SAND2015-7610PE

Architecture-aware Task Placement

Unstructured Mesh Technologies

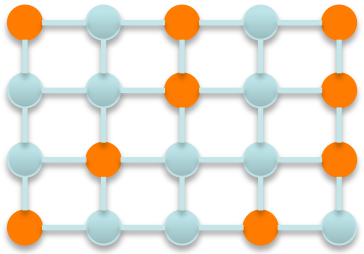
Deveci, Devine, Leung, Prokopenko, Rajamanickam
Sandia National Laboratories

With collaborators at Knox College, Ohio State University, Sandia National Laboratories (ASC and LDRD program)

FASTMath SciDAC Institute

At high core counts, scalability can drop due to bad placement of tasks in nodes and network

- **Motivation**: In a typical parallel computing environment:
  - Applications' MPI tasks are assigned to node allocations without regard for app's communication patterns and data locality
  - Allocations may be sparse; spread far across network
  - Communication messages may travel long routes
- **Status in 2011**:
  - Scalability lost above 8K cores even in simple stencil-based apps
  - Grouping with respect to node only (without considering the network) recovers some *but not all* scalability
- **Key gaps:**
  - Ideal metric to optimize (max hops, avg hops, congestion, etc.) unknown
  - Software for general, inexpensive task placement lacking
  - Analysis of new topologies needed



Non-contiguous node allocation in a mesh network

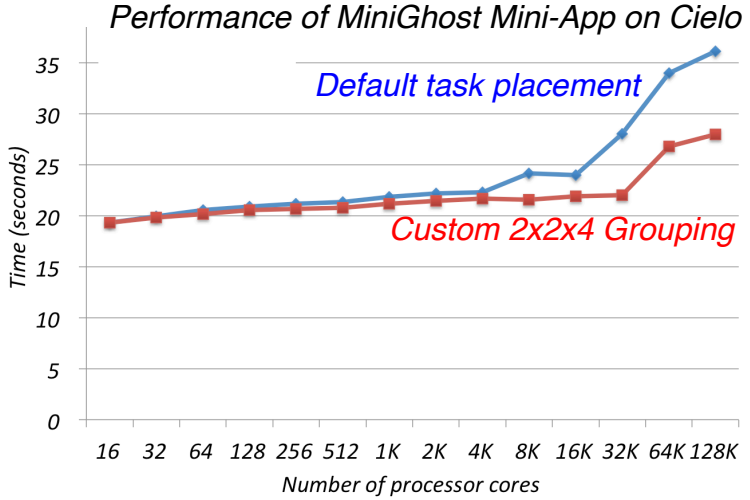
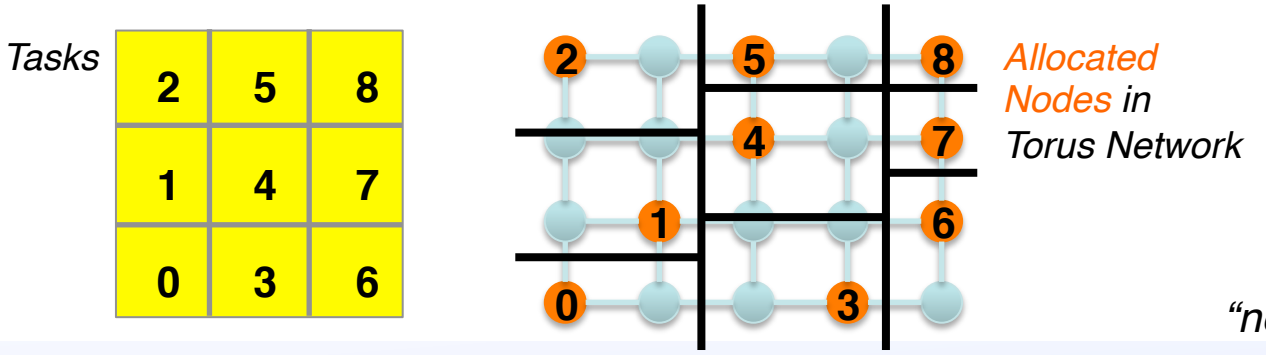


Figure courtesy of Barrett & Vaughan (Sandia)

