Contribution ID: **445** Type: **Oral Presentation**

MPI Job Manager

Monday, August 8, 2022 4:50 PM (20 minutes)

MPI Job Manager (MPI_JM) is "scheduler" designed enable users to make maximum use of heterogenous architectures, particularly which require a "swarm" of independent MPI tasks is required for a complete calculation - such as lattice QCD calculations of correlation functions on pre-existing configurations. MPI_JM managers all these tasks through lightweight C++ code supported by Python3. MPI_JM allows users to describe the resource requirements of their tasks (GPU-intense, CPU-only, number of nodes, wall clock time, etc) as well as their dependencies. MPI_JM then schedules these tasks on an allocation on an HPC platform based upon user defined priority and dependencies. Jobs with GPU-intense and CPU-only requirements are placed upon the same nodes, maximizing the use of all node resources. This is all managed with a single mpirun call, minimizing the requirements of the service nodes that manage an HPC system. Planned features include (among others):

- Multiple job-configurations: as the wall clock of the allocation nears the end, the optimal run configuration may not have enough time to complete, but doubling the nodes at a performance loss would allow a job to complete in time. MPI_JM can try alternate configurations specified by the user, to use up the otherwise idle cycles towards the end of a job allocation
- Try again: sometimes, the GPUs on a node will just fail to start up in time, causing a job to time out. MPI_JM can be instructed to try N-times before giving up and trying a new job, or removing those nodes from the allowed ones to be used in the allocation.
- Use real wall-clock time rather than user specified estimate: Optinoally, MPI_JM will track performance of similar jobs in a database, and then use this information to provide more reliable estimates of wall-clock time requirements than what is specified by the user.
- etc.

Primary authors: WALKER-LOUD, Andre (Lawrence Berkeley National Laboratory); MCELVAIN, Ken (UC Berkeley)

Presenter: WALKER-LOUD, Andre (Lawrence Berkeley National Laboratory)

Session Classification: Software development and Machines

Track Classification: Software development and Machines