J. Exp. Biomed. Sci. 10 (2004) 523-528

Implementation of Integrated Analysis System for Bioinformatics Analysis

Bong-Oh Koo[1†] and Yong-Won Shin[2]

[1]Department of Physical Therapy, [2]Department of Healthcare Management, Catholic University of Pusan, Busan 609-757, Korea

The core factor of the study is integrated environment based PC-Cluster system and high speed access rate up to 155 Mbps, continuous collection system for bioinformatics information at home and abroad. The results of the study are establishment and stabilization of information and communication infrastructure, establishment and stabilization of high performance computer network up to 155 Mbps, development of PC-Cluster system with 32 nodes, a parallelized BLAST on Cluster system, which can provides scalable speedup in terms of response time, and development of collection and search system for bioinformatics information.

Key Words: KOERN, Cluster system, Sequence analysis, Bioinformatics, BLAST, mpiBLAST

서 론

Watson과 Crick의 염기구조 발견이후 생화학 및 분자생물학의 발달과 1990년대의 인간 유전체 프로젝트 (Human Genome Project)로 인하여 생물학적 정보의 양은 급속도로 증가하게 되었다. 특히 1970년대 이후 염기나 단백질의 서열을 자동으로 분석할 수 있는 각종 기기 (Automatic DNA sequencer, DNA microarray, Image analyser, Mass spectroscopy)와 고속처리 탐색 기술 (High throughput screening)의 개발로 생물학적 정보의 양은 기하급수적으로 증가하고 있다. 최근에는 다양한 방식을 통해 데이터를 얻는 작업 자체를 위해서도 여러 가지 전산적인 도구들이 필요하게 되었다. PC 클러스터 시스템은 저가의 고성능 PC 및 워크스테이션을 고속 네트워크로 연결하여 고성능 서버의 성능을 발휘하는 방식으로, 저가의 상용제품을 사용함으로써 기존의 고성능 서버보다 수배에서 수십배 적은 비용으로 동일한 성능의 시스템을 구현할 수 있어 가격대 성능비가 높은 장점을 가진다. 또한, 사용자가 직접 상용부품을 사용하여 손쉽게 확장할 수 있기 때문에 시스템 유지비용이 감소하고, 사용이 편리한 PC 및 워크스테이션의 환경을 그대로 사용할 수 있기 때문에 병렬 프로그램의 연구, 개발 및 운영이 용이하다. 이미 전세계 슈퍼컴퓨터 상위 500위 중 43.8%에 해당하는 219개 컴퓨터가 클러스터 시스템 방식으로 구축되어 있으며 (http://www.top500.org), 클러스터 시스템의 장점을 활용한 효율적인 병렬형 서열검색 시스템은 유전체 연구기관간 유기적 네트워크 형성과 공동연구의 초석으로 활용될 수 있다. 본 연구에서는 서열검색 시스템을 위한 클러스터 시스템의 효율적인 구조를 제안하고, 구축된 서열검색 시스템에 대한 성능 평가를 통해 향후 개선 방향을 살펴본다.

*논 문 접 수: 2004년 9월 7일
수정재접수: 2004년 12월 6일
†교신저자: 구봉오, (우) 609-757 부산광역시 금정구 부곡 3동 9번지, 부산가톨릭대학교 보건과학대학 물리치료학과
Tel: 051-510-0574, Fax: 051-510-0578
e-mail: kbo905@cup.ac.kr

