Bioinformatics and Biosystems 2007, Vol. 2, No. 2, pp. 37-45

바이오그리드 컴퓨팅과 생명과학 연구에의 활용

(Bio Grid Computing and Biosciences Research Application)

김 태 호[1], 김 의 용[1], 염 재 범[1], 고 원 규[2], 곽 회 철[3], 주 현[1*]

Taeho Kim[1], Euiyong Kim[1], Jae Boum Youm[1], Weon-Gyu Kho[2], Heui Chul Gwak[3], and Hyun Joo[1*]

[1]인제대학교 의과대학 생리학교실, [2]인제대학교 의과대학 기생충학교실, [3]인제대학교 의과대학 정형외과

[1]Department of Physiology and Biophysics, [2]Department of Parasitology, [3]Department of Orthopedic Surgery, College of Medicine, Inje University, Busan, Korea

초 록

생물정보학은 컴퓨터를 이용하여 방대한 양의 생물학적 데이터를 처리하고 그 결과를 분석하는 학문으로서 IT의 고속성장과 맞물려 점차 그 활용도를 넓혀가고 있다. 특히 의학, 생명과학 연구에 사용되는 데이터는 그 종류도 다양하고 크기가 매우 큰 것이 일반적인데, 이의 처리를 위해서는 고속 네트워크가 바탕이 된 그리드-컴퓨팅(Grid-Computing) 기술 접목이 필연적이다. 고속 네트워크 기술의 발전은 슈퍼컴퓨터를 대체해 컴퓨터 풀 내에 분산된 시스템들을 하나로 묶을 수 있는 그리드-컴퓨팅 분야를 선도하고 있다. 최근 생물정보학 분야에서도 이처럼 발전된 고성능 분산 컴퓨팅 기술을 이용하여 데이터의 신속한 처리와 관리의 효율성을 증대시키고 있는 추세이다. 그리드-컴퓨팅 기술은 크게 데이터 가공을 위한 응용 프로그램 개발과 데이터 관리를 위한 데이터베이스 구축으로 구분 지을 수 있다. 전자에 해당하는 생물정보 연구용 프로그램들은 mpiBLAST, ClustalW-MPI와 같은 MSA서열정렬 프로그램들을 꼽을 수 있으며, BioSimGrid, Taverna와 같은 프로젝트는 그리드-데이터베이스 (Grid-Database)기술을 바탕으로 개발되었다. 본 고에서는 미지의 생명현상을 탐구하고 연구하기 위하여 현재까지 개발된 그리드-컴퓨팅 환경과 의생명과학 연구를 위한 응용 프로그램들, 그리고 그리드-데이터베이스 기술 등을 소개한다.

키워드: 그리드-컴퓨팅 기술, 생물정보학, 의생명과학, 생물학적 데이터 탐색, 서열정렬, 단백질 구조 및 기능, 바이오툴킷, 통합 생물정보 데이터베이스

Keyword: Grid-Computing Technology, Bioinformatics, Biomedical Science, Biological Data Mining, Sequence alignment, Protein Structure and Function, Biotool Kit, Integrated Bioinformatics Database

왜 바이오그리드(BioGrid) 시스템인가?

얼마 전 수학에서 난제로 뽑히던 소수 관련 문제의 해결을 그리드-컴퓨팅 기술을 이용해 시도하는 뉴스를 접한 적이 있다. 전 세계에 인터넷으로 연결되어 있는 무수히 많은 컴퓨터 자원을 이용하여 계산시간을 단축한다는 아이디어다. 생물정보학에서도 이와 같이 진보된 IT기술을 이용하여 소규모 실험실 규모에서 연구소 단위까지 방대한 양의 생물학적 데이터를 다루고 분석하는 시도가 많이 이루어지고 있다. 문제는 생명과학 분야에서 다루는 생물학적 데이터는 크기가 매우 커서 입출력(input/output)뿐만 아니라, 정보검색 및 가공시 연산시간이 상당히 오래 걸린다는 점이다. 특히 게놈프로젝트(genome project)를 전후로, 지난 20여 년간 GenBank, EMBL, DDBJ 등에 축적된 DNA와 RNA 서열의 양은 최근들어 기하급수적으로 늘어나 현재는 165000개에 이르는 개별유기체(organism) 수에 그 염기총량(total base) 정보량만 100기가 베이스($1x10^{11}$ base)에 이르렀다. 편의상 염기하나를 1 byte단위로 간주한다면, 이들 전체 염기 총량을 현재의 저장매체에 담는 것은 그리 어렵지가 않다. 너무도 손쉬운 일이지만, 이들 저장매체에 사용자가 필요로 하는 특정 정보를 탐색하고 일련의 연산과정을 거쳐 고급수준의 정보로 재 가공하기 위해서는 상상

교신저자 : 주현 (Email: phyjoo@inje.ac.kr)

본 논문은 산업자원부 해양생물산업 지역혁신체계구축 지역역량강화사업 및 인제대학교 의과대학 바이오클러스터 구축 사업(2006~2007)의 지원으로 수행되었음.

