Korea@Home : P2P형 인터넷기반 분산컴퓨팅

한국과학기술정보연구원 박찬열 · 길준민 · 최장원 · 박학수 · 이필우

1. 서 론

냅스터(Napster)[1]로부터 촉발된 피어투피어(Peer-to-Peer: P2P)에 대한 관심이 개인용 컴퓨터의 쓰임새를 더욱 높이 평가하게 만들었으며, 냅스터 이후에도 수없이 많은 P2P 응용 기술들이 나타나고 있다. 이는 무어의 법칙을 능가하는 컴퓨팅 파워의 향상과 인터넷의 발전 및 확산으로 인하여 컴퓨팅 패러다임이 열린 구조로 전환되고 인터넷의 최종 끝자락까지 펼쳐지게 되었음을 의미한다.

P2P 시스템의 목적은 다른 여느 컴퓨팅 시스템과 마찬가지로 이 시스템을 활용하는 사용자의 요구를 충족시키려는 응용을 지원하기 위한 것이다. 따라서 일반인들 사이에서 P2P의 전부인 것처럼 인식되는 파일 공유 목적의 Napster, Gnutella, e-Donkey, KaZaA 등의 응용 이외에도 P2P 시스템 활용과 관련된 여러 가지 분야가 존재하며 이와 관련된 여러 연구들이 진행되고 있다.

그러한 여러 활용과 연구들에 있어서 P2P의 핵심은 '공유(sharing)'에 있다고 볼 수 있다. 현재 대부분의 P2P 응용 프로그램들이 '파일 공유'를 목적으로 하고 있지만, 협업(collaboration)에 활용되는 P2P 시스템 역시 '인적 자원의 공유'로 볼 수 있다. 그리고 또다른 P2P 응용 중 하나가 컴퓨팅 파워를 공유하는 분산컴퓨팅 분야이다. 이러한 방식의 시초는 1997년 시작된 거대 Mersenne 소수를 찾기 위한 GIMPS[2]와 암호 메시지 해독을 위한 distributed.net[3] 프로젝트이며, 가장 규모가 크고 널리 알려진 SETI@Home[4]이 1999년에 시작되었다. 국내에서도 이러한 분산컴퓨팅을 위한 일반적인 플랫폼을 제공하고자 하는 Korea@Home 프로젝트[5]가 수행 중이다. 이들 P2P 시스템들에 대해 [6]에서는 시스템의 목적에 따라 분류된 현재 널리 알려진 응용 시스템의 예를 그림 1과 같이 기술하고 있다.

이 논문에서는 ***P2P형 인터넷기반 분산컴퓨팅 프로젝트인 Korea@Home을 소개하고, 이와 관련된 P2P 관련 연구와 이슈들, Korea@Home 프로젝트에 적용되는 이슈들과 해결 방안, 그리고 향후 방향 등에 대해 기술한다.***

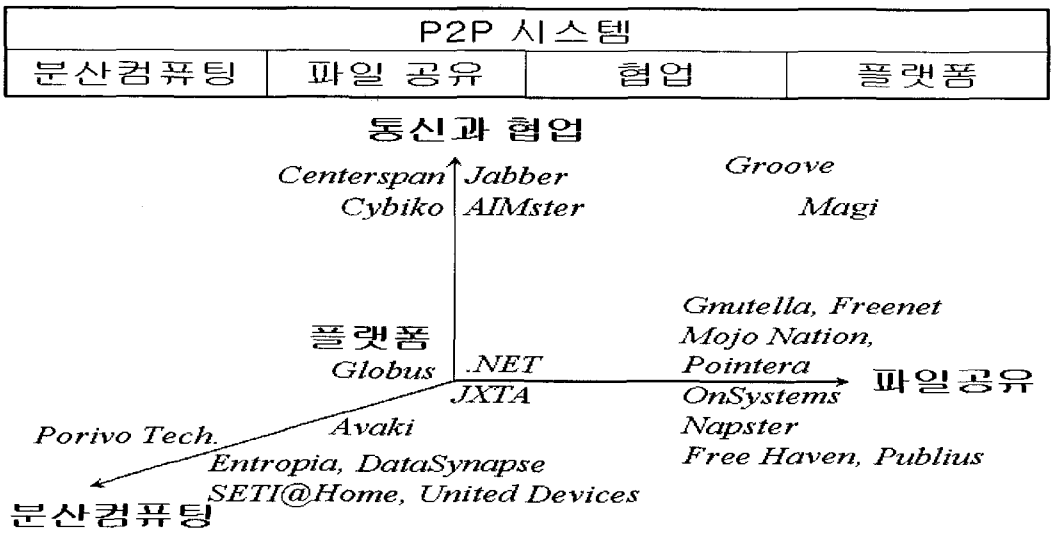


그림 1 P2P 시스템의 분류

2. P2P 이슈들 및 관련 연구

P2P의 개념은 완전히 새로운 것이 아니어서 기존의 여러 응용 중에서도 뉴스 서버의 운영에 사용되는 UUCP[7]나 교환 망(Switched Network)[8] 등에서 이미 활용되어 왔다. 하지만 여러 관점에서 새롭게 조명을 받으며 복잡성과 유용성에 대해 다양한 연구가 진행되고 있으나 아직 성숙되지 않은 단계라고 볼 수 있다. 그만큼 많은 관심을 끌며 인터넷의 차세대 주자로서 인정받고 있으며, 기술적인 연구뿐 아니라 사회 및 문화적 관점에서의 연구[9,10]도 다양하게 진행되고 있다.

