

웹 기반 시물레이터의 구현

김중은, 조경산
단국대학교 전산 통계학과
e-mail:semico,kscho@dankook.ac.kr

Development of a Web-Based Simulator

Jongun Kim, Kyungsan Cho
Dept. of Computer Science & Statistics, Dankook University

요약

웹은 지난 수년간 급속도로 발전하였으며 웹의 다양한 활용 분야 중에서 시물레이션은 웹의 특성을 가장 잘 이용하는 분야 중 하나로, 웹 기반 시물레이션의 구현에 대한 연구가 활발히 이루어지고 있다. 또한 Java 언어의 출현은 웹에서 실질적인 애니메이션과 애니메이션들간의 상호동작을 가능하게 한다.

웹 기반 분산 시물레이션은 웹의 분산 특성과 자바의 객체지향 특성을 이용한 분산 시물레이션이다. time-warp 기법을 사용하는 웹 기반 분산 시물레이션에서 speedup에 대한 성능은 rollback과 통신 지연이 가장 중요한 요인이다. rollback이 발생한 경우 시물레이션을 다시 수행하여 시물레이션을 매우 느리게 한다. 이러한 rollback과 통신 지연의 방대한 오버헤드는 시물레이션 모델의 지역적 분할을 사용할 때 발생한다. 본 발표에서는 time-warp을 기본 구조로 자바의 RMI를 사용하는 웹 기반 분산 시물레이션에서 통신 지연에 의한 오버헤드 및 거대한 병렬성과 분산을 고려한 시물레이션의 구현 모델을 제안하고 구현한다.

I. 서론

컴퓨터 시물레이션은 실제 또는 이론적인 물리 시스템의 모델을 설계, 모델의 실행하고 실행 결과를 분석하는 학문이다. 웹 기반 시물레이션은 웹에서 컴퓨터 시물레이션 방법들과 응용들에 대한 변화를 제시한다. 즉, 웹 기반 시물레이션은 시물레이션 모델의 접근 방법뿐만 아니라 시물레이션 모델들의 생성, 실행, 배포를 위한 플랫폼으로써 웹의 사용을 의미한다.

웹에서의 분산은 웹 문서를 이용하고 웹의 문서는 정적, 동적, 활성의 3가지 유형으로 구현될 수 있다.

제안된 웹 기반의 시물레이션의 구현은 자바 언어의 애플릿을 이용한 활성 문서로 구현한다[조

97]. 자바의 실행 환경은 인터넷을 통한 소켓 기능 및 RMI를 제공한다. 기존의 분산 시물레이션은 분산 및 병렬처리하기 위해 멀티 프로세서 시스템이나 또는 한 네트워크에 연결된 동일 운영 구조를 사용하였으나, 웹 기반 분산 시물레이션은 자바를 지원하는 웹 브라우저를 갖고 인터넷에 연결된 모든 컴퓨터에서 실행이 가능하다 [Bu96][Fi96].

웹 기반 분산 시물레이션은 클라이언트-서버 모델을 기본 구조로 한다. 시물레이션 모델의 방대한 지역적 분할과 분할된 시물레이션 개체 사이의 빈번한 메시지 교환은 서버에 통신 부하를 가

중시켜 큰 통신 지연을 발생시키나, 웹 기반 시뮬레이션이 방대한 시뮬레이션 모델을 수행하기 위해서는 방대한 분산과 병렬성을 가져야한다 [Am92].

본 발표에서는 다양한 클라이언트에서 분산과 병렬 처리가 가능한 웹 기반의 분산 시뮬레이션 모델을 제시하고 구현한다. 이를 위하여 II장에서는 웹 기반 분산 시뮬레이션의 구현 환경과 사례를 기술한다. III장은 낙관적 time-warp 기법을 이용하고 시뮬레이션 모델의 지역 분할을 고려한 시뮬레이션 구조와 구현 특성을 설명한다. IV장에서는 실제 구현 시뮬레이터에서 시뮬레이션을 수행하여 그 활용성을 제시하고자 한다.

