy.
•   MP "R21-M1" is busy.
•   MP "R40-M0" is busy.
•   MP "R41-M0" is busy.
•   Not enough resources for this step as top-dog.
•   Shape 1x1x1x3 does not fit machine 1x5x2x2.
•   Shape 1x1x3x1 does not fit machine 1x5x2x2.
•   Shape 3x1x1x1 does not fit machine 1x5x2x2.
•   MP "R00-M0" is busy.
•   MP "R00-M1" is busy.
•   MP "R01-M0" is busy.
•   MP "R01-M1" is busy.
•   MP "R20-M0" is busy.
•   MP "R20-M1" is busy.
    MP "R21-M0" is busy.
```

BG_SIZE = 1536 # 3 MD
BG_CONNECTIVITY = TORUS

The job will not start. It’s not possible to have the TORUS connection for all directions.

LL COMMANDS
“llq –l” output



llq -l <job_id>

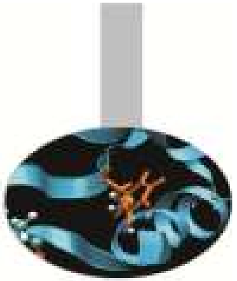
- Specifies that a long listing will be generated for each job for which status is requested.
- In particular you'll be notified about the bgsize you requested and the real bgsize allocated:

```
   Queue Date: Thu 06 Mar 2014 08:42:51 AM CET
Eligibility Time: Mon 10 Mar 2014 07:52:29 AM CET
Dispatch Time: Mon 10 Mar 2014 06:52:17 PM CET

  …………………………………...
BG Size Requested: 1024
BG Size Allocated: 1024
BG Shape Requested:
BG Shape Allocated: 1x1x1x2
BG Connectivity Requested: Mesh
BG Connectivity Allocated: Torus Torus Torus Torus
……………………………………..
```

llcancel

llcancel <job_id>

QUEUES
BE and FE nodes



- Serial (FE nodes)
  - DATA PROCESSING and TRANSFER - (1 core, up to 6 h)
- Debug (BE nodes)
  - TEST Short time - (64 compute nodes, up to 30 min)
- Longdebug (BE nodes)
  - TEST Long time - (64 compute nodes, from 31 min up to 24 h)
- Smallpar (BE nodes)
  - PRODUCTION - (128 compute nodes, up to 24 h)
- Parallel (BE nodes)
  - PRODUCTION - (from 256 to 512 compute nodes, up to 24 h)
- Bigpar (BE nodes)
  - PRODUCTION - (from 1024 to 2048 compute nodes, up to 24 h)
- Keyproject (BE nodes)
  - Very parallel jobs (authorized from the user support superc@cineca.it)

```
# @ wall_clock_limit = up to 6 h
# @ resources = ConsumableMemory (2 GB) # From 2 GB (default) to 4 GB
# @ class = serial
```

```
# @ job_type = bluegene
# @ wall_clock_limit = up to 24 h
# @ bg_size = from 64 to 2048 nodes
# @ class = keyproject  #For bg_size
    > 2048 (upon authorization):
```

SCHEDULER
JOB State

- Queueing state. The job has been submitted (queue time) and has been scheduled to start (**elegibility date**)
  - I: job is in the idle state
  - R: job is in the running state (dispatch time)
- Not queueing state. The job has been submitted (queue time), but it has not been scheduled to start (**no elegibility date**)
  - NQ: job is in the not queueing state. This is the state of a single step (multistep job) or a job whose user has already reached its “max queued jobs number” available for the specific queue

SCHEDULER
JOB State

user “max queued jobs number” **debug: 1**

user “max queued jobs number” **longdebug: 2**

user “max queued jobs number” **smallpar: 4**

user “max queued jobs number” **parallel: 2**

user “max queued jobs number” **bigpar: 2**

– H: job is in hold state. The user can place and release its job into and from this state by using llhold command in order not to schedule the job

SUPERC MODULE

```
>module help superc
```

bgtop (draws a full-terminal display of nodeboards and jobs)

topdog (shows the jobs that are the current top-dogs)

jobtyp (provides useful information about job in the LL queues - user, tasks, times, ...)

sstat (provides useful information about the system status - jobs in the LL queues, allocated nodes, ...

sstat2 (provides a more complete information about the system status - Midplane avail/down/drained, jobs in the LL queues, allocated nodes, ...

…………

```
>module load superc
>bgtop
```

Advanced jobs

- MULTISTEP JOBS

  LoadLeveler scheduler allows to chain many jobs in a single multi-step job

  - BE nodes steps
    - User guide link
  - FE nodes and BE nodes steps
    - data processing (BE) and data transfer (FE)
    - User guide

Advanced jobs

- SUB BLOCK JOBS
  - It is possible to lunch multiple runs in the minimum allocatable block of 64 compute nodes. Sub-blocking tecnhique enables you to submit jobs in which 2, 4, 8, 16, 32, or 64 simulations are simultaneously running, each occupying 32, 16, 8, 4, 2, 1 compute nodes, respectively
  - User guide link

SALDO

saldo -b

Prints budgets info for your username:

- validity ranges
- consumed resources both on the local cluster and on all clusters
- percentage of consumed resources

```
------------------------------------------------------------------------------------------------------------------
account           start       end        total       localCluster            totConsumed      totConsumed
                                       (local h)   Consumed(local h)   (local h)            %
------------------------------------------------------------------------------------------------------------------
```

SALDO

saldo -r

Prints daily resources usage report on the local cluster for

- selected username (-u)

```
>saldo –r –u <user_name>
```

- selected account (-a)

```
>saldo –r –a <account_name>
```

```
-----------------Resources used from 201101 to 201212--------------

date       username    account          localCluster       num.jobs
                                              Consumed/h
-----------------------------------------------------------------------------
```

CONSUMED RESOURCES

- Remember that you are consuming the ALLOCATED resources and not necessarily the REQUESTED resources

(allocated compute nodes)*(16cores)*(execution time)

ARCHIVING SPACE

- CART

>cart_dir

  - **long-term** storage
  - **500 GB default quota**
  - **upon authorization (**contact our HPC support superc@cineca.it)**)**
  - **cart commands**
  - **user guide link**