

인터넷 기반 분산컴퓨팅 테스트베드 구축

최장원, 박찬열, 박학수, 이필우, 황일선
한국과학기술정보연구원 슈퍼컴퓨팅센터
e-mail : jwchoi@kisti.re.kr

Construction of Internet-based Distributed Computing Testbed

Jang-Won Choi, Chan Yeol Park, Hark-Soo Park, Pill-Woo Lee,
Il-Sun Hwang
Supercomputing Center, KISTI

요 약

이 논문에서는 인터넷 기반 분산컴퓨팅 테스트베드를 구축하고 이를 활용하여 '단백질과 리간드 결합 세기예측을 통한 신약후보물질탐색'을 수행한 결과를 보인다. 윈도우즈 계열의 임의 자발적 인터넷 사용자 1,217대의 PC를 대상으로 수행되었으며, 시스템이 목표로하고 있는 확장성, 가용성, 신뢰성, 적응성 등이 충분히 적용됨을 보이고, 실험을 통해 얻어진 교훈을 피드백으로 향후 연구 방향을 제시한다.

1. 서론

컴퓨터 네트워크와 웹 기술의 발전으로 인터넷을 이용하는 사용자가 기하급수적으로 증가되고 있다. 이에 비례하여 다양한 정보를 제공받기 위한 사용자의 요구사항 또한 질적·양적으로 높아져 응용분야가 넓어졌으며, 요구되는 계산양도 늘어나 거대한 컴퓨팅 파워를 필요로 하게 되었다. 상당히 많은 연산시간을 요구하는 대용량 어플리케이션을 단일 프로세서 상에서 작업을 수행시키는 것은 비효율적인 면이 있다. 실제로 이와 같은 작업들은 슈퍼 병렬 컴퓨터 즉, SGI Origin 2000, CrayT3e 및 IBM SP2 등과 같은 MPP(Massively Parallel Processing)시스템에서 작업을 수행하였다. 하지만, 일반 컴퓨터 사용자나 중급 수준의 사용자가 슈퍼 병렬 컴퓨터를 이용하여 작업을 하기란 현실적으로 상당한 경제적 부담과 컴퓨터 이용에 따른 기술적 문제가 있다. 이것을 해결하기 위해서 다수의 컴퓨터 자원을 사용하여 작업을 나누어 수행하는 병렬 처리에 대해 관심을 갖는 연구가 지속되고 있다. 그러나 LAN이라는 지역적으로 한정된 범위를 이용한다는 제한사항으로 어떤 임의의 작업을 수행하기 위해서는 여러 가지 환경 설정이 진제되어야 하는 치명적인 단점을 지니게 되었다. 이에 반하여 웹은 무수히 많은 호스트들을 연결하는 가상 시스템(virtual system)이 될 수 있다. 전세계에 걸친 이 가상 시스템을 따르는 멀티프로세서(multiprocessor) 시스템에 의해 많은 연산시간을 요구하는 어플리케이션의 병렬화를 이룬다면 높은 처리량(throughput)을 얻어 낼 수 있다.

현재 우리나라의 초고속 정보통신인프라 구축현황과 인터넷 이용률은 세계 최상위권에 속한다. 국내 PC 보급률과 인터넷 이용자 수는 매우 높지만 PC의 이용형태를 분석하여 보면 대부분 문서작성, 웹서핑 또는 게임 등에 이용하는 경우가 대부분이다. 특히 CPU의 성능과 메모리 용량은 기하급수적으로 증가하는 반면, 실제 CPU의 이용률

과 정보처리를 위한 메모리의 사용률은 낮다는 분석이 나온다. 다시 말하면 CPU의 유휴시간(idle time)이 많다는 이야기이다. 이와 같이 컴퓨터의 CPU를 이용하지 않는 시간에 작업을 수행하도록 하여 수백~수천만 유휴 컴퓨팅 자원을 통해 슈퍼컴퓨팅급의 작업 실행을 얻어 저비용, 고효율의 대용량 컴퓨팅 환경을 구현하고자 하며, 이를 @Home 프로젝트라 명명하였다.