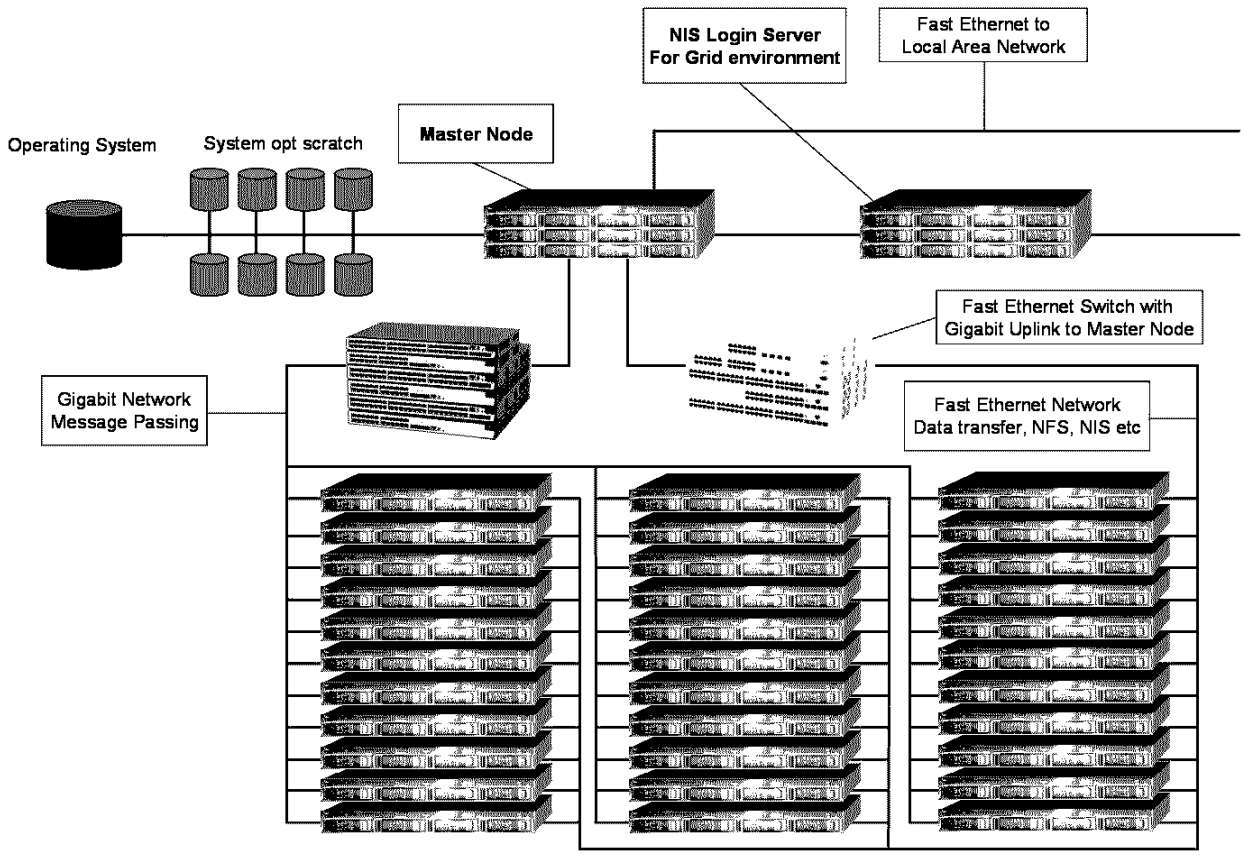

Fig. 1. 그리드-컴퓨팅(Grid-Computing) 시스템은 마스터 노드, 계산 노드, 네트워크 기기로 구성되며 때로는 보안을 위해 NIS서버와 같은 계정관리 서버를 따로 두는 경우도 있다. 클러스터 시스템의 토폴로지는 보통 두 개의 네트워크로 구성된다. 하나는 데이터의 송수신에 사용되며, 초당 기가비트(gigabit)단위의 데이터를 전송할 수 있는 빠른 네트워크를 사용한다. 또 다른 네트워크는 시스템 관리에 사용되는 데이터의 송수신에 관여하며, Fast Ethernet과 같은 비교적 느린 네트워크로 구성될 수 있다. 마스터 노드는 전체 컴퓨팅 시스템의 운영 및 관리를 담당하며, 이를 위해 NIS, NFS등의 통신 프로토콜을 사용한다. 계산노드는 실제 데이터의 연산기능을 담당하게 되며, 마스터 노드에서 할당 받은 데이터를 가공하여 다시 마스터 노드로 돌려보내는 역할을 수행한다.

할 수도 없이 많은 양의 메모리와 연산속도를 요구한다. 생물정보학에서 일컫는 '메모리 법칙'은 다음과 같다.

$$\text{Sequence, } n = \{m, \text{byte}\}^n$$

즉, m개의 염기수를 지닌 n개의 개별서열들이 존재할 때, 이들을 모두 일대일 비교하기 위하여 연산에 요구되는 메모리 총량은 m 바이트(byte)의 n 승을 필요로 한다. 동적 프로그래밍(dynamic programming, DP) 과정까지 고려한다면 더 많은 메모리가 소요된다. 가까운 미래에 연구자가 실제 접할 수 있는 염기총량은 가히 상상을 초월하는 수를 넘어설 것이다. 일반적으로 세포 하나에 약 3만개의 단백질이 존재함을 감안할 때, 16만종의 개별유기체 x 30,000개의 단백질 x 1500개의 염기 수(평균적으로 30 kD 크기의 단백질 발현을 고려할 때)를 계산하면, 무려 7조2천억 개의 염기총량이 새로운 정보단위로 축적될 것이다. 이와 같이 대규모 메모리용량과 이에 따른 고속 연산기능을 뒷바침하기 위해서는 컴퓨팅 기술도 병행하여 발전되어야 한다.

그리드-컴퓨팅은 소위 '분산 컴퓨팅(distributed computing)' 기술의 한 분야이다. 이는 빠른 로컬 네트워크와 인터넷을 통해 소규모 연구단위체 혹은 세계 각지에 흩어져 분포하는 수천에서 수십만 대의 컴퓨터 자원과 데이터 저장 능력을 하나로 공유하는 기술로 정의된다. 특히, 지난 10여 년간 이루어진 High-Speed 네트워크(Network) 기술의 발전은 슈퍼컴퓨터를 대체해 범용 컴퓨터급에서도 슈퍼컴의 연산과 데이터

가공이 가능케 하는 새로운 컴퓨팅 기술을 이끌게 되었다. 실제로 데이터의 공유와 빠른 계산을 필요로 하는 생물정보학분야에서 고성능 컴퓨팅 기술의 발전은 매우 중요한 부분을 차지한다. 이렇게 발전된 그리드-컴퓨팅 기술은 생물의학 및 생명공학 어플리케이션, *in silico* 약물 탐색 및 기능 분석, 단백질 분석, 독성 구조 확인 및 관련 데이터베이스 구축, 생물장치 공정설계 최적화, 자동화 어플리케이션 분야에 이미 널리 사용되고 있다. 예를 들면, 현재 프로테오믹스나 시스템즈생물학 연구용으로 구축된 리눅스 클러스터(Linux Cluster)도 고성능 컴퓨팅 기술의 한 부분으로 여러 대의 컴퓨터를 하이스피드 네트워크로 묶어 하나의 작업을 여러 대의 컴퓨터가 동시에 처리할 수 있도록 고안된 것이다.

실제 생물정보학 분야에 이용되는 그리드-컴퓨팅 기술은 크게 두 가지로 나눌 수 있다. 계산 및 데이터 가공을 위한 데이터 처리형과 대형 스토리지 서버 등의 운용을 위한 데이터 저장형이다. 연산 및 데이터 처리를 위해 이용되는 그리드-컴퓨팅 기술은 복잡한 단위연산을 연산 노드별로 분할하여 계산량을 고르게 분배하는 시스템으로서, 이들 연산 노드별 결과의 임시저장 및 입출력을 위하여 여러 대의 컴퓨터를 빠른 네트워크로 연결하여 한대의 컴퓨터처럼 운용하는 고성능 컴퓨팅 기술이다 (Fig. 1). 데이터 저장을 위해 이용되는 그리드-컴퓨팅 기술은 데이터의 효율적인 관리를 위해 데이터를 임의로 여러 노드에 분산 저장하거나 반대로 여러 곳에 분산되어 있는 데이터를 집약시키는 기술을 뜻한다. 후자의 경우, 데이터 전송의 안정성 및 보안 문제가 통신 프로토콜과 직결되어 있다. 본 고에서는 실제 그리드-컴퓨팅 기술이 생물정보학에서 과연 어떻게 적용되고 있으며 어떤 어플리케이션들을 제공하고 있는지에 대하여 자세히 살펴보기로 한다.

