ISSN(Print): 2287-5905

http://dx.doi.org/10.3745/KTSDE.2013.2.2.097

Workflow-based Bio Data Analysis System for HPC

Shinyoung Ahn† · ByoungSeobKim† · Hyun-Hwa Choi† · Seunghyub Jeon† · Seungjo Bae†† · Wan Choi†††

ABSTRACT

Since human genome project finished, the cost for human genome analysis has decreased very rapidly. This results in the sharp increase of human genome data to be analyzed. As the need for fast analysis of very large bio data such as human genome increases, non IT researchers such as biologists should be able to execute fast and effectively many kinds of bio applications, which have a variety of characteristics, under HPC environment. To accomplish this purpose, a biologist need to define a sequence of bio applications as workflow easily because generally bio applications should be combined and executed in some order. This bio workflow should be executed in the form of distributed and parallel computing by allocating computing resources efficiently under HPC cluster system. Through this kind of job, we can expect better performance and fast response time of very large bio data analysis. This paper proposes a workflow-based data analysis system specialized for bio applications. Using this system, non-IT scientists and researchers can analyze very large bio data easily under HPC environment.

Keywords : HPC, Genome Analysis, Bio-Informatics, WMS, RMS, Supercomputer, MAHA Supercomputer

HPC 환경을 위한 워크플로우 기반의 바이오 데이터 분석 시스템

안 신 영† · 김 병 섭† · 최 현 화† · 전 승 협† · 배 승 조†† · 최 완†††

요 약

인간 게놈 프로젝트의 완성 이후 유전체 분석 비용은 매우 빠르게 감소하고 있다. 이에 따라 인간 유전체 분석 요구가 급증할 것으로 예상된다. 인간 유전체 분석과 같은 대규모 바이오 데이터 분석을 고속으로 수행하기 위해서는 비IT 전문가들이 다양한 특성의 바이오 응용들을 고성능컴퓨팅 시스템을 통해 효과적으로 실행할 수 있어야 한다. 이를 위해서는 여러 응용들이 조합되어 순서를 갖고 실행되어야 하는 바이오 응용들을 워크플로우 형태로 쉽게 정의할 수 있어야 하며, 이 워크플로우를 HPC 클러스터 시스템에서 최적 자원을 할당 받아 분산 병렬 수행시켜야 한다. 이를 통해 바이오 데이터 분석 성능과 응답시간의 개선을 기대할 수 있다. 본 논문에서는 HPC 환경에 익숙하지 않은 비IT 바이오 연구자들이 쉽게 바이오 데이터 분석을 할 수 있도록 바이오 워크플로우를 쉽게 정의하고 실행할 수 있는 바이오 특화된 워크플로우 기반 대규모 데이터 분석 시스템을 제안한다.

키워드 : 고성능 컴퓨팅, 유전체 분석, 바이오 데이터 분석, 워크플로우 관리, 자원 관리, Supercomputer, MAHA Supercomputer

1. 서 론

전체 인간 유전자 지도는 1993년 시작되어 2003년 완성된 인간 게놈 프로젝트를 통해 처음으로 밝혀졌다. 이 프로젝트에는 총 30억 달러가 소요되었는데[1], 이후로 유전체 분석 비용은 매우 빠르게 줄어들고 있다. 2012년 현재 한 사람의 인간 유전체 분석 비용은 일주일에 수천$ 수준으로 서비스가 되고 있으며[2] 수 년 후에는 수 시간에 100$ 유전체 분석 시대가 열릴 것으로 기대되고 있다.

1인당 유전체 분석 시간 및 비용의 감소에 따라 더 많은 사람들의 유전체 분석 요구가 기하급수적으로 증가할 것으로 예상된다. 유전체 서열 분석뿐 만 아니라 단백질 구조 분석 등 관련된 바이오 데이터의 총량은 급속도로 증가하고 있다.

유전체 분석은 유전체(게놈)을 물리/화학적으로 조각낸 후 광학적으로 A(아데닌), T(티민), C(시토신), G(구아닌) 염색체 서열을 읽어내는 1단계(시퀀싱 머신이 수행)와 텍스트 형태로 저장된 서열 단편들로부터 레퍼런스 유전체를 참조하여 전체 유전체 서열을 찾아내고 변이를 찾는(컴퓨터에서 수행) 2단계로 구성된다. 시퀀싱 머신이 담당하는 1단계의 시간과 비용은 급속히 감소하고 있으나, 1단계의 결과로

※ 본 연구는 지식경제부 및 한국산업기술평가관리원의 IT산업원천기술개발사업의 일환으로 수행하였음[10038768, 유전체 분석용 슈퍼컴퓨팅 시스템 개발].

† 정 회 원 : 한국전자통신연구원 클라우드컴퓨팅연구부 선임연구원
†† 정 회 원 : 한국전자통신연구원 클라우드컴퓨팅연구부 책임연구원
††† 정 회 원 : 한국전자통신연구원 클라우드컴퓨팅연구부장

논문접수 : 2013년 1월 8일
수 정 일 : 1차 2013년 1월 23일
심사완료 : 2013년 1월 23일

* Corresponding Author : ByoungSeobKim(powerkim@etri.re.kr)

나오는 대용량(수십 ~ 수백 기가바이트)의 유전체 단편 정보를 분석하는 2 단계는 여러 단계의 분석 절차를 거치면서 최대 수 테라바이트(TB)의 데이터 처리를 요구하기 하기 때문에 단일 컴퓨터에서 수행할 경우에는 매우 오랜 시간이 걸리는 문제점이 있다[3][4][5].

이러한 대규모 바이오 데이터의 폭증으로 인해 대규모 데이터를 고속으로 안정적으로 계산하고 관리할 수 있는 고성능 컴퓨팅(High Performance Computing : HPC) 시스템이 필요하다.

바이오 응용은 여러 특성을 가진다. 어떤 응용은 계산이 많거나(CPU-Intensive) 또는 입출력이 많거나(IO-Intensive) 또는 메모리 사용량이 매우 많은(Memory-Intensive) 응용들이 있을 수 있다. 대규모 바이오 데이터를 위한 HPC 시스템은 이렇게 다양한 특성의 바이오 응용을 모두 처리할 수 있어야 한다.

또한 일반 비 IT전문가인 바이오 과학자, 의사, 분석자들이 쉽게 사용할 수 있어야 한다. 기존에 HPC 환경을 사용하기 위해서는 HPC를 구성하는 하드웨어(컴퓨팅 노드, 계산 네트워크, 스토리지 및 파일시스템 등)에 대한 일정 지식을 가져야 할 뿐 만 아니라 다수의 계산 노드에 분산 작업을 실행하고 관리해주는 자원 관리 시스템(Resource Managemenet System: RMS)의 사용법을 숙지해야 한다[6][7][8]. 이는 비IT 바이오 연구자들의 고성능컴퓨터 사용을 막고 있는 장벽이 되어 왔다.

