인터넷 기반의 분산된 병렬 처리를 지원하기 위한 분산 처리 지원 도구의 보안 기능과 운영 방안

이상윤,[O] 안철웅
대원과학대학[O], 계명문화대학
sylee@daewon.ac.kr[O], homepig@km-c.ac.kr

KISS Korea Computer Congress 2005

Sangyun Lee,[O] Cheolwoong Ahn
Dept. of Computer Information Processing, Daewon Science College,[O] Dept. of Multimedia, Keimyung College

요 약

분산된 컴퓨팅 환경은 프로세서의 개수를 적응적으로 활용하는 병렬 처리 환경으로 활용할 수 있다. 병렬 처리에 의한 수행 시간 단축 효과에 가장 많은 영향을 주는 것은 활용되는 프로세서의 개수와 병렬 처리 요소 상호 간의 통신 오버헤드이다. 분산된 컴퓨팅 환경으로 구성한 병렬 처리는 통신 오버헤드에 의한 단점과 프로세서의 개수를 자유롭게 활용할 수 있다는 장점이 상반되는 특성을 가지며 레이트레이싱에 의한 렌더링과 같이 계산량이 많고 병렬 처리 요소 상호 간의 통신량이 적은 응용 분야에 효과적이다. 분산된 컴퓨팅 환경을 병렬 처리에 활용하기 위하여 기존의 분산 처리 모델을 적용하면 통신 오버헤드 이외에 부수적인 오버헤드(프로그래밍 및 활용 절차)로 인하여 실효성을 발휘하기 어렵다. 단일 컴퓨팅 환경을 위한 절차와 서비스를 그대로 적용하여 분산된 컴퓨팅 환경을 구성하는 여러 대의 컴퓨터를 통합하여 활용하는 방안은 이와 같은 부수적인 오버헤드를 해결할 수 있으며 본 연구팀에서 이미 발표한 TORB(Transparent Object Request Broker)는 프로그래밍 투명성의 제공을 통하여 분산된 컴퓨팅 환경을 활용하기 위한 프로그램을 단일 컴퓨팅 환경을 위한 프로그래밍 기법을 적용하여 작성할 수 있도록 지원한다. 지속적인 연구를 통하여 프로그래밍 투명성의 범위를 확장함과 동시에 활용 절차의 투명성을 지원하는 방안을 추가하였고 새로운 분산 처리 모델을 설계하여 이러한 절차와 서비스를 체계적으로 정립하였다. 인터넷에 연결된 컴퓨터는 적절한 수준의 컴퓨팅 능력을 갖추고 있고 상호 간의 정보 교환을 할 수 있는 상태이므로 "TORB"와 같이 잘 정의된 패러다임으로 이들을 통합하여 운영하면 병렬 처리에 참여하는 프로세서의 개수를 자유롭게 활용하여 수행시간 감소 효과(병렬 처리에 의한)를 극대화할 수 있다. 그러나 인터넷을 기반으로 하는 분산된 병렬 처리를 지원하기 위해서는 "TORB"가 이미 제공하는 투명성 외에 불특정한 타인이 작성한 프로그램 코드가 "TORB"를 통하여 자신의 컴퓨터에서 실행되어도 악의적인 동작을 수행하지 못하게 하는 보안 기능과 인터넷에 연결된 방대한 수의 컴퓨터를 "TORB"에 의해 구성되는 분산된 컴퓨팅 환경에 참여시키는 시나리오가 필요하다.

