韓國 컴퓨터情報學會 論文誌
第13卷 第6號, 2008. 11.

2008-13-6-1-5

워크플로우 환경에서의 대규모 서열 유사성 검색 웹 서비스에 관한 연구

정 진 영*

A Study on Web Services for Sequence Similarity search in the Workflow Environment

Jin-Young Jung*

요 약

최근 생물정보학에서의 워크플로우 관리 도구를 이용한 생명 현상에 대한 연구가 활발하게 진행되고 있다. 워크플로우 관리 도구는 서비스의 재사용과 공유를 통해 연구자들이 서로 협업할 수 있는 기반으로 MyGrid 프로젝트의 Taverna를 비롯하여 Kepler, BioWMS 등의 다양한 워크플로우 관리 도구들이 오픈소스로 개발되어 사용 되고 있다. 이러한 워크플로우 관리 도구는 공간적으로 떨어진 서로 다른 서비스들을 웹 서비스 기술을 기반으로 하나의 작업 공간에서 연구 과정을 모델링하고 자동화 할 수 있도록 해준다. 생물정보학에서 사용되는 많은 도구와 데이터베이스들이 웹 서비스 형태로 제공되어 워크플로우 관리 도구에서 사용되고 있다. 이러한 상황에서 생물정보학에서 기본으로 사용되는 서열 유사성 검색에 대한 웹 서비스의 개발과 안정적인 서비스 제공은 생물정보학 분야에서 필수적이라 할 수 있다. 본 논문에서는 리눅스 클러스터를 기반으로 생물학 서열 데이터의 유사성 검색 속도를 향상시키는 한편, 이를 웹 서비스 형태로 개발하여 워크플로우 관리 도구와의 연동하여 단시간에 서열 유사성 검색을 가능하게 하였다.

Abstract

In recent years, a life phenomenon using a workflow management tool in bioinformatics has been actively researched. Workflow management tool is the base which enables researchers to collaborate through the re-use and sharing of service, and a variety of workflow management tools including MyGrid project's Taverna, Kepler and BioWMS have been developed and used as the open source. This workflow management tool can model and automate different services in spatially-distant area in one working space based on the web service technology. Many tools and databases used in the bioinformatics are provided in the web services form and are used in the workflow management tool. In such the situation, the web services development and stable service offering for a sequence similarity search which is basically used in the bioinformatics can be essential in the bioinformatics field. In this paper, the similarity retrieval speed of biology sequence data was improved based on a Linux cluster, and the sequence similarity retrieval could be done for a short time by linking with the workflow management tool through developing it in the web services.

▸ Keyword : 웹서비스(webservices), soap, xml, 생물정보학(Bioinformatics), 서열검색(Sequence Search)

• 제1저자 : 정진영
• 접수일 : 2008. 8. 27, 심사일 : 2008. 10. 30, 심사완료일 : 2008. 11. 26.
* 대전보건대학 바이오정보과 조교수

Ⅰ. 서 론

생물정보학은 대규모 생물학 연구로부터 얻어진 대량의 데이터를 가치 있는 정보로 만들어내는 연구 분야로 고전적인 실험 방법을 통해 얻어지는 연구 결과에 비해서 많은 분량의 결과를 짧은 시간에 얻을 수 있기 때문에 그 중요성이 커지고 있다. 이와 같은 대량의 생물학 데이터를 처리하기 위해서는 고성능의 컴퓨터의 사용이 필수적이다. 대량의 데이터를 고속으로 처리하기 위한 컴퓨팅 자원은 모든 연구자가 갖추기는 어려울 뿐만 아니라 분산된 각각의 생물학 도구 및 데이터에 접근하기 위해서 생물정보학 연구에서 최근 워크플로우 관리 도구의 사용이 점차 증가되고 있다. 워크플로우 관리 도구는 생명정보 연구과정을 모델링하고 자동화하기 위한 시스템으로 영국의 MyGrid 프로젝트를 통해 생물정보학 분야에 특화된 Taverna라는 워크플로우 관리 도구를 비롯하여 Kepler, Biopipe, Bioworks등의 수많은 오픈소스 기반의 워크플로우 관리 도구들이 있다[1,2,3,4]. 이러한 워크플로우 관리도구들은 서로 공간적으로 분리되어 있는 여러 생물정보학 도구와 데이터를 웹 서비스 기술을 통하여 하나의 작업공간에서 수행할 수 있도록 하고 있을 뿐만 아니라 워크플로우를 공유하고 원격에서 실행함으로써 연구자들이 서로 협업 할 수 있는 기반을 제공하고 있다[5]. 따라서 기존에 클라이언트/서버 또는 로컬 환경, 웹 인터페이스를 사용하는 생물정보학 도구들에 대한 웹 서비스 형태의 제공은 매우 중요하다.