고속 연산 기능을 위한 바이오그리드 시스템

생물정보학에서 때로는 수천~ 수만 개 이상에 이르는 방대한 데이터들과 복잡한 수식들을 연결시켜 어떠한 결과를 이끌어내야 하는 경우가 많다. 대표적인 예가 생물종 사이 혹은 게놈(genome)들간의 서열 유사성 분석을 위한 다중서열정렬(Multiple Sequence Alignment의 약자로서, 흔히 MSA로 줄여 부름)이다. 이러한 경우 항상 염두에 두어야 하는 문제는 최소 비용으로 빠른 시간 내에 원하는 정보를 얻을 수 있는가에 대한 질문이다. 아키텍쳐의 구성상 하나의 거대 메모리를 공유하는 다중코어 집합체로 구성된 슈퍼컴퓨터는 매우 빠른 처리속도를 제공하지만, 경제성에 있어서는 좋은 해결책이 되지 못한다. 실제 일반 유저의 접근이 매우 제한적이기 때문이다. 따라서, 이러한 문제를 해결하기 위해 고안된 것이 리눅스클러스터(Linux Cluster)로 대표되는 고성능 컴퓨팅 기술이다. 이 시스템은 수대에서 수십 대의 비교적 저렴한 컴퓨터들을 기가비트급 이상의 빠른 네트워크를 이용하여

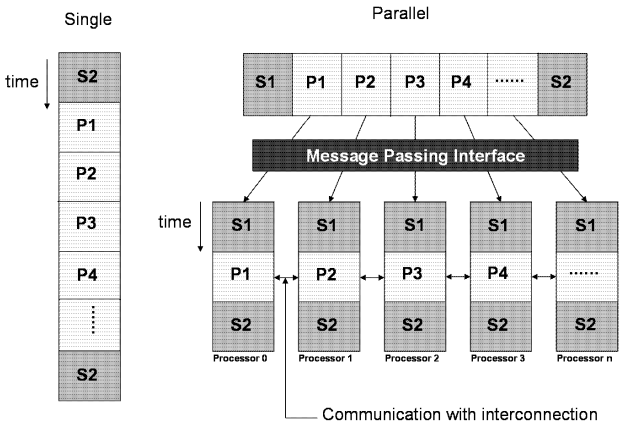

Fig. 2. MPI환경에서의 프로세스처리. 싱글 프로세서 시스템에서는 프로세스 (P) 처리가 시간에 따라 순차적으로 진행된다. 반면, 병렬 프로세서 시스템에서는 마스터 노드에서 각 계산노드의 유휴 상태를 검사 후 작업량을 노드수에 맞추어 분할한다. 각 노드에 분산된 프로세스는 처리가 완료된 후 다시 마스터 노드로 보내진다. 따라서 전체 프로세스 (P1+P2+P3…)의 처리시간은 계산노드의 수에 반비례하고, 네트워크의 오버헤드에 비례해서 증가하게 된다.

연결하는 시스템으로, 처리속도와 경제성을 모두 고려하여 제작된다. 그리드-컴퓨팅기법은 현재 생물정보학, 유체역학, 생물물리학 등의 분야에서 매우 흔하게 사용되고 있다. 특히 이들 고성능클러스터 시스템들은 각 노드별 전송언어로서 MPI(Message Passing Interface)나 PVM(Parallel Virtual Machine)같은 병렬처리기술을 기반으로 모든 컴퓨터 (노드)에서 동시에 데이터를 처리하는 특징을 가진다. (Fig. 2). 특히 개별 노드별로 생성된 각 작업을 여러 대의 컴퓨터가 동시에 분산 처리하여 그 처리속도를 획기적으로 빠르게 만드는데 목적이 있다. 주로 공개 운영체제인 리눅스를 이용하여 구축되기 때문에 리눅스 클러스터로 널리 알려져 있다.

고성능 컴퓨팅 기술 현황

매6개월마다 "Top500 사이트"(http://www.top500.org) 는 전 세계에 구축되어 운영되는 고성능 컴퓨터의 성능을 비교하여 상위 500위까지 리스트를 작성하고 있다. 작년까지 보고된 세계에서 가장 빠른 컴퓨터는 IBM이 제작하여 미국의 로렌스 리버모어 국립 연구소에서 운용하고 있는 컴퓨터로 측정된 성능은 280600 기가플롭스(GFlops), 즉 초당 약 2.8 x 1014번의 부동 소수점 연산명령을 실행할 수 있는 성능을 가졌다. 이는 일반 데스크톱용으로 많이 사용되는 펜티엄6 컴퓨터 78000대를 동시에 사용하여 내는 성능치와 맞먹는 엄청난 시스템이다. Top500 리스트에 포함된 대부분의 고성능

컴퓨터는 슈퍼 컴퓨터급이지만 비교적 저가로 판매되고 있는 인텔 또는 AMD 64비트 프로세서 기반으로 구현되어 연산능력과 경제성을 동시에 해결한 리눅스 클러스터와 같은 고성능 컴퓨터도 지난 3년간 약 40대에서 200대 이상으로 5배 이상 증가하였다. 이러한 증가세는 계속 될 전망이다. 이것은 일반 데스크톱에서 사용되고 있는 프로세서의 성능이 증가한 것도 한 원인이 되고 있다. 또한 현재 Top500에 등재된 한국의 고성능 컴퓨터는 총 6대로 캐나다, 스페인에 이어세계 11위에 랭크 되어있다. 그러나 불운하게도 생물정보학 연구를 위하여 운용되고 있는 고성능 컴퓨터는 아직 한대도 TOP500에 링크되어 있지 못한 실정이다.

그리드-컴퓨팅에서 사용되는 가장 일반화된 병렬처리 프로그램들은 MPI, OpenMP, 혹은 PVM들을 들 수 있다. 이들 언어들은 병렬화 프로그램 라이브러리를 사용하여, 각 노드간의 원활하고 효율적인 통신을 위해 구현 되었다. 이들 라이브러리들(MPI, OpenMP, PVM)사이의 차이점은 여러 가지를 들 수 있겠지만, 크게 컴퓨터 노드간의 처리형태에 따라 구분되어 진다. 먼저, 병렬화된 각 노드를 구성하는 프로세서의 타입이 서로 같은 동종 시스템인지 혹은 서로 다른 이종간 시스템인지에 따라 구별되며, 동종 시스템에는 MPI를, 이종간의 시스템에서는 PVM을 사용한다. 그리고 프로세스 단위로 처리하고자 하면 MPI를, 스레드(thread) 단위로 처리를 하고자 하면 OpenMP를 주로 이용한다. 하지만, 생물정보학용으로 구현되고 있는 대다수의 클러스터 시스템은 인텔 혹은 AMD CPU기반의 동종시스템에서 프로세스 단위로 처리하는 MPI(Message Passing Interface) 라이브러리가 주로 사용된다.

하나의 작업을 다수의 프로세스를 이용해 분산 처리하는 병렬환경에서 각 프로세스간의 상호통신을 위해 MPI가 필요하다. 하나의 프로세스내에 존재하는 여러 스레드 (thread)들을 분산처리하기 위하여 개발된 OpenMP와는 달리 MPI는 "프로세스"를 기반으로 노드별 작업량 할당과 분산처리가 동시에 가능하여 이종간의 높은 이식성을 특징으로 한다. MPI 라이브러리를 이용한 병렬환경의 프로그램 개발은 C-언어 혹은 Fortran에서 {함수} 혹은 {서브루틴} 형태로 호출되어 사용된다. Fig. 2는 단일 시스템에서의 프로세스 처리방식과 MPI를 이용한 병렬환경에서의 프로세스 처리간의 차이점을 보여준다. 단일 시스템에서 프로세스 처리는 프로세스가 생성된 순서에 따라 순차적으로 처리를 진행하지만, MPI를 이용한 병렬환경에서의 프로세스 처리는 생성된 프로세스를 마스터노드에서 각 종속노드로 분산시켜 동시에 처리하게 된다. 이때, 모든 노드의 상태를 관리하는 마스터노드의 역할은 커지게 된다. 마스터노드는 각 노드들이 현재 작업을 하고 있는지의 여부를 파악해 프로세스를 생성하게 되고, 처리된 프로세스를 리턴 받아 작업 재할당 여부 및 노드별로 전송된 결과들의 저장 혹은 종합적인 처리를 수행한다. 따라서 병렬환경에 포함되는 노드 수에 비례해서 주어진 작업 처리속도는 증가하게 된다.

