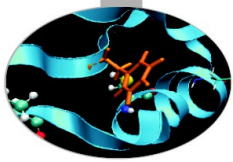
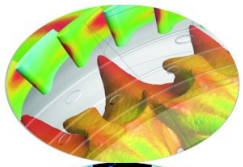
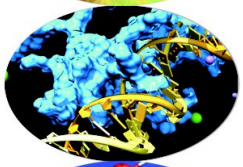
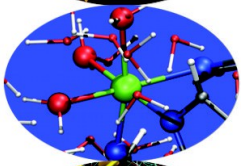
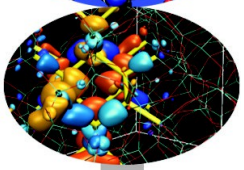
CINECA SCAI
SuperComputing Applications and Innovation



Knights Landing configuration and production environment on MARCONI

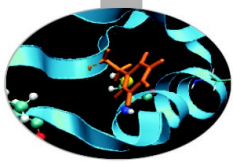





Alessandro Marani - a.marani@cineca.it

April 21st, 2017

CINECA

Agenda



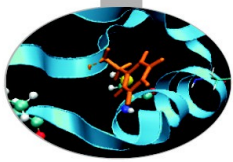
In this presentation, we will discuss…

- How we interact with KNL environment on MARCONI
- How to navigate on the new module system
- How to compile for KNL and how to submit a KNL job
- Accounting and budget linearization
- Miscellanea and documentation

DISCLAIMER: This presentation assumes that you are familiar with general HPC environment at CINECA and focuses on what is specific for MARCONI-KNL.

You can refer to our userguides (links in the last slide) for a basic assistance on our environment.

Before KNL: login on MARCONI



Login: **ssh <username>@login.marconi.cineca.it**

At login, you will be prompted with our "**Motto of the Day**" with the technical detail of the cluster and latest news from User Support

WARNING: you may sometimes find an unstable situation on login (like inability to see the filesystems). In such cases, it doesn't necessarily mean that the entire front-end is affected, try switching login nodes by being specific among which of the three you want to use:

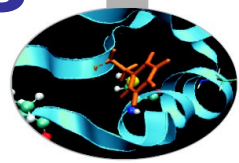
```
********************************************************************************
* Welcome to MARCONI /                                                         *
*            MARCONI-fusion @ CINECA - NeXtScale cluster - CentOS 7.2!         *
*                                                                              *
* Broadwell partition - 1512 Compute nodes:                                    *
*    - 2 18-core Intel(R) Xeon(R) E5-2697 v4 @ 2.30GHz per Compute node        *
*    - 128 GB RAM per Compute node                                             *
* KNL partition - 3600 Compute nodes:                                          *
*    - 1 68-core Intel(R) Knights Landing @ 1.40GHz per Compute node           *
*    - 16 GB MCDRAM per Compute node                                           *
*    - 93 GB RAM per Compute node                                              *
* Intel OmniPath (100Gb/s) high-performance network                            *
* PBSpro 13 batch scheduler                                                    *
*                                                                              *
* wiki.u-gov.it/confluence/display/SCAIUS/UG3.1%3A+MARCONI+UserGuide            *
* for a guide on Marconi                                                       *
*                                                                              *
* mailto: superc@cineca.it for support                                         *
********************************************************************************
* This system is in full-production. *
================================================================================
IMPORTANT:
- A new version of "module" is installed. Applications are available through
  domain-based profiles to be loaded before the module itself. Use the "modmap"
  command to identify the correct profile ("modmap -h" for help).
- Marconi is little-endian (like GALILEO and PICO), in contrast with FERMI
  which was big-endian.
- Daily cleaning of the scratch area is not active yet.
- Load the module "env-knl" to switch from Broadwell to KNL environment.
  Unload it or load "env-bdw" to revert to Broadwell environment.
================================================================================
  Since March 15th accounting is enabled on KNL partition. The computing hours
  spent with your jobs will be now detracted from your budget, as it already
  happens for Broadwell and other CINECA HPC clusters.
================================================================================
```

ssh <username>@login01.marconi.cineca.it

ssh <username>@login02.marconi.cineca.it

ssh <username>@login03.marconi.cineca.it

Before KNL: filesystems on MARCONI



At login, you will find yourself into your “Home” space ($HOME). It is a permanent back-upped space with a 50 Gb quota, suited for compilation and small debugging.

