CORBA를 이용한 멀티스레드 분산 시뮬레이션 환경

‡ 강원석° † 김기형
‡ 한국과학기술원, 첨단정보기술연구센터(AITrc)
† 영남대학교, 컴퓨터공학과
‡ wskang°@world.kaist.ac.kr, † kkim@yu.ac.kr

A Multi-threaded Distributed Simulation Environment Using CORBA

‡ Wonseok Kang° † Kihyung Kim
‡ AITrc, KAIST
† Dept. of Computer Engineering, Yeungnam University

요 약

DEVS(Discrete Event System Specification) 형식론은 계층적이고 모듈화된 형태로 이산사건 시스템을 기술한다. 본 논문에서는 CORBA를 이용한 Multi-threaded 분산 시뮬레이션 방법을 제시한다. 본 논문에서 제시한 시뮬레이션 방법은 기존에 선행 연구된 DEVSCluster를 기반으로 한다. DEVSCluster는 계층적 DEVS 모델들을 비 계층적 모델로 구성하여 시뮬레이션한다. DEVSCluster는 전통적인 계층적인 시뮬레이션 시 발생하는 overhead를 제거한다. CORBA기반으로 DEVSCluster를 구성함으로써 산업 표준에 맞는 확장을 가지는 분산 시뮬레이션이 가능해졌다. 그리고 CORBA를 이용함으로써 기존에 분산 시뮬레이션 동기화를 위한 새로운 패러다임도 적용이 가능해 졌다. 제시한 시뮬레이션 방법의 효용성을 보이기 위해 Windows 시스템에서 분산 시뮬레이션 엔진을 구현하여 대규모 물류 시스템으로 성능을 측정하였다.

1. 서 론

일반적으로 이산 사건 시뮬레이션은 시스템의 성능 분석 및 측정에 많이 사용된다. 그러나 대규모 시스템 시뮬레이션에 있어서 시뮬레이션 수행 시간 증가의 문제점이 남아있다. 분산 시뮬레이션은 이들 시뮬레이션 수행 시간을 줄이는 방법들을 제시하고 있다. 또한 분산 시뮬레이션은 대규모 시스템을 다루기 때문에 모델 검증, 모델의 타당성, 모델의 재사용성과 사용자에 대한 투명성 등을 제공하여야한다. Zeigler에 의해 개발된 DEVS(Discrete Event Systems Specification) 형식론은 이산사건을 모듈화 및 계층적으로 기술 할 수 있는 정형화된 구조를 가진다[1].

DEVS 형식론은 계층적인 모델을 구성 할 수 있는 특성으로 기본적 구성요소들을 연결 모델 및 더 복잡한 연결 모델 작성을 제공한 다. 연결 모델은 입/출력 포트들을 연결함으로써 구성된다. 계층적 모델 구성은 다음과 같은 특징을 가진다. 첫 번째 기본요소 시스템들은 상위 시스템들과 통합 전에 생성 및 분석을 가능하게 한다. 두 번째 특별한 목적으로 개발된 모델들은 다른 상위 시스템 구성 시 재사용을 가능하게 한다. 세 번째 모듈화 및 계층적 모델들은 실 시스템 모델 반영을 가능하게 한다. 이러한 시스템 이론적 특성 때문에 DEVS 형식론에 기반 한 모델링과 시뮬레이션은 프로그램 언어에 관계없이 구현할 수 있다. 분산 시뮬레이션 방법 또한 이러한 특성에 부합하기에 DEVS 형식론 많이 이용한다.

DEVS 형식론에 기반 한 분산 시뮬레이션은 전통적인 프로세스 기반 분산 시뮬레이션과 다음과 같은 차이점을 가진다[3]. 첫 번째로 DEVS 형식론은 시뮬레이션 모델에 대하여 외부/내부 사건에 대해 구별한다. 두 번째로 계층적인 모델을 구성한다. 이러한 차이점으로 병렬성을 제공하는 DEVS 형식론으로 인해 분산 시뮬레이션은 이를 적용하고 있다[2-6].

