

수상함 전투체계 육상시험체계용 개체생성기 구현에 적합한 병렬처리기법에 관한 연구

A Study on the Parallel Processing of the Object Generator in a Surface Combat System LBTS

김창진* 오광백* 정용환**
Chang-Jin Kim Gwang-Baek Oh Young-Hwan Jung

Abstract

Object Generator is a software to provide simulation object data(aircraft, ship, submarine, missile, torpedo) for simulators in LBTS(Land Based Test System). but there is a burden to the system, because Object generator needs to send many object's data, display objects in a tactical screen, show object's information in a list in 1 second. This paper suggests a parallel software structure taking a few factors(deadlock, dependency) into consideration. At last, the paper shows the performance of the parallel structure's software compared with the former structure's software.

Keywords : Land Based Test System(LBTS)

1. 서론

개체생성기는 수상함 전투체계의 육상시험체계를 구성하는 하나의 시뮬레이터로서 운용자에게 시나리오작성 기능을 제공하고 이를 시뮬레이션하여 매주기마다 개체의 정보를 제공하는 소프트웨어이다. 또한 여러 가지 사용자인터페이스(UI)를 통해 개체의 정보를 전시하고 사용자에게 개체 편집등의 편의기능을 제공한다. 현재 수상함 전투체계는 요구조건으로 약 3,000개

표적 처리를 명시하고 있는데 이와 같은 요구조건에 충족하는 시뮬레이션 환경을 위해서는 개체생성기 또한 시뮬레이션마다 3,000개 이상의 개체정보를 시뮬레이션 환경에 제공해야한다. 그러나 주기마다 3,000개 이상의 개체의 정보를 시나리오 맞게 처리하고 이를 UI에 전시할 시에 전시 지연, 개체 정보 전송 누락 등의 여러 가지 문제를 발생시킨다.

현재 대부분 PC사양은 듀얼코어등 다수의 CPU를 가지는 멀티프로세서 환경을 제공한다. 그러나 멀티프로세서 시스템이 곧 자원의 병렬 사용을 의미하는 것은 아니고 개발자가 이를 이용하는 소프트웨어를 개발해야 시스템의 성능을 최대한 이용하는 것이라 할 수 있다.

본 논문에서는 멀티프로세서 환경에서 개체생성기

† 2010년 6월 28일 접수~2010년 9월 10일 게재승인

* 삼성탈레스(Samsung Thales)

** 국방과학연구소(ADD)

책임저자 : 김창진(yukichang@naver.com)

소프트웨어의 병렬처리를 통하여 다수개체 처리 시 성능을 향상시키는 방안을 제안한다.

2. 개체생성기 소개

육상시험체계는 전투체계를 구성하는 다수의 장비를 모사한 시뮬레이터들과 전투체계시스템과의 연동을 육상에서 모의해 보는 체계이며, 개체생성기는 이러한 육상시험체계를 구성하는 시뮬레이터 중 하나로서 육상시험체계의 각 장비 시뮬레이터에 개체의 정보를 제공한다. 각 장비 시뮬레이터는 개체생성기가 제공하는 개체의 정보를 수신하여 각 장비의 탐지 알고리즘을 적용하여 각자의 표적정보로 생성하고 이를 전투체계시스템에 제공한다. 아래의 그림은 육상시험체계에서 개체생성기와 다른 시뮬레이터와의 연동 구성도이다.

