Road Network Simulation using FLAME GPU

Peter Heywood, Dr Paul Richmond & Dr Steve Maddock

Department of Computer Science,
The University of Sheffield

The University Of Sheffield.

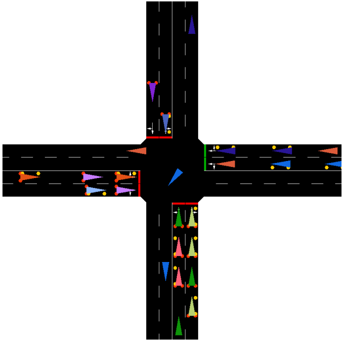
Overview

Introduction

Road Network Simulation

Need for improved road simulation systems [4, 8]

- Increasing number of vehicles globally
- Poor utilisation of existing infrastructure
- Relatively cheap
- Decision making



An example of traffic microsimulation (SUMO)

The University Of Sheffield.

Why Agent Based Simulation on the GPU

Why ABS / Microsimulation?

- More natural method of description
- Allow emergence of more complex behaviour
- Good for modelling congested networks

Why GPGPU?

- *Not* embarrassingly parallel but it is *well suited* for GPGPU computing
- Aspects are SIMD (*Same Instruction Many Data*) in nature
- Has been demonstrated as GPGPU suitable [7, 9]
- Speed-up allows for increased complexity / scale

The University Of Sheffield.

Aims

- Demonstrate performance of road network simulation using FLAME GPU
- Evaluate performance scalability using an artificial road network.
  - Scale population size
  - Scale population and environment
- Demonstrate interactive visualisation using instancing

Gipps' Car Following Model

Car Following

- Key vehicle behaviour
- Drive at desired speed without colliding into other vehicles
- Considering factors such as *reaction time*, *vehicle limitations*, *neighbouring vehicles* ...
- Many car following models exist
  - Safety-distance models
  - Psycho-physical models

Gipps' Car Following Model

Gipps' Car Following Model defined in 1981 by Peter Gipps

- Safety Distance Model
- Considers driver & vehicle characteristics
- Only considers the preceding vehicle
- One of the most commonly used models

Aims - Gipps' Car Following Model

> "The model should mimic the behaviour of real traffic" [3]

> "parameters which correspond to obvious characteristics of drivers and vehicles" [3]

> "should be well behaved when the interval between successive recalculations of speed and position is the same as the reaction time" [3]

Notation - Gipps' Car Following Model

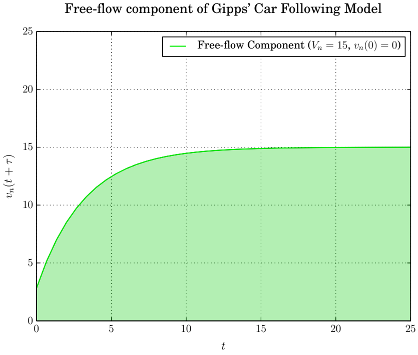
a_n	the maximum acceleration of vehicle n
b_n	the most severe braking that the vehicle n will undertake
s_n	the effective size of vehicle n, including a margin
V_n	the target speed of vehicle n
$x_n(t)$	the location of the front of vehicle n at time t
$v_n(t)$	the speed of vehicle n at time t
τ	constant reaction time for all vehicles
$\hat{b}$	estimate of leading vehicles most severe braking

Notation for variables used by Gipps' car following model

Equations - Gipps' Car Following Model

Free-flow Conditions

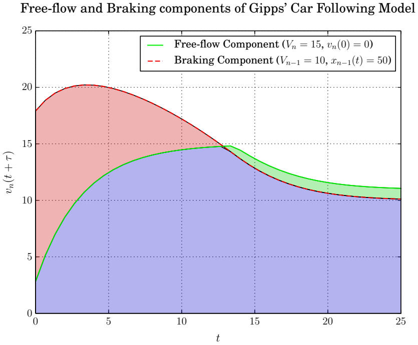
$$v_n(t+\tau) <= v_n(t) + 2.5 a_n \tau (1 - v_n(t)/V_n)(0.025 + v_n(t)/V_n)^{\frac{1}{2}}$$



Equations - Gipps' Car Following Model

Congested Conditions (Braking)

$$v_n(t+\tau) <= b_n\tau + \sqrt{{b_n}^2\tau^2 - b_n(2[x_{n-1}(t) - s_{n-1} - x_n(t)] - v_n(t)\tau - v_{n-1}(t)^2/\hat{b})}$$