이 논문에서는 한국과학기술정보연구원에서 수행하는 @Home 프로젝트인 인터넷기반 분산컴퓨팅 테스트베드 구축에 대해 설명하고, 이를 이용해 단백질과 리간드의 결합 세기예측을 통한 신약후보물질탐색 실험결과를 보여준다.

2. 기존 연구

지난 3-4년간 빠른 인터넷 및 네트워크 환경과 흠어져 있는 컴퓨팅 파워를 이용하려는 여러 연구가 국내외에서 이루어져왔다. 클러스터 컴퓨터를 구축하려는 시도를 비롯해 전통적으로 병렬컴퓨팅에 사용되는 Parallel Virtual Machine(PVM)과 Message Passing Interface(MPI)는 원격지 주스의 호스트에 존재하는 프로세스간에 서로 메시지를 전달하는 기본적인 단계를 마련해주었다. JPVM, JAPE는 기존의 PVM과 MPI를 자바 언어로 이식하여 이질적 환경을 극복하는 방안을 제시하였다. 그러나, 이들은 모두 LAN이라는 환경에 한정되어 있기 때문에 WAN에서 중요하게 생각되어야 하는 부하 균등 방법과 성능변화에 따르는 해결책은 제시하지 못했다. 특히, 프로그램 코드를 각 호스트에 배포하기 위해 그 호스트에 대한 계정(account)을 필요로 하는 등 사전 작업이 필요한 단점을 가지고 있어 동적으로 변화하는 프로세서들에 대한 문제를 해결하기 어렵다.

안정적인 속도와 신뢰성을 제공하지 못하며 중앙관리하기 어려운 특성을 가진 인터넷 환경에서 이질적인 컴퓨팅 파워들을 이용하고자 하는 여러 실험적 연구들이 진행되

어왔다.

JavaParty^[11]는 JavaRMI나 일반 소켓프로그램을 이용하여 원격객체를 구현할 때 발생하는 코딩 오버헤드를 줄이기 위해 임의의 호스트에서 생성된 객체를 여러 호스트에 자동으로 분산시키는 원격 객체에 대한 메커니즘을 마련하여 공유 메모리 공간을 형성해준다. ATLAS^[2]는 웹 자원을 이용한 컴퓨팅에 따른 여러 고려사항을 제시하며 자바와 Clik의 프로그래밍 기술을 접합하여 밀티쓰레드로 작성된 프로그램 기술을 네트워크 기반으로 확장하여 병렬 수행할 수 있는 설계 기법을 제시하였다. ParaWeb은 자바 병렬 클래스 라이브러리(Java Parallel Class Library)의 구축을 통해 메시지 전달 방법을 사용하여 웹 컴퓨팅의 자원을 활용할 수 있는 해결책을 제시하였다. 그러나 순수 Java를 사용하지 않고 JVM 인터프리터를 직접 수정하여 병렬 컴퓨팅 환경을 구현하였기 때문에 호환성 측면에 단점을 가지고 있다. Charlotte^[3]은 순수 자바를 사용하여 인터넷에 연결된 어떤 호스트라도 웹 브라우저만 있으면 자원을 제공할 수 있도록 설계된 자바 기반 병렬 플랫폼이다. JET^[4]는 마스터/작업 프로세서의 구조로 웹 기반 자바 병렬 라이브러리를 구축하고, 웹 환경에서 동작하는 여러 호스트들의 결합내성을 제공하기 위하여 체크포인팅(checkpointing)과 로깅(logging) 기법을 적용하여 프로세서들과 마스터의 태스크 제실정과 접속상태를 주기적으로 저장하는 방법을 사용하였다. POPCORN^[5]은 원격지에서 객체를 생성하여 작업을 수행한 후 결과를 돌려주는 Computelet라는 클래스를 정의하여, 이것을 기반으로 프로그래밍 모델을 제시하고 있다. Javelin^[6]은 기존의 자바 가상 머신과의 호환성을 목적으로 실행 시간에 작업 할당 전략과 비슷한 실시간 라이브러리(Load-Balancing Runtime Library)를 구현하였다. 광선 추적법(Raytracing) 및 메르센 소수(Mersenne Primes)의 탐색 프로그램을 어플리케이션으로 적용하여 그 수행 결과를 보였고, 낙관적 작업 전파(optimistic load propagation)에 기반한 간단한 작업 할당 전략을 제시하였다. 자바 애플릿간의 직접 통신이 가능하고 메시지 전달 방식으로 병렬 태스크간의 통신이 이루어지지만 결합내성과 동적 작업 프로세서 관리 기법을 제공하지 못하였고, 하나의 프로그램을 사용하므로 확장성이 부족하다는 단점 등이 있다. Java/^[7]는 지금까지 웹 컴퓨팅 기반의 객체 지향 병렬 프로그래밍을 지원하는 가장 적당한 프레임워크로 평가받고 있다. 하지만, 하위 레벨에서 분산 객체들간의 행동 방식과 프로그래머 입장에서의 그 객체들을 다루는 방법에 대한 연구에 집중하고 있고 동적으로 변화하는 작업 프로세서들에 대한 해결책을 지원하지 않으므로 웹의 컴퓨팅 자원들을 최대한 많이 이용할 수 있는 기반 아이디어는 부족하다. 국내에서도 몇몇 연구들이 진행되고 있다. 하지만 아직까지는 규모가 크지 않으며, 실용적인 수준에 머무르고 있어, 실제 다양한 응용 분야에 적용가능한 기반 기술로 발전하지 못하고 있다. 본 논문의 실용은 연구에만 그치지 않고 실제 임의의 자발적 인터넷 사용자를 대상으로 대용량의 시스템을 구축하고 그 결과를 살펴보았다.