본 논문에서는 계층적 DEVS 모델 구조를 위한 CORBA를 이용한 분산 시뮬레이션 방법론을 제시한다(CORBA-DEVS-Cluser). 기존에 우리는 DEVSCluster라는 분산 시뮬레이션을 개발 하였다. DEVSCluster는 복잡한 계층적 시뮬레이션 모델 구조를 간단한 비 계층적인 구조로 시뮬레이션을 수행한다[7]. 본 논문에서 제시하는 시스템은 DEVSCluster의 시뮬레이션 메시지 전달 방법을 CORBA을 이용하여 처리한 것이다. CORBA를 이용함으로써 유연한 확장성을 가질 수 있으며 시뮬레이션 모델들에 대한 위치 추적 및 시뮬레이션 메시지 전달을 쉽게 처리할 수 있다. 그러나 CORBA를 이용함으로써 멀티쓰레드 기반으로 DEVSCluster가 시뮬레이션을 수행하기 때문에 기존 DEVSCluster에서 발생하지 않았던 deadlock이 발생을 하였다. 본 논문에서는 이러한 해결책을 함께 제시한다. CORBA-DEVSCluster의 효율성을 보이기 위해 대규모 물류 시스템 모델을 디자인 하여 성능을 측정하였다. CORBA-DEVSCluster는 정확한 분산 시뮬레이션 결과를 보여주었다. 기존 DEVSCluster보다는 수행성능 결과가 좋지 않았으나 CORBA를 이용함으로써 산업 표준화에 근거한 분산 시뮬레이션이 가능하게 되었다. 본 논문에서는 2장에서 CORBA-DEVSCluster에 대해 설명하고 3장에서 실험 결과 및 평가에 대해 설명한다. 마지막 4장에서는 결론 및 향후 계획에 대해 설명한다.

2. CORBA-DEVSCluster

본 장에서는 CORBA-DEVSCluster에 대해 설명한다. CORBA는 server와 client 관계에 있어서 client가 server에 포함된 operation을 내부적으로 procedure call을 수행 할 수 있게 하는 분산 객체 관리 기술이다[8]. 분산 시뮬레이션 방법론은 시뮬레이션 환경을 여러 대의 컴퓨터 노드들에서 시뮬레이션 동작을 수행함으로써 시스템 모델러들에게 좀 더 빠른 결과를 얻게 하는 것이다. 분산 시뮬레이션 시 시뮬레이션 모델들이 여러 대의 컴퓨터에 나누어져 있기 때문에 분산된 시뮬레이션 모델들 간에 시뮬레이션 메시지(이하 : 메시지)를 교환할 수 있게 하여야 한다. 본 장에서는 이를 처리하기 위해 CORBA의

분산 객체 관리 기술을 이용하는 방법에 대해 설명한다. CORBA의 분산 관리 객체 기술을 이용 시 여러 대에 분산된 시뮬레이션 모델들에게 메시지를 전송하기 위해 CORBA 객체로 구현된 인터페이스를 이용해 메시지를 전달해야 한다. CORBA는 어떤 시스템들이 CORBA로 구현된 객체를 이용 시 인터페이스에 정의된 operation을 invocation하기 위해서 multi-thread형태로 이들을 처리한다. 이 때문에 분산 시뮬레이션 실행 중에 메시지 전달 시 시뮬레이션 동기화를 위한 작업이 필요하다. 2.1 절에서는 CORBA를 이용한 multi-thread DEVSCluster에 대해 설명하고 2..2절에서는 2.1절에서 기술한 방법을 이용할 때 발생하였던 deadlock 해법에 대해 설명한다.

2.1 CORBA 객체를 이용한 DEVSCluster

그림 1은 CORBA 기반의 multi-thread DEVSCluster의 시뮬레이션 방법을 나타낸 것이다. CORBA-DEVSCluster가 C-ORBA를 이용하여 분산 시뮬레이션을 하기 위해서는 시뮬레이션 모델들에 대해 시뮬레이션 시간 동기화와 외부 전달 메시지 처리에 대한 인터페이스가 필요하다. 본 시스템은 이를 위해 CORBA Servant를 구성하여 그림 1과 같은 인터페이스로 분산 시뮬레이션을 수행한다. 그림 2는 분산 시뮬레이션을 하기 위한 CORBA Servant IDL 모듈이다.