관련 연구

1. Cluster 기반 BLAST의 단계별 병렬화

생물정보학 (Bioinformatics)에서 가장 기본적인 작업 중 하나는 새롭게 서열을 알게 된 염기와 이미 기록되어 있는 염기서열 간의 유사성 (Similarity)이나 상동성 (Homology)을 검색하는 것이다. 이러한 작업을 통해서 연구자들은 새롭게 얻어진 유전자 서열에 의해 부호화된 단백질의 종류를 예측할 수 있다. BLAST (Basic Local Alignment Search Tool) (http://www.ncbi.nlm.nih.gov/BLAST)는 웹상에서 가장 대중화되고 사용자에게 친숙한 서열 유사성 검색도구로서, 미국생명공학연구소 (NCBI: National Center for Biotechnology Information)의 유전자 서열 데이터베이스 (GenBank) 뿐만 아니라 SWISS-PROT이나 PDB 등과 같은 모든 주요 서열 데이터베이스를 검색할 수 있는 도구이다. 1990년대 인간 유전체 프로젝트가 본격화된 후 유전자 서열 데이터베이스 크기의 급격한 증가 (http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html)때문에 BLAST 검색시간의 단축이 생물정보 연구에서 매우 중요한 요소로 자리 잡게 되었다. 따라서 BLAST 검색 시간의 단축을 위해서 SMP (Symmetric Multiprocessing) 호스트 기반의

Table 1. Parallelization phase of BLAST

	Fine grained	Medium grained	Coarse grained
Subject (s)	1 sequence	1 sequence	N sequence
Target (s)	1 sequence	M sequence (in database)	M sequence (in database)
Parallelism	Multiple alignments on single sequence pair	Partitioned database Multiple targets examined at once	Replicated database Partition input sets

Table 2. Configuration of hardware and Operating System of genome sequence analysis system

역할 분담	Operating System	H/W Detail
Server Node	RedHat 7.3 (Kernel Version 2. 4. 18)	2-way P-III 1.2 GHz CPU 1 GB memory Intel Ethernet Pro100
calculation Node	RedHat 7.3 (Kernel Version 2. 4. 18)	P-IV 2GHz 1 GB memory Intel Ethernet Pro100
storage Server	SUN-Fire Workstation	SunOS 5.8 sun4u

병렬화가 주로 진행되었으며, 그 결과 SMP 호스트의 쓰레드에 기반한 NCBI BLAST가 개발되었다. 하지만, 프로세서 개수가 제한되는 SMP 구조의 특성상 제한된 확장성 (Scalability)을 가지는 문제점이 존재한다.

Braun 등이 수행한 클러스터 시스템 상에서의 BLAST 병렬화 연구 (Anne Julich, 1995; Braun et al., 2001; David Mathog, 2003, 2003; Hong Soog Kim)에서는 BLAST 병렬화를 세 가지 방식으로 나눈다. 세립형 (Fine grained) 병렬화에서 다중 정렬 비교는 상호 독립적으로 이루어지기 때문에 매우 효과적이며, 특정 하드웨어를 사용할 경우 더욱 효과적이다. 중립형 (Medium grained) 병렬화는 계산 노드별로 데이터베이스를 분할하는 것이다. 이렇게 함으로써 데이터베이스에서 요구하는 메모리양을 줄일 수 있다. 중립형 병렬화에서 서버 노드는 사용자 질의를 분할하고 각 계산 노드에 분배하며, 검색 완료 후 각각의 결과를 취합한다. 이러한 구조는 중복성이 배제됨으로써 데이터베이스양 증가에 따라 컴퓨팅 파워를 추가할 수 있는 장점이 있다. 조립형 (Coarse grained) 병렬화는 BLAST 엔진 소스의 수정없이 각 계산노드에 데이터베이스 모두를 저장해 두고, 사용자 질의를 분할하여 배치처리 (Batch processing)하는 것이다. 이 방식은 단일 사용자 질의의 검색시간은 동일하나 웹 서비스와 같이 동시다발적인 다수의 사용자 질의 검색 서비스에 유용하다. 하지만, 전체 데이터베이스가 각 계산 노드에 적재되기 위해서는 많은 양의 메모리가 필요하고, 사용자 질의 서열 비교 시 응답 시간의 향상이 없고, 복제 데이터베이스의 일관성을 관리해야 하는 단점이 있다.