<표 1>과 <표 2>는 대표적인 @Home 프로젝트의 사례들이다.

3. 프레임워크

구축하고자하는 시스템의 목적은 인터넷을 기반으로 일반 사용자의 유휴 컴퓨팅 자원을 활용하여 고성능 대용량 컴퓨팅 환경을 구축하는 것이다. 이러한 목적을 위해 갖추어야할 특징을 다음과 같이 설정하고, 디자인 목표로 하였다.

<표 2> 해외의 대표적 @Home 프로젝트

프로젝트명	참여기관	프로젝트 내용	시작년도/회원수
SETI @Home	UC Berkeley 대학	@HOME 프로젝트 원조, 외계진과 탐색	1999년/3,243,725명
Folding @Home	Stanford 대학	단백질 구조분석 프로젝트	2000년/23,000여명
Genome @Home	Stanford 대학	Folding@home과 형제 프로젝트로 Genome의 진화와 작용 연구	2001년/11,326명
Intel-UD Cancer Research	Intel, 미국립암연구소, 암협회, United Devices, 옥스퍼드대	단백질 상호작용 분석을 통한 암 신약선	2000년/908,846명
FightAIDS @Home	미국 Scripps Research Institute, Entropia	Autodock이라는 Protein-ligand docking simulation 프로그램으로 HIV 바이러스에 대한 신약신도발견 개발	2000년/34,566명
Moneybee	독일 i42사	신경망 이론을 바탕으로 주가 예측	2000년/1만명
Golem @Home	Brandies 대학, Entropia	가상 로봇을 이용한 시판레이션을 통하여 로봇 설계 연구	2000년/24,000명
Casino-21	영국 COAPEC, 옥스퍼드 대학	연속된 데이터로 이용해 가상모형을 평가하고 모델의 치명적인 결함 검증	1999년/4,155명
GIMPS	Entropia	Mersenne 소수 중 가장 큰 소수 찾기	1996년/19만명
Grub	미국 Grub사	P2P 방식의 점액 로봇엔진 프로젝트	2001년/400여명

<표 2> 해외의 @Home 프로젝트 성공사례

프로젝트명	참여기관	프로젝트 개요	결과물
Anthrax Research	UD, 미국립암 연구재단, 영국 옥스퍼드 대학 화학과	· 탄저균(Anthrax)을 제어할 수 있는 신약 후보물질의 탐색을 목적으로 24일간 90만대의 PC가 참여 · 총 35의 7천만개의 화합분자를 탄저균 단백질과 가상적으로 반응	탄저균에 대한 신약이 될 수 있는 매우 유망한 12,000개의 후보물질과 376,064개의 후보물질결론을 발견
Safer Market Project	미국 Entropia 사, 펜실바니아 대학, 로게스터 대학	· 나스닥과 S&P 지수 계산 및 환율변동 데이터를 이용하여 주식시장의 휘발성 문제 예측 · 9개월간 9,335대의 PC가 참여	· 970,885개의 태스크 계산 · 주식투자자들이나 주식전문가가 주식투자시 사용할 수 있는 예측 알고리즘을 일반에 공개할 예정
EvoChess Project	독일 도르트문트 대학	향후 진행될 몇 번의 수를 예측할 수 있는 체스 알고리즘의 설계 · 목적으로 6개월간 1,700대의 PC 참여	2,228,343개의 게임이 플레이되어 5번에서 10번의 수를 내다볼 수 있는 체스 알고리즘 도출을 도출