1. 서 론

컴퓨팅 자원을 공유하기 위하여 분산된 컴퓨팅 환경을 구성하는 것은 심각한 보안상의 위험을 감수 하거나 이를 피하기 위한 대책이 마련되어야 한다. 마이크로소프트의 액티브엑스 기술을 활용한 분산 처리 서비스는 사용자의 명시적인 동의에 의해서 서비스 제공자의 코드가 사용자의 컴퓨터에 설치되어 동작한다. 이때, 서비스 제공자가 배포한 코드가 악의적인 동작을 하지 않는다는 보장은 액티브엑스 기술이 아니고 서비스 제공자의 공신력에 의존해야만 한다.[1]

냅스터(napster), 소리바다(soribada), 프루나(pruna), 당나귀(eDonkey) 등 P2P에 의한 파일공유 기술은 불특정 다수의 대상이 보유하는 정보를 공유하기 위하여 구성된 분산된 컴퓨팅 환경이며 방대한 수의 컴퓨터가 구성원에 참여한다.[2,3,4,5] 이러한 구성원의 방대함은 원하는 정보의 검색 및 다운로드의 성공적인 달성을 위하여 필수적인 것이다. 인터넷에 연결된 불특정 타인이 접근 할 수 있는 권리를 사용자가 보유하고 있는 파일 중 허가한 파일과 다운로드 중인 파일로 제한하는 보안상의 기술적인 보장이 참여하는 구성원의 방대함을 유도하기 위한 가장 중요한 요건으로 평가 할 수 있다. 보안상의 기술적인 보장을 제공하지 않는다면, P2P 환경의 구성원이 사용자의 자발적인 의사에 의하여 유지되므로, 이는 구성원이 방대하게 유지되는 것을 방해하는 가장 심각한 요인이 된다.

네트워크로 연결된 여러 대의 컴퓨터들로 구성되는 분산된 컴퓨팅 환경은 프로세서의 개수를 적응적으로 활용하는 병렬 처리 환경으로 활용할 수 있다. 병렬 처리에 의한 수행 시간 단축 효과에 가장 많은 영향을 주는 것은 활용되는 프로세서의 개수와 병렬 처리 요소 상호 간의 통신 오버헤드이다. 분산된 컴퓨팅 환경으로 구성한 병렬 처리는 통신 오버헤드에 의한 단점과 프로세서의 개수를 자유롭게 활용할 수 있다는 장점이 상반되는 특성을 가지며 레이트레이싱에 의한 렌더링과 같이 계산량이 많고 병렬 처리 요소 상호 간의 통신량이 적은 응용 분야에 효과적이다.

분산된 컴퓨팅 환경을 병렬 처리에 활용하기 위하여 기존의 분산 처리 모델을 적용하면 통신 오버헤드 이외에 부수적인 오버헤드(프로그래밍 및 활용 절차)로 인하여 실효성을 발휘하기 어렵다. 단일 컴퓨팅 환경을 위한 절차와 서비스를 그대로 적용하여 분산된 컴퓨팅 환경을 구성하는 여러 대의 컴퓨터를 통합하여 활용하는 방안은 이와 같은 부수적인 오버헤드를 해결할 수 있으며 본 연구팀에서 이미 발표한 TORB(Transparent Object Request Broker)는 프로그래밍 투명성의 제공을 통하여 분산된 컴퓨팅 환경을 활용하기 위한 프로그램을 단일 컴퓨팅 환경을 위한 프로그래밍 기법을 적용하여 작성할 수 있도록 지원한다.[6,7] 지속적인 연구를 통하여 프로그래밍 투명성의 범위를 확장함과 동시에 활용 절차의 투명성을 지원하는 방안을 추가하였고 새로운 분산 처리 모델을 설계하여 이러한 절차와 서비스를 체계적으로 정립하였다. 인터넷에 연결된 컴퓨터는 적절한 수준의 컴퓨팅 능력을 갖추고 있고 상호 간의 정보 교환을 할 수 있는 상태이므로 "TORB"와 같이 잘 정의된 패러다임으로 이들을 통합하여 운영하면 병렬 처리에 참여하는 프

