JXTA 기반 단백질 구조 비교 시스템

정효숙[†] · 안진현[††] · 박성빈[†††]

요 약

단백질 구조 비교는 각 단백질에 존재하는 수많은 원자들을 처리해야 하므로 많은 컴퓨팅 자원을 요구하는 작업이다. 이러한 작업을 처리하기 위한 접근법으로써 그리드 환경에서 시간 소모가 큰 계산 작업을 분산 처리하는 것이 널리 사용되어 왔다. 그러나 이러한 그리드 환경을 통제하는 것은 비전문가들에게 쉽지 않을 수 있다. 본 논문에서는 비전문가들도 쉽게 그리드 환경을 통제할 수 있는 JXTA 기반의 단백질 구조 비교 시스템을 제안한다. 쿼리 단백질과 비슷한 단백질들을 찾기 위해 사전처리 단계 와 인식단계로 구성된 기하학적 해싱 알고리즘이 사용되었다. 실험 결과에 의하면 주어진 쿼리 단백질 구조와 일치하는 단백질 구조를 시스템이 정확히 찾고 또한 제안된 시스템은 쉽게 단백질 다킹 문제를 해결하도록 확장될 수도 있다. 본 논문에서 제안하는 시스템은 비전문가들, 특히 생물학이나 화학을 전공하는 대학생들처럼 일반적으로 분산 시스템에 대한 숙련된 지식이 없는 사용자들에게 도움이 되리라 기대된다.

주제어 : JXTA, 단백질 구조 비교

A JXTA-based system for protein structure comparison

Hyo-sook Jung[†] · Jin-hyun Ahn[††] · Seong-bin Park[†††]

ABSTRACT

Protein structure comparison is a task that requires a lot of computing resources because many atoms in proteins need to be processed. To address the issue, Grid computing environment has been employed for processing time-consuming jobs in a distributed manner. However, controling the Grid computing environment may not be easy for non-experts. In this paper, we present a JXTA-based system for protein structure comparison that can be easily controled by non-experts. To search proteins similar to a query protein, the geometric hashing algorithm that consists of preprocessing and recognition was employed. Experimental results indicate that the system can find the correct protein structure for a given query protein structure and the proposed system can be easily extended to solve the protein docking problem. It is expected that the proposed system can be useful for non-experts, especially users who do not have sophisticated knowledge of distributed systems in general such as college students who major in biology or chemistry.

Keywords : JXTA, Protein Structure Comparison

† 종신회원: 고려대학교 컴퓨터교육학과 박사과정
†† 종신회원: KAIST 전산학과 박사과정
††† 종신회원: 고려대학교 컴퓨터교육과 교수(교신저자)
논문접수: 2009년 5월 9일, 심사완료: 2009년 6월 25일

1. 서 론

분자 모델링 기법을 이용한 의약품 설계는 잠재적 약물들을 확인하기 위해서 화학 데이터베이스의 수많은 분자 화합물들을 확인하는 것과 관련되어 있다. 이러한 과정을 분자 다킹이라 하는데, 이는 과학자들이 작은 분자들이 어떻게 3차원 구조로 된 효소나 단백질 수용체와 결합하는지를 예측하는데 도움을 주고 있다. 분자 다킹은 계산 및 데이터 집약적 작업을 요구하기 때문에, 효율적으로 이를 처리하기 위해 최근 다양한 분산 시스템들이 개발되고 있다[9].

분산 컴퓨팅은 거대한 계산 문제를 작은 작업으로 분할하여 인터넷에 연결된 여러 컴퓨터들에 할당시킴으로써 문제를 해결하려는 온라인 협력의 새로운 형태이다. 강력한 처리 능력을 지닌 고가의 컴퓨터를 구입하지 않더라도 수많은 개인용 컴퓨터들의 처리 능력을 이용하여 컴퓨팅 집약적 문제를 해결할 수 있기 때문에 단백질 데이터와 같이 방대한 데이터를 성공적으로 처리할 수 있는 방식이라 볼 수 있다[10]. 그러나 분산 컴퓨팅에 대한 전문적인 지식이 없는 사용자들이 사용하기에 쉽지 않다는 단점을 지니고 있다. 특히, 전문가를 대상으로 설계된 시스템을 운영하고 있는 실험실이나 연구실에 방문하지 않고도 언제 어디서나 필요한 정보를 얻고자 하는 사람들이 분산 컴퓨팅 환경을 활용한다는 것은 어려울 수밖에 없다.