2. mpiBLAST

mpiBLAST (http://www.epowergate.co.kr/biz/clu_ezcon.html)는 (Table 1)의 구분에 따르면 중립형 병렬화에 해당하며, BLAST 알고리즘 자체는 수정하지 않고, 프로세스 제어와 생성, 데이터 통신에만 수정을 가한 BLAST 프로그램이다. mpiBLAST의 사용은 데이터베이스가 분할되어 공유 저장장치에 저장되는 단계와 mpiBLAST 질의가 각 계산 노드에서 검색되는 두 단계로 이루어진다. mpiBLAST의 검색 단계는 검색에 필요한 데이터베이스 파티션 복사 단계와 실제 검색 단계, 그리고 결과 취합 단계로 다시 나눌 수 있다. 즉, 주어진 사용자 질의에 대해 필요한 데이터베이스 파티션을 각 계산 노드로 복사하고, 복사된 각 파티션 내에서 서열 유사성 검색을 수행한 후 각각의 파티션에 대한 검색 결과를 취합하는 과정을 거친다.

3. 클러스터 시스템 기반의 유전자 서열 분석 시스템 구축

1) 시스템 구성

본 연구에서 구축한 유전자 서열 분석 시스템은 베오울프 (Beowulf) 계열의 리눅스 클러스터 시스템을 기반으로 하였다. 플랫폼에 관한 자세한 사항은 (Table 2)에 나타나 있으며, 병렬화 된 BLAST를 수행하기 위해 MPI (Message Passing Interface)를 사용한다. 총 4대의 역할 분담 서버, 32대의 계산 노드, 그리고 1대의 스토리지 서버가 100 Mbps 이더넷 (Ethernet)으로 연결되어 있으며, 역할 분담 서버는 역할에 따라 웹 서버, 관리 서버, FTP 서버, DNS 및 메일 서버로 나뉜다.

본 연구에서 구축한 클러스터 시스템은 각 서버와 계산 노드의 상태 확인 및 관리를 위해 ezCon (http://www.epowergate.co.kr/biz/clu_ezcon.html) 클러스터 관리 도구를 사용하며, 작업 스케쥴링을 위해 PBSpro (http://altair.com/software/pbspro.html) 큐잉 시스템을 사용한다. 정확성 보장과 개발 기간 단축을 위해 BLAST 알고리즘의 수정을 배제하고 배치작업의 처리율을 높이는데 기본적인 초점을 맞추고 있다. 사용자의 질의는 웹 페이지를 통해 클러스터 시스템의 전단에 위치하는 웹 서버로 전달되고, CGI 프로그램에 의해 생성된 PBS 스크립트가 PBS 큐에 추가된다. PBS 스케쥴러는 32대의 계산 노드들 중 가장 부하가 적은 노드를 선택하여 사용자 질의를 수행한다. (Fig. 1)은 일련의 질의 검색 과정을 도식화 한 것이다.

2) 시스템 최적화

본 연구에서 구축한 클러스터 시스템은 제공하는 서비스

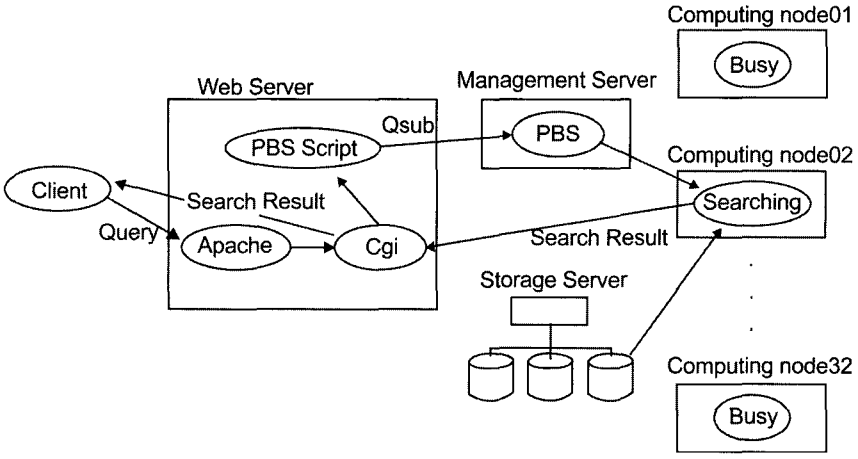

Fig. 1. Inquiry execution process of sequence analysis cluster system.

(http://www.biohealthinfo.org)의 특성상 다음과 같은 특징을 가진다.

(1) 웹을 통해 사용자 질의를 수신하고 이를 큐잉 시스템에 전달하는 웹 서버의 CPU 및 네트워크 부하가 높다.

