Overview

- MPI+OpenMP is a privileged option for implementing hybrid applications in the context of the U.S. DOE Exascale Computing Project (ECP)
- MPI and OpenMP implementations are not designed to work together
- It is difficult for users to precisely control the deployment of MPI+OpenMP applications
  - By default, the OpenMP runtime will assume that all local resources can be used
  - The definition of OpenMP places limits how an hybrid application can be deployed
  - Impossible to support complex “layouts”

Key Challenges

- How can users describe complex MPI+OpenMP layouts to run on compute nodes?
- Can users change the layout of an application at runtime?
  - *OpenMP limitations*: places are set during initialization and cannot be changed
  - *MPI limitations*: the number and placement of MPI ranks can be changed; however, how can we set a new OpenMP layout on a per-MPI rank basis

Layout – Example

- Goal: applications take full benefit of the underlying hardware
- Example based on the architecture of OLCF Summit’s compute nodes:

Layout Description

- Definition of the layout illustrated above
- Passed through the mpiexec command line:
  `mpirun --mca rmaps explicit --mca rmaps_explicit_layout "[MPI, ...]" --np 128 myapp.exe`

Example focusing on the context of a single rank

```c
int main (int argc, char **argv) {
(...)
MPI_Init (&argc, &argv);
MOC_Init (&argc, &argv);
(...)
MOC_Finalize ();
MPI_Finalize ();
}
```

Execution

1: mpirun triggers our mapper which parses the layout, publishes the layout through PMIs and places the rank
2: the app calls MOC_Init; MOC retrieves the layout via PMIs specific to the current rank and set the OMPI PLACES environment variable (the OpenMP runtime is not yet initialized), specifically for the current MPI rank
3: the app enters the first OpenMP regions, which initializes the OpenMP runtime; the threads are deployed according to the OMPI PLACES