서열유사성 검색은 생물정보학에서 가장 중요하고 기초적인 작업으로 아미노산이나 염기서열에 대한 서열의 유사성 검색 작업을 통해 상동성을 찾아내는 작업은 유전자의 기능을 예측하거나 단백질의 2차 구조의 예측 및 분자 모델링 디자인 연구에 기반이 된다. 이러한 서열 검색 작업에서 동적프로그래밍 알고리즘은 최선의 정렬을 찾기 위한 모든 가능성을 결정하기 때문에 항상 최적의 결과를 얻을 수 있다. 그러나 많은 결정에 대한 순서들이 생성되기 때문에 이를 구현하기 위해서는 많은 메모리와 시간을 필요로 한다[6]. 따라서 휴리스틱 접근 알고리즘을 사용하여 결과의 정확성 대신 속도를 얻는 BLAST(Basic Local Alignment Search Tool) 알고리즘이 서열 검색에 널리 사용되고 있다[8]. BLAST 알고리즘의 특징은 첫째, 알려지지 않은 서열에서 유용한 정보를 확인하는데 강력한 도구라는 점과 둘째, 빠르게 작동 한다는 점 셋째, 통계적 입장과 소프트웨어 개발 관점 두 가지에서 정확하다는 점과 마지막으로 많은 서열 분석 시나리오를 유연하게 적용할 수 있다는 것이다[7].

생물학에서의 서열에 대한 시퀀싱(Sequencing)의 자동화와 컴퓨터 프로그래밍의 도입으로 서열 데이터는 매우 크고 그 성장 속도 또한 빠르게 이루어지고 있는 시점에서 서열 검색에 대한 속도는 매우 중요한 요소로 대두되게 되었다. 대규모의 서열 검색 작업에 대한 병렬화를 통한 검색 작업의 단축[8]과 이를 웹 서비스 형태로의 제공은 워크플로우 관리 도구의 사용에 있어 필수적이라 할 수 있다. 본 논문에서는 리눅스 기반의 클러스터 환경에서 질의 서열을 분할하여 BLAST를 병렬 수행하는 웹 서비스를 개발하였다. BLAST 웹 서비스는 표준의 HTTP를 통해 접근하여 생물학 워크플로우 관리 도구와 연동 할 수 있도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 생물학 데이터의 통합과 분석 프로세스의 자동화 및 워크플로우 관리 시스템에 대해 알아보고, 생물학 서열검색과 이를 병렬화하는 방법에 대해 알아본다. 3장에서는 본 논문에서 제안하는 질의 분할 방법을 통한 병렬화 알고리즘을 웹 서비스로 제공하기 위한 시스템을 디자인한다. 4장에서는 BLAST 웹 서비스를 위한 WSDL을 기술하고 웹 서비스 형태로 구현한다. 5장에서는 실제 워크플로우 관리 도구에서 생물학 도구와 BLAST 웹 서비스를 서로 연결하는 시나리오를 작성하고, 6장에서는 결론을 맺는다.

Ⅱ. 관련 연구

본 장에서는 생물학 데이터의 통합과 분석 프로세스의 자동화 및 생물학 서열 유사성 검색에 대해서 살펴본다.

2.1 생물학 데이터의 통합과 XML

2.1.1 XML을 통한 생물학 데이터 통합

생물학 연구 결과로 지속적인 새로운 데이터와 데이터 형태가 나타나고 있다. 이러한 생물학 데이터는 마이크로어레이와 같은 high-throughput 기술로 인해 그 크기 또한 지속적으로 증가하고 있는 추세이다. 그러나 이러한 생물학 데이터는 공급자들 간의 상이한 데이터베이스 구조, 포맷 형식, 데이터를 정의 하는 용어의 차이와 서로 다른 컴퓨터 프로그래밍 언어를 사용하는 등의 문제가 발생한다. 이러한 이질적인 생물학 데이터를 분석하기 위한 생물학 데이터와 분석 소프트웨어와의 연동 또한 필수적이 되었다[9].

생물 정보의 연동을 위한 생물학 데이터 표준화 방안은 생물학 데이터 대한 XML 스키마를 통한 모델, XML을 기반으로 하는 데이터 표현과 저장, 데이터 교환과 연동을 위한 웹

서비스, 생물학적 분석 프로세스를 실행하고 정의하기 위한 워크플로 시스템을 필요로 한다. (표 1)은 생물학 데이터에 대해서 XML을 이용한 표현 방법으로, 서열 및 단백질 데이터 등의 생물학 데이터를 XML로 표현하는 방법과 데이터 처리 결과물에 대한 XML의 표현 방법 등 여러 가지 형태로 XML이 사용되고 있다.

표 1. 생물학 XML 데이터
Table. 1 XML for Biology

데이터	XML 표현
Sequence	BSML, Agave
Proteins	SPML
NCBI ouputs	BlastXML
Microarray	MAGE-ML
System Biology Markup Language	SBML
Biological Variation Markup Language	BVML

2.1.2 생물학 웹 서비스

웹 서비스는 인터넷 상에서 표준을 이용하여 서로 다른 시스템 상의 어플리케이션들 간에 데이터 교환과 공유를 제공하는 서비스 통합 기술이다. 웹 서비스는 상이한 플랫폼, 운영체제, 프로그래밍 언어, 데이터베이스 간의 프로그램들이 표준 기반으로 서로 통신할 수 있도록 상호운용성을 보장해 준다. 웹 서비스는 UDDI(Universal Description, Discovery and Integration), WSDL(Web Service Description Language), SOAP(Simple Object Access Protocol)로 이루어져 서비스 공급자와 요청자 사이에 인터넷 표준 통신 프로토콜인 HTTP를 이용하여 XML 문서의 형태로 정보를 교환한다. 서비스 요청자는 UDDI를 통해 서비스 검색한 후 해당 서비스의 위치정보를 가져오게 되며, 서비스 공급자는 웹 서비스 작성, UDDI에 WSDL 형태로 서비스를 등록하게 된다. 서비스 요청자는 작성된 서비스 명세인 WSDL을 가져와 SOAP을 통해 작성된 웹 서비스를 실행하게 된다. (표2)는 대표적인 생물학 웹 서비스 제공기관과 제공 서비스를 보여준다[9].