P2P 모델과 시스템에 대한 정의도 여러 곳에서 다양하게 표현하고 있는데, [6]에서는 "중앙집중식이 아닌 방식으로 주요 기능을 수행하기 위해 분산된 자원을 이용하는 시스템들과 응용들의 집합체"로 P2P 시스템을 정의하고 있으며, [11]에서는 "참여하는 컴퓨터들 간의 직접적인 통신을 통해 자원 공유를 위한 응용을 수행하는 네트워크로 연결된 컴퓨팅 모델"로서 P2P 모델을 정의하고 있다. 이외에도 여러 연구들[12,13,14]에서 P2P 시스템에 대한 정의들을 기술하고 있는데, 대부분 다음과 같은 내용을 공통적으로 포함하고 있다.

- 분산된 자원(distributed resources) : 지리적으로 한 곳에 모여있지 않은 컴퓨팅 파워, 데이터(스토리지, 컨텐츠, 파일 등), 네트워크 대역폭, 인적 자원(human resource) 등의 자원들이 주체가 되어 시스템을 구성
- 단말의 참여(participating at the edges of networks) : 수행하고자 하는 주요 기능이 서버 중심의 자원 활용이 아닌 연결된 네트워크(예를 들어 인터넷)의 개인 사용자 PC와 같은 최종 단말을 통해 이루어짐
- 공유(sharing)를 통한 주요 기능 수행 : P2P 시스템의 목적에 따라 분산컴퓨팅, 컨텐츠 공유, 메시징 서비스, 협업, 플랫폼 서비스 등에 분산된 자원의 공유를 통해 제공

이러한 정의에서 이미 P2P 시스템의 특징을 일부 포함하고 있으며, [6]에서 그림 2와 같이 Client-Server 방식과 P2P 시스템의 비교를 보여준다. 일반적인 P2P 시스템의 장점으로 내세울 수 있는 특징들은 다음과 같다.

2.1 확장성(Scalability)

P2P 시스템은 자원의 소유가 분산화 되어 시스템에 참여와 배제가 보다 자유스러워 확장이 용이하다.

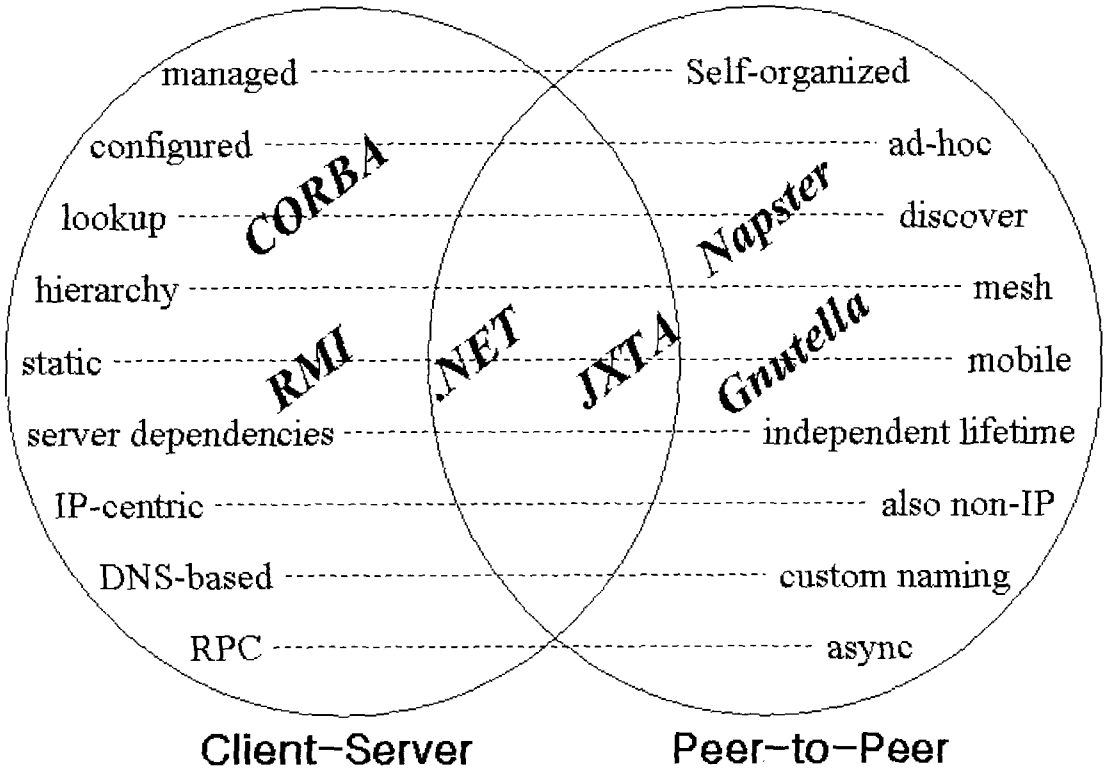


그림 2 Client-Server와 P2P 방식의 비교

2.2 성능(Performance)

확장성과 함께 분산된 처리 및 저장 능력, 그리고 네트워킹 능력의 적절한 제어를 통해 성능 향상을 얻을 수 있다. 하지만 성능 향상의 효과는 아키텍처에 따라 발생하는 오버헤드로 인해 달라질 수 있는데, Napster나 SETI@Home과 같이 중앙 제어의 방식으로 수행되는 아키텍처와 Gnutella[15]와 같이 완전 분산된 아키텍처를 결합한 Hybrid 방식[16]이 두 방식의 장단점을 절충하여 사용되기도 한다.

