

분산환경에 적용될수 있는 DVM 및 DESPL의 제안

양일등^o, 이석희, 김성열
청주대학교 컴퓨터 정보공학과

fbiskr@hanmail.net

DVM and DESPL for the distributed environment

Il-Deung Yang^o, Seok-Hee Lee, Soeng-Ryeol Kim
Dept. of Computer Information Engineering, Chong-ju University

요약

컴퓨터의 파워가 증대됨에 따라 데이터의 처리속도도 증가했지만 더불어 사회의 복잡도가 증가함에 따라 처리해야 할 데이터도 증가하고 있다. 이에 병렬처리, 클러스터, P2P, 그리드등 분산된 컴퓨터들의 파워를 하나로 묶어 이러한 문제를 처리하려는 움직임이 있다. 하지만 이러한 연구들은 최종사용자들에게 분산환경에 대한 인지를 요구하고 있어 개발에 어려움이 있다. 또한 일반 프로그래밍 언어로 개발된 프로그램을 분산된 환경에서 실행하는 것은 쉽지 않다. 이에 언어적인 차원에서 분산환경을 지원하는 DVM 및 DESPL을 제안하고자 한다.

1. 서론

컴퓨터 분야의 기술은 시간을 더하면서 다양하고 빠르게 발전하고 있다. 각 분야의 다양한 발전과 인터넷이라는 거대한 네트워크의 힘까지 입어 구축된 지식정보는 그 수를 헤아릴수 없을 만큼 많지만 사회적으로 혹은 과학기술적으로 존재하는 많은 데이터를 처리하고 결과를 얻어내야 하는 것이 지금의 현실이기도 하다. 이에 전 세계에 분산된 컴퓨터 파워를 하나로 모아서 이런 분야의 연구활동을 하려는 시도가 있다. 그러한 시도는 과거 병렬처리부터 지금의 그리드 컴퓨팅까지 존재해 왔고 앞으로도 계속 될것이다. 하지만 언어적인 차원에서 이러한 개념을 지원한 것은 없었다. 기반 환경과 존재하는 언어의 결합으로 인하여 많은 데이터 처리를 신속하게 하고 있지만 일반 사람들이 분산의 개념이나 프로세스 관리라는 것들을 가지고 접근하기란 쉽지 않기 때문이다.

이는 컴퓨터의 실제 메모리를 프로그래머가 고려하지 않고 메모리 요청을 하면 OS에서 가상메모리 개념을 가지고 이에 대한 처리를 해주고 파일 시스템의 구성에 상관없이 디스크에 접근할수 있는 것도 같은 예라고 볼수 있다. 기계어의 불편함 때문에 고급언어의 개발이 있었듯이 분산된 자원을 하나로 묶어주며 개념적으로 물리적으로나 이러한 지원을 해주는 언어의 개발은 필요하다.

이에 본 논문은 분산 환경 프로그래밍을 실현하게 해주는 DVM(Distributed Virtual Machine) 및 DESPL(Distributed Environment Support Programming Language)의 제안하고자 한다. 순서는 분산 환경 연구에 관한 전반사항을 2장에서 다루고, 3장에서 현재 분산 환경 프로그램 개발에 대한 문제를 제기하고 4장과 5장에서 DVM 및 DESPL을 제안한다.

2. 관련연구

2.1 병렬처리(parallel processing)

병렬처리는 싱글 태스크 밖에 지원되지 않던 과거의 운영체제 등에서 동시에 두가지 작업을 수행하고 했던

개념에서 출발하였고 많은 다른 개념으로 변형되었다 [1].

그 변형중에는 멀티프로그래밍 혹은 멀티태스킹과 멀티 프로세싱 등이 있다. 멀티프로그래밍은 하나의 CPU에서 여러 개의 태스크를 조각내어 시분할한 다음 실행하는 것이고, 멀티 프로세싱은 한 시스템에 2개 이상의 CPU를 가지고 코드의 종속성이 없는 부분으로 나누어 실행하는 방식이다. 오늘 날의 운영체제에는 이 두 가지 모두를 지원하는 것이 대부분이다.

