Data-Parallel Agent-Based Microscopic Road Network Simulation using GPUs

Peter Heywood[1], Steve Maddock[1], Mark Brackstone[2], Jordi Casas[2], David Garcia[2], Paul Richmond[1]

[1] The University of Sheffield, UK.
[2] Aimsun, Spain.

p.heywood@sheffield.ac.uk

Introduction

- Simulations are used for planning and management transport networks.
- Microscopic simulations are low-level simulations:
  - Bottom-Up simulations
  - Simulate individual vehicles and Local interactions
  - Complex behaviour emerges from simple models
  - **Computationally Expensive**

Figure 1 - Active Traffic Management in the UK [1]

- Existing tools used in industry use CPUs
  - I.e. Aimsun[2], SUMO[3], Paramics[4], VISSIM[5], etc
  - Single-core or multi-core applications
  - Task-parallel or coarse-grained data-parallel applications
  - Long run-times for large-scale networks
  - Diminishing returns from additional CPU cores (Figure 2)
  - Increased performance and scalability are required

- We have implemented a GPU accelerated road network microsimulation
- Cross-validated our implementation against Aimsun, a commercial, multi-core CPU simulator
- Benchmarked both simulators using a procedurally generated network and compared performance [6]

https://doi.org/10.1016/j.simpat.2017.11.002

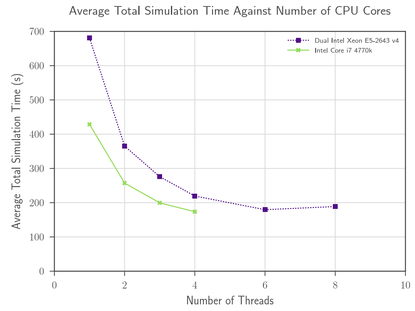

Figure 2 - Aimsun simulation performance for different CPU core counts

Models

- Implemented a subset of models implemented in Aimsun
  - Gipps' car following model [7] [8]
  - Aimsun gap-acceptance model [7]
  - Turning probabilities [7]
  - Single-lane roads
  - Stop signs
  - Vehicle detectors [7]
  - Constant vehicle arrival with virtual-queues for overflow [7]

Benchmark Network

- Artificial benchmark network
- Procedurally-generated
- Manhattan-style grid
- One-way, single-lane roads
- Demand input from outer edges
- 50%-50% turning probabilities for internal junctions
- Reduced turning probability to leave the simulated region

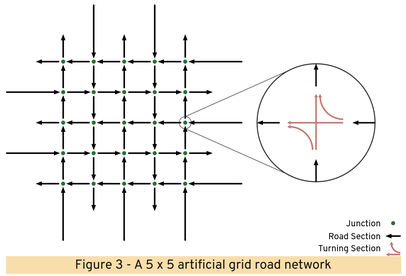

Figure 3 - A 5 x 5 artificial grid road network

[1] CC BY 2.0 Highways England https://www.flickr.com/photos/highwaysagency/9950013283/
[2] https://aimsun.com
[3] http://sumo.dlr.de
[4] http://www.sias.com/paramics
[5] https://www.ptvgroup.com/en/solutions/products/ptv-vissim/
[6]Heywood, Peter, et al. "Data-parallel agent-based microscopic road network simulation using graphics processing units." Simulation Modelling Practice and Theory (2017)
[7] Transport Simulation Systems, Aimsun 8 Users Manual (2014).
[8] P. Gipps, A behavioural car-following model for computer simulation, Transportation Research Part B: Methodological 15 (2) (1981) 105–111. doi:10.1016/0191-2615(81)90037-0.
[9] Richmond, P. "Flame gpu technical report and user guide." Department of Computer Science Technical Report CS-11-03 (2011).
[10] Richmond, Paul. "Resolving conflicts between multiple competing agents in parallel simulations." European Conference on Parallel Processing. Springer, Cham, 2014.