2.3 신뢰성(Reliability)

P2P 시스템의 중요한 장점 중 하나가 결함이나 고장의 발생 지점이 한 곳에 집중되어 있지 않아 결함 포용(fault-tolerance) 능력이 있다는 것이다. 이를 통해, 전체 시스템을 구성하는 요소들 중 일부분에 발생한 오류로 인해 전체 시스템이 부정확하게 동작하는 경우를 방지할 수 있다. 또한 일부의 오류로 인해 원하는 자원으로의 접근이 가능하지 않을지라도 같은 기능을 하는 다른 자원으로의 접근이 가능하기 때문에 가용성(availability)도 높아지게 된다.

그러나 이러한 특징들을 만족하기 위해서는 해결해야 할 여러 가지 이슈들이 존재한다.

2.4 복잡도의 증가

클라이언트와 서버의 역할을 동시에 수행하는 대규모의 공유 자원들이 참여함으로써 시스템의 제어와 관리뿐 아니라 결함에 대한 대처, 필요한 자원의 검색 등에 있어서 복잡도가 매우 높아짐에 따라 이를 해결할 수 있는 기술적 기법들이 요구된다.

2.5 불안정한 환경

일반적으로 인터넷의 끝자락에 연결된 각 자원들은 운영과 연결 형태가 매우 다양하다. 예를 들어, 저속의 모뎀으로 연결된 PC 뿐 아니라 xDSL이나 케이블, ATM 네트워크 장비, 또는 무선 연결 등의 다양한 속도의 네트워크로 연결될 수 있으며, 프록시(Proxy)나 방화벽, 또는 NAT 뒤에 가려진 채 연결되기도 한다. 또한 운영체제나 하드웨어 역시 매우 이질적(heterogene-ous)이며 자원의 소유가 집중적이지 않고 분산되어 있어 행동 양식이 매우 다양하여 자주 연결이 끊기는 등 영속적인 연결이 보장되지 않으므로 이에 대한 대처가 필요하다.

2.6 보안의 위협

자원 소유의 분산으로 인해 신분이 확실하지 않은 자원이 참여할 가능성이 많으며, 다양한 형태의 보안 위협에 노출되어 있다. 익명성(anonymity)으로 인해 신분을 위조하거나 메시지의 가로채기, 메시지의 위변조 등을 통한 개인정보 노출의 위험도 존재한다.

2.7 상호 호환성

매우 많은 P2P 시스템들이 존재하지만 아직 표준화된 방식이 존재하지 않기 때문에 충분한 규모를 확보하지 못할 때가 많다. 표준화에 대한 노력이 꾸준히 이루어지고 있긴 하지만 아직 명확한 표준은 나타나지 않고 있으며, 통신 방식이나 운영 방식에 대한 표준화가 요구된다.

다음 장에서는 P2P 시스템의 한가지 활용인 분산컴퓨팅과 국내에서 수행 중인 Korea@Home 프로젝트를 소개하고 이와 관련된 이슈와 해결 방안에 대해 기술한다.

표 1 그리드와 P2P 인터넷기반 분산컴퓨팅의 차이점

	그리드	P2P 인터넷기반 분산컴퓨팅
구성원	컴퓨터 과학 뿐 아니라 다양한 응용 과학 분야에 걸친 전문가 집단	많은 수의 다양하고 익명 구성원
자 원	보다 다양하고 높은 성능으로써 고성능 네트워크로 연결됨	인터넷에 연결된 데스크탑 컴퓨터가 주를 이루며 불안정한 환경
관 리	각 자원의 소유가 확실하며 명확하게 관리되는 경우가 대부분	자원의 소유와 관리가 집중식으로 이루어지지 못함
응 용	대용량 데이터를 다루는 전문적인 응용	특정 응용에 따라 운영
규 모	수십에서 수백	수천에서 수백만
인프라	GGF 등을 통한 표준화 추진	공통적인 플랫폼 미비

3. 인테넷 기반 분산 컴퓨팅

3.1 관련연구

고성능 컴퓨팅을 위한 시도는 전산학 분야에서 끊임없이 이어져 왔다. 전통적인 방법은 하드웨어적으로 더 빠른 성능의 기계를 고안하거나 소프트웨어적으로 알고리즘을 개선하는 방법이다. 하지만 이와 더불어 단위 요소들의 결합을 통해 보다 높은 성능을 얻고자 하는 시도

역시 지속적으로 이루어지고 있다. 그러한 시도들로서 나타난 결과물들이 대칭형 다중처리장치(Symmetric MultiProcessing; SMP)나 고도화된 병렬처리(Massively Parallel Processing; MPP)를 위한 슈퍼컴퓨터 또는 클러스터 컴퓨터 등이다.

여기에 더해 최근에는 이러한 고성능 장비들을 지역적인 제약없이 활용하여 더 높은 성능을 얻고자 하여 그리드 컴퓨팅(Grid Computing)이 시도되고 있다. 대규모의 분산된 자원을 활용하여 고성능 대용량 컴퓨팅 파워를 얻어내고자 하는 그리드는 결국 P2P 분산컴퓨팅과 동일한 목적을 가지고 있지만, 표 2과 같이 이들 두 기술 간의 차이점이 존재하여 구현에 있어 서로 다른 접근방법을 사용한다.