표 2. 생물학 웹 서비스
Table. 2 Web Services for Biology

웹 서비스	제공기관 및 제공형태
EMBOSS, XEMBL, Interpro	EBI
eUtils	NCBI
caBIO	NCICB
KEGG API	DDBJ
bioMOBY	directory
Soaplab	tools

2.2 생물학 프로세스의 자동화와 표준화

워크플로우 관리시스템은 소프트웨어 라이브러리, 독립 시스템, 웹 인터페이스의 3가지 형태로 제공되고 있다. 이러한 워크플로우 관리 시스템은 서로 다른 언어를 사용하고 있으며, WfMC, W3C등의 서로 다른 기관의 표준을 수용하고 있다[1,2]. 이러한 생물학 워크플로우 관리시스템은 (표3)과 같다.

표 3. 워크플로우 관리 시스템
Table. 3 Workflow Management System

소프트웨어	형태	XML	배포형태
Taverna Workbench	Stand-alone	XScufl	Open source
Kepler	Stand-alone	MoML	Open source
BioWMS	Web Interface, remote services	XPDL	Public use
Biowep	portal	XScufl, XPDL	Open source
Pegasys	Stand-alone	Pegasys DAG	Open source
Wildfire	Stand-alone	GEL	Open source
Triana	Stand-alone	Triana WL	Open source
FreeFluo	Library	WSFL & XScufl	Open source
Biomake	Library	N/A	Open source

2.3 BLAST 알고리즘

BLAST는 공개된 서열 데이터베이스 내에서 분석하고자 하는 서열과 비교를 통해 유사성을 지닌 아미노산이나 염기서열을 찾고 그에 대한 기능적 특징을 알아내는데 목적이 있다. NCBI BLAST는 blastall이라는 하나의 실행파일을 통해 blastn, blastp, blastx, tblastn, tblastx의 서로 다른 다섯 가지 프로그램을 하나의 인터페이스를 이용하여 수행할 수 있다. 이러한 NCBI BLAST 프로그램은 (표 4)와 같이 검색하려는 데이터베이스와 질의 서열의 성질에 따라 나누어진다.

표 4. BLAST 프로그램
Table. 4 Traditional BLAST Programs

Program	Database	Query
blastn	Nucleotide	Nucleotide
blastp	Protein	Protein
blastx	Protein	Nucleotide translated into protein
tblastn	Nucleotide translated into protein	Protein
tblastx	Nucleotide translated into protein	Nucleotide translated into protein

BLAST 알고리즘은 (그림 1)과 같이 검색을 원하는 질의 서열로부터 3개의 아미노산이나 11개의 염기로 이루어진 짧은 서열(words)을 만든다. 이때 만들어지는 짧은 서열은 식(1)을 통해 구할 수 있다. 예를 들어 500-base의 아미노산의 경우 총 498개의 짧은 서열을 만들 수 있다.

$$Maximum\ words = L - w + 1$$

(L = 질의 서열의 총 길이, w = 3(아미노산), 11(염기)) ..... 식(1)

생성된 짧은 서열은 두 아미노산의 유사성을 나타내는 매트릭스인 BLOSUM(Blocks Amino Acid Substitution Matrices)62를 사용하여 경계 값(T) 이상의 점수를 기록하는 모든 목록을 구성한다. BLOSUM62는 한 아미노산이 다른 아미노산으로 바뀔 가능성 즉 두 아미노산의 유사성을 나타낸다. BLOSUM62 점수행렬에서 양수는 두 아미노산이 서로 잘 바뀔 수 있는 경우를 의미하며, 음수는 두 아미노산이 서로 잘 바뀌지 않는 경우를 의미한다. 0은 특별한 의미 없이 두 아미노산이 우연히 바뀔 수 있는 경우를 의미한다. 두 서열을 비교해서 점수가 높으면 친족관계(homology)가 더 있다고 볼 수 있다[10].

경계 값 이상의 짧은 서열 목록과 서열 데이터베이스에서 일치하는 서열을 찾아 서열 데이터베이스에서 양쪽 방향으로 갭이 없는 로컬 정렬 방식으로 확장하게 된다. 확장을 마친 후 서열 데이터베이스의 서열 중 일정 값 이상의 HSP(High-scoring Segment Pair)를 가진 서열들을 추출한다. 이때 중복되지 않는 각각의 HSP들은 통계적인 테스트를 거쳐 연결되어 최종 결과를 생성한다[7].