(2) 사용자 질의를 데이터베이스에서 검색하는 계산 노드는 CPU 부하가 높다.

(3) 대용량의 데이터베이스를 각 계산 노드에 제공하는 스토리지 서버는 CPU, 메모리, 네트워크 부하가 높다.

(4) 웹 서버와 관리 서버는 모든 사용자 질의에 대해 1개 이상의 TCP 커넥션을 생성하기 때문에 커넥션 관리 부하가 높다. 특히 모든 사용자 질의에 대해 웹 서버와 1개 이상의 TCP 커넥션을 생성하고, 32대의 계산 노드와 작업 분배를 위한 1개 이상의 TCP 커넥션을 생성하는 관리 서버의 네트워크 부하가 가장 높다.

(5) 스토리지 서버의 파일 시스템을 NFS (Network File System)로 공유하기 때문에 모든 계산 노드와 웹 서버, 관리 서버, FTP 서버는 시간 동기화가 필요하다.

위에서 열거한 문제점들을 해결하기 위해 리눅스 커널의 수정 및 설정 최적화를 수행하고, 모든 작업 분배에 TCP 커넥션을 생성하는 OpenPBS를 배제하고 UDP를 사용하는 PBSpro를 사용하며, 인터넷 데몬 (Internet daemon)의 설정을 최적화하였다. 웹 서버와 관리 서버가 겪게 되는 대부분의 네트워크 부하는 TCP 연결의 능동 닫기 (Active close)로 인한 커넥션 수의 증가와 어느 순간 TCP 커넥션 개수가 폭발적으로 증가할 경우 새로운 커넥션 생성이 불가능하다는 두 가지로 요약할 수 있다. 이를 해결하기 위해 TCP의 TIME WAIT 시간을 단축하고 순간적으로 사용자 질의가 증가할 것을 대비해 TCP의 소켓당 백로그 (Backlog) 항목수를 늘려주었다. 원활한 NFS의 동작을 위해 모든 계산 노드 및 서버 노드에 시간 동기화를 위한 NTP (Network Time Protocol)를 설치하고, 웹 서버를 NTP 서버로 활용하였다. 따라서, 모든 서버 및 계산 노드는 주기적으로 웹 서버와 통신하며 서로의 시간을 정확히 맞추게 된다. 마지막으로, 각 서버의 인터넷 데몬 설정을 변경하여 사용자 질의가 특정 시간에 폭발적으로 들어오더라도 견딜 수 있는 내구성을 갖추었다. 일반적인 리눅스 배포판은 인터넷 데몬 기본 설정이 특정 서비스의 최대 허용량이 60개로 고정되어 있고, 초당 25개 이상의 TCP 연결 시도가 있을 경우 30초간 모든 TCP 연결 시도를 비활성화 시킴으로써 제한된 서비스만을 제공할 수 있도록 설정되어 있었던 것에 반해, 본 연구에서 구축한 클러스터 시스템의 웹 서버와 관리 서버는 단위 시간당 더 많은 서비스를 제공할 수 있도록 하기 위함이다. 최적화를 위해 사용한 구체적인 값들은 아래와 같다.

(1) 시간 동기화를 위해 웹 서버에 NTP 서버를 설치하고, 상위 시간 동기화 서버로 time.nuri.net을 지정하였다. 따라서, 웹 서버는 주기적으로 time.nuri.net 서버와 통신하며 정확한 시계를 유지한다. 그 외 계산 노드 및 스토리지 서버 등은 웹 서버의 NTP 서버와 통신하며 시간을 동기화함으로써 클러스터를 구성하는 모든 컴퓨터들이 동기화된 시간으로 동작할 수 있도록 하였다.

(2) 일반적으로 배포되는 리눅스 커널의 TCP_TIMEWAIT_LEN 값은 기본 설정이 60초로 되어 있지만, 이는 WAN (Wide Area Network) 환경을 고려한 여유있는 값이다. 클러스터 시스템과 같이 안전한 LAN 환경에서 빈번한 TCP 연결 설정 및 해제를 반복하는 경우 TCP_TIMEWAIT_LEN의 값을 단축하여 시스템 성능을 극대화 할 수 있다. 따라서, TCP_TIMEWAIT_LEN의 값을 15초로 단축하고 커널 재컴파일을 수행하여 최적화된 커널을 사용하여 시스템 성능을 향상시킬 수 있었다.

