EUROfusion users: Marconi and D.A.V.I.D.E.-Fusion environment

The EUROfusion community has a dedicated partition of Marconi, consisting in 806 nodes of the A1 partition (from Jul 2016 to Aug 2017).

EUROfusion community has also a dedicated partition on Marconi-A2 (Knights landing) with 449 nodes. Starting from January 2018 the number of available nodes on Marconi-A2 are 288.

Starting from August 2017 Marconi-fusion is being upgraded to 1512 node on Marconi-A3 (SkyLake), increasing the peak performance from 2 to 5 PFlops. By November 2018 the number of available nodes on Marconi-A3 has been increased up to 2410.

The EUROfusion community had a dedicated partition of D.A.V.I.D.E., consisting in 40 nodes between January 2019 and January 2020. Starting from January 2018 the activity on D.A.V.I.D.E. has been stopped.

Starting from April 2020, the EUROfusion community has a dedicated partition on the new accelerated platform Marconi100. Please refer to the general Marconi100 guide for details on the EF production environment.

The general environment defined on Marconi for the EUROfusion community is the same as the one defined for all the users of the cluster. The general environment refers to:

- Access
- Accounting
- Disks and Filesystems
- Module environment
- Programming environment

Please refer to the Marconi UserGuide for a description of the previous topics, as well as of the system architecture.

The general environment defined on D.A.V.I.D.E. for the EUROfusion community is the same as the one defined for all the users of the cluster. Please refer to the D.A.V.I.D.E. UserGuide for a description of D.A.V.I.D.E. environment (Access, Accounting, Disks and Filesystems, Modules environment, Programming environment and Additional information).

Data storage and Filesystems

The storage organization conforms to the CINECA infrastructure (see Section Data Storage and Filesystems). In addition to the home directory $HOME, for each user is defined a scratch area $CINECA_SCRATCH, a large disk for the storage of run time data and files. A $WORK area is defined for each active project on the system, reserved for all the collaborators of the project. This is a safe storage area to keep run time data for the whole life of the project. Note that this area is not available on D.A.V.I.D.E.

Storage Production Areas: $WORK and $CINECA_SCRATCH

These two areas share the same physical device, have the same block size and they also have the same performance in terms of data throughput. $WORK and $CINECA_SCRATCH are conceived as working directories for large files used and produced by batch jobs. Also, the blocking features make these areas more suitable for large binary files.

* $WORK: permanent, project specific, local *

There is one $WORK area for each active project on the machine that all users belonging to can use for production runs and storage of their output data. The owner of the main directory is the PI (Principal Investigator), but all collaborators are allowed to read/write in there. Collaborators are advised to create a personal directory in $WORK for storing their personal files. By default the personal directory will be protected (only the owner can read/write), but protection can be easily modified, for example by allowing write permission to project collaborators through "chmod" command. The default quota for a project $WORK area is 1TB, but it is possible to consider a quota extension if needed (please mailto: superc@cineca.it). File retention in the $WORK area is related to the life of the project. Files in this area will be conserved up to 6 months after the project expiring date, and then they will be cancelled. Please note that there is no back-up on this area.

To manage the different WORK areas for different projects please use the "chprj" command. For a brief description of the command, just type "chprj --help" to print the help page.

To check for the occupancy of this area please use the "cindata" command, that will list all filesets containing any file owned by your username.

* $CINECA_SCRATCH: temporary, user specific, local *
The main difference of this area with respect to $WORK is that it is user specific (not project specific) and that it can be used for sharing data with people outside your project. There is one $CINECA_SCRATCH area for each username on the machine. By default, file access is open to everyone, in case you need more restrictive protections, you can set them with "chmod" command. On this area a periodic cleaning procedure is applied, with a normal retention time of 40 days: files are cancelled on a daily basis by an automatic procedure if not accessed for more than 40 days. Please take in mind that this time interval of 40 days may be reduced in case of critical usage ratio of the area. In this case, users will be notified via HPC-News. When files are deleted, a file listing all deleted files for a given day will be created: CLEAN_<yyyymmdd>.log, where <yyyymmdd> = date when files were cancelled.

$CINECA_SCRATCH does not have any disk quota. However, it is strongly recommended to maintain a low occupancy of this area in order to prevent the very dangerous filling condition. To check for the occupancy of this area please use the "cindata" command, that will list all filesets containing any file owned by your username.