로세서의 개수를 자유롭게 활용하여 수행시간 감소 효과(병렬 처리에 의한)를 극대화할 수 있다. "TORB"는 투명성을 제공하는 분산 처리 지원 도구 이므로 여러 대의 컴퓨터에서 "TORB 서버"를 기동하는 단순한 절차에 의해 분산된 컴퓨팅 환경을 구성할 수 있으나 인터넷에 연결된 컴퓨터를 대상으로 하기 위해서는 투명성 이외에 부가적인 문제를 해결해야 한다. 인터넷 환경에서 "TORB 서버"를 기동하는 것은 불특정한 타인이 작성한 프로그램 코드가 자신의 컴퓨터에서 실행되는 것을 허용하는 의미를 가지는 것이고, 사용자의 자발적인 의사에 전적으로 의존하는 것이기 때문이다. 이에 따라, 본 논문에서는 불특정 타인이 작성한 프로그램 코드가 악의적인 동작을 하지 못하게 하는 기술적인 보장을 제공하기 위한 보안 기능의 수용 방안과 인터넷에 연결된 컴퓨터들이 "TORB 서버"를 기동하도록 유도하는 시나리오를 제시한다.

2 보안 기능과 인터넷 기반의 분산된 컴퓨팅 환경

본 연구팀에서 제시한 분산 처리 모델이 기존의 것과 차이나는 점은 투명성을 제공하는 것이다. 이에 따라 여러 대의 컴퓨터가 참여해야만 구현 가능한 본질적인 응용 프로그램을 작성하기 위하여 단일 컴퓨팅 환경을 위한 프로그래밍 기법을 적용할 수 있고, 이미 작성된 레거시 프로그램이 분산된 컴퓨팅 환경을 구성하는 컴퓨터들을 활용하며 동작하도록 하여 성능 향상을 기대 할 수 있다. 또한 "TORB"는 이러한 서비스를 성공적으로 제공 할 수 있는 기능을 보유하고 있다.

참여하는 컴퓨터의 수를 방대하게 확보하여 분산된 컴퓨팅 환경을 구성하면 병렬 처리를 목적으로 작성된 응용 프로그램이 필요로 하는 프로세서를 충분히 확보 할 수 있으므로 수행시간 감소 효과를 극대화 할 수 있다. 그러나 인트라넷 환경에서 분산된 컴퓨팅 환경을 구성 할 때, 참여하는 컴퓨터의 수는 제한적이며 이를 통하여 수행 시간 감소 효과를 극대화 할 수 있다고 평가하기 힘들다. 이에 반해 인터넷에 연결된 컴퓨터는 일정 수준의 컴퓨팅 능력을 갖추고 있고 상호간의 정보 교환을 할 수 있는 상태이므로 "TORB"와 같이 잘 정의된 패러다임으로 이들을 통합하여 운영하면 수행시간 감소 효과(병렬 처리에 의한)를 극대화 하는 목적을 실현 할 수 있다.

2.1 보안 기능

"TORB"를 활용하면, 여러 대의 컴퓨터에서 "TORB 서버"를 기동하는 단순한 절차에 의해 분산된 컴퓨팅 환경을 구성 할 수 있다. 그러나, 인터넷 환경에서 "TORB 서버"를 기동하는 것은 불특정한 타인이 작성한 프로그램 코드가 자신의 컴퓨터에서 실행되는 것을 허용하는 의미를 가지므로 이 코드가 악의적인 동작을 하지 못한다는 기술적인 보장이 선행되어야 한다. "TORB 서버"를 기동하는 것은 사용자의 자발적인 의사에 전적으로 의존하는 것이므로 이러한 보장을 제공하지 않는 것은 구성원이 방대한 분산된 컴퓨팅 환경을 유지하는 것을 방해하는 가장 심각한 요인이 된다. "TORB"를 이용하여 인터넷에 연결된 방대한 수의 컴퓨터가 참여하는 분산된 컴퓨팅 환경을 구성하고 활용하기 위해서는 이러한 보안상의 보장을 할 수 있는 기술을 추가로 도입하여야 한다.

