Topology Aware Application Execution

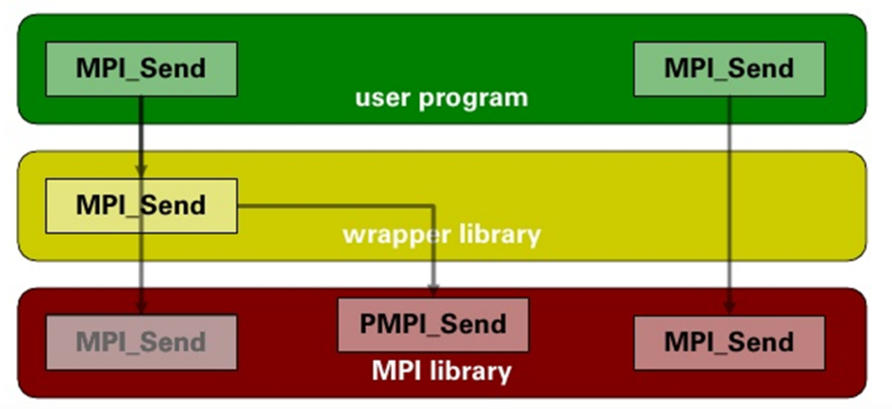
Profiling tools, Pattern detection, and Task-aware Mapping

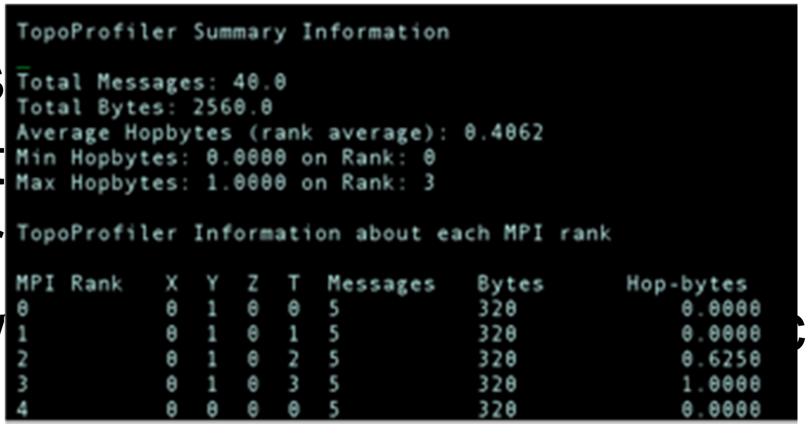
Lead:	Laxmikant Kale,
Team:	Akhil Langer, Nikhil Jain
Collaborators:	Abhinav Bhatele

Profiling Tools and Pattern Detection

- Transparent interception of communication calls to capture important information

- Use cases:





```
TopoProfiler Summary Information

Total Messages: 40.0
Total Bytes: 2560.0
Average Hopbytes (rank average): 0.4062
Min Hopbytes: 0.0000 on Rank: 0
Max Hopbytes: 1.0000 on Rank: 3

TopoProfiler Information about each MPI rank

MPI Rank     X  Y  Z  T  Messages   Bytes      Hop-bytes
0            0  1  0  0  5          320           0.0000
1            0  1  0  1  5          320           0.0000
2            0  1  0  2  5          320           0.6250
3            0  1  0  3  5          320           1.0000
4            0  0  0  0  5          320           0.0000
```

reassignment

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