

온톨로지 기반의 그리드 자원선택 시스템

노창현*, 장성호*, 김태영*, 이종식**

Ontology-based Grid Resource Selection System

Chang Hyeon Noh *, Sung Ho Jang *, Tae Young Kim *, Jong Sik Lee **

요약

그리드 컴퓨팅 환경에서 컴퓨팅 자원은 매우 다양한 네트워크와 시스템으로 구성되어 있다. 이기종의 환경에서 기존의 자원선택 기법으로 사용자가 원하는 자원을 검색 및 선택하는 것은 자원정보의 저장 구조상 한계가 있다. 본 논문은 사용자의 요구사항과 데이터 특성에 맞는 자원을 선택하기 위해 그리드 자원을 온톨로지로 구축하고, SWRL을 이용하여 정의한 규칙을 바탕으로 추론 엔진을 거쳐서 자원을 선택 및 제공하는 온톨로지 기반의 그리드 자원선택 시스템을 제안한다. 실험 결과는 본 논문에서 제안한 온톨로지 기반의 그리드 자원선택 시스템이 기존 그리드 자원선택 시스템인 Condor-G와 Nimrod-G 보다 더 높은 작업 처리율 및 자원 이용률과 적은 작업 손실 및 처리 시간을 보임으로써 그리드 자원선택을 지능적이며 능동적으로 할 수 있고, 자원 이용에 더 효과적이라는 사실을 증명한다.

Abstract

Grid resources are composed of various communication networks and operation systems. When a grid system searches and selects grid resources, which meet requirements of a grid user, existing grid resource selection systems are limited due to their storage methods for resource information. In order to select grid resources suitable for requirements of a grid user and characteristics of data, this paper constructs an ontology for grid resources and proposes an ontology-based grid resource selection system. This system provides an inference engine based on rules defined by SWRL to create a resource list. Experimental results comparing the proposed system with existing grid resource selection systems, such as the Condor-G and the Nimrod-G, verify the effectiveness of the ontology-based grid resource selection system with improved job throughput and resource utilization and reduced job loss and job processing time.

▶ Keyword : 온톨로지(Ontology), 추론(Inference), 그리드 자원선택(Grid Resource Selection)

• 제1저자 : 노창현

• 접수일 : 2008. 3. 13, 심사일 : 2008. 4. 10, 심사완료일 : 2008. 5. 24.

*인하대학교 정보공학과

**인하대학교 정보공학과 부교수

※ 이 논문은 인하대학교의 지원에 의하여 연구되었음.

I. 서론

인터넷과 같은 통신 수단의 발달로 다양한 종류의 많은 양의 데이터가 발생함에 따라 의미 있는 정보를 취득하기가 점점 더 어려워지고 있다[1]. 그리드 컴퓨팅에서 자원의 종류 및 양은 전 세계의 자원을 대상으로 하기 때문에 그 수는 헤아리기 어려울 정도로 많다. 이기종의 시스템과 네트워크로 수많은 종류의 자원으로 구성된 그리드 컴퓨팅 환경[2]에서 사용자가 자신이 원하는 자원을 하나씩 선택한다는 것은 쉬운 일이 아니며 수백~수천 개에 이르는 자원을 직접 선택하는 일은 불가능하다.

효율적으로 그리드 자원을 활용하기 위해서는 사용자가 그리드 자원을 사용 할 때 데이터가 처리 될 때까지 소요되는 시간, 저장 매체의 크기, 자원의 위치 및 네트워크의 대역폭, 지불 할 수 있는 금액[3]과 사용자가 선호하는 자원 등등을 고려하여 적합한 자원을 선택해야 한다. 하지만 현재 그리드 상에서 자원을 선택 할 수 있는 대표적인 그리드 미들웨어인 Condor-G[4]와 Nimrod-G[3]는 사용자의 이러한 요구사항들을 만족시킬 수 없다. 그리드 자원정보가 관계형 데이터 베이스에 저장 되어있는 구조상의 한계로 간단한 질의 및 검색으로는 다양한 형태의 사용자의 요구를 만족 시킬 수 없기 때문이다. 그래서 최근 몇 년 동안 가상조직(Virtual Organization)상에서 자원을 보다 더 쉽게 공유하고 사용[5]하기 위한 방안으로 온톨로지를 이용한 그리드 자원 구축 모형이 연구되어 왔다. 위에서 언급하였듯이 수많은 종류의 자원 중 그리드 사용자의 다양한 요구사항을 만족시키면서 데이터 특성에 맞는 자원을 선택하기 위해서는 그리드 자원정보 데이터에 의미 정보를 부여하고, 사용자의 선호 자원과 데이터 특성에 대한 연관 관계를 규칙으로 정의하여 지능적이고 고급 정보 처리가 가능하도록 그리드 자원정보를 온톨로지로 구축해야 한다. 구축 된 온톨로지로부터 사용자가 원하는 처리시간 및 비용, 네트워크 요소와 데이터 특성을 고려하여 원하는 자원을 추론하고 적합한 자원 목록을 제공함으로써, 사용자의 만족도를 개선시키고 전체 그리드 컴퓨팅 시스템의 성능을 향상 시킬 수 있다.