II. 관련 연구

인터넷에 연결된 많은 컴퓨터들을 이용한 웹 기반의 분산 시뮬레이션의 구현에 대한 연구가 활발하다. 본 장에서는 웹 기반 분산 시뮬레이션 구현 언어로서 자바의 특성, 비동기식 분산 시뮬레이션의 time-warp 기법 및 웹 기반 분산 시뮬레이션의 구현 사례를 설명한다.

1. 웹 기반 시뮬레이션 구현 환경[조98]

웹 기술은 시뮬레이션 모델들의 개발, 실행, 분석과 문서화를 위한 방법으로 강력한 기술이다. 또한, Java 언어의 출현은 웹에서 실질적인 애니메이션과 애니메이션들간의 상호동작을 가능하게 한다.

자바는 개체 지향 언어이고 자바의 애플릿은 웹에서 활성 문서를 제공한다. 자바와 웹의 특성을 이용한 분산 객체간의 통신은 자바의 RMI를 통해 구현될 수 있다. 자바의 RMI는 기존의 분산처리에서 활용하던 RPC의 객체 지향적 구현이다. RMI를 활용한 자바의 분산 객체 모델은 마스터-슬레이브 구조를 근간으로 구현하며, 그림 1과 같은 세

개의 계층으로 구성된다. 자바의 RMI는 인터페이스를 통해 원격 객체의 메소드를 호출하도록 한다. 원격 참조 계층은 원격 객체 참조에 대한 인터페이스며, 트랜스 포트 계층은 클라이언트와 서버간의 연결을 설정/관리하고 원격호출을 원격객체에게 전달한다 [Su96].

RMI 구조에서 마스터-슬레이브간의 전송횟수는 마스터의 통신 부하에 큰 영향을 미친다. RMI를 사용하는 웹 기반 분산 시뮬레이션은 방대한 분산과 빈번한 원격 메소드 호출은 서버의 통신 지연에 대한 오버헤드를 증가시켜 시뮬레이션 수행을 느리게 한다.

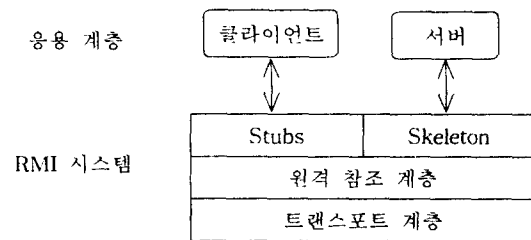


그림 1. RMI 구성

2. Time Warp 기법

비동기 분산 시뮬레이션은 보수적 기법과 낙관적 기법이 적용되어 왔다. 보수적 기법은 분산 처리되는 모든 사건들이 시간 순서로 수신되고 처리되도록 보장한다. 낙관적 기법은 실행 효율성을 위해 분산된 프로세스들 사이의 메시지 전달은 시간 순서가 어긋날 수 있도록 허용하고 프로세스들은 내부적으로 사건들을 시간 순서대로 처리한다. time-warp는 낙관적 분산 시뮬레이션의 가장 보편적인 구현 기법이다[Pr90].

time-warp 기법에서 사건들은 내부 사건과 외부 사건으로 분리된다. 프로세스는 사건들을 시간 순서로 처리하기 위해 rollback을 사용한다. rollback은 시뮬레이션 진행 시간 이전에 처리되

어야하는 외부 사건 메시지 또는 처리된 사건을 취소하는 외부 사건 메시지가 도착하면 발생한다. rollback을 위하여, 각 프로세스는 수신된 외부 사건들에 대한 정보, 전송한 외부 사건들에 대한 정보와 사건 처리에 대한 상태 정보를 유지해야 한다[Pr90][Am92].

3. 웹 기반 시물레이터의 구현 사례

최근에 웹의 분산적 특성과 자바의 특성을 이용한 다음과 같은 웹 기반 시물레이터들이 구현되었다.

Simjava는 자바로 작성된 애니메이션 기능을 프로세스 기반의 이산 사건 시물레이터로서, 복잡한 시스템 모델을 설계할 수 있는 애플릿 단위의 클래스들로 구성되어있다. 시물레이션 프로그램은 중앙 제어되는 각각의 스레드로 실행되는 엔티티들로 구성된다[Bu96].

