

효율적인 시뮬레이션 정보 수집을 위한 실험 환경 설계

최창범*, 김정훈*, 김탁곤*

Designing Experimental Frame for Efficient Simulation Data Collection

Changbeom Choi, Jeong Hoon Kim, Tag Gon Kim

Abstract

Due to the multi-core system has come into wide use, the simulation software users perform a simulation in a multi-core system. Accordingly, the research on performing a simulation while utilizing the characteristics of the multi-core system is making progress.

This paper proposes an design of Experimental Frame to collect simulation data from existing simulation software. Proposed Experimental Frame enables the batch simulation of standalone/interoperable simulation software based on the process of simulation software. Moreover, the Experimental Frame reduces the total execution time of batch simulation, and manages the execution of standalone/interoperable simulation software without user's intervention.

Key Words : 멀티코어 시뮬레이션, 실험 환경

1. 서론

모델링 및 시물레이션(Modeling & Simulation: M&S) 공학은 수많은 산업 분야에서 핵심적인 역할을 맡아 제품 개발 및 수요 예측 등을 통하여 비용을 절감시키며 유체 역학 시물레이션과 전투 실험과 같은 다양한 학문 분야에서도 연구비용을 감소하는 효과를 가져왔다[1]. 이러한 시물레이션 소프트웨어는 모의 대상을 실세계를 최대한 모사하기 위하여 연속 시간 모델링, 이산 시간 모델링, 이산 사건 모델링 등의 방법들을 사용하여 모의 대상을 모의하였다. 하지만, 실세계를 반영한 시물레이션 모델들은 복잡하여 표현하기 어려우며, 표현하더라도 시물레이션 수행 시간이 길어 시물레이션 소프트웨어를 활용하기 어렵다. 또한, 시물레이션 소프트웨어를 사용하여 데이터를 수집하는 경우 몬테 카를로스 시물레이션과 같은 기법을 사용하여 시물레이션 데이터를 수집하면 전체 시물레이션 소요 시간은 시물레이션 소프트웨어의 수행 시간의 n 배가 된다. 이와 같은 이유로 대부분의 시물레이션 모델들은 실세계 모델들을 추상화하여 모델링하여 시물레이션을 수행한다. 그러나 추상화된 모델을 수행함으로써 시물레이션의 수행 시간을 감소하지만 추상화로 인한 시물레이션 결과의 부정확성은 증가하는 단점이 존재한다. 이러한 모의 대상의 수준에 따른 시물레이션 수행시간과 시물레이션의 정확성 문제를 해결하기 위하여 분산 처리 시스템을 이용한 시물레이션과 멀티코어 시스템을 활용한 시물레이션이 있다.

분산 처리 시스템은 오랜 시간동안 연구되어 온 학문으로 여러 개의 컴퓨터 시스템들이 네트워크를 통해 서로 연결되어 마치 하나의 시스템처럼 동작하는 시스템이다. 분산 시물레이션 시스템은 미들웨어를 통하여 모델 상호간의 메시지 전송 및 시물레이션 시간 동기화를 위한 인터페이스가 필요하고, 이들을 처

리하며, 이러한 특징으로 인하여 확장성과 모듈성을 가진다[2]. 하지만, 분산 처리 시스템은 오랫동안 연구되어왔음에도 불구하고 하나의 모델이 분산된 시스템에서 동작하게 만들기 위한 분산 알고리즘이 어려워 하나의 모델을 분산 시스템으로 만들기 어렵고, 개별 모델들을 각각 분산 시스템의 서브시스템에 사상하여도 네트워크의 성능이 컴퓨터의 성능에 따라가지 못해서 성능이 저하되는 문제점을 가진다.