예를 들어, 대학생들이 수업 시간에 단백질 구조 비교에 대한 개념을 배운다고 가정하자. 한 강의실에서 사용할 수 있는 컴퓨터의 개수가 부족하여 다른 강의실에 있는 컴퓨터들을 동시에 연결하여 단백질 구조 비교 실험을 할 수 있는 분산 컴퓨팅 환경을 구축하려고 한다. 이때, 각 강의실에 필요한 시스템들이 미리 설치되 있어야 하고 또한 강의실마다 방화벽과 같은 보안 시스템이 존재할 수도 있으므로 이를 어떻게 처리할 것인지에 대해서도 대비해야 한다. 학생들은 서로 다른 강의실에 있는 학생들과의 커뮤니케이션이 가능해야 하며, 만일 학생들이 어디에 있던지 시스템에 접근할 수 있도록 모바일 기기 사용이 가능하다면 더욱 편리할 것이다.

본 논문에서는 네트워크 환경과 기기의 유형에 상관없이 쉽게 설치할 수 있고 개발자들에게 방화벽이나 기종에 상관없이 데이터 전송이 진행되는 것을 직접적으로 드러내지 않는 API를 제공하는 경량 프레임 워크를 사용하고자 한다. 특히, JXTA는 이러한 문제를 해결할 수 있는 표준 프로토콜을 제공한다고 볼 수 있다.

본 논문에서 제안한 시스템은 어떤 컴퓨팅 환경에서도 쉽게 설치하고 사용할 수 있도록 개발된 시스템으로써, 기존의 단백질 구조 비교 시스템의 성능을 개선하기 보다는 학생들이 학교의 유휴 컴퓨터 자원을 활용하여 쉽게 단백질 구조 비교 실험을 수행할 수 있는 환경을 제공하는 교육용 시스템으로써 활용하고자 한다. 비록 단백질 구조 비교를 위한 고가의 장비나 시스템에 대한 전문적 지식이 없더라도 본 논문에서 제안한 시스템을 사용한다면, 학생들은 유휴 컴퓨터의 메모리나 컴퓨팅 능력을 활용하여 쉽게 분산 처리 환경을 구축하고 단백질 구조 비교 실험을 수행할 수 있을 것이다.

2장에서는 본 논문과 관련된 연구들을 소개하고 3장에서는 제안된 시스템을 보다 자세하게 설명할 것이다. 또한 4장에서는 시스템의 실험 결과를 소개하고 5장에서 본 논문의 결론 및 향후 연구 과제를 제시하고자 한다.

2. 관련 연구

단백질 구조 비교는 바이오인포매틱스에서 다루는 주요 문제 중 하나로써, 3차원 단백질 분자 구조를 비교하는 것은 힘든 작업으로 알려져 있다. 의약품 설계 과정, 단백질 구성의 새로운 형태 확인, 단백질의 유기적 구조 분석 등을 지원할 수 있고, 예측하지 못했던 단백질들의 진화 및 이들 간의 기능적 관계를 발견할 수 있도록 도와야 하므로 단백질 구조 비교를 위한 효과적인 해결 기법에 대한 연구가 요구되고 있다[11].

PDB 아카이브는 수많은 단백질과 그 외 생물학적 고분자를 설명하기 위해서 이들의 원자 좌표 및 다른 관련 정보를 저장하고 있다. PDB 아카이브에서 단백질과 핵산의 구조뿐만 아니라, 리

보솜, 종양 유전자, 약물 표적, 바이러스 등에 대한 구조도 찾을 수 있다[8]. PDB 데이터는 생물학 분자들의 좌표 정보를 파일로 저장하고 있다. 이 파일들은 각 단백질의 원자들 및 이들의 3차원 위치를 기술하고 있으며, PDB, mmCIF, XML 등 다양한 포맷을 지원하고 있다[8].

기하학적 해싱은 후보 객체들 중에서 주어진 객체를 인식해내는 컴퓨터 비전 (computer vision) 분야에서 개발된 알고리즘이다 [4][5]. 기하학적 해싱은 단백질 구조를 비교하는데 사용되고 있다. 사전 처리 단계에서 모든 단백질 객체들의 기하학적 정보를 추출하여 해시 테이블에 중첩적으로 해시한다. 인식 단계에서 주어진 쿼리 단백질과 동일한 해시 값을 갖는 후보 단백질을 우선 검색한 후, 그 중에서 유사한 단백질 구조를 갖고 있는 단백질을 찾아낸다. 모든 객체를 검색하는 것이 아니라 해시 값이 일치하는 후보 객체들을 먼저 검색한 후, 실제 일치하는 객체를 찾기 때문에 보다 효율적인 검색을 가능하게 한다.