서열정렬 및 탐색에의 이용

서열간 유사성(혹은 호몰로지) 검색 및 글로벌 서열정렬을 위해 생명과학 연구에 빈번히 사용되는 BLAST와 같은 생물정보 탐색및 분석용 프로그램도 그리드-컴퓨팅 환경에서 사용이 가능하도록 BeoBlast, Soap-HT-BLAST, mpiBLAST, GridBLAST, W.ND BLAST, Squid와 같이 다양한 병렬프로그램으로 개발되어 있다. 이 프로그램들의 특징은 수천개 이상의 데이터를 빠른 시간에 처리할 수 있도록 제작되어 있다. 또한 최근 개발된 ABCGrid(Sun et al., 2007) 는 NCBI_BLAST, Hmmpfam, CE등의 프로그램을 통합하여 UNIX/Linux, Windows, Mac OS X 등 다양한 운영체제 환경의 바이오 클러스터 시스템에서 사용할 수 있도록 개발되었다. ABCGrid는 ABCUser, ABCMa- ster, ABCWorker로 구성된다. 사용자가 ABCUser를 이용하여 작업을 생성하면 ABCUser는 생성된 작업을 ABCMaster로 보내 작업수행에 필요한 분석을 거쳐 최종 노드별로 업무량 분할작업을 수행한다. 이때 ABCMaster는 분할된 작업들을 각 노드의 ABCWorker에게 보낸다. 이 과정에서 ABCGrid는 생성된 작업을 현재 사용 가능한 노드를 자동으로 찾아 보내는 알고리즘을 적용하였다. 이것이 병렬 응용프로그램에서 중요하게 생각하는 요소인 로드밸런싱(load balancing)이다. 각 노드의 ABCWorker를통해 처리된 최종 결과는 다시 ABCMaster에 보내져 결과를 종합하여 출력하게 된다. 이 알고리즘의 구현을 통해 ABCGrid는 AMD Sempron 2200+ CPU 30개로 이루어진 리눅스 클러스터에서 1000개의 핵산(nucleic acid) 서열정보가 들어있는 파일을 처리한 결과 약 1.8배 처리속도를 증가시켰다. 이는 결코 작은 처리속도의 향상이 아니다. 예로 6시간 걸려 처리될 작업량이 불과 3시간으로 줄 수 있다는 의미를 지니고 있다.

국부적(local) 다중서열정렬(MSA)을 위하여 초기 싱글처리 버전으로 개발되었던 ClustalW는 Li에 의해 MPI 라이브러리가 사용된 병렬화 프로그램으로 다시 완성되었다. ClustalW-MPI(Li, 2003)라 명명된 이 병렬화 국부정렬 프로그램은 Posix threads기반의 '공유메모리' 알고리즘을 기반으로 개발되었던 parallel ClustalW(Mikhailov et al., 2001)와는 다르게 '분산메모리' 아키텍처를 기본으로 하는 리눅스 클러스터 환경에 최적화 시킬 수 있는 기반을 제공하였다. ClustalW 알고리즘은 주어진 서열정렬을 수행하기 위하여 모두 세 단계를 거친다. 첫 단계는 쌍정렬(pairalign)을 통하여 각 서열간 거리 매트릭스(distant matrix)를 계산한다. 두 번째 단계는 서열간 토폴로지를 결정한다. 마지막으로 순차적인 다중 서열정렬을 실행하는 방식으로 이루어진다. 이와 같은 방식의 알고리즘을 '프로그레시브' 다중서열정렬법이라 부른다. ClustalW-MPI는 첫 번째 단계인 쌍정렬의 거리매트릭스 계산을 위하여 일정한 사이즈의 데이터를 각 프로세스에 분산시키는 고정-크기 청킹(fixed-size chunking) 스케쥴링(Hagerup, 1997)을 사용하였다. 이 알고리즘 구현을 통하여

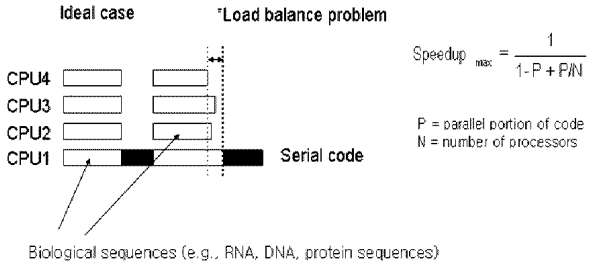

Fig.3. 바이오-그리드 시스템에서의 로드 밸런싱 문제. 가장 이상적인 형태는 동일 크기의 서열정보들이 제공되어야 하나, 실제 생물학적 서열들은 이러한 균질 크기를 지니기 어렵다. 이에 Li등은 고정-크기의 파일을 각 노드상에 분산시키는 기법을 적용시켜 CPU상의 오버헤드 시간을 단축시킬 수 있었다. 병렬화 그리드-컴퓨팅을 통한 스피드업(speed up)은 크게 코드의 병렬화 정도(0~1), 분산된 프로세스 수의 함수로서 표현이 가능하며, 이를 Amdahl의 법칙이라 부른다(Amdahl, 1967). 이 수식에는 로드밸런싱은 고려되지 않은 상태이다. 완전히 병렬화 코딩이 이루어진 경우, 속도향상은 단지 프로세스의 수에 의하여 결정되나, 실제 이와같은 이상적인 시스템은 구현이 어렵다.

네트워크상의 오버헤드를 일정하게 유지하고 각 노드별 연산기의 유휴시간을 줄이는 것으로 ClustalW-MPI는 주어진 서열의 정렬 속도를 펜티엄III CPU 16개로 구성된 리눅스 클러스터에서 단일 시스템에 비하여 약 14.5배 증가시켰다 (Fig. 3 참조).

MrBayes 3(http://mrbayes.csit.fsu.edu/)는 진화적 거리(evolutionary distance)를 계산하거나 생물정보학에서 중요한 phylogenetic tree를 생성하는데 필요한 응용프로그램으로서 리눅스 혹은 매킨토시 환경의 클러스터 시스템에도 구동이 가능하다.

