韓國 컴퓨터情報學會 論文誌
第13卷 第3號, 2008. 5.

2008-13-3-1-7

응용 프로그램 온톨로지 기반 시맨틱 그리드 관리 시스템

김 민 성*, 이 관 수*

Semantic Grid Management System based on the Ontology of Application Software

Min-Sung Kim*, Gwan-Su Yi*

요 약

그리드 컴퓨팅은 분산된 컴퓨팅 자원들을 서로 연결하여 방대한 양의 계산 작업을 가능케 하는 기술로써 다양한 분야에서 사용되고 또한 발전되어 왔다. 하지만 기존의 연구들은 주로 그리드 자원을 효율적으로 관리하고 사용하는 것에 초점을 두었지, 실제 그리드 환경에서 수행되는 응용 프로그램들의 정보에 대한 체계적인 이해 및 관리는 제대로 이루어지지 않았다. 따라서 응용 분야의 사용자가 그리드 컴퓨팅 환경을 사용하기 위해서는 매번 응용 프로그램의 그리드 관련 요구사항들을 구체적으로 이해하고 기술할 수 있는 전문적인 지식이 요구되었다. 본 연구에서는 이러한 문제점을 보완하기 위한 응용 프로그램 온톨로지 기반 시맨틱 그리드 관리시스템을 구현하였다. 이 시스템은 응용 프로그램들에 대한 정보와 이들이 요구하는 그리드 관련 정보를 온톨로지 형태로 구축하였다. 이를 기반으로 입력 데이터와 인수에 따른 자원 요구량 예측, 최적 자원 할당 등의 정보를 추론하고, 이 정보를 그리드 미들웨어가 다룰 수 있는 형태로 재해석할 수 있도록 하였다. 또한 본 논문에서는 구축된 시스템을 고성능 컴퓨팅 자원이 요구되는 생물정보학 분석환경 구축에 실제 적용해보고 이를 기존 시스템들과 비교함으로써 본 시스템이 유용하게 사용될 수 있음을 설명하였다.

Abstract

Grid Computing has enabled enormous amount of computational jobs by connecting distributed computing resources. This technology has developed and widely used in various fields. Previous researches usually focused on how to efficiently manage and use the grid resources. However, there was not enough tries to understand and manage information of application softwares in a well-defined structure. Therefore users in application domain need to know about grid deeply to identify and describe the resource requirements matching for each jobs. We introduce a semantic grid management system based on application ontology to overcome this problem. We design and implement the ontology to store various information of the applications. With the ontology, this system can infer the resource requirements from input parameters and input data of the application software and automatically assign appropriate resources by matching the requirement. Also it can transform the information to other forms which grid middlewares can handle. We apply the system to construct an analysis environment of bioinformatics and compare it with other grid systems to explain usefulness of the system.

▸ Keyword : 온톨로지(Ontology), 시맨틱 그리드(Semantic Grid), 생물정보학(bioinformatics), 그리드 관리 시스템(Grid Management System)

• 제1저자 : 김민성 교신저자 : 이관수
• 접수일 : 2008. 4. 16, 심사일 : 2008. 4. 18, 심사완료일 : 2008. 5. 24.
* 한국정보통신대학교 공학부
※이 논문은 2005년 교육인적자원부의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2005-003-C00159)

I. 서 론

그리드 컴퓨팅(Grid Computing) 기술은 지리적으로 분산된 컴퓨터들을 연결하여 상호 이용할 수 있도록 해주는 기술이다. 특히 계산 그리드(Computational Grid)는 컴퓨팅 자원들을 서로 연결하여 방대한 양의 계산 작업을 가능케 함으로써 최근 급속한 데이터양의 증가에 의해 분석량이 증가하고 있는 생명과학과 나노기술 등의 첨단 과학 기술 분야에서 큰 주목을 받고 있다. 그리드 컴퓨팅 기술은 초기 메타컴퓨팅(Metacomputing)이라고 불리어지는 프로젝트들[1,2]로부터 시작하여 대표적 미들웨어 시스템인 글로버스(Globus)[3], Legion[4] 등으로 발전되어 왔다. 이렇게 발전된 그리드 미들웨어 및 관리 시스템들은 그리드 환경을 효율적으로 활용하기 위하여 자원 할당, 자원들 간의 소통, 보안 등의 기본적인 기능들을 제공한다.

하지만 이러한 기술적 발전에도 불구하고 그리드 환경에서 분석을 수행하기 위해서는 사용자는 응용 프로그램의 정보를 수집하여 필요로 하는 자원 요구사항들을 파악하고 그리드 자원에 속한 각 노드들이 이러한 요구사항들을 만족하는지 검토한 후 최종 수행정보를 그리드 미들웨어가 요구하는 형식에 맞춰 기술해야만 한다. 이렇듯 응용 프로그램의 특성을 파악하는 것뿐만 아니라 그리드 미들웨어에 대한 전문적인 지식 및 그에 수반하는 준비 작업들은 그리드 응용분야 확대에 걸림돌이 되고 있다.

이러한 상황에서 미래의 e-Science 기반 구축을 목표로 [5] 시맨틱 그리드(Semantic Grid) 개념이 등장하였다. 시맨틱 웹(Semantic Web) 기술의 발전으로 WWW(World Wide Web) 상에 존재하는 다양한 데이터들을 인간뿐만 아니라 컴퓨터가 이해 가능한 언어로 표시하고 이를 가공하기 위한 RDF(Resource Description Framework)[6], OWL(Web Ontology Language)[7]과 같은 표준화된 온톨로지 언어들이 만들어졌다. 시맨틱 그리드에서는 이러한 기술들을 활용하여 그리드 자원 정보를 온톨로지로 구축하고 추론엔진을 통한 자원 관리 및 자원 할당 등을 제공하는 기술로 발전시켜왔다.

그러나 이러한 기술들은 그리드 자원을 중심으로 한 최적화 시도로서, 자원 요구사항들이 미리 사용자에 의해 주어진 상황 아래에서 이에 적합한 자원을 제시하는 해결책이다. 즉 그리드 자원 정보를 온톨로지화 하여 자원 할당의 효율적인 자동화를 지원하고 있으나, 실제 수행되는 대상인 응용 프로그램들의 정보를 체계적으로 관리하고 이 정보들을 그리드 미들웨어에서 다룰 수 있도록 재해석 하는 부분은 제대로 이루어지지 않아왔다.

본 논문에서는 이러한 문제를 보완하기 위하여 프로그램들의 다양한 특성들 중 그리드 환경에서 수행되기 위해 필요한 다양한 하드웨어 및 소프트웨어적 요구사항 정보들을 기술할 수 있도록 설계된 온톨로지를 구축하였다. 또한 이렇게 구축된 온톨로지 데이터와 웹 서비스를 이용하여 수집된 자원 정보를 바탕으로 응용 프로그램의 입력 데이터 및 인수에 따라 변하는 각 자원의 요구량 예측, 다양한 요구사항들에 부합하는 그리드 자원목록 작성 등의 추론 과정을 수행하고, 이를 사용하려는 그리드 미들웨어에 맞게 변환할 수 있는 응용 프로그램 온톨로지 기반 시맨틱 그리드 관리시스템을 제안하고자 한다. 본 시스템을 생물정보학의 대표적인 응용 분야인 서열정렬 분석환경 구축에 실제 적용해 보고 이를 기존 시스템들과 비교함으로써 본 시스템이 유용하게 활용될 수 있음을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련된 연구들에 대해 개괄적으로 설명하고, 3장은 시스템의 구조 및 응용 프로그램 온톨로지에 대해서 서술한다. 4장에서는 실제 분석 과정 구축에 적용된 예를 바탕으로 기존 시스템들과의 비교를 설명하며, 5장에서는 결론 및 향후 연구를 기술한다.

II. 관련 연구

시맨틱 그리드(Semantic Grid)는 기존 그리드의 확장된 개념으로써 시맨틱 웹에서 사용된 다양한 개념 및 기술들을 기존 그리드와 통합하여 그리드의 다양한 정보 및 서비스들을 좀 더 명확한 형태로 제공하고, 이를 통해 좀 더 나은 사람과 컴퓨터 간의 협력을 가능케 하는 것[8]으로 정의 된다. 이러한 목적을 위하여 웹 서비스(Web Service), 온톨로지(Ontology), 소프트웨어 에이전트(Software Agent)와 같은 다양한 기술들이 활용되고 있다.

