Parallel Path-finding

Daksha Shrivastava, Akash Hegde
Carnegie Mellon University

The Problem

- Path-finding is integral to applications like movement of bots in video games, analysis of social networks, navigation and exploration.
- However, path-finding is computationally expensive.
- The A* algorithm helps ameliorate path-finding by using a heuristic function along with the actual distance to find the shortest path between two endpoints.

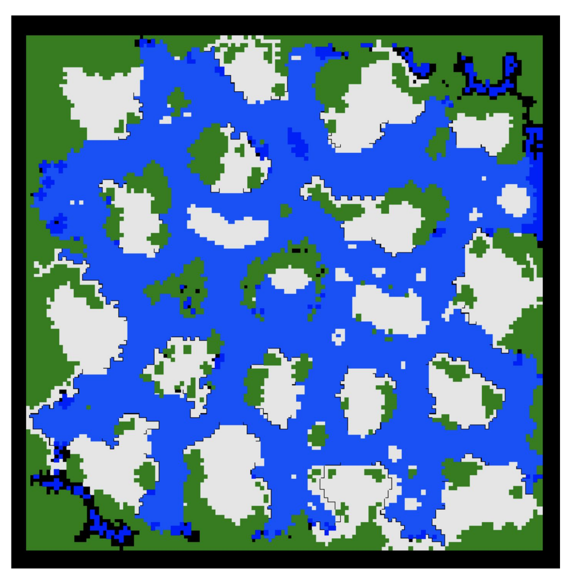
  $$f(n) = g(n)+h(n)$$

  g(n): Path cost from *Start* to *n*

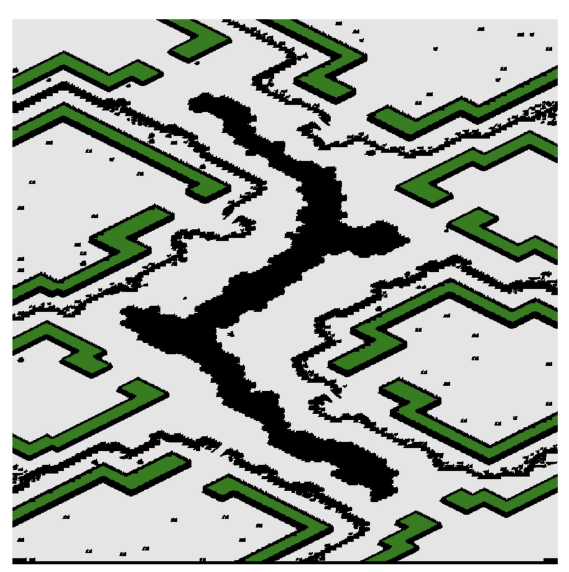
  h(n): Heuristic function to estimate path cost from *n* to *Goal*
- Even though A* algorithm gives much better performance, it is an intrinsically sequential algorithm and poses significant challenges to parallelize efficiently.

Evaluation Benchmarks:

2D Grid World Benchmarks [1]: This dataset contains simplified representations of game maps made by BioWare. This benchmark was specifically chosen to evaluate the performance of the class of problems that have grid-based representations.