그리드와 P2P 인터넷기반 분산컴퓨팅은 발생 자체가

표 2 해외의 대표적 인터넷기반 분산컴퓨팅 프로젝트 예

프로젝트명	참여기관	프로젝트 내용	시작년도/회원수
SETI@Home	UC Berkeley 대학	@HOME 프로젝트 원조, 외계 전파 탐색	1999년/400만
Folding @Home	Stanford 대학	단백질 구조분석 프로젝트	2000년/2만3천
Genome @Home	Stanford 대학	Folding@home과 형제 프로젝트로Genome의 진화와 작용 연구	2001년/1만1천
Intel-UD Cancer Research	Intel, 미국립암연구재단, 암협회,United Devices, 옥스퍼드대	단백질 상호작용 분석을 통한 암 신약선도물질 분석	2000년/90만
FightAIDS @Home	미국 Scripps Research Institute, Entropia	Autodock이라는 Protein ligand docking simulation 프로그램으로 HIV 바이러스에 대한 신약선도물질 개발	2000년/3만4천
Moneybee	독일 i42사	신경망 이론을 바탕으로 주가 예측	2000년/1만
Golem @Home	Brandies 대학, Entropia	가상 로봇을 이용한 시뮬레이션을 통하여 로봇 설계 연구	2000년/2만4천
Casino-21	영국 COAPEC, 옥스퍼드 대학,	관측된 데이터를 이용해 가상모델을 평가하고 모델의 치명적인 결함 검증	1999년/4만5천
GIMPS	Entropia	Mersenne 소수 중 가장 큰 소수 찾기	1996년/19만
Grub	미국 Grub사	P2P 방식의 검색 로봇엔진 프로젝트	2001년/400

표 3 Kor시스템의 설계 목표

구성요소	목 표	설 명
응용 수행자 (Customer)	편리성(Convenience)	기존 응용 작업에서 이 시스템을 이용하기 위한 전환 작업이 용이해야 한다.
	정확성(Correctness)	악의적인 보안 위협들을 극복하고 정확한 결과를 제공해야 한다.
	유연성(Flexibility)	특정 응용이 아닌 다양한 응용 분야에 활용될 수 있어야 한다.
자원 제공자 (Resource Provider)	투명성(Transparency)	자원 제공자의 실제 사용에 방해를 주지 말아야 한다.
	신뢰성(Reliability)	어떠한 보안 위협에도 대처해야 한다.
	보상정책(Rewards)	어떤 식으로든 보상이 존재해야 한다.
Korea@Home 서버	확장성(Scalability)	자원 제공자가 많아지더라도 성능과 안정성에 문제없이 확장가능해야 한다.
	적응성(Adaptability)	인터넷의 다양한 환경에 대처할 수 있어야 한다.

달라 서로 다른 접근 형태로 목적을 달성하고자 구현되어 왔고 차이점과 장단점이 존재하지만 그리드에서는 OGSA[17] 등을 통해 상호 호환성을 확보하고자 노력하고 있으며, 인텔, HP 등을 중심으로 만들어진 산업계 표준화 커뮤니티인 P2P 워킹그룹이 GGF(Global Grid Forum)에 흡수됨으로써 같은 목적을 가진 두 분야는 서로 통합되어가는 추세에 있다.

인터넷기반 분산컴퓨팅(Internet-based distributed computing)은 공공 컴퓨팅(Public Computing)[14, 18] 또는 자발적 컴퓨팅(Volunteer Computing)[19], 군중 컴퓨팅(Hive Computing)[20] 등으로도 불리고 있다. 표 2은 해외의 대표적인 인터넷기반 분산컴퓨팅 프로젝트의 사례들을 보여주고 있으며, 이들 프로젝트들은 아직 호환성이 없이 수행되는 응용작업에 따라 별도로 운영되고 있다. 하지만 최근에는 분산컴퓨팅 프로젝트들도 다양한 응용을 수행할 수 있는 공통 플랫폼을 마련하고자 하는 시도가 이루어지고 있어서 BOINC[21]나 GPU[22]와 같은 것들이 대표적인 예이다. 학계에서의 연구 차원에서 수행되는 것으로 그치지 않고 상업적인 목적의 인터넷기반 분산컴퓨팅을 수행하기 위한 제품들도 나타나기 시작하였으며[23,24,25,26,27], 국내에서도 인터넷기반 분산컴퓨팅을 위한 기반 연구와 개발 및 서비스를 위한 Korea@Home 프로젝트[5]가 진행되고 있다.

3.2 Korea@Home 설계 목표

Korea@Home은 세계 최고 수준으로 발전한 국내의 네트워크 및 컴퓨팅 인프라를 적극 활용하여 대용량 고성능 컴퓨팅 파워를 창출함으로써 국내 연구 및 산업 발전에 이바지 하고자 하는 목적으로 2002년도에 시작되었다. 앞서 언급한 BOINC나 기타 P2P 분산컴퓨팅 시스템들과 같은 목적을 가지고 있지만, 단순히 시스템의 구축과 서비스 뿐 아니라 보다 일반적인 플랫폼으로서 역할을 수행하기 위해 다양한 응용 API 지원과 SDK 지원을 추구하고 있다.

Korea@Home 시스템을 구성하는 요소를 세 가지로 분류하면, 첫 번째는 이 시스템을 이용하여 고성능 컴퓨팅 파워를 실제로 사용하는 응용 개발자, 두 번째는 이 시스템에 컴퓨팅 파워를 제공하는 자원 제공자, 그리고 마지막으로 이 둘을 연결하여 필요한 정보를 저장하고 작업 관리, 자원 관리, 스케줄링 등을 담당하는 서버이다. 이 요소별로 Korea@Home 시스템이 만족시켜야 할 목표를 표 3과 같이 수립하였다.

이러한 Korea@Home의 설계 목표를 달성하기 위한 여러 시도들과 걸림돌이 바로 해결해야 할 이슈들이다. 다음 절에서는 Korea@Home에서 해결해야 할 이슈들과 해결 방안들을 기술한다.

