

GPU 기반 LCTM 교통 시뮬레이션에서의 성능 측정

경민기, 신인수, 조민규, 민덕기¹

전국대학교 컴퓨터공학과

e-mail: {moonend, readen, whalsrb1226, dkmin}@konkuk.ac.kr

Time Measurement on GPU-based LCTM Simulation

MinGi KYUNG, In-soo Shin, Min-Kyu Cho, Dugki Min¹

College of Computer Science,

Konkuk University, South Korea

요약

본 연구에서는 메소스코픽 교통 시뮬레이션 모델의 하나인 LCTM(Lane Cell Transmission Model) 모델을 GPU 기반의 병렬 교통 시뮬레이션의 형태로 구현하여, 수행한 시뮬레이션 시간을 측정하였다. 본 논문에서는 LCTM 교통 시뮬레이션의 병렬화 고려사항들을 언급하고, GPU를 사용한 병렬 교통 시뮬레이션 구현 시, 성능에 영향을 미치는 요소들을 분석한 후, 측정하였다.

1. 서론

교통 시뮬레이션[1]은 그 시뮬레이션이 얼마나 자세한 시뮬레이션을 하도록 설정되었느냐에 따라 서로 다른 구현 요소와 다른 요구 조건들, 시뮬레이션 수행을 위한 측정 데이터를 사용하기 때문에 하나의 일 반화된 시간 측정 모델을 제공할 수 없다. 교통 시뮬레이션은 다른 병렬 시뮬레이션 기법과는 달리, 상대적으로 계산량은 적으면서, 한 지역에서 이루어진 시뮬레이션 결과가 다른 지역의 시뮬레이션 결과와 연동되어야 하므로 메모리 접근 시간 및 동기화에 많은 시간이 소요되기 때문이다. 이러한 시뮬레이션(Simulation)은 실험 장비의 비용과 실험 환경의 문제로 인해, 실제 수행하기 어려운 실험이 제한된 자원을 이용하여 행하는 모의실험이기에 실험 환경이 자세할수록, 많은 환경 요소들에 관해 묘사할수록 정확해지기 때문에, 컴퓨터와 GPU의 계산 능력 향상과 메모리 접근 속도 향상이 시뮬레이션의 성능을 높이는 가장 확실한 수단이다.

본 연구팀은 KAIST에서 개발한 CTM 모델[2]을 개량한 메소스코픽 교통 시뮬레이션 모델의 하나인 LCTM(Lane Cell Transmission Model) 모델을 GPU 기반의 병렬 교통 시뮬레이션의 형태로 구현하여 수행한 시뮬레이션 성능 향상을 측정한다. 이를 위해 본 논문에서는 LCTM 교통 시뮬레이션의 병렬화 고려사항들 및 병렬 교통 시뮬레이션 구현 시, 성능에 영향을

미치는 요소들을 분석하는데, 특히 GPU를 구성하는 블록(Block)과 스레드인블록(Thread In Block) 요소가 시뮬레이션 시간에 끼치는 영향을 측정한다.

2. 교통 시뮬레이션 도구들간의 특성 비교

Kai Wang et al.(2012)은 ATS(Artificial Transportation System) 프레임워크를 구축하여 CUDA를 이용한 병렬 교통 시뮬레이션을 수행하였다 [3]. 그리고 Yan Xu et al. (2014)는 오프라인 대규모 시뮬레이션 기반 교통 계획 및 온라인 시뮬레이션 교통 관리를 지원하기 위해 ETSF 프레임워크 및 Mesoscopic Traffic Simulation 프레임워크를 개발 및 적용하였다[4].

이를 통해 연구자들은 GPU가 교통 시뮬레이션을 위한 잠재적인 고성능 플랫폼이 될 수 있는지를 확인하고자 하였고, 그 가능성을 확인할 수 있었다.

하지만 이들 연구진이 개발한 CPU 기반의 교통시뮬레이션 데이터 구조에서 GPU 기반의 교통 시뮬레이션 데이터 구조를 사용하기 위해 링크드 리스트를 배열로 단순히 옮기기만 하는 방법론은 GPU의 메모리 구조 등에 대한 고려가 부족하여, 성능 저하가 발생할 수 있다. 이 때문에 GPU의 메모리 아키텍처를 고려한 시뮬레이션 이론 및 최적화 연구가 필요하다.

