Rubik's ParaCube: A Collection of Parallel Implementations for Optimal Rubik's Cube Solver

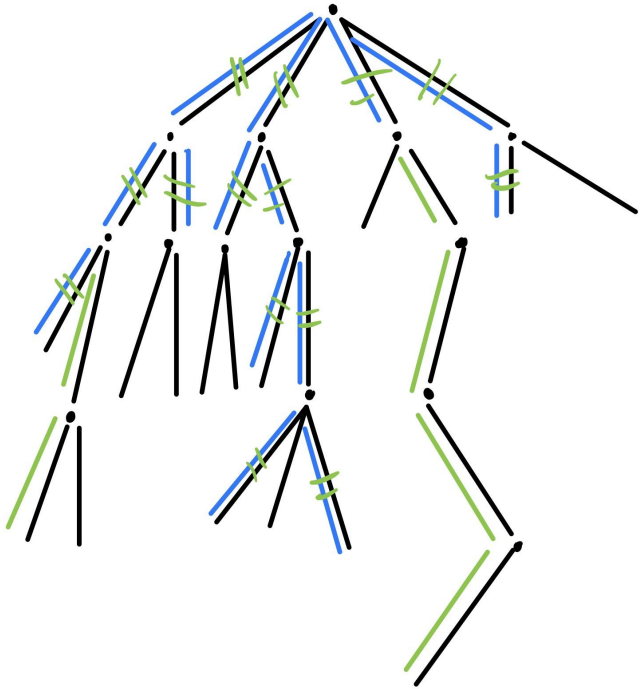
Art Zhu, Chengzhi Huang

December 14, 2020

Project Background

- Rubik’s Cube
  - Representations: F, B, L, R, U, D, F1, ..., U3, D3
  - Solver: input a series of scrambles, output a series of solution steps
- Why Iterative Deepening A*?
  - Thistlethwaite's 52-move algorithm (not optimal)
  - Optimality, BFS Search?
  - High branching factor (K ~= 12), Memory…
  - DFS? May never terminate!

IDA* and Korf's Algorithm



IDA Iterations

0 ...

...

i: visit all blue paths

i+1: visit blue + green paths

...

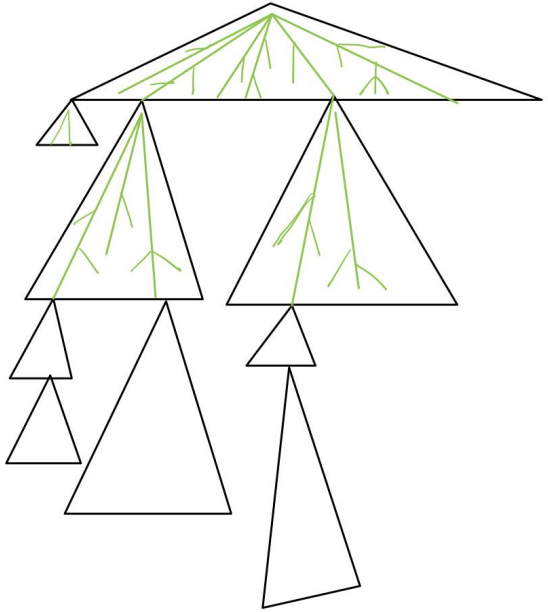
Korf Pattern Database

Map cube condition to least number of moves needed to achieve goal state.

To find the correct bound, (which nodes are leaf nodes for iteration i)

OpenMP Implementation

- Spawn enough tasks
  - To provide load balance
- Allow freedom of task management and multi level task spawn through task steal
  - Exposing worker private problem to peers