단백질 구조와 기능예측에의 응용

단백질의 구조와 기능연구에는 '분자동력학적(Molecular dynamics)' 혹은 '브라우니안 동력학적(Brownian dynamics)' 분자운동 시뮬레이션 전산기법을 이용한다. 분자동력학적 시뮬레이션은 대상 모델을 구성하는 각 분자간의 물리학적 상호작용력을 뉴톤의 운동방정식으로 수식화하여 각 분자를 구성하는 원자들의 운동궤적을 수치적으로 생성한다. 정해진 시간별로 기록된 궤적값들을 하나의 파일로 모으면 각 분자들의 움직임과 상호작용을 알아볼 수 있다. 이에 반하여, 브라우니안 동력학은 주로 액체상의 거동 모사에 많이 사용된다. 이러한 시뮬레이션 기법들은 분자내 모든 원자들의 위치를 시간의 함수로 계산해야 하기에 매우 많은 양의 계산이 필요하고 상당한 시간이 요구된다. 이를 해결하기 위한 방법 중의 하나로 그리드-컴퓨팅 기법이 분자운동 시뮬레이션에 도입되었다. Gromacs(Van Der Spoel et al., 2005), NAMD, Charmm, AMBER등은 병렬화된 그리드- 컴퓨팅 환경을 제공하여 빠른 시간안에 방대한 양의 분자거동 궤적을 수치화 하여 나타낼 수 있다. 이중 NAMD(Phillips et al., 2005)는 병렬화 연산기능을 가지기 위하여 별도로 Charm++ 병렬프로그래밍 시스템과 런타임 라이브러리를 요구한다. Charm++는 message-driven object 프로그래밍 기법을 이용하여 클러스터 내 데이터 송수신시 발생하는 통신지체 현상을 최소화 하고, 시스템 노이즈 현상도 자연스럽게 해결할 수 있도록 구성되어 있다. 또한 시뮬레이션 시작단계에 각 프로세서가 하나의 작업을 해결하는 시간을 미리 측정하여 프로세서의 유휴시간을 효과적으로 관리하는 실측치 반영 로드밸런싱(measurement-based load balancing) 기법이 이용된다. 이는 처리속도와 프로세서 관리의 효율을 동시에 높일 수 있다. 단백질의 구조와 기능연구 외에도 단백질간 결합반응이나, 다른 소 분자체(small molecule) 등의 구조 및 상호 결합력 예측 연구에도 이러한 그리드-컴퓨팅 기술이 사용된다.

의생명과학 연구 및 과학적 진단 도구로의 활용

인체의 생명현상을 과학적으로 분석하고, 올바른 예측 및 진단이 가능케 하는 메디칼 어플리케이션들의 개발이 전세계적으로 활발히 진행되고 있다. 멀티미디어 전문업체로 알려진 일본의 SONY사는 피지옴(Physiology:생리 + -ome:전체의 합성어로 '생리체 (physiome)' 라고도 부름) 연구를 위하여 전세계적인 핵심 기술인력을 대폭 보강하였을 뿐만 아니라, 이에 대한 플렛트폼 개발에 박차를 가하고 있다. 한마디로 각 기능을 수행하는 인체 기관들의 생체 내 조건 및 외부자극에 대한 반응을 디지털화된 컴퓨터에서 실제와 똑같이 묘사하겠다는 것이다. 이와 밀접한 요소기술들에는 디지털 X-ray와 연동되는 해부학적 영상화 기술, 지능화된 병태생리학적 진단 및 분석기술들로서 항상 바쁜 시간을 보내야 하는 임상의가 신속하고도 정확하게 병의 진단 및 처치, 그리고 수술방법(미세수술까지 포함)까지도 시뮬레이션을 통하여 재현해 볼 수 있다는 점이다. 환자의 생명을 다루는 중요한 외과적 수술과 진단에 단 한번의 실수도 인정하지 않는다는 목표아래, 미래 첨단의료 기술들이 연구 중이다. 현재까지 임상 실습 및 연구용으로 개발된 응용프로그램들도 상당수가 존재한다. 가상심장 모델 구축에서부터 체내 내분비 시스템의 변화들을 신속하게 분석하고, 응급상황이나, 약물 투여 등에 따른 체내 변화의 진단을 위한 지표로의 활용 그리고 이에 대한 적절한 처치 등을 효율적으로 진행할 수 있다. 피지옴 연구의 최전선은 바로, 국제 생리과학 연합기구(International Union of Physiological Sciences)에서 운영하는 웹 기반의 연구정보사이트(http://www.physiome.org) 에서 확인할 수 있다 (Bassingthwaighte, 2005). 이에는 심혈관계 운동, 세포생리, 내분비계, 통합생리, 신경계, 호흡계, 행동 및 체내 물질 이동

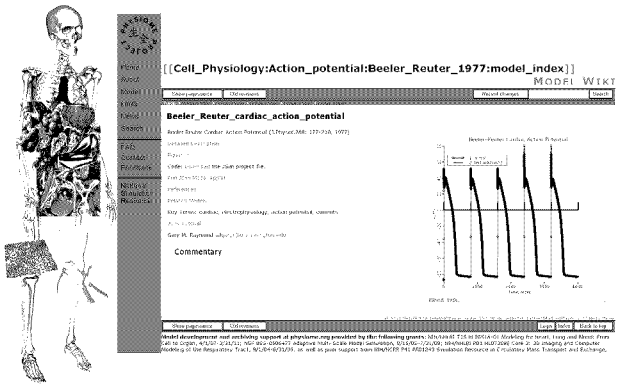

Fig. 4. 디지털 휴먼 구현을 목표로 진행중인 IUPS 피지옴 기구. 작게는 세포신호전달 경로에서, 크게는 인체 각 기관별 시뮬레이션을 통한 가상 인간체를 만들고자 한다. 한 예로, 심근세포에 작동중인 활동전압 모델과 이의 연산결과를 표시하는 수식, 설명 등이 수록되어 있다. 이러한 작은 부분 단위체 모델들은 XML을 공통언어로 기록되며 항상 실시간으로 업데이트되어 많은 연구자와 공유 및 통합이 가능하도록 설정되어 있다.

현상 등으로 세분화되어 개별 카테고리 내에서 각 연구자들은 발표 혹은 미발표된 모델에 대한 주석과 함께 애플릿 프로그램 코딩을 공유하게끔 구성되어 있다. 선 마이크로시스템즈사는 시스템 구현을 위하여 JAVA 애플릿을 제공한다.

피지옴 연구는 전체(wholeness)속에서 모든 인체의 생명현상을 규명하고자 하는 목표를 지니고 있다. 연구자들은 이러한 세포 혹은 각 기관별 생명현상을 하나로 통합할 수 있는 방안을 모색중이다. 대표적인 예로서 'CellML (http://www.cellML.org)' 모델링 기술 통합체 운영을 통하여 각 연구자가 XML 기반의 프로그램 언어들을 이들 데이터베이스에서 검색하고 하나로 모을 수 있는 기회를 제공하는 일도 활발히 진행 중이다. 특히 이들 모델들의 신속한 출력을 위해서는 그리드-컴퓨팅과의 연결이 필수적이다. 복잡한 인체의 생리활동들을 신속하고 정확하게 시뮬레이션하기 위해서는 단일 컴퓨터상에서의 운영은 절대 불가능하다. 이를 위하여 병렬화 처리가 요구되는 바이오그리드 시스템과의 접목이 시도되고 있다. 한 예로, 1960년대에 옥스포드 대학의 데니스 노블(Noble, 1960)에 의하여 구현된 호지킨-헉슬리 모델에 기반을 둔 가상 심장의 초기 컴퓨터 모델은 많은 개선이 진행되어 왔으며, 오늘날 영국 CLRC(http://www.clrc.ac.uk) 주도로 'e-science'그룹에 의하여 그리드-컴퓨팅이 가능한 병렬화 언어로 이미 탈바꿈하였다.