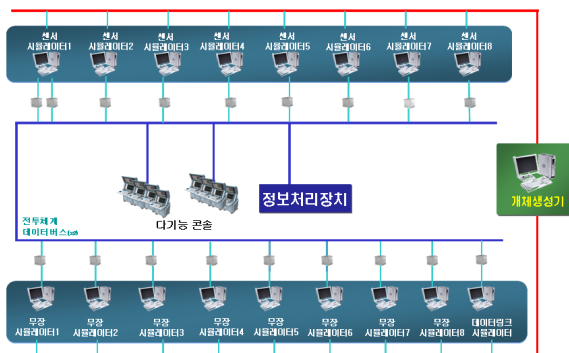


Fig. 1. 육상시험체계 구성도

개체생성기의 주요 기능 구조는 개체관리모듈과 시나리오관리 모듈, 정보전시 모듈로 크게 나눌 수 있다. 개체 관리 모듈은 각 개체를 해당 타입에 맞게 생성, 삭제, 시간별 처리를 담당한다. 시나리오 관리 모듈은 여러 개체를 하나의 시나리오로 설정하여 시뮬레이션 시간에 맞게 실행하고 시나리오를 파일로 저장하는 역할을 담당한다. 개체전시모듈은 개체의 정보를 지도상에 전시하는 전술화면과 이외의 리스트 전시, 개체 상세정보를 전시하는 윈도우로 구성되어 있다.

개체생성기는 객체지향(Object Oriented) 구조를 가지고 있다. 개체를 구성하는 클래스는 서로 상속성(Hierarchy)을 가지고 있고, 해당 개체 타입에 맞는 클래스로 동적 생성되어 다형성(Polymorphism)을 최대한

이용하는 방안으로 설계 되어 있다. 각 객체는 해당 타입의 클래스로 생성되며 위치정보 계산은 공통 부모 클래스 CTrack클래스에서 나머지 정보는 각 개체클래스에서 처리된다.

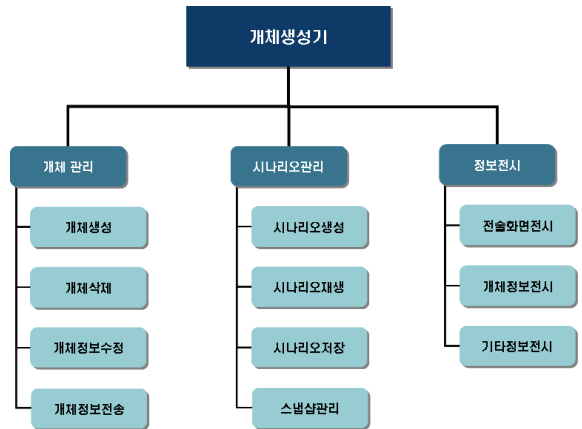


Fig. 2. 개체생성기 기능 구조도

개체 클래스는 다음과 같다.

(1) CTrack 클래스

개체클래스 중 최상위 클래스로 CObject 클래스를 상속받는다. CTrack 클래스는 모든 개체의 공통속성인 위치 정보, 시간에 따른 예상이동점을 멤버변수로 가지고 있으며 좌표변환 기능등 개체 공통 처리 기능을 수행한다.

(2) CAirTrack 클래스

CTrack의 자식클래스로서 대공개체의 상세속성정보(피착식별정보, 전파방사특성 등)를 멤버변수로 가지고 있다.

(3) CSurfaceTrack 클래스

CTrack의 자식클래스로서 위치성분중 고도가 0인 대함개체의 상세속성정보를 멤버변수로 가지고 있다.

(4) CSubmarineTrack 클래스

CTrack의 자식클래스로서 대잠개체의 상세속성(음파탐지특성)정보를 멤버변수로 가지고 있다.

(5) CMissile 클래스

CAirTrack의 자식클래스로서 미사일 개체정보를 처

리하고 특수함수(호밍기능)를 포함한다.

(6) CTorpedo 클래스

CSubmarineTrack의 자식클래스로서 어뢰개체정보처리하고 특수함수(호밍기능)를 포함한다.

이러한 개체클래스의 상속도는 아래와 같다.