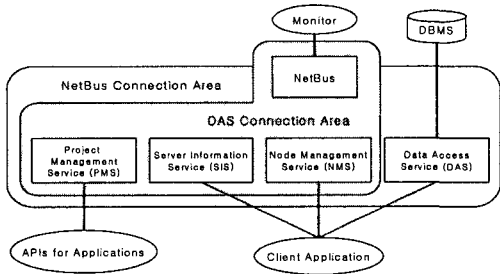
- 확장성(scalability) : 인터넷에 연결된 수많은 자원을 충분히 활용할 수 있도록 성능 향상이 되어야 하며, 이질적인 자원들간에 상호 호환성이 유지되어야 한다.
- 적용성(applicability) : 다양한 자원 및 응용 분야에 활용될 수 있도록 풍부한 API들을 제공하여야 한다.
- 신뢰성(reliability) : 자원에 대한 개인보호, 보안, 및 올바른 기능 수행이 보장되어야 한다.
- 적응성(adaptability) : 인터넷이라는 동적인 특성과 자원제공자의 자원활용을 방해하지 않도록 적응적이어야 한다.

구축하는 시스템의 주요 기능별 모듈들을 살펴보면 대규모 작업을 분산 처리에 적합하도록 분배하고 스케줄링하는 부분과 계산수행에 참여하는 컴퓨터, 즉 클라이언트의 작업 진행현황을 모니터링하고 자원을 관리하여 부하를 분산시키는 부분이 있다.

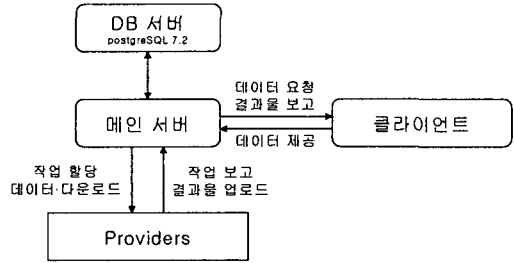
또한 수행된 작업결과를 모아 결과 데이터에 대한 신뢰도를 검증하는 모듈과 클라이언트의 유휴시간 동안 수행된 결과를 서버에 전송하여 주는 모듈로 구성되어 있으며 각 모듈별 기능은 아래와 같다.

① PMS(Project Management Service)

- 관리자가 정의한 프로젝트를 분산컴퓨팅 시스템의 분산 응용으로 생성 및 초기화
- 관리자가 프로젝트를 통해 수행하고자 하는 작업들을 Task



<그림 1> 시스템 아키텍처



<그림 2> 테스트 시스템 구성도

<표 3> 테스트 시스템 기능 설명

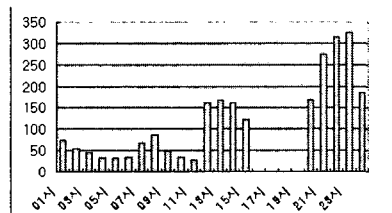
구성	주요 기능	기능 설명
Server	Work Spec 제공	Provider에게 Work Spec 제공
	Provider/Storage Access 관리	Provider와 Client 및 Storage의 Access 관리
	메시지 처리	Provider/Client/Storage 메시지통신관리
	스트림서비스 할당	Provider의 업로드/다운로드 스트림 관리
	데이터베이스소스 관리	Work 데이터베이스 관리
	데이터 제공	시버를 통해 Work 데이터 제공
Client	결과 저장	Work 수행 결과물 저장
	프로그램 실행	자동 실행 (작업은 Background 방식으로 수행)
	OS Type 분석	OS Type(win95/98/98SE/2000/XP)
Provider	CPU 분석	CPU Type, Cpu Time(98이하의 불가능)
	IP address 분석	사용자의 IP Address 분석
	Work Spec 다운로드	용용개발자가 정의한 작업 Spec 다운로드
	Work 다운로드	Work SPEC에 정의된 Work 다운로드
Provider	Work수행, Upload	Work을 수행하고 서버에 접속 Upload
	Run, Stop, Exit	사용자가 Run(작업시작),Stop(작업멈춤),Exit를 시행 가능