GPU Implementation

- Implemented models using FLAME GPU
  - Flexible Large-scale Agent Modelling Environment for the GPU
  - Template-based simulation environment for high performance, GPU-accelerated simulations [9]
  - Agents represented as X-Machines with message lists for communication
  - Provides a high level interface for describing agents, abstracting the CUDA programming model [10]
  - http://www.flamegpu.com

- CSR representation of the network graph
- Individual vehicle models implemented using one or more states and agent functions (Figure 4)
- Cross validated against CPU implementation (Aimsun), showing statistically similar results

FLAME GPU

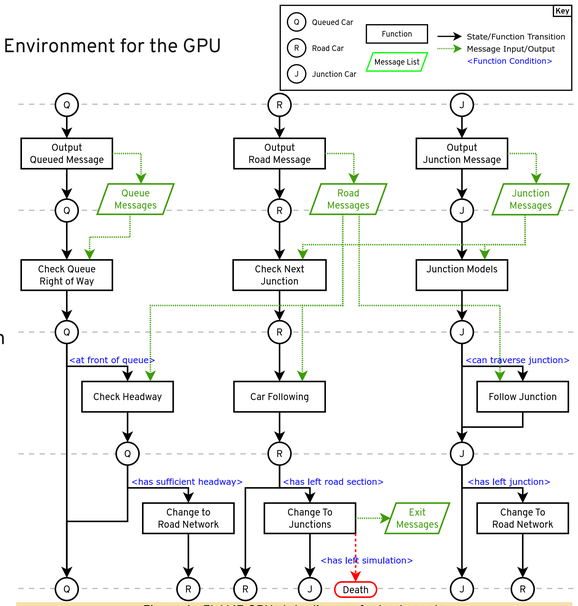

Figure 4 - FLAME GPU state diagram for 'car' agents

Agent Communication

- Agents communicate using message lists
- Message lists must bet iterated by each agent, to find the relevant message(s)
- FLAME GPU provides specialised messaging techniques
  - All to All (Global)
  - Spatial Partitioning

- Improves performance by reducing the number of messages iterated per agent
- This can still have a significant performance impact

- Road network models typically require information from a small number of local vehicles
- New *graph-based* communication strategy implemented
  - Restricts messages to those from the relevant parts of the graph
  - Allows data-parallel access to relevant messages

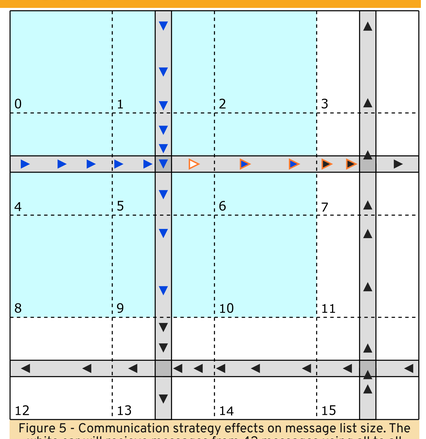

Figure 5 - Communication strategy effects on message list size. The white car will recieve messages from 42 messages using all to all communication, 18 messages in spatially partitioned (blue agents) and 5 messages using graph-based communication (orange borders).

- Figure 5 shows how each communication strategy can impact the number of messages iterated for a model such as the car following model.
- Figures 6 and 7 show effect on performance of each communication strategy
  - Graph-Based communication has higher output cost, but much lower message iteration time.

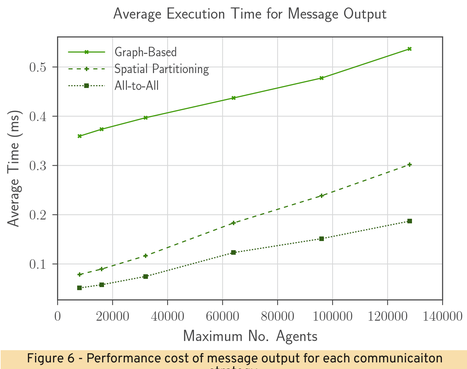

Figure 6 - Performance cost of message output for each communicaiton strategy.

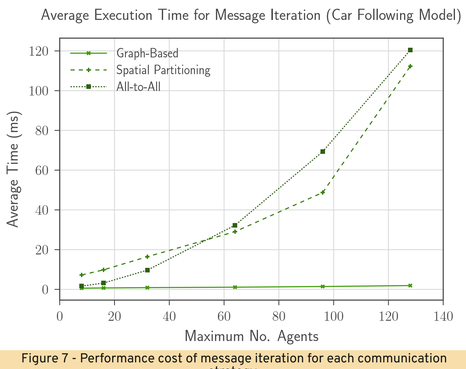

Figure 7 - Performance cost of message iteration for each communication strategy.