Pennec과 Ayache [6][7]는 기하학적 해싱 알고리즘을 사용하여 단백질 구조를 비교하는 방식을 제안하였다. 단백질의 원자들 중에서 α-helix와 β-sheet를 만드는 시작 잔기인 N-Cα-C 구조를 추출하여 레퍼런스 프레임을 구성하고 단백질을 Cα에 의해 연결된 아미노산의 시퀀스, 즉, 일차원 구조로 표현된 단백질을 다룬다. Ferrari 등 [3]은 이차원 구조로 표현된 단백질 데이터를 이용하여 단백질 구조를 비교하기 위해서 기하학적 알고리즘을 기반으로 분산 시스템을 제안하였다.

그리드 컴퓨팅은 대용량의 컴퓨팅 작업을 처리하기 위해서 수많은 컴퓨터들을 연결시켜 가상의 네트워크를 형성하여 지리적으로 분산된 자원들을 공유하면서 협력적으로 문제를 해결하는 것을 말한다[12]. 의약품 설계를 위한 분자 모델링은 고도의 연산 능력을 요구하기 때문에, 그리드 기술을 이용한 애플리케이션들이 개발되고 있다[9].

그러나 그리드 미들 웨어와 같은 복잡한 소프트웨어를 일반 인터넷 사용자가 설치하고 사용하기에 너무 어렵다기 때문에, 그리드 기술을 단지 학문적 목적 뿐만 아니라, 일반적 목적으로 활용하는데 어려움이 따른다. 반면 JXTA와 같은 P2P 기술을 이용할 경우, 다양한 분산 서비스 및 애플리케이션 개발이 가능하므로, 실제 그리드 환경은 아니지만, 상대적으로 소규모 컴퓨팅 작업을 수행하는 그리드 환경을 구축할 수 있다[13].

JXTA는 애플리케이션 개발을 위한 미들웨어로써 네트워크상에 연결된 상호 이질적인 장치들 간에 통신과 협력을 가능하게 하는 P2P 프로토콜이다. JXTA 피어는 한 개 이상의 프로토콜을 구현한 네트워크 장치로써 세 가지 형태의 피어가 존재할 수 있다. 에지 피어는 메시지를 다른 피어에 전송할 수 있는 가장 기본적인 피어이다. 랑데부 피어는 XML로 작성된 다른 피어들의 정보를 보유하고 있는 피어로써, 여러 개의 에지 피어가 만나는 지점이 된다. 랑데부 피어는 피어 그룹의 초기 접근 포인트를 제공하며, 에지 피어도 동적으로 랑데부 피어가 될 수 있다. 즉, 일반적인 서버 타입 P2P 프로토콜과 달리 중앙 서버가 필수적이지 않다. 릴레이 서버는 일종의 중계자의 역할을 수행한다. 큰 데이터를 임시 저장할 정도의 메모리가 없고 상대적으로 작업 처리 속도가 느린 모바일 기기는 릴레이 서버를 통해서 데스크 탑에 메시지를 전송하게 된다. 릴레이 서버는 방화벽 내부에 있는 피어에 메시지를 전송하는 것뿐만 아니라 다른 서브 네트워크의 보호를 받고 있는 피어들이 서로 상호작용할 수 있도록 지원한다 [1].

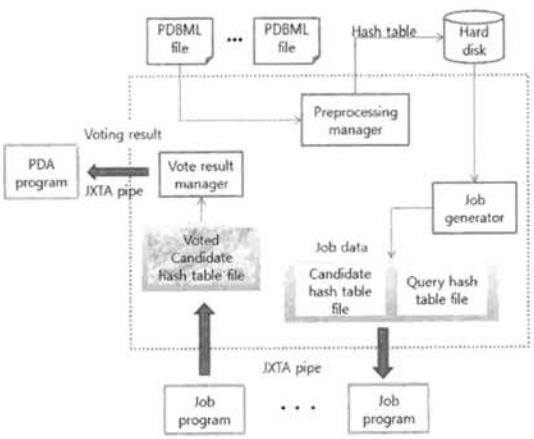
개발자들은 방화벽 내부의 피어들에게 메시지를 전송하거나 방화벽 밖의 피어로부터 메시지를 전송 받기 위해서 환경설정 파일에 릴레이 피어의 IP 주소와 포트 번호를 설정할 수 있다. JXTA 네트워크에서 다른 피어에 대한 정보를 얻으려는 대부분의 작업은 XML로 작성된 각 피어의 광고 문서를 탐색해냄으로써 달성된다. 광고 문서에는 피어, 피어 그룹, 피어 파이프 등에 대한 정보가 표현되어 있다. 피어 및 피어 그룹 광고는 피어의 이름, ID, 프로토콜, 서비스 등에 대한 정보를 포함하고 있다[2].