유전체 분석과 같은 대규모 바이오 데이터 분석은 한 응용의 출력 결과가 다른 응용의 입력으로 들어가는 파이프라인 형태를 가지며, 이런 응용들을 묶어 워크플로우를 구성하여 분석하는 경우가 대부분이다. 분석자가 한 작업이 종료되는 것을 기다려서 다음 작업을 수작업으로 실행하는 것은 시간 및 노력이 많이 낭비되기 때문에 이렇게 응용들간에 순서와 데이터 종속성이 있는 경우 단일 시스템에서는 스크립트를 작성하여 순차적으로 작업을 수행하여 왔다. 그러나 바이오 분석자들의 경우 스크립트를 작성하고 관리하는 것도 매우 어렵기 때문에, 분석 작업들을 GUI 도구의 도움을 받아 워크플로우로로 정의하고 실행하고 관리하려는 시도들이 있어 왔다. 바이오 응용 뿐 아니라 다양한 과학 응용들도 유사한 속성을 가지기 때문에 출현한 것이 워크플로우 관리 시스템(WMS:Workflow Management System)이다[9][10][11]. WMS의 출현으로 과학자 및 데이터 분석 서비스 프로바이더(Service Provider)와 같은 사용자들은 WMS를 이용하여 좀더 쉽고 다양한 워크플로우(파이프라인)를 구성하여 데이터 분석을 할 수 있게 되었다.

그러나 기존 WMS는 HPC 클러스터 시스템 연동에 적합한 워크플로우 정의 및 실행 방법을 제공하지 못하며, 클러스터 시스템 내부에서 여러 응용을 이용한 워크플로우 정의 방법 및 실행 방법에 대한 고려가 부족하다. 또한, 대규모 컴퓨팅 노드를 이용한 분산 병렬 실행 방법에 대한 고려가 부족하며, 응용이 분산 병렬 실행 정의를 지원하더라도 사용자가 응용의 병렬화 정도를 모두 결정하도록 되어 있어 비IT 연구자에 의해 워크플로우를 현재 클러스터 환경의 HPC 자원에 적합하게 수행되도록 정의하기가 어렵다.

이와 같이 다양한 특성을 가지는 응용들로 구성되는 대규모 바이오 데이터 분석 워크플로우를 HPC 시스템에서 효과적으로 분석하기 위해서는 바이오 워크플로우를 제어 흐름에 따라 쉽게 정의할 수 있어야 하며 워크플로우로 정의된 분석 응용들을 고성능컴퓨팅 환경을 이용하여 응용 특성에 맞는 계산 자원 할당 및 효과적인 분산 처리 작업을 스케줄링을 할 수 있어야 한다. 이를 통해 대규모 바이오 데이터 분석의 성능과 응답시간을 획기적으로 개선할 수 있다.

본 논문에서는 비IT 연구자들이 HPC 환경을 쉽게 이용할 수 있도록 바이오 워크플로우를 쉽게 정의하고 실행할 수 있는 바이오 특화된 워크플로우 기반 대규모 데이터 분석 시스템을 제안한다. 이 시스템은 사용자로 하여금 바이오 워크플로우의 특성과 자원 요구량 정보를 분석하는 것을 도와줄 뿐만 아니라 자동으로 사용자가 정의한 워크플로우를 다수의 HPC 작업으로 변환하여 실행하고 그 결과를 관리해 줄 수 있다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구들을 분석하고 3장에서는 MAHA 시스템과 MAHA 시스템 워크플레이스에 대해 소개한다. 4장에서는 워크플로우 기반 바이오 데이터 분석 시스템의 개요와 워크플로우의 정의, 변환, 실행, 그리고 분석에 대해 설명한다. 5장에서는 워크플로우 모델을 변환하고 실행하는 과정을 구현 관점에서 설명하고 6장에서는 결론을 맺는다.

2. 관련 연구

바이오 데이터 분석이 점점 더 복잡해지고, 여러 응용들의 실행을 요구하게 되면서, 워크플로우 관리 시스템은 바이오인포매틱스 분야에서 가장 중요한 도구로 인식되고 있다. 그리하여, 바이오인포매틱스 분야를 목표로 다양한 워크플로우 관리 시스템들이 개발되었으며, 이러한 워크플로우 관리 시스템은 워크플로우 구성 및 실행 기법에 따라 크게 두 종류로 분류할 수 있다.

첫 번째 워크플로우 관리 시스템들은 바이오 과학자들이 특정 수준의 스크립팅을 하거나, 로컬 도구 및 웹 형식들을 사용할 수 있도록 개발되었다. 이러한 워크플로우 관리 시스템은 보통 이미 존재하는 데이터 집합에 대한 질의 및 결과 데이터의 변형 기능을 포함하며, Taverna[9], Kepler[10] 및 Triana[11]가 대표적인 예이다.

Taverna[9]는 SCUFL(Simple Conceptual Unified Flow Language)라는 워크플로우 마크업 언어를 제공하며, 다양한 로컬 및 원격 프로세서들에 해당하는 태스크들을 바탕으로 워크플로우를 구성할 수 있는 GUI 기반의 워크벤치를 제공한다. 자바 기반의 동적 스크립트와 R 스크립트를 지원하며, myExperiment라는 워크플로우 공유 환경을 제공하고 있다.

Kepler[10]는 UC Berkeley에서 개발한 Ptolemy II 시스템 기반 워크플로우 관리 시스템이다. Kepler는 액터(actor)라는 독립적인 컴포넌트들의 구성을 통해 워크플로

우를 작성할 수 있도록 지원한다. 워크플로우의 실행은 입력이 가용한 액터들의 독립적인 실행을 통해 이뤄진다. Kepler의 액터들은 기본적으로 로컬 자바 쓰레드를 통해 실행되나, 웹 및 그리드 서비스를 위해서 분산 실행 쓰레드로 확장 실행된다.

Triana[11]는 많은 이종의 서비스를 바탕으로 복잡한 워크플로우를 구성하고, 구성된 워크플로우를 P2P나 그리드 네트워크를 바탕으로 실행할 수 있도록 지원한다. 특히, Triana는 WServe API를 통해 그래픽 환경 하에서 웹 서비스의 발견, 실행, 구성 및 등록이 가능하도록 지원한다.

한편, 로컬 및 컴퓨팅 노드들의 클러스터를 바탕으로 워크플로우 실행을 관리하기 위한 워크플로우 관리 시스템들이 개발되었다. 이러한 워크플로우 시스템에는 Biopipe[12], Wildfire[13], Pegasys[14], Ergatis[15], BioWMS[16] 등이 있다.

Biopipe[12]는 컴퓨팅 노드들의 클러스터를 바탕으로 워크플로우의 실행을 제공하기 위하여, 실행 가능한 프로그램들의 인터페이스를 위한 랩퍼들(wrappers)과 데이터 어댑터들을 제공한다. Biopip는 단지 프로그램들의 순차적인 실행(pipeline)만을 고려하기 때문에, 반복 수행을 위한 루프(loops)을 지원하고 있지 않다.

Wildfire[13]는 Jemboss[17]를 바탕으로 워크플로우를 구성하고 실행하기 위한 통합 환경을 제공한다. Biopipe와 같이 컴퓨팅 노드 클러스터를 바탕으로 워크플로우를 실행할 수 있으나, GEL(Grid Execution Language)[18]을 통해 직접적인 프로그램 실행을 지원한다는 점에서 차이가 있다.