For production, move into $CINECA_SCRATCH or $WORK filesystems. They are large, parallel filesystems suited for intensive I/O activity:

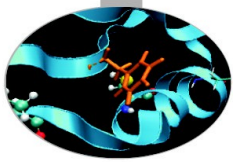
-$CINECA_SCRATCH: personal area, with no backup and no quota. Cleaning procedure on scratch is not active yet (will be on April 27th).

-$WORK: area shared with all the collaborators of an account (i.e. project). It is not backupped and has a quota of 1 Tb (extendable upon request to User Support)

If you work on different projects, use the command “chprj” to switch between Work folders. It modifies your $WORK environment variable accordingly

CINECA SCAI
SuperComputing Applications and Innovation

Before KNL: filesystems on MARCONI



It is possible to make a request for a DRES (Data Resource), a resource meant for storage purposes. DRES can be of three types: filesystem (high throughput disks), archive (magnetic tape) or repo (smart repository based on iRODS).

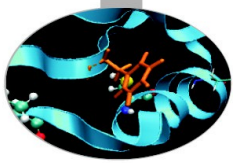
All data stored in regular filesystems and DRES will last for 6 months after the end of the project.

The command “cindata” can help you to keep track of the disk occupancy of your areas:

```
[amarani0@r000u07l02 ~]$ cindata -u mbaldi00
```

USER	AREADESCR	AREAID	FRESH	USED	QTA	USED%	aUSED	aQTA	aUSED%
mbaldi00	/marconi_work/	marconi_work	2hou	0	--	--%	444T	7.0P	6.2%
mbaldi00	/marconi_work/IscrB_SIMCODE1	marconi_work-IscrB_SIMCODE1	2hou	0	--	--%	0	1T	0.0%
mbaldi00	/marconi_work/IscrC_P3SIMCD	marconi_work-IscrC_P3SIMCD	2hou	0	--	--%	119G	1T	11.7%
mbaldi00	/marconi/	marconi_hpc	3hou	1.6G	--	--%	8.0T	200T	4.0%
mbaldi00	/marconi/home	marconi_hpc-home	3hou	1.6G	50G	3.2%	4.4T	--	--%
mbaldi00	/marconi_scratch/	marconi_scr	3hou	651G	--	--%	1.6P	2.5P	65.0%
mbaldi00	/gss/gss_work/DRES_SIMCD	work_OFFLINE-DRES_SIMCD-FS	5hou	8.4T	--	--%	8.4T	195T	4.3%
mbaldi00	/gss/gss_work/	work_OFFLINE	5hou	8.4T	--	--%	719T	--	--%
mbaldi00	/gss/gss_work/DRES_SIMCD	work_ONLINE-DRES_SIMCD-FS	5hou	34T	--	--%	44T	186T	23.6%
mbaldi00	/gss/gss_work/	work_ONLINE	5hou	34T	--	--%	978T	1.4P	70.3%

KNL environment: module env-knl



When you login on MARCONI, you'll find yourself in an environment studied for work with Broadwell partition.

Your jobs will be submitted on Broadwell nodes and other commands such as *qstat* display only this side of the cluster.