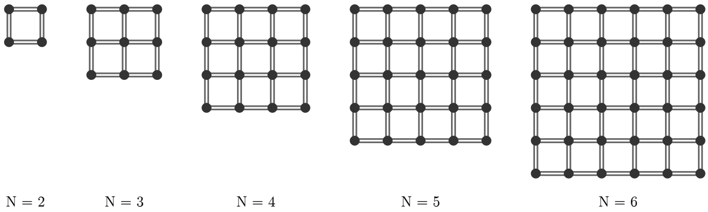
Limitations - Gipps' Car Following Model

- Time-step should be set to reaction time τ
- Assumes drivers:
  - Drive in a safe manner
  - Can make accurate observations

Implementation

Artificial Road Network

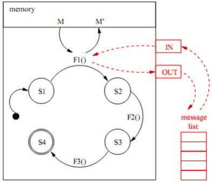
- Scales consistently unlike real world networks
- Single lane uniform grid
- Grid made of N rows and columns
- 2 sections of road between each adjacent junction
- N^2 junctions and $4N(N-1)$ one-way roads



FLAME GPU

FLAME GPU is a "template based simulation environment" for agent based simulation on Graphics Processing Unit (GPU) architecture [6]

- Agents are represented as X-Machines
- Agents can communicate via globally accessible message lists
- Messages are crucial for interaction
- Message lists can be partitioned to "ensure the most optimal cycling of messages"[6]



The University Of Sheffield.

FLAME GPU Messaging

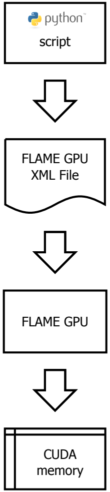
There are currently 3 defined message partitioning schemes

- **Non-partitioned messaging**
  - All to All
- **Discrete partitioned messages**
  - 2D non-mobile agents only (i.e. Cellular Automata)
- **Spatially partitioned messages**
  - Continuous space
  - Requires *radius* and *environment bounds*

Aims to reduce the size of message lists

Implementing Gipps' Car Following Model using FLAME GPU

- Each vehicle represented by an agent
- Initial values generated with *python* script and stored in a *FLAME GPU XML file*
- Road network stored in *CUDA constant memory*
    - Does not change
    - Agents interact with same network
    - *CUDA Read-Only Data Cache* could allow larger road networks (> 64kB of memory)

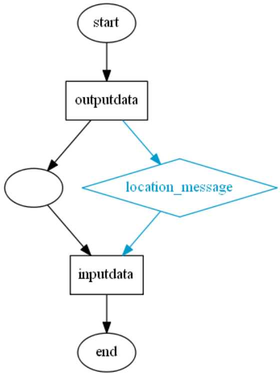


The University Of Sheffield.

Implementing Gipps' Car Following Model using FLAME GPU

For each step in the simulation

- Agents output their observable properties (`outputdata`)
- Agents iterate through their message lists for the lead vehicle (`inputdata`)
    - Gipps' car following model is applied using the lead vehicle information
    - *Forward Euler* used to calculate location and velocity
    - New roads randomly assigned at junctions



Experiments & Results

Experiments, Model Parameters, Hardware

Experiments

	Grid Size	Agent Count	Road Length
Fixed Grid	$N = 16$	256 to 262144	$10000m$
Scaled Grid	$N = 2$ to $N = 24$	512 to 141312	$1000m$

(64 vehicles per $1000m$)

Model Parameters proposed by Gipps

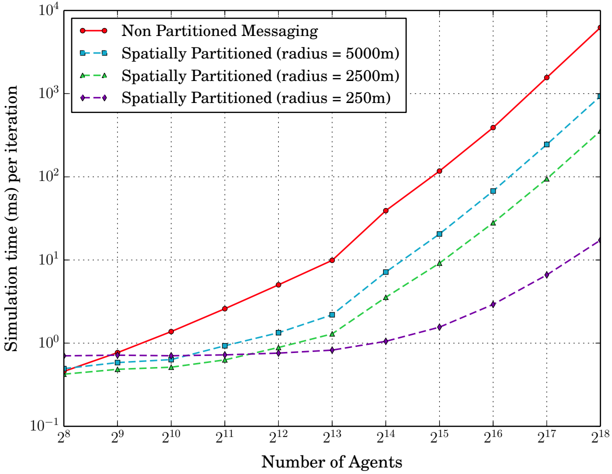
a_n	sampled from the normal distribution $N(1.7, 0.3^2)$ m/sec^2
b_n	$-2.0a_n$
s_n	sampled from the normal distribution $N(6.5, 0.3^2)$ m
V_n	sampled from the normal distribution $N(20.0, 3.2^2)$ m/sec
τ	2/3 seconds
$\hat{b}$	the minimum of -3.0 and $(b_n - 3.0)/2$ m/sec^2