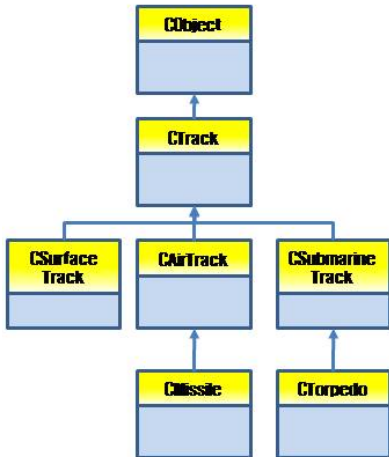


Fig. 3. 클래스 상속도

3. 병렬처리 시스템 구조 설계

가. Hot Spot

병렬처리를 위해서는 가장 시간이 많이 걸리는 부분인 Hot Spot을 우선적으로 찾아야 한다. 개체생성기는 1초마다 개체정보전송 뿐 아니라 개체의 정보를 운용자에게 전시해야 한다. 개체생성기는 타이머를 통해 1초마다 각 개체의 현재위치를 갱신하고 정해진 개체 메시지를 생성하여 발송하며, 수정된 정보를 리스트 및 전술화면에 전시해야 한다. 이 처리 과정의 시간은 전송주기 1초보다 작아야 하며 추가 UI전시(개체정보리스트전시, 운용자 개체선택)등을 위해서는 이 보다 훨씬 짧아야 한다.

매 주기마다 각 개체는 다음의 처리 루틴을 직렬적으로 처리한다.

- (1) 시간에 따른 현재 위치 계산 및 상세정보 변경
- (2) 메시지 구성 및 발송, 파일 로깅
- (3) 현재 위치로 전술화면 전시

아래의 그래프는 처리루틴 별 평균소요시간을 보여준다.

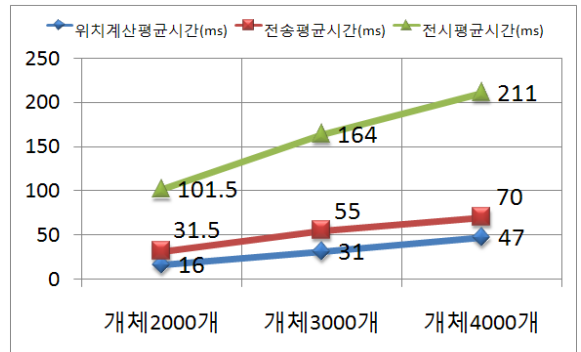


Fig. 4. 처리루틴 별 평균 소요시간

Fig. 4를 살펴보면 전술화면 전시가 가장 많은 시간이 걸릴 것을 알 수 있다. 그러나 전술화면에 대한 효율적 전시처리는 하나의 연구주제가 될 수 있으므로 이 논문에서는 제외한다.

나. 동기화

병렬적인 구조를 위해서는 동기화를 고려해야 한다. 동기화를 고려하지 않으면 Thread끼리 교착상태(DeadLock) 상황이 발생할 수 있으며 순서에 종속적 처리루틴은 올바르게 못한 결과를 발생시킬 수 있기 때문이다. 개체생성기 소프트웨어에서 동기화를 고려할 부분은 다음과 같다.

(1) 자함 처리

개체생성기가 전송하는 메시지에는 개체별로 자함을 중심으로 한 상대위치정보를 포함한다. 따라서 자함을 제외한 개체는 자함의 정보를 통해 자신의 위치정보를 다시 계산해야 하므로 자함은 다른 개체 보다 우선적으로 처리되어야 한다.

(2) 소켓 전송

개체생성기는 개체의 정보를 소켓을 통해 멀티캐스트로 전송한다. 모든 개체는 하나의 소켓객체를 공유하여 전송하므로 병렬처리 시 소켓 자원을 두고 각 Thread가 경쟁상황에 빠질 수 있다.

(3) 전술화면 처리

각 개체는 현재 시뮬레이션 시간으로 위치 정보를

계산한 후 전술화면에 위치정보를 갱신한다. 각 개체가 위치 계산을 종료하기 전에 전술화면을 갱신하면 전술화면과 실제 위치정보는 오차를 갖게 될 것이다.

