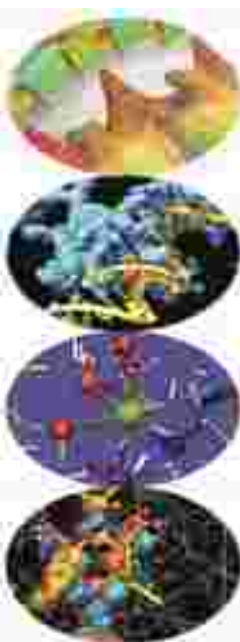




Knights Landing Production Environment on Marconi

HPC User Support @ CINECA

23/10/2017





Agenda

In this presentation, we will discuss:

- 🔑 How to interact with Marconi environment
- 🔑 How to navigate on the new modules system
- 🔑 How to compile for KNL and how to submit a KNL job
- 🔑 Accounting and budget linearization
- 🔑 How to open a graphic session
- 🔑 Data management and storage
- 🔑 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-A2 partition (KNL).

You can refer to our userguides for a basic assistance on our environment.



Marconi environment

When you login to Marconi, you will find yourself in an environment studied for work with Marconi-A1 partition (BDW).

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

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

🔧 **module load env-knl**

Then everything will be set for working on KNL partition and your jobs will be submitted to KNL nodes

🔧 To return to BDW environment, either unload the module or load the proper module:

🔧 **module load env-bdw**

Marconi environment



```
(KNL) [shuenonig@000u06101 ~]$ module load env-knl
(KNL) [shuenonig@000u06101 ~]$ qstat -Q
```

Queue	Max	Tot	Ins	Str	Que	Run	Hld	Wrt	Trn	Ext	Type
special	0	1	yes	yes	0	0	1	0	0	0	Exec
cin_container	0	0	yes	yes	0	0	0	0	0	0	Exec
prod	0	2087	yes	yes	641	342	1102	0	0	1	Exec
metaserv	0	0	yes	yes	0	0	0	0	0	0	Exec
system	0	33	yes	yes	0	0	33	0	0	0	Exec
quarantine	0	1	yes	yes	0	0	1	0	0	0	Exec
admin	0	0	yes	yes	0	0	0	0	0	0	Exec
test	0	0	yes	yes	0	0	0	0	0	0	Exec
gig2	0	0	yes	yes	0	0	0	0	0	0	Exec
metanets	0	0	yes	yes	0	0	0	0	0	0	Exec
route	0	0	yes	yes	0	0	0	0	0	0	Route
debug	0	67	yes	yes	46	8	13	0	0	0	Exec
visualproc	0	0	yes	yes	0	0	0	0	0	0	Exec
metaspar	0	0	yes	yes	0	0	0	0	0	0	Exec
serial	0	1	yes	yes	0	1	0	0	0	0	Exec
bigprod	0	20	yes	yes	2	5	12	0	0	0	Exec
xfuagp	0	18	yes	yes	0	18	0	0	0	0	Exec
xfuagpdebug	0	0	yes	no	0	0	0	0	0	0	Exec

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

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

```
[shuenonig@000u06101 ~]$ module load env-knl
(KNL) [shuenonig@000u06101 ~]$ qstat -Q
```

Queue	Max	Tot	Ins	Str	Que	Run	Hld	Wrt	Trn	Ext	Type
knlsystem	0	179	yes	yes	179	0	0	0	0	0	Exec
kniquarantine	0	0	yes	yes	0	0	0	0	0	0	Exec
knlroute	0	0	yes	yes	0	0	0	0	0	0	Route
knidebug	0	140	yes	yes	141	5	2	0	0	0	Exec
xfuaknidebug	0	1	yes	yes	0	0	1	0	0	0	Exec
xfuaknlprod	0	46	yes	yes	27	17	0	0	0	0	Exec
knlprod	0	468	yes	yes	64	228	174	0	0	1	Exec
knitest	0	0	yes	yes	0	0	0	0	0	0	Exec
knladmin	0	0	yes	yes	0	0	0	0	0	0	Exec

Modules environment



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:

🌳 profile/base [default]

🌳 profile/advanced

🌳 domain profiles:

🌳 astro

🌳 bioinf

🌳 chem

🌳 deeplrn

🌳 eng

🌳 knl

🌳 lifesc

🌳 phys



A 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 <module_name>`
- 🔑 `modmap -p <profile_name>`
- 🔑 `modmap -c <category_name>`
- 🔑 `modmap -h`

```

[shuenomi@r000u06101 ~]$ modmap -m namd
Profile: advanced
Profile: archive
      namd
      2.11
Profile: astro
Profile: base
Profile: bioinf
Profile: chem
      namd
      2.12
Profile: deeplrn
Profile: eng
Profile: knl
      namd
      2.12_knl
Profile: lifecs
Profile: lifesc
      namd
      2.12
      2.12_knl
Profile: phys
      namd
      2.12
  