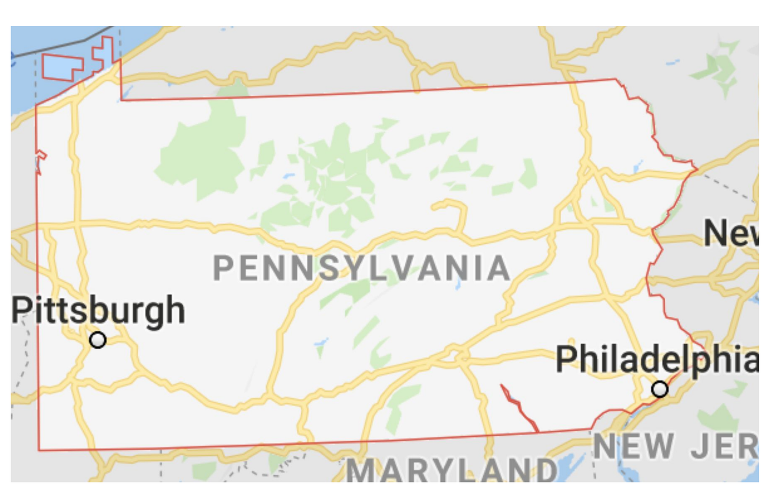


Map	Dimensions
Arena	49x49
Swamp of Sorrows	512x512
Arctic Station	768x768
Simple Maze	1024x1024
Octopus	1024x1024



SNAP: Road Network Pennsylvania[2]: This is a road network of Pennsylvania. Intersections and endpoints are represented by nodes, and the roads connecting these intersections or endpoints are represented by undirected edges.

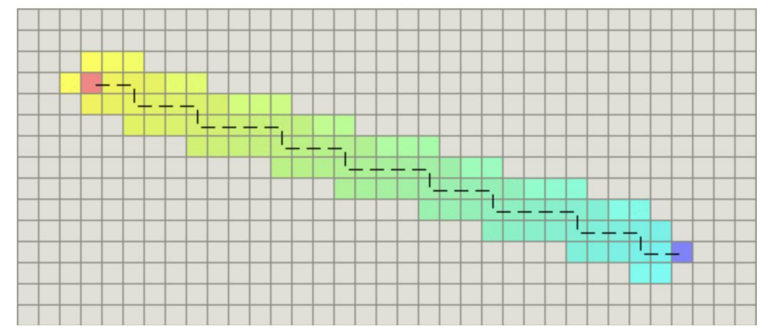
Nodes	1,088,092
Edges	3,083,796
No. of triangles	67,150
Avg. Clustering Coeff.	0.0465
Diameter	786



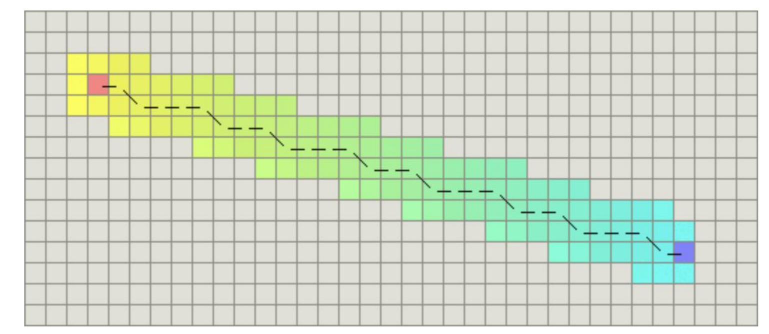
Approach

Acceleration of Heuristic Functions: The heuristic function *h(n)* tells A* an estimate of the minimum cost from any vertex *n* to the goal. It's important to choose a good heuristic function.

- Low divergence, can be done in parallel for all neighbors.
- Accelerated using ISPC-SIMD and OpenMP



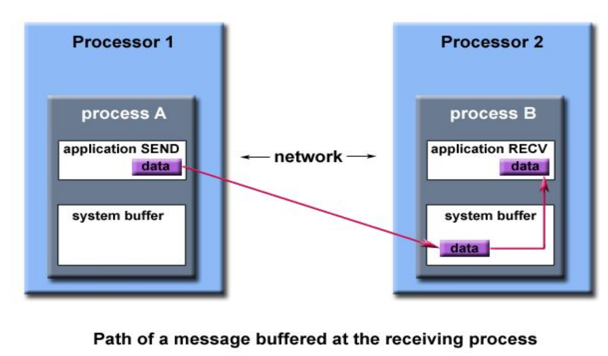
Manhattan Distance (4 directional)