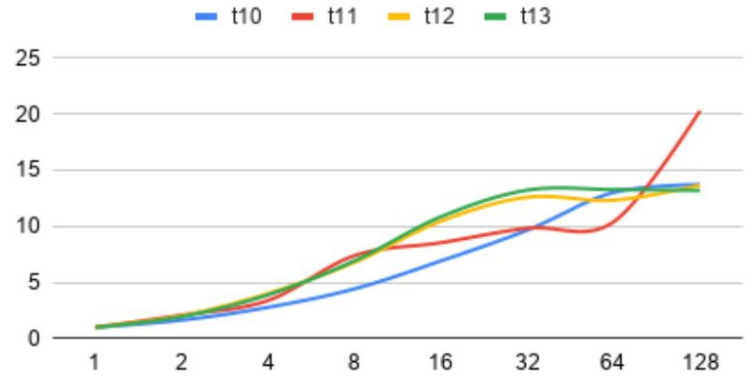
MPI Implementation

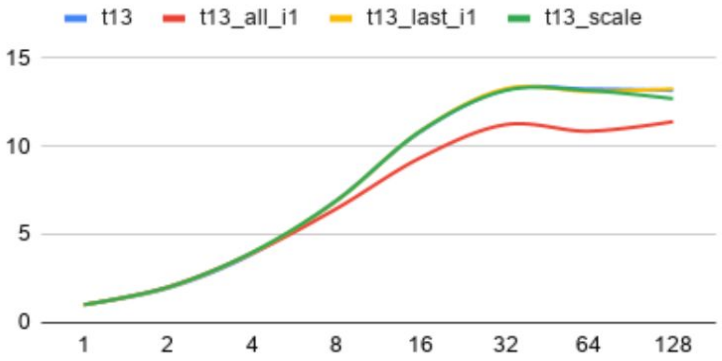
- Frontier Nodes Generation: "Best-First" Expansion
  - Nodes with similar f-values will have sub-trees of comparable sizes
  - Expand nodes with smaller f-value until we have enough frontier nodes
- Dynamic Work Assignment
  - Master generate enough frontier nodes (18*18*18 in our case)
  - Master node send node to worker nodes where they independently perform IDA* searching
  - Worker nodes report back searching results, if optimal solution detected

    master node call MPI_Abort(), otherwise, master node update its bound

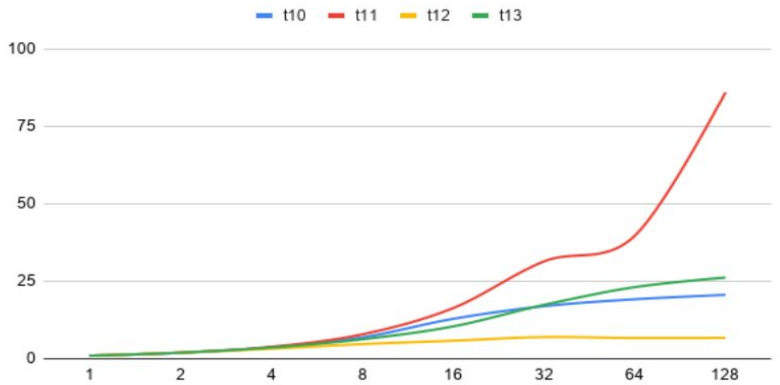
Results

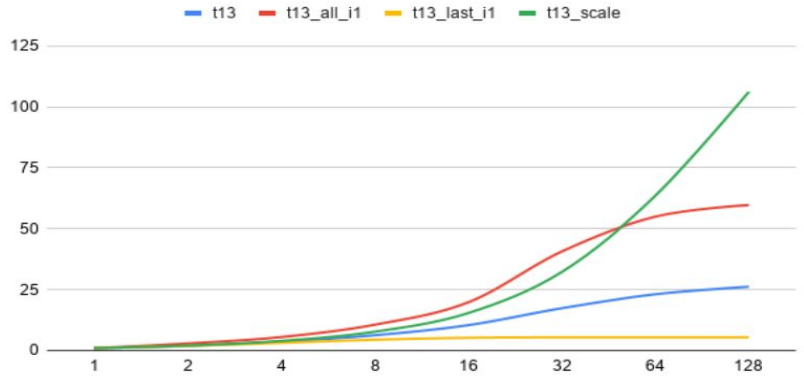
Speedup Factors for OpenMP on Latedays





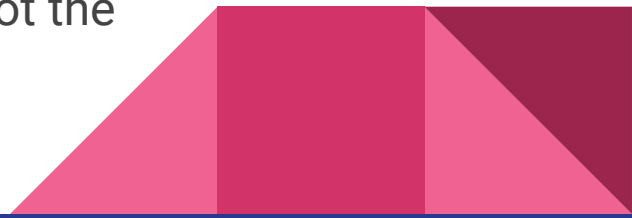
Speedup Factors for OpenMPI on Bridges





Analysis

- Property of IDA
  - For a given iteration, we have a graph to search, G_i
  - Not the entire G_i is searched in the last iteration for sequential algorithm
  - For the previous iterations, fully searched G_{i-x} has 1/K^x as much work
- Efficiency Factor
  - Traversing in different orders will yield different speedup results
  - Parallelization, incur unnecessary work (when not the entire graph G_i is searched)



Summary: Efficiency Analysis

Why parallelizing IDA is hard!

And why we are proud of our results!