3. 단백질 구조 비교 시스템

본 논문에서 제안한 시스템은 서버 프로그램, 작업 프로그램, PDA 프로그램으로 구성되어 있다. 서버 프로그램은 사전 처리 단계를 수행한다.

서버 프로그램은 사용자의 쿼리를 받았을 때, 이를 처리하기 위해서 작업 데이터를 작업 프로그램으로 보내고 쿼리와 일치하는 단백질을 찾는 인식 단계를 완료한 후, PDA 프로그램에게 투표 결과를 보낸다. 작업 프로그램은 인식 단계를 수행하기 위해서 작업 데이터를 기다리고 있다가 서버로부터 작업 데이터를 받으면 인식 단계를 수행한 후, 처리 결과를 서버 프로그램으로 보낸다. PDA 프로그램은 사용자가 쿼리 단백질을 선택하여 이를 서버로 보내고 서버로부터 받은 투표 결과를 볼 수 있는 사용자 인터페이스를 제공한다. 각각의 기능을 보다 자세히 살펴보면 다음과 같다.

<그림 1>은 서버 프로그램의 구조로써 JXTA 파이프를 이용하여 다른 프로그램과 상호작용하는 과정을 설명하고 있다. PDBML 파일은 단백질 데이터로써 단백질에 대한 정보를 갖고 있는 XML 문서이다 [8]. 각 PDBML 파일은 단백질 구성요소들의 집합으로써 단백질에 존재하는 원자들에 대한 설명을 제공한다. 각 구성요소마다 각 원자의 이름, 3차원 좌표값, 잔기 이름 등의 정보를 포함하고 있다.



<그림 1> 서버 프로그램의 구조

사전 처리 단계에서 처리할 PDBML 파일들은 서버 프로그램에 미리 저장되어있다. 사전 처리 관리자 모듈은 서전 처리할 PDBML 파일들을 선택하여 각 PDBML 파일의 해시 테이블이 생성한다. 이는 나중에 하나의 쿼리 단백질로 사용될 것이다. 모든 PDBML 파일들의 해시 테이블은 파일의 형태로 하드 디스크에 저장된다.

후보 해시 테이블은 후보 단백질로부터 생성된 해시 테이블이고 쿼리 해시 테이블은 쿼리 단백질로부터 생성된 해시 테이블이다. 쿼리를 인식하는 작업을 작업 프로그램에 쉽게 분배하기 위해서 후보 해시 테이블과 쿼리 해시 테이블은 같은 구조를 갖고 있도록 생성된다.

서버 프로그램은 PDA 프로그램의 사용자 인터페이스를 통해 쿼리 단백질로 사용할 수 있는 단백질의 ID를 사용자에게 보여준다. 사용자가 쿼리 단백질 하나를 선택하면, 작업 생성자 모듈이 쿼리 단백질과 유사한 단백질을 찾기 위해서 그 쿼리 단백질의 해시 테이블 파일과 후보 해시 테이블 파일들을 준비한다. 현재 JXTA 네트워크에 연결된 작업 프로그램의 개수에 따라서, 작업 데이터가 서브 작업 데이터들의 집합으로 분할된다. 서브 작업 데이터는 쿼리 해시 테이블과 작게 분할된 후보 해시 테이블로 구성되어 있다. 작업 생성자 모듈은 작업 프로그램의 상태 즉, 네트워크에 연결되어 있는지, 현재 서브 작업 수행 중인지, 서브 작업을 마치고 새로운 작업 수행을 위해 대기 중인지 등을 확인하면서 서브 작업 데이터를 반복적으로 작업 프로그램들에게 보낸다.

작업 프로그램은 서버 프로그램으로부터 받은 쿼리 단백질의 해시 테이블 파일과 후보 해시 테이블 중 일부를 비교하여 쿼리와 일치하는 정보를 갖는 단백질에 투표하는 인식 작업을 수행한 후, 후보 해시 테이블 파일에 투표된 결과를 서버 프로그램에 보낸다. 이때, 후보 해시테이블 파일에 투표된 결과만 필요하므로 쿼리 해시 테이블 파일이 서버 프로그램으로 보내질 필요는 없다.