(3) 외부의 사용자들에게 서비스를 제공하는 시스템의 경우

사용자 요구가 한꺼번에 몰리는 상황에 대비해야 한다. 따라서, 일반적인 리눅스 배포판의 경우 1024로 설정되어 있는 TCP 백로그 값을 증가시켜 충분한 백로그 큐 항목 (Backlog queue entry)을 확보하는 것이 필요하다. 이를 위해 백로그 값을 8192로 변경하였다.

(4) 웹 서버와 관리 서버의 경우 CPU 부하보다는 외부의 사용자 질의를 처리하고 작업을 분배하기 위한 네트워크 부하가 시스템의 성능을 좌우한다. 웹 서버와 관리 서버의 인터넷 서버 설정을 조정하여 단위 시간당 처리 가능한 사용자 질의를 증가시키기 위해 최대 서비스 인스턴스의 개수를 무제한으로 설정하고, 단위 시간당 처리 가능한 TCP 연결 수락의 개수를 25개에서 60개로 증가시켰으며, 단위 시간당 처리 가능한 TCP 연결 수락의 개수를 초과하는 연결 시도가 들어올 경우 인터넷 서버를 비활성화 시키는 시간을 30초에서 20초로 조정하였다. 이를 통해 단발적인 서열 검색 요청의 네트워크 부하를 최소화 시킬 수 있었다.

Table 3. Sizes and kinds of database

Kind of Database	Ecoli	NT
Size of Database	1.34 MB	2.61 GB

4. 클러스터 시스템 기반의 유전자 서열 분석 시스템 성능 평가

1) 성능평가 환경

클러스터 시스템에 기반한 유전자 서열 분석 시스템의 성능을 평가하기 위해 앞에서 언급된 클러스터 시스템 상에서 (Table 3)과 같은 생물정보학 데이터베이스를 사용하였다. 비교적 작은 용량의 Ecoli 데이터베이스와 대용량의 NT 데이터베이스에 대해 9226C 염기 서열과 1461C 단백질서열을 사용하여 질의검색 시 걸린 시간을 측정하였다. 성능 측정 결과에

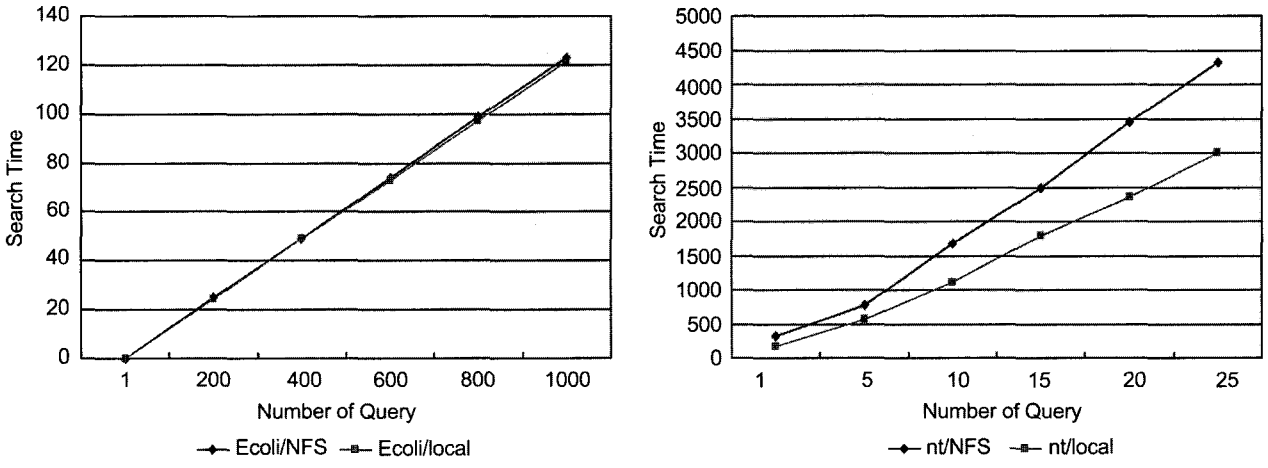

Fig. 2. Genome sequence analysis at single node.