대표적인 사례로 최근 글로버스에서는 WS-Resource Framework를 구축하여 웹 서비스를 기반으로 자원관리, 보안, 파일 전송 등 미들웨어의 기본 기능들이 수행되도록 하였다.

또한 시맨틱 그리드의 핵심 기술인 온톨로지를 이용하여 다양한 그룹에서 그리드 자원을 온톨로지로 표현하는 방법[9]이나 온톨로지 기반의 최적 자원 매칭 기술[10,11] 등을 연구하여 왔다. 이러한 연구의 주목적은 자원에 대한 정보를 RDF, OWL 등의 온톨로지 언어로 구축하고 이를 자원 관리 및 자원 할당 과정에 효율적으로 사용하는 기술의 개발이라고

할 수 있다.

또한 시맨틱 그리드라는 새로운 개념이 등장하고 발전되는 과정과 더불어 그리드 자원을 효율적으로 활용하여 안정적이며 빠른 분석을 지원하기 위한 연구들 또한 진행되고 있다. 이러한 연구들은 주로 각 그리드 자원들의 성능을 대략적으로 측정하고 이를 바탕으로 적절한 작업을 할당하기 위한 새로운 그리드 스케줄링 모델을 제시한다.[12,13] 또한 여기에 작업의 연산량 및 네트워크 특성 등을 반영하여 좀 더 효율적인 작업 할당을 시도하기도 하였다.[14,15]

현재까지의 연구들은 앞서 서술된 바와 같이 그리드 자원에 대한 정보를 파악하고 더 나아가 이를 온톨로지 형태로 체계적으로 정리하여 자원 관리 및 할당 과정에 활용하는데 그치고 있다. 이는 그리드 환경을 관리하는 차원에서는 유용할 수 있으나 시맨틱 그리드의 주요 목표중의 하나인 사용상의 편의성을 개선하는 데는 큰 효과를 거둘 수 없다.

또한 작업의 특성을 자원 할당과정에 반영하려는 시도들도 작업들의 특성을 몇 가지로 단순화하고 이를 바탕으로 접근해 왔기 때문에 실제 응용 프로그램들이 가질 수 있는 다양한 특성들을 제대로 반영하지 못하는 한계를 지니고 있다. 특히 입력 데이터와 인수 등에 의해서 같은 작업이라도 연산량 등이 전혀 달라질 수 있는 것이나, 하드웨어적 요구사항 뿐만 아니라 응용 프로그램이 소프트웨어적으로 가질 수 있는 다양한 요구사항들 등을 제대로 다루지 못하고 있다.

이것은 전체 과정의 후반부인 그리드 자원에 대한 정보만을 온톨로지로 체계화 했을 뿐, 전반부 즉 수행 조건에 따른 응용 프로그램들의 다양한 요구사항에 대한 정보를 체계적으로 이해 및 관리하고, 이를 그리드 미들웨어에서 다룰 수 있는 형태로 재해석하는 방법에 대한 연구는 제대로 이루어지지 않았기 때문이다.

따라서 본 연구에서는 이러한 기존 연구의 문제점을 보안하기 위하여 응용 프로그램의 다양한 특징들 중 그리드 환경에서 요구되는 의미체계로 단순화하여 온톨로지 형태로 구성하는 방식을 정의함으로써 응용 프로그램들에 대한 정보를 체계적으로 관리할 수 있는 환경을 만들었다.

또한 이렇게 구축된 온톨로지 데이터와 수집된 그리드 자원에 속한 각 노드들의 자원 정보를 바탕으로 응용 프로그램의 분석 조건에 맞는 각 자원의 요구량 예측, 다양한 요구 사항들에 부합하는 그리드 자원목록 작성 등의 추론과정을 수행하고, 이 정보를 그리드 미들웨어가 다룰 수 있는 형태로 변환할 수 있는 시맨틱 그리드 관리 시스템을 구성하였다.

III. 응용 프로그램 온톨로지 기반 시맨틱 그리드 관리 시스템 설계

3.1. 응용 프로그램 온톨로지 구조

앞서 설명한 응용 프로그램의 다양한 정보들을 온톨로지 형태로 설계하기 위하여 시맨틱 웹의 표준 온톨로지 언어인 OWL(Web Ontology Language)이 사용되었다. 또한 OWL에서 제공하는 추론 기능을 이용하기 위하여 3가지 형태 중의 하나인 OWL-DL을 준수하여 설계되었다.

해당 온톨로지는 개념적으로 크게 세 가지 파트로 나뉘어 설계되었다. 이중 가장 핵심 내용인 프로그램 온톨로지는 응용 프로그램에 대한 기본적인 정보뿐만 아니라 각종 하드웨어적, 소프트웨어적인 요구사항 등의 내용을 담게 된다. 또한 분석 온톨로지는 프로그램들에 대한 정보를 단순히 나열하는 것이 아니라 분석의 목적에 따라 프로그램들을 체계적으로 관리되도록 한다. 마지막으로 데이터 온톨로지는 프로그램의 입력으로 들어가는 데이터들에 대한 체계적인 정리가 밑바탕이 되어야 프로그램의 입력에 대한 정확한 정보 기술 및 데이터의 종류와 그 크기에 따른 자원 요구량 추론이 가능하기 때문에 설계되었다.

이렇게 설계된 온톨로지는 온톨로지 시각화 툴인 Jambalaya[16]를 이용하여 다음 그림 1과 같은 구조로 표현된다. 화살표로 연결된 선은 각 클래스들 간의 관계 즉 OWL의 Object Property를 의미하는 것으로 전체 구조를

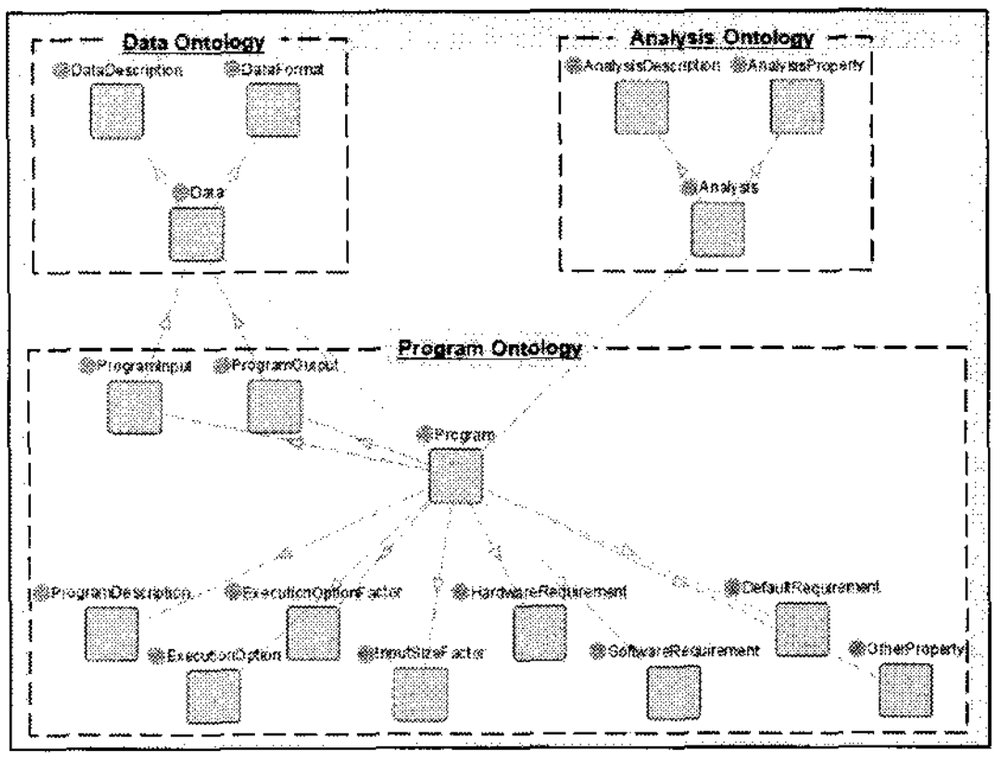

그림 1. 응용 프로그램 온톨로지의 전체 구조
Fig 1. Whole structure of application ontology

한눈에 표현하기 위하여 각 클래스들의 상세 하부 클래스들은 표현되지 않았다.