Architecture-Aware Geometric Task Placement

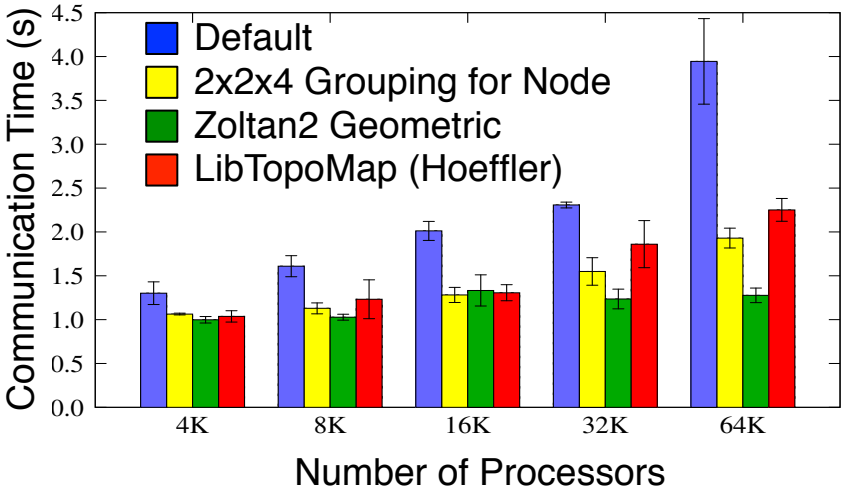
- **Goal:** Given a (possibly non-contiguous) allocation of nodes in a parallel computer, assign interdependent MPI tasks to "nearby" allocated nodes within the network
- **Related work:**
  - Much work focused on contiguous allocations (e.g., IBM BlueGene)
  - Several graph-based approaches (LibTopoMap, Scotch, Jostle)
- **Approach:**
  - For tasks, use geometric proximity as a proxy for interdependence
  - For nodes, use nodes' geometric coordinates in torus/mesh network
  - Apply inexpensive geometric partitioning algorithms to both application tasks and nodes, giving consistent ordering of both



Zoltan2's MultiJagged Geometric Partitioner (IEEE TPDS 2015) assigns related tasks to "nearby" nodes in the torus.

Accomplishment and Impact

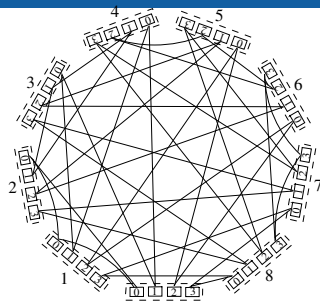
- **Accomplishment:**
  - Experimentation/simulation indicates *Average Number of Hops* is good proxy for communication costs in task placement algorithms (*ACM PPoPP14*)
  - Zoltan2's Geometric Task Placement reduced MiniGhost execution time on 64K cores (*IEEE IPDPS14*)
    - by 34% relative to default
    - by 24% relative to node-only grouping



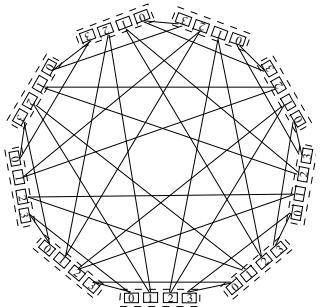
- **Impact:** Adopted by Trilinos' MueLu multigrid solver
  - Applying geometric task placement at the finest multigrid level reduced overall solve time by >10%

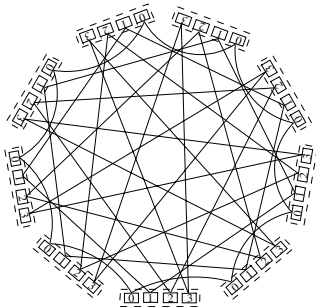
Upcoming Systems: Analysis of Dragonfly Networks

- **Challenge:**
  - Dragonfly topology is not completely specified
  - Inter-group topology affects bisection bandwidth, task placement
- **Accomplishments** (*IEEE Cluster 2015*):
  - Characterized bisection bandwidth as function of the numbers of groups and switches per group, and the ratio of global to local bandwidths for three common topologies (Camarero et al.)
- **Impact:**
  - Absolute topology (used by Cray, IBM) is most straightforward
  - But Absolute has ***constant*** bisection bandwidth even as switches and groups are added
  - With Relative and Circulant, bisection bandwidth increases ***linearly*** with number of groups, ***quadratically*** with number of switches per group



Absolute


Relative


Circulant