In order to move on KNL environment, you need to load a proper module:

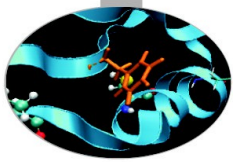
module load env-knl

Then everything will be set for working on the new partition, and your jobs will we submitted on KNL nodes. To return on Broadwell, either unload the module or load:

module load env-bdw

CINECA SCAI
SuperComputing Applications and Innovation

KNL environment: module env-knl



An example: qstat -Q (list of all the available queues on a partition)

On Broadwell:

```
[amarani0@r000u07102 ~]$ module load env-bdw
(BDW) [amarani0@r000u07102 ~]$ qstat -Q
Queue              Max   Tot Ena Str   Que   Run   Hld   Wat   Trn   Ext Type
---------------- ----- ----- --- --- ----- ----- ----- ----- ----- ----- ----
xfualongprod         0   217 yes yes   177    40     0     0     0     0 Exec
xfuaprod             0   132 yes yes   118     6     8     0     0     0 Exec
xfuadebug            0     2 yes yes     1     1     0     0     0     0 Exec
xfuabigprod          0     6 yes yes     2     3     1     0     0     0 Exec
xfualowprio          0     0 yes yes     0     0     0     0     0     0 Exec
test                 0     0 yes yes     0     0     0     0     0     0 Exec
serial               0     1 yes yes     0     0     1     0     0     0 Exec
system               0     0 yes yes     0     0     0     0     0     0 Exec
route                0     0 yes yes     0     0     0     0     0     0 Rout
meteopar             0     9 yes yes     5     4     0     0     0     0 Exec
meteoser             0     1 yes yes     0     1     0     0     0     0 Exec
special              0     0  no yes     0     0     0     0     0     0 Exec
debug                0     7 yes yes     1     6     0     0     0     0 Exec
xfuagwdebug          0     0 yes yes     0     0     0     0     0     0 Exec
quarantine           0     0 yes yes     0     0     0     0     0     0 Exec
prod                 0  2203 yes yes   951   256   994     0     0     0 Exec
admin                0     0 yes yes     0     0     0     0     0     0 Exec
xfuaspecial          0     0  no yes     0     0     0     0     0     0 Exec
bigprod              0    21 yes yes    19     1     1     0     0     0 Exec
xfuagw               0     2 yes yes     0     2     0     0     0     0 Exec
(BDW) [amarani0@r000u07102 ~]$
```

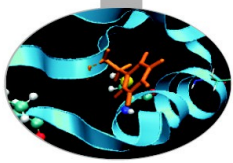
On KNL:

```
[amarani0@r000u07102 ~]$ module load env-knl
(KNL) [amarani0@r000u07102 ~]$ qstat -Q
Queue              Max   Tot Ena Str   Que   Run   Hld   Wat   Trn   Ext Type
---------------- ----- ----- --- --- ----- ----- ----- ----- ----- ----- ----
knlroute             0     0 yes yes     0     0     0     0     0     0 Rout
knlprod              0   336 yes yes   115   203    18     0     0     0 Exec
knlsystem            0     0 yes yes     0     0     0     0     0     0 Exec
knldebug             0     5 yes yes     1     1     3     0     0     0 Exec
xfuaknlprod          0    33 yes yes    27     6     0     0     0     0 Exec
knlquarantine        0     0 yes yes     0     0     0     0     0     0 Exec
knladmin             0     0 yes  no     0     0     0     0     0     0 Exec
knltest              0     1 yes yes     0     1     0     0     0     0 Exec
(KNL) [amarani0@r000u07102 ~]$
```

First rule for KNL: when you want to work with KNL environment, first thing to do is to load the env-knl module!

CINECA

Speaking of modules...



Since the beginning of MARCONI, a new module system has been implemented. Modulefiles are now divided in **profiles**, and you have to load the proper profile in order to access to their modules (module load profile/profilename).

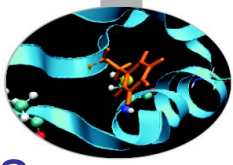
Profiles currently defined are the following:

- **profile/base** (default): all serial and parallel compilers, most common libraries (Intel or Python compiled), most common tools for debugging and profiling
- **profile/advanced**: libraries compiled with Gnu, Openmpi-intel, mvapich2 and other “less recommended” modules
- **profile/archive**: older versions of updated modules, for retrocompatibility
- **domain profiles**: here you can find all the application softwares. They are divided in profiles based on scientific domains.