2.2 클러스터(cluster)

클러스터는 막대한 비용이 들어가는 슈퍼컴퓨터의 파워를 저렴한 컴퓨터들의 연결을 통하여 이루어내려는 분야이다. 전 세계적으로 보았을 때 클러스터의 연구는 대부분 표준화된 라이브러리인 MPI(Message Passing Interface)을 사용하기 때문에 클러스터의 기술적인 접근은 거의 일치하지만 아직까지 클러스터 프로그램 사용에 대한 뚜렷한 연구 결과가 나오지 않은 상태이며, 이는 분산 시스템의 구현에 앞서 클러스터를 이용한 병렬 프로그램의 개발에 어려움이 있기 때문이라고 분석된다[2].

2.3 분산객체(distributed object)

분산객체 분야의 대표적인 것들은 CORBA, RMI, DCOM, SOAP, RPC등이 있는데 그중 CORBA, RMI, DCOM을 살펴보면 다음과 같다.

COBRA는 하드웨어와 소프트웨어에 독립적인 객체 지향 언어를 기반으로 하는 소프트웨어 버스를 제공하기 위하여 OMG(Object Management Group) 의해 1991년에 시작되어 현재 800개 이상의 산업체가 참여하여 구현이 아닌 표준 인터페이스 규격을 정의하고 있다[3].

RMI(Remote Method Interface)는 Sun에서 개발한 미들 티어로서 자바 언어를 사용하는 환경에서 분산 객체를 구현 할수 있는 환경을 제공해 준다[3].

DCOM(Distributed Component Object Model)은

MS에서 사용하고 있는 분산객체 표준이며, COM을 네트워킹으로 확장하여 사용 할수 있다.

각각의 것들은 각자의 환경에 만든 미들 티어(Middle Tier)를 개발하여 분산 객체 시스템을 지원하고 있다.

2.4 다중 에이전트 및 객체 관리 시스템

다중 에이전트 및 객체 관리 시스템은 분산객체에서의 문제점인 해당 컴퓨터에서 서버 객체의 존재에 대한 문제점을 해결하기 위해 발전하였다[4.5.6.7]. 여기에서는 많은 방향으로 연구가 되었는데, 그중에 IBM Aglets는 IBM에서 개발한 자바 기반의 자율적인 이동 에이전트 시스템으로 서로 다른 네트워크상의 컴퓨터들에 설치되어 있는 에이전트 서버 사이를 에이전트가 쉽게 이주할 수 있으며, 자바를 기반으로 하기 때문에 컴퓨터의 플랫폼에 독립적인 장점을 가지고 있다. 다중 에이전트 시스템은 하나의 에이전트로 해결하지 못하는 복잡한 문제를 여러 에이전트간의 협력해 효과적으로 수행하기 위해 제안되었다.

2.5 P2P(Peer To Peer)

P2P 통신 방식은 과거의 클라이언트/서버 방식에서 각 피어(Peer)가 동등한 자격을 가지고 컴퓨터 통신을 할수 있어서 특정한 컴퓨터에 의존하지 않을 수 있으며, 어느 한 지역과 지역의 통신회선이 두절되더라도 또 다른 회선을 통해 상호연락이 문제없이 사용할 수 있는 통신체계이다[8]. 그 대표적 예로는Guntella와 한국형 Ktella 그리고 Freenet 이 밖에도 응용 프로그램으로는 국내에서의 소리바다, 씨프렌드, 구루구루를 비롯해서 다양하게 존재하고 있다[9].

2.6 그리드(grid)

이기중 컴퓨팅 자원과 대용량 저장장치, 다양한 고성능 연구·장비들을 통합한 환경을 그리드라고 표현한다[10]. 그리드는 P2P와 더불어 인터넷을 통한 WAN 구간에 분산 되어 있는 계산 자원을 활용하여 분산컴퓨팅 및 병렬처리를 수행하는 기술이다. 그리드는 기존의 클러스터링과는 달리 지리적으로 멀리 떨어져 있는 컴퓨터들의 자원을 공유하는데 그 목적을 갖는다. 이러한 컴퓨터들을 계산에 참여시키기 위해 모바일 코드(Mobile Code)나 분산 객체를 사용할 수 있다[7]. 현재 시장 표준인 글로버스(Globus) 미들웨어 툴킷은 그리드 서비스를 제공하는 그리드 미들웨어로서 전 세계적인 그리드 개발 과제들에서 가장 많이 사용되고 있다. 글로버스 툴킷이 널리 사용되는 이유는 글로버스 툴킷이 분리될 수 없는 단일 시스템이 아니라, 그리드에서 필요로 하는 다양한 서비스들을 독립적인 요소로서 제안하고 있기 때문이다.