JSIM은 애니메이션 상속, 시물레이션을 위한 이벤트 스케줄링 방법, 프로세스 상호동작을 위한 패키지 라이브러리이다. 즉, JSIM은 자바 기반의 시물레이션과 애니메이션 환경을 제공한다. 시물레이션 모델과 시물레이션 결과를 저장하기 위한 시물레이션 클래스들은 데이터베이스와 상호동작하기 위해 연결된다[Na96]. 또한, 이산 사건 시물레이터인 simkit, DIS 시물레이터등이 제안되었다[조98].

III. 시물레이터 구현

본 연구에서는 기존 분산 시물레이션의 실행 환경의 제한 및 웹 기반 시물레이션의 미약한 분산과 병렬처리의 단점을 보완하기 위하여, 비동기 분산 시물레이션의 낙관적 time-warp기법을 기본 구조로 하는 웹 기반 분산 시물레이터를 제안한다.

1. 시물레이터의 구현 원리

제안된 비동기 분산 시물레이션을 처리하는 논리 프로세스는 II장에서 소개된 time-warp기법

을 기본 구조로 하며, 동작 구조는 그림 2와 같이 시물레이션 서버, 분산된 시물레이션 개체, 웹 기반 통신 시스템의 3요소로 구성된다. 시물레이션 서버와 시물레이션 개체들은 웹 기반 통신 시스템을 통하여 공간에 대한 분산된다.

시물레이션 서버는 시물레이션 실행에 필요한 초기화를 설정하고, 분산된 시물레이션 개체들과 RMI의 원격호출을 이용한 메시지(외부 사건 메시지와 제어 메시지) 통신을 중앙 제어하고 시물레이션 결과를 분석/처리한다. 이를 위하여 시물레이션 서버는 웹 기반 통신 시스템을 통하여 분산된 시물레이션 개체들간의 외부 사건 메시지들을 중재하고 시물레이션 진행 결과를 저장해야 한다.

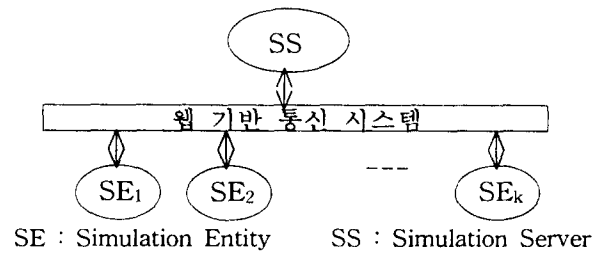


그림 2. 시물레이션의 동작 구조

분산 시물레이션 개체는 그림 3과 같이 외부 통신 인터페이스와 논리 프로세스들로 구성되고 웹 기반 통신 시스템을 통하여 분산되는 시물레이션에 대한 활성 문서이다. 분산 시물레이션 개체간의 통신은 시물레이션 서버에 원격 호출을 통하여 메시지를 교환방식이다. 즉, 외부 인터페이스는 동일한 분산 시물레이션 개체의 논리 프로세스들간 또는 분산된 시물레이션 개체들에서 논리 프로세스간의 외부 사건 메시지의 송수신을 처리한다.

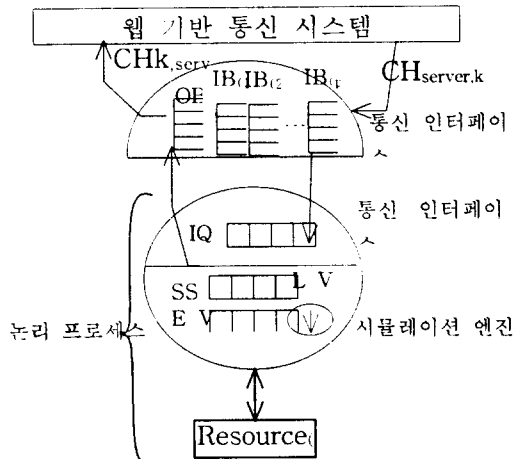


그림 3. 시물레이션 개체(SE_k)

분산 시물레이션 개체(SE_k)의 CH_{server,k}는 시물레이션 서버로부터 외부 통신 인터페이스에 연결된 통신 채널이다. 시물레이션 개체의 외부 통신 인터페이스는 시물레이션 서버로부터 외부 사건 메시지를 수신하여 해당 IB에 시간 순서로 입력한다. 다른 시물레이션 개체의 논리 프로세스에게 전송할 외부 사건 메시지는 OB에서 시간 순서로 관리하고 시물레이션 서버에게 전송한다.