  Current available domain profiles are:

  - **profile/astro**
  - **profile/bioinf**
  - **profile/chem**
  - **profile/lifesc**
  - **profile/phys**

An useful command: modmap



"modmap" is an useful tool for navigate in our modules environment. It lets you know which profile you have to load in order to find a specific module

Usage examples:

modmap -m <modulename>

to know where to find a specific module

```
[amarani0@r000u07l02 ~]$ modmap -m namd
Profile: advanced
Profile: archive
                namd
                 2.11
Profile: astro
Profile: base
Profile: bioinf
Profile: chem
                namd
                 2.12
Profile: knl
                namd
                 2.12_knl
Profile: lifesc
                namd
                 2.12
Profile: phys
                namd
                 2.11
                 2.12
```

modmap -p <profilename>

to get the list of modules contained in a specific profile

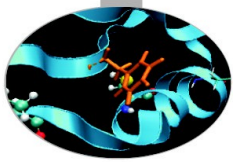
modmap -c <categoryname>

to get the list of all the modules in a specific category (tools, libraries,…), divided by profile

modmap -h

to get an help about the command usage

Module environment and KNL



For KNL applications, a specific **profile/knl** is available.

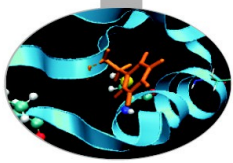
KNL modules are also identified by an "**_knl**" in their name.

```
[amarani0@r000u08l03 ~]$ modmap -p knl
          applications
                  amber
                   16.0_knl
                  cp2k
                   4.1_knl
                  cpmd
                   4.17_knl
                  gromacs
                   2016.1_knl
                  lammps
                   17nov2016_knl
                  namd
                   2.12_knl
                  qe
                   6.0_knl
                  siesta
                   4.1-b2_knl
                  vasp
                   5.4.1_knl
          compilers
          data
          environment
          libraries
          tools
```

Currently, fifteen application modules are available for KNL usage.

This configuration is still under development, there is work in progress for adding more KNL applications and libraries to the profile. For the time being, what is listed in regular profiles but not in profile/knl is to be considered the correct choice for both environments (although it may not be optimized for Knights Landing)

Compiling for KNL



While regularly compiled applications can run on KNL, performance may not be as good as you expected.

To exploit the benefits of Knight Landing vectorization, add to your compiling line (assuming you are using Intel compiler suite) the following flag:

mpiicc **-xMIC-AVX512** -o myexe mycode.c

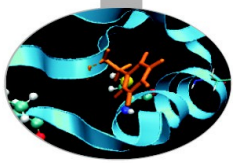
This will generate AVX-512 instructions to derive better performance from these nodes. However, the application compiled this way will not run on Broadwell. To generate a portable, vectorized application use:

mpiicc **-axMIC-AVX512** -o myexe mycode.c

Intel recommends that you keep two separate binaries, for the two partitions

Please check this guide for more tips about exploiting the vectorization benefits:
https://wiki.u-gov.it/confluence/display/SCAIUS/How+to+Improve+Code+Vectorization

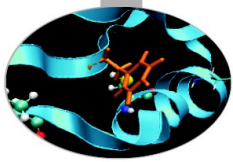
Submitting a job on KNL nodes



```
#!/bin/bash
#PBS -n jobname
#PBS -e job.err
#PBS -o job.out
#PBS -l walltime=24:00:00      #maximum walltime requirable
#PBS -l select=5:ncpus=68:mpiprocs=68:mem=93GB
#PBS -A <account_no>

cd $PBS_O_WORKDIR
module load autoload intelmpi/2017--binary
mpirun -n 340 ./myexe
```

Let's take a moment to discuss the resources you can ask!