Repository area for collaborative work among different projects and across platforms: $DRES

This is a data archive area available only on-request (please mailto: superc@cineca.it), shared with all CINECA HPC systems and among different projects. $DRES is not mounted on the compute nodes. This means that you cannot access it within a batch job: all data needed during the batch execution has to be moved to $WORK or $CINECA_SCRATCH before the run starts.

MARCONI-A1

Starting from October 15th, 2017 the EUROfusion activity on Marconi-A1 has been transferred to Marconi-A3.

MARCONI-A2

A personalized production environment is in place for EUROfusion also on the A2 (KNL) partition.

288 nodes are reserved to EUROfusion with "cache" memory model.

Since this Marconi partition is used by several users at the same time, all production jobs must be submitted using a queuing system. This guarantees that the access to the resources is as fair as possible.

As on the rest of the cluster, batch jobs are managed by the SLURM batch scheduler, that is described in section Batch Scheduler SLURM.

In addition to reserved compute nodes, a set of dedicated partitions have been defined in the scheduler configuration.

The maximum number of cores and the maximum walltime depend on the chosen SLURM partition and QOS.

In the following table you can find all the main features and limits imposed on the current partitions and QOS. However for up-to-date information, use the "sinfo" and "scontrol show partition <partition_name>" commands on the system itself. There is only one production partition (knl_fuad) and you need to explicitly request the knl_qos_fuadbg to give a higher priority to the debug jobs.

<table>
<thead>
<tr>
<th>partition</th>
<th>QOS (quality of service)</th>
<th>#nodes</th>
<th>max walltime</th>
<th>max memory (MB)</th>
<th>priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>knl_fuad</td>
<td>knl_qos_fuadbg</td>
<td>min = 1 max = 2</td>
<td>00:30:00</td>
<td>86000</td>
<td>85</td>
</tr>
<tr>
<td>(no QOS)</td>
<td></td>
<td>min = 1 max = 284</td>
<td>48:00:00</td>
<td>86000</td>
<td>40</td>
</tr>
<tr>
<td>knl_qos_fuadwprio</td>
<td></td>
<td>min = 1 max = 284</td>
<td>48:00:00</td>
<td>86000</td>
<td>0</td>
</tr>
</tbody>
</table>

The maximum number of nodes available for one job is 284 with a maximum walltime of 48 hours.

The maximum memory per node is 86000MB:

```
#SBATCH --mem=86000
```

For information on how to submit and manage jobs on KNL partition, please refer to the Marconi UserGuide.
According to the amount of resources (ncpus and walltime) needed by your job, specify the appropriate partition by using the \texttt{SLURM directive "--partition"}. For example, for production jobs requiring more than 2 nodes and/or more than 30 minutes of walltime please use:

\texttt{#SBATCH --partition=knl\_fua\_prod}

and for debug jobs please add also the corresponding QOS (quality of service) by using the \texttt{SLURM directive "--qos"}:

\texttt{#SBATCH --partition=knl\_fua\_prod}
\texttt{#SBATCH --qos=knl\_qos\_fuadbg}

As usual on systems with the SLURM scheduler, you submit a job script with the command:

\texttt{> sbatch <options> script}

You can get a list of defined SLURM partitions (on the entire cluster - you will be allowed to submit jobs only to the "*\_fua\_*" partitions) with the command:

\texttt{> sinfo}

For more information and \texttt{examples of job scripts}, see section Batch Scheduler SLURM.

\section*{MARCONI-A3}

This partition, made of 2410 nodes (SkyLake, 48 cores, 192000MB) is in production since August 2017 (initially with 1512 nodes, enlarged to 2410 nodes since November 2018) and is reserved to the EUROfusion community.

As on the rest of the cluster, all production jobs must be submitted using a queuing system. Batch jobs are managed by the SLURM batch scheduler, that is described in section Batch Scheduler SLURM.

The \textbf{maximum number of cores per job} and the \textbf{maximum walltime} depend on the chosen partition.

In the following table you can find all the main features and limits imposed on the SLURM partitions. For up-to-date information, use the "sinfo" and "scontrol show partition <partition\_name>" commands on the system itself.