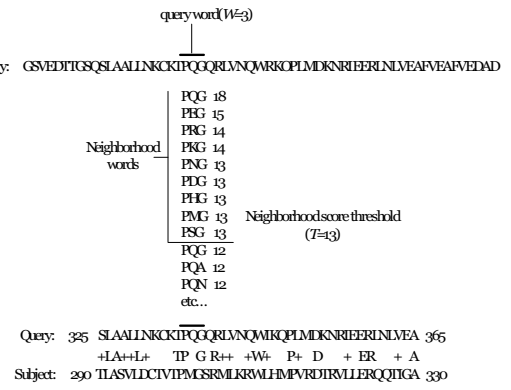

그림 1. BLAST 검색 알고리즘
Fig. 1 BLAST Search Algorithm

2.4 BLAST의 병렬화

본 절에서는 BLAST에 대한 속도 향상을 위한 병렬화 방법을 살펴본다.

2.4.1 하드웨어 병렬화

하드웨어 가속 기능을 통한 BLAST의 속도 향상은 R.K. Singh[11]에 의해 처음 보고 되었으며, TimeLogic 사의 DeCyper 악셀레이터는 BLAST의 최적화를 위해서 BLAST 알고리즘을 FPGA(Field Programmable Gate Array)에 이식시켜 이를 컴퓨터에 장착하여 사용한다. DeCypher 엑셀레이터는 FPGA 보드에 최적화한 Tera-BLAST를 수행하여 BLAST 알고리즘을 실행한다[12]. 이러한 하드웨어 기반의 BLAST의 속도 향상은 많은 양의 BLAST 검색 시 유용하게 사용되며 클러스터 보다 저렴한 비용으로 구축할 수 있다는 장점이 있지만, BLAST 작업에만 특화되어 범용성이 떨어진다는 단점이 있다.

2.4.2 질의 분할을 통한 병렬화

BLAST의 질의 분할을 통한 병렬화는 클러스터의 각 노드 또는 SMP 장비의 CPU별로 질의 서열을 분할하여 실행하는 방법으로 서로 다른 질의를 병렬로 수행할 수 있다. 이 방법은 각 노드의 로컬 디스크나 원격의 공유디스크에 존재하는 데이터베이스에 대하여 분할된 질의를 노드가 개별적으로 수행하게 된다[13].

2.4.3 데이터베이스 분할을 통한 병렬화

BLAST의 데이터베이스 분할을 통한 병렬화는 각 프로세서나 클러스터의 노드 단위로 서열 데이터베이스를 독립적으

로 분할하여 검색하는 방법으로 이렇게 함으로써 데이터베이스에서 요구하는 메모리양을 줄일 수 있다. 이러한 구조는 중복성이 배제됨으로써 데이터베이스 양 증가에 따라 컴퓨팅 파워를 추가할 수 있는 장점이 있다. 이러한 데이터베이스를 분할 방법은 TurboWorx사의 TruboBLAST가 있다. TurboBLAST는 워크스테이션과 네트웍을 통한 데이터베이스 분할과 분산 실행을 수행하며, 이때 TurboHub를 통하여 스케줄링과 로드밸런싱을 수행하며 클러스터 환경에 적용하여 사용할 수 있다[14].

데이터베이스 분할의 또 다른 구현은 parallelblast로 Sun Grid Engine/PVM 환경과 몇 개의 스크립트로 구성되어 있다. 그러나 SGE/PVM 환경을 필요로 한다는 단점이 있다[15].

mpiBLAST는 BLAST 알고리즘 자체는 수정하지 않고, 프로세스 제어와 생성, 데이터 통신에만 수정을 가한 BLAST 프로그램이다. mpiBLAST는 데이터베이스가 분할되어 공유 저장장치에 저장되는 단계와 mpiBLAST 질의가 각 계산 노드에서 검색되는 두 단계로 이루어진다. mpiBLAST의 검색 단계는 검색에 필요한 데이터베이스 파티션 분할 단계와 실제 검색 단계, 그리고 결과 취합 단계로 다시 나뉘어 진다. 즉, 주어진 사용자 질의에 대해 필요한 데이터베이스 파티션을 각 계산 노드로 복사하고, 복사된 각 파티션 내에서 서열의 유사성 검색을 수행한 후 각각의 파티션에 대한 검색 결과를 취합하는 과정을 거친다[16].

III. 병렬 서열 검색 웹 서비스 디자인

본 장에서는 병렬 서열 검색을 위한 BLAST를 병렬화 알고리즘과 병렬화된 BLAST의 웹 서비스에 대해서 살펴본다.

3.1 질의 분할 병렬 알고리즘

병렬 서열 검색을 위한 BLAST 병렬화는 마스터 노드와 계산 노드의 두 부분으로 구분된다. 마스터 노드는 입력된 질의를 분할하고 노드 정보를 통해 얻어진 값을 기반으로 질의를 수행할 노드를 선별하는 과정과 계산 노드에서 실행된 결과를 취합하는 과정을 담당한다. 계산 노드는 마스터 노드로부터 전달된 질의를 수행하고 마스터 노드에 노드 정보 및 질의 결과를 전달한다. (그림 2)는 BLAST 병렬화 알고리즘으로 MPI 통신자의 선언으로 시작해서 MPI 통신자를 해제함으로써 종료된다. 마스터 노드의 작업은 BLAST 작업을 수행할 수 있는 계산 노드가 존재한다는 조건 하에서 시작된다. 계산 노드가 존재한다면 각 계산 노드에 전달할 질의를 분할하게 된다. 여기서 질의 분할 작업은 모든 클러스터 노드에서 진행되는 것이 아니라 마스터 노드에서 이루어지게 된다.