논리 프로세스는 자바의 Thread 클래스로부터 생성된 클래스로 내부 통신 인터페이스와 시물레이션 엔진으로 구성된다. 컴퓨터의 프로세서들은 논리 프로세스들을 Thread로 병렬 처리한다. 논리 프로세스(LPp)는 시물레이션 개체의 IB(p)로부터 외부 사건 메시지를 수신하여 처리하고 외부 사건 메시지 전송은 내부 인터페이스를 통하여 외부 통신 인터페이스에게 전달한다. 내부 통신 인터페이스는 외부 사건 메시지를 외부 통신 인터페이스를 통하여 논리 프로세스들간 외부 사건 메시지를 처리한다.

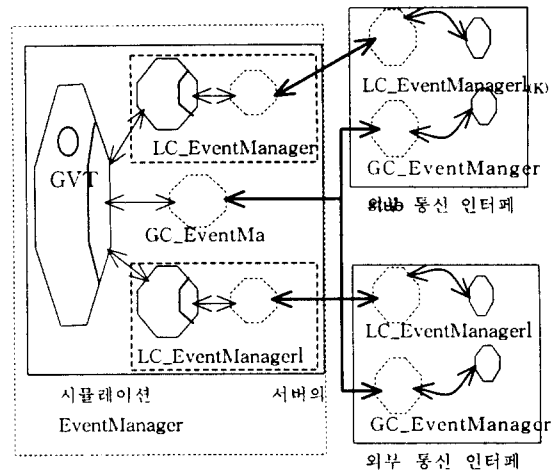


그림 4. 시물레이션 서버와 시물레이션 개체의 상호 동작

웹 기반 통신 시스템에서 시물레이션 서버와 분산 시물레이션 개체간의 외부 사건 메시지와 제어 메시지를 교환하기 위한 상호동작 구조는 그림 4와 같다. 메시지 교환을 위한 상호 동작은 자바의 RMI를 이용한 원격 메소드 호출을 통해서 이루어진다[조98].

2. 패키지 구성

본 절에서는 자바를 사용하여 재사용과 시물레이션 모델의 설계 및 시물레이션 구현을 지원하도록 구축된 클래스 라이브러리를 설명한다.

클래스 라이브러리는 event, queue, process, server, variate, statistics, visual 클래스 패키지로 구성된다. event 패키지는 Event 클래스로부터 내부 사건과 외부 사건에 대한 클래스들을 정의한다. queue 패키지는 기본적인 자료구조를 정의하는 클래스 패키지이고 Queue 클래스로부터 SQueue(Single Linked Queue), DQueue(Double Linked Queue), T_DQueue, T_SQueue 등을 정의한다. proces 클래스는 사건을 관리하는 자료구조와 기본적인 시물레이션 동작을 정의하는 클래스들을 포함한다. 이 클래스 패키지는 InputQueue, EventVirtualList, StatusStack, InputBuffer, LogicalProcess 등을 정의한다.

server 패키지는 ComServer, EventManager, ResultManager, ControlManager 클래스등을 정의한다. variate 패키지는 Variate 클래스로부터 Poisson, Exponential, Geometric, Uniform 클래스 등을 저의한다. statistic 클래스 패키지는 시뮬레이션 결과를 분석하기 위한 클래스들로 구성된다. 그림 6. 스위치1의 시뮬레이션 개체 실행 화면이며, 현재 진행중이다. visual 클래스 패키지는 ControlInfo, ProcessInfo, CharLine, ChartArc, ChartBar 클래스 등을 정의한다.

3. 구현 예

본 절에서는 구현된 웹 기반 분산 시뮬레이션을 그림 5의 ATM 스위치 모델에 본 연구에서 적용하여 그 활용성을 제시한다.