2.7 JAVA와 VM(Virtual Machine)

자바의 유래는 1991에 James Gosling이 이끄는 Sun 엔지니어 그룹에서 케이블 TV 스위치박스 같은 가전제품에 이용되는 간단한 컴퓨터 언어를 설계하고자 했다. 이러한 장치는 많은 용량의 전력이나 메모리를 가지고 있지 않기 때문에 언어는 간단하면서도 견고한 코드를 생성해야만 했다. 또한 서로 다른 제조 회사들이 제각기 다른 중앙처리장치를 선택할수 있기 때문에 하나의 구조에 얽매이지 않는 것이 중요했다. 이러한 모든 특성을 살려서 자바가 탄생했으며 C++ 다음의 객체지향 언어로 자리 잡았다[12].

VM은 자바 바이트 코드 실행되기 위한 가상 환경인데, 자바는 중간 출력결과로 바이트 코드라는 *.class 파일을 생성한다. 이러한 클래스 파일을 자바

실행기(java)를 통하여 VM상에서 실행하게 된다. 이 가상 머신은 각 OS 별로 존재하므로 한번 작성되면 수정없이 다른 운영체제 상에서 실행이 가능하다. 이 점이 자바의 이식성을 아주 높이는 결과를 가져 왔다.

3. 분산 환경 프로그램 문제

3.1 개발 문제

현재 분산처리 환경부터 그리드 환경까지 개발은 기존 언어(C, C++, Java) 등과 연계 할수 있는 라이브러리를 제공하고 OS와 연계한 시스템을 기반으로 개발되고 있다.

하지만 이러한 분산 프로그램 개발은 최종사용자에게 컴퓨터 프로그래밍 외에 분산 환경 라이브러리와 분산 환경 시스템의 개념에 대한 지식을 요구하고 있다.

3.2 운영상의 문제

운영상의 문제는 각 분산 환경 마다 조금씩 다른데 가장 대표적인 것은 분산 환경 설치의 복잡성 및 분산 환경의 관리에 있다. 이는 대부분 전문 종사자에 의해 설치가 이루어 져야 하며 설치후에도 해당 시스템에 적절한 서비스를 제공하는 객체들을 지속적으로 관리하기 위한 노력이 필요하기 때문이다.

3.3 존재하지 않는 문제

'**존재하지 않는 문제?**'란 지금 까지 분산 환경에서 어떠한 방식으로 운영되었던 시스템일지라도 특정 환경에 맞게 프로그램을 작성해야 했었다. 즉 분산 환경의 개념이 들어가지 않은 프로그램을 작성한 것은 분산 환경에서 실행할수 없음을 의미하며 이것이 '**존재하지 않는 문제?**'인 것이다.

크게 보면 전 세계에 있는 분산 자원을 지원하는 시스템이 있고, 한 물리학자가 분산 환경이 지원되는 언어로 물리량에 대한 엄청난 계산을 요구하는 프로그램을 만들면 시스템이 자동으로 작업을 분산시켜 결과를 돌려주는 것이다. 여기에는 프로그래밍에 관한 사항만 이 필요한 것이다.

작게 보면 분산 환경이 지원 가능한 언어로 포토샵, HWP, 3D Studio MAX 등을 만들어 분산 환경 언어를 지원하는 것에서 실행한다면 현재의 시스템에서 운영하는 것보다 더 나은 성능을 기대할수 있을 것이다. 하지만 지금의 개발은 분산 환경에서 실행할수 있는 특수 포토샵, 특수 HWP, 특수 3D Studio MAX등을 요구했었다.

이에 DVM 및 DESPL을 제안하고 하는 것이다.