MPI를 초기화하는 과정에서 클러스터간의 메시지 송/수신을 위한 통신자를 설정하게 된다. 통신자가 0이라면 이는 작업을 수행하는 마스터 노드를 가리키기 때문에, 통신자가 0인 마스터 노드에서만 분할 작업을 수행하도록 한다. 이 모든 작업은 계산 노드가 존재 할 때 시작하도록 하여 네트워크의 경쟁 등으로 인한 성능 감소를 사전에 차단한다. 분할 작업 후 생성된 질의는 모든 노드가 질의를 할당 받을 때 까지 계산 노드들에게 질의를 할당한다. 질의를 할당 받은 계산 노드는 질의를 수행한 계산 노드 집합에 해당 노드를 추가 시켜 더 이상 질의를 할당 받지 않도록 한다. 질의가 계산 노드에게 할당이 되면 계산 노드들은 BLAST 작업을 수행한다.

3.2 병렬 서열 검색 웹 서비스

병렬화된 BLAST는 로컬 서비스만이 가능하기 때문에 이를 웹 서비스 형태로 제공되어야 워크플로우 관리 도구에서 사용이 가능하다. BLAST를 웹 서비스로 제공하기 위해서는 네트워크에서의 서비스 품질과 유효성 및 접근 제한에 대한 문제가 해결되어야 한다. 또한 서비스 작성에 있어서 생물학 데이터 처리에 걸리는 긴 시간에 대한 고려와 함께 대용량 데이터의 I/O에 대한 고려가 이루어져야 한다. 워크플로우 관리 시스템은 사용자 인터페이스, 데이터 재사용 및 데이터 캐싱과 이질적인 웹 서비스와 웹 서비스 I/O의 복잡성 문제 및 이질적인 데이터에 대한 포맷 변환을 위한 어댑터에 대한 고려가 이루어져야 한다. 이러한 문제를 해결하기 위해 본 논문에서는 병렬 BLAST의 웹 서비스에 대해서 Job Name을 제공하고 웹 서비스 사용자가 실행 상황을 주어진 Job Name을 통해서 모니터링 할 수 있도록 하여 사용자의 웹 서비스의 실행 과정에서 디버깅이 가능하도록 하였다.

```
질의 서열 집합 Q = {q1; q2; q3...}
할당되지 않은 질의 Unassigned
질의가 수행되고 있지 않은 노드 Unsearched
클러스터 계산 노드 집합 W = {w1; w2; w3...}
질의를 할당받은 계산 노드의 집합 F = {F1; F2; F3...}
통신자에서 자신의 순번(마스터 프로세스 0) iproc
통신자에서 모든 노드의  합 nproc

include MPI Library // MPI 함수
MPI_Init //MPI 통신자를 선언한다.
if |W| ≠ 0 //계산 노드가 존재해야만 한다.
    if iproc = 0 //마스터 노드인 경우
        Split query and save //질의 분할을 수행한다.

    while |Unassigned| ≠ 0 do //모든 노드에게 질의가 할당 될 때까지
        Send a query qi to a worker wj //질의를 계산 노드에 할당한다.
        Remove qi from Unassigned
```

```
    //질의를 할당되지 않은 질의에서 제거한다.
    Add qi to Fj //질의를 할당받은 계산 노드 집합에 추가한다.
end while

Receive Broadcast message from workers
//계산 노드로부터 질의를 할당 받았다는 메시지를 수신한다.

while |Unsearched| ≠ 0 do
//모든 계산 노드가 질의를 수행할 때까지
    Receive a message from worker wi
    //계산 노드로부터 질의 결과를 수신한다.
    if message is SEARCH_COMPLETE
    //계산 종료 메시지를 수신하면
        Receive result //질의 결과를 수신 받는다.
        Remove Fi from Unsearched
        // 질의가 수행되지 않는 노드로 환원한다.
    end while

if iproc = 0 //마스터 노드인 경우
    Merge the results  //결과 취합
end if

PI_Finalize //MPI 통신자 해제
```

그림 2 BLAST 병렬화 알고리즘
Fig. 2. Algorithm of Parallel BLAST

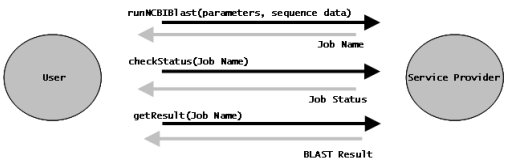

그림 3. 병렬 BLAST 웹 서비스의 메소드와 메시지 흐름
Fig. 3 BLAST Web Services Method and Message Flow