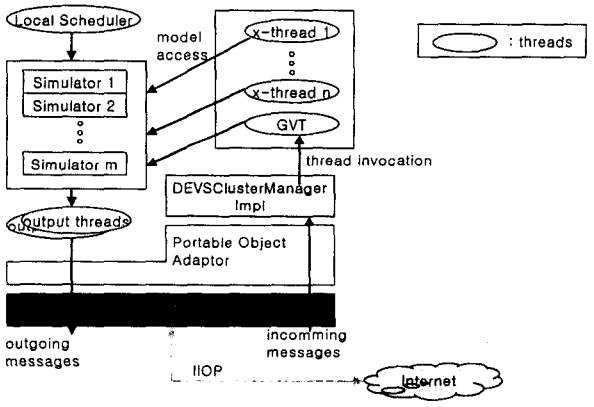

그림 1 CORBA-DEVSCluster

그림 2에서 SetGVT 및 Cal_Lvt는 분산된 시뮬레이터 간의 동기화를 위한 Servant Implementor이고 분산된 시뮬레이터 모델들 간의 입/출력 메시지를 주고받기 위한 Servant Implementor는 SendToXMesgThread이다.

```
module DEVSCluster
{
    struct Commbuf{
        unsigned long       src;
        long long       time;
        unsigned long       dst;
        long long       tN;
        unsigned long       simcount;
        unsigned long       priority;
        unsigned long       dupcount;
        unsigned long       type;
        unsigned long       sign;
        unsigned long       mesgid;
        unsigned long       func;
        string          buf;
    };
    interface DEVSClusterManager{
    void RunSimulation();
         void SendToXMesgThread(in CommBuf
buf);
    long long Cal_Lvt(in long long lvt);
    void SetGVT(in long long gvt);
    };
};
```

그림 2 CORBA-DEVSCluster를 위한 Servant 모듈

2.2 CORBA 객체를 이용 시 deadlock 해결책

DEVS 형식론에 기반 한 분산 시뮬레이션 모델의 정의에 따르면 분산 시뮬레이터 상호간에 시뮬레이션 동기화를 위한 메시지를 주고 받아야한다. 이러한 메시지 중 하나가 external transition 메시지이다[1]. 이 메세지는 어떤 한 시뮬레이터에서 시뮬레이션을 수행하고 그 결과를 다른 시뮬레이터에게 전달할 때 사용하는 메시지이다. 그림 3과 같이 external transition 메시지를 자신의 컴퓨터 노드의 다른 시뮬레이터에게 보내면 deadlock이 발생을 하지 않으나 external transition 메시지를 보내고자 하는 시뮬레이션 모델이 다른 컴퓨터 머신에 있을 때 deadlock이 발생할 수 있다. 컴퓨터(Node1)의 $Simulator_1$이 컴퓨터(Node2)의 $Simulator_2$에게 external transition 메시지를 전달하고자 할 때 컴퓨터(Node2)의 $Simulator_2$가 같은 시점에서 컴퓨터(Node1)에게 external transition 메시지를 전달 할 때 각각 $Simulator_1$과 $Simulator_2$는 다른 컴퓨터에서 들어온 external transition 메시지에 대한 처리 thread와 자신의 상태 정보를 보호하기 위해 lock을 걸고 작업을 수행한다. 이 때문에 deadlock이 발생한다. deadlock에 대한 해결은 그림 3에서 보는 것과 같이 CORBA Servant에 구성된 external transition 메시지 처리 Procedure를 바로 수행하지 않고 향후 수행되게 또 따른 thread로 실행하면 된다. 이것은 메모리 큐에 buffering하는 기능과 같은 효과이다. 이들 thread는 다음 스케쥴 때 즉시 실행된다.