2.2 자바의 보안 및 인증 기능

2.2.1 단위 동작의 제한

자바 개발 키트에서 제공하는 라이브러리 객체 "java.lang.SecurityManager"와 "Policy File"을 활용하면 자바 응용 프로그램이 악의적인 목적을 가진 코드를 수행하지 못하도록 단위 동작을 제한 할 수 있다. 이 기능은 그림1의 상단에 표현된 것과 같이 자바 응용 프로그램을 기동 할 때 적용 할 수 있고, 그림1의 하단에 표현된 것과 같이 응용 프로그램 자체에 적용 할 수도 있다. 두 경우 모두 응용 프로그램이 수행하는 단위 동작을 제한하기 위한 정보는 "MyPolicy"라는 정책파일에 보관되어 있으며 이는 자바 개발 키트에 포함되어 있는 "policytool"을 이용하여 사용자가 직접 작성한다.

```
java -Djava.security.manager -Djava.security.policy=MyPolicy YourApp
```

```
o o o
String PolicyFileName = "MyPolicy";
System.setProperty("java.security.policy",PolicyFileName);
SecurityManager TORBSecurity = new SecurityManager();
System.setSecurityManager(TORBSecurity);
o o o
```

그림1 : 자바의 보안 기능을 활용하는 두 가지 방법

그림1의 상단에 표현 된 것은 안정성이 검증되지 않은 상태로 배포된 프로그램(그림1의 YourApp)이 사용자가 직접 작성하여 허용한 정책 정보(MyPolicy)의 범위 내에서 실행됨을 보장하는 것이고, 하단에 표현 된 것은 제시된 코드를 실행한 이후에 동일한 내용을 보장할 수 있으므로 배포된 프로그램이 변형되지 않음을 추가로 검증하여 안정성을 보장하는 것이다.

2.2.2 인증서 기능과 TORB의 배포

"TORB 서버"는 자체가 기동하기 위한 적절한 단위 동작을 수행 한 후 그림1의 하단에 나타난 코드를 실행하도록 구현되어 있다. "TORB 서버"가 이 코드를 실행 한 후에는 사용자가 직접 작성하여 허용한 정책 정보의 범위 내에서 단위 동작이 허용되므로 악의적인 목적을 가진 코드가 수행 될 수 없다. 그러나 "TORB 서버"가 이렇게 신뢰 할 수 있는 동작만을 수행한다는 보장을 하기 위해서는 사용자가 소유한 "TORB 서버"가 공식적으로 배포되어 변형되지 않은 것임을 보장해 주어야 한다.

자바는 배포하고자 하는 자바 코드에 전자 서명 정보를 포함시키고 이에 대한 인증서를 별도로 생성하며 인증서를 기반으로 하여 보안 정책을 적용하는 보안 기능을 제공한다. 전자 서명 정보가 포함된 "TORB"패키지와 이에 대한 인증서는 동일한 사이트에서 배포되므로 변형된 "TORB 서버"는 같이 배포된 인증서와 일치하지 않는다. 이 기능을 활용하면 사용자는 자신이 소유한 "TORB 서버"가 신뢰 할 수 있는 단위 동작만 수행하는 것임을 보장 받을 수 있고, "TORB 서버"가 수행하는 객체 서비스가 자신이 직접 작성하여 허용한 정책 정보의 범위 내에서 단위 동작을 수행하도록 할 수 있다.

2.3 허용해야 할 단위 동작

"TORB 서버"가 수행하는 모든 단위 동작은 사용자가 직접 작성하는 정책 정보에 의해 제한되므로 사용자가 모든 단위 동작을 금지하는 정책을 적용하면 "TORB 서버"에 의해 제공되는 분산 처리 동작 자체가 운영 될 수 없다. 이러한 이유로 "TORB 서버"의 의미 있는 운영을 위해 최소한의 단위 동작들을 허용하는 정책을 적용해야 한다. 이때 허용하는 단위 동작들 및 이들의 조합들은 악의적인 코드의 실행이 성공 할 수 없는 범위에 있어야 하며 그림2에 이들을 표현하였다.