4. DVM 제안

4.1 DVM의 요건

DVM의 갖추어야 할 요건은 다음과 같다.

1. 각 DVM간에 위치가 동등해야 한다. : P2P에서의 방식은 중점의 클라이언트/서버 방식에서 탈피하여 각 Peer들 간에 정보 공유의 위치에 있어서 동등한 권한을 가지고 있다. 이는 IBM Aglets과 같이 Master/Slave의 관계가 아닌 누구나 환경에서 프로그램을 실행하기위해 자원을 요청하고 실행할수 있어야 한다는 것이다(P2P 개념).
2. DVM을 설치후에 해당 객체는 공유될수 있어야 한다. : 해당 사용자의 컴퓨터의 프로그램이 실행하면 필요한 다른 모듈들은 다른 분산 환경으로 자동 전송될수 있어야 한다(비 CORBA 개념).
3. 각 자원들에 대해 효율적인 관리할수 있어야

한다. : 자원은 크게 대형 저장장치부터 작게는 PC의 메모리등이 있는데 이러한 것들을 효율적으로 관리할수 있어야 하며 세부적으로 다음과 같이 이루어 져야 한다.

- 프로세서 : 한 작업을 분산 시켜 수행하기 위해 시스템이 각 분산 환경의 프로세서를 적절히 이용할수 있어야 한다. 특히 객체를 분리할수 있어야 하거나 코드 차원의 종속성을 파악하여 분산 실행할수 있어야 한다(클러스터, P2P, 그리드 개념).
 - Memory : 각 분산 환경에서 할당된 메모리를 가상 메모리의 개념처럼 단일 메모리와 같이 제공해 주어야 한다.
 - Disk : 각 분산 환경에서 할당된 디스크를 가지고 파일 시스템처럼 사용자에게 제공해 주어야 한다(DISK RAID 개념).
 - Graphic : (X Window System 개념).
 - Network : 각 분산 환경의 네트워크 자원을 모아서 하나의 논리적인 Network를 제공해 주어야 한다.
 - 기타 Device : 각 분산 환경에서 제공하는 것을 공유할수 있어야 한다.
4. 각 분산 환경의 자원을 상대적으로 평가할수 있어야 한다. : 분산 환경의 자원은 메인 프레임부터 개인용 PC 까지 다양하다. 이에 각 자원의 파워를 상대적으로 계산하여 가장 효율이 좋은 곳으로 처리 할당을 할수 있어야 한다.
 5. 동적인 분산 환경 정보를 갱신할수 있어야 한다. : 분산 환경은 실시간으로 다른 분산 환경의 정보를 얻고 갱신 할수 있어야 한다.
 6. 자신의 자원에 대해 쉽게 제공 및 해제를 할수 있어야 한다. : 좋은 인터페이스를 제공하여 사용자로 하여금 쉽게 자신의 자원을 제공 할수 있어야 한다.
 7. 오류가 일어났을 경우에 대처 할수 있어야 한다. : 정전으로 인한 사고나 해당 시스템의 결함으로 인하여 객체의 손실 및 기타 문제가 발생할 경우 대처 하고 복구 할수 있어야 한다.
 8. 각 분산 환경에 대한 정보 보호가 이루어 져야 한다. : 자신이 사용하고 있을 경우에 사용을 금하는 것과 같은 예와, 허락된 사용자만이 자신의 시스템을 이용할수 있도록 해야 하며 정보의 유출을 막을수 있어야 한다(정보보호개념).
 9. 각 분산 환경들은 동적으로 가입 및 탈퇴가 이루어 질수 있어야 한다. : 자신의 자원을 가입시키고 탈퇴시키는 것이 동적으로 이루어 질수 있어야 하며 이것이 쉽게 이루어 질수 있어야 한다(P2P개념).
 10. 플랫폼 독립적이어야 한다. : 각 분산 환경은 플랫폼 독립적으로 하부 구조를 사용자에게 은폐할수 있어야 하며 다른 플랫폼에서도 동일한 기능을 제공해 주야 한다(JAVA의 VM 개념).
 11. 표준 프로토콜을 재정하여 다른 기관들의 DVM과의 연결이 용이 해야 한다.(TCP/IP 개념)
 12. 싱글 환경에서도 수행이 가능해야 한다. : 다른 분산 환경 자원이 없으면 자신의 환경에서 수행하기 적절한 설정을 통하여 싱글 환경처럼 수행될수 있어야 한다.
 13. 설치가 쉬어야 한다. : Just One Click 만으로 설치가 이루어져야 하며 전문 지식이 없이도 사용할수 있어야 한다(Window 프로그램의 쉬운 설치 개념).