투표 결과 관리자 모듈은 각 작업 프로그램에서 처리하여 보낸 후보 해시 테이블에 대한 투표한 결과를 모아서 전체 후보 해시 테이블에 대한 결과를 갱신한 후, 투표 결과를 PDA 프로그램으로 보내어 사용자가 확인할 수 있도록 한다. 서버 프로그램, 작업 프로그램, PDA 프로그램 간의 모든 데이터 전송은 JXTA 파이프를 통해서 이루어진다. JXTA는 다양한 디지털 장비에 P2P 시스템을 적용할 수 있으므로 프로그램을 실행하기 위해서 어떤 기기를 사용할 것인가는 시스템 아키텍처의 선택에 따라 달라진다. 서로 다른 방화벽 보호 안에 있는 기기에서 프로그램이 운영되더라

도 릴레이 피어를 통해 연결될 수 있다. 즉, 릴레이 피어가 알맞게 설정되어 있다면, 임의적인 네트워크 환경에서도 프로그램들이 배치될 수 있다.

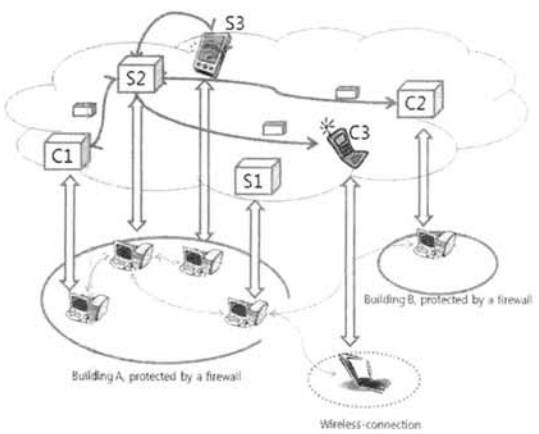
4. 실 험

본 논문에서 제안한 세 가지 프로그램을 6개의 기기에 각각 설치하고 11개의 단백질 데이터 파일을 사전 처리 및 인식한 실험 결과를 제시하고자 한다. <표 1>은 각 기기의 하드웨어 정보와 실험에서 수행한 역할을 보여주고 있다.

<표 1> 각 실험 기기의 환경 설정 정보

Index	Roles	OS	Device type
S1	Relay peer Rendezvous peer	Windows	Desktop
S2	Server program	Windows	Desktop
S3	PDA program	Windows	Desktop
C1	JXTA-J2SE	MacOS	Desktop
C2	JXTA-J2SE	Linux	Desktop
C3	JXME program	Windows	Laptop

<그림 2>는 실험에 사용된 기기들이 두 개의 다른 건물에 어떻게 배치되어 있고, JXTA 네트워크에서 어떻게 연결되어 있는지 보여주고 있다. 빌딩 A에는 하나의 방화벽 보호 아래 데스크 탑 네 대가 존재한다. 빌딩 B에는 또 다른 방화벽 보호 아래 데스크 탑 한 대가 존재한다. 랩톱 한 대는 무선 랜으로 연결되어 있다. 실험에서 실제 모바일 기기를 사용하는 대신에 데스크 탑에 에뮬레이터를 설치하여 사용하였다.



<그림 2> 물리적 네트워크와 JXTA 네트워크간의 대응 관계

그림에서 입방체는 J2SE-JXTA 기반 프로그램을 나타낸다. PDA 이미지는 PalmOS 에뮬레이터를 실행하고 있는 JXME 기반 프로그램을 나타낸다. 모바일 폰 이미지는 J2ME 에뮬레이터를 실행하고 있는 JXME 기반 프로그램을 나타낸다. 물리적인 네트워크 수준에서 보면, 빌딩 A의 방화벽 보호 때문에, 빌딩 B에 있는 컴퓨터와 무선 랜으로 연결된 컴퓨터는 빌딩 A의 S2로 표시된 컴퓨터로 데이터를 전송할 수 없다. 그러나 JXTA 수준에서 이것이 가능하다. 즉, 빌딩 A의 S1로 표시된 컴퓨터가 서로 다른 방화벽 내에 존재하는 컴퓨터들 간의 메시지를 연결해 주는 릴레이 피어 역할을 하기 때문이다.