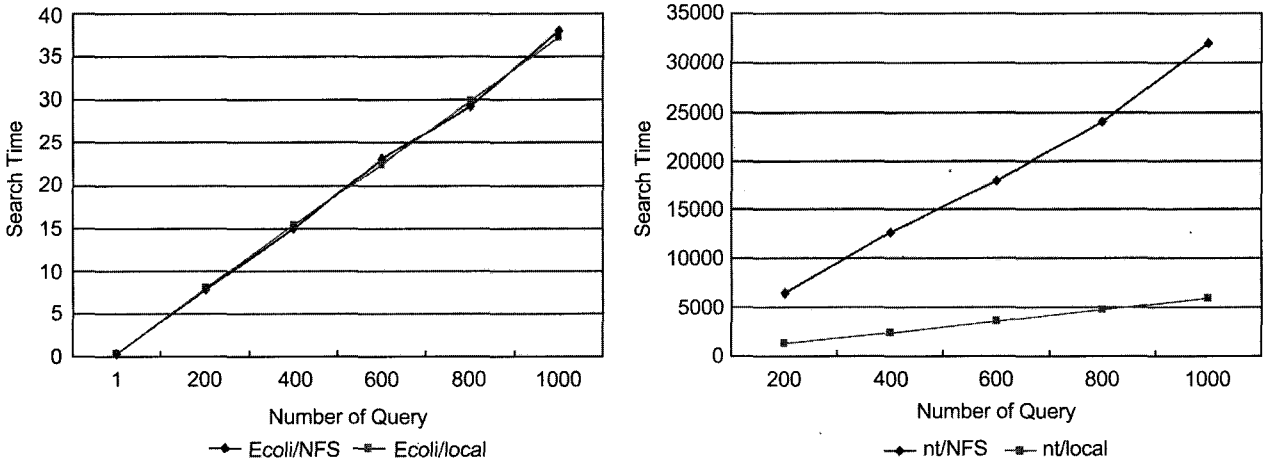

Fig. 3. Genome sequence analysis at cluster system.

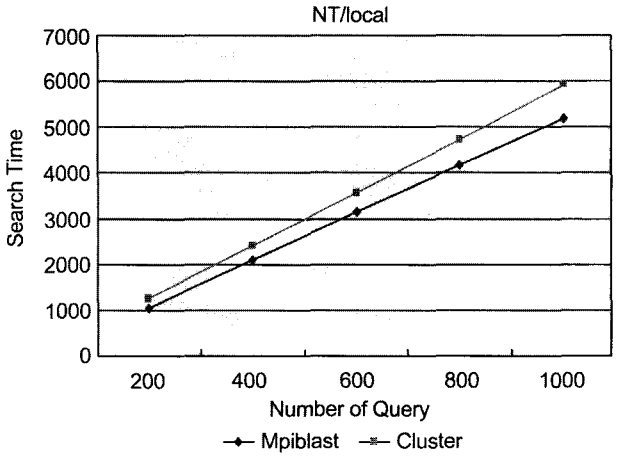

Fig. 4. Genome sequence analysis using mpiBLAST

서 나타난 수행 시간은 모든 사용자 질의가 수행 완료된 시간을 의미한다.

