

GPU 기반 메소스코픽 트래픽 시뮬레이터 설계

김성수^o, 민옥기
한국전자통신연구원

{sungsoo, ogmin}@etri.re.kr

The Design of GPU-Based Mesoscopic Traffic Simulator

Sung-Soo Kim^o, Okgee Min
ETRI

요약

대도시의 교통혼잡으로 인한 사회적 비용이 날로 늘어감에 따라, 복잡한 도로 네트워크상의 교통 혼잡을 예측하거나 교통 시뮬레이션을 통해 보다 효과적인 교통신호 데이터를 구축하는 것이 중요한 과제로 부상하고 있다. 특히, 대규모 도시 전체에 대한 교통 시뮬레이션은 기존의 교통 시뮬레이션 소프트웨어를 통해 수행하는 것은 현실적으로 불가능하다. 따라서, 대규모 교통 시뮬레이션을 지원하기 위해 시뮬레이션의 분산/병렬처리가 필수적이다. 본 논문에서는 대규모 교통 시뮬레이션의 처리 및 성능 개선을 위해 도로 네트워크의 공간 분할 및 각 분할된 도로 네트워크에 대해 병렬 트래픽 시뮬레이션 처리를 수행하는 GPU 기반 메소스코픽 트래픽 시뮬레이터에 대한 시스템 설계를 제안한다.

1. 서론

대도시의 교통혼잡으로 인한 사회적 비용이 날로 늘어감에 따라, 복잡한 도로 네트워크상의 교통 혼잡을 예측하거나 교통 시뮬레이션을 통해 보다 효과적인 교통신호 데이터를 구축하는 것이 중요한 과제로 부상하고 있다. 교통 시뮬레이션 소프트웨어는 교통 전문가, 도시계획 연구자, 그리고 복잡한 교통 역학 (traffic dynamics) 을 이해하기 위해 교통공학 연구자들에게 도움을 주는 도구다[1]. 대표적인 도심 교통 시뮬레이션 공개 소프트웨어로 Simulation of Urban MObility (SUMO)가 있다 [2]. 하지만, 기존의 교통 시뮬레이션 소프트웨어는 컴퓨팅 자원의 한계와 분산/병렬처리 지원이 미흡하여 도시 내 일부 제한된 구역의 주요 교차로들을 대상으로 시뮬레이션 수행이 가능한 것이 현실이다. 이러한 대규모 교통 시뮬레이션을 지원하기 위해서는 분산/병렬처리를 통해 성능을 개선하는 것이 중요한 문제다.

본 논문은 GPU 기반 메소스코픽 트래픽 시뮬레이터의 시스템 구조를 제안하고, 성능 개선 측면에서 기여한 바를 요약하면 다음과 같다.

- **대규모 교통 시뮬레이션 (Massive Traffic Simulation):** 대규모 도로 네트워크를 공간 분할을 통해, 시뮬레이션 태스크를 분산 처리하고 각 분할된 태스크는 GPU를 통해 병렬처리함으로써 전체적인 시뮬레이션을 가속화 할 수 있다.
- **메소스코픽-수준 병렬화 (Mesoscopic-level Parallelism):** 메소스코픽 교통 시뮬레이션을 각 GPU 워커를 통해 수행함으로써, 마이크로스코픽 교통 시뮬레이션과 비교했을 때 유사한 시뮬레이션 결과를 제공하면서 처리 성능을 개선할 수 있다.

이와 같이, 본 논문은 대도시 규모의 교통 시뮬레이션 처리를 가속화하기 위해, 마스터-워커 아키텍처기반의 메소스코픽 트래픽 시뮬레이션 시스템의 구조 설계를 제안한다.

2. 연구 배경

기술적인 측면에서 트래픽 시뮬레이션은 차량 운전자 관점에서의 수요 (demand) 부분과 교통 흐름 관점에서의 공급 (supply) 부분으로 두 단계로 구성된다. 본 논문에서는 트래픽 시뮬레이션에서 공급 부분에 대한 시뮬레이션 분산/병렬처리에 초점을 두고 있다.

