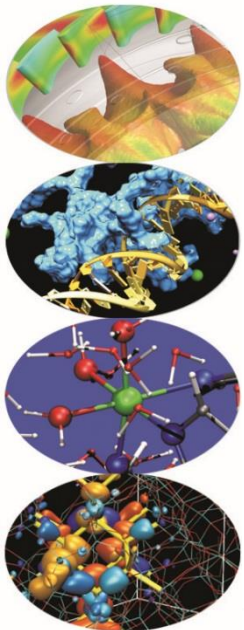


Parallel I/O and management of scientific data

Giorgio Amati, Luca Ferraro,
Gianni Morelli, Mario Tacconi
CINECA



The (real) agenda

Wednesday, 18th

- ✓ 09.30-09:45 Registration
- ✓ 09.45-10.30 I/O: State of the art (G. Amati)
- ✓ 10:30-11:30 Management of large scientific data (G. Morelli)
- ✓ 11:30-13:00 MPI2-IO: theory and practice (L. Ferraro)
- ✓ 13:00-14:15 Lunch
- ✓ 14:15-18:00 MPI2-IO: theory and practice (L. Ferraro)

Thursday, 19th

- ✓ 09:30-10:30 MPI2-IO: theory and practice (L. Ferraro)
- ✓ 10:30-13:00 HDF5: theory and practice (G. Amati/M. Tacconi)
- ✓ 13:00-14:15 Lunch
- ✓ 14:15-18:00 HDF5: theory and practice (G. Amati/M. Tacconi)

Hands-out

- Hands-out and examples can be downloaded at:
 1. <https://hpc-forge.cineca.it/files/CoursesDev/public/>
 2. go to [2016](#)
 3. go to [Rome](#)
 4. go to [Parallel I O and management of large scientific data/](#)

Using desktop

- Use module
 - ✓ `module purge`
 - ✓ `module available (av)`
 - ✓ `module list (li)`
 - ✓ `Module load`
 - ✓ `Module load autoload`

```
[caspurc-05-usere@caspurc-05] $ module av
----- /usr/local/Modules/3.2.10/modulefiles -----
autoload                hdf5/intel-serial/1.8.16
gcc/5.2                  intel/compilers/pe-xe-2016
grace/5.1                intel/mkl/11.3
gromacs/5.0.4            intel/vtune/16.1
hdf5/gnu-api16-serial/1.8.16 openmpi/1.10.1/gcc-5.2
hdf5/gnu-parallel/1.8.16  openmpi/1.8.5/gcc-4.8
hdf5/gnu-serial/1.8.16   paraview/4.4.0
hdf5/intel-parallel/1.8.16 vmd/1.9.2
```

Using desktop

- Loading compiler
 - ✓ `module load gcc/5.2`
 - ✓ `module load intel/compilers/pe-xe-2016`
- Using mpi
 - ✓ `module load autoload openmpi/1.10.1/gcc-5.2`
- Using serial hdf5
 - ✓ `module load autoload hdf5/gnu-serial/1.8.16`
 - ✓ `module load autoload hdf5/intel-serial/1.8.16`
 - ✓ `module load autoload hdf5/gnu-api16-serial/1.8.16`
- Using parallel hdf5
 - ✓ `module load autoload hdf5/gnu-parallel/1.8.16`
 - ✓ `module load autoload hdf5/intel-parallel/1.8.16`

exampleHDF5.tar

To expand `exampleHDF5.tar` file, syntax:

...

```
tar -xvf exampleHDF5.tar
```

...

Now a directory HDF5 is created, with this structure:

HDF5

|— `u_00001000.h5`

|— `sample.h5`

|— **PARALLEL**

| └— **RUN**

└— **SERIAL**

 └— **RUN**