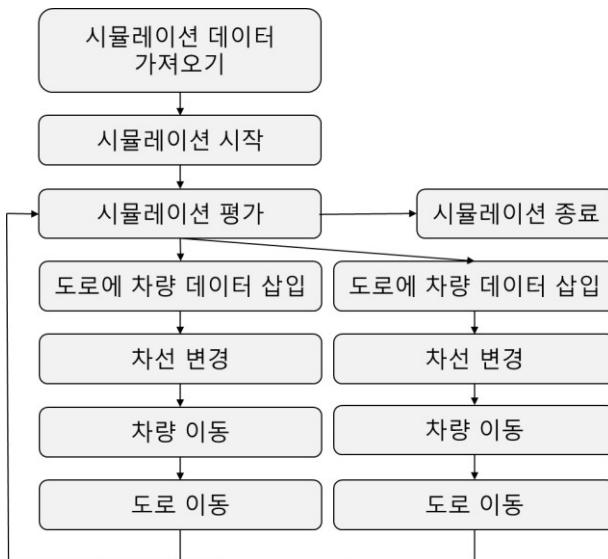
3. LCTM 기반한 교통 시뮬레이션

LCTM (Lane Cell Transmission Model)은 CTM에 차로(Lane)와 차량 에이전트(Vehicle)을 추가하여 차량

¹ 교신저자. 이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. 2017-0-00121, 도시 교통 문제 개선을 위한 클라우드 기반 트래픽 예측 시뮬레이션 SW 기술 개발)

흐름을 더 자세하게 파악하려고 시도하는 메소스코픽 교통 시뮬레이션 (Mesoscopic Traffic Simulation) 모델이다. 이를 이용한 교통 시뮬레이션 처리 순서는 (그림 1)과 같다.

우선 첫 번째로 도로, 신호등, 교차로 데이터 등 시뮬레이션 환경을 구성하는 데이터 및 차량이 도로에 진입하는 순서 및 차량 여행 경로를 포함하는 데이터를 먼저 읽어들이는 작업을 수행한다.



(그림 1) LCTM 기반의 병렬 교통 시뮬레이션 순서

두 번째로 교통 시뮬레이션을 시작하면서, 교통 시뮬레이션을 수행하기 위한 가상 시계에 대한 초기화 작업 및 시뮬레이션 종료 시간에 대한 설정을 수행하고, 여러 GPU 코어에서 서로 다른 도로 환경에 대한 병렬 교통 시뮬레이션이 수행되도록 동작을 시작한다.

세 번째에서 모든 GPU 코어에서 동작하는 교통 시뮬레이션 환경에 대한 평가를 수행하여 더 이상의 교통 시뮬레이션이 가능한지 아닌지를 평가한다.

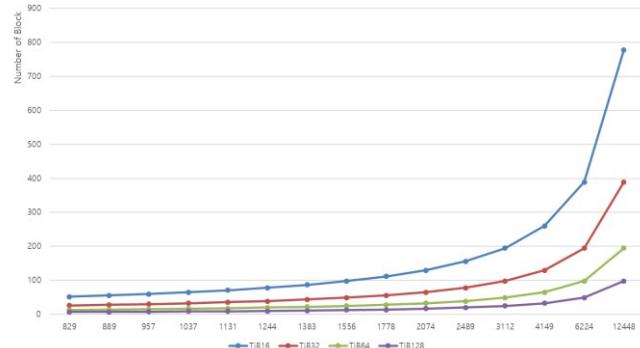
네 번째에서 일곱 번째 과정은 각각의 GPU 코어에서 도로를 키 값으로 이용한 교통 시뮬레이션을 수행하는데, 차량 데이터를 읽어들여 교통 시뮬레이션 환경의 도로 위에 동작하도록 하며, 차량이 도로 안에서 이동하고, 차선 변경을 수행하는 작업 및 도로 사이를 오가는 작업을 수행하게 된다.