교통 시뮬레이션 방법은 크게 마이크로스코픽 (microscopic) 시뮬레이션, 메소스코픽 (mesoscopic) 시뮬레이션, 매크로스코픽 (macroscopic) 시뮬레이션 이렇게 세가지로 구분할 수 있다. 첫번째, 마이크로스코픽 시뮬레이션은 시간이 흐름에 따라 도로 네트워크의 각 개별차량들의 위치를 업데이트함으로써 시뮬레이션을 수행하는 방법이다. 두번째, 메소스코픽 시뮬레이션은 대도시의 교통정책을 수립하기 위해 수행하는 오프라인 대규모 트래픽 시뮬레이션과 온라인 시뮬레이션기반 교통관리를 지원할 수 있는 트래픽 시뮬레이션 기술이다. 일반적으로 메소스코픽 트래픽 시뮬레이터에서 빠른 시뮬레이션 처리 성능을 제공하기 위해, 시뮬레이션 대상 에이전트를 각 차량별로 수행하는 것이 아니라, 특정 단위 도로링크를 나눈 세그먼트 (segment) 단위로 시뮬레이션을 처리한다. 메소스코픽 시뮬레이션을 적용하는 핵심적인 이유는 마이크로스코픽 시뮬레이션과 유사한 정확성을 제공함과 동시에 빠른 시뮬레이션 처리 성능을 제공하는 것이다. 끝으로, 매크로스코픽 시뮬레이션은 도로네트워크의 링크의 평균속도 정보와 같은 교통 흐름정보를 빠르게 계산하는 장점을 제공하지만, 세부적인 도로네트워크의 교통 상황을 표현하기 어렵다는 단점이 있다.

메소스코픽 시뮬레이션 수행에 대한 시간 복잡도는 하나의 타임스텝 (time step) 동안 하나의 차선에 대해 수행하는 경우, 해당 차선에 존재하는 차량의 수가 n 이라고 할 때, $O(n)$ 의 계산 시간이 걸린다. 따라서, 복잡하고 교통이 혼잡한 도로망을 포함하고 있는 대도시 전체를 하나의 머신에서 한번에 트래픽 시뮬레이션을 수행하는 것은 실질적으로 어려운 일이다. 최근에 관련 연구자들이 이

러한 도시 전체에 대한 시뮬레이션을 목표로 고성능 트래픽 시뮬레이션 플랫폼을 제안하고 있다 [3, 4].

본 논문에서는 도로 네트워크상에서 이동하는 많은 차량들에 대한 동적인 교통흐름을 모델링하기 위해, 많은 계산이 요구되는 공급 부분의 성능향상을 위한 GPU기반 메소스코픽 트래픽 시뮬레이션에 대한 병렬처리 설계를 제안한다.

3. 분산/병렬 트래픽 시뮬레이션 시스템 설계

마스터-워커 아키텍처 (Master-Worker Architecture): 대규모 트래픽 시뮬레이션 처리를 위해, 본 논문에서 제안하는 GPU 기반 대규모 교통 시뮬레이션의 분산/병렬처리 흐름도는 그림 1과 같다. 본 논문에서 제안하는 설계에서는 시뮬레이션 단위별 병렬처리를 GPU를 통해 수행하며, 시뮬레이션 수행은 메소스케일 트래픽 시뮬레이션을 병렬처리하는 접근방법으로 설계한다. 기본적인 가정으로 단일 GPU를 사용할 경우, 시뮬레이션 입력은 각 GPU 메모리 크기보다는 작아야 한다. 또한, 대규모 교통 시뮬레이션을 위해서는 복수개의 GPU를 이용하여 수행할 수 있다. 이때, 복수 GPU를 이용하더라도 각 세부 시뮬레이션을 위한 각 GPU 입력 데이터 크기도 각 GPU 메모리 크기보다 작아야 한다.