Pegasys[14]는 게놈 서열(genomic seqeunce) 분석에 관한 다양한 도구(tool)을 지원할 뿐만 아니라, 워크플로우의 실행을 통해 발생하는 결과 데이터를 GFF(General Feature Format) 형태로 관계형 데이터베이스 시스템에 저장하고 추후 분석에 활용할 수 있도록 지원한다는 특징을 가진다.

Ergatis[15]은 Pegasys와 같이 게놈 서열 분석에 특화된 워크플로우 관리 시스템으로, 워크플로우 실행 결과를 Chado DB를 통해 관리할 수 있다. 그러나, Pegasys와 달리 Ergatis는 OGE(Oracle Grid Engine)[7]을 워크플로우 실행 엔진으로 사용하고 있으며, 웹 기반의 인터페이스를 제공한다.

마지막으로 BioWMS[13]는 에이전트 기반의 미들웨어 형태로 개발된 웹 기반 워크플로우 관리 시스템이다. BioWMS는 사용자 워크플로우로부터 동적으로 워크플로우 엔진을 생성한다. 생성된 워크플로우 엔진은 여러 에이전트로 구성되는 시스템으로, 에이전트들은 서로 협력하여 워크플로우를 실행한다.

본 논문에서 제안하는 바이오 데이터 분석 시스템은 두 번째 워크플로우 관리 시스템에 속하는 것으로, GUI 기반으로 워크플로우 정의, 실행 요청 및 실행 결과를 관리할 수 있도록 지원한다. 한편, 워크플로우 실행에 있어 최적의 설정 정보를 찾을 수 있도록 워크플로우 분석 기능을 제공한다는데 차이가 있다.

3. MAHA 시스템 워크플레이스

본 논문에서 제안하는 “HPC 환경을 위한 워크플로우 기반의 바이오 데이터 분석 시스템”은 MAHA(MAny-core Hpc system for bio-Application) 시스템 워크플레이스의 일부 기능들을 기반으로 하고 있다.

MAHA 시스템은 범용 프로세서(CPU)와 함께 GPGPU (General Purpose computing on GPU)[19], MIC(Many Integrated Core)[20] 등과 같은 이기종 매니코어 기반 성능 가속 장치를 포함하는 이종 자원 기반의 고성능 컴퓨팅 시스템으로 바이오, 특히 유전체 및 단백질 분석과 같은 신산업 융합분야의 사용자들을 위한 HPC 시스템이다.

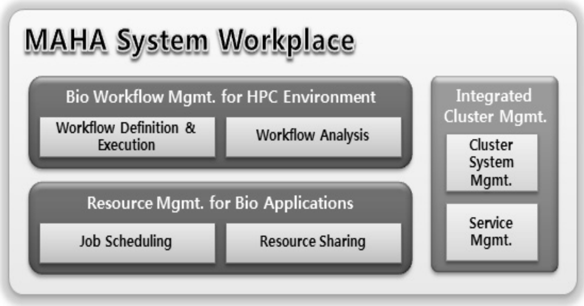

Fig. 1. The functionality of MAHA system workplace

MAHA 시스템 워크플레이스는 MAHA 시스템 운용을 위한 시스템 소프트웨어이다. MAHA 시스템 워크플레이스는 이종 자원 기반의 MAHA 시스템 상에서 바이오, 특히 유전체 및 단백질 분석과 같은 바이오 응용 서비스를 효과적으로 실행하기 위해서 이질적인 특성을 갖는 다양한 형태의 이종 자원들을 효과적으로 관리하고 사용 환경의 복잡도를 최소화 할 수 있는 성능 가속 시스템 SW이다. 이를 통해 성능 최적화 및 고성능 컴퓨팅의 활용편의성을 확대할 수 있다.

MAHA 시스템 워크플레이스는 일반 사용자들이 쉽게 바이오 워크플로우를 정의하고 HPC 시스템에서 효과적으로 실행할 수 있도록 도와주며 시스템 관리자에게는 통합 HPC 클러스터 관리 도구를 제공한다.

MAHA 시스템 워크플레이스는 MAHA 시스템에 대한 충분한 사전 지식이 없는 일반 사용자도 MAHA 시스템을 편리하게 사용(응용을 실행)할 수 있는 MAHA 시스템 사용 환경을 제공하고 바이오 워크플로우 기반 사용 환경을 지원한다. 워크플로우를 구성하는 응용의 특성을 분석하고 자원 요구량을 추출하며 분석을 통해 생성된 응용 분석 정보를 기반으로 사용자가 정의한 추상화된 워크플로우를 MAHA 시스템에 최적화된 분산/병렬 실행 작업으로 변환할 수 있다. 또한 이종 자원의 상태 정보를 바탕으로 최적의 자원 사용 계획을 수립하여 작업 부하를 효율적으로 배분할 수 있다. 마지막으로 관리 편의성을 위한 단일 관리점 기반의 MAHA 시스템 통합 관리를 통해 형상 및 운영관리, 응용

및 서비스 제공을 위해 필요한 소프트웨어의 설치 및 구성, MAHA 시스템을 구성하는 자원 상태 모니터링, 소프트웨어 구성요소에 대한 관리를 지원한다.

MAHA 시스템 워크플레이스는 Fig. 1에서와 같이 바이오 워크플로우 관리, 이종 자원 관리, 클러스터 통합 관리 기능을 포함한다. 본 논문에서는 MAHA 시스템 워크플레이스의 최상위 기능인 바이오 워크플로우 관리 기능을 중심으로 이종 자원관리 기능을 포함하는 워크플로우 기반의 바이오 데이터 분석 시스템을 기술한다.

4. 워크플로우 기반의 바이오 데이터 분석 시스템

4.1 시스템 개요

워크플로우 기반 바이오 데이터 분석 시스템(Workflow-based Bio data Analysis System)은 사용자를 두 부류로 구분한다. 일반사용자는 바이오 과학자로서 자신의 워크플로우를 정의하고 실행할 수 있는 사람이고, 관리자는 일반사용자의 사용자 워크플로우 및 실행 워크플로우를 관리하고, 일반 사용자의 요청시 워크플로우 분석 기능을 대행하는 사람이다. 워크플로우 분석은 일반적으로 실행보다 많은 자원을 일시적으로 투입하여 응용 분석을 수행하므로 관리자 권한으로만 실행할 수 있다.

상기 Fig. 2에서는 본 논문에서 제안하는 바이오 데이터 분석 시스템의 구조를 도시하였다. 본 시스템은 바이오 워크플로우 관리자와 이종 자원 관리자로 구성된다.

바이오 워크플로우 관리자는 최종 사용자를 위한 사용자 인터페이스를 포함하며, 크게 3개의 기능 모듈로 구성된다. 워크플로우 정의 및 실행 관리 모듈은 사용자가 바이오 워크플로우를 정의할 수 있는 정의 모델 제공 및 정의한 워크플로우 정보 관리 기능을 제공하며, 워크플로우 실행 제어, 상태 관리, 이력 관리 기능을 제공한다.