- 객체로 생성하며 DAS를 통해 프로젝트 고유의 Task로 저장 및 관리
- 관리자는 초기화된 프로젝트를 전체 분산컴퓨팅 시스템에서의 수행에 대해 제어가능
- ② SIS(Server Information Service)
- 보유모듈을 활용, 노드의 확장과 트랜잭션 규모의 확대에 따라 다중적, 중층적으로 구성된 서버 시스템에 대한 정보를 서버, 노드, 사용자에게 제공
 - 노드에서 접근 가능한 최적의 서버 검색 서비스 제공
 - 서버 시스템의 확장 또는 특정 서버의 변화의 전체 시스템에 대한 투명성 관리
- ③ NMS(Node Management Service)
- 보유모듈을 활용, 개별 클라이언트가 분산컴퓨팅 시스템의 노드로서 작동할 수 있는 서비스
 - 노드에서 발생하는 기본적인 요청 처리
 - 노드의 heartbeat, 작업 수행 및 보고 등 노드의 상태를 추적하고 보고
 - 서버 시스템에서 발생하는 상태 변화, 제어 정보 등 노드 관련 이벤트를 각 노드에 서비스
- ④ DAS(Data Access Service)
- 보유모듈을 활용, 데이터베이스와 연동해 작업결과 및 프로젝트 관련 객체를 저장 및 제공
 - 분산컴퓨팅 시스템의 상태정보를 관리 및 일관성 유지
 - 접근 권한을 가진 서버와 노드의 데이터 및 상태 정보에 대한 요청을 처리
- ⑤ NetBus Communicator
- 보유 모듈을 활용하여 서버시스템간 이벤트 교환을 수행 제어함으로써 서버시스템의 단순화와 유연한 확장을 가능하게 함
 - 각 서버는 자신의 이벤트를 알리기 위해 다른 서버들에 대한 사전 정보를 갖고 있지 않아도 되며 이벤트 교환 모듈을 통해 보고하면 됨
- ⑥ APIs for Applications
- 용용 프로젝트를 인터넷 분산컴퓨팅 시스템에 적용하기 위해 작성하는 API
 - 용용 개발자가 개발한 용용을 가상탐색용 인터넷 분산컴퓨팅 시스템에 적용할 수 있도록 제공
- ⑦ 분산컴퓨팅 클라이언트 에이전트
- 작업에 참여할 사용자들의 PC에 설치되는 가상탐색용 분산컴퓨팅 에이전트 소프트웨어
 - 프로젝트 수행에 필요한 데이터 및 파일을 수집
 - DAS로부터 태스크 및 데이터를 공급받으며 작업 수행
 - 각 작업 수행에 따른 결과 또는 오류사항 보고

4. 테스트베드 구축 및 실험

구축된 시스템에서 수행한 용용은 '신약후보물질 탐색'이다. 이는 단백질 상호작용을 분석하여 질병과 연관된 타겟 단백질을 찾아내고, 선정된 타겟단백질과의 상호작용 분석을 위해서 타겟단백질과 상호작용 시뮬레이션에 적용할 화합물 정보를 구축하는 것을 목표로 한다. 실제 연산 내용은 리간드 데이터베이스로부터 추출된 분자를 타겟 단백질의 활성자리에 도킹하여 에너지 계산 및 이 에너지 기반의 결합세기 함수를 통한 결합의 세기를 예측한다.