2) 성능평가 결과

(Fig. 2)는 단일 계산 노드에서 서열 분석을 수행할 경우 유전자 서열 데이터베이스가 계산 노드의 지역 저장 장치에 있을 경우 (DB/local)와 NFS로 공유되는 저장 서버의 공유 저장 장치에 있을 경우 (DB/NFS)의 검색시간 차이를 나타낸다. 작은 용량의 Ecoli 데이터베이스에 대한 검색의 경우 Ecoli/local과 Ecoli/NFS의 경우에 대해 검색 시간 차이가 크지 않고, 동시에 수행되는 사용자 질의 개수가 증가해도 검색시간은 선형적으로 증가한다. 이러한 현상은 두 경우 모두에 대해 운영체제의 파일 캐쉬가 Ecoli 데이터베이스를 모두 캐쉬하기 때문이다. 또한, 데이터베이스 크기가 크지 않기 때문에 스와핑 (Swapping)이 발생하지 않고, 이로 인해 사용자 질의 개수가 증가해도 검색 시간은 선형적으로 증가한다. 하지만, 대용량 NT 데이터베이스에 대한 검색의 경우 검색시간 차이가 크게 나타난다. 단일 사용자 질의를 검색하는데 NT/local의 경우 178.79초가 걸린 반면, NT/NFS의 경우 321.11초로 검색 시간이 약 80%정도 증가한다. 대용량의 NT 데이터베이스는 운영체제의 파일 캐쉬에 모두 캐쉬 될 수 없고, 스와핑이 빈번하게 발생하기 때문에 동시에 수행되는 사용자 질의 개수가 증가하면 시스템 멈춤 (System hanging) 현상이 발생한다. 하지만 적당한 개수의 사용자 질의가 동시에 수행될 경우 파일 캐쉬 적중으로 인한 이득이 있으며, 순차적 검색에 비해 검색 시간이 단축되는 슈퍼리니어 (Super-linear) 효과가 나타난다. 이는 기울기가 1보다 작은 (Fig. 2)의 우측 그래프로부터 관찰할 수 있다.

(Fig. 3)은 32대의 계산 노드로 구성된 클러스터 시스템에서 유전자 서열 분석을 수행했을 때의 검색 시간을 나타낸다. R.C.Braun의 분류에 따르면 조립형 병렬화에 해당하는 이 실험에서 사용자 질의는 각 계산 노드로 분배되어 검색된다. 단일 노드에서 단일 사용자 질의를 검색할 경우 0.12초가 걸린데 반해, 클러스터 시스템에서 단일 사용자 질의를 검색할 경우 0.36초로 3배 가량 검색 시간이 증가한다. 이는 큐잉 시스템에 검색 작업을 삽입하고, 검색 결과를 복사하는 과정에서 발생하는 오버헤드가 있기 때문이다. 하지만 1,000개의 사용자 질의에 대한 Ecoli/local 검색 시간은 단일 노드와 클러스터 시스템의 경우 각각 127.15초와 37.21초로 클러스터 시스템이 더 짧다. 즉, 사용자 질의 개수가 증가하면 클러스터 시스템을 사용한 검색이 단일 노드를 사용한 검색에 비해 단축된 검색 시간을 보인다. 작은 용량의 Ecoli 데이터베이스에 대한 검색 시간은 사용자 질의 개수와 비례해서 선형적으로 증가하고, Ecoli/local과 Ecoli/NFS 사이의 큰 차이가 없다. 이는 단일 노드에서 설명된 것처럼 운영체제의 파일 캐쉬 영향 때문이다. 대용량의 NT 데이터베이스에 대한 검색은 단일 노드에서 불가능했던 수백개 이상의 사용자 질의 검색도 가능하며, 검색 시간은 NT/local이 NT/NFS에 비해 짧게 나타난다. 이는 다수개의 검색이 동시에 수행될 때 발생하는 공유 저장 서버의 병목 현상 때문이다.

(Fig. 2)와 (Fig. 3)의 실험을 통해 다음과 같은 사실을 확인할 수 있다. 첫째, 작은 용량의 데이터베이스에 대한 검색은 단일 노드로도 충분하며, 클러스터 시스템을 사용하는 것이 오히려 오버헤드를 줄 수 있다. 둘째, 대용량의 데이터베이스에 대한 검색은 클러스터 시스템을 사용하여 검색 작업을 계산 노드에 분배하는 것이 효과적이다. 셋째, 클러스터 시스템을 사용한 대용량 데이터베이스에 대한 검색은 공유 저장 장치를 사용하는 것보다 지역 저장 장치를 사용하는 것이 효과적이다.