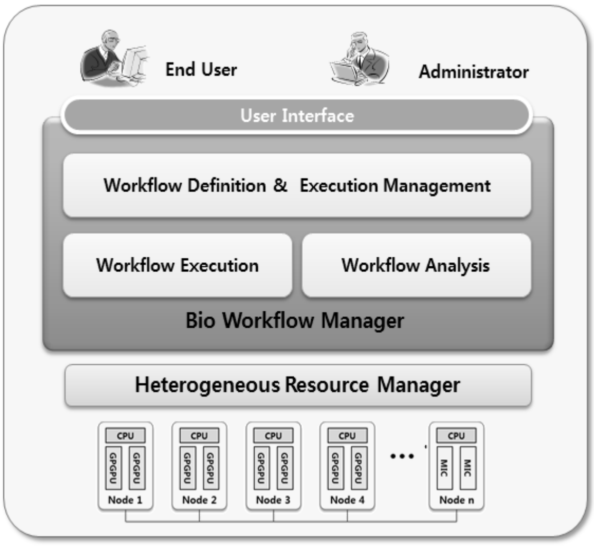

Fig. 2. Bio data analysis system's architecture

워크플로우 실행 모듈은 하부의 이종 자원 관리자 시스템을 이용하여 사용자가 정의한 워크플로우를 대규모 계산 노드에 분산 처리하는 기능을 제공한다. 이를 위해 사용자가 정의한 바이오 워크플로우의 개별 워크들을 이종 자원 관리자 시스템이 계산 노드에 실행할 수 있는 작업 형태로 변환한다. 또한 워크플로우 분석 모듈에서 응용 분석 정보를 제공하면 최적 실행을 위한 변환 기능을 제공한다.

워크플로우 분석 모듈은 시스템에서 반복적으로 사용하고자 하는 사용자의 바이오 워크플로우를 사전에 분석하여 구축된 계산 노드의 자원에서 최적으로 실행될 수 있도록 워크플로우를 구성하는 응용 프로그램의 처리 특징을 분석하여 사용자 및 워크플로우 실행 모듈에 제공한다.

이종 자원 관리자는 GPGPU, MIC 등과 같은 이종 매니코어 기반 성능가속 장치를 포함한 계산 노드들에 바이오 워크플로우 관리자가 제출한 작업들을 할당하고 실행하는 역할을 수행한다. 이종 자원 관리자는 SLURM[6], OGE[7] 등과 같은 기존의 자원 관리 시스템을 활용할 수 있다. 본 논문에서는 바이오 워크플로우 관리자를 중심으로 설명한다.

4.2 워크플로우의 정의 및 실행관리

1) 워크플로우 정의

본 시스템은 사용자가 워크플로우를 정의하기 위한 전용 워크플로우 모델(이하 MAHA 워크플로우 모델)을 제공한다. MAHA 워크플로우 모델은 사용자가 처리하고자 하는 데이터 및 데이터를 처리할 응용프로그램들을 엮어서 전체 데이터 분석을 위한 워크플로우를 정의할 수 있는 규격을 제공한다.

MAHA 워크플로우 모델은 '워크플로우 정의 모델'과 '워크플로우 실행 설정 모델'로 구성된다. '워크플로우 정의 모델'은 Fig. 3에서 보는 바와 같이 입출력 데이터 설정, 입출력 데이터와 명령어(CMD)와의 관계 및 동시 수행 가능 여부 등을 포함하는 워크(Work 또는 Task)를 정의할 수 있다. 또한 워크와 워크는 중간 데이터를 이용하여 연결이 가능하며 이러한 연결을 통하여 여러 단계로 구성된 워크플로우 정의가 가능하다. '워크플로우 실행 설정 모델'은 이와 같이 정의된 워크플로우를 대상으로 실행할 때마다 변화될 수 있는 입력 데이터 정보 및 병렬 수행 정보와 같은 실행 설정 정보를 정의할 수 있는 규격을 제공한다.

MAHA 워크플로우 모델 규격은 XML Schema 형식이며, 사용자는 MAHA 워크플로우 모델의 XML Schema 규격에 맞는 워크플로우를 정의함으로써 사용자 워크플로우를 시스템에 제출하여 처리할 수 있다. (MAHA 워크플로우 모델의 XML Schema 규격은 지면 관계상 생략한다.)

MAHA 워크플로우 모델은 네 가지 특징적인 기능을 제공한다. 첫 번째로 명령어 기반 쉬운 모델링을 제공한다. 기존 프로그램 명령어 및 배치 스크립트와 유사한 개념의 모델을 제공함으로써, 사용자는 단일 노드에서 실행하는 것처럼 워크플로우를 정의할 수 있다. 다만, 워크의 병렬 수행

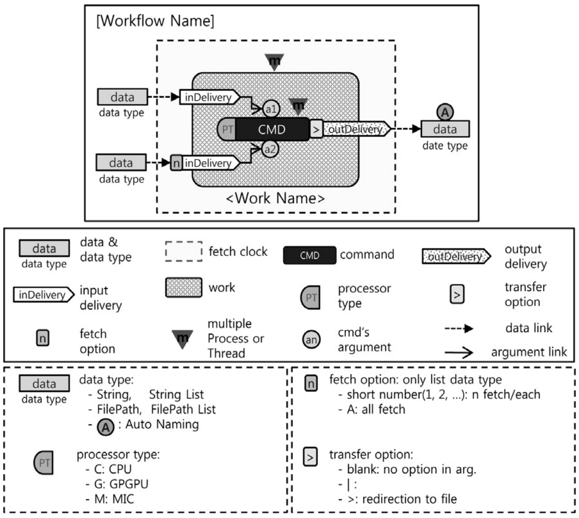

Fig. 3. MAHA workflow definition model

가능성(MultiProcess) 및 응용 프로그램의 멀티쓰레드(MultiThread) 지원 여부 등을 설정하면 워크플로우를 대규모 컴퓨팅 노드에서 수행 할 수 있다. 두 번째로는 사용자가 쉽게 워크플로우를 정의할 수 있는 추상 워크플로우를 제공한다. 세 번째로 보조 연산 장치 설정 기능을 제공한다. CPU 기반의 응용프로그램뿐만 아니라 GPGPU, MIC와 같은 성능 가속 장치에 대한 실행 설정이 가능하여 이와 같은 이종 자원 클러스터 환경을 지원한다. 마지막으로 다양한 데이터 전달 방법을 제공한다. 워크 사이의 데이터 전달 방법을 파일, 메모리, 소켓 등 다양한 전달 방법을 제공한다. 파일의 경우 단지 워크 사이의 데이터 전달을 위해서 임시로 생성되는 중간 결과의 경우 사용자가 파일명을 명시하지 않아도, 시스템이 내부에서 중간 파일을 할당하고 파일 및 메모리 등의 중간 매체를 활용하여 다음 단계의 워크로 결과를 전달 하는 방법을 설정할 수 있는 방법을 제공한다.