바이오인포마틱스 툴킷에의 응용

생물정보학 분야에 이용되는 리눅스/유닉스 기반의 응용 프로그램은 매우 다양할 뿐만 아니라, 이들 프로그램들은 그 사용목적에 따라 개별적으로 존재하기 때문에 실제 정보의 발굴부터 가공, 그리고 분석을 위해서는 각각의 응용 프로그램들을 선정하여 수 많은 개별 수작업들을 거쳐야 한다. 그러나 세포군 혹은 집합체 단위의 생명정보를 다루는 새로운 omics(전체를 뜻함) 연구의 태동과 함께 최근 이들 개별 프로그램 도구들을 사용자 임의대로 짜맞추어 일괄 처리가 가능케 하는 생명정보학 툴킷들의 개발이 활발히 진행되고 있다. 이도 물론 바이오그리드 기반으로 구축중이다. 즉, 개별적으로 구성된 생물학적 정보들을 하나의 거대한 툴안에 넣어 이들 사이의 연관, 배치 등을 빠른 시간안에 통합적으로 연구하고자 하는 것이다. 실로 세포 내 존재하는 단백질량만 해도 수만종 이상이 존재한다. 그 중 프로테오믹스 연구기법 중의 하나인 다차원 크로마토그래피 혹은 이차원 전기영동(2-D)겔을 거쳐 확인이 가능한 단백질들은 약 1만종 정도로 예상된다. 특히 서열 구조 자체가 *de novo*한 경우는 이의 분석자체가 용이하지가 않다. 이러한 문제점을 해결하기 위하여 적용된 시스템으로 '시퀘스트클러스터' (Sequest Cluster)를 들 수 있다. 이는 스크립스 연구소의 Yates박사에 의하여 고안된 알고리즘으로, Genome 혹은 개별 단백질 서열정보 DB상에 존재하는 수많은 유전자 혹은 단백질 서열들로부터 가상의 질량분석값을 가공하여 생성하고, MS/MS 질량분석기를 통하여 획득된 단편화 이온들의 질량정보와 비교 분석하는 작업에 필수적으로 사용된다(Tabb et al., 2002).

이와 같이 툴킷은 목적에 따라 필요한 프로그램들을 패키지 형태로 통합시켜 여러 단계를 거쳐 처리되어야 할 데이터를 하나의 툴킷으로 일괄 처리할 수 있도록 개발된 것이다. 단백질 서열분석을 위해 독일 막스-프랑크 연구소에서 개발된 생물정보학 툴킷(Bioinformatics Toolkit http://toolkit.tuebingen.mpg.de), 16S rRNA 유전자 서열 분석을 위해 영국 Cardiff 대학에서 개발된 생물정보학 툴킷 (Bioinformatics Toolkit; http://www.cardiff.ac.uk/ biosi/research/biosoft) 등과 같이 공개형 툴킷이 있는 반면, IBT(Informatics Benchmark Toolkit)와 같은 상업용 툴킷도 존재한다. 특히, 막스-프랑크 연구소에서 개발되어 웹 서비스 중인 생물정보학 툴킷은 MPI기반 병렬환경으로 개발되었다. 이 툴킷에는 사용자가 원하는 형태의 다양한 데이터가공 및 처리를 위하여 내부에 NucleotideBLAST, ProteinBLAST, PSI-BLAST, fastHMMER, HHsenser; ClustalW, MUSCLE, Mafft, ProbCons; HHrep, PCOILS, REPPER; Quick2D; HHpred, Modeller; CLANS, ANCESCON, PHYLIP; Reformat, RetrieveSeq, gi2promoter를 포함한다. 이 프로그램들은 상호 유기적으로 연결되어 서열검색, 다중서열정렬, 단백질 이차 및 삼차 구조예측과 분류를 한번에 처리할 수 있는 환경을 제공한다. 툴킷에 포함되어 있는 각종 BLAST 프로그램들로부터 검색된 서열들은 바로 ClustalW, MUSCLE과 같은 다중서열정렬 프로그램과 연동되어 로컬 정렬(local alignment)을 수행한다. 정렬 결과는 곧 바로 HHpred, Modeller, Reformat

들과 같은 프로그램이 이용되어 단백질 이차 · 삼차 구조의 예측 및 분류를 진행한다. 이 일련의 과정은 단 한번의 작업으로 모두 실행 가능하며, 처리속도의 향상을 위하여 각각의 프로그램은 모두 바이오그리드 시스템에서 사용이 가능하게 병렬 코드화되어 있다. Fig. 5는 미지의 단백질 서열을 대상으로 데이터베이스 검색, 서열정렬, 단백질 이차구조 예측까지 요구되는 일련의 작업 과정을 툴킷을 이용하여 실행한 결과이다(Biegert et al., 2006).

IBT는 cross-platform 기반의 리눅스/유닉스 응용프로그램 벤치마킹 프레임워크로서 사용되고 있는 각종 생물정보학 프로그램들이 특정 시스템에서 어떠한 성능을 나타내는지 실시간으로 자체 진단할 수 있는 환경을 제공한다. Table 1은 IBT 홈페이지에서 제공하는 벤치마킹 자료로 IBT를 이용하여 BLAST(Altschul et al., 1990), BLAT (Kent, 2002), gromacs, hmmer(Eddy, 1998)를 임의 시스템 A, B, C에서 자체 진단한 결과를 나타낸다. 사용자는 이러한 통계자료를 기반으로 시스템 튜닝 작업을 용이하게 진행할 수 있으며, 특히 특정 프로그램에 대한 리소스 제한 및 확장을 통하여 최적의 환경을 구현할 수 있다.

통합 데이터 관리를 위한 바이오그리드 시스템

생물정보학에서 중요한 포인트는 많은 양의 데이터를 빠른 시간 내에 처리하는 것 이외에도 방대한 양의 데이터들을 효율적으로 관리할 수 있는 시스템 구축에 대한 문제들이다. 그 예로 많은 생물학 데이터의 체계적인 분류와 관리를 위해 만든 컨소시엄인 유전자 온톨로지 컨소시엄(Gene Ontology Consortium http://www.geneontology.org) 이 있다. 또한 데이터의 효율적 관리를 위하여 GenBank, DDBJ, EMBL은 주기적으로 서로의 데이터를 동기화시켜 상호교환 및 관리하고 있다.

최근에는 그리드 환경하에서 데이터의 효율적인 관리와 처리를 동시에 수행할 수 있는 미들웨어(middleware)들이 개발되고 있다. 그 예로 Taverna(Oinn, T., et al, 2004), Pegasys (Shah, S.P., et al, 2004), Wildfire (Tang, F., et al, 2005)등을 꼽을 수 있다. 특히, Taverna는 워크프로우 언어(workflow language)를 지원하고, GUI를 이용하여 사용자로 하여금 데이터관리와 데이터 처리결과의 가독성을 높였다. 또한 웹서비스를 통하여 사용자의 접근을 용이하도록 돕고 있다. 하지만, 이 예들은 데이터 관리의 효율성에만 중점을 두고 있어, 데이터의 효과적인 가공은 매우 미비하다. 따라서, 미들웨어들은 생물정보학적 워크플로우(bioinformatics workflow)를 적용시켜 서열데이터의 단순 저장이나 저장된 데이터를 불러 서열정열 연산을 수행하는 일에만 한정되지 않고, 사용자 위주의 인터페이스 제공과 서열정열을 기반으로 한 데이터의 2차 가공 지원 등이 동시에 이루어지도록 발전되어야 한다. 특히 국내에서 개발된 BioGridPSE(Sun et al, 2006)는 이러한 결과의 좋은 예이다.