```



Modules environment for KNL

```
[chancellor@rockwell18] ~]$ module -q knl
application
```

```

  abinit
  8.0.0_knl
  -----
  16.0_knl
  co2k
  4.1_knl
  cymad
  4.1_knl
  dl_gmly
  4.00_knl
  groonga
  2016.1_knl
  lumps
  17nov2016_knl
  10jul2016_knl
  num
  7.12_knl
  nuChem
  8.6_knl
  openfoam
  v1612_knl
  plumed
  2.3.1_knl
  qt
  6.0_knl
  slesta
  4.1-b2_knl
  uasp
  5.4.1_knl
  yambo
  4.1.2_knl
```

```
library
```

```

  dealii
  8.5.0_knl--openmpi--2.10.3--g++--4.1.0
  fftw
  3.3.5_knl--intelmpi--2017--binary
  pSfft
  2.7.5_knl--intelmpi--2017--binary
```

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

KNL modules are identified by a “**_knl**” in their name.

For the time being, what is listed in regular profiles but not in profile/knl is to be considered the correct choice for all 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 Knights 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 BDW or SKL. To generate a portable, vectorized application use:
 - ☛ `mpiicc -axMIC-AVX512 -o myexe mycode.c`
- ☛ Intel recommends that you keep separate binaries, for different Marconi 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=90GB
#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= ...

Maximum value equals total **68** physical cores on a KNL node

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 (not advisable...)

mem= ...

Every node is in **cache** mode, so you can ask for up to **90GB** of memory per node (suggested 86GB)

numa,mcdram= ...

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

#PBS -l select=1000:ncpus=68:mpiprocs=272:mem=90GB



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-A2 (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

Other queues for KNL



- 🔑 In addition to production queues, there is also a special queue called **knltest** meant for testing and development.
- 🔑 You have to ask to **superc@cineca.it** to be authorized to access.
- 🔑 After that, you have to specify its usage on the jobscript
#PBS -q knltest
#PBS -W group_list=<account_name>
- 🔑 To access to flat partition, add to your request line:
#PBS -l select=2:ncpus=68:mpiprocs=68:mcdram=flat:mem=105GB

Jobs 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” does not return the full jobid! Some characters may be cut, and if you copy/paste what you see, results may be unexpected.

```
(KML) [shuenomi@000-07102 ~]$ qstat -u shuenomi
```

```
qstat@06501:
```

Job ID	Username	Queue	Jobname	SessID	NCs	TSK	Req'd Memory	Req'd Time	Elap. S Time
416717.r004-06501	shuenomi	knlpred	STDIN	---	1	68	80gb	01:00	---
416719.r004-06501	shuenomi	knlpred	STDIN	15060	1	68	80gb	01:00	01:00

🔑 “qstat -u \$USER -w” solves the problem!

```
(KML) [shuenomi@000-07102 ~]$ qstat -u shuenomi -w
```

```
qstat@06501:
```

Job ID	Username	Queue	Jobname	SessID	NCs	TSK	Req'd Memory	Req'd Time	Elap. S Time
416717.r004-06501	shuenomi	knlpred	STDIN	---	1	68	80gb	01:00	---
416719.r004-06501	shuenomi	knlpred	STDIN	15060	1	68	80gb	01:00	01:00:48

Accounting



#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 deduct 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 information about your KNL accounts

Use the option **--skl** to get information about your SKL accounts

```
[Knl] [chiomacip-000.02102 ~]$ saldo -n --knl
```

account	start	end	total (local #)	localCluster Consumed(local #)	totConsumed (local #)	totConsumed %	monthTotal (local #)	monthConsumed (local #)
cto_staff	20110125	20200125	1000000000	1777553	129817736	0.1	10100549	813499
cin_priorit	20111115	20191231	150000000	405643	914030	0.1	134579	0
cin_external_W	20150120	20171231	200000	8300	8300	0.2	1007	220

```
[Skylake] [chiomacip-000.02101 ~]$ saldo -n --skl
```

account	start	end	total (local #)	localCluster Consumed(local #)	totConsumed (local #)	totConsumed %	monthTotal (local #)	monthConsumed (local #)
cto_staff	20110125	20200125	1000000000	217029	1111000	0.1	3124007	32007
cto_external_W	20170029	20171231	1000	0	0	0.0	1012	0

Accounting policy



- 🔧 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 the **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`
 - 🔧 `saldo -b --skl`