"ProgramDescription" 클래스는 해당 프로그램의 정식 명칭 및 간단한 설명, 버전 정보 등의 기본적인 정보를 가지고, "ExecutionOption" 클래스는 실행 시에 들어오는 여러 인수들에 대한 정보 및 가능한 조합들에 대한 정보를 가진다.

또한 해당 프로그램의 입력 및 출력 데이터에 대한 정보를 가지는 "ProgramInput" 클래스와 "ProgramOutput" 클래스가 존재하여, "hasDataInfo"라는 Object Property를 이용 데이터 온톨로지의 "Data" 클래스와 연결됨으로써 입출력 데이터에 대한 정확한 명시가 가능하다.

그리고 "HardwareRequirement" 클래스의 하위 클래스들을 이용하여 해당 프로그램이 요구하는 하드웨어적 요구사항들을 하드웨어 별로 기입이 가능하게 하고, "Software Requirement" 클래스의 하위 클래스들을 이용하여 해당 프로그램이 요구하는 운영체제, 시스템 라이브러리, 특정 프로그램 등의 다양한 소프트웨어적 요구사항들을 기입 가능하도록 하였다.

요구사항은 이에 그치는 것이 아니라 CPU, 메모리와 같은 각 자원에 대한 요구량의 정도를 추측하기 위한 클래스들이 존재하는데 "DefaultRequirement" 클래스는 해당 프로그램이 기본적으로 가지게 되는 각 자원에 대한 요구량의 정도를 Very Low, Low, Middle, High, Very High 중 하나의 구간값으로 가지게 된다. 또한 "ExecutionOption Factor" 클래스는 프로그램 실행 시에 입력으로 들어오는 인수에 따라서, "InputDataSizeFactor" 클래스는 입력으로 들어오는 데이터의 종류 및 크기에 따라서 어떻게 각 자원에 대한 요구량이 달라지는 지에 대한 가변적인 자원 요구량 정보를 가지고 있다.

마지막으로 "OtherProeprty" 클래스의 하위 클래스들을 이용하여 프로그램의 MPI, PVM 지원 여부 사항과 같은 기타 고려되지 못한 정보를 기입할 수 있도록 설계되었다.

분석 온톨로지는 특정 분야에 존재하는 다양한 분석들이 계층적 구조로 정리되어 있는 "Analysis" 클래스와 분석에 대한 다양한 정보를 표현하기 위한 "AnalysisAttribute" 클래스의 하위 클래스들로 나뉘고, "hasAnalysis" 라는 Object Property를 통해 "Program" 클래스와 연결이 됨으로써 각 프로그램이 속한 분석에 대한 정보를 기술할 수 있도록 하였다.

"AnalysisAttribute" 클래스의 하위 클래스에는 해당 분석의 명칭 및 간단한 설명 등의 기본적인 정보를 가지는 "AnalysisDescription" 클래스와 해당 분석에 사용되는 프로그램들 간에 가지는 상대적인 특성 정보, 즉 정확성, 연산 속도 등의 정보 및 그에 연관된 프로그램들의 목록을 가지는 "AnalysisProperty"클래스가 있다.

데이터 온톨로지는 특정 분야에 존재하는 다양한 분석 데이터들이 계층적으로 정리되어 있는 "Data" 클래스와 데이터에 대한 다양한 정보를 표현하기 위한 "DataAttribute" 클래스의 하위 클래스들로 나뉘고, 앞서 설명되었듯이 프로그램의 입력 및 출력 정보를 표시하는 클래스들과 "hasDataInfo" 라는 Object Property에 의해 연결되어 있다

"DataAttribute" 클래스의 하위 클래스에는 해당 데이터가 담고 있는 내용의 종류 및 설명 등의 기본적인 정보를 가지는 "DataDescription" 클래스와, 해당 데이터가 어떠한 포맷으로 존재하는 지에 대한 정보를 가지는 "DataFormat" 클래스가 있다.

3.2. 시스템 구조

해당 시스템을 이용한 그리드 기반의 분석 작업의 흐름은 다음의 그림 2에 표현된 바와 같다. 가장 먼저 사용자는 여러 인터페이스를 통해 수행하고자 하는 분석에 대한 작업 기술파일을 작성한다. 이렇게 작성된 기술파일은 시맨틱 그리드 관리 시스템의 입력으로 들어가 추론과정을 거치게 된다. 이 추론과정은 온톨로지 데이터 저장소의 해당 프로그램에 대한 정보와 그리드 자원에 속한 각 노드들에 설치된 리소스 에이전트(Resource Agent)들로부터 웹 서비스를 통해 얻어진 자원에 대한 상세 정보를 바탕으로 이루어진다. 이렇게 추론된 정보는 사용하는 그리드 미들웨어가 다룰 수 있는 형태로 변환됨으로써 실제 분석 작업이 그리드 환경에서 실행된다.

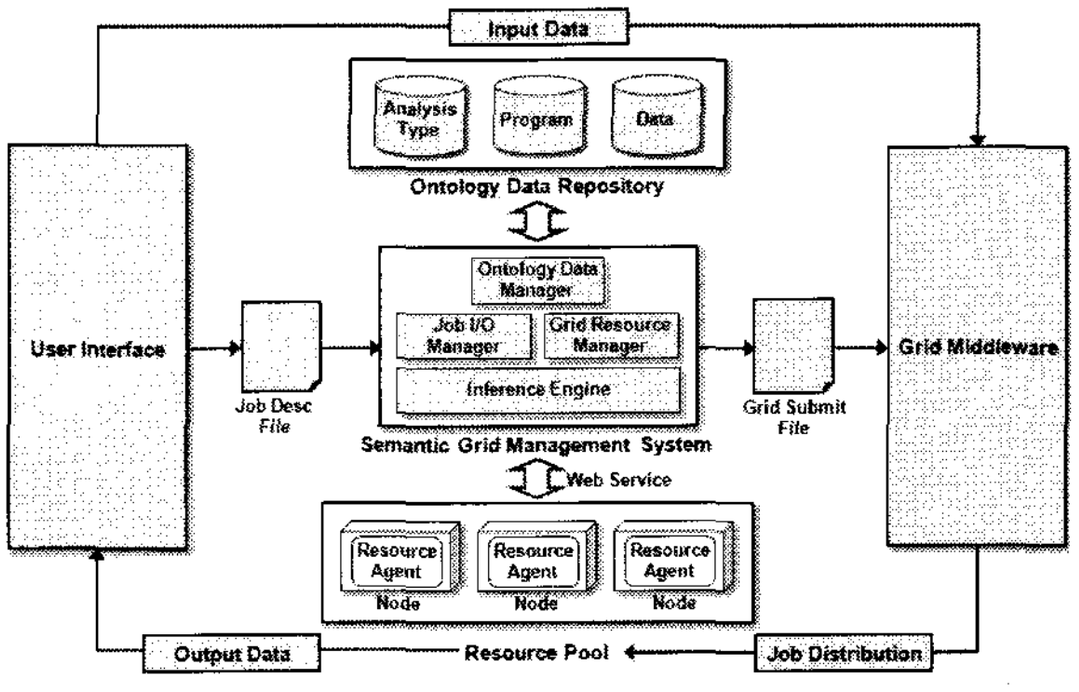

그림 2. 시맨틱 그리드 관리 시스템 기반의 작업흐름
Fig 2. Job flow based on semantic grid management system

해당 시스템은 크게 3개의 관리자, 즉 온톨로지 데이터 관리자(Ontology Data Manager), 작업 입출력 관리자(Job I/O Manager), 그리드 자원 관리자(Grid Resource manager)와 추론 엔진(Inference Engine), 온톨로지 데이터 저장소(Ontology Data Repository) 그리고 각 노드에 설치된 리소스 에이전트(Resource Agent)들로 구성되어 있다.