Performance

- Several benchmark experiments
  - Increase scale and population of network (Fig 8)
  - Increase population for fixed size networks (Fig 9-11)
  - Inspect per simulation-iteration performance (Fig 12)
- Up to **43.8x** faster than CPU
  - 0.5 million vehicles
  - **28x real time** (~0.5x on CPU)
- Workstation
  - i7 4770k
  - Titan X (Pascal)
  - GeForce GTX 1080
- Nvidia DGX-1
  - Xeon E5 2698 v4
  - Telsa P100 SMX2

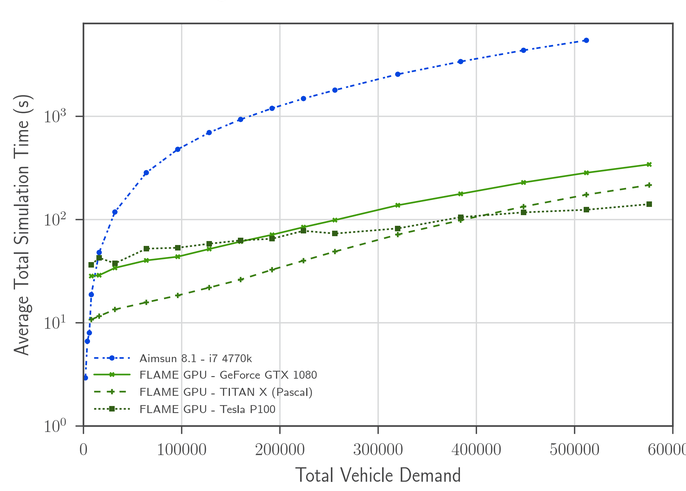

Figure 8 - Average simulation time for 1 hour simulations of the benchmark network at various scales

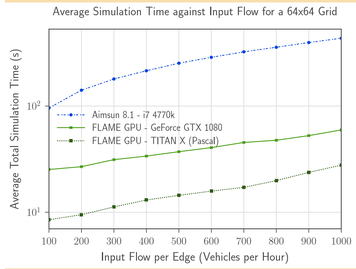

Figure 9 - Average simulation time for a 1 hour simulation of the 64x64 grid network at various input flow rates

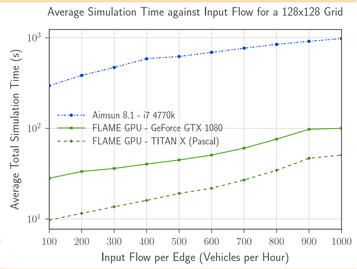

Figure 10 - Average simulation time for a 1 hour simulation of the 128x128 grid network at various input flow rates

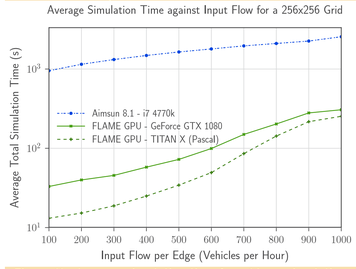

Figure 11 - Average simulation time for a 1 hour simulation of the 256x256 grid network at various input flow rates

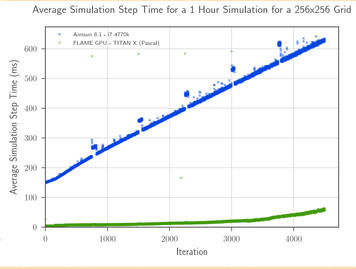

Figure 12 - Average time per simulation iterateration for each simulator.

Conclusion

- GPUs offer significant performance improvements for large-scale microscopic simulations of road networks networks
- Up to **43x** speed-up compared to equivalent multi-core CPU simulation
- Demonstrated **500,000 vehicles** simulated at **25x real time**
- Reducing global memory accesses through specialised agent communication technique enables high performance

Acknowledgements

This work was supported by a Department for Transport (DfT) Transport Technology Research Innovation Grant "Accelerating Transport Microsimulation: Demonstrating the impact of future many core simulations" and additional support through EPSRC fellowship "Accelerating Scientific Discovery with Accelerated Computing" (EP/N018869/1)