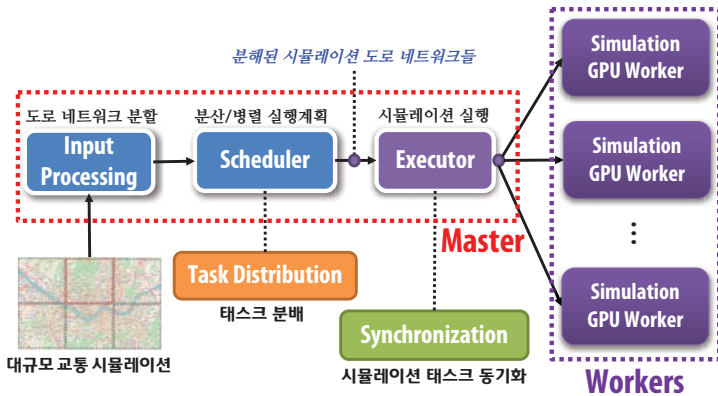


그림 1: GPU기반 대규모 트래픽 시뮬레이션 분산/병렬처리 흐름도

제안하는 구조는 마스터-워커 (master-work) 아키텍처를 따른다. 먼저, 분산 컴퓨팅 환경의 시뮬레이션을 수행하는 각 GPU 워커로 입력을 할당하기 위해서 시뮬레이션 대상인 대규모 도로 네트워크를 분할한다. 공간 분할 (spatial decomposition)된 각 도로 네트워크에 대한 작업할당 및 실행 계획은 스케줄러에서 수행한다. 마스터 노드는 스케줄러를 통해 전달받은 실행계획을 바탕으로 GPU 워커들에게 시뮬레이션 실행을 할당/요청한다.

경계 처리 (Boundary Processing): 각 GPU 워커는 할당받은 도로 네트워크에 대한 트래픽 시뮬레이션을 수행한다. 여기서, 마스터는 전체 도로 네트워크를 분할할 때 생기는 경계 (boundary)에 대한 시뮬레이션을 순차 시뮬레이션 처리결과와 동일하게 제공될 수 있도록 태스크 동기화 처리를 수행한다. 도로 네트워크의 분할한 경계 부분에서 일반적으로 링크 (link)들이 나누어지게 된다. 분산/병렬 시뮬레이션의 결과가 대규모 트래픽 시뮬레이션의 순차 시

뮬레이션 (sequential simulation) 결과와 동일한 결과를 얻기 위해서, 이 경계 부분에 대해 차량들의 이동 계산을 정확하게 처리해 주어야 한다. 각 세부 도로 네트워크와 주변 도로네트워크 사이의 경계에서 발생하는 동기화 처리는 외부 별도 서버를 통해 동기화를 수행한다. 동기화 수행 방법으로는 각 GPU 간의 각 도로 네트워크의 경계 진입 및 이탈에 대한 이벤트를 동기화 서버에 전달하는 방식으로 처리 할 수 있다. 아래와 같이, GPU 이벤트 생성 및 처리 함수를 이용하여 동기화를 수행할 수 있다. 이벤트 생성 함수 (invokeBoundaryEvent)는 입력으로 도로 네트워크상의 링크식별자, 해당 도로를 진입인지 이탈인지 여부, 시뮬레이션 대상 이동객체 정보를 제공하고, 출력으로 이벤트 발생, 처리 완료 등 상태를 리턴한다.

```
SIM_EVENT invokeBoundaryEvent(long linkID,
                               eventType inOut, simInfo info);
```

시뮬레이션 이벤트 동기화 함수 (synchSimulation)의 입력은 이벤트 생성함수와 동일하다.

```
SIM_SYNC synchSimulation(long linkID, eventType
                          inOut, simInfo info);
```