Submitting a job on KNL nodes



- select=…

You can ask up to **1000** nodes on KNL partition

- ncpus=…

Since a KNL node has **68** cores, that is the maximum number to put in this entry (also the default, since KNL is node-partitioned)

- mpiprocs=…

Hyper-threading is active on KNL. Each physical core can behave as 4 virtual cores. So you can ask for up to **272** mpirocs!

- mem=…

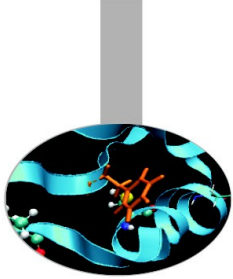
Every node is in cache mode, so you can ask for up to **93GB** of memory per node (also the default, since KNL is node-partitioned)

- numa,mcdram=…

Do **NOT** specify them, as every node is defaulted to quadrant/cache and such configuration can't be changed

```
#PBS -l select=1000:ncpus=68:mpiprocs=272:mem=93GB   # maximum
```

Queues for KNL



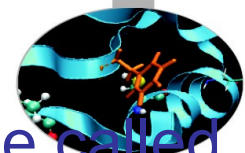
As it is now common in our HPC systems, the queue has **not** to be specified. PBS will decide it depending on the amount of resources you are asking.

On MARCONI-KNL, there are two possible queues you can end up:

- **knldebug**: 2 racks are reserved for debugging and small production, you will access them if you ask for less than **2 nodes** and **30 minutes** in your jobscript

- **knlprod**: jobs requiring higher resources will end up in regular, production queue and compete with all the other production jobs for the resources

Queues for KNL



In addition to regular queues, there is also a special queue called "**knltest**". It points to two racks, one is cache/quadrant and the other is flat/quadrant.

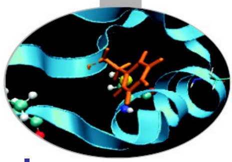
You have to ask to superc@cineca.it to be authorized to access the queue. After that, you have to specify its usage on the jobscript:
#PBS -q knltest
#PBS -W group_list=<account_name>

You can ask up to **72** nodes with it. To access to flat partition, add to your request line:
#PBS -l select=4:ncpus=68:mpiprocs=68:**mcdram=flat**:mem=108GB

Note that with flat equipped you can ask up to **108GB** per node

NOTE: knltest is a queue meant for testing and development only! It is not a queue suited for production!

Job submission



If you have loaded env-knl module, you can submit your job as usual, with “qsub <jobscript>”, and check its status with “qstat”.

Warning: “qstat -u $USER” doesn’t return the full jobid!

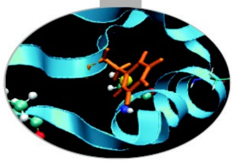
Some characters may be cut, and if you copy/paste what you see, results may be unexpected.

```
(KNL) [amarani0@r000u08103 ~]$ qstat -u $USER

r064u06s01:
                                                            Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
81241.r064u06s0 amarani0 knlprod  simpleknl     --   20 136 1860gb 04:00 Q   --
(KNL) [amarani0@r000u08103 ~]$ qstat -f 81241.r064u06s0
qstat: illegally formed job identifier: 81241.r064u06s0
```

“qstat -w -u $USER” solves the problem

```
(KNL) [amarani0@r000u08103 ~]$ qstat -w -u $USER

r064u06s01:
                                                                                                            Req'd  Req'd   Elap
Job ID                         Username        Queue           Jobname         SessID   NDS  TSK   Memory Time  S Time
------------------------------ --------------- --------------- --------------- -------- ---- ----- ------ ----- - -----
81241.r064u06s01               amarani0        knlprod         simpleknl             --   20  1360 1860gb 04:00 R  --
(KNL) [amarani0@r000u08103 ~]$ qstat -f 81241.r064u06s01
Job Id: 81241.r064u06s01
    Job_Name = simpleknl
    Job_Owner = amarani0@r000u08103-hfi.marconi.cineca.it
    resources_used.cpupercent = 490
```

Accounting & KNL



#PBS -A ???