그림2의 A는 전자 서명 정보가 포함된 "TORB"패키지에 포함된 코드를 위하여 허용해야 할 단위 동작이다. "java.net.SocketPermission"은 "TORB 서버"가 다른 컴퓨터에서 네트워크를

통하여 전달되어 오는 "요청"을 수용하고 이들에 대한 공식적인 인터넷 주소를 확인하는 동작을 정상적으로 수행 할 수 있도록 허용하는 것이고, "java.lang.RuntimePermission"의 "accessClassInPackage"는 "TORB 서버"가 자바 표준 라이브러리 코드를 접근 할 수 있도록 허용하는 것이다.

A) permission for officially distributed "TORB Server" code
permission java.net.SocketPermission "*", "accept"
permission java.net.SocketPermission "*", "resolve"
permission java.lang.SocketPermission accessClassInPackage, "*"
B) permission for A) and arbitrary wrote code execute by "TORB Server"
permission java.lang.RuntimePermission "createClassLoader"
permission java.net.SocketPermission "*", "connect"

그림2 : 전자 서명된 코드와 불특정 타인이 작성한 코드를 위하여 허용할 단위 동작

그림2의 B는 전자 서명 정보가 포함된 "TORB"패키지에 포함된 코드와 "TORB 서버"에 의하여 수행될 임의의 코드를 위하여 허용해야할 단위 동작이다. "java.net.SocketPermission"은 "TORB 서버"를 구성하는 코드 및 클래스 로더에 의해 전달받은 코드가 제3의 컴퓨터에 존재하는 객체를 접근하는 것을 허용하기 위한 것이고, "java.lang.RuntimePermission"의 "createClassLoader"는 "TORB 서버"를 구성하는 코드 및 클래스 로더에 의해 전달받은 코드가 제3의 컴퓨터로부터 실행 코드(클래스 파일)를 동적으로 전달받아 해당 객체를 생성하기 위한 것이다.

그림2에 허용된 단위 동작 및 이들의 조합을 활용하는 것으로는 "TORB 서버"가 실행되고 있는 컴퓨터의 지역 정보를 절대로 접근 할 수 없으므로 클래스 로더에 의해 동적으로 배포된 타인의 악의적인 코드가 성공적으로 수행 될 수 없음을 보장 할 수 있다.

3. 운영 시나리오

자바의 쓰레드와 동기화 메커니즘을 활용하여 작성한 병렬처리 응용 프로그램이 불특정 다수의 컴퓨터를 동시에 이용하여 수행 시간 감소 효과를 얻고자 하는 것은 프로그래밍 투명성과 함께 "TORB"가 제공하는 두 가지 분산 객체 서비스(객체의 위치 개념을 명시적, 암시적 활용) 중 하나를 활용하는 것이다. 이러한 활용이 가치를 가지기 위해서는 가능한 한 많은 수의 컴퓨터가 "TORB 서버"를 실행하고 있어야 하고 이들의 URL목록을 확보해야 한다. 인터넷에 연결된 불특정 다수의 컴퓨터들은 충분한 유휴 컴퓨팅 자원을 보유하면서 상호간의 정보교환을 할 수 있는 상태이므로 이러한 목적을 달성하기에 가장 적합하다. 다음은 인터넷에 연결된 컴퓨터의 사용자가 "TORB 서버"를 실행하도록 유도 할 목적으로 적용 할 수 있는 몇 가지 운영 시나리오이다.

3.1 동등한 목적

P2P에 의한 파일공유 기술은 자신이 원하는 정보를 얻기 위하여 다른 이가 원하는 정보를 암시적으로 제공하고 있으며 참여하는 구성원의 방대함으로 인하여 성공적으로 운영되고 있다.

