DATA SCIENCE &
SCIENTIFIC COMPUTING

openMPI library usage (materials to be revised)

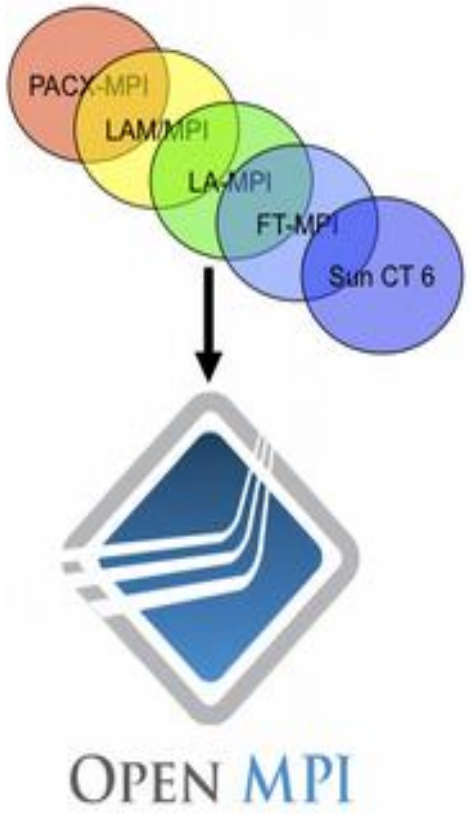
- Stefano Cozzini
- CNR-IOM and eXact lab srl

Parallel programming

- MPI is a standard with many implementations
- You need a library to link to your MPI-enable parallel code
- Many implementation available:
  - OpenMPI
  - MVAPICH
  - IntelMPI
  - MPICH
  - Etc..

openMPI

- Evolution of several prior MPI's
- Open source project and community
- Production quality
- Vendor-friendly
- Research- and academic-friendly
- MPI-3.0 compliant



https://www.open-mpi.org/

OpeMPI and compilers

- OpenMPI works with several compiler suites

```
Module load openmpi/3.1.3..

openmpi/3.1.3/gcc/4.8.5-z2zfbgq
openmpi/3.1.3/gcc/8.2.0-qh4llbm
openmpi/3.1.3/pgi/18.10-ahjhvki
```

Mpirun/mpiexec

- Mpirun and mpiexec
  - Completely identical (in OpenMPI)
- General form:
  - `mpirun -np X your_exe`
  - `mpirun [-np X] --hostfile hostfile your_app`
- If using a scheduler,no need for hostfile or-np

How to map mpirun processess..

- Map proceess by speficied object foo
  - `mpirun –mapby  <foo>`
  - `Supported options include slot,hwthread, core, L1cache, L2cache, L3cache, socket, numa, board, node, sequential, distance, and ppr.`