이러한 과정을 통해 교통 시뮬레이션을 수행하게 되면, 각각의 도로들이 데이터 공유 및 전달을 위해 GPU의 글로벌 메모리(Global Memory)에 접근하는 과정을 필수적으로 요구하게 된다. 이와 같은 상황에서 교통 시뮬레이션에서 병렬 LCTM 교통 시뮬레이션에 영향을 주는 요소는 교통 시뮬레이션을 수행하는 스레드의 숫자임을 가정할 수 있다.

3.1 최적 교통 시뮬레이션을 위한 GPU 설정 비교

본 논문에서 교통 시뮬레이션은 기본적으로 스레드 1 개가 1 개의 도로 링크를 시뮬레이션하도록 가정하였고, (그림 2)는 이러한 가정하에서 교통 시뮬레이션을 수행할 때, 링크 수에 따른 시뮬레이션에 사용될 최적의 GPU 블록(Block) 숫자와 스레드인블록(Thread In Block) 값을 보여준다.

아래 스레드인블록의 수와 블록 수는 다음과 같은 반비례 관계가 성립된다.

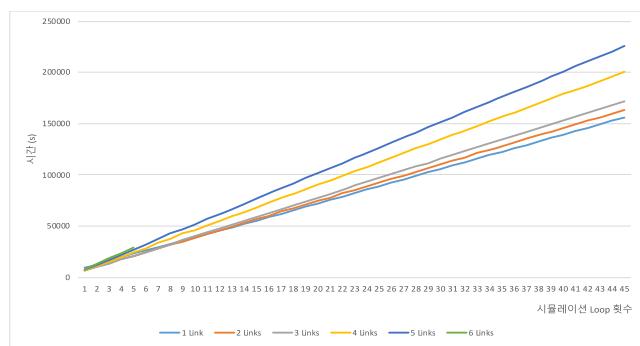


(그림 2) 링크 수에 따른 Block과 TiB (Thread In Block)의 상관관계

교통 시뮬레이션을 수행할 링크 개수가 고정되어 있으면, 스레드인블록의 수가 증가할 때 블록 수는 감소하는 반비례 관계를 맺기 때문에 그 관계를 아래와 같이 표현할 수 있다.

$$\text{Number of Block} = \frac{\text{Number of Link}}{\text{Thread in Block}}$$

GPU 코어 1 개에 링크 1 개를 배정하거나 그 이상을 처리하도록 할 수 있으나, 시뮬레이션의 특성상 매 시뮬레이션 시간 경과 시에 차량 데이터를 입력받아야 하며, 링크의 현재 상태를 알려주어야 하는 등, 메모리 접근 횟수 자체를 줄일 수 없다.



(그림 3) GPU 코어 1 개당 처리할 도로 갯수에 따른 병렬 시뮬레이션 실행시간 차이

(그림 3)에 나온 그래프에서는 스레드인블록이 128

이고, 링크 12,448 개에 대한 시뮬레이션 테스트를 수행한 그래프이다. GPU 스레드 1 개에 링크 여러개에 대한 연산을 수행할 수 있지만, 메모리 접근 시간이 더 증가하기 때문에 링크 1 개를 처리할 때, 시뮬레이션 처리 시간이 제일 적게 나온다.

4. GPU 기반 LCTM 교통 시뮬레이션 성능 실험 결과

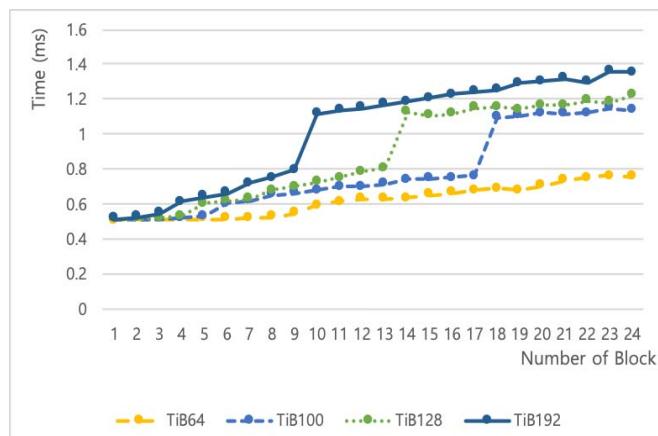
4.1 실험 환경

교통 시뮬레이션을 위해 NVidia GTX 780 기종을 테스트 환경으로 사용하였는데, SM (Streaming Multiprocessor)이 12 개가 있고, SM마다 192 개의 GPU 코어가 존재하고 있어 2,304 개의 GPU 가 동시에 동작할 수 있는 환경을 구성할 수 있다. 이를 이용하여 12,448 개의 링크를 가지고 있는 강동구 지도 데이터를 이용한 교통 시뮬레이션을 수행하였다.