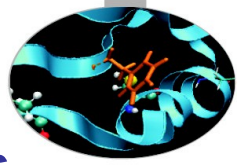
Usually, the command “saldo” is able to display the account name that you have to add to your job in order to let it know from where it has to dectract the cpu hours spent.

However, accounts for MARCONI-BDW are different from the ones for MARCONI-KNL, and saldo is able to display only the former (regardless of the environment module loaded)

Use the option **--knl** to get informations about your KNL account

```
(KNL) [amarani0@r000u08103 ~]$ saldo -b --knl

----------------------------------------Budgets available on Knights Landing--------------------------------------------------------------
account                 start         end         total        localCluster   totConsumed      totConsumed     monthTotal     monthConsumed
                                               (local h)   Consumed(local h)   (local h)                %      (local h)         (local h)
------------------------------------------------------------------------------------------------------------------------------------------
cin_staff            20110323    20200323    1600000016                   0     108775531            6.8       14598540             75093
cin_priorit          20131115    20191231       8000000                   0       8039559          100.5              0                 0
train_scA2017_0      20170213    20170305         12000                   0             0            0.0              0                 0
(KNL) [amarani0@r000u08103 ~]$
```

A quick review about accounting policy



After a period of pre-production, in March 15th, 2017 accounting **has been enabled** for KNL nodes, and your jobs submitted to such partition will be regularly accounted on the proper budget

As it is now common in our HPC environment, a budget linearization policy is active. Each month, a monthly quota will be set for your account, and priority of your jobs will decrease as much as this quota is consumed.

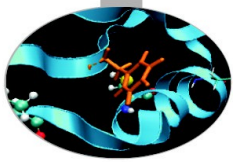
This priority parameter will reach its minimum when the monthly quota is completely spent. After that moment, you will still be able to consume your global budget, but at a reduced priority. At the first day of the month, the situation will reset and you will be able to submit again at full priority, while consuming the new monthly quota.

This is to encourage a linearization of your consumption, and to incentivate fairness in sharing the resources with all other users.

You can check your global and monthly consumption with:

saldo -b --knl

KNL environment: alternative setup



While loading the env-knl module is still our recommended choice of actions, first day users may still prefer the original method of interacting with KNLs, that this slide will briefly review

You can submit your jobs while staying on Broadwell environment, by launching the command like this:

qsub -q knlroute@knl1 jobscript.sh

“knl1” refers to the primary PBS KNL server, you can alternatively use “knl2” (the secondary server) if it doesn’t work.

Other PBS commands change consequently:

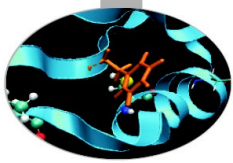
qstat -w -u $USER @knl1 # knl1 has to be written after a space

qdel <jobid>@knl1 # no spaces this time

Using the module sets up automatically all the environment variables for KNL, and you don’t need to remember particular PBS options, thus avoiding confusion.

CINECA SCAI
SuperComputing Applications and Innovation

Useful links and documentation



General userguides related to CINECA's HPC environment

https://wiki.u-gov.it/confluence/display/SCAIUS/UG2.0%3A+General+Information

MARCONI (Broadwell and KNL) specific userguide

https://wiki.u-gov.it/confluence/display/SCAIUS/UG3.1%3A+MARCONI+UserGuide

Informations about PBS Batch Scheduler

https://wiki.u-gov.it/confluence/display/SCAIUS/UG2.5.1%3A+Batch+Scheduler+PBS

Useful tips about improving code vectorization

https://wiki.u-gov.it/confluence/display/SCAIUS/How+to+Improve+Code+Vectorization

Useful e-mails:

superc@cineca.it - Helpdesk, write here for any problem or question related to our HPC environment

corsi@cineca.it - For informations about training activites (courses, schools,…) at CINECA

CINECA