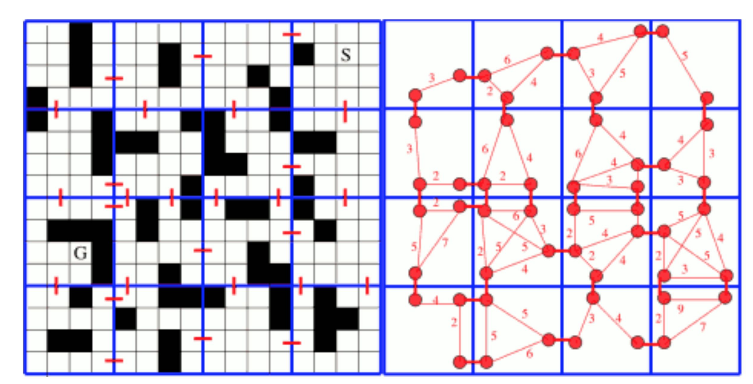


Diagonal Distance (8 directional)

Hierarchical division of the graph into clusters: Search spaces represented as grid-based graphs, like game maps, can be divided into clusters, with marked entry and exit points.

- Each individual cluster can be mapped to a processing element as an independent pathing problem.
- Once the shortest path between the marked endpoints in each cluster is determined, they can be combined to find an optimal path between the original source and destination.
- Accelerated by splitting work among compute nodes using MPI



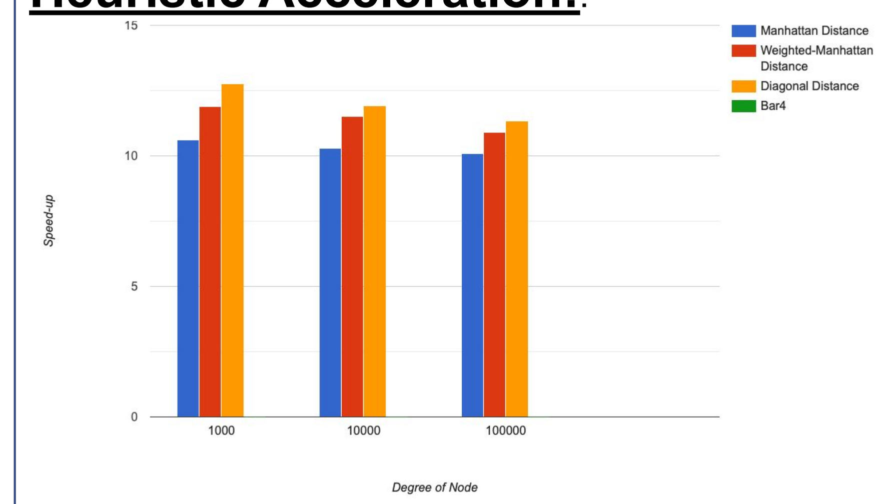


Multiple Simultaneous Source and Destination Nodes:

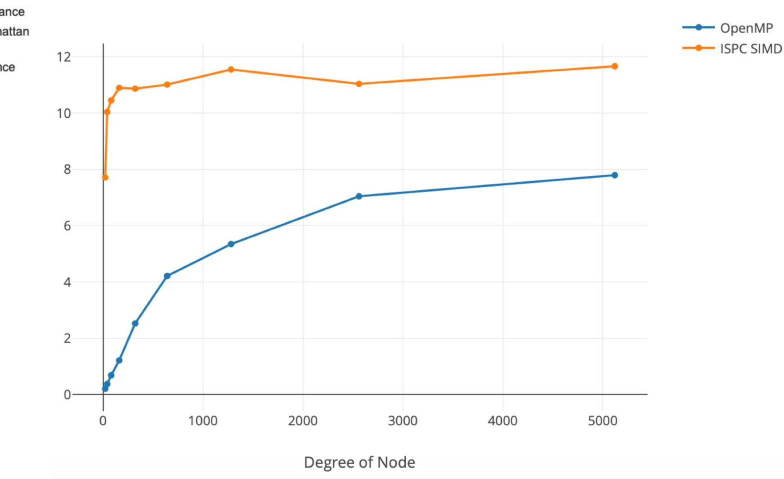
- Accelerated the multi-agent problem using OpenMP
- Agents share graph data-structures eliminating redundant copies of data, while maintaining private copies of data-structures tracking progress, thereby enabling simultaneous computation of paths for different agents.
- Each agent's source and destination nodes are different on the same map/graph.