온톨로지 데이터 관리자(Ontology Data Manager)는 크게 앞서 설명된 구조로 이루어진 응용 프로그램 온톨로지에 특정 클래스의 수정 및 삭제 등의 제한적인 구조 변경의 역할을 하는 온톨로지 스키마 핸들러(Ontology Schema Handler)와 구축된 온톨로지 데이터의 각 클래스들에 실제 객체(Individual)를 추가하는 역할을 하는 온톨로지 객체 핸들러(Ontology Individual Handler) 그리고 이러한 모듈들을 쉽게 사용할 수 있도록 사용자 인터페이스를 제공한다.

이렇게 구현된 온톨로지 데이터 관리자를 이용하여 시스템 관리자, 프로그램 개발자, 분석을 진행하는 사용자 등 다양한 사용자들이 제공되는 인터페이스를 이용하여 응용 프로그램 온톨로지의 구조를 변경하거나 실제 프로그램이나 분석 혹은 데이터에 대한 정보들을 추가해 나간다.

작업 입출력 관리자(Job I/O Manager)는 사용자로부터 입력된 작업 기술파일의 내용을 읽어 들이고, 해당 파일의 입력 포맷이 적합한지에 대한 유효성 검사를 거쳐 추론 엔진에 그 내용을 전달한다. 또한 해당 작업이 실행된 후 그 결과를 사용자에게 전달하는 역할을 한다. 입출력 관리자에서 받아들이는 작업 기술파일의 내용은 표 1에서와 같다.

표 1. 작업 기술파일의 내용
Table1. Contents of Job Description File

변수 이름	내용
WorkingDir	해당 작업이 이루어질 경로
ProgramName	프로그램 이름
AnalysisName	분석 이름
AnalysisProp	분석의 특성 정보
Executable	실행 파일 이름
Parameter	실행 시에 입력되는 인수
InputData	입력 데이터 정보 (데이터의 위치, 종류, 크기)
ExecResult	실행 결과의 내용이 저장될 파일 이름
TransferFile	각 노드에 전달해야할 파일들 의 목록

그리드 자원 관리자(Grid Resource Manager)는 그림 3과 같은 구조로 구현되었다.

자원 상태 모니터링(Resource State Monitoring)모듈은 그리드 자원의 각 노드에 설치된 리소스 에이전트(Resource Agent)들로부터 웹서비스를 통해 해당 노드의 상세 정보를 입력 받음으로써 해당 노드를 그리드 자원으로 편입시킨다. 해당 정보는 하드웨어 정보 및 운영체제와 같은 기본적인 정보들뿐만 아니라 관리자에 의해서 기입된 각종 소프트웨어 라이브러리 및 설치된 프로그램들의 목록과 같은 자세한 사항들을 전달함으로써 기존의 그리드 미들웨어에서 다루지 못했던 다양한 소프트웨어적 요구사항들을 다룰 수 있도록 한다. 또한 해당 모듈은 각 노드의 현재 상태를 TCP 포트를 이용하여 작동 유무 확인 및 그리드 미들웨어를 통하여 각 노드의 현재 상태를 지속적으로 모니터링 하는 역할을 한다.

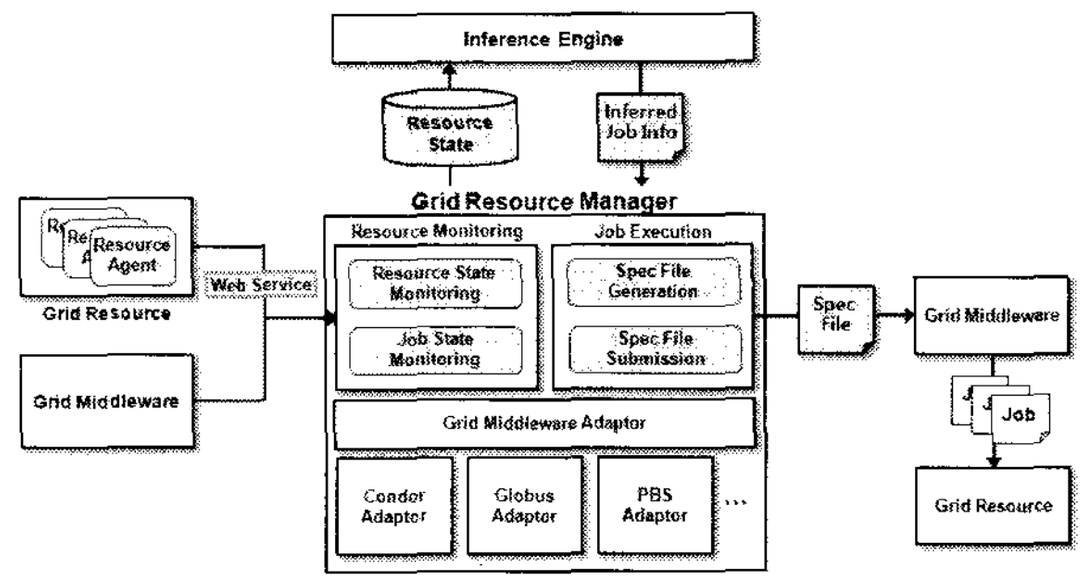

그림 3. 그리드 자원관리자 구조
Fig 3. Structure of Grid Resource Manager

작업 상태 모니터링(Job State Monitoring) 모듈은 현재 그리드 미들웨어에 던져진 작업들의 상태를 모니터링 함으로써 사용자로 하여금 작업의 정상적인 작동 여부를 확인할 수 있도록 한다. 이렇게 관리된 자원 상태 및 작업 상태 정보들은 하나의 자원 관리 테이블 형태로 관리된다.

또한 작업 실행(Job Execution) 모듈은 추론엔진으로부터 전달받은 작업 정보를 바탕으로 사용하려는 그리드 미들웨어에서 요구하는 형태로 정보를 작성하는 역할을 하는 명세 파일 생성자 (Specification File Generation) 모듈과 작성된 정보를 그리드 미들웨어에 전달하여 실제 작업이 수행되도록 하는 역할을 하는 명세 파일 제출자(Specification File Submission) 모듈로 나뉘어 진다. 또한 앞서 그리드 미들웨어에서 요구하는 형태로 정보를 변환하기 위하여 그리드 미들웨어 어댑터(Grid Middleware Adaptor) 모듈을 사용하였고 이 모듈은 그리드 미들웨어에 독립적으로 설계하여 시스템의 확장성을 높이도록 하였다.

추론엔진(Inference Engine)은 해당 시스템의 가장 핵심적인 역할을 하는 모듈로써 각 관리자로부터 전달받은 정보들 및 온톨로지 저장소에 있는 응용프로그램 정보를 바탕으로 다음과 같은 모듈들을 거치게 된다.

온톨로지 데이터 추론(Ontology Data Inference) 모듈은 온톨로지 저장소에서 관리되고 있는 응용 프로그램 온톨로지 정보들을 단순히 가져오는데 그치지 않고 특정 분석을 하는데 사용될 수 있는 프로그램 추론, 특정 데이터를 분석하는데 사용될 수 있는 프로그램 추론 등의 온톨로지 추론 기능을 가지고 있다. 이러한 추론을 위하여 Pellet[17]이라는 기존의 DL(Description Logic) Reasoner를 DIG Interface를 통해 연결하여 사용을 하게 된다.

최적 프로그램 선택(Appropriate Program Selection) 모듈은 작업 기술파일에 수행하고자 하는 특정 프로그램이 아닌 수행하고자 하는 특정 분석과 그 분석의 특성에 대한 정보만이 주어졌을 때 앞서 서술된 분석 온톨로지의 "Analysis Property"에 저장되어 있는 해당 분석의 특성정보와 매칭을 시켜 공통적으로 포함되는 여러 프로그램들 중 가장 적합한 프로그램을 선택하여 실행시키는 기능을 한다. 이와 같은 모듈은 해당 분석에 대한 자세한 지식이 없는 사용자가 간단히 분석을 수행해보고자 할 때 유용하게 사용될 수 있다.