Table 1. IBT를 이용한 벤치마크 결과. 테스트를 위하여 임의 시스템 A, B, C를 적용시켜 blat, blast, gromacs, hmmer의 처리속도를 시간별로 측정하였다. 각 시스템의 사양에 따라 처리속도가 다른 것을 알 수 있으며, 여기서 각 시스템의 사양과 그래프 형태의 결과는 제공하지 않는다.

프로그램 종류	자체진단 시스템	시간(초)
Blat	System A	710
	System B	830
	System C	900
Blast	System A	920
	System B	950
	System C	1045
Gromacs	System A	130
	System B	154
	System C	194
hmmer	System A	1220
	System B	1350
	System C	1400

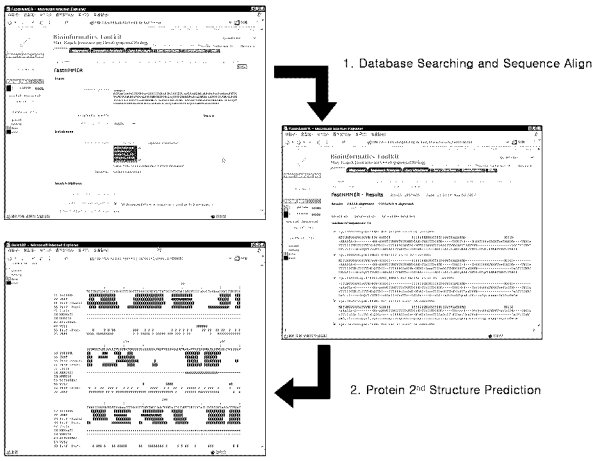

Fig. 5. 툴킷을 이용한 단백질 2차구조 예측. 미지의 단백질 서열로부터 2차구조를 예측하는 과정은 일반적으로 10여 단계를 거치게 되나 툴킷을 이용하여 단지 3단계로 매우 간편하고 빠르게 구조예측을 할 수 있다. 사용자 편의를 위하여 웹페이지에 미지의 단백질 서열을 입력하고, fast_hmmer를 거쳐 데이터베이스 검색과 동시에 다중 서열정렬을 실행할 수 있도록 만들어져 있다. 수행 결과는 Quick2D 프로그램으로 링크되어 바로 단백질 2차 구조를 예측 할 수 있다. 모든 과정은 불과 10분 이내에 진행된다. 사용자 임의로 필요에 따라 서열정렬 혹은 2차 구조 예측의 정확도와 빠르기를 조절이 가능하며, 툴킷에 포함된 여러 가지 프로그램을 선택적으로 지정하여 데이터를 처리할 수 있다.

그리드 시스템에서 또 하나의 중요한 사항은 바로 보안이다. 데이터가 여러 곳에 분산되어 저장되거나, 여러 사람이 동시에 사용할 수 있도록 디자인되어야 하는 그리드 시스템에서 항상 보안에 문제가 발생할 수 있는 소지는 충분히 존재한다. 따라서 많은 생물정보학 관련 데이터베이스들은 보안을 중요하게 생각하고 있다. 하지만, 보안에 너무 치우치게 되면 자칫 데이터의 공공성이라는 근본취지를 훼손하거나, 방해하게 되는 일이 발생할 수 있다. 이러한 문제의 해결책으로 우리는 2006년 Life Science Grid Workshop(LSGrid)에서 발표된 "*Shibboleth*"기술에 주목할 필요성이 있다. 이것은 "Web Single SignOn (SSO)"기술을 제공하여, 그리드-데이터베이스 서버와 로컬 데이터베이스 서버간의 인증관리를 하게 된다(Sinnott et al., 2006).

Grid-DBMS(Alosio et al., 2005)는 다양한 형태의 데이터 관리를 위하여, 상이한 아키텍쳐로 구성된 데이터베이스 환경하에서도 전혀 다른 성격의 데이터들을 문제없이 관리하기 위하여 개발되었다. 이러한 통합 데이터베이스 관리 시스템은 생물정보 데이터를 다루는 최적의 방식으로 인식되고 있으며 ORACLE사는 물론 각국의 개별 연구진들을 통하여 최근 개발이 활발히 진행되고 있다. 이 시스템이 지니는 특징 중의 하나로, 데이터의 종류도 다양하고 이를 운영하는 데이터베이스 시스템 또한 매우 다양하여 사용자 임의대로 여러 개의 상이한 데이터베이스를 통합하거나 그리드로 묶을 필요가 있을 때 이 Grid-DBMS의 이용은 매우 중요한 요소로 널리 인식되고 있다.

Oxford대학이 중심이 되어 만들어진 BioSimGrid (http://www.biosimgrid.org/)는 생화학 분자(biomolec- ules)의 특성과 거동들을 하나의 구심점으로 하는 방대한 규모의 컴퓨터 시뮬레이션들을 제공한다(Ng et al., 2006). 현재까지 약 20만개 이상의 단백질 서열과 3만 3천 개 이상의 알려진 단백질 구조들이 BioSimGrid에 축적되어 있으며, 궁극적으로는 구조 유전체학(structural genomics)과 시스템즈바이올로지(systems biology)사이의 상호 연결고리를 제공하기 위하여 다양한 형태의 생화학적 물성 데이터와 각종 시뮬레이션 결과들을 제공한다. 즉, BioSimGrid는 그리드 시스템에 기반을 둔 데이터베이스 구성을 통하여 단백질 서열들을 관리하고, 단백질간의 분자구조 시뮬레이션 수행이 가능케 할 뿐만 아니라, 획득된 결과를 바로 분석할 수 있도록 시스템이 구축되어 있다. 이와 같이 시뮬레이션 데이터의 신속한 비교분석을 위해 BioSimGrid는 고성능 컴퓨팅 환경인 그리드-데이터베이스에 초점을 맞추어 개발되었다. 특히 이중-그리드(dual-grid) 시스템 구축을 통하여 데이터의 처리뿐만 아니라 이의 효율적인 관리까지도 동시에 수행이 가능하다.