(그림3)은 병렬 BLAST의 웹 서비스에서 메시지 흐름을 보여준다. 사용자가 runNCBIBlast() 메소드를 통해서 BLAST를 호출하면 해당 작업에 대한 Job Name을 사용자에게 반환한다. 이것으로 병렬 BLAST에 대한 작업의 호출은 마치게 되며, 향후 Job Name을 통해 해당 작업에 대한 수행 상태와 수행 결과를 checkStatus() 메소드와 getResult() 메소드를 통해서 얻을 수 있다. 따라서 사용자는 자신이 호출한 BLAST 작업에 대한 디버깅이 가능하도록 하여 생물학 데이터 처리에 걸리는 긴 시간에 대한 웹 서비스가 가능하도록 하였다. 병렬 BLAST의 웹 서비스에서 제공하는 메소드는 총 5개로 (표 5)와 같다. 5개의 메소드 중에서 findJobId() 메소드는 사용자가 제출한 작업에 대한 Job Id를 찾는 메소드로 사용자가 직접 호출하여 사용하지 않는 메소드로 병렬 BLAST 내부 웹 서비스 코드에서 호출하여 사용한다. 따라서 findJobId() 메소드를 제외한 4개의 메소드를 사용자가 사용할 수 있다.

표 5. 병렬 BLAST 웹 서비스의 사용 가능한 메소드
Table. 5 BLAST Web Services Methods

메소드	설명
runNCBIBlast	BLAST를 호출하는 메소드로 BLAST에서 사용하는 다양한 옵션과 검색 서열을 입력으로 받는다.
findJobId	사용자가 제출한 작업에 대한 Job ID를 찾는 메소드로 웹 서비스의 디버깅 작업에 사용한다.
checkSequence	사용자가 제출한 서열이 유효한 서열인지를 확인한다. (유효한 서열 True, 유효하지 않은 서열 False)
checkStatus	사용자가 제출한 BLAST 작업의 진행 상황을 확인한다.(종료되면 True, 실행중이면 False)
getResult	해당 Job ID에 대한 서열 유사성 검색 결과를 반환한다.

병렬 BLAST 웹 서비스와 병렬화된 BLAST는 (그림 4)와 같은 구조로 되어 있다. BLAST 웹 서비스는 병렬 BLAST를 호출하고 그에 대한 Job Name을 사용자에게 반환한다. 병렬 BLAST는 각 클러스터 노드들에 작업을 할당하게 되고, 병렬 BLAST 작업이 종료되면 그 결과와 수행 상태를 로컬에 저장하게 된다. 사용자가 BLAST 웹 서비스를 통해 해당 작업에 대한 Job Name을 통해 작업 상태나 결과를 호출하면 BLAST 웹 서비스는 해당 Job Name을 통해 해당 작업에 대한 상태 및 결과 정보를 사용자에게 웹 서비스를 통해 반환한다.

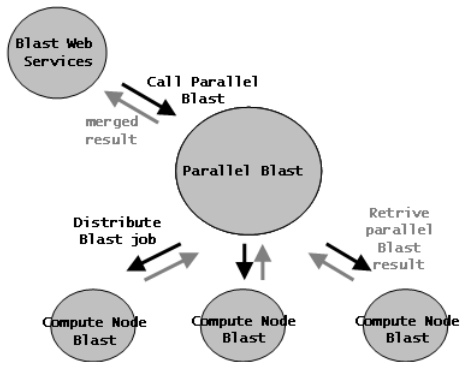

그림 4. Parallel Blast와 웹 서비스 구성
Fig. 4 BLAST and Web Service Architecture

Ⅳ. 구현

BLAST의 병렬화 버전은 리눅스 클러스터로 구성된 컴퓨팅 노드에서 수행하며, 웹 서비스는 별도의 웹 서버에서 제공된다. 각각의 시스템 자원은 (표 6)과 같다.

표 6. 웹 서비스 시스템 자원
Table. 6 Specification of Web services System

분류	웹 서비스 제공 서버	Blast 컴퓨팅 노드
OS	RedHat Linux 7.3	RedHat Linux 7.3
CPU	Intel PIV	Intel PIV
Memory	1GB	1GB
Software	Tomcat Java(JAX-WS)	MPI

웹 서비스를 제공하는 서버는 RedHat Linux를 기반으로 Tomcat과 JAX-WS를 통해 웹 서비스를 제공한다. 병렬 BLAST를 수행하는 리눅스 클러스터는 RedHat Linux 기반의 운영체제를 사용하고 있으며, 총 28노드의 계산노드로 100Mbps의 네트워크 대역폭을 가지고 있다. 스토리지 노드는 모든 계산노드에 NFS로 공유된 마운트 포인트를 제공한다. BLAST의 바이너리 파일과 질의 파일들은 MPI 환경에서 수행되기 때문에 마스터 노드와 계산 노드의 공유 디스크에 위치한다.

BLAST 데이터베이스의 저장 위치에 대해서는 공유디스크나 계산 노드의 로컬 디스크에 저장할 수 있다. 공유디스크를 이용할 경우 네트워크를 통한 계산 노드간의 부하가 생기지만, 데이터베이스의 업데이트 등의 관리적인 측면에서는 유용하다. 데이터베이스가 각 계산 노드에 존재할 경우 데이터베이스 업데이트 시 동기화의 과정을 필요로 한다는 단점이 있지만 각 계산 노드로 데이터베이스를 로드 하는데 걸리는 네트워크 부하를 줄일 수 있기 때문에 본 논문에서는 각 계산 노드의 로컬 디스크에 BLAST 데이터베이스를 위치시켰다.