자원 요구량 추측(Resource Requirement Prediction) 모듈은 해당 프로그램을 실행하는데 있어 입력으로 들어온 데이터의 크기 및 인수에 따라서 필요한 각 자원의 요구량을 대략적으로 추측하기 위한 모듈로써 전체적인 추측 방법은 그림 4와 같다. 자원 요구량은 기본적으로 앞서 서술된 프로그램 온톨로지의 "DefaultRequirement" 클래스에 저장된 CPU, 메모리, 데이터 사이즈 등과 같은 각 자원 별 기본 요구사항을 갖게 된다.

하지만 만약 "InputDataSizeFactor" 클래스와 "Execution OptionFactor" 클래스가 정의되어 있고 입력으로 들어온 데이터의 종류 및 크기 그리고 실행 옵션이 매칭되는 것이 있을 경우 각각의 매칭되는 가중치(Weight Factor)들을 각각의 자원 별로 모아서 산출하는 방식이다. 이때 자원 요구량의 값은 0 ~ 5까지의 정량화된 값을 가지게 되고 만약 5이상이 넘는 경우는 해당 프로그램을 보통의 노드 하나에 실행하기에는 무리가 있는 작업이라고 판단하고 사용자에게 경고 메시지를 보내거나 앞서 최적 프로그램 선택 모듈이 실행됐을 경우 다음으로 적합하다고 판단되는 프로그램을 선택하여 자원요구량을 다시 계산하게 된다.

최적 노드 할당(Appropriate Node Matching) 모듈은 그리드 자원 관리자로부터 그리드 자원에 속한 각 노드의 정보를 담고 있는 자원 관리 테이블과 앞서 온톨로지 데이터에서 가져온 하드웨어적 소프트웨어적 요구사항들 그리고 앞서 추측된 각 자원에 대한 요구량을 바탕으로 해당 작업을 수행하는데 적합한 노드들의 목록을 작성하는 역할을 한다.

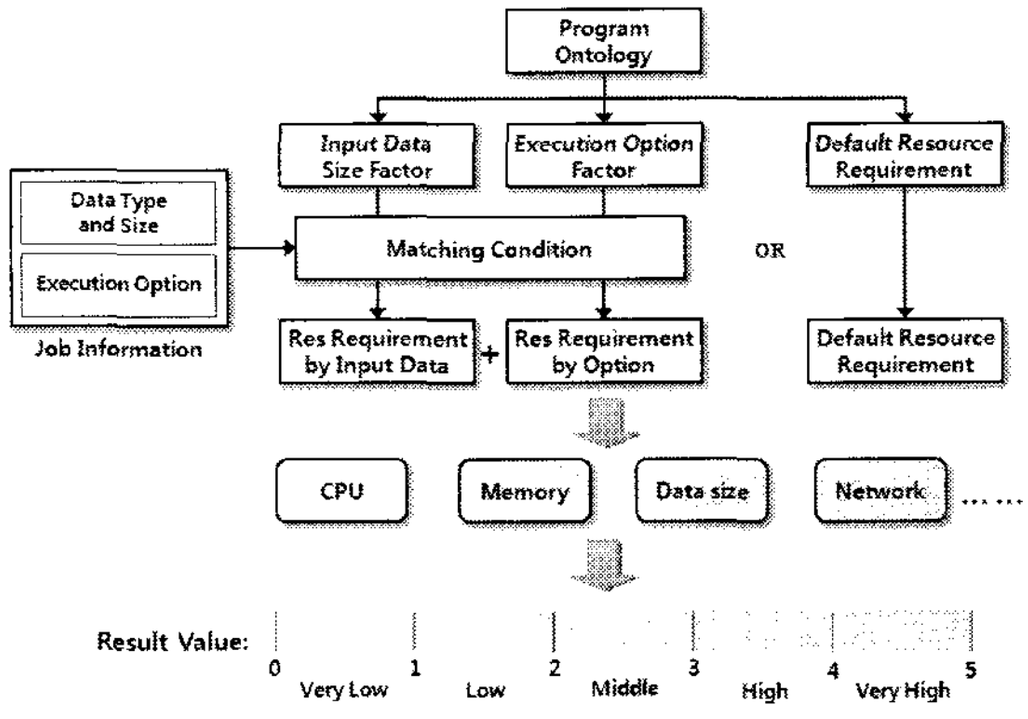

그림 4. 자원 요구량 추측 방법
Fig 4. Method to predict resource requirement

이렇게 각 모듈을 거쳐 작성된 하드웨어적 소프트웨어적 요구사항들과 이에 적합한 노드들의 목록 그리고 입력으로 들어온 작업 기술파일의 내용을 그리드 자원 관리자에게 넘겨주어 해당 작업이 그리드 미들웨어에서 실행되도록 한다.

Ⅳ. 평가

구현된 그리드 관리 시스템을 평가하기 위하여 생물정보학 분석 환경 구축에 실제 적용해본 후 이를 온톨로지 측면과 그리드 사용 측면에서 기존 시스템들과 비교해보았다.

생물정보학 분석 환경 구축을 위하여 다양한 분석들 중 서열정렬 분석이 선택되었다. 서열정렬은 염기 서열 혹은 단백질 서열들 간의 유사성을 분석하는 것으로 생물정보학의 가장 기본적인 분석으로써 중요한 의미를 지니고 있다. 또한 해당 분석은 대용량의 유전자 서열들을 다뤄야 함으로써 높은 연산량을 필요로 하기 때문에 그리드 환경에 적합한 분석이다.

해당 분석환경 구축을 위하여 가장 먼저 그림 5 와 같은 응용 프로그램 온톨로지를 기존 온톨로지 데이터에 추가하였다. 그림 왼쪽의 계층적 구조는 서열정렬과 관련한 정보를 분석 온톨로지에 적용한 그림이고, 오른쪽은 각각의 분석에 포함되는 프로그램 온톨로지의 객체로서 입력된 프로그램들의 목록이다.

또한 분석 환경 구축을 위하여 20여대의 노드들로 구성된 그리드 자원을 구축하였고 이 노드들을 관리하는 미들웨어로써 Condor[18] (Version 6.6.10)를 사용함으로써 다양한 서열정렬 프로그램들을 그리드 기반으로 수행할 수 있는 환경을 마련하였다.

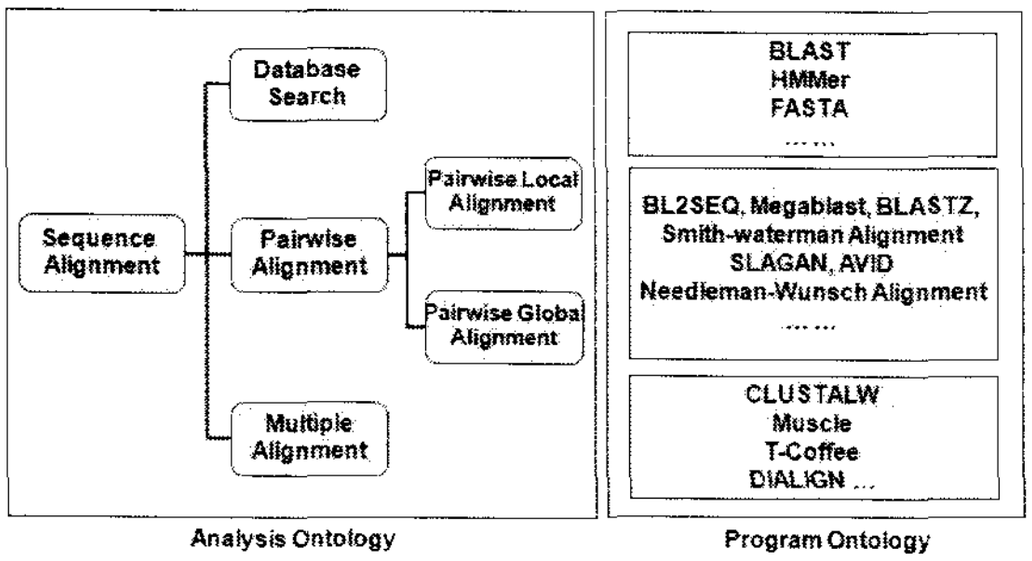

그림 5. 서열정렬 분석 환경 구축을 위해 구현된 온톨로지
Fig 5. Ontology for sequence alignment

구축된 분석환경을 바탕으로 기존의 여러 그리드 프로젝트들을 구축하기 위해 설계된 다양한 온톨로지 데이터들 중 해당 연구와 가장 유사하다고 판단된 과학 정보(Scientific Information)를 담기 위한 온톨로지를 설계하고 이를 ESG(Earth System Grid)에 적용한 사례[19]와 비교해 보았다.