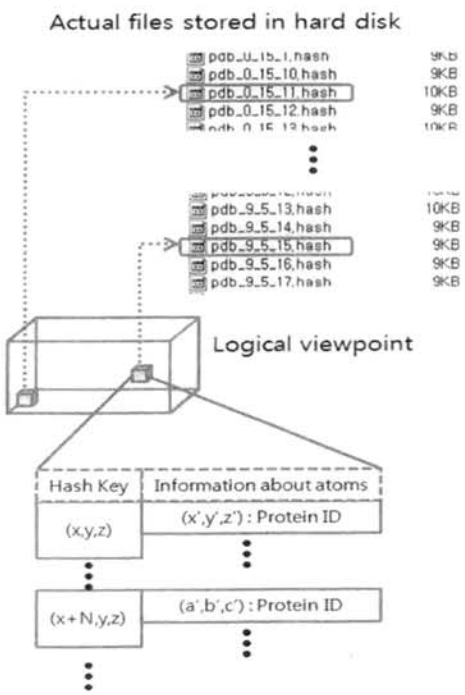
시스템은 N, Cα, C로 표시된 원자들만 추출하고 다른 원자들은 배제한다. 단백질을 일차원구조로 보는 관점에서, 3차원 좌표에서 원자 N의 분포는 단백질들을 구별해주는 특징이 된다. 따라서 단백질을 구성하고 있는 모든 원자를 비교하기보다는 원자 N으로 구성된 이미지를 비교함으로써 단백질 구조 비교에 드는 시간과 노력을 줄일 수 있다. 또한 일차원 구조에서 N, Cα, C는 작은 한 영역에 정기적으로 함께 나타난다. Cα는 아미노산들을 서로 유기적으로 연결시켜주는 지점이 된다. 따라서 이러한 원자를 갖지 않은 단백질들은 본 시스템에서 다루지 않는다. 세 개의 원자로부터 세 개의 벡터로 구성된 레퍼런스 프레임을 구성하는 과정은 다음과 같다.

첫 번째 벡터는 Cα에서 N으로 가는 방향으로부터 얻어진다. 두 번째 벡터는 Cα에서 C로 가는 벡터의 외적을 계산하여 구한다. 세 번째 벡터는 앞서 구한 두 벡터의 외적을 계산하여 얻는다. 이러한 방법으로 3차원 공간에 세 개의 벡터가 직교하게 되고 이를 레퍼런스 프레임으로 사용한다. N으로 표시된 원자들은 새로운 프레임을 기준으로 새로운 위치에 변환된다. 새로운 프레임의 원점은 프레임을 구성하는데 사용된 원자 중 하나인 Cα에 있다.

해시 함수는 좌표 변형을 통해 실수로 구성된 세 튜플로부터 정수로 구성된 세 튜플을 계산해 내고 이를 해시 테이블의 해시 키로 사용한다. 어떤 원자들이 특정 해시 슬롯에서 충돌할 수도 있기 때문에 레퍼런스 프레임에 대한 정보를 배제하는 것이 기하학적으로 부정확한 결과를 생성할

수도 있다. 물론 레퍼런스 프레임을 추가하여 4차원 해시 테이블을 만들 경우, 이러한 문제점은 거의 발생하지 않을 것이다. 그러나 해시 테이블의 차수가 증가할수록 컴퓨팅 자원이 더 많이 요구되므로 이는 일종의 트레이드-오프의 문제가 된다. 따라서 해시 테이블 구조를 어떻게 설계할 것인가는 애플리케이션의 목적에 따라 달라질 수 있다.

<그림 3>은 해시 테이블이 어떻게 하드 디스크에 파일로 저장되는지를 논리적인 관점에서 보여주고 있다. 그림에서 입방체는 해시 테이블의 3차원 공간을 표현한다. 그림 아래의 데이터는 그 입방체의 서브 입방체 조각들에 대한 데이터가 어떻게 저장되어 있는지 표현하고 있다. 하나의 서브 입방체는 여러 해시 슬롯을 갖고 있다. 해시 슬롯의 해시 키는 특정 상수 범위에 있다.



<그림 3> 해시 테이블 생성 예제

예를 들어, 상수가 10으로 설정되어 있다면, 하나의 해시 슬롯은 1000개의 해시 키를 포함할 수 있다. 각 해시 키는 동일한 해시 키를 갖는 원자들을 포함한다. 파일 "pdb_0_15_11.hash"는 인덱스가 (0, 15, 11)인 해시 슬롯 하나에 해시된 원자들의 데이터를 저장하고 있다. 즉, (0, 15, 11)에서 (10, 25, 21)까지의 모든 해시 키는 3차원 좌표와 단백질 ID로 구성된 원자의 정보에 대하여 해시된다. 해시 테이블을 구성해 놓으면, 해시 키의 범위 내에 있는 파일을 로딩시킴으로써 특정 해시 키로 인덱스된 원자들의 정보를 얻는 것이 가능하다.