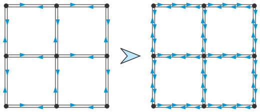
Hardware/Software

- FLAME GPU 1.4 for CUDA 7.0
- Intel Core i7 4770K
- NVIDIA Tesla K20c

The University Of Sheffield.

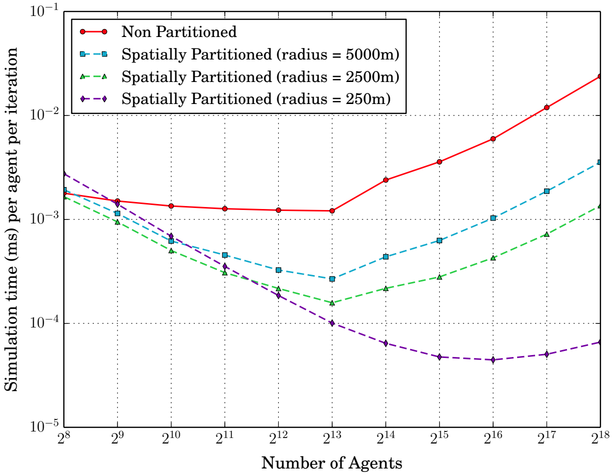
Fixed Grid Network





- Spatially partitioned messaging outperforms non-partitioned messaging
- Smaller radii outperforms larger radii beyond overhead
- Distinct gradient change at 2^{13} agents

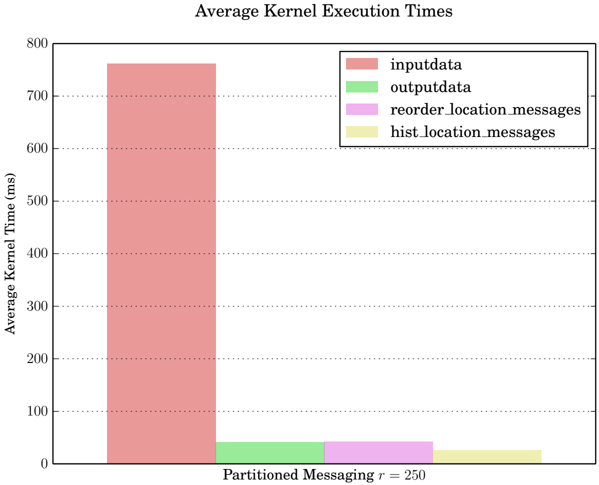
Fixed Grid Network - Per Agent



- Distinct gradient change at 2^{13} agents - **hardware utilisation** vs **larger message lists**
- Non-partitioned outperformed by partitioned messaging
- $r = 250$ scales much better per agent
- Maximum message count

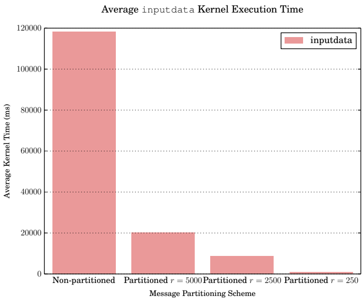
Non-partitioned	262144
Partitioned $r = 5000$	19662
Partitioned $r = 2500$	9720
Partitioned $r = 250$	309

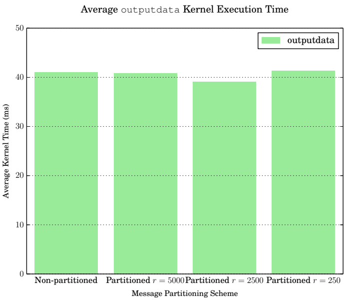
Fixed Grid Network - Kernel Profiling



- Kernel times averaged over 10 iterations
- Some Kernels omitted
- 32768 Agents
- Spatial Partitioned messaging with $r = 250$
- `inputdata` kernel is dominant