<table>
<thead>
<tr>
<th>partition</th>
<th>QOS (quality of service)</th>
<th>min/max # nodes per job</th>
<th>min/max # nodes per user</th>
<th>max walltime</th>
<th>max memory (MB)</th>
<th>priority</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ski_fua_dbg</td>
<td>\textit{no QOS}</td>
<td>4 max per job</td>
<td>4 max per user</td>
<td>2:00</td>
<td>182000</td>
<td>40</td>
<td>12 dedicated nodes Job submission to the partition is limited to max 10 jobs</td>
</tr>
<tr>
<td>ski_fua_prod</td>
<td>\textit{no QOS}</td>
<td>65 max per job</td>
<td></td>
<td>24:00</td>
<td>182000</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>ski_fua_prod</td>
<td>ski_qos_fuabprod</td>
<td>66 min per job</td>
<td>512 max per job</td>
<td>24:00</td>
<td>182000</td>
<td>85</td>
<td>Max 1024 nodes total</td>
</tr>
<tr>
<td>ski_fua_prod</td>
<td>ski_qos_fualprod</td>
<td>11 max per job</td>
<td></td>
<td>72:00</td>
<td>182000</td>
<td>85</td>
<td>Max 22 nodes/user Max 66 nodes total</td>
</tr>
<tr>
<td>ski_fua_prod</td>
<td>ski_qos_fuspecial</td>
<td>782 max per job</td>
<td></td>
<td>180:00</td>
<td>182000</td>
<td>100</td>
<td>\texttt{on request write to: superc@ci_neca.it}</td>
</tr>
<tr>
<td>ski_fua_prod</td>
<td>ski_qos_fualowprio</td>
<td>65 max per job</td>
<td></td>
<td>24:00</td>
<td>182000</td>
<td>0</td>
<td>Max 2 jobs/user Max 950 nodes total for exhausted active projects</td>
</tr>
</tbody>
</table>
For information on how to submit and manage jobs for SKL partition, please refer to the Marconi UserGuide.

As usual on systems with the SLURM scheduler, you submit a batch job script with the command:

```
> sbatch <options> script
```

You can get a list of defined partitions (on the entire cluster - you will be allowed to submit jobs only to the "_fua_" partitions) with the command

```
> sinfo
```

For more information and examples of job scripts, see section Batch Scheduler SLURM.

**D.A.V.I.D.E.**

*Starting from January 2020 the activity on D.A.V.I.D.E. has been stopped.*

Starting from January 2019, 40 nodes (Power8, 16 cores, 256GB) of D.A.V.I.D.E. are reserved to the EUROfusion community.

As on others clusters, all production jobs must be submitted using a queuing system. Batch jobs are managed by the SLURM batch scheduler, that is described in section Batch Scheduler SLURM.

The **maximum number of cores** and the **maximum walltime** depend on the chosen partition.

In the following table you can find all the main features and limits imposed on the SLURM partitions. For up-to-date information, use the "sinfo -d" and "scontrol show partition <partition_name>" commands on the system itself.

<table>
<thead>
<tr>
<th>partition</th>
<th>QoS</th>
<th>core per job</th>
<th>cores per user</th>
<th>GPUs per node</th>
<th>max wall time</th>
<th>max running jobs per user</th>
<th>max memory per node</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>dvd_all_serial</td>
<td>noQOS</td>
<td>max = 1</td>
<td></td>
<td></td>
<td>0</td>
<td>04:00:00</td>
<td>15200MB</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(max mem = 15200MB)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dvd_fua_prod</td>
<td>noQOS</td>
<td>min = 1</td>
<td>min = 1</td>
<td>min = 1</td>
<td>4</td>
<td>24:00:00</td>
<td>252000MB</td>
<td>--gres=gpu:N(N=1,4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>max = 512 (32 nodes)*</td>
<td>max = 512 (32 nodes)*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dvd_fua_dbg</td>
<td>noQOS</td>
<td>min = 1</td>
<td>min = 1</td>
<td>min = 1</td>
<td>4</td>
<td>00:30:00</td>
<td>252000MB</td>
<td>--gres=gpu:N(N=1,4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>max = 32 (2 nodes)*</td>
<td>max = 32 (2 nodes)*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(*) 16 core per node, 8 threads per core

For information on how to submit and manage jobs on DAVIDE, please refer to the D.A.V.I.D.E. UserGuide.