이에 비해 멀티코어 시스템을 활용한 시물레이션은 최근 멀티코어 시스템의 보급으로 일반 개인이 사용하는 컴퓨터에서도 멀티코어 시스템이 보편화되고 있으며, 네트워크를 통한 분산처리가 아닌 CPU 내부에서 다수의 CPU코어에서 동작을 수행함으로써 네트워크의 부하 없이 시스템을 성능을 증가시킬 수 있다. 하지만, 기존의 분산 시스템과 마찬가지로 하나의 시물레이션 모델을 다수의 코어에 사상시키기 어려워, 보편화되고 있는 수행 환경에 반해 시물레이션 소프트웨어는 멀티코어 시스템을 거의 지원하지 않지 않다. 또한 기존에 개발된 시물레이션 소프트웨어들은 대부분 싱글코어 기반으로 설계 및 구현되었기 때문에 이를 멀티코어 기반으로 변경시키기 어렵다.

본 연구에서는 멀티 코어 기반 환경에서 멀티코어Frame한 사용하여 시물레이션을 효율적으로 관리하는 실험 환경(Experimental Frame)을 제안한다. 실험 환경의 개념은 B.P. Zeigler에 의해 처음 제안되었다[3]. 본 연구에서는 Zeigler에 의해서 제안된 실험 환경을 확장하여 멀티코어 환경에서 하나의 시물레이션 소프트웨어가 하나의 목적을 위하여 시물레이션을 수행하는 단독형 시물레이션과 다수의 시물레이션 소프트웨어가 하나의 목적을 위하여 시물레이션을 수행하는 연동형 시물레이션

을 제어한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 제시하고, 3장에서는 제안하는 실험 환경을 소개한다. 4장에서는 제안하는 실험환경을 사용한 실험 결과를 소개하고 각 실험 결과를 분석한다. 마지막으로 5장에서는 결론 및 향후 연구를 제시한다.

2. 관련연구

멀티코어 시스템을 활용한 시뮬레이션은 크게 GPU의 코어를 활용한 시뮬레이션과 CPU의 코어를 활용한 시뮬레이션으로 나눌 수 있다.

GPU를 활용한 시뮬레이션으로 대표적인 것이 NVIDIA 사의 그래픽 카드를 활용한 CUDA(Compute Unified Device Architecture)가 있다[4]. 시뮬레이션 모델 개발자들은 CUDA는 개발자들로 하여금 CUDA GPU안의 병렬 계산 요소의 고유 명령어 집합과 메모리에 접근할 수 있게 만들어, 각 시뮬레이션 모델이 GPU의 코어에 할당되어 시뮬레이션을 수행할 수 있게 된다. GPU를 이용한 시뮬레이션은 다수의 GPU 코어를 사용하여 병렬처리를 수행할 수 있는 장점이 있지만, CPU 코어에서 동작하는 모듈과 GPU 코어에서 동작하는 모듈 사이의 통신과 GPU 코어 내의 통신이 빈번한 경우 이들을 처리해야 하는 오버헤드가 존재한다.

다수의 CPU 코어를 사용하는 시뮬레이션의 경우는 모의하고자 하는 대상의 모델을 다수의 코어에 할당하는 방법이나 시뮬레이션 모델과 시뮬레이션과 관련 없는 모듈을 분리하여 사용하는 방법들이 있다. 즉, 네트워크를 관리하는 네트워크 쓰레드와 로그를 남기는 로그 쓰레드를 별도의 코어에서 동작시키는 방법들이 활용되었다.

3. 정보 수집을 위한 실험 환경

실험 환경(Experimental Frame)의 개념은 B. P. Zeigler에 의해서 처음으로 제시되었다 [3]. B. P. Zeigler가 제시한 실험환경은 Generator와 Acceptor, 그리고 Transducer로 이루어져 있다. Generator는 실험 대상 시스템에 주어질 입력을 생성하는 역할을 담당한다. 예를 들어 Generator는 실험 대상 시스템에 대하여 실험할 입력 패턴을 생성하여 실험 대상 시스템에 입력을 넣는 모델이다. Transducer는 시스템에 입력된 입력 값에 대한 출력에 대하여 분석할 수 있도록 출력 값을 변경시킨다. Acceptor는 Generator에서 입력한 값에 따라 Transducer에서 받은 출력 값이 사용자가 원하는 조건을 만족하였는지를 확인하는 모델이다. 즉, 모의 대상 시스템이 입력 값에 따라 생성한 출력 값이 사용자의 요구사항을 만족하는 지를 검사하는 모델이다.