(Fig. 4)는 BLAST의 중립형 병렬화 버전인 mpiBLAST를 사용하여 NT 데이터베이스를 검색했을 경우의 검색 시간을 나타낸다. 동일한 단일 사용자 질의에 대해 클러스터 시스템 상에서 조립형 병렬화 형태로 수행했을 경우 Tc=179.68초가 걸린데 반해, 단일 사용자 질의를 계산 노드 수에 따라 분리하여 중립형 병렬화 형태로 수행 할 경우 Tm=5.19초가 걸렸다. 단일 사용자 질의를 중립형 병렬화를 통해 수행한 경우 (Tm)가 순차적으로 수행한 경우 (Tc)에 비해 32배 이상 빠르다는 사실을 알 수 있다. 이러한 슈퍼리니어 (Super-linear) 현상 [8]은 분할된 데이터베이스가 파일 캐쉬에 모두 포함될 수 있어서 저장 장치로의 I/O 트래픽이 줄어들었기 때문이다. 사용자 질의의 개수가 일정 개수 이상 증가하면 중립형 병렬화 BLAST 검색과 조립형 병렬화 BLAST 검색의 검색 시간 차이가 227.1초, 328.7초, 414.8초, 563.6초, 745.8초로 꾸준히 증가하는 점을 관찰할 수 있다. 이는 사용자 질의의 개수와 상관없이 중립형 병렬화 BLAST 검색이 항상 좋은 성능을 보임을 의미한다. 전체적으로 동일한 개수의 CPU와 메모리를

사용하지만, 검색을 수행하는 계산 노드의 입장에서는 중립형 병렬화 검색이 조립형 병렬화 검색보다 더 작은 크기의 분할된 데이터베이스를 검색한다. 따라서, 요구되는 데이터베이스 조각들이 운영체제의 파일 캐쉬에 포함될 확률이 높아지고, 그로 인해 저장 장치로의 I/O 트래픽이 감소하기 때문이다.

결 론

NCBI BLAST는 SMP 호스트의 쓰레드 기전을 이용하여 병렬화되어 있다. SMP 시스템은 시스템 자체의 구조적 특성에 의해 일정수의 프로세서 이상은 확장되지 않기 때문에 제한적인 속도향상 밖에 달성하지 못한다. 반면에, 155M 초고속 선도망 기반 저비용 고효율의 클러스터 시스템을 이용한 유전자 서열 분석 시스템은 저가의 하드웨어를 이용함으로써 SMP 시스템에 비하여 비용-효과적인 병렬 프로세서 시스템을 쉽게 구현할 수 있다.

본 연구에서는 클러스터 시스템 구축시 운영체제의 최적화 방안을 제안하였고, 대용량 서열 데이터베이스를 이용한 성능평가를 통하여 우수하고, 향상된 속도향상을 제공하였다. 이를 통해 유전체 연구기관간 유기적 네트워크 및 공동연구의 기반으로 활용될 수 있을 것으로 예상한다. 향후 연구에서는 클러스터 시스템내의 효과적인 자원 할당, 데이터베이스 크기에 따른 검색 작업 분배 방법, 다양한 생물정보 응용 프로그램의 병렬화 등에 관해 진행할 예정이다.

REFERENCES

Anne Julich. Implementations BLAST for Parallel Computers. CABIOS. 1995. 11: 3-6.

Braun RC, Pedretti KT, Casavant TL, Scheetz TE, Bitkett CL, Roberts CA. Parallelization of local BLAST service on workstations clusters. FGCS. 2001. 17: 745-754.

Darling AE, Lucas Carey, Wu-chun Feng. The Design, Implementation and Evaluation of mpiBLAST. ClusterWorld 2003 conference, 2003

David R.Mathog. Parallel BLAST on split databases. Bioinformatics. 2003. 19: 1865-1866.

Hong Soog Kim. Hyper-BLAST:A Parallelized BLAST on Cluster Systems. School of Engineering Information and Communication University. 2003. 1:15-16.

http://www.altair.com/software/pbspro.htm

http://www.biohealthinfo.org

http://www.epowergate.co.kr/biz/clu_ezcon.html

http://www.ncbi.nlm.nih.gov/BLAST/

http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html

http://www.top500.org