Ⅴ. 성능 분석

본 장에서는 워크플로우 관리 도구를 이용하여 웹 서비스를 호출하여 병렬화된 BLAST를 호출하고 타 생물학 웹 서비스와 연동하도록 하겠다.

5.1 웹 서비스 등록

생물학 워크플로우 관리 도구인 Taverna를 통해 병렬화된 BLAST 웹 서비스를 추가 한다. Taverna는 Add new WSDL scavenger를 통해 외부에서 제공하는 웹 서비스를 추가할 수 있도록 하고 있다. (그림 5)의 웹 서비스의 WSDL 문서를 Taverna의 서비스에 추가한다.

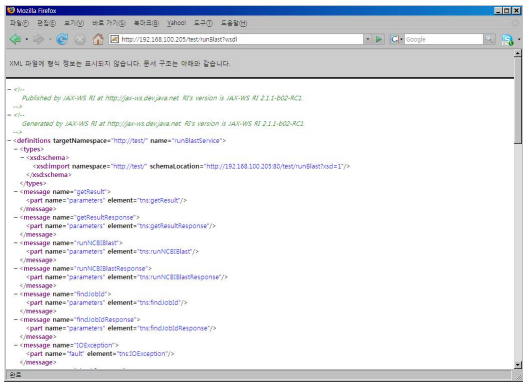

그림 5. BLAST 웹 서비스 WSDL 문서
Fig. 5 WSDL for BLAST Web Service

WSDL을 추가하면 해당 웹 서비스에서 사용가능한 메소드의 목록이 나타나게 된다. (그림 6)은 Taverna에 등록된 웹 서비스의 메소드 목록을 보여준다.

그림 6. 추가된 BLAST 웹 서비스
Fig. 6 BLAST Web Service in Taverna

5.2 워크플로우 작성

EBI(European Bioinformatics Institute)에서는 생물학 데이터베이스로부터 정보를 가져오는 WSDbfetch 웹 서비스를 제공하고 있다. WSDbfetch 서비스는 Nucleotide 서열, Protein 서열, Protein 구조 등의 데이베이스로부터 해당 정보를 가져 온다. WSDbfetch는 http://www.ebi.ac.uk/Tools/ webservices/ws이/WSDbfetch.wsdl에 WSDL 문서가 위치하고 있으며, 이를 통해 서비스를 등록할 수 있다. 본 논문에서는 단백질에 대한 서열 및

기능에 대한 주석을 포함하고 있는 UniProt 데이터베이스에 대해서 SLPI_MOUSE Antileukoproteinase precursor(ALP)에 대한 서열을 가져온다.

가져온 서열에 대해서는 994,617개의 서열로 구성된 non-redundant(nr) 단백질 데이터베이스와 병렬 BLAST 웹 서비스를 통해 서열의 유사성 검색을 수행한다. runNCBIBlast에 WSDbfetch를 통해 가져온 ALP 서열을 입력으로 주고 checkStatus를 통해 해당 작업의 진행 상황을 모니터링 하다가 작업이 완료되면 getResult를 통해 결과를 가져온다. (그림 7)은 작성된 워크플로우로서, 해당 워크플로우를 실행하면 자동으로 일련의 작업을 수행하고 그 결과를 반환한다.

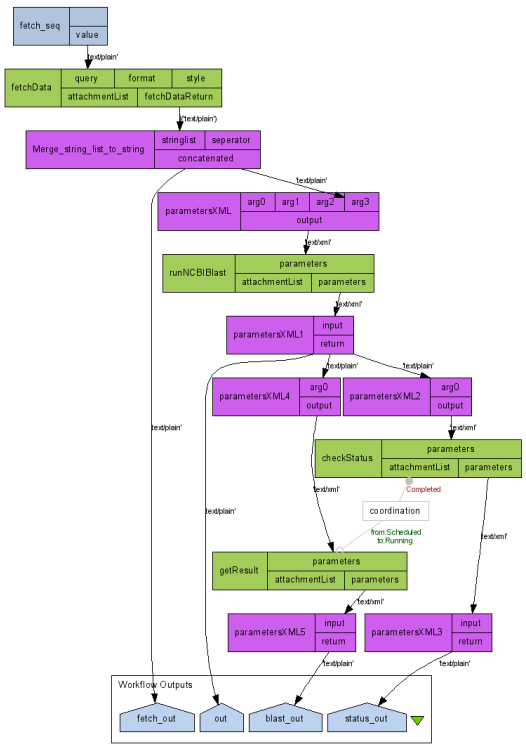

그림 7. Taverna를 이용한 BLAST 워크플로우
Fig. 7 Workflow for BLAST using Taverna

5.3 서열 유사성 검색 결과

워크플로우를 실행 결과는 (그림 8)과 같으며, 로컬 Blast를 수행한 결과와 동일한 결과를 보여준다.