MAHA 워크플로우 모델을 이용한 워크플로우 정의에 대한 이해를 돕고자 아래와 같은 데이터 분석 시나리오를 예시로 워크플로우 정의 방법을 설명한다.

[시나리오] 웹 방문 로그 파일을 분석하여 user1의 방문 횟수를 구하라.

상기 시나리오는 웹 로그 파일들을 입력으로 grep 명령어를 이용하여 'user1'이 포함된 줄을 추출하여, wc(word count) 명령어를 이용하여 추출한 전체 줄 수를 구하는 명령을 순차적으로 실행하여 결과를 얻을 수 있다.

사용자는 이와 같은 정의 모델을 이용하여 쉽게 워크플로우를 정의할 수 있으며, 예제 시나리오에 대한 워크플로우 정의 모습 예시가 Fig. 4와 같다.

사용자는 정의한 워크플로우를 실행하기 위해서 제출하기 전에 도구의 실행 설정 인터페이스를 이용하여 다음과 같은 실행 설정 정보들을 설정할 수 있다.

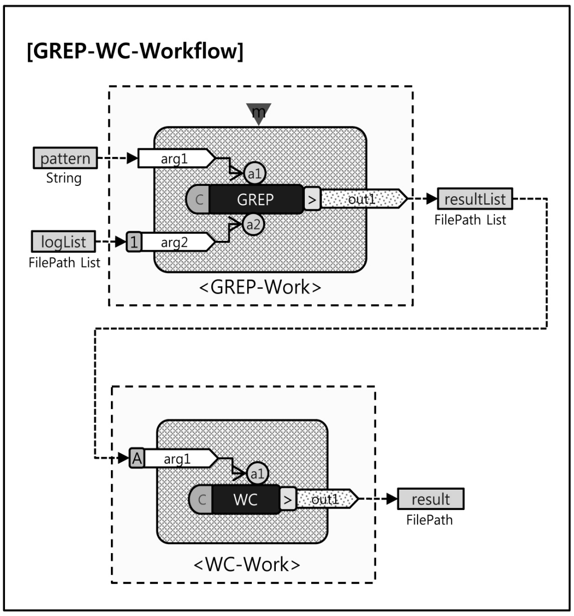

Fig. 4. GREP-WC workflow

Table 1. Execution configuration info. for GREP-WC workflow

Parallel Info.	Input Data	Intermediate Data	Output Data
m=5	pattern user1 String logList location: /web/log pathlist: 1.log 2.log 3.log 4.log 5.log FilePathList	resultList location: /result pathlist: 1.out 2.out 3.out 4.out 5.out FilePathList	result location: /out filename: visit_count.txt FilePath

결국, 사용자가 워크플로우 정의 도구를 이용하여 정의한 워크플로우 및 실행 설정 정보는 Fig. 5, Fig. 6과 같이 XML 형식으로 서버에 전달되어 처리된다.

2) 바이오 워크플로우 실행 관리

a) 바이오 워크플로우 정보 관리

바이오 서비스를 위한 워크플로우를 정의하는 것은 많은 시간과 노력을 필요로 하는 작업으로 정의된 바이오 워크플로우의 반복 활용을 위하여 바이오 워크플로우의 등록 및 삭제를 포함하는 관리 기능을 제공한다. 다중 사용자 간의 워크플로우 협업 및 공유를 지원하기 위하여 등록된 워크플로우는 공유 여부를 설정할 수 있다.

```
...
<!--+++++ Work1:GREP +++++-->
  <Work name="GREP-Work" isMultiProcessing="true">
    <InputDelivery name="arg1">
      <InputDataLink inputDataName="pattern"/>
      <ArgumentLink commandName="GREP" argumentName="a1"/>
    </InputDelivery>
    <InputDelivery name="arg2" fetchOption="1">
      <InputDataLink inputDataName="logList"/>
      <ArgumentLink commandName="GREP" argumentName="a2"/>
    </InputDelivery>

    <CMD name="GREP" processorType="CPU" command="grep"
isMultiThreading="false">
      <Argument name="a1"/>
      <Argument name="a2"/>
    </CMD>
...

<!--+++++ Data Set +++++-->
<!-- GREP-Work: input -->
  <Data name="pattern" type="String"/>
  <Data name="logList" type="FilePathList"/>
...
```

Fig. 5. GREP-WC 워크플로우 정의 XML

```
...
  <MultipleConfig>
    <Work name="GREP-Work" multiProcessNumber="5"/>
  </MultipleConfig>
  <DataSet>

    <Data name="pattern">
      <String>user1</String>
    </Data>

    <Data name="logList">
      <FilePathList>
        <location>/web/log</location>
        <FileNameList>
            1.log
            2.log
            3.log
            4.log
            5.log
        </FileNameList>
      </FilePathList>
    </Data>
...
```

Fig. 6. Execution configuration XML for GREP-WC workflow

b) 바이오 워크플로우 접근 제어

바이오 워크플로우 관리자 블록은 등록된 바이오 워크플로우와 워크플로우의 실행 및 분석 이력에 대한 사용자(관리자, 일반 사용자) 별 접근 제어 기능을 제공한다.

c) 바이오 워크플로우 실행 결과 통보

바이오 데이터 분석 서비스는 비교적 긴 시간의 처리를 요구하는 워크플로우이므로 전체 워크플로우 실행이 완료되면 사용자가 지정한 이메일(e-mail)로 성공/실패 등의 실행 결과 상태를 통보하는 기능을 제공한다.

4.3 워크플로우 변환 및 실행

1) 워크플로우 변환

워크플로우 변환은 추상화된 사용자 정의 워크플로우를 HPC 시스템에서 실행 가능한 실행워크로 구성되는 구체화된 워크플로우로 변환하는 단계이다.

사용자 워크플로우는 각 바이오 응용의 특성 및 실행 방법, 바이오 응용 간의 실행 순서 및 데이터 흐름 관계 정보

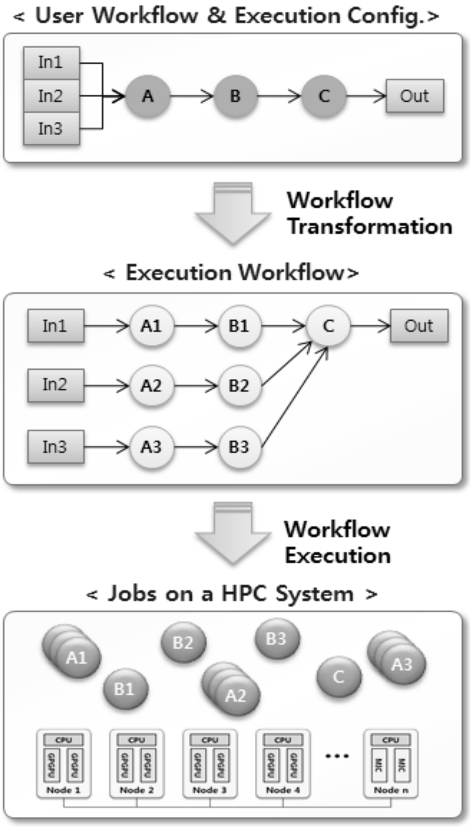

Fig. 7. Workflow transformation and execution

를 포함하는 추상화된 워크플로우이다. 일반 사용자는 워크플로우 정의 모델을 이용하여 자신만의 바이오 워크플로우를 정의한다. 이렇게 사용자가 정의한 워크플로우는 입력데이터를 바꾸어 가면서 계속 재활용 가능하다. 바이오 워크플로우에 입력과 출력 파일 그리고 옵션 등은 워크플로우 실행설정 모델을 이용하여 정의한다(사용자는 GUI를 통해 필요한 입력, 출력 파일과 응용 옵션을 정의하면 시스템에서 자동으로 실행설정 모델로 저장 가능하다). 예를 들어 Fig. 7에서 A, B, C는 사용자 워크플로우이고 in1, in2, in3, out은 실행설정 데이터이다.