데이터 시뮬레이션 이벤트 처리 함수를 통해, 경계에서 이동하는 이동객체에 대한 시뮬레이션을 서로 다른 GPU 상에 정보를 이동해가며 수행할 수 있다. 분산 처리의 경우에도 이와 같은 방식으로 분산 데이터를 동기화해 주는 역할을 수행하는 서버를 통해 처리가 가능하다. 분산 서버들간의 세부적인 처리를 위해서, 할당 시뮬레이션 태스크와 해당 도로네트워크를 식별하는 코드(권역코드 등)을 해쉬 맵 (hash map) 자료 구조 등을 이용해서 구축한 맵핑한 정보를 생성 후 사용해야 한다. 그림 2는 병렬 메소스케일 시뮬레이션 호스트-커널 구성도를 보여주고 있다. GPU 디바이스에서 수행되는 주요한 커널 함수들로는 메소스코픽 시뮬레이션 수행, 경계에 대한 세그먼트 이벤트 처리, 세그먼트 정보 관리, 세그먼트내 선두 이동 차량을 관리하는 커널 함수가 있다.

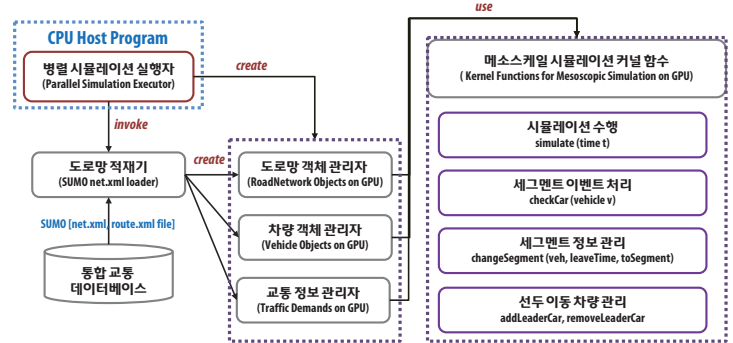


그림 2: 병렬 메소스케일 시뮬레이션 호스트-커널 구성도

각 GPU에서 메소스케일 교통 시뮬레이션을 처리 설계는 CUDA 프로그래밍 모델에 따라, 호스트 부분에 대한 설계와 GPU상에서

수행되는 커널 영역에 대한 설계 부분으로 구성된다. CPU에서 수행되는 호스트 프로그램은 병렬 시뮬레이션 실행자로 GPU 클라우드로 마스터로부터 전달받은 시뮬레이션 작업을 실행하고 결과를 요청한 마스터에게 전달해 주는 역할을 수행한다.

병렬 시뮬레이션 실행자의 주요 처리 절차는 다음과 같다.

1. 할당받은 시뮬레이션 입력을 로드하기 위해 도로망 적재기에 게 처리를 요청한다.
2. 도로망 적재기는 도로네트워크에 대한 정보를 통합 교통 데이터베이스에서 읽어서 적재한다.
3. 2단계에서 적재하면서 GPU 메모리에 전달하기 위한 시뮬레이션 입력 객체들(도로망, 차량, 교통 정보 등)을 생성한다.
4. 생성한 시뮬레이션 입력 객체를 GPU 메모리에 복사한다.
5. 메소스케일 교통 시뮬레이션 커널 함수를 이용하여, 시뮬레이션을 수행한다.
 - 시뮬레이션 수행시, 분할된 도로네트워크상에 이벤트 발생 시, 이벤트 생성 함수를 호출한다. (invokeBoundaryEvent)

시뮬레이션 시스템의 각 GPU 워커 노드에서 수행하는 시뮬레이션 설계를 위한 기본적인 아이디어는 Yan Xu 등이 제안한 메소스코픽 트래픽 시뮬레이션 방법을 참조하였다 [5]. 그림 3은 GPU 내에 적재 가능한 도로 네트워크 객체들; 도로 네트워크, 노드, 링크, 세그먼트, 차선, 차량 등의 클래스 다이어그램을 보여주고 있다. GPU 메모리상의 연속적인 접근을 위해 각 객체들을 배열로 정의한 후, 디바이스 메모리 영역에 복사한다. 본 설계에서는 도로 네트워크와 노드, 링크, 세그먼트, 차선을 기본적으로 정의하였으며, 차량객체에 대한 인덱스 정보를 가지고 있다. 디바이스 메모리에 생성된 객체들은 메소스케일 시뮬레이션 커널 함수들의 입력으로 제공되고 GPU를 통해 병렬 메소스코픽 시뮬레이션이 수행된다.