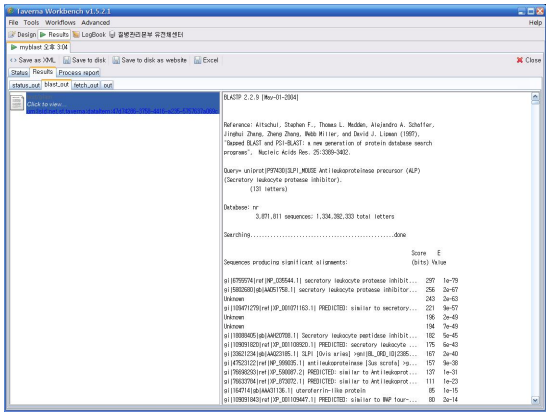

그림 8. BLAST 웹 서비스 실행 결과
Fig. 8 Result for BLAST Web Services

Ⅵ. 결론 및 향후 연구

본 논문에서는 대규모 생물학 서열 데이터를 분석하기 위한 서열 유사성 검색 도구인 BLAST를 병렬화하고 이를 웹 서비스를 통해 제공함으로써 워크플로우 관리 도구와 손쉽게 연동되도록 하였다. 서열 검색에 있어서 질의 서열을 분할하고 이를 리눅스 클러스터상에서 작업을 수행하도록 하였다. 이는 별도의 소프트웨어적인 작업 없이 리눅스 클러스터에서 로컬 BLAST를 사용 가능하다는 장점이 있다. 병렬 서열 분석 웹 서비스는 Taverna 워크플로우 관리 도구와 타 웹 서비스와 결합하여 일련의 생명현상 연구 과정을 워크플로우화하여 자동으로 수행할 수 있도록 해준다.

향후 서열 유사성 검색뿐만 아니라, 구조 분석이나, 텍스트 마이닝 작업등 대용량의 데이터와 대규모 컴퓨팅파워를 필요로 하는 생물학 작업에 대한 웹 서비스 제공 연구가 이루어져야 한다.

참고문헌

〔1〕 Oinn T, Addis M, Ferris J, Marvin D, Senger M, Greenwood M, Carver T, Glover K, Pocock MR, Wipat A, Li P: Taverrna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics 2004, 20(17):3045-54.

〔2〕 Altintas, I., Barney, O., & Jaeger-Frank, E. Provenance collection support in the Kepler Scientific Workflow System. In International

Provenance and Annotation Workshop (IPAW), LNCS, Provenance and Annotation of Data, 4145: 118-132, 2006.
〔3〕 Bio Community, http://biocommunity.kr/
〔4〕 Bioworks, http://bioworks.kisti.re.kr/
〔5〕 David De Roure, Carole Goble and Robert Stevens. Designing the myExperiment Virtual Research Environment for the Social Sharing of Workflows. e-Science 2007 - Third IEEE International Conference on e-Science and Grid Computing, 2007. Bangalore, India, 10-13 December 2007. 603-610.
〔6〕 T.R. Smith and M.S. Waterman. Identification of common molecular subsequences. J. Mol. Bio1., 195-197. 1981.
〔7〕 S.Altschul, W.gish, W. Miller, E.Myers, and D. Lipman. Basic local alignment search tool. Journal of Molecular Biology, 215:403,1990.
〔8〕 홍창범, 클러스터 환경에서의 MPI 기반 병렬 서열 유사성 검색에 관한 연구. 한국컴퓨터정보학회, 69-78. 2006.
〔9〕 H Sugawara, S Miyazaki. Biological SOAP servers and web services provided by the public sequence data bank. Nucleic Acids Research, 3836-3839. 2003.
〔10〕 S. Henikoff and J.G. Henikoff. Amino acid substitution matrices from protein block. Proceedings of National Academy of Science, 89:10915-10919,1992.
〔11〕 R. K. Sin호, W. D. Dettloff, V. L. Chi, D. L. Hoffman, S. G. Tell, C. T. White, S. F. Altschul, and B. W. Erickson, BioSCAN: A Dynamically Reconfigurable Systolic Array for Biosequence Analysis, Research on Integrated System, 1993.
〔12〕 TimeLogic. Adaptable hardware acclerated systems for bioinformatics. Technical report, TimeLogic, 2002
〔13〕 N. Camp, H. Cofer, and R. Gomperts. High-throughput BLAST, September 1998.
〔14〕 R. D. Bjorson, A. H. Sherman, Weston, N. Willard, and J. Wing. TurboBLAST: A parallel implementation of BLAST based on the TurboHub process integration architecture. Technical report, TurboGenomics, Inc, 2002.
〔15〕 parallelblast, ftp://saf.bio.caltech.edu
〔16〕 A. Darling, L. Carey, and W. Feng. The Design, Implementation and Evaluation of mpiBLAST(Best Paper: Applications Track) ClusterWorld Conference & Expo in conjunction with the 4th Internationl Conference on Linux Clusters: The HPC Revolution 2003, San Jose, CA, June 2003.

저 자 소 개

정진영
2002년8월 한남대학교 컴퓨터공학과 공학박사
1997년 3월~ 현재 대전보건대학 바이오정보과 교수
관심분야 : 운영체제, 웹기반정보시스템, 생물정보학, 정보검색