사용자가 정의한 바이오 워크플로우 모델과 워크플로우 실행설정 데이터 모델을 시스템에 제출하면, 워크플로우 변환 및 실행 모듈에서 사용자 워크플로우를 HPC 시스템의 작업으로 맵핑될 수 있는 구체화된 실행 워크플로우로 동적으로 변환한다.

변환과정을 통해 실행 설정 모델로부터 입출력 파일명을 포함한 인자들을 받아서 추상화된 사용자 워크플로우 정의 모델의 정보와 결합하여 실제 실행될 구체 실행워크들의 워크플로우가 새로 생성된다. 실행 워크플로우에서 사용자 바이오 워크플로우의 각 응용의 특성, 실행순서 등의 정보는 이종 자원 관리자가 인지 할 수 있는 시스템 정의 인자로 변환된다.

예를 들어 Fig. 7에서 A, B, C 세 개의 응용으로 구성되는 사용자 워크플로우에서 A, B 응용은 입력파일별로 실행

되어야 하는 워크이고, C 응용은 입력파일별 실행결과를 취합하는 워크라 가정하자. 실행 워크플로우에서는 입력파일(In1, In2, In3) 별로 A, B 실행워크가 각각 표현(A1-3, B1-3)되고, 세 개의 입력 파일별 A, B 작업의 결과가 하나의 C 실행워크로 취합되는 형태로 구체화된다. 이렇게 사용자 워크플로우에서는 3개의 워크로 간단하게 표현되었을 지라도 실행 워크플로우에서는 입력 설정 정보의 수에 따라 다수의 구체화된 실행 워크로 변환된다.

2) 워크플로우 실행 및 제어

변환된 실행 워크플로우의 각 실행워크는 HPC 시스템의 작업으로 맵핑되어 이종 자원 관리자에 제출됨으로써 실행된다. 제출된 작업들에 대한 중지(stop), 일시 정지(pause), 재개(resume)을 통해 실행 워크플로우의 제어 기능을 제공한다. 또한 실행중인 워크플로우에 대한 감시 기능을 통해 현재 실행중인 워크플로우의 상태정보와 진행률 정보 등을 제공한다.

3) 실행워크플로우 이력관리

실행 완료된 실행워크플로우의 이력 정보를 저장하고, 삭제하며, 정보를 제공한다. 이 이력정보는 실행워크플로우의 제출시간, 시작시간, 종료시간 등의 모든 정보와 각 실행 작업별로 CPU 사용량, 메모리 사용량, 디스크 사용량 등의 통계 정보도 함께 제공된다.

4.4 워크플로우 분석

본 시스템은 바이오 연구자와 같은 클러스터 환경에 익숙하지 않은 사용자가 대규모 클러스터 환경에서 워크플로우를 쉽고 빠르게 처리될 수 있도록 실행 설정 정보를 제시해주는 워크플로우 사전 분석 기능을 제공한다.

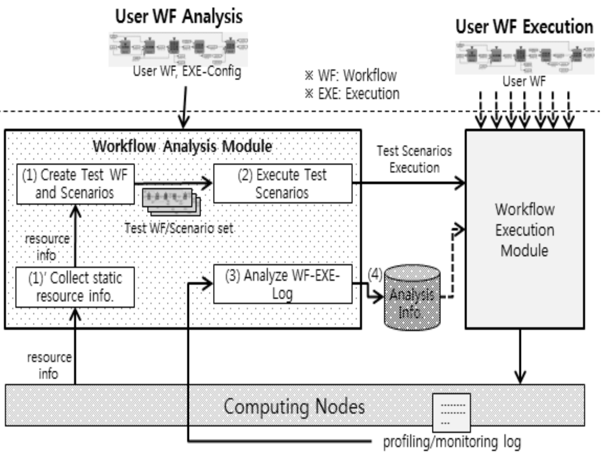

Fig. 8. User workflow pre-analysis procedure

Fig. 8에서 보는 바와 같이 워크플로우 사전 분석 기능은 등록된 사용자 워크플로우를 반복적인 실행 또는 다수 사용자에 의해 널리 실행되기 전에 사전 분석을 통하여 사용자 워크플로우가 최적 실행될 수 있도록 워크플로우 실행기에 최적 정보를 제공한다.

'사용자 워크플로우 분석' 주요 흐름은 (1) 사용자가 정의한 사용자 워크플로우와 컴퓨팅 노드 자원 규격 정보를 기반으로 다양한 실험 워크플로우(사용자 워크플로우 형식) 및 실험 시나리오(실행 설정 정보 형식)를 생성한다. (2) 생성한 실험 시나리오를 워크플로우 실행기에 제출하여 실행한다. (3) 각 실험 시나리오가 실행되는 동안 기록한 자원 사용 로그 파일을 분석하여 워크플로우 실행을 위한 최적 정보를 추출하여 저장한다. 최적 정보는 다음과 같은 병렬 수행 정보 및 워크 실행 시 필요한 자원 요구 정보 등을 포함한다.

* 병렬 실행 정보: 다중 쓰레드 수,
다중 프로세스 수 등
* 자원 요구 정보: CPU, 메모리, I/O 요구량 등

분석 이후 '사용자 워크플로우 실행'이 요청되면, 워크플로우 실행 모듈은 이러한 최적 정보를 이용하여 사용자 워크플로우를 빠르게 실행한다.

5. 구 현

Fig. 9는 본 논문에서 제안하는 워크플로우 정의 모델로 모델링한 유전체 분석 사용자 워크플로우의 한 예 이다.

유전체 분석 워크플로우는 유전체 단편들이 저장된 입력 파일을 읽어 참조 유전체와 비교하여 그 단편들의 순서를 맞추면서 전체 유전체 순서를 완성하는 re-sequencing 방법에 대한 것이다. 유전체 서열분석 응용은 bwa[21], samtools [22]를 이용한다. Fig. 9에서와 같이 유전체 분석 워크플로우는 5개의 워크로 구성된다.