3.3 Korea@Home 개발 방향

3.3.1 성능의 확장

시스템에 참여하는 자원의 수가 증가할수록 성능이 선형적으로 증가하지는 않겠지만, 제어와 관리, 통신 등의 오버헤드로 인해 성능이 오히려 저하되어서는 안될 것이다. 실제로 성능의 저하를 발생시키는 병목은 (1) 동기화와 같이 제어가 필요한 연산을 위한 처리, (2) 저장을 위한 입출력 부분, (3) 통신 지연 등에서 발생한다. 각 참여 자원에서 저장을 위한 입출력에 대한 병목 해결은 자원별로 특성이 존재하고 해결하는데 한계가 존재하지만, 제어 연산과 통신 지연에 대해서는 다양한 해결 기법들이 연구되고 있다.

이를 위한 국내외 연구 방향을 살펴보면 알고리즘의 개선을 통한 연구가 주를 이루고 있는데, 제어 연산의 경우 분산컴퓨팅 분야에서 기초적인 모델의 제시와 분석을 통해 오래 전부터 연구되어 오고 있다[28,29]. 특히 P2P 시스템이 인터넷을 기반으로 하고 있지만 잦은 접속 단절이나 컨텐츠의 이동 등 불안정한 환경으로 인하여 DNS와 같은 일반 인터넷의 주소 탐색 방식을 그대로 사용할 수 없기 때문에 별도의 명명법(Naming)과 원하는 자원을 찾는 기법을 필요로 한다. 그리고 시스템의 규모가 자원의 참여도에 따라 가변적으로 변하므로 제어 연산과 통신 지연의 많은 부분이 여기에서 발생한다.

이 기법들은 조회를 위한 중앙 서버가 존재 여부로 크게 달라지는 P2P 시스템의 아키텍처에 따라 서로 다르게 적용되는데, 1세대의 방식이 Napster나 SETI@Home과 같이 중앙에 조회 서버를 가지고 있는 방식이라면, 2세대의 방식은 Gnutella와 같이 브로드 캐스팅을 이용하는 방식이다. 최근에 보다 일반적인 목적으로 활용되고 확장성과 결함 포용력이 좋은 방법으로서 Chord[30], CAN[31], Pastry[32], Tapestry[33]와 같은 DHT(Distributed Hash Table)을 이용하는 연구들이 3세대 방식으로 등장하였다.

Korea@Home에서는 제어와 관리의 편의를 위해 중앙 서버를 두고 작업의 분배 스케줄링을 수행한다. 보안과 관련된 제어와 감시가 필요한 부분을 제외하고 대부분의 기능들을 각 노드들이 담당하도록 하여 보다 자율적이고 지능적인 피어를 구현하는 것을 목표로 하고 있다.

3.3.2 다양한 환경에 대한 적응

앞 절에서 설명한 바와 같이 인터넷에 연결되어 분산컴퓨팅에 참여하는 자원들은 하드웨어, 운영체제, 네트워크 연결 등이 매우 다양하며, 중앙집중 방식과 달리 자원의 관리 권한 및 소유가 분산되어 행동 양식이 매우

불안정하다. 또한 P2P 시스템은 노드들 간 또는 서버와 노드들과의 직접적인 연결이 이루어져야 하는데 방화벽이 존재하거나 NAT 등을 통해 사설 주소 등을 사용하는 경우 이러한 직접 연결이 불가능하다.

이러한 문제를 해결하기 위해 Korea@Home에서는 두 가지 방안을 사용한다. 첫 번째는 거의 모든 노드들이 HTTP를 이용한 인터넷 접속이 가능하기 때문에 항상 HTTP와 HTTPS 프로토콜을 이용하여 연결하도록 하는 것이고, 두 번째는 자원 제공자에 설치되는 에이전트를 보다 지능적이고 자율적으로 기능하도록 하는 것이다. 전달되는 메시지는 항상 SOAP을 통해 HTTP 프로토콜로 이루어지며, 직접 연결이 가능하지 않은 경우에만 서버를 경유하도록 한다. 향후 확장성을 위해 JXTA[14] 등에서 활용하는 중간 노드(superpeer)를 이용할 것이다. 또한 다양한 환경에 대해 가장 잘 알고 있는 자원 자신이 환경 변화에 대한 대응을 어떤 방식으로 수행할 것인지 결정하도록 하여 네트워크 연결이 끊긴 상태에서도 자율적인 작동을 수행하도록 [34]에서와 같이 접속 단절 상태에서의 연산을 지원하고, 작업의 중복 배분과 수행하는 작업의 검사점(checkpoint) 기록을 통해 작업 수행의 효율성과 연속성을 보장하여 보다 자율적인 수행을 지원한다.

3.3.3 자원 제공자에 대한 보안 위협

P2P 시스템에서 가장 중요하게 대두되는 것 중 하나가 시스템에 참여하는 자원들의 보안 문제이다. 특히 중앙집중식의 제어가 느슨한 P2P 시스템에서 익명성으로 인한 보안의 위협에 대한 대처는 반드시 필요하다. Korea@Home에서는 표 4와 같은 시나리오들을 자원 제공자에 대한 보안 위협 형태로 분류하고 이에 대한 대응 기술을 적용한다.

