ParaTreeT

Joseph Hutter

New Ideas

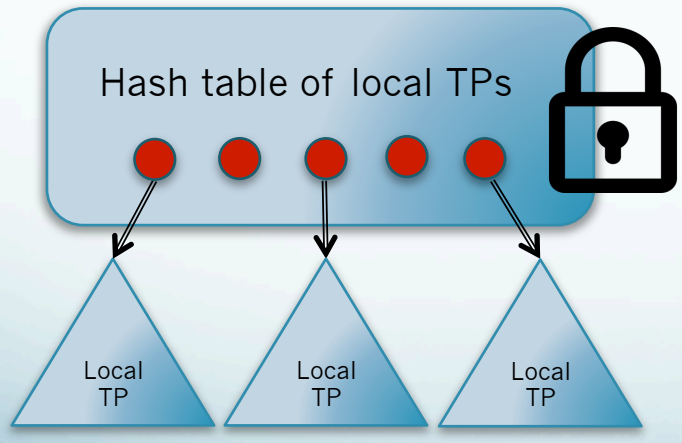
- Long-awaited generic tree walk
  - Abstracted tree type, traversal type
  - Visitor model
- Primary Optimizations
  - TreeElements
  - SMP tree-in-cache
- Absorb the power of ChaNGa

Interface

- Function calls
  - Construct proxies
  - *driver.load(configuration, input_file)*
  - *driver.run(traversal_type, num_iterations)*
- Visitor
  - *::node(Node* const source, Node* target)*
  - *::leaf (Node* const source, Node* target)*
- Data
  - *::Data(), operator+=(Data&)* for parent nodes
  - *::Data(Particle*, num_particles)* for leaf nodes

Cache SMP Model

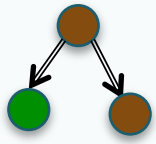
- Cached parent node (read only)
- Cached leaf node (read only)
- Root of local TreePiece
- Placeholder for remote node



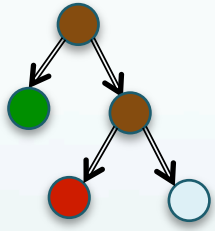
1. Receive remote data

3 nodes, 5 particles

2. Reconstruct subtree



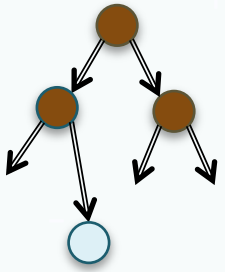
3. Check if child is a local TP
   If not, use placeholder



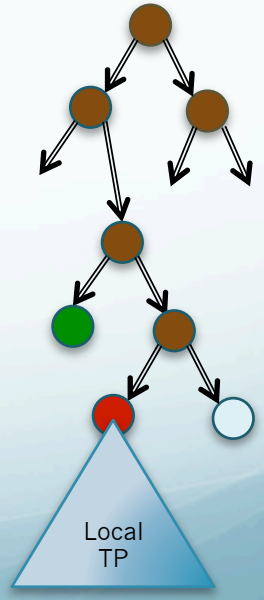
4. Atomic swap child pointer

5. Resume waiting traversals

Node-wide tree before

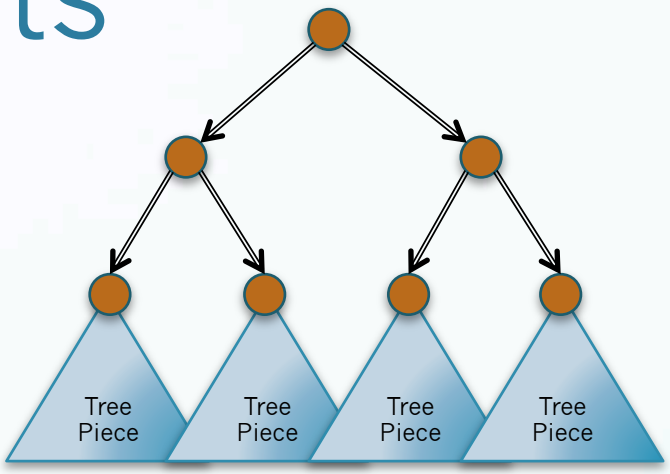


Node-wide tree after



TreeElements

- Provides inherent parallelism
  - Upward traversals
  - Dual-tree traversals
- Reduces memory footprint
  - No need to duplicate these nodes
- Indexed by keys
  - Charm++'s demand creation feature
- Stores locations of TreePieces