4.2 블록 수에 따른 시뮬레이션 시간 증가

(그림 4)는 스레드인블록(Thread In Block)이 32, 64, 100, 128, 192 일 때, 블록 수가 증가함에 따라 링크 1 개의 시뮬레이션 처리 시간이 어떻게 변하는지를 보여준다.

별별 교통 시뮬레이션 작업을 처리하는 블록 숫자가 늘어남에 따라 링크 간의 데이터 교환에 대한 딜레이 오버헤드가 증가하고 있는 것을 볼 수 있다.



(그림 4) Block과 TiB 값에 따른 링크 1 개당 처리시간

4.3 링크 수에 따른 시뮬레이션 시간 증가

링크의 숫자를 12,288 개에서 1,920,000 개로 변화시키면서 교통 시뮬레이션 및 모델의 링크 1 개 당 시뮬레이션 처리 시간 값을 같이 비교하면 결과가 <표 1>로 나오게 된다. <표 1>은 블록 당 최적의 도로 배치를 위해 스레드인블록(Thread In Block)에서 처리하는 링크 수를 192로 설정했을 때, 블록 수가 증가함에 따라 링크 1 개의 시뮬레이션 처리 시간이 어떻게 변하는지를 보여준다.

<표 1> 링크 수와 블록 수의 증가에 따른 링크처리시간

Link	TiB	# of Blocks	링크 처리 시간
64	192	1	0.51002
1000	192	6	0.636794
2000	192	11	1.111431
3000	192	16	1.193326
4000	192	21	1.31076
5000	192	27	1.423665
6000	192	32	1.487325
7000	192	37	1.800316
9000	192	47	2.00522
10000	192	53	2.122455

시뮬레이션하는 링크 숫자가 증가할수록, 교통 시뮬레이션을 수행할 때는 블록 수를 늘려야만 더욱 적은 시뮬레이션 처리 시간을 보여줄을 확인할 수 있다.

5. 결론

메소스코픽 교통 시뮬레이션 (Mesoscopic Traffic Simulation) 방법인 LCTM 모델을 사용하여 조건에 따른 교통 시뮬레이션 시간 측정을 수행하였다.

이를 통해 본 연구는 LCTM 교통 시뮬레이션의 성능에 영향을 미치는 요소들에 대해 변경하는 작업을 수행할 수 있다. 각 스레드가 GPU 의 글로벌메모리에 접근하는 횟수를 제한하도록 교통 시뮬레이션 알고리즘을 개량하는 방법과 각 스레드에 도로를 배정하는 방식이 아니라 블록별로 지역을 담당하는 도로를 배정하여 스레드인블록 및 블록 수의 증감에 따른 성능 향상 요소들을 찾아내는 방법 등을 시도할 수 있다. 이러한 방법들의 시도를 통해 앞으로 GPU 기반 LCTM 교통 시뮬레이션의 성능 향상을 추구할 수 있다.

참고문헌

- [1] Pursula, M. 1999. SIMULATION OF TRAFFIC SYSTEMS : AN OVERVIEW. Journal of Geographic Information and Decision Analysis.
- [2] Daganzo, C. F. 2015. THE CELL TRANSMISSION MODEL: NETWORK TRAFFIC (pp. 1–22).
- [3] Wang, K., & Shen, Z. (2012). A GPU Based TrafficParallel Simulation Module of Artificial Transportation Systems. Presented at the Service Operations and Logistics, and Informatics (SOLI), 2012 IEEE International Conference on, IEEE., pp. 160–165
- [4] Yang, X., & Recker, W. (2005). Simulation studies of information propagation in a self-organizing distributed traffic information system. Transportation Research Part C: Emerging Technologies, 13(5-6), 370–390.