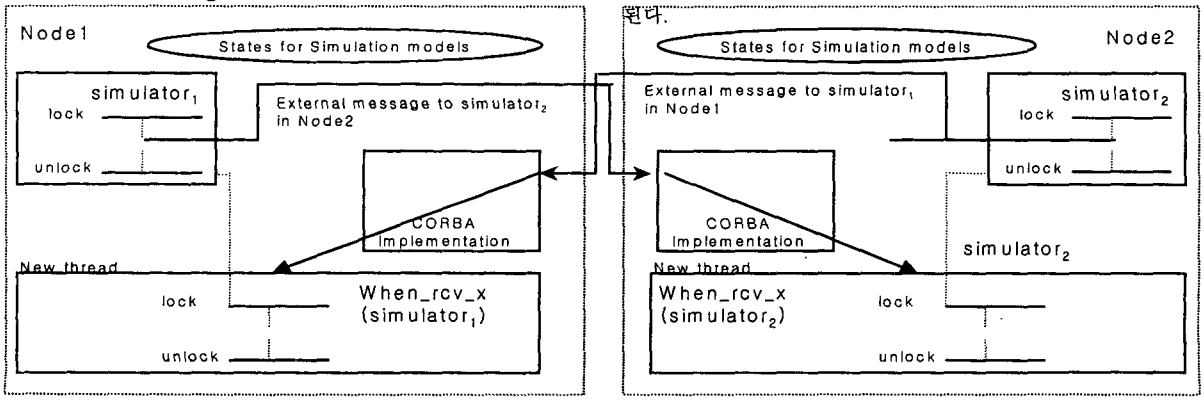

그림 3 CORBA 이용 시 external transition 메시지 처리 방법

3. 실험 결과 및 평가

본 논문에서는 CORBA-DEVSCluster 성능을 위한 효율성을 측정하기 위해 대규모 그래프를 가지는 물류 시스템을 대상으로 시뮬레이션 수행 후 그 분산 시뮬레이션 결과에 대해서 설명한다. 본 시스템은 Pentium IV, WindowsXP, 100Mbps에서 실험되었다. 그리고 컴파일러로는 VC++6.0을 사용하였다.

그림 4는 2대와 4대의 컴퓨터 노드에서 본 시스템을 이용하여 MPI버전과 CORBA를 이용하여 메시지를 보낼 때의 비 동기적 전달과 동기적 전달로 수행한 결과이다.

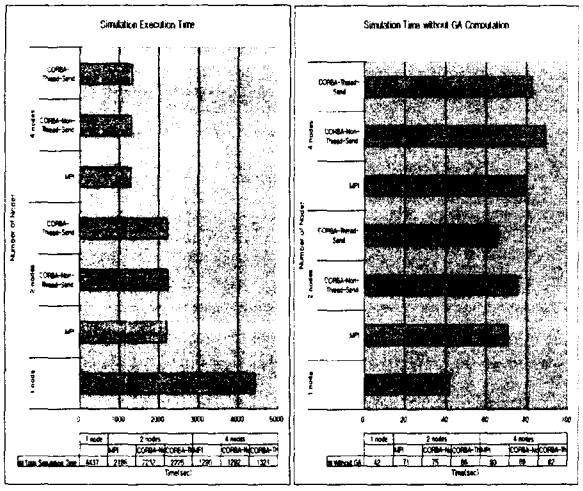

그림 4 CORBA-DEVSCluster와 기존버전간의 비교

그림 4에서 보는 바와 같이 거의 선형구조 형태의 성능을 보이고 있다. 이러한 결과로 보아 CORBA-DEVSCluster는 병렬성을 충분히 충족할 수 있는 시스템이다. 그림 4에서 보는 것과 같이 CORBA를 이용할 때 보다 전체적으로 MPI를 이용할 때가 분산 시뮬레이션 성능이 우수하다. 이는 MPI가 CORBA보다 미들웨어 적으로 처리하는 성능이 우수하기 때문이다[8-9]. 그림 4에서 왼쪽의 시뮬레이션 결과는 물류 시뮬레이터 수행 시 Genetic Algorithm(GA)를 사용하였기 때문에 CORBA 비 동기적 메시지 처리와 동기적 메시지 처리가 가변적이다. 이를 증명하기 위해 그림 4의 오른쪽 결과에서는 GA계산을 빼고 수행을 한 결과를 제시한다. 그 결과 CORBA를 이용한 비 동기적 메시지 처리가 성능이 더 높음을 보여준다. 이는 메시지 처리 시 비 동기적으로 처리함으로써 그 만큼의 메시지 전달 시간이 감소한 결과이다.