인식 단계에서 작업 데이터는 작업 프로그램으로 전송된다. 작업 데이터는 후보 해시 테이블 파일과 쿼리 해시 테이블 파일로 구성되어 있다. 작업 프로그램은 후보 해시 테이블 파일의 해시 키 중에서 쿼리 해시 테이블과 같은 해시 키에만 투표한다. 어떤 원자가 득표할 수 있는가를 결정하는 다양한 방식이 존재할 수 있다. 본 시스템에서 적용한 방식은 가장 간단한 것으로 쿼리 해시 테이블에 존재하는 해시 키와 같은 해시 키를 갖는 모든 원자에 투표하는 방식이다. 그러나 좀 더 정확한 비교를 위해서 해시 키가 일치하는 원자들 중에서 각 원자의 좌표값을 비교하여 특정 거리 내에 존재하는 원자들에게만 투표하는 방식이 사용될 수도 있다. 각 원자가 얻은 득표수에 따라 정렬된 결과는 후보 단백질 중에서 어떤 단백질이 쿼리 단백질과 유사한지를 알려준다.

<그림 4>는 실험을 통해 수행된 투표 결과를 보여준다. 본 실험에서, 11개의 후보 단백질이 사전 처리되었고 1A0B가 쿼리 단백질로 선택되었다. 첫 번째 열은 사전 처리된 후보 단백질의 ID를 나타낸다. 두 번째부터 네 번째 열까지는 각 기기의 작업 프로그램이 인식 단계에서 각 단백질 ID에 투표한 결과를 나타낸다. 마지막 열은 서버 프로그램이 작업 프로그램으로부터 투표된 후보 해시 테이블 파일을 받은 후, 각 단백질의 득표수의 총합을 계산한 결과이다. 아래에서 두 번째 행은 각 작업 프로그램이 수행한 작업의 개수를 나타낸다.

이와 동일한 방식으로 여러 개의 PDBML 단백질 데이터를 사전 처리하여 해시 테이블을 생성한 후, 서로 다른 쿼리 단백질을 입력하였을 때, 시스템 원하는 단백질을 정확하게 인식해내는지를 확인하는 실험을 수행하였다. 우선 Protein Data Bank로부터 단백질 15개를 선택하여 각각의 PDBML 파일을 다운로드하고, 이들을 사전 처리하여 후보 단백질 해시 테이블을 생성하였다. 각

단백질의 원자 개수는 평균 489.27개이다.

	C3	C1	C2	Total
1A04	236	2106	2295	4637
1A07	59	601	623	1283
1A08	75	537	585	1197
1B4B	69	583	565	1217
1B4E	147	1299	1421	2867
1A06	104	992	1021	2117
1A0A	20	179	219	418
1KRJ	144	996	1211	2351
1KRS	14	147	168	329
1KRQ	25	355	378	758
1A0B	**2738**	**24849**	**27593**	**55180**
# of jobs	402	9	10	421
Time(min)	24.299	70.065	51.153	*

<그림 4> 각 기기에서 수행한 투표 결과

<표 2>은 15개의 단백질 중에서 3개를 선택하여 각각 쿼리 단백질로 입력하였을 때의 결과를 보여준다. 단백질 1BX8, 1B2J, 1BPT 등 서로 다른 단백질의 쿼리 해서 테이블을 생성한 후, 이를 후보 단백질 해시 테이블과 비교하였을 때, 투표된 결과이다. 쿼리로 입력한 단백질의 총 득표수가 월등히 높게 나오므로, 본 논문에서 제안한 시스템은 찾으려고 하는 단백질과 일치하는 단백질을 정확하게 찾아냄을 확인할 수 있다.

<표 2> 단백질 실험 결과

Protein ID	# of Atoms	# of Votes Query: 1BX8	# of Votes Query: 1B2J	# of Votes Query: 1BPT
1BX7	449	7	5	14
1B7I	532	9	9	16
1B18	472	4	6	8
1B2J	461	7	5226	11
1BBI	536	6	9	20
1BX8	397	3575	8	7
1B7J	531	6	15	18
1B17	483	3	3	13
1BPT	482	4	9	7701
1BL1	501	5	3	2
1B7K	523	10	7	12
1BBA	582	3	1	7
1B19	467	3	5	7
1B13	464	5	9	13
1BE7	459	2	6	11
Time(min)		45.0	52.4	62.2

5. 결론 및 향후 연구

본 논문에서는 JXTA 네트워크에서 실행되는 단백질 구조 비교 시스템을 제안하였다. 컴퓨터에 대한 전문적인 지식이나 기능을 갖고 있지 않은 일반 컴퓨터 사용자들도 JXTA 환경을 설정하여 단백질 구조 비교 실험을 쉽게 수행할 수 있을 것이다. 특히, 임의의 네트워크 환경에서 릴레이 피어를 알맞게 설정함으로써 방화벽에 상관없이 시스템을 설치하고 운영하는 것이 가능하므로, 많은 양의 데이터를 분산 처리할 수 있다.