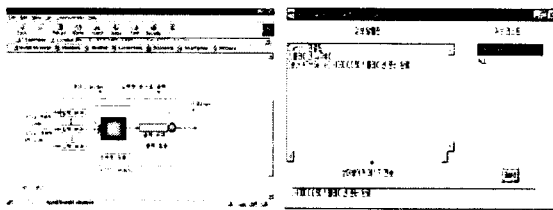


그림 5의 스위치1, 스위치2 및 스위치3는 각각 별도의 클라이언트에서 브라우저를 통하여 분산 수행되는 시뮬레이션 개체로 구현하였다[김97].

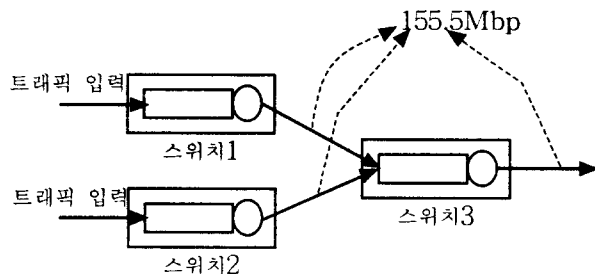


그림 5. ATM 스위치 시뮬레이션

ATM 스위치 모델의 설정은 다음과 같다. 각 ATM 스위치의 출력 버퍼는 500(셀) 크기이고, 출력 전송 속도는 155.5Mbps, 하나의 셀 전송 시간은 $4 \mu s$ 로 한다., 출력 버퍼는

FCFS(First-Come First Service)으로 동작하고, 버퍼 크기를 초과하여 입력된 셀은 폐기한다.

IV. 결론

최근 시뮬레이션 분야는 분산 시뮬레이션과 웹의 특성을 활용한 웹 기반 시뮬레이션의 구현에 대한 연구가 활발해지고 있다. 본 발표에서는 기존의 분산 시뮬레이션 또는 웹 기반 분산 시뮬레이션의 단점을 보완하여 분산 및 병렬 수행이 가능한 웹 기반의 분산 시뮬레이션을 자바 기법으로 구현하고, 이를 ATM 스위치의 분산 시뮬레이션에 적용하였다.

웹 기반 시뮬레이션을 구현하기 위하여 낙관적 time-warp 기법을 기본 구조로 하여 웹의 분산 특성과 자바의 객체 지향 특성, RMI 원격 호출 기능 및 스레드를 사용하여 시뮬레이션 클래스 라이브러리를 구현했고 진행중이다. 본 발표에서는 웹 기반 분산 시뮬레이션의 주목적은 멀티 프로세서 시스템에서의 분산 시뮬레이션과는 달리 시뮬레이션 수행 처리의 상호 운용에 있다. 또한 시뮬레이션 모델에 대한 방대한 분산과 병렬성을 고려한 시뮬레이션 구조를 제안한다.

참고 문헌

- [Bu96] A. Buss and K. Stork, "Discrete Event Simulation on The World Wide Web Using Java," Procs. of the 1996 Winter simulation Conference, pp. 780 - 785, 1996.
- [Pr90] B. Preis, "Performance of Discrete Event Simulation on a Multiprocessor using Optimistic and Conservative Synchronization", Dept. of Electrical and Computer Engineering University of Waterloo, 1990.
- [Am92] H. Ammar, "Time Warp Simulation

Using Time Scale Decomposition", ACM Transaction on Modeling and Computer Simulation, pp. 158 - 177, 1992.

- [Fi96] P. Fishwick, "Web-based Simulation: Some personal Observations," Procs. of the 1996 Winter simulation Conference, pp. 772 - 779, 1996.
- [Na96] R. Nair, et. al., "Java-Based Query Driven Simulation Environment," Procs. of the 1996 Winter simulation Conference, pp. 786 - 793, 1996.
- [Su96] Sun Microsystems, "Java Remote Method Invocation Specification", 1996, <http://www.sun.com>
- [김97] 김종은, 안효범, 조경산, "ATM 망에서 다중 임계를 이용한 트래픽 스케줄링 연구", 정보 처리학회 논문지, 제4권 7호, pp.1781-1787, 1997
- [조97] 조 경산, 컴퓨터 네트워크와 인터넷, 도서출판 그린, 1997.
- [조98] 조경산, 김종은, "웹 기반 분산 시뮬레이터의 구현 및 활용", 단국대학교 논문집, 제 33권, pp.329-339, 1998