5. DESPL 제안

1. 객체 지향 언어이어야 한다. : 사용자가 객체의 개념으로 프로그래밍 할수 있도록 객체를 지원해야 한다.(JAVA 개념)
2. 분산 환경에 대한 개념을 은폐할수 있어야 한다. : 분산 프로그래밍을 하기 위해 특정 라이브러리를 요구하거나 코드를 요구해서는 않된다.
3. 언어적으로 분산에 대한 지원을 할수 있어야 한다. : 사용자가 특정 분산 환경의 자원의 사용을 원할 경우에 이에 대한 언어적 지원이 있어야 하며 JAVA를 기본으로 해서 다음과 같은 예가 있을수 있다.
 - `int:(Computer A) b[] = new int[100];`
 - `Thread:(Computer B) t = new Thread();`
4. 배우기 쉬워야 한다. : (JAVA 개념)

6. 결론

앞으로의 연구과제는 제안된 사항을 가지고 DVM 및 DESPL을 설계하고 구현해야 한다. 이를 뒷 받침 하기 위해 현재 Java의 VM환경에 대한 연구를 선행하고 P2P 및 Grid의 프로세스 관리 기법과, 객체관리 기법과 종속성을 분석한 코드 분할 및 클러스터와 RAID 를 이용한 디스크 관리등에 연구를 해 나가야 할 것이며, 표준화된 프로토콜을 설계하여야 할 것이다.

참고문헌

- [1] 강성룡, "<http://terms.co.kr/parallelprocessing.htm>", 가나다식 온라인 컴퓨터 용어사전 * 텀즈", 검색어, parallel processing, 1999. 6
- [2] 김영태, 이용희, 최준태, 오재호, "초고속 네트워크를 이용한 PC 클러스터의 구현과 성능 평가", 한국정보과학회 논문지, 시스템 및 이룬 제 29 권 제 2 호, 2002. 2
- [3] 장중현, 이동길, 한치문, "개방형 네트워크 환경을 위한 멀티쓰리드 기반 코바 설계 및 구현", 한국정보처리학회, 2001. 11
- [4] 김경하, 김영학, 오길호, "IBM Aglets를 기반으로 하는 가상 병렬 컴퓨팅 시스템에서 작업 할당 기법과 성능 비교", 한국정보과학회, 2002. 4
- [5] 전병국, 이근상, 최영근, "Java 언어를 이용한 객체이동시스템의 설계 및 구현", 한국정보처리학회, 1998. 11
- [6] 신필섭, 김신덕, "자바 인터넷 컴퓨팅 환경을 위한 효율적인 자원 관리 시스템의 설계 및 분석", 한국정보처리학회, 1999.
- [7] 변상선, 진현숙, 고영용, 유혁, "그리드 컴퓨팅을 위한 프로세스 원격수행 모델", 한국정보처리학회, 2002.
- [8] 이광현, 전형수, 유철중, 장욱배, "서버 부하 감소를 위한 P2P 기반 데이터 서비스 시스템의 설계 및 구현", 정보처리학회, 2002. 10.
- [9] 이석희, 한정식, 이태희, 조 상, "Hybrid P2P의 그룹관리와 신뢰성을 위한 시스템 설계", 한국정보과학회, 2002.
- [10] 강경우, "이기종 슈퍼컴퓨팅 자원들을 활용한 Globus 기반 그리드 구축에 관한 연구", 한국정보처리학회, 2002.
- [11] 이종숙, 홍정우 "그리드 미들웨어 해부", 마이크로소프트웨어, 2002. 7.
- [12] CAY S. HORSTMANN, GARY CORNELL, "Core Java2", SunSoft Press, pp 13-14, 2000