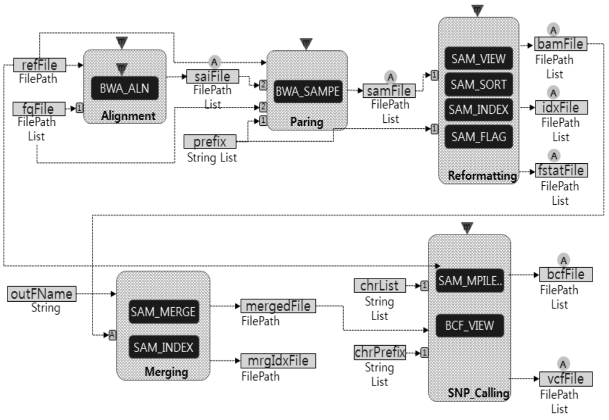

Fig. 9. Genome analysis user workflow(example)

'Alignment' 워크는 염기서열 분석기로부터 나오는 유전체 단편들이 텍스트 형태로 저장된 sequence reads 파일(FASTQ 파일 포맷)을 읽어, 각 유전체 단편이 참조 유전체의 어느 부분과 유사한지를 검색한다. 'Pairing' 워크는 'Alignment' 단계에서 검색된 정보를 이용하여 SAM

(Sequence Alignment/Map) 포맷의 결과파일을 출력한다. ‘Reformatting’ 워크는 텍스트 포맷의 SAM 파일을 바이너리 포맷의 BAM 파일로 변환하고 추후 작업의 속도 향상을 위해 정렬과 인덱싱을 수행한다. Merging 워크는 ‘Reformatting’ 작업의 결과인 BAM 파일을 모두 묶어 하나의 파일로 저장하고 인덱스를 생성한다. ‘SNP_calling’ 워크는 단일염기변이를 확인할 수 있도록 Pileup 파일을 만드는 작업을 수행한다.

각 워크는 다수의 응용 명령으로 구성될 수 있다. ‘Alignment’, ‘Pairing’ 워크는 하나의 명령으로 구성된 워크이나 ‘Reformatting’, ‘Merging’, ‘SNP_calling’과 같은 워크는 두 개 이상의 명령으로 구성된다. 하나의 워크는 워크플로우 변환시 하나의 스크립트로 맵핑되기 때문에 하나의 워크에 들어가는 명령들은 동일한 RMS 작업을 통해 순차적으로 실행된다. 따라서 각 명령들이 필요로 하는 자원이 매우 다를 경우(예를 들어 첫번째 명령은 CPU core를 4개 사용하는데 두 번째 명령은 1개만 사용하는 경우)에는 워크를 분리하는 것이 바람직하다.

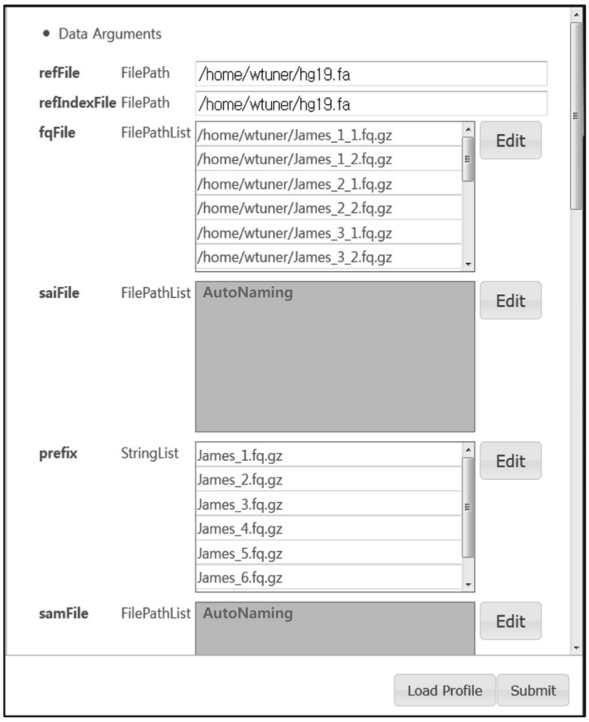

Fig. 10. Execution configuration for genome analysis user workflow(example)

Fig. 9에서 워크들에 점선으로 연결된 작은 사각 박스들(refFile, fqFile, saiFile 등)은 실제 워크플로우 실행을 위한 정보(입력파일명, 출력파일명, 응용별 옵션 등)이다. 이 정보들은 워크플로우 모델에 정의된 규격으로 워크플로우 제출시에 Fig. 10과 같이 별도의 웹 인터페이스를 통해 사용자가 설정 해준다. 워크플로우 모델에서 Ⓐ로 표시된 AutoNaming이 설정된 정보들은 중간에 임시로 생성되는 파일명으로 모델 정의 규칙에 의해 시스템이 자동 생성한다.

사용자 워크플로우와 실행설정데이터는 Fig. 11와 같이 구체화된 실행워크플로우로 변환된다. Fig. 11은 지면 관계상 실행워크플로우의 일부분만을 도시하였다. 실행 워크플로우는 실행설정 데이터에서 리스트로 제공된 입력 파일, 출력파일 그리고 옵션 리스트들을 읽어 와서 실제 응용을 실행할 수 있는 명령행 스크립트를 만들 수 있는 정보를 포함하는 형태로 변환된다. 실행 워크플로우의 각 실행워크는 RMS를 통해 실행 가능한 스크립트로 자동 생성되어 계산 노드에서 수행된다.

Fig. 12는 상기 유전체 분석 실행워크플로우의 각 실행워크들이 RMS의 한 종류인 SLURM[6]을 통해 작업이 실행되는 상황을 보여준다.

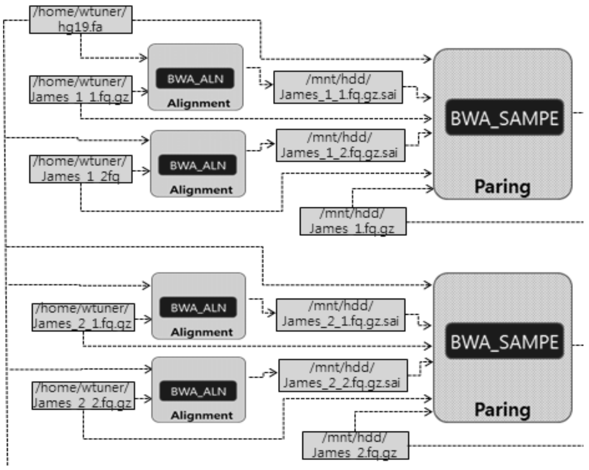

Fig. 11. Execution workflow of genome analysis (example)

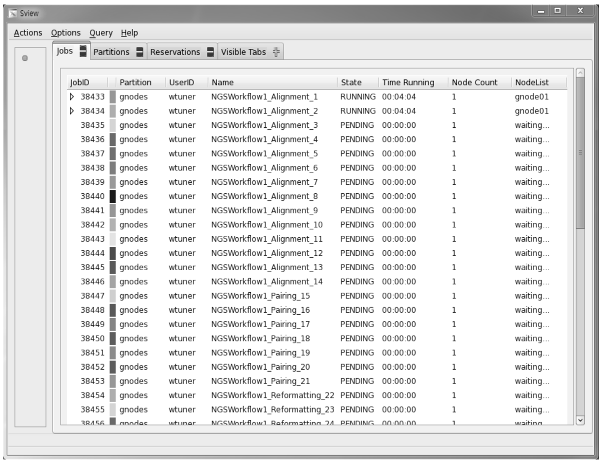

Fig. 12. RMS(SLURM)'s job list(example)

6. 결 론

본 논문에서는 HPC 관련 충분한 지식이 없는 바이오 연구자들도 대규모 바이오 데이터를 HPC 환경에서 쉽고 효

과적으로 분석할 수 있는 바이오 데이터 분석 시스템을 제안하였다.

