Topology Aware Application Execution

Profiling tools, Pattern detection, and Task-aware Mapping

Lead: Laxmikant Kale, Akhil Langer, Nikhil Jain
Team: Abhinav Bhavele
Profiling Tools and Pattern Detection

- Transparent interception of communication calls to capture important information

Use cases:

- Performance analysis
- Automatic categorization of jobs to BW queues
- Suggest geometry for job execution
- Input data for task-aware mapping and dynamic reassignment

![Diagram of MPI communication and profiling tools](image)
Topology Aware Mapping

- Rearranging placement of tasks at start up and dynamically during the execution
- Variety of algorithms: tree-based, spacing filling curves e.g. Hilbert, polynomial time approximation scheme (PTAS), grid-based
- Specialized algorithms for common geometries: aligned, corned, affine, embed
- Fully automated based on trial runs: nodelist-based reordering just before the job starts
Collaboration with App Teams

❖ On look out for applications that *are not* sure
  ➢ If they are communication bound
  ➢ Which BW queues should they use
  ➢ Whether topology aware mapping can improve performance

❖ We can help perform topology aware mappings for applications!

❖ Examples of successful applications of mapping
  ➢ NAMD has seen performance benefits up to 25%
  ➢ OpenAtom’s performance improved by up to 40% and is less susceptible to allocation geometry
  ➢ Up to 25% improvements for MILC and pF3D using Rubik