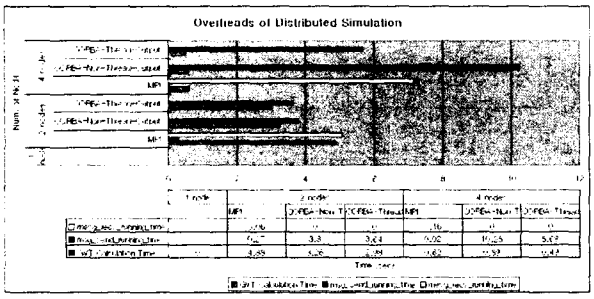

그림 5 CORBA-DEVSCluster의 모듈별 수행 성능

그림 5는 분산 시뮬레이션 시 시뮬레이터들만의 결과를 측정한 것이다. 여기에서 측정한 결과는 external message에 대한 처리와 분산 시뮬레이션 시 필요한 시뮬레이션 시간 동기화을 위한 메시지 처리에 대한 결과이다. 그림 5의 결과에서 보듯이 CORBA를 이용 시 메시지를 받는 시간은 0이다. 이는 다른 컴퓨터 노드에서 Procedure Call 형태로 수행되기 때문이다. 대신 Procedure Call을 수행하는 시간이 메시지 보내는 곳에 추가된다. MPI가 월등히 수행 성능이 높은 것은 실질적인 메시지 처리가 포함되어져 있지 않고 메시지만 버퍼링 하기 때문이다. 본 논문에서 기술 하지는 않았으나 CORBA를 이용함으로써 가능하게 된 시뮬레이션 동기화를 위한 새로운 GVT(Global Virtual Time)계산법이 적용되어 기존 DEVSCluster보다 빠른 GVT 계산을 수행할 수 있다. 이에 대해 간략히 언급을 하면 기존 MPI를 이용할 때는 GVT를 계산할 때 3-phase를 수행하였지만 CORBA를 이용할 때는 2-phase만에 가능하게 되었다. 이러한 결과는 그림 5에서 GVT 계산 측정을 통하여 증명할 수 있다.

4. 결론 및 향후 계획

본 논문에서 CORBA 기반 멀티쓰레드로 구동되는 DEVS 형식론 기반의 분산 시뮬레이션 환경을 제시하였다. 본 시스템의 성능을 증명하기 위해 대규모의 물류 시스템을 모델링하여 수행하였다. 본 논문에서 제안한 시스템은 기존에 수행되어졌던 MPI 보다는 수행 성능면에서 떨어지지만 CORBA를 이용함으로써 GVT계산과같은 새로운 패러다임적인 방법이 가능하게 되었다. 그리고 CORBA를 이용함으로써 산업체 표준 환경하에서의 좀 더 확장성있는 분산 시뮬레이션이 가능하게 되었다.

- 참고문헌 -

1. Zeigler, B.P., Praehofer, H., and Kim, T.G.,"Theory of Modeling and Simulation"
2. Chandy, K.M. and Misra, J.,"Distributed Simulation: A Case Study in Design and Verification of Distributed Programs", IEEE Trans. on Software Eng. vol. 5, no. 5, (1978) 440-452
3. Fujimoto, R.M.,"Optimistic approaches to parallel discrete event simulation", Transactions of the Society for Computer Simulation International, vol. 7, no. 2, October, (1990) 153-1-91
4. Fujimoto, R.M.,"Parallel and Distributed Simulation Systems", Proceedings of the 2001 Winter Simulation Conference (2001) 147-157
5. Kim, K.H., Seong, Y.R., Kim, T.G., and Park, K.H.,"Distributed Simulation of Hierarchical DEVS Models: Hierarchical Scheduling Locally and Time Warp Globally.", Trans. of SCS vol. 13. no.3. (1996) 135-154
6. Kim, K.H., Seong, Y.R., Kim, T.G., and Park, K.H,"Ordering of Simultaneous Events in Distributed DEVS Simulation", Practice and Theory, vol. 5, (1997) 253-268
7. Kihyung Kim, Wonseok Kang, Bong Sagong, Hyungon Seo, "Efficient Distributed Simulation of Hierarchical DEVS Models: Transforming Model Structure into a Non-Hierarchical One",33rd Annual Simulation Symposium, (2000), 227-236
8. Object Management Group,"The Common Object Request Broker: Architecture and Specification", 2.2 ed., Feb. (1998)
9. Message Passing Interface Forum,"MPI-2: Extensions to the Message-Passing Interface", (1997)