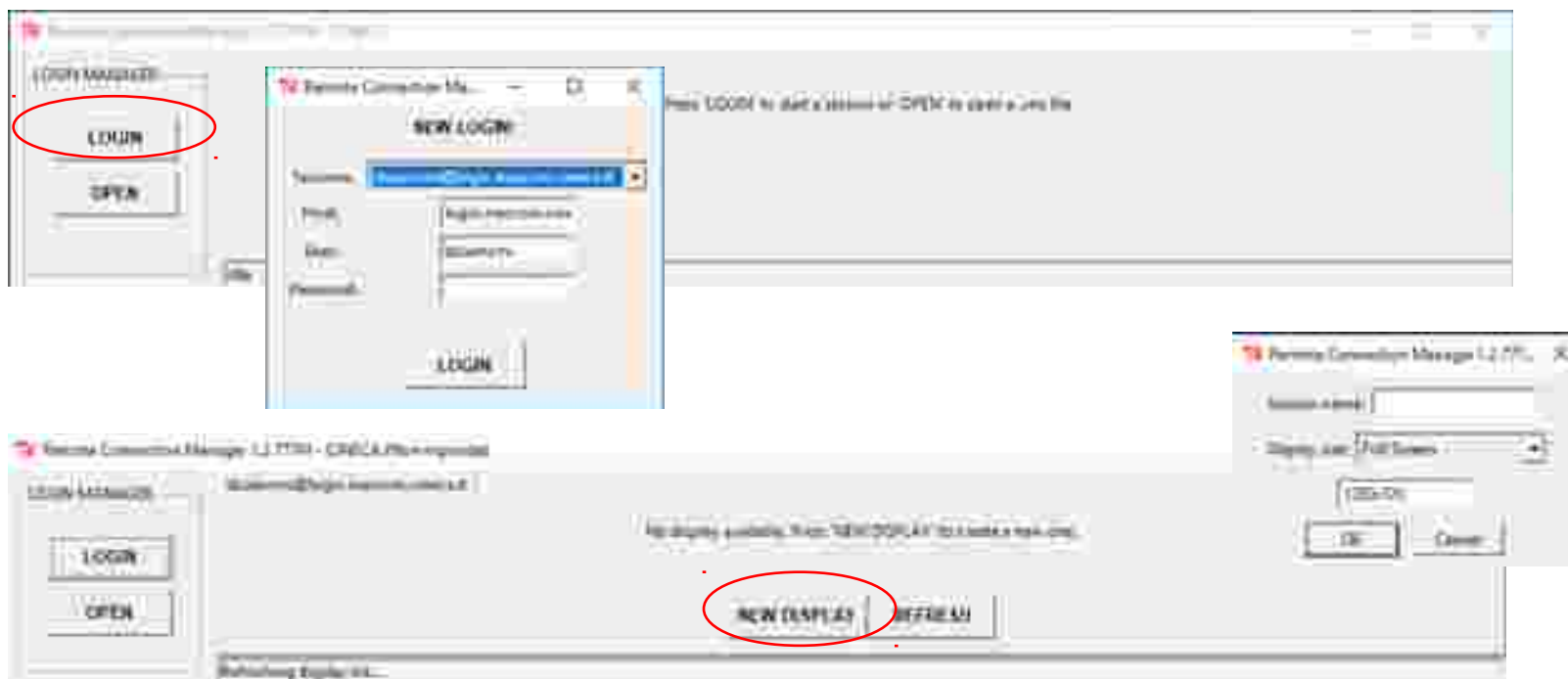
Graphic session on Marconi



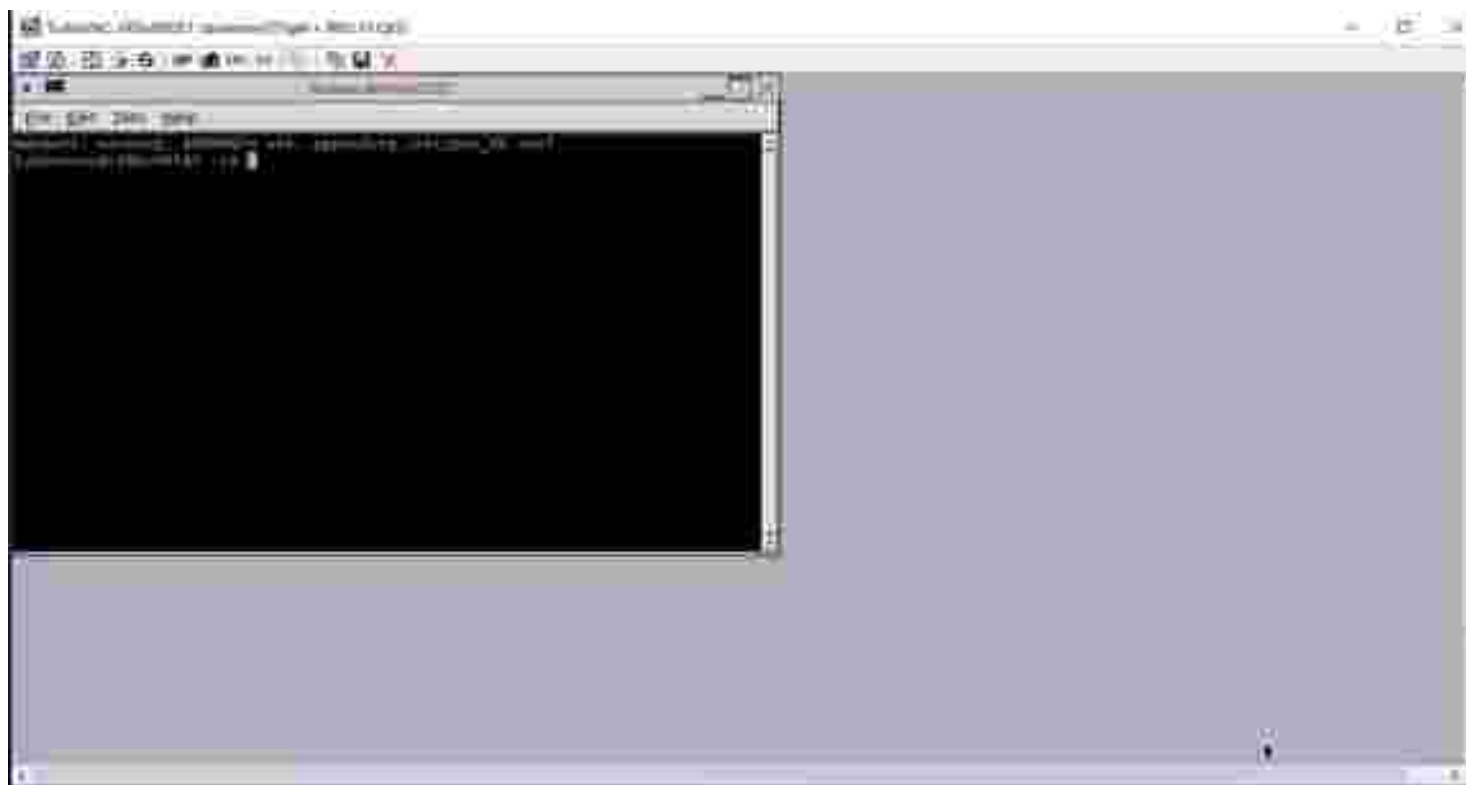
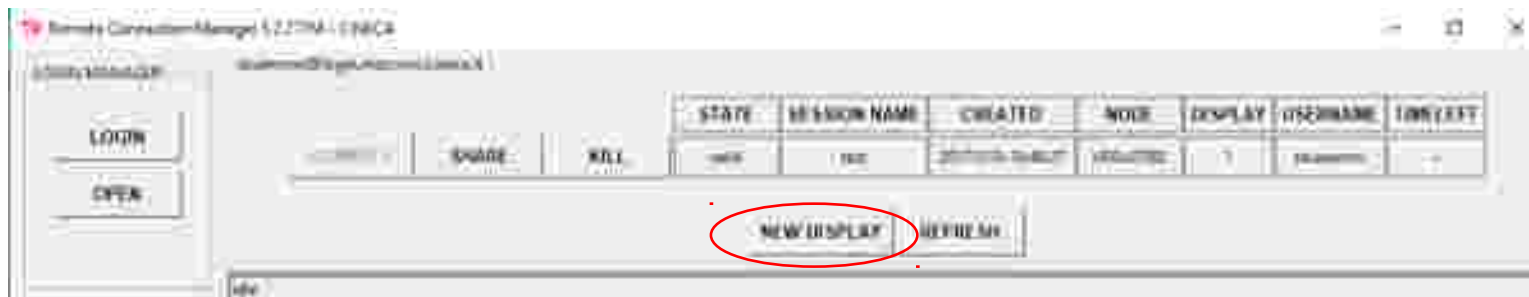
🔑 We recommend the tool **RCM** (Remote Connection Manager) to run a graphic session on Marconi

🔑 <https://hpc-forge.cineca.it/svn/RemoteGraph/branch/multivnc/build/dist/Releases/?p=817>

🔑 Start the application double clicking on the RCM application icon (Windows) or launching the RCM binary from a shell (Linux):



Graphic session on Marconi





Useful links and documentation

🔑 General User Guides related to CINECA's HPC environment

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

🔑 Marconi (BDW, KNL, SKL) specific User Guide

🔑 <https://wiki.ugov.it/confluence/display/SCAIUS/UG3.1%3A+MARCONI+UserGuide>

🔑 Information about PBS Batch Scheduler

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

🔑 Tips about improving code vectorization

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

🔑 e-mail addresses:

🔑 superc@cineca.it

Helpdesk, write here for any problem or question related to our HPC environment

🔑 corsi@cineca.it

For informations about training activities (courses, schools,...) at CINECA