본 논문에서 제안한 시스템은 고가의 상용 소프트웨어 구입이 어렵고 시스템에 대한 전문적인 지식을 갖추지 못한 일반 사용자들, 특히 대학교에서 생물학이나 화학을 전공하는 학생들이 학교나 기관의 유휴 컴퓨터 자원들을 활용하여 단백질 구조 비교 실험을 실시하는데 유용하게 활용될 수 있으리라 기대된다.

향후, 모바일 폰과 PDA와 같은 실제 모바일 기기를 이용하여 프로그램을 실행하여 단백질 구조 비교 실험을 수행하고자 한다. 예를 들어, UMPC (Ultra-Mobile PC)는 보다 향상된 사용자 인터페이스와 PDA보다 높은 컴퓨팅 능력을 제공한다. 또한 컴퓨터 화면에 단백질의 일부분을 선택하여 시각적으로 보여주는 펜을 이용할 수 있으므로 UMPC을 사용할 수 있는 시스템으로 설계한다면 보다 향상된 기능을 수행하는 시스템 개발이 가능할 것이다.

참 고 문 헌

[1] Traversat, B., Arora, A., Abdelaziz, M., Duigou, M., Haywood, C., Hugly, J. C., Pouyoul, E., & Yeager, B. (2006). Project JXTA 2.0 : Super-Peer Virtual Network. Sun Microsystems, Inc.

[2] Gong, L. (2001). JXTA : A Network Programming Environment. *IEEE Internet Computing*, Industry Report. 88-95.

[3] Ferrari, C., Guerra, C., & Zanotti, G. (2003). A grid-aware approach to protein structure comparison. *The Journal of Parallel and Distributed Computing, 63,* 728-737.

[4] Lamdan, Y., & Wolfson, H. J. (1988).

Geometric hashing: a general and efficient model-based recognition scheme. *Proceedings of the Second International Conference on Computer Vision.*

[5] Wolfson, H. J. (1997). Geometric Hashing : An Overview, *IEEE Computational Science & Engineering*, 10-21.

[6] Pennec, X., & Ayache, N. (1994). An O(n2) Algorithm for 3D Substructure Matching of Protein. *Proceedings of the 1st International Workshop on Shape and Pattern Matching in Computational Biology*, 25-40.

[7] Pennec, X., & Ayache, N. (1998). A geometric algorithm to find small but highly similar 3D substructures in proteins. *Bioinformatics, 14*(6).

[8] RCSB Protein Data Bank (2008). [Online] available: http://pdbml.rcsb.org.

[9] Buyya, R, Branson, K., Giddy J., & Abramson, D. (2003). The Virtual Laboratory: a toolset to enable distributed molecular modelling for drug design on the World-Wide Grid, *Concurrency and Computation Pracice and Experience, 15,* 1-25.

[10] Holohan, A., & Garg, A. (2005). Collaboration online: The example of Distributed Computing. *Journal of Computer-Mediated Communication, 10*(4).

[11] Pelta, D. A, González, J. R., & Vega, M. M. (2008). A simple and fast heuristic for protein structure comparison, *BMC Bioinformatics, 9.*

[12] Foster, I. (2002). What is the Grid? A Three Point Checklist, *GRIDtoday, 1*(6).

[13] Mikkone, H.(2005). Enabling Computational Grids Using JXTA-technology, Helsinki University of Technology, Seminar on Internetworking.

정 효 숙

1998 서울교육대학교 초등교육과 (교육학학사)
2001 서울교육대학교 교육대학원 컴퓨터교육학과(교육학석사)
2003~현재 고려대학교 컴퓨터교육학과 박사과정
관심분야: 컴퓨터교육, 시맨틱 웹, 소셜 네크워킹
E-Mail: est0718@comedu.korea.

안 진 현

2005 고려대학교 사범대학 컴퓨터교육과 (이학사)
2007 고려대학교 사범대학 컴퓨터교육학과 (이학석사)
2008~현재 KAIST 전산학과 박사과정
관심분야: 웹 사이언스, 소셜 컴퓨팅, 시맨틱 웹
E-Mail: jhahncs@gmail.com

박 성 빈

1990 고려대학교 전산과학과 (이학사)
1993 University of Southern California (전산학 석사)
1999 University of Southern California (전산학 박사)
2006~현재 고려대학교 컴퓨터교육과 부교수
관심분야: 하이퍼텍스트, 컴퓨터과학교육 알고리즘, 계산이론
E-Mail: psb@comedu.korea.ac.kr