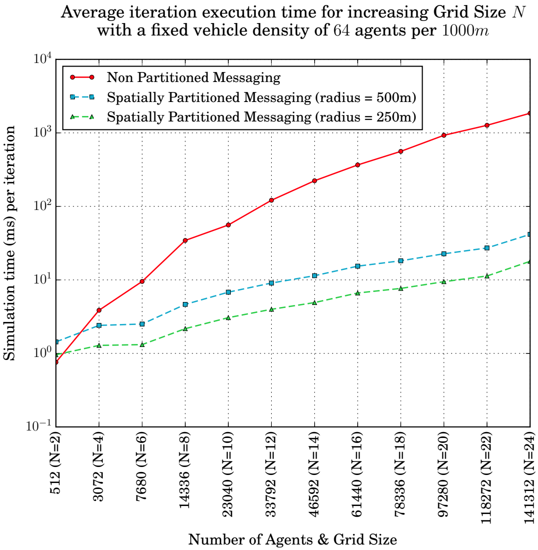
Fixed Grid Network - Kernel Profiling

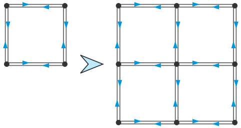




The University Of Sheffield.

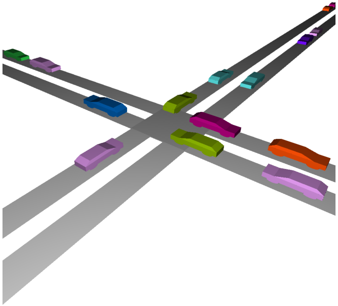
Scaled Grid Network



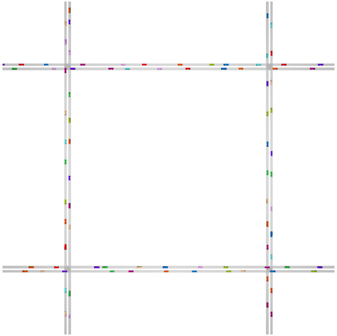


- As scale increases performance decreases
- Spatially partitioned messaging outperforms non-partitioned beyond overhead
- Spatial partitioning scales better
- Up to 103x performance increase for spatial partitioning than non-partitioned

Interactive Visualisation



Nearby



Overview

- Cross platform `C++`, `OpenGL` & `libSDL`[2]
- `OpenGL Interop`[5] & `instanced rendering`[1] used to avoid unnecessary host-device memory transfers
- $N = 8$, length $1000m$, 8192 vehicles & 1000 iterations
- NVIDIA GeForce GTX 660
-

Console	$15079ms$
Visualisation	$16291ms$
Increase	$1.08x$

Conclusions & Future Work

Conclusions

- Two experiments carried out, demonstrating suitability of FLAME GPU for road network simulation
- Scaling behaviour has been investigated
- Performance difference between messaging communication schemes highlighted

Future Work

- Message partitioning techniques for network based communication
- Support wider range of road networks
- Non-uniform vehicle distribution
- Increased accessibility through visualisation of aggregate data on the GPU
- Increased variation of vehicles using procedural instancing

Thank You

ptheywood.uk
ptheywood1@sheffield.ac.uk

flamegpu.com

The University Of Sheffield.

References I

[1] OpenGL SDK glDrawArraysInstanced manpage. https://www.opengl.org/sdk/docs/man/html/glDrawArraysInstanced.xhtml

[2] Simple DirectMedia Layer (libSDL). https://www.libsdl.org/

[3] Gipps, P.G.: A behavioural car-following model for computer simulation. Transportation Research Part B: Methodological 15(2), 105–111 (1981)

[4] Neffendorf, H., Fletcher, G., North, R., Worsley, T., Bradley, R.: Modelling for intelligent mobility (Feb 2015)

[5] Nvidia, C.: Cuda c programming guide. http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf (Mar 2015), last accessed 2015-03-30

The University Of Sheffield.

References II

[6] Richmond, P.: Flame gpu technical report and user guide. Tech. rep., technical report CS-11-03. Technical report, University of Sheffield, Department of Computer Science (2011)

[7] Strippgen, D., Nagel, K.: Multi-agent traffic simulation with cuda. In: High Performance Computing & Simulation, 2009. HPCS'09. International Conference on. pp. 106–114. IEEE (2009)

[8] UK Department for Transport: Quarterly Road Traffic Estimates: Great Britain Quarter 4 (October - December) 2014 (Feb 2015)

[9] Wang, K., Shen, Z.: A gpu based trafficparallel simulation module of artificial transportation systems. In: Service Operations and Logistics, and Informatics (SOLI), 2012 IEEE International Conference on. pp. 160–165. IEEE (2012)