제안하는 실험 환경은 다음 그림과 같다.

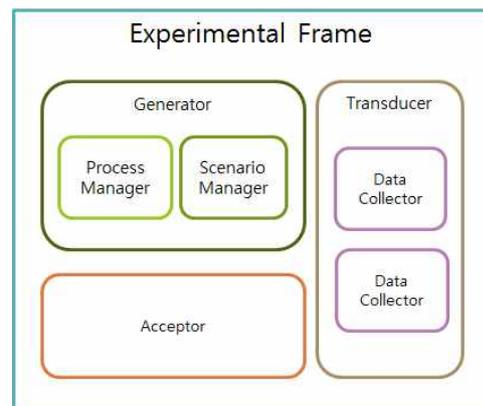


그림 1 정보 수집을 위한 실험 환경

그림1의 정보 수집을 위한 실험 환경은 B. P. Zeigler가 제시한 실험 환경과는 달리 다수의 시뮬레이션 모델과 상호 작용하며 시뮬레

이션을 실행하고, 데이터를 수집하는 전 과정을 관리한다. 제안하는 실험환경은 기본적으로 B. P. Zeigler가 제안한 실험 환경과 같은 형태로 Generator, Transducer와 Acceptor를 가진다.

Generator 모델은 시물레이션 모델의 프로세스를 생성하는 프로세스 관리자와 생성된 시물레이션 모델에 입력할 입력 값의 시퀀스를 관리하는 시나리오 관리자로 구성된다. Generator 모델의 프로세스 관리자는 멀티코어를 활용하여 시물레이션 모델을 실행시킨다. 프로세스 관리자는 사용자가 정의한 설정 값에 따라 시물레이션 모델을 실행 환경의 코어에 맵핑시켜 시물레이션을 관리한다. 시나리오 관리자는 몬테 카를로스 시물레이션과 같이 특정 변수 값의 변화에 따른 시물레이션을 수행하기 위하여 현재 시물레이션의 시나리오를 관리하며 전체적인 시물레이션 횟수를 관리한다. 이 수행되는 실행 시나리오 관리자는 몬테 카를로스 시물레이션과 같이 값을 변경시키며 시물레이션을 수행시킬 수 있도록 현재 진행되고 있는 시물레이션의 시나리오 값과 전체 시물레이션의 시나리오 값을 관리한다.

Transducer 모델은 시물레이션 모델로부터 생성된 출력 값을 받아들이기 위한 Data Collector를 가진다. 제안하는 정보를 수집하기 위한 실험 환경은 단독형 시물레이션 소프트웨어와 연동형 시물레이션 소프트웨어를 고려하여 데이터를 수집하는 다수의 Collector 모델을 가질 수 있다. 즉, 복수의 시물레이션 소프트웨어가 동시에 실행되는 연동형 시물레이션의 경우 각 시물레이션 소프트웨어의 출력값을 처리하기 위하여 Transducer 모델은 시물레이션 소프트웨어들의 고유 출력 포맷을

처리하는 Data Collector를 가질 수 있다.

Acceptor 모델은 정보를 수집한 후 이를 처리하는 후처리 과정을 담당한다. 즉, 엑셀을 사용한 그래프 생성 혹은 생성된 출력 값들의 평균, 실행 시간의 측정 등, 시물레이션 모델이 만족해야 하는 조건이나 시물레이션이 종료된 이후에 처리해야 하는 과정을 처리한다.

4. 사례 연구

정보 수집을 위한 실험 환경은 다른 시물레이터와 연동없이 홀로 동작하는 단독형 시물레이션 소프트웨어와 HLA/RTI와 같은 미들웨어를 사용하여 타 시물레이션 소프트웨어와 연동하여 시물레이션을 수행하는 연동형 시물레이션 소프트웨어에 적용할 수 있다. 본 장에서 단독형 시물레이션 소프트웨어와 연동형 시물레이션 소프트웨어에 정보 수집을 위한 실험 환경을 적용하여 실험한 결과를 분석한다.