표 4 자원 제공자에 대한 보안 위협 시나리오

보안 위협 형 태	설 명
참여자로의 위장	자원 제공자로 위장하여 서버 또는 다른 자원 제공자에게 공격을 시도
네트워크 감시	네트워크를 통해 송수신되는 메시지들을 감시하여 자원 제공자의 정보를 탈취
악성 코드의 유입	자원제공자가 실행할 작업 코드로 위장하여 악의적인 실행을 시도

Korea@Home에서는 공개키 기반의 인증 시스템을 활용하여 위의 문제들을 해결한다. 먼저 자원 제공자는 반드시 정당한 자원 제공자로서 참여하기 위해 서버로부터 유일한 아이디와 인증서를 발급받아야 하며, 이 인증서를 통해 자원 제공자의 신원을 확인한다. 메시지를 주고받을 때에는 항상 인증서를 통해 신분 확인을 수행하므로 자원 제공자로 등록되지 않은 공격자가 악의적인 데이터를 담은 메시지 전달은 가능하지 않다. 또한 이러한 메시지 전달 과정에서 민감한 데이터가 포함된 메시지의 경우 SSL(Secure Socket Layer) 채널을 통해 전달하도록 하여 네트워크 감시를 통한 정보 탈취 역시 가능하지 않다. 마지막으로 악성 코드의 유입에 대한 대응으로는 전자서명 방식을 이용한다. Korea@Home에서 주고 받는 모든 파일들은 서버에 의해 사전에 서명되어야 한다. 즉, 파일들은 MD5 해싱을 통해 요약되고 이것은 다시 암호화되어 전달되며, 이러한 전자 서명을 통해 자원 제공자는 실행가능한 파일의 경우 서버에 의해 사전에 서명된 것인지 사전에 검증 절차를 거쳐야 한다. 여기에 더해 샌드박싱 기법 중 Korea@Home 시스템에 더 적합하다고 생각되는 PCC(Proof-Carrying Code) 기법[35]을 적용할 예정이다.

3.3.4 정확한 응용의 수행

응용이 수행되는 설계와 실행 흐름이 정확하다는 가정 하에서, 응용 수행의 정확성에 해를 입히는 주요 요인 역시 앞 절의 보안 위협과 관련이 있는데, 임의의 공격자 또는 자원 제공자가 악의로 잘못된 결과를 생성해 냄으로써 정확한 수행을 방해할 수 있다. 이에 대한 대응으로 Korea@Home에서는 중복 수행을 통해 자원 제공자가 생성해 낸 결과의 정확성을 검증한다. 즉, 서로 관련이 없는 둘 이상의 자원 제공자가 같은 결과를 내 놓은 경우 이 결과를 정확한 것으로 인정한다. 하지만 주의해야할 점은 반드시 관련이 없고 독립적인 자원 제공자가 수행하도록 작업이 배분되도록 해야한다는 것이다.

3.3.5 다양한 응용의 편리한 수행

응용을 수행하는 과학자 또는 개발자는 일반적으로 자신들이 사용하기 편리하게 손에 익은 도구나 개발 언어들이 존재한다. 이러한 응용 수행자들이 보다 손쉽게 Korea@Home을 활용할 수 있도록 지원함으로써 Korea@Home의 활용율을 높일 수 있다. 이를 위해서는 이전에 사용하던 도구들과 같은 또는 유사한 기능과 인터페이스를 제공해야 한다. 주로 사용하는 상용화 도구에 대한 기능 지원은 쉽지 않겠지만, 고성능 병렬처리를 위해 표준화된 PVM[36] 또는 MPI[37]의 개발 인터페이스와 유사하게 활용할 수 있도록 지원하는 것이 필요하다.

또한 P2P의 장점을 충분히 활용할 수 있도록 다양한 API를 지원할 필요가 있는데, 이러한 시도로서 JXTA와 같은 플랫폼이나 [38]와 같은 연구가 이루어지고 있다. Korea@Home도 역시 이러한 공통 플랫폼 API를 설계하고 지원하는 것을 목표로 하고 있다.

3.3.6 자원 제공에 대한 보상

자원 제공자는 자신의 자원을 제공하는 대가로 어떤 식으로든 보상을 받아야 한다. 그 보상의 형태가 비록 물질적인 대가이거나 부가적인 서비스를 받는 등이 아닌 심리적인 만족일지라도 분명히 기여에 대한 보상이 존재해야 한다. 이러한 보상 정책이 존재하기 위해서는 기여치에 대한 측도가 존재해야 하고 정확한 측정이 가능해야한다. 생각할 수 있는 간단한 측도로는 자원을 제공한 시간이나 자원이 수행한 단위 작업의 개수를 들 수 있다. 하지만 단지 하나의 응용만을 수행하는 것이 아니라 여러 개의 응용을 수행하기 때문에 각 응용별로 단위 작업의 수행 시간 등이 서로 다르고 자원의 능력이 달라 자원 제공에 대한 기여도를 정확하기 측정하는 것이 손쉽게 이루어지지는 않는다. 이를 위해 보다 균형있는 측도를 개발할 필요가 있다.

3.3.7 표준화와 상호 호환성

앞 절에서 언급한 설계 목표와 관련된 이슈들 외에도 여러 가지 해결해야 할 문제들이 존재한다. 먼저 표준화와 상호 호환성에 관련된 것을 들 수 있는데, 명확한 표준화 단체 또는 기업 표준이 정해져 있지 않은 상태에서 여러 시도가 이루어지고 있다. 이러한 시도들 중 최근 주목받고 있는 것이 BXXP(Blocks Extensible Exchange Protocol)[39]과 SOAP, WSDL 등을 활용하는 웹 서비스[40] 등이 있다.