테스트된 시스템의 다양한 구성도는 <그림 2>와 같으며, 약 4일간 1,217대의 PC를 이용하여 시스템의 안정성 및 성능, 연결된 PC들의 평균 접속률, 적합한 작업의 크기, 대역폭, PC들의 평균작업 시간 등을 살펴봐왔다. 시스템에서 수행되는 작업의 양에 관련된 파라미터로서, 리간드는 140개를 사용하였고 이 리간드와 한 종의 단백질에 대해 결합세기 함수의 3가지 파라미터를 달리하여 작업을 수행하였다. 각 개인 컴퓨터가 수행할 작업을 위해 서버로부터 받는 파일의 크기는 40~55KByte 가량이었고, 각 개인 컴퓨터가 작업을 마치고 돌려보내는 결과 파일의 크기는 약 190KByte 크기였다. 각 개인 컴퓨터에서 수행되는 작업 시간의 평균은 약 20분이었으며, 전체 작업이 수행되는 데 약 2시간 30분 가량이 소요되었다. 같은 작업을 한대의 컴퓨터(Pentium3 550MHz 기준)로 수행하는데 걸리는 시간은 약 160시간 가량이다. 이들 작업에 대한 개략적인 흐름은 <표 4>에서 보여준다.



<그림 3> 시간별 작업결과물 수

<표 5> 테스트 진행 과정

No	테스트 진행 과정
①	사용자가 Provider 다운로드(ActiveX Control)
②	Provider가 서버에 접속하여 Work Spec(XML포맷) 다운로드
③	Work Spec의 정의에 따라 Work 다운로드
④	작업 수행
⑤	작업완료시 서버 접속 및 결과 업로드
⑥	서버에서는 결과를 비교하여 내용이 같을 경우 삭제, 내용이 다를 경우 보전
⑦	결과를 응용쪽 Client에 보냄
⑧	Client에서는 파일 형태로 저장

<그림 3>에서 보여주듯 22시~24시까지 사용자들이 인터넷을 가장 활발히 사용하는 시간대에 작업을 제일 많이 수행하였으며, 총 1,217대가 작업 수행에 참여하였지만 어느 순간에 작업에 참여하는 자원제공자의 수는 최대 326, 최소 29이었다. 테스트 참여 PC의 기본적인 통계는 아래와 같다.

<표 6> OS 종류별 분포 (Windows ME는 Windows98 SE에 포함, Windows XP는 Windows2000에 포함)

OS 종류	대수(대)	구성비율(%)
Windows 98	94	7.7
Windows 98 SE	767	63.0
Windows 2000	340	27.9
미인식	16	1.3
합 계	1,217	100.0

<표 8> CPU 클럭속도별 분포

CPU 클럭속도	대수(대)	구성비율(%)
500MHz 미만	230	18.9
500MHz~1GHz 미만	703	57.4
1GHz 이상	282	23.2
미인식	2	0.1
합 계	1,217	100.0

<표 9> RAM 용량별 분포

RAM 용량	대수(대)	구성비율(%)
128M 미만	242	19.9
128M~256M미만	524	43.0
256M	350	28.8
256M 초과	99	8.1
미인식	2	0.2
합 계	1,217	100.0

테스트 기간동안 최고 성능은 53.5Gigaflops로 이는 P-III 550MHz 전용 PC 107대, P-IV 1GHz PC 53대를 병렬로 연결하여 수행하는 속도에 해당되며 일일평균 성능은 31.78 Gigaflops로 나타났다.

$$Teraflops = AM * MHz * Utiliz. * flops/cycle \quad [111]$$

* AM = Active Members(results received last two weeks)
 * MHz = Average MHz in pool
 * Utiliz. = Average utilization the program receives
 * Flops/cycle = flops capable for processor

이번 테스트를 통해 분산컴퓨팅 시스템 부문에서는

- 사용자 모듈의 독립 프로그램으로의 패키징
- 스크린셰어 모드의 필요성
- 작업 스펙 확장 및 사용자 API 보완
- 작업 에러를 통계 보완

응용 부문에서는

- 응용프로그램의 분산컴퓨팅 시스템용 최적화
- 결과물 검증 프로그램 개발
- 운영체제별 응용프로그램 최적화

의 문제가 대두되었다.

실험을 통해 도출된 교훈을 반영하여 향후 연구 및 개발 계획은 다음과 같다.