Support

- Traversals
  - Top-down
  - Bottom-up or Up-and-down
  - Dual-tree (requires *Visitor::cell*)
- Tree Types
  - K-ary trees
  - Octrees
  - K-D trees
- Other
  - Various decomposition schemes
  - Various prefetching schemes
  - GPU offloaded work

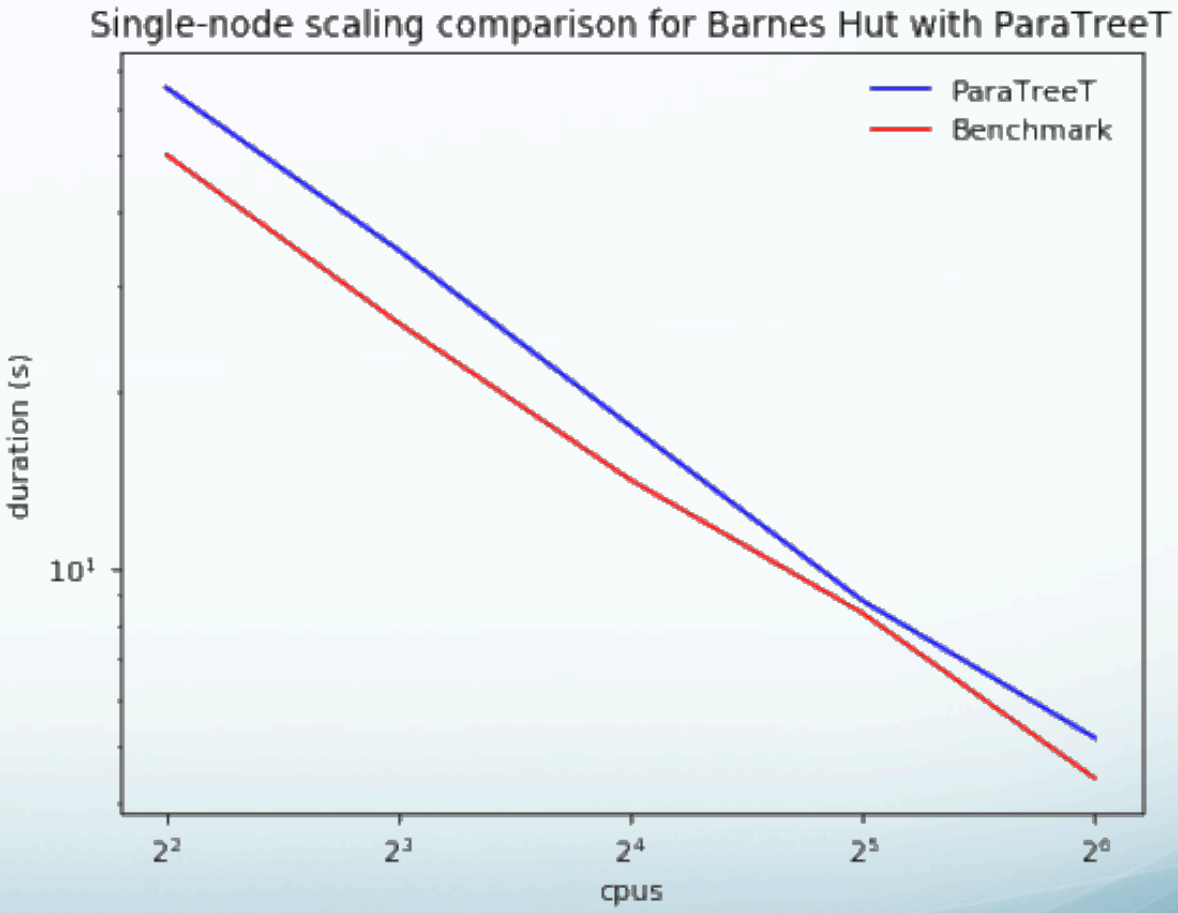
Barnes Hut

< 200 lines of user code

Stampede2 KNL

1m particles

Competitive on a single node



Barnes Hut

< 200 lines of user code

Stampede2 KNL

1m particles

Scales to 250 particles per cpu