As usual on systems with the SLURM scheduler, you submit a batch job script with the command:

```
> sbatch <options> script
```

You can get a list of defined partitions (on the entire cluster - you will be allowed to submit jobs only to the "_fua_" partitions) with the command

```
> sinfo
```

For more information and examples of job scripts, see section Batch Scheduler SLURM.

**Other remarks**

**PLEASE NOTE THE FOLLOWING IMPORTANT REMARKS:**

1) **to impose a one-node granularity on your jobs we imposed a scattered arrangement and exclusive placement of resources** for your jobs.
Hence, each resource will be placed on a different node and each node will be job-exclusive. We also defined a default value of cpus per chunk equal to the full number of cores/node (36 on A1, 68 on A2, 48 on A3). Hence, if you need for example 3 nodes you can simply write:

```bash
#SBATCH --nodes=3
```

You can always specify the other resource, e.g. mpiprocs, for example for a hybrid job using 24 OMP threads and 2 MPI processes per node (on A3):

```bash
#SBATCH --nodes=3 --ntasks-per-node=2 --cpus-per-task=24
```

Please note that you can overwrite the default value if you ask explicitly for a specific number of ncpus, nevertheless you will pay always for the full node.

2) following the granularity request on the resources, we also set the default memory for each chunk to be equal to the total memory available on a compute node. Again, the parameter only applies if the user do not ask explicitly for a specific amount of memory.

3) **You always have to specify the partition (and when needed the qos)** according to the needed resources. For example, if you need to run on 64 nodes the "skl_fua_prod" queue is what you need. As an example consider a pure-MPI job requiring 64 nodes and 1 hour of walltime on Marconi-A3:

```bash
#!/bin/bash
#SBATCH --nodes=64 --ntasks-per-node=36
#SBATCH --mem=182000
#SBATCH --time=01:00:00
#SBATCH --account=FUA32_XXXXX
#SBATCH --partition=skl_fua_prod
<load some modules or set some env variables>
<execute your code>
```

4) Please note that the recommended way to launch parallel tasks in slurm jobs is with **srun**. By using srun vs mpirun you will get full support for process tracking, accounting, task affinity, suspend/resume and other features.

5) Controlling the processes and threads affinity is crucial to ensure the optimal performances on Marconi-A2 and Marconi-A3. Do not rely on slurm autoaffinity and use the proper SLURM --cpu-bind option.

6) All users with active (i.e., not expired) projects but exhausted budget are automatically enabled to use the QOS **knl_qos_fualowprio** (for the projects defined on Marconi-A2) and **skl_qos_fualowprio** (for Marconi-A3), which allow to keep submitting jobs with no priority.

Marconi-A2:

```bash
#SBATCH --account=FUA2X_XXXXX
#SBATCH --partition=knl_fua_prod
#SBATCH --qos=knl_qos_fualowprio
```

Marconi-A3:

```bash
#SBATCH --account=FUA3X_XXXXX
#SBATCH --partition=skl_fua_prod
#SBATCH --qos=skl_qos_fualowprio
```

For more information and examples of job scripts, see section **Batch Scheduler SLURM**.

7) All users with active and non exhausted budgets can request to be enabled to use the fualowprio QOS by association to FUA34_LOWPRIO (for Marconi-SKL) projects by writing to superc@cineca.it. You can then submit job (at low priority) using these accounts as above:

Marconi-A2:

```bash
#SBATCH --account=FUA23_LOWPRIO
#SBATCH --partition=knl_fua_prod
#SBATCH --qos=knl_qos_fualowprio
```

Marconi-A3:

```bash
#SBATCH --account=FUA34_LOWPRIO
#SBATCH --partition=skl_fua_prod
#SBATCH --qos=skl_qos_fualowprio
```

8) EUROfusion community users are suggested to use some particular settings for the Intel *I_MPI_ADJUST* family environment variables. This choice should guarantee optimal performances controlling the corresponding collective algorithm selection. In particular:

```bash
I_MPI_ADJUST_GATHERV=3
```
has been found optimal for a given workload containing a huge number of collective calls and for a large number of nodes.

9) INTEL compiler options: processor specific optimizations flags

To optimize the performance of your code on SKL and KNL architectures follow the guidelines reported here.

10) (OLD) CPU binding on Marconi

Please refer to this document (Courtest of T. Ribeiro), that provides practical guidelines on how to use the process pinning options within the IntelMPI library together. The document refers to the PBS batch scheduler system, that was available on Marconi before the transition to SLURM.