미래 연구를 위한 제언

앞서 살펴본 바와 같이 기하 급수적으로 늘어가는 DNA, RNA, 단백질 서열과 같은 서열정보와 단백질 구조 및 상호작용 등에 대한 수 많은 데이터, 연산, 그리고 복잡한 결과들을 신속하게 처리하고, 효과적으로 관리하기 위하여 바이오그리드 시스템은 하나의 해결점으로서 자리매김하고 있다. 불과 10여년 전만 하더라도 소규모 연구실에서 한 명의 연구자 혹은 학생이 다루는 유전자나 단백질은 많아야 서너개 정도였다. 그러나, 현재는 한 명이 하나의 염색체 단위 혹은 세포 내 단백질 군, 유전자 군을 다루어야 하는 시대로 연구환경이 급변하였다. 전체(-ome)를 다루는 집합체에 대한 연구는생체 내 연관 및 직간접적 메커니즘 규명 연구의 중요한 열쇠를 제공하기 때문이다. 무수히 많은 유전자와 단백질 집합체들의 기능이상이 하나의 표현형으로 나타나는 경우가 상당수 존재하는데, 당뇨(diabetes) 등이 그 대표적인 예가 될 수 있다. 또한 단백질구조 및 기능예측 부분에서도 잠시 언급되었지만, 이제 걸음마 단계를 벗어나고 있는 이들 분자동력학적 시뮬레이션 기술들은 더욱 많은 개선과 바이오그리드 시스템과의 접목이 요구된다. 생체반응의 대부분을 차지하는 반응용 단백질, 예로 효소, 들은 그 구조 기능 해석만 가지고는 이의 반응성 (reactivity) 분석이 절대 불가능하다. 유사한 구조를 지닌 단백질이라 하더라도 이의 반응특이성은 단지 하나의 아미노산치환을 통해서도 급격하게 달라지기 때문이다 (Joo et al., 1999). 이를 해결하기 위해서는 퀀텀(quantum) 레벨의 전자궤도 함수를 구현하고 이를 계산하기 위한 노력이 별도로 동원되어야 한다. 이 모든 것을 하나의 시뮬레이션 툴에 넣기 위한 작업들이 이제 막 태동하고 있다. 마지막에 언급되었듯이 BioSimGrid는 효율적인 데이터 처리뿐만 아니라, 효과적인 관리측면에서도 좋은 시도로 받아들여질 수 있다. 하지만, 대용량 생물정보학 분야발전에 커다란 원동력이 된 그리드-컴퓨팅 기술은 그 바탕이 Unix/Linux에 기반을 두기에 현실적으로 파생될 수 밖에 없는 일부 부정적인 면도 없지는 않다. 그 이유로 그리드-컴퓨팅 환경에서 구동가능하게 구현된 응용프로그램 혹은 데이터베이스 시스템은 아직까지는 일반유저들에게 널리 활용되고 있지 못하다. 병렬처리 환경이 대부분 Unix/Linux 기반으로 구현되고 있기 때문에, 이들 연구는 Unix/Linux운영체제 환경에 익숙한 소수의 연구자들에 의해 주도되고 있기 때문이다. 따라서, 윈도우환경에 익숙한 대다수의 연구자들을 위하여 병렬화 코딩 라이브러리 및 이의 소스를 손쉽게 얻을 수 있는 윈도우의 개발은 매우 고무적인 일이다. 보다 많은 사용자를 위해 Unix/Linux 기반으로 구현된 병렬 프로그램들의 윈도우로의 이식이 현재 가장 시급한 과제가 아닐까 생각된다.

참 고 문 헌

[1] Alosio, G., Cafaro, M., Fiore, S., Mirto, M. (2005) The grid-DBMS: towards dynamics data management in grid environments, *ITCC 2005*,

2:199-204.

[2] Altschul, S., Gish, W, Miller, W., Meyers, E., Lipman, D. (1990) Basic Local Alignment Search Tool, *J. Mol. Biol.* **215**:403-410.

[3] Amdahl, G. (1967) Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities", *AFIPS Conference Proceedings*, 30:483-485.

[4] Bassingthwaighte, J. B. (2005) Strategies for the physiome project. *Annal. Biomed. Eng.* **28**:1043-1058.

[5] Biegert, A., Mayer, C., Remmert, M., Soding, J., Lupas, A.N. (2006) The MPI Bioinformatics Toolkit for protein sequence analysis, *Nucleic Acids Res.* **34**:W335-W339.

[6] Bioinformatics Toolkit, http://www.cardiff.ac.uk/biosi/research/biosoft/index.html.

[7] CellML Modeling Language, http://www.cellML.org.

[8] CLRC, http://www.clrc.ac.uk.

[9] Eddy, S.R. (1998) Profile Hidden Markov Models, *Bioinformatics*, **14**:755-763.

[10] Informatics Benchmarking Toolkit, http://web.bioteam.net/metadot/index.pl?iid=2378.

[11] IUPS physiome project, http://www.physiome/org.

[12] Joo, H., Lin, Z., Arnold, F. H. (1999) Laboratory evolution of peroxide-mediated cytochrome P450 hydroxylation. *Nature* 399: 670-673.

[13] Kent, W.J. (2002) BLAT-The BLAST-Like Alignment Tool, *Genome Res.* 12:656-664.

[14] Konagaya, A. (2006) Trends in life science grid: from computing grid to knowledge grid, *BMC Bioinformatics*, **7 (Suppl 5)**: S10.

[15] Li, K.B. (2003) ClustalW-MPI: ClustalW analysis using distributed and parallel computing, *Bioinformatics*, **19**: 1585-1586.

[16] Message Passing Interface, http://www-unix.mcs.anl.gov/mpi/.

[17] MrBayes: Bayesian Inference of Phylogeny, http://mrbayes.csit.fsu.edu.

[18] Ng, M.H., Johnston, S., Wu, B., Murdock, S.E., Tai K., Fangohr, H., Cox, S.J., Essex, J.W., Sansom, M., Jeffreys, P. (2006) BioSimGrid: Grid-enabled biocolecular simulation data storage and analysis, *Future Gener. Comput. Syst.* **22**:657-664.

[19] Noble, D. (1960) Cardiac action and pacemaker potentials based on the Hodgkin-Huxley equations. *Nature* **188**: 495-497.

[20] Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greendwood, M., Carver T., Glover, K., Pocock, M.R., Wipat, A., Li, P. (2004) Taverna: a tool for the composition and enactment of bioinformatics workflows, *Bioinformatics*, **20**:3045-3054.

[21] Phillips, J., Braun R., Wang W., Gumbart J., Tajkhorshid, E., Villa, E., Chipot, C., Skeel, R., Kale, L., Schulten K. (2005) Scalable molecular dynamics with NAMD, *J. Comp. Chem.*, **26**: 1781-1802.

[22] Shah, S.P., He, D., Sawkins, J. Druce, J., Quon, G., Lett, D., Z heng, G., Xu, T., Ouellette, F. (2004) Pegasys: software for executing and integrating analyses of biological sequences, *BMC Bioinformatics*, **5**:40.

[23] Shibboleth Project, http://shibboleth.internet2.edu.

[24] Sinnott, R., Ajayi, O., Stell, A. Jiang, J., Watt, J. (2006) User-Oriented Access to Secure Biomedical Resources through the Grid, *Proceedings of the LSGRID2006*, **2006**:71-86.

[25] Sun, C.-H. Yi, G.-S. (2006) Bioinformatics Analysis System Using Current Grid Technology, *Bioinformatics and Biosystems*, **1**:157-164.

[26] Sun, Y., Z hao, S., Yu, H., Gao, G. and Luo, J. (2007) ABCGrid: Application for Bioinformatics Computing Grid, *Bioinformatics*, in press.

[27] Tabb, D.L., McDonald, W.H., Yates, J.R. III. (2002) DTASelect and Contrast: tools for assembling and comparing protein identifications from shotgun proteomics. *J. Proteome Res.* **1**:211-215.

[28] Tang, F., Chua, C.L., Ho, L., Lim, Y.P., Issac, P., Krishnan, A., (2005) Wildfire: distributed, Grid-enabled workflow construction and execution, *BMC Bioinformatics*, **6**:69.

[29] Van Der Spoel D., Lindahl, E., Hess B., Groenhof, G., Mark, A.E., Berendsen, H.J. (2005) GROMACS: fast, flexible, and free, *J. Comput. Chem.* **26**:1701-1718.

[29] World Community Grid, http://www.worldcommunitygrid.org