"TORB"를 활용하면 자신의 컴퓨터에서 "TORB 서버"를 기동하는 것만으로 자신이 작성한 병렬 처리 프로그램이 필요로 하는 프로세서를 충분히 확보하고 활용 할 수 있다. "TORB"의 이러한 메커니즘은 P2P에 의한 파일공유 기술과 충분히 동등하며 성공적인 운영을 기대 할 수 있다. 또한, "TORB"는 이러한 운영을 실현하기에 충분한 체제를 갖추고 있으며 공식적인 배포를 준비하고 있다.

3.2 사이버 머니 제공

인터넷에 연결된 컴퓨터의 유휴 컴퓨팅 자원을 활용하여 특별한 서비스(계산양이 방대한 컴퓨팅 결과 제공)를 제공하는 사업을 고려한다면 "TORB 서버"를 기동하고 있는 컴퓨터의 목록을 방대하게 확보하고 이를 통하여 배타적인 분산 및 병렬처리를 수행 할 필요가 있다. 인터넷을 통한 온라인 게임에 참여하는 컴퓨터의 방대함을 고려한다면, 온라인 게임을 즐기기 위한 약간의 사이버 머니를 지급하는 것으로 이들이 "TORB 서버"를 기동하게 하는 것을 유도 할 수 있으며 사업에서 제공하고자 하는 서비스를 위하여 충분한 컴퓨팅 자원을 확보할 수 있다.

4. 결론

인터넷에 연결된 컴퓨터는 일정 수준의 컴퓨팅 능력을 갖추고 있고 상호간의 정보 교환을 할 수 있는 상태이므로 "TORB"와 같이 잘 정의된 패러다임으로 이들을 통합하여 운영하면 수행시간 감소 효과(병렬 처리에 의한)를 극대화 할 수 있다. "TORB"를 활용하면, 여러 대의 컴퓨터에서 "TORB 서버"를 기동하는 단순한 절차에 의해 분산된 컴퓨팅 환경을 구성할 수 있다. 그러나 인터넷 환경에서 "TORB 서버"를 기동하는 것은 불특정한 타인이 작성한 프로그램 코드가 자신의 컴퓨터에서 실행되는 것을 허용하는 의미를 가지므로 이 코드가 악의적인 동작을 하지 못한다는 기술적인 보장이 선행되어야 한다. "TORB 서버"를 기동하는 것은 사용자의 자발적인 의사에 전적으로 의존하는 것이므로 이러한 보장을 제공하지 않는 것은 구성원이 방대한 분산된 컴퓨팅 환경을 유지하는 것을 방해하는 가장 심각한 요인이 된다. 이에 따라, 구현된 분산 처리 지원 도구에 이러한 보안상의 안정성을 추가로 제공하기 위하여 자바가 제공하는 보안 기능을 적용하는 방안과 인터넷에 연결된 방대한 수의 컴퓨터를 분산된 컴퓨팅 환경에 참여시키는 시나리오를 제시하였다.

참고문헌

[1] CERT Coordination Center, "Results of the Security in ActiveX Workshop Pittsburgh", Pittsburgh, Pennsylvania USA, August 22-23, 2000

[2] Napster Web Team, "Napster - All the music you want. Any way you want it", On-Line Document, http://www.napster.com/

[3] (주)소리바다, "Share the Rhythmn", On-Line Document, http://www.soribada.com/index.html

[4] 프루나, "쉽고 편한 파일 공유 프루나", On-Line Document, http://www.pruna.com/

[5] MetaMachine, "eDonkey2000 - Overnet", On-Line Document, http://www.edonkey2000.com/

[6] 이상윤, 김승호, "프로그래밍 투명성을 지원하는 분산 프로그래밍 도구의 설계", 한국정보과학회 논문집 제 31권 3호, pp. 259-268, June 2004

[7] 이상윤, 김승호, "자바를 위한 분산된 병렬 컴퓨팅 환경", 대한전자공학회 논문지 제41권 CI편 제6호, pp. 533-547, November 2004