다. 소프트웨어 구조 설계

기존의 개체생성기 구조에서는 매 주기마다 각 개체별로 개체위치 처리 및 전송, 화면 설정을 하고 모든 개체의 처리가 끝난 후에 전시화면을 재전시하는 직렬적 구조를 가지고 있다. Fig. 5는 개체생성기의 기존 구조를 보여준다.

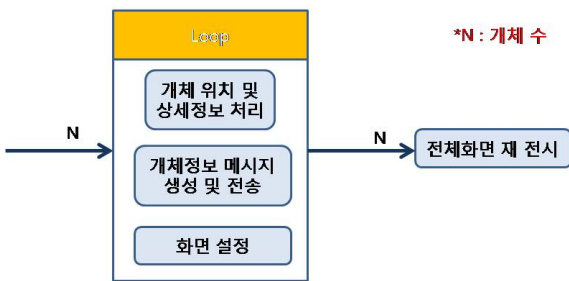


Fig. 5. 기존 소프트웨어 구조

동기화 문제등을 고려하여 병렬처리를 위해서 재설계된 개체생성기 소프트웨어의 구조는 다음과 같다.

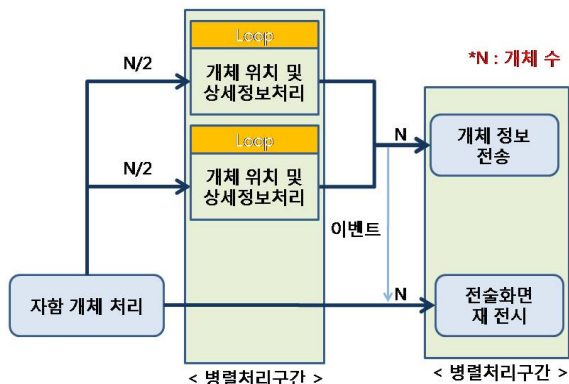


Fig. 6. 병렬처리 소프트웨어 구조

매주기마다 동기화를 위해 자함 만 우선적으로 처리한다. 이후에 각 개체의 위치 계산 및 상세정보 처리, 전송 메시지 구성을 병렬적으로 처리한다. 개체정보처리 루틴도 2개의 병렬구간으로 각각 처리하도록 설정하였다.

두 개의 분기로 나누어져 처리된 개체의 위치정보 처리가 종료되면 소켓을 통해 개체별로 생성된 개체정보를 일렬적으로 전송한다. 이와 동시에 이벤트 설정으로 전술화면 재전시 처리를 시작함으로써 메시지 전송과 전술화면전시를 병렬 처리한다. 메시지 전송은 소켓이라는 공통 리소스를 사용하기 때문에 병렬처리가 될 수 없다. 그러나 전술화면 전시 루틴과는 공통리소스를 사용하지 않으므로 병렬처리가 가능하다.

4. 시뮬레이션 조건 및 결과

시뮬레이션 환경은 Intel Core 2 Duo E6750 2.66Ghz CPU, 메모리사양 2GB인 PC를 사용했으며 개체처리에는 OpenMP API를 이용하여 병렬 처리하였다.

Fig. 7은 3,000개의 개체를 시뮬레이션 하는 개체생성기 소프트웨어의 주 화면이다.

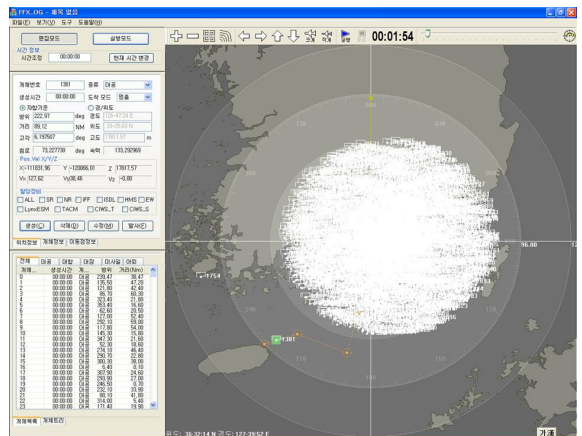


Fig. 7. 개체생성기 주 화면(개체수 N=3000)