BXXP는 IETF BEEP 워킹그룹에서 제안하였으며 응용 프로토콜 구현을 위한 XML 기반 메시징 프레임웍이다. BXXP는 인터넷 상의 XML 문서를 다루기 위한 도구로서 설계되었고, TCP 위의 레이어로 동작하는데 연결지향적(connection-oriented)이며 동시 다중 메시지 교환이 가능하면서도 MIME 인코딩이 가능하여 클라이언트-서버 메시징을 위한 HTTP를 대체할 수 있는 표준 프로토콜로서 P2P 메시징을 위한 호환성 있는 프로토콜로 등장하고 있다. 또한 XML 기반으로 RPC 형태의 호환성 있는 실행 흐름을 분산화 할 수 있는 웹 서비스가 널리 구현되어 사용되어 가고 있다. Korea@Home에서도 TCP 위의 레이어로서 SOAP 기반의 응용 메시징 프로토콜을 통해 향후 호환성을 염두에 두고 개발하고 있다.

4. 결 론

이제까지 P2P의 개괄과 이슈들, 그리고 Korea@Home을 중심으로 인터넷기반 분산컴퓨팅과 관련된 이슈들과 관련 연구들을 살펴 보았다. Korea@Home 시스템은 P2P 시스템으로서의 관점에서 볼 때 설계 목표를 만족하는 아키텍처를 그림 3과 같이 구성하고 있다.

앞에서 살펴본 이슈들 이외에도 성공적인 인터넷기반 분산컴퓨팅이 이루어지기 위해서는 비즈니스적인 측면이나 전통적인 분산 및 병렬 컴퓨팅 연구들이 많은 부분 적용되어야 할 것이다. 세계 최고 수준의 네트워크 인프라와 컴퓨팅 자원 수준에 비추어 볼 때 지속적으로 인터넷기반 분산컴퓨팅에 관련된 연구와 기술 개발이 뒷받침되어 준다면 매우 커다란 영향력을 가진 플랫폼으로 성장할 것이 틀림없다.

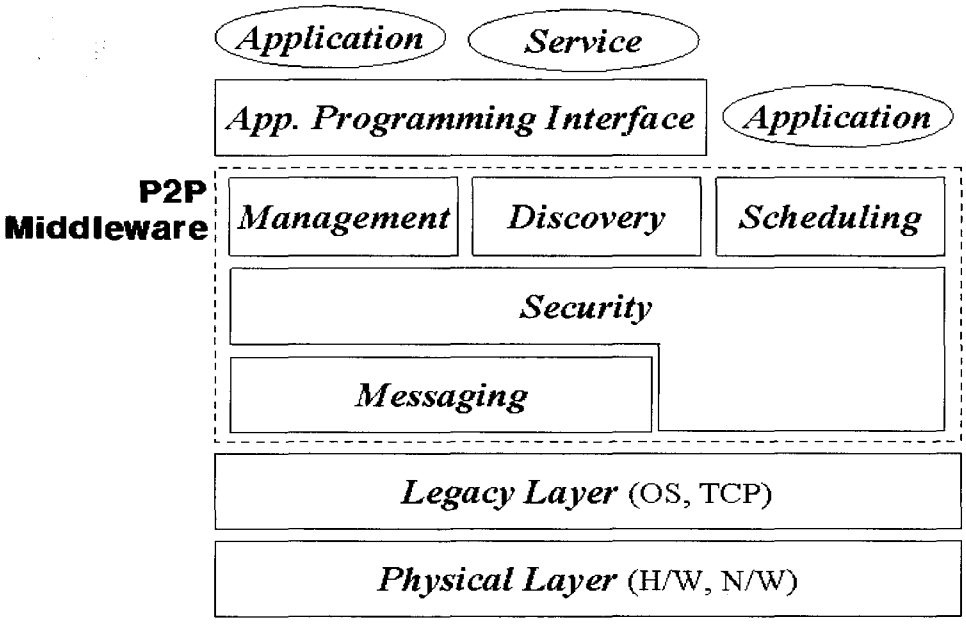


그림 3 Korea@Home 아키텍처

참고문헌

[1] Napster, http://www.napster.com/

[2] Mersenne Prime Searche, http://www.mersenne.org/prime.htm

[3] distributed.net: Node Zero, http://www.distributed.net/

[4] D.P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, "SETI@home: an experiment in public-resource computing," Communications of the ACM, Vol.45, No.11, pp.56-61, Nov. 2002 (See also http://setiathome.berkeley.edu)

[5] Korea@Home, http://www.KOREAatHOME.org/

[6] D.S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu, "Peer-to-Peer Computing," Technical Report HPL-2002-57, HP Labs. 2002

[7] D. Nowitz, UUCP Implementation Description, UNIX Programmer's Manual, Bell lab. Oct. 1978

[8] A.S. Tanenbaum, Computer Networks, Prentice-Hall, 1981

[9] P.E. Agre, P2P and the Promise of Internet Equality, Communications of the ACM, Vol.46, No.2, pp.39-42, Feb. 2003

[10] J. Lee, An End-User Perspective on File-Sharing Systems, Communications of the ACM, Vol.46, No.2, pp.49-53, Feb. 2003

[11] David Barkai, "Peer-to-Peer Computing: Technologies for Sharing and Collaborating on the Net," Intel- Press, 2002

[12] Andy Oram, "Peer-to-Peer : Harnessing the Power of Disruptive Technologies," O'Reilly & Associates, 2001

[13] Clay Shirky, "What is P2P...And What isn't", http://www.openp2p.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html

[14] D. Brookshier, D. Govoni, and N. Krishnan, "JXTA: Java P2P Programming", SAMS, 2002

[15] Gnutella, http://gnutella.wego.com

[16] B. Yang and H. Garcia-Molina, Comparing Hybrid Peer-to-Peer Systems, the VLDB Journal, pp.561-570, Sep. 2001

[17] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Integration, Technical Report Globus Project, 2002