Results and Analysis

Heuristic Acceleration:



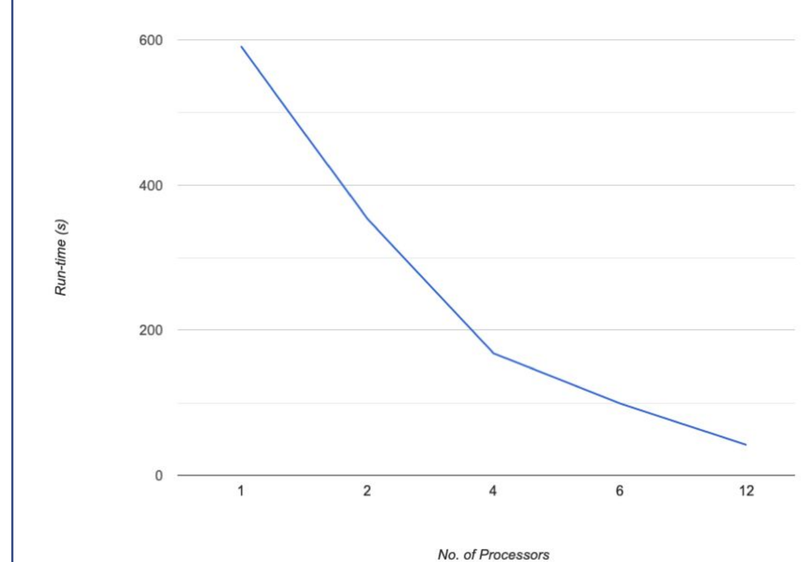
ISPC-based SIMD Speed-up of heuristic functions in grid-based game maps



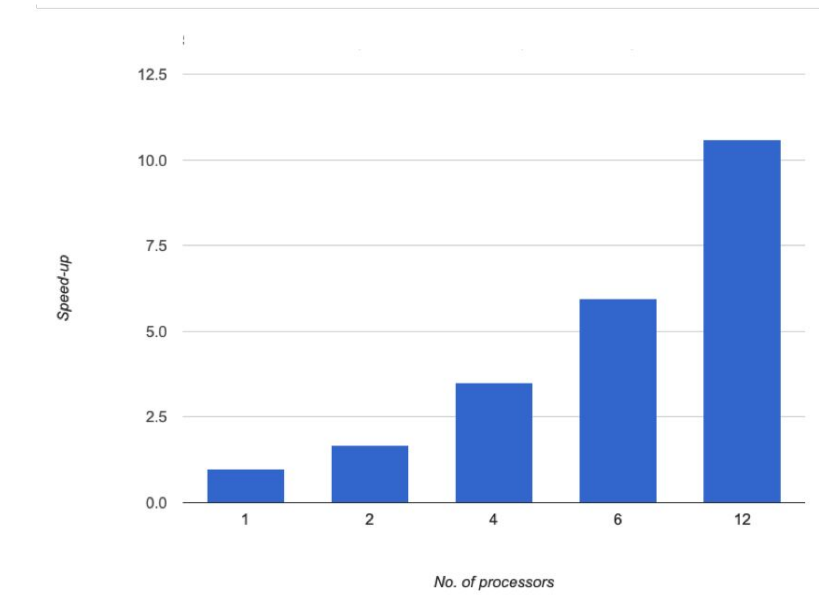
Speed-up comparison of Diagonal Distance heuristic (ISPC & OpenMP) in grid-based game maps

- ISPC-SIMD works out better, since the heuristic computation is data-parallel, and ISPC has lesser overheads than OpenMP.