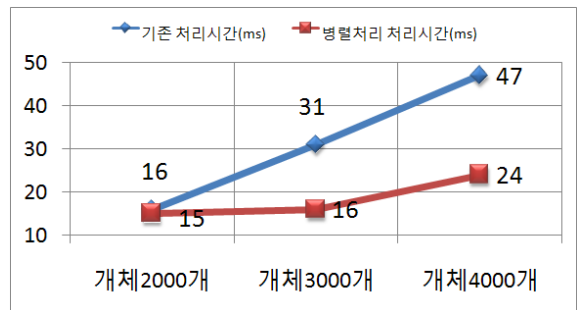


Fig. 8. 개체위치 및 정보처리 시간 비교

Fig. 8의 그래프는 개체의 각 초마다의 처리시간을 비교한 그래프이다. 개체수가 3,000이상부터 약 50%의 시간단축을 보여준다.

Fig. 9는 개체수 3,000개 일때 메시지 전송루틴과 전술화면 처리루틴의 병렬처리 시 성능 향상의 예를 보여준다. 소켓을 통한 메시지 전송루틴의 시간이 전술화면 처리루틴의 시간에 중첩되어 사라짐을 알 수 있다.



Fig. 9. 메시지전송과 전시 병렬처리 비교

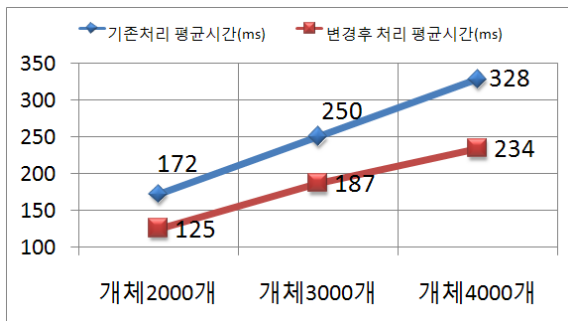


Fig. 10. 주기별 시간 비교

Fig. 10의 그래프는 주기마다 각 개체의 처리루틴(위치계산, 전송, 전시)시간을 개체개수 별로 비교한 그래프이다. 병렬처리를 이용한 구조설계를 통해 평균적으로 약 25%의 시간단축이 이루어졌으며 개체의 수가 증가함에 따라 성능은 더욱 개선됨을 알 수 있다. 구조변경 후 4,000개의 개체처리가 이전구조의 3,000개 개체처리 때보다 더 적은 시간이 걸림을 아래의 그래프에서 확인 할 수 있다.

5. 결론

본 논문에서는 다수의 개체를 개체생성기에서 효율적으로 처리하여 제한시간(전송주기/전시주기)내에 개체 처리 및 전송/전시하는 방법을 제안하였다. 많은 수의 개체처리 동기화를 고려하여 최대한 병렬처리함으로써 주어진 시간을 단축하고, 듀얼코어 시스템을 최대한 이용하여 성능을 향상시켰다. 수상함 전투체계는 3,000개 개체처리가 요구조건이나 앞으로 워게임이나 체계검증용 시뮬레이션 환경에서는 이보다 많은 개체와 처리 루틴이 포함될 것으로 판단되며, 이 경우 본논문이 제안한 병렬처리 소프트웨어 구조는 효과적으로 적용될 수 있다. 이와 더불어 개체 전시처리 루틴이 처리시간의 많은 비중을 차지하고 있으므로 이에 대한 연구가 차후에 진행되어야 할 것으로 보인다.

Reference

[1] C++ 병렬 분산 프로그래밍, Cameron Hughes, Tracey Hughes.