4.1 단독형 시물레이션

제안하는 정보 수집용 실험 환경은 단독형 시물레이션 소프트웨어의 변수값을 변경하면서 반복적으로 수행하는 경우 실험 환경 내의 프로세스 매니저와 시나리오 관리자를 사용하여 전체 시물레이션 시간을 단축시킬 수 있다. 즉, 동일한 시물레이션 소프트웨어에 대하여 시나리오 관리자가 서로 다른 시나리오 정보를 주고, 프로세스 관리자에서 시물레이션 소프트웨어를 복수 개를 수행함으로써 전체 시물레이션 시간을 단축시킬 수 있다. 단독형 시물레이션 소프트웨어의 수행시 시물레이션 정보 수집을 위한 실험환경의 효과를 측정하기 위하여 이차원으로 구성된 환경 모델에서 수상함의 생존율을 측정하는 단독형 시물레이

션을 이용하였다[5].

다음 그림2는 실험 환경을 사용하여 단독형 시뮬레이션을 20회 수행하였을 경우 시뮬레이션의 종료 시까지 걸린 시간을 측정한 결과이다.

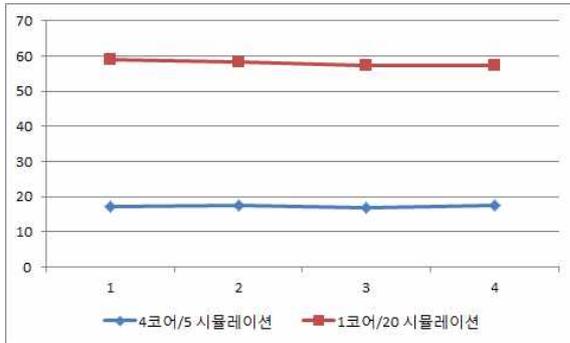


그림 2 실험 환경을 사용한 시뮬레이션 실험 결과

위 그림에서 확인할 수 있듯이 1 개의 코어를 사용하여 20회의 시뮬레이션을 수행한 경우와 코어를 4개를 사용하여 시뮬레이션을 수행한 경우와 비교하여 대략 3.34배 정도 시뮬레이션 수행시간을 단축시킴을 확인할 수 있다. 이상적으로 총 4개의 코어를 사용하여 시뮬레이션을 수행하는 경우 4배의 성능향상을 볼 수 있어야 하지만 실제 시뮬레이션 소프트웨어를 각 코어에 할당하는 등 시뮬레이션 초기화를 수행하는 시간으로 인하여 이상적인 성능 향상이 발생하지는 않음을 확인할 수 있다.

4.2 연동형 시뮬레이션

연동형 시뮬레이션에서 시뮬레이션 정보를 수집하기 위한 실험 환경은 프로세스 관리자에서 다수의 시뮬레이션 소프트웨어를 수행할 때 각 시뮬레이션 소프트웨어를 멀티코어의 서로 다른 코어에 배치시키며 전체 시뮬레이션을 시나리오 관리자가 관리함으로써 시뮬레

이션의 수행시간을 감소시킨다. 연동형 시뮬레이션에서 시뮬레이션 정보 수집을 위한 실험 환경의 효과를 측정하기 위하여 수상함의 생존율을 측정하는 연동형 시뮬레이션 소프트웨어를 사용하였다[6].

다음 그림은 실험 환경을 사용하지 않고, 사용자가 연동 시뮬레이션을 수행하는 시간을 측정한 것이다.