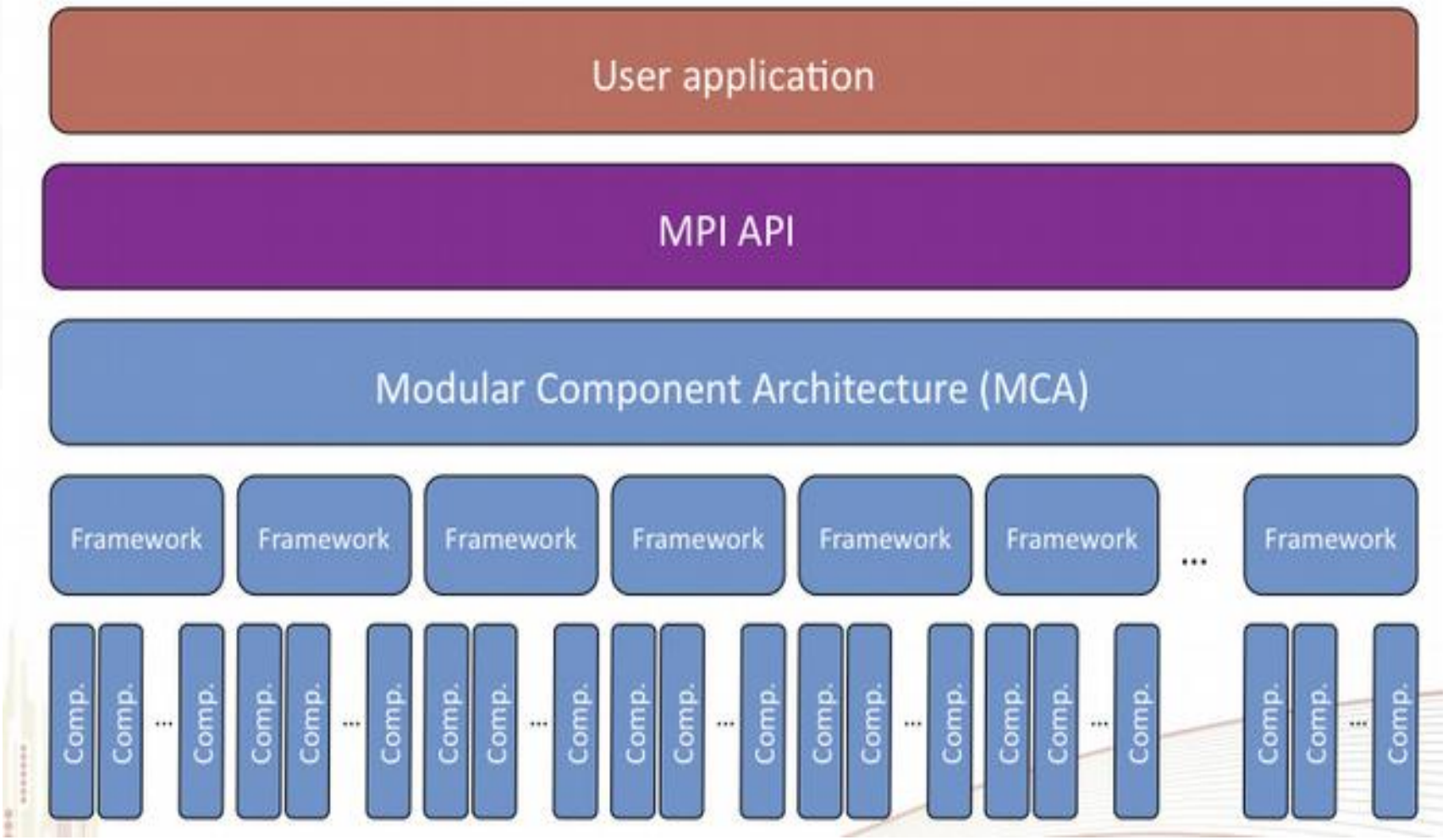
Mpirun useful options

- Assign only a certain number of MPI process on one node
  - `-npernode  X`
- Indicates how many cores to bind per process
  - --cpus-per-proc <#perproc>
- Show how processes are bind to cores/sockets etc..
  - --report-bindings

OpenMPI is Based on Plugins

- Lots and lots of plugin types
  - Back-end network
  - Resource manager support
  - Operating system support
- All can be loaded (or not) at runtime
  - Choice of network is a runtime decision

Plugin high level view



MCA parameters

- Run-time tunable values
  - Per layer
  - Per framework
  - Per component(“plugin”)
- Change behaviors of code at run-time
  - Does not require recompiling/relinking

Example: specify BTL

- BTL:Byte Transfer Layer
  - Framework for MPI point-to-pointcommunications
  - Select which network to use for MPI communications

```
mpirun --mca btl tcp,self -np 4 my app
```

- Components
  - tcp:TCP sockets
  - self:Loopback (send-to-self)

Example:specify openIB BTL

```
mpirun --mca btl openib,self -np 4 my
app
```

- Components
  - openib:OpenFabricsverbs(InfiniBand)
  - self:Loopback(send–to–self)

What does this do ?

```
mpirun  -np 4 my app
```

- Use all available components
  - tcp,sm,openib,…
- TCP too?
  - Yes and no..
  - TCP is automatically disable itself in the presence of better network/protocol

What does this do ?

```
mpirun  -np 4 my app
```

- More specifically:
  - Open each BTL component
  - Query if it wants to be used
  - Keep all that say“yes”

    Rank by bandwidth and latency rank

you can check with `--verbose` option

What does this do ?

```
mpirun  -np 4  --mca btl ^tcp my app
```

- Use all available components except tcp
- More specifically:
  - Open eachBTL component except tcp
  - Query if it wants to be used
  - Keep all that say“yes”

    Rank by bandwidth and latency rank

MPI freely available benchmarks (2)

- IMB-4.0 (now IMB2017) (INTEL MPI benchmark)
  - MPI protocol ()
  - https://software.intel.com/en-us/articles/intel-mpi-benchmarks
- OSU benchmarks: http://mvapich.cse.ohio-state.edu/benchmarks/

Suggested activities

- Play with Intel MPI benchmark
- Compile it using openMPI with different compiler
- Submit your job using two or more nodes
- Play with different BTL
- Report/understand difference