- 자원제공자의 컴퓨팅 자원에 대해 소유자의 사용에 방해가 되지 않기 위해, 보다 융통성 있는 설정 제공
- 다양한 응용에 활용될 수 있도록 작업에 대한 스펙을 정의한 규칙 확장
- 자원제공자에 자원이용에 대해 충분한 이해와 공공성을 알리기 위한 가시화 모듈 개발
- 응용 프로그램의 최적화 방법론 개발
- 보다 다양한 플랫폼 및 디바이스의 활용 테스트

5. 결론

방대한 계산량 때문에 100만~300만 정도의 화확분자 DB를 구성한 뒤 필터링 방식을 적용하여 1만~2만개의 화확분자만을 가상탐색을 해오던 기존 시스템의 한계점을 인터넷에 연결된 PC를 이용하여 해결하기 위해서 1,217대의 PC로 가상탐색을 실시하였다. 시스템의 구조는 서버/클라이언트/자원제공자(Provider)의 형태로 되어있으며, 480개의 작업을 자원제공자에게 분배하여 모든 작업 완료 시까지 총 2시간 30분 가량 작업을 수행하였다. 이를 한대의 개인용 컴퓨터에서 수행하는 경우 160시간의 작업이 필요하다. 작업을 할당받은 자원제공자는 자신의 유휴자원을 이용하여 작업을 하게되며 190K정도의 결과물이 생성되며 이를 서버에 보낸다.

우리나라의 발달된 네트워크 인프라 덕분에 해외의 유사 프로젝트와 비교할 때 같은 자원제공자수에도 더 좋은 성능을 얻을 수 있음을 알 수 있었으며, 이번 실험은 약 1200여대의 일반 사용자를 대상으로 수행되었지만 도출된 개선사항을 보완하여 올해 말에 5000대 규모, 내년 초에 1만대 이상의 규모로 테스트를 수행할 예정이다.

참고문헌

- 11) Michael Philippsen and Matthies Zenger. "JavaParty-Transparent Remote Objects in Java." In the ACM Workshop on Java for Science and Engineering Computation, 1997
- 12) J. E. Baldeschwieler, R. D. Blumofe, and E. A. Brewer. "ATLAS : An Infrastructure for Global Computing." In Proc of the 7th ACM SIGOPS European Workshop : System Support for Worldwide Application, September 1996.
- 13) A. Baratloo, M. Karaul, Z. M. Kedem, and P. Wychoff. "Charlotte : Metacomputing on the Web." The 9th International Conference on Parallel and Distributed Computing Systems, Dijon, France, <http://www.cs.nyu.edu/milan/charlotte/index.html>, September 1996.
- 14) Hernani Pedroso, Luis M. Silva, Victor Batista, Paulo Martins, Guilherme Soares and Telmo Menezes. "Web-based Metacomputing with JET." In Proc. of Concurrency : Practice and Experience, June 1997.
- 15) N. Camiel, S. London, N. Nisan, O. Regev. "The POPCORN Project : Distributed computing over the Internet in Java." In Proc. 6th International World Wide Web Conference, April 1997.
- 16) B. O. Christiansen, P. Cappello, M. F. Ionescu, M. O. Neary, K. E. Schauer, D. Wu. "Javelin : Internet-Based Parallel Computing Using Java." ACM Workshop on Java for Science and Engineering Computation, June 1997.
- 17) D. Caromel, W. Kausser, J. Vayssiere. "Java// Towards Seamless Computing end Meta- computing in Java", Concurrency Practice and Experience, pp. 1043-1061, September, 1998
- 18) 한인회 · 박찬원 · 정영식 · 황종신, "성능 기반 태스크 할당을 이용한 웹 기반 병렬처리 시스템의 설계", 한국정보과학회논문지, 제27권 3호, pp.264-276, 2000년
- 19) 신필섭, 정준목, 명해신, 홍원기, 김신덕, "인터넷 기반의 병렬 컴퓨팅을 위한 사용자 라이브러리 설계 및 성능 분석", 정보처리학회 논문지, Vol. 6, No. 11, pp.2932-2946, 1999년
- 110) 김승조, 이상성, 손경우, 하병연, 조진연. "인터넷 슈퍼컴퓨팅 기술의 구현" 한국항공우주학회지 제29권 제3호 pp.28-37, 2001년
- 111) The Intel Philanthropic Peer-to-Peer Program, <http://www.intel.com/cure/>