다음 표 2는 해당 시스템들에 사용된 온톨로지를 서로 비교한 내용이다. 과학 정보 온톨로지는 과학 분야의 특정 분석을 위한 전체 과정을 Pedigree 클래스에 의해 Project로 정의하고 각각의 과정을 구성하는 분석, 실험, 관찰들의 정보를 표현하는 Scientific_Use 클래스, 그리고 각각의 과정에서 사용되는 데이터를 표현하는 Dataset 클래스를 핵심으로 한다. 표에서도 서술된 바와 같이 해당 온톨로지는 각 과정을 연결해서 여러 연구에 사용되는 데이터들을 어떻게 다룰 것인지를 기술하는데 그 초점을 두고 있다.

이를 응용 프로그램 온톨로지와 비교하면 해당 온톨로지는 분석 과정을 기술하는 데 초점을 두었기 때문에 전체 분석 과정 설계에는 효율적일 수 있지만 각각의 과정에서 수행되는 실제 프로그램들의 다양한 정보를 기술하는 데는 다소 부족함이 있다. 또한 전체 분석 과정을 온톨로지를 바탕으로 세세히 기술해야 하므로 큰 프로젝트가 아닌 작은 단위의 보통의 분석들에 일일이 적용하기에는 한계가 있다는 것을 알 수 있다.

그에 반해 응용 프로그램 온톨로지는 이미 서술된 바와 같이 프로그램 온톨로지를 이용하여 응용 프로그램이 그리드 환경에서 수행되는데 필요한 사항들을 기술하는 데 초점을 맞춤으로써 좀 더 다양한 요구사항들을 다룰 수 있고, 데이터 온톨로지, 분석 온톨로지를 통해서 해당 프로그램들을 분석 종류 및 다룰 수 있는 데이터에 따라서 관리가 가능하므로 효율적인 프로그램 관리가 가능하다. 또한 해당 응용 프로그램의 정보를 온톨로지에 추가해 놓으면 간단한 분석 혹은 복잡한 분석 과정의 일부로써 사용이 가능하다.

표 2. 응용 프로그램 온톨로지와 과학 정보 온톨로지의 비교
Table 2. Comparison between application software ontology and scientific information ontology

비교 항목	응용 프로그램 온톨로지	과학 정보 온톨로지
온톨로지 타겟	그리드 기반 응용 프로그램 정보	그리드 기반 과학 정보
대표 클래스	Program, Analysis, Data	Pedigree, Scientific_Use, Dataset, Service, Access
온톨로지 초점	응용 프로그램의 각종 요구사항 기술	다양한 데이터들을 어떤 과정을 통해서 다룰 것인지를 기술

그리드 사용 측면에서 기존 시스템들과 비교를 하기 위하여 BLAST[20] 프로그램을 이용한 서열정렬 분석을 진행하였다. BLAST의 경우 생물정보학에서 가장 널리 사용되는 서열정렬 프로그램이나 알고리즘의 특성상 대용량의 서열을 서로 비교 분석하기에는 어려운 특성을 가지고 있다. 따라서 이를 극복하기 위해 입력 서열을 적당한 크기(100k bp)로 잘라 각 노드에서 분석 가능케 하고 이 결과를 모아 후처리할 수 있는 Perl script 기반 BLAST 분석 프로그램을 구현하였다.

이렇게 구현된 프로그램을 이용하여 NCBI (National Center for Biotechnology Information)에서 제공 중인 유전체 서열들 중 서열간의 유사성이 높다고 알려진 인간 염색체 20번(62M bp)과 쥐 염색체 2번 서열(182M bp)을 정렬하는 분석을 진행하였다.

다음 그림 6은 상기의 분석 과정을 기존 그리드 미들웨어 혹은 그리드 관리 시스템들을 이용하여 분석을 진행하는 과정과 구현된 응용 프로그램 온톨로지 기반 시맨틱 그리드 관리 시스템을 적용하여 진행하는 과정을 비교한 것이다.

그림에서 설명된 바와 같이 해당 시스템을 적용함으로써 기존 매 작업마다 미리 수행되어야만 했던 프로그램의 요구사항들을 파악하고 그리드 자원에 속한 각 노드들이 이러한 요

구사항과 매칭되는지를 살펴본 후 사용하려는 그리드 미들웨어에 맞게 명세 파일을 작성하는 일련의 과정들이 줄어들게 된다.

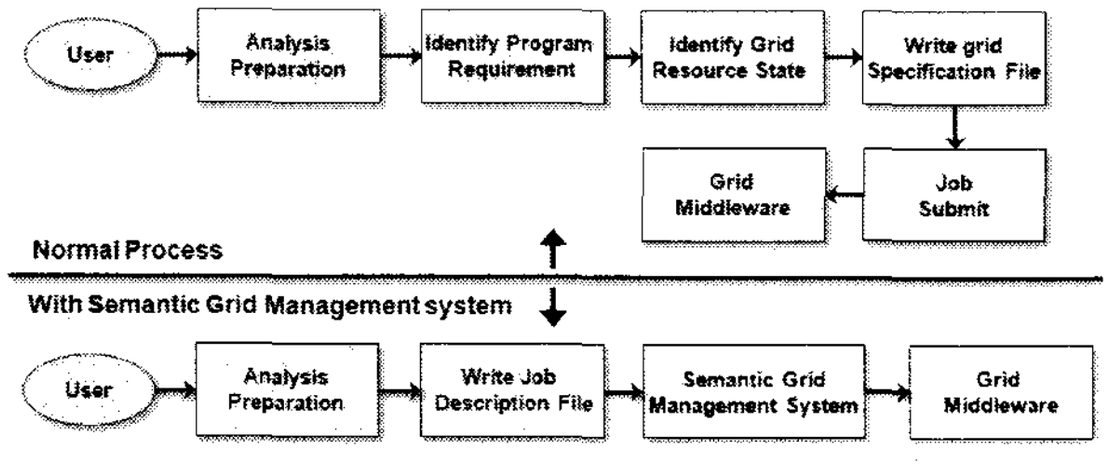

그림 6. 기존 분석 과정과 해당 시스템 기반의 분석과정 비교
Fig 6. Comparison between analysis process based on the semantic grid management system and normal process based on current grid middlewares

특히 이번 분석과 같은 경우는 Perl기반의 프로그램이기 때문에 수행하려는 노드에 특정 버전이상의 Perl이 설치되어 있어야만 한다. 따라서 기존시스템은 이러한 자세한 소프트웨어적 요구사항을 다루지 못하기 때문에 사용자가 직접 그리드 자원에 포함된 노드들의 정보를 파악하고 이에 적합한 노드들의 목록을 파악하여 그리드 미들웨어가 요구하는 형식에 맞게 건네줘야만 했다.

다음 표 3은 시맨틱 그리드 관리 시스템을 사용하기 위해 작성되는 작업 기술파일(Job Description File)과 그리드 미들웨어를 사용하기 위해서 작성되는 그리드 명세 파일에 기술해야하는 정보를 비교한 내용이다.

표 3. 작업 기술파일과 그리드 명세파일 비교
Table 3. Comparison between job description file and grid specification file

작업 기술파일	그리드 명세파일
실행파일 정보	실행파일 정보
인수 정보	인수 정보
전송파일 정보	전송파일 정보
프로그램 이름	적합한 노드의 목록
입력 데이터 정보	자원 요구사항

두 파일의 내용에서 볼 수 있듯이 그리드 환경에서 응용 프로그램을 수행하기 위해서는 실행파일 및 인수정보 그리고 실행 파일과 함께 전송되어야만 하는 파일들의 목록이 공통됨을 알 수 있다.