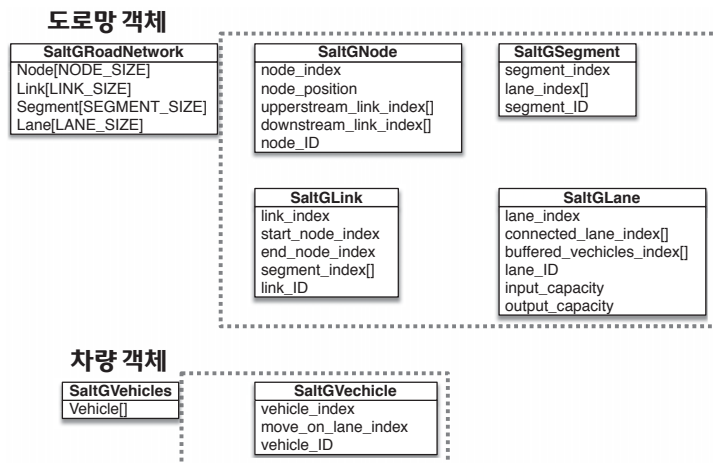


그림 3: 도로 네트워크 객체와 차량 객체에 대한 클래스 다이어그램

4. 결론

본 논문은 대규모 교통 시뮬레이션의 처리 및 성능 개선을 위해, 대규모 도로 네트워크의 공간 분할 및 각 분할된 도로 네트워크에 대해 병렬처리를 수행하는 GPU 기반 메소스코픽 트래픽 시뮬레이터에 대한 시스템 설계를 제안하였다. 제안한 시스템은 마스터-워커 아키텍처를 기반으로 분할된 시뮬레이션 태스크를 분산하고, 할당된 태스크는 각 GPU 워커상에서 병렬화하는 방식을 적용하였다.

제안한 GPU 기반 메소스코픽 트래픽 시뮬레이터를 사용하면, 도시계획 전문가, 도로정책 수립자등과 같은 사용자는 트래픽 시뮬레이션을 보다 빠르게 수행할 수 있는 장점을 제공한다. 그리고, 초기 각 소규모 지역별 트래픽 시뮬레이션을 수행하는 경우, 병렬화된 GPU 기반 메소스코픽 트래픽 시뮬레이션 엔진을 통해 효과적으로 수행할 수 있는 장점을 제공한다.

감사의 글

본 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (2017-0-00121, '도시 교통 문제 개선을 위한 클라우드 기반 트래픽 예측 시뮬레이션 SW 기술 개발')

참고 문헌

- [1] P. Toshniwal, M. Hanai, and E. S. Liu, "Towards a Benchmark for the Quantitative Evaluation of Traffic Simulators," in *Proceedings of the 2017 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation, SIGSIM-PADS '17*, pp. 259–262, ACM, 2017.
- [2] SUMO, "http://www.sumo.dlr.de/userdoc/index.html," 2017.
- [3] M. Hirabayashi, S. Kato, M. Edahiro, and Y. Sugiyama, "Toward GPU-accelerated Traffic Simulation and Its Real-Time Challenge," in *Proc. of the International Workshop on Real-Time and Distributed Computing in Emerging Applications*, 2012.
- [4] T. Suzumura, G. McArdle, and H. Kanezashi, "A High Performance Multi-modal Traffic Simulation Platform and Its Case Study with the Dublin City," in *Proceedings of the 2015 Winter Simulation Conference, WSC '15*, pp. 767–778, IEEE Press, 2015.
- [5] Y. Xu, G. Tan, X. Li, and X. Song, "Mesoscopic Traffic Simulation on CPU/GPU," in *Proceedings of the 2nd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation, SIGSIM PADS '14*, pp. 39–50, ACM, 2014.