JIVE

Joint Institute for VLBI ERIC

MPI and SLURM

SFXC Workshop

Mark Kettenis, 22 September 2025

What is MPI?

- Message Passing Interface
- (de-facto) Standard API for programming parallel computers
- Many implementations: OpenMPI and MPICH are the most popular
- Vendor implementations are typically based on OpenMPI
-

SFXC and MPI

- Sample data streams don’t use MPI API
  - Typically TCP sockets
- All other communications do use MPI
  - Model use MPI broadcast
- SFXC isn’t aggressively multithreaded
  - Use single core for correlator “nodes”
  - Use up to two cores for input “nodes”
  - Use multiple cores for the output “nodes”
    - For correlations producing large output date rates

The OpenMPI ranks file

- Necessary to assign specific SFXC sub-processes (rank) to CPU cores (slots) on cluster nodes
- rank 0: Manager “node”
  - Only needs a single core
  - This node needs access to the delay directory
- rank 1: Log “node”
  - Only needs a single core
- rank 2: Output “node”
  - Assign two or more cores if possible
  - This node needs access to the output directory
- rank 3-*n*: Input “nodes”
  - Assign two cores if possible (per rank)
  - One for each input datastream
  - These nodes need access to the data!
- rank *n*+1-*m*: Correlator “nodes”
  - Only needs a single core (per rank)
  - Need at least one rank for each (output) sub-band

```
# job.ranks
rank 0=headnode slot=0
rank 1=headnode slot=1
rank 2=headnode slot=2,3
rank 3=flexbuf-1 slot=0,1
rank 4=flexbuf-3 slot=0,1
rank 5=flexbuf-1 slot=2,3
rank 6=mark6-3 slot=0,1
rank 7=node-a0 slot=0
rank 8=node-a1 slot=0
rank 9=node-b0 slot=0
rank 10=node-b1 slot=0
rank 11=node-a0 slot=1
rank 12=node-a1 slot=1
rank 13=node-b0 slot=1
rank 14=node-b1 slot=1
```

The OpenMPI machines file

- Need to list all machines used in the job
- With the number of available slots
- OpenMPI will bind slot to cores by default

```
# job.ranks
rank 0=headnode slot=0
rank 1=headnode slot=1
rank 2=headnode slot=2,3
rank 3=flexbuf-1 slot=0,1
rank 4=flexbuf-3 slot=0,1
rank 5=flexbuf-1 slot=2,3
rank 6=mark6-3 slot=0,1
rank 7=node-a0 slot=0
rank 8=node-a1 slot=0
rank 9=node-b0 slot=0
rank 10=node-b1 slot=0
rank 11=node-a0 slot=1
rank 12=node-a1 slot=1
rank 13=node-b0 slot=1
rank 14=node-b1 slot=1
```

```
# job.machines
headnode slots=4
flexbuf-1 slots=16
flexbuf-3 slots=16
mark6-3 slots=8
node-a0 slots=20
node-a1 slots=20
node-b0 slots=20
node-b1 slots=20
```

Starting an MPI job

- Use mpirun on mpiexec to start a correlation job
- `mpirun –machinefile sfxc.machines –rankfile sfxc.ranks –np <num_processes> /full/path/of/sfxc <ctrlfile> <vexfile>`
  - `<num_processes>` need to be at least 3 + *n* + *m* with
  - *n* = number of input data streams (typically number of stations)
  - *m* = number of sub-bands to be correlated

What is SLURM?

- Job scheduling system
- Includes MPI support
- Typically used in batch mode
  - But supports interactive mode as well
- Simplifies the “easy” case where input data is on a cluster filesystem
  - We’re still figuring out how to handle EVN production correlation this way

SLURM commands

- salloc: allocates resources and starts sub shell
- srun: runs job interactively
- sbatch: starts batch job
- squeue: inspect queues
- scancel/skill: kill (batch) job

Questions?