하지만 그리드 명세파일의 경우는 어떤 노드에서만 해당 작업이 실행되어야 하는지 그리고 해당 작업이 각각의 자원에 대한 어떠한 요구사항을 가져서 작업을 어떻게 할당해야만 하는지에 대한 정보를 기술해야만 한다. 가령 현재 사용 중인 Condor의 예를 들어 설명하면, 우선 Submit Description File을 작성하고 Requirements 옵션을 이용하여 상기의 적합한 노드들의 목록과 해당 프로그램의 하드웨어적 그리고 운영체제와 관련한 요구사항들을 Condor에서 정의한 ClassAd 표현 방식을 이용하여 기술해야 한다. 또한 해당 작업이 특정 자원을 많이 요구하는 작업이라면 Rank 옵션을 이용하여 특정 자원에 대한 높은 노드들을 위주로 할당되도록 해야 한다. 특히 해당 서열정렬 분석은 높은 연산량과 메모리를 요구하기 때문에 CPU 및 메모리 성능이 높은 자원에 먼저 할당되도록 해야 한다.

그에 반해 작업 기술파일의 경우는 온톨로지 데이터에서 해당 프로그램의 정보를 가져오기 위한 프로그램의 정식 명칭과 입력으로 들어가는 데이터들의 정보만을 기술하면 된다.

또한 입력 데이터 정보 및 인수 정보를 기술하였기 때문에 입력 데이터의 종류 및 크기 그리고 인수에 따라서 각 자원에 대한 요구량의 정도를 상기 언급된 추론엔진을 통해 대략적으로 추측할 수 있어 좀 더 효율적인 작업 할당을 가능케 한다. 특히 해당 분석과 같은 경우는 잘려진 각 데이터의 사이즈가 동일하다 하더라도 비교 대상인 염색체 서열에는 유의미한 "A, T, G, C" 서열을 제외하고 아직 서열이 밝혀지지 않아 무의미한 N으로 표시된 긴 서열들을 포함하고 있다. 따라서 이런 무의미 서열을 가지고 있는 작업들은 실제 분석해야할 데이터의 사이즈가 작아짐으로써 다른 작업들 보다 낮은 자원 요구량을 가지게 된다. 이 작업들은 추론엔진의 자원 요구량 추측 모듈에 의해서 상대적으로 낮은 자원 가중치 값을 갖게 되고, 이 정보가 그리드 자원 관리자의 명세파일 생성자에 건네져 높은 가중치 값들을 가지는 작업들을 CPU와 메모리가 더 높은 자원에 먼저 할당되도록 그리드 명세파일이 작성되게 된다.

따라서 해당 시스템은 기존 그리드 미들웨어 혹은 관리 시스템과 비교하여 상기와 같은 분석과정의 간소화를 가능케 함으로써 분석 작업의 효율성을 높일 수 있고, 사용자가 매 작업 시 기술해야 하는 정보를 그리드 시스템에 독립적인 분석과 관련한 기본적인 정보만을 작성케 함으로써 쉽게 사용이 가능하여 사용자의 편의성을 높일 수 있다.

마지막으로 응용 프로그램 온톨로지 기반 시맨틱 그리드 관리 시스템을 기반으로 수행된 상기 BLAST 기반 서열정렬

의 결과를 설명하면 인간 20번 염색체와 쥐2번 염색체를 정렬하는데 20여대의 노드를 이용하여 약 450분 정도가 걸렸다. 또한 정렬 결과로써 BLAST의 옵션 중의 하나인 expectation value의 값을 0.1로 설정하였을 경우 1,086,963 HSPs(High Scoring Pairs 혹은 정렬 결과)를 얻게 되었다. 이렇게 나온 결과물을 상기 설명된 Perl 기반의 후처리 과정을 거쳐 가장 의미 있을 것이라고 생각되어지는 30,673 정렬 결과들을 얻게 되었고 이를 gnuplot[21]을 이용하여 2차원 그래프 형태로 정렬 결과로 도시화한 것이 그림 7이다.

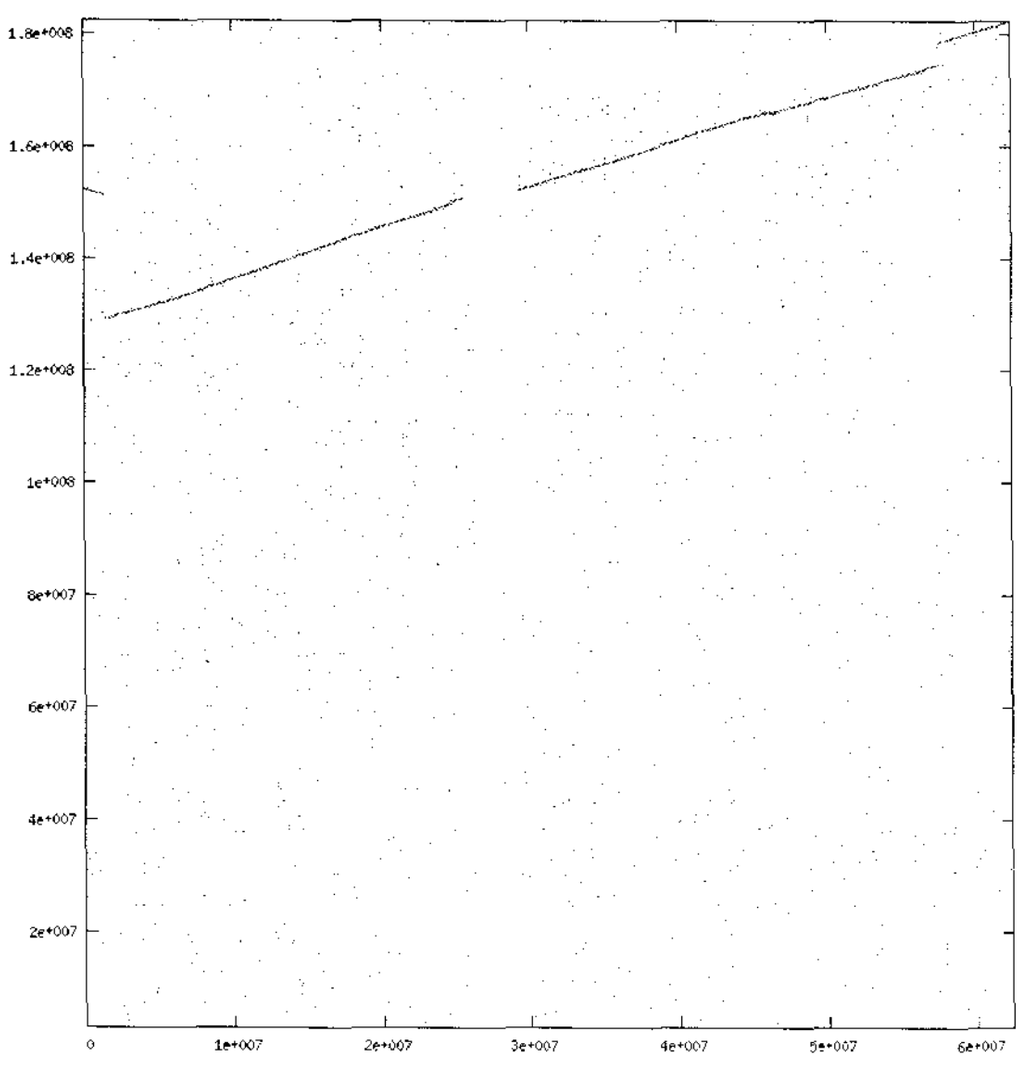

그림 7. 인간 20번 염색체와 쥐 2번 염색체를 정렬한 결과
Fig 7. Result of aligning human 20 chromosome to mouse 2 chromosome

이 그래프에서 x좌표는 정렬 결과가 나타난 인간 20번 염색체에서의 위치이고, y좌표는 쥐 2번 염색체에서의 위치를 의미한다.