[18] D.P. Anderson, Public Computing: Reconnecting People to Science, Proc. of the Conf. on Shared Knowledge and the Web, Madrid, Spain, Nov. 2003

[19] Luis F. G. Sarmenta, Volunteer Computing, Ph.D. thesis. Dept. of Electrical Engineering and Computer Science, MIT, Mar. 2001

[20] Hive Computing, http://www.tsunamiresearch.com/products/hivecomputing/

[21] BOINC: Berkeley Open Infrastructure for Net-work Computing, http://boinc.berkeley.edu

[22] GPU: Giga@lobal Processing Unit, http://gpu.sourceforge.net

[23] Avaki Data Grid, Avaki Corporation, http://www.avaki.com

[24] TurboWox, TurboWorx Inc., http://www.blackstonecomputing.com

[25] DCGrid, Entropia Inc., http://www.entropia.com

[26] Platform LSF, Platform Computing Inc., http://www.platform.com

[27] GridMP, United Devices, http://www.ud.com

[28] G. Tel, Introduction to Distributed Algorithms, Cambridge University Press, 1994

[29] N. Lynch, Distributed Algorithms, Morgan Kaufmann, 1996

[30] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications, Proc. of ACM SIGCOMM, pp.149-160, San Diego, Aug. 2001

[31] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, A Scalable Content-Addressable Network, Proc. of ACM SIGCOMM, pp.161-172, San Diego, CA, Aug. 2001

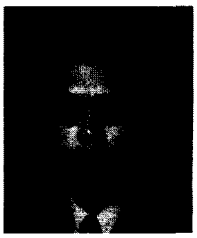
[32] A. Rowstron and P. Druschel, Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, Proc. of

IFIP/ACM Int'l Conf. on Distributed Systems Platforms, Nov. 2001
[33] K. Hildrum, J. Kubiatowicz, S. Rao, and B. Zhao, Distributed Object Location in a Dynamic Network, Proc. of 14th ACM Symp. on Parallel Algorithms and Architecture (SPAA), pp.41-52, Aug. 2002
[34] J.J. Kistler, Disconnected Operation in a Distributed File System, Lecture Notes in Computer Sciences, Springer-Verlag, 1995
[35] G.C. Necula, Proof-carrying code, Proc. of the 24th ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages, pp.106-119, Paris, France, Jan. 1997
[36] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V.S. Sunderam, PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Network Parallel Computing, MIT Press, 1994
[37] Message Passing Interface(MPI) Forum, http://www.mpi-forum.org/
[38] F. Dabek, B. Zhao, P. Druschel, and I. Stoica, Towards a Common API for Structured Peer-to-Peer Overlays. Peer-to-Peer Systems II: Second International Workshop, IPTPS 2003, pp.33-44, Berkeley, CA, Feb. 2003
[39] M. Rose, The Blocks Extensible Exchange Protocol Core, IETF RFC3080, http://www.ietf.org/rfc/rfc3080.txt?number=3080
[40] W3C Web Services, http://www.w3.org/2002/ws/

박 찬 열

1999. 11~2002. 3 (주)이씨오 연구개발 부장
2000. 2 고려대학교 컴퓨터학과, 이학박사
2002. 4~현재 한국과학기술정보연구원 슈퍼컴퓨팅센터 선임연구원
관심분야 : Distributed Computing, Mobile Computing, P2P, Fault-Tolerance
E-mail : chan@kisti.re.kr

길 준 민



1994. 2 고려대학교 전산학과, 이학사
1996. 2 고려대학교 대학원 전산학과, 이학석사
2000. 8 고려대학교 대학원 컴퓨터학과, 이학박사
1998. 3~2002. 9 고려대학교 기초과학연구소 연구원
2001. 6~2002. 5 Univ. of Illinois at Chicago, Post-Doc
2002. 10~현재 한국과학기술정보연구원 슈퍼컴퓨팅센터 초청연구원
관심분야 : 분산컴퓨팅, 인터넷 컴퓨팅, P2P 네트워크,이동무선 네트워크, 그리드 컴퓨팅 등
E-mail : jmgil@kisti.re.kr

최 장 원

1996 홍익대학교 전자공학과, 공학사
1998 홍익대학교 대학원 전자공학전공, 공학석사
1998~현재 한국과학기술정보연구원 슈퍼컴퓨팅센터 연구원
2003~현재 고려대학교 정보통신대학원 컴퓨터학과 박사과정
관심분야 : 분산컴퓨팅, 그리드 컴퓨팅, ATM, Network
E-mail : jwchoi@kisti.re.kr

박 학 수

1989 한남대학교 전자계산공학과(학사)
1991 한남대학교 대학원 컴퓨터공학과(석사)
2003 한남대학교 대학원 컴퓨터공학과(박사)
1991~현재 한국과학기술정보연구원 선임연구원
관심분야 : 소프트웨어공학, 분산컴퓨팅, 멀티미디어응용, 차세대네트워크(QoS, MPLS), 보안공학
E-mail : hspark@kisti.re.kr

이 필 우

1988. 2 동국대학교 전자공학과, 공학사
1991. 3 Univ. of Tsukuba 이공학연구과 컴퓨터공학전공, 공학석사
1999. 3 Univ. of Tsukuba 공학연구과 컴퓨터공학전공, 공학박사
2000. 4~현재 한국과학기술정보연구원 슈퍼컴퓨팅센터 선임연구원
관심분야 : 소프트웨어공학, 분산컴퓨팅, 멀티미디어응용, 차세대네트워크(QoS, MPLS), 보안공학
E-mail : pwlee@kisti.re.kr