제안된 시스템은 바이오 연구자들이 일반적으로 사용하는 명령어 및 배치 스크립트와 유사한 MAHA 워크플로우 모델을 제공하고, 이 모델을 따르는 사용자 워크플로우를 GUI 환경에서 쉽게 정의할 수 있는 도구를 제공함으로써 바이오 데이터 분석의 편의성을 제공한다. 그리고 이러한 도구를 통해 정의된 사용자 워크플로우를 SLURM 등과 같은 HPC 시스템의 자원 및 작업 관리 SW의 작업으로 자동으로 변환하고 실행하는 워크플로우 실행 기능을 제공함으로써, 분산 병렬처리에 대한 충분한 지식이 없고 HPC 시스템에 익숙하지 않은 사용자가 HPC 환경에서 워크플로우를 쉽게 실행할 수 있는 실행 환경을 제공한다. 또한, 사용자 워크플로우를 서비스 전에 사전 분석할 수 있는 사용자 워크플로우 분석 기능을 제공함으로써 서비스 제공자는 대규모 자원을 효과적으로 운영하여 서비스를 좀 더 빠르게 제공할 수 있는 대규모 바이오 데이터 분석 서비스 운영 환경을 제공한다.

참 고 문 헌

[1] J. C. Venter et al., “The Sequence of the Human Genome”, Science, Vol.291 No.5507, pp.1304-1351.

[2] NHGRI Genome Sequencing Program, http://www.genome.gov/sequencingcosts/

[3] Human Genome Project, Wikipedia, http://en.wikipedia.org/wiki/Human_Genome_Project

[4] Biology 2.0, Special report, The Economist, 2010, http://www.economist.com/node/16349358

[5] Yunku Yeu et al., “A survey of sequence alignment algorithms for next-generation sequencing read”, KIISE Database Society Journal, Vol.28 No.1 pp.33-51, 2012.

[6] Simple Linux Utility for Resource Management (SLURM), https://computing.llnl.gov/linux/slurm/

[7] Oracle Grid Engine(Sun Grid Engine), http://www.oracle.com/technetwork/oem/grid-engine-166852.html

[8] TORQUE Resource Manager, http://www.adaptivecomputing.com/products/open-source/torque/

[9] P. Missie et al., “Taverna reloaded”, In Proc. of SSDBM, 2010

[10] I. Altintas et al., “Kepler: An Extensible System for Design and Execution of Scientific Workflows”, In Proc. of SSDBM, pp.423-424, 2004.

[11] S. Majithia et al., “Triana: A Graphical Web Service Composition and Execution Toolkit”, In Proc. of ICWS, pp.514-421, 2004.

[12] S. Hoon et al., “Biopipe: A Flexible Framework for Protocol-Based Bioinformatics Analysis”, Genome Research, Vol.13, No.8, pp.1904-1915, 2003.

[13] F. Tang et al., “Widlfire: distributed, Grid-enabled construction and execution”, BMC Bioinformatics, Vol.6, pp.69, 2005.

[14] S. P. Shan et al., “Pegasys: software for executing and integrating analyses of biological sequences”, BMC Bioinformatics, Vol.5, pp.40, 2004.

[15] J. Orivs et al., “Ergatis: a web interface and scalable software system for bioinformatics workflows”, Bioinformatics, Vol.26, No.12, pp.1488-1492, 2010.

[16] E. Bartocci et al., “BioWMS: a web-baed Workflow Management System for bioinformatics”, BMC Bioinformatics, Vol.8(Suppl I), S2, 2007.

[17] T. Carver and A. Bleasby, “The design of Jemboss: a graphical user interface to EMBOSS”, Bioinformatics, Vol.19, No.14, pp.1837-1842, 2003.

[18] C. C. Lian et al., “GEL: Grid Execution Language”, Parallel and Distributed Computing, Vol.65, No.7, pp.857-869, 2005.

[19] GPGPU, Wikipedia, http://en.wikipedia.org/wiki/GPGPU

[20] Intel MIC, Wikipedia, http://en.wikipedia.org/wiki/Intel_MIC

[21] Burrows-Wheeler Aligner, http://bio-bwa.sourceforge.net/

[22] SAMtools, http://samtools.sourceforge.net/

안 신 영

e-mail : syahn@etri.re.kr

1997년 성균관대학교 정보공학과(학사)

1999년 성균관대학교 전기전자및컴퓨터공학부(석사)

1999년~현 재 한국전자통신연구원 클라우드컴퓨팅연구부 선임연구원

관심분야 : High Performance Computing, Software Architecture

김 병 섭

e-mail : powerkim@etri.re.kr

1997년 전북대학교 정보통신공학과(학사)

1999년 전북대학교 정보통신공과(석사)

1999년~현 재 한국전자통신연구원 클라우드컴퓨팅연구부 선임연구원

관심분야 : 분산 병렬 처리 시스템, 스트림 처리 시스템, 정보 통합 시스템

최 현 화

e-mail : hyunwha@etri.re.kr

2000년 충남대학교 컴퓨터공학과(공학사)

2002년 포항공과대학교 컴퓨터공학과(공학석사)

2002년~현 재 한국전자통신연구원 클라우드컴퓨팅연구부 선임연구원

관심분야 : 분산 병렬 처리, 데이터베이스, 이벤트 스트림 처리

전 승 협

e-mail : shjeon00@etri.re.kr
1998년 고려대학교 컴퓨터학과(학사)
2001년 고려대학교 통신시스템기술(석사)
2009년~현 재 한국전자통신연구원 클라우드컴퓨팅연구부 선임연구원
관심분야 : High Performance Computing, Virtualization

배 승 조

e-mail : sbae@etri.re.kr
1987년 연세대학교 전산과학과(학사)
1992년 Syracuse University 컴퓨터과(석사)
1997년 Syracuse University 컴퓨터과학(박사)
1997년~현 재 한국전자통신연구원 클라우드컴퓨팅연구부 책임연구원
관심분야 : High Performance Computing, Parallel Computing

최 완

e-mail : wchoi@etri.re.kr
1981년 경북대학교 전자공학과(학사)
1985년 KAIST 전산학과(석사)
1985년~2003년 ETRI, TDX/CDMA 전전자교환기용 실시간OS/DBMS/MW/컴파일러 개발책임자
2000년~2011년 한국정보처리기술사회 이사
2004년~2007년 ETRI, 클라우드서비스 기술 개발팀장
2007년 불교명예철학 박사
2008년~2010년 ETRI, SW콘텐츠미래기술연구부장
2011년~현 재 한국전자통신연구원 클라우드컴퓨팅연구부장
관심분야 : Cloud Computing, High Performance Computing