V. 결론

본 연구에서는 기존 그리드 연구에서 제대로 다뤄지지 못했던 실제 그리드 환경에서 수행되는 응용 프로그램이 가질 수 있는 다양한 정보를 온톨로지 형태로 구축하고 이를 이용한 시맨틱 그리드 관리 시스템을 제안하였다. 또한 생물정보학의 가장 기본적이면서도 중요한 분석인 서열정렬 분석환경 구축에 실제 적용해 보고 이를 기존 시스템들과 비교함으로써 유용성을 설명하였다.

이렇게 구축된 시스템을 통하여 사용자는 그리드 미들웨어에 대한 깊은 지식이 없더라도 비교적 간단한 형태의 작업 기술파일을 작성함으로써 손쉽게 그리드 환경에서 원하는 작업들을 실행할 수 있도록 하여 사용자의 편의성을 증가시킬 수 있다.

여기에 분석 프로그램을 그리드 환경에서 실행하기 위해 기존 사용자가 일일이 고려해야만 했던 다양한 요구사항들 특히 기존 그리드 미들웨어에서 다루기 힘들었던 다양한 소프트웨어적 요구사항들 까지도 각 노드에 설치된 리소스 에이전트들을 이용한 자원 정보 수집과 추론엔진을 통한 최적 자원 할당을 통해 관리 시스템에서 다뤄질 수 있도록 하였다.

또한 응용 프로그램과 관련된 다양한 정보를 온톨로지 형태로 구축함으로써 체계적인 관리가 가능하게 됨은 물론 수정의 용이성과 확장의 용이성 등 온톨로지 언어가 가지는 장점들을 얻을 수 있었다. 또한 구축된 응용 프로그램 온톨로지는 단순히 그리드 사용에 한정된 것뿐만 아니라 지금도 개발되고 있는 수많은 응용 프로그램들을 체계적으로 정리할 수 있는 하나의 방법으로써 제공될 수 있다. 여기에 온톨로지 추론 기능을 이용하여 특정 분석에 적합한 프로그램 추론 및 특정 데이터 혹은 특정 프로그램의 결과물을 분석하는데 사용되는 프로그램 추론 등과 같이 사용자에게 참고할 수 있는 유용한 정보를 제공해 줄 수도 있다.

그리드 미들웨어 측면에서는 추론 엔진으로부터 전달 받은 분석 작업에대한 각 자원의 대략적인 가중치 값을 이용하여 동시에 들어온 작업들이라 하더라도 가중치 값이 큰 작업들을 적합한 자원에 먼저 할당함으로써 효율적인 작업할당을 가능하도록 할 수 있다. 또한 기존의 몇 가지로 단순화된 작업의 특성을 바탕으로 이를 반영하여 전체 수행시간을 단축시키려 했던 연구들이 지녔던 한계를 벗어나 프로그램들에 따라 심지어는 같은 프로그램이라도 입력 데이터의 종류 및 크기와 인수에 따라서 연산량 등이 달라지는 것 등을 반영함으로써 좀 더 현실에 맞게 전체 수행시간을 단축시킬 수 있는 성능향상의 기회를 제공할 수 있다.

향후 연구에서는 다양한 응용 프로그램들을 온톨로지에 추가하는 과정에서 나타나는 다양한 특성들을 파악하고, 이를 온톨로지 데이터 구조에 반영시켜 나감으로써 시스템을 좀 더 범용 적으로 사용가능하게 만들도록 한다. 또한 시스템의 입력 파일인 작업 기술파일을 좀 더 쉽게 작성할 수 있도록 도와주는 사용자 인터페이스와 API를 구축할 예정이다.

참고문헌

[1] I. Foster, J. Geisler, W. Nickless, W. Smith, S. Tuecke "Software Infrastructure for the I-WAY High Performance Distributed Computing Experiment", 5th IEEE Symposium on High Performance Distributed Computing, pp. 562-571, 1997.

[2] FAFNER, http://www.npac.syr.edu/factoring.html

[3] I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", International Journal of Supercomputer Applications, Vol. 11, No. 2, pp.115-128, 1997.

[4] Andrew S. Grimshaw, Wm. A. Wulf, and the Legion team, "The Legion Vision of a Worldwide Virtual Computer", Communications of the ACM, Vol. 40, No.1, 1997.

[5] D. De Roure, N.R. Jennings, and N. Shadbolt, "The Semantic Grid: A Future e-Science Infrastructure", Grid Computing: Making the Global Infrastructure a Reality, F. Berman, A.J.G. Hey, and G. Fox, eds., John Wiley & Sons, pp. 437 - 470, 2003.

[6] E. M. Frank Manloa, "RDF primer". W3C, 2004.

[7] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, and L.A. Stein, "OWL web ontology language references", W3C, 2004.

[8] D.R. David, N.R. Jennings, N.R. Shadbolt, "The Semantic Grid: Past, Present, and Future", Proceedings of the IEEE, pp. 669-681, 2005.

[9] A.M. Pernas and M.A.R. Dantas, "Using ontology for description of grid resources", Proceedings of the 19th International Symposium on High Performance Computing Systems and Applications, pp 223-229, 2005.

[10] H. Tangmunarunkit, S. Decker and C. Kesselman, "Ontology-based Resource Matching - The Grid meets the Semantic Web", Second International Semantic Web Conference (ISWC), Sanibel Island, Florida, USA, 2003.

[11] H. L. Kim, T.N. Kim, C.S. Jeong, "Grid Resource Management System and Semantic Information System", 22nd International Conference on Advanced Information Networking and Applications, 2008.

[12] J. Luo, P. Ji, X. Wang, Y. Zhu, F. Li, T. Mas, and X. Wang, "Resource management and task scheduling in grid computing" Computer Supported Cooperative Work in Design, 2004. Proceedings. The 8th International Conference on Volume 2, pp.431-436, 2004.

[13] 박다혜, 이종식, "계산 그리드 컴퓨팅에서의 자원성능 측정을 통한 그리드 스케줄링 모델", 한국컴퓨터정보학회지, 제11권, 제5호, pp.87-94, 2006.

[14] K. L. Park, H. J. Lee, Y. J. Lee, O. Y. Kwon, S. Y. Park, H. W. Park, and S. D. Kim, "An Efficient Collective Communication Method for Grid Scale Networks", International Conference on Computational Science, pp.819-828, 2003.

[15] 조수현, 김영학, "계산 그리드 상에서 프로그램의 특성을 반영한 작업 프로세스 수의 결정에 관한 연구",한국컴퓨터정보학회논문지, 제11권, 제1호, pp.72-85, 2006.

[16] Storey, M.-A., N.F. Noy, M. Musen, C. Best, and R. Fergerson, "Jambalaya: an interactive environment for exploring ontologies", in Proceedings of the International Conference on Intelligent User Interfaces, pp. 239, 2002.

[17] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur and Y. Katz. "Pellet: A practical OWL-DL reasoner", Journal of Web Semantics, Vol. 5, No.2, pp.51-53, 2007.

[18] Condor, http://www.cs.wisc.edu/condor/

[19] L. Pouchard, et.al. "An Ontology for Scientific Information in a Grid Environment: the Earth System Grid" In Proc. of the 3th IEEE/ACM International Symposium on Cluster Computing and the Grid, pp 626-632, 2003.

[20] S.F. Altschul, W. Gish, W. Miller, E.W. Myers and D.J. Lipman, "Basic local alignment search

tool", Journal of Molecular Biology, 215, pp. 403-410, 1990.
[21] Gnuplot, http://www.gnuplot.info/

저 자 소 개

김 민 성

2005년 한국정보통신대학교 공학부 졸업(학사)
2005년~현재 한국정보통신대학교 공학부 석박사 통합과정
관심분야: 그리드 컴퓨팅, 생물정보학

이 관 수

1988년 서울대학교 동물학과 졸업(학사)
1990년 한국과학기술원 생물공학과 졸업(석사)
1993년 한국과학기술원 생물공학과 졸업(박사)
2002년~현재 한국정보통신대학교 공학부 부교수
관심분야: 그리드 컴퓨팅, 생물정보학