Hierarchical Clustering:



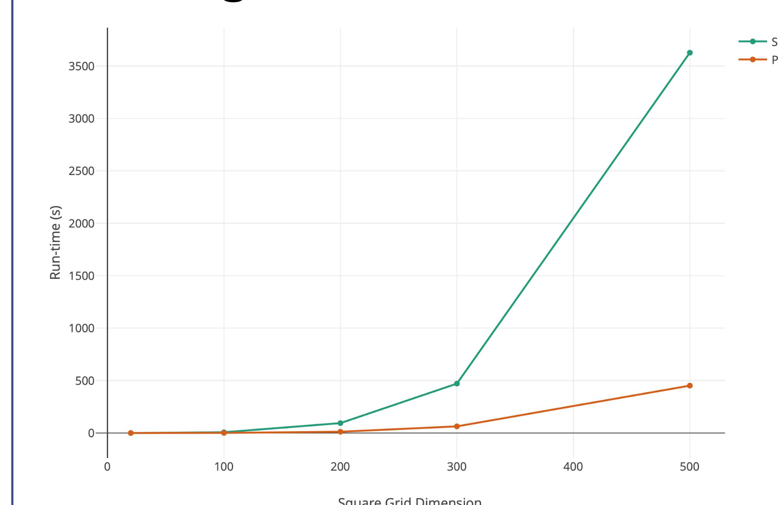
Run-time vs Processing Elements Analysis for Clustering PA road networks (Distance = 196) (MPI)



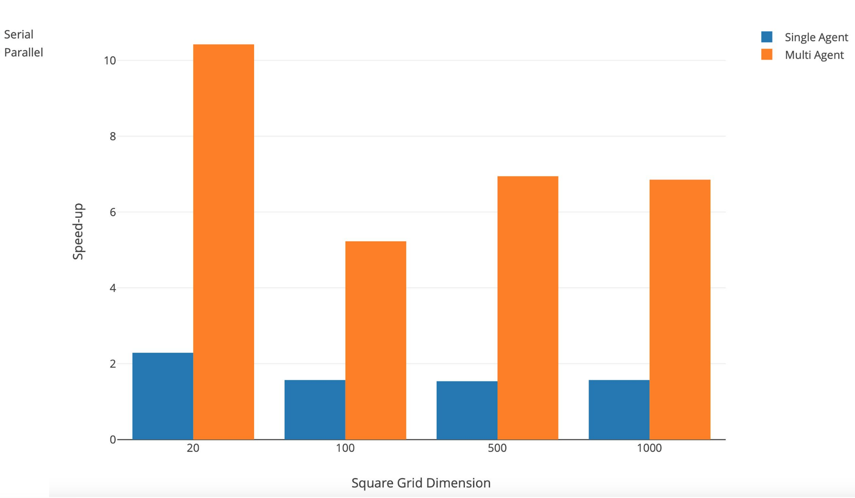
Speed-up vs Processing Elements Clustering PA road networks (Distance = 196) (MPI)

- Huge performance gain for large graphs (PA road network)
- Potential load imbalance

Multi Agent :



Run-time analysis of Multi-Agent path-finding in grid-based game maps (OpenMP)



Speed-up comparison of Multi-Agent path-finding in grid-based game maps (OpenMP)

References

[1] Stanford Network Analysis Project https://snap.stanford.edu/data/roadNet-PA.html
[2] 2D Pathfinding Benchmarks https://www.movingai.com/benchmarks/grids.html
[3] International Journal of Computer Games Technology https://www.hindawi.com/journals/ijcgt/2008/873913/
[4] Amit Patel, Introduction to the A* Algorithm https://www.redblobgames.com/pathfinding/
[5] Parallel Computer Architecture and Programming, Spring 2019 http://www.cs.cmu.edu/~418/