그림 3 수동 연동 실험 및 EF 실험의 비교

위 실험은 사용자가 수동으로 시나리오와 시뮬레이션에서 사용될 랜덤 시드값을 입력하여 처음 시뮬레이션을 수행하였을 때부터 시뮬레이션의 종료될 때까지의 시간을 측정하였다. 이와 반대로 실험 환경을 사용한 경우 코어를 지정하지 않고, 시뮬레이션을 수행하여 각 시뮬레이션의 시간을 측정한 것이다. 위 실험에서 시나리오 1번의 시드 2의 경우와 시나리오2번의 시드 3번의 경우 수동으로 실험한 것이 실험 환경을 사용한 것보다 좋게 나온 것을 확인할 수 있다. 실험 환경을 사용한 연동형 시뮬레이션은 전체 시뮬레이션에서 시뮬레이션 소프트웨어가 종료되는 시간과 초기화하는 시간을 프로세스 관리자가 관리하는데 각 시뮬레이션에 따라 그 종료 시점과 초기화 시점이 불분명하기 때문에 일정시간 기다려야

한다. 이에 반해 사용자가 수동으로 하는 경우에는 종료 시점 혹은 초기화 시점에 즉시 시물레이션을 수행할 수 있기 때문에 몇몇 경우 시물레이션을 빨리 끝낼 수 있었다.



그림 4 멀티 코어 지정 및 미지정 실험

그림 4는 실험 환경을 사용하여 연동형 시물레이션을 수행하는 경우 프로세스 관리자에 시물레이션 소프트웨어를 다양한 경우를 고려하여 코어에 개별적으로 수행하도록 할당한 경우와 프로세스 관리자를 사용하지 않고, 운영체제에서 자체적으로 프로세스를 관리하는 경우와 비교하였다. 연동형 시물레이션에서 코어의 개수 이상의 시물레이션 소프트웨어를 실행시켜야 하기 때문에 시물레이션 수행 중 부하가 큰 프로세스를 개별 코어에 할당하여 시물레이션을 수행하였을 경우 그렇지 않은 경우와 비교하여 시물레이션 성능이 향상됨을 확인할 수 있다. 또한 실제 부하가 크지 않은 프로세스를 개별 프로세스에 할당한 경우에는 몇몇 경우에서 시물레이션 성능이 저하되는 것을 확인할 수 있었다.

5. 결론 및 향후 연구

본 논문은 멀티코어 환경에서 다양한 시물레이션 도구들을 사용하여 시물레이션을 수행하였을 경우 멀티코어 자원을 최대한 사용하여 시물레이션의 성능을 향상시키는 방법으로 시

물레이션 정보를 수집하기 위한 실험 환경을 제안하였다. 제안하는 실험 환경은 각 시물레이션 소프트웨어의 프로세스를 개별적으로 관리하여 코어에 할당하여 시물레이션을 수행하고 그 결과를 분석할 수 있도록 한다.

Acknowledgements

본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2010년도 문화콘텐츠산업기술지원사업의 연구결과로 수행되었음

참고문헌

- [1] 김탁곤, "모델링 시물레이션 공학(M&S Engineering)," 정보 처리 학회지, Vol.14, No. 6, pp. 3-17, 2007년 11월
- [2] Ajay, D. Kshemkalyani, Mukesh Singhal, *Distributed Computing: Principles, Algorithms, and Systems*, Cambridge University Press, 2008
- [3] Bernard P. Zeigler, Herbert Praehofer and Tag Gon Kim, *Theory of Modelling and Simulation (2nd Edition)*, Academic Press, 2000.
- [4] 석문기, 김탁곤, "GPGPU를 활용한 PDEVS 시물레이터 개발 방법론," 한국시물레이션학회 '10 추계학술대회, 2010년 5월.
- [5] 최창범, 김정훈, 김탁곤, "이차원 배열 DEVS 모델로 표현된 에이전트 기반 국방 M&S 기법에 관한 연구," 한국군사과학기술학회 '10 종합학술대회, pp. 368 - 371, 2010년 6월.
- [6] 김정훈, 최창범, 문일철, 김탁곤, "임무-교전급 시물레이션 모델의 연동을 통한 전투실험: 함대 대공 방어 효과도 분석 사례," 한국군사과학기술학회 '10종합학술대회, pp. 2022 - 2025, 2010년 6월.