본 논문에서는 그리드 사용자에게 적합한 자원 검색 기능을 제공하고, 선택하기 위해서 그리드 자원 정보를 온톨로지로 구축하고 사용자 요구 사항과 데이터 특성을 입력 받아 데이터 값으로 추출한다. 구축한 온톨로지 재사용성을 고려하여 기존의 관계형 데이터베이스에 저장된 자원정보 값으로 인스턴스를 생성한다. 인스턴스들과 미리 정의한 규칙을 포함하여 추론

엔진을 거쳐 그리드 사용자에게 가장 적합한 자원 목록을 제공하는 온톨로지 기반의 그리드 자원선택 시스템을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구의 기초가 되는 기존의 그리드 자원선택 시스템과 프로티지(protégé) 온톨로지 구축 툴, 국내 기술로 만든 보쌈 추론엔진에 대하여 소개하고, 3장에서는 본 논문에서 제안하는 온톨로지 기반의 그리드 자원선택 시스템에 대하여 설명한다. 그리고 4장에서 시뮬레이션을 통해 성능을 평가하고 5장에서 결론을 맺는다.

II. 관련 연구

2.1 그리드 자원선택 시스템

그리드 컴퓨팅 환경에서 자원을 선택 할 수 있는 기능을 제공하는 대표적인 미들웨어는 Globus Toolkit과 Condor-G, Nimrod-G 등이 있다. Globus Toolkit은 MDS(Metacomputing Directory Service)를 이용하여 자원에 대한 정보를 그리드 사용자에게 제공[6]하지만, 간단한 검색 수준에 머물고 있으며 자원선택에 관한 방안을 제시하고 있지 않다[7]. Condor-G[4]는 자원이 광고하는 내용과 사용자가 요구하는 자원수를 매칭 시켜주는 역할을 하는 ClassAds Match-Making을 사용하지만 실제 자원이 광고하는 내용을 완전히 신뢰 할 수 없고, 하나의 자원만 관리 하므로 다수의 필요한 자원을 선택하기에는 한계가 있다. Nimrod-G[3]는 사용자의 작업이 처리되는 종료 시점 중심의 선택방법과 사용자가 그리드 자원을 사용 할 때 지불 할 수 있는 예산 중심의 선택 방법이 있다.

하지만 이러한 기존의 관계형 데이터베이스에 저장 된 그리드 자원정보를 바탕으로 한 검색 방법으로는 수많은 종류의 자원 중 그리드 사용자의 다양한 요구사항을 만족 시키면서 데이터 특성에 맞는 적합한 자원을 선택하기 어렵다. 따라서 본 논문에서는 기존의 관계형 데이터베이스에 저장 된 그리드 자원정보를 온톨로지로 구축하고, 사용자의 선호 자원과 데이터 특성에 따른 선택 자원에 대한 연관 관계를 규칙으로 정의하여 적합한 자원 목록을 추론해서 그리드 사용자에게 제공한다. 정확하고 적합한 추론 결과를 바탕으로 제공 된 자원 목록은 그리드 사용자의 만족도를 개선시키고, 지능적이고 자동적인 검색 기능을 제공 할 수 있다.

2.2 온톨로지 구축 도구 및 추론엔진

본 논문에서 그리드 자원선택을 위한 온톨로지를 구축하기 위해서 프로티지를 사용하였다. 프로티지[8]는 스탠포드 대학에서 개발한 지식기반 모델의 설계와 모델링 및 편집을 지원하는 오픈소스 무료소프트웨어이다. 프로티지[9]는 플랫폼에 독립적인 환경을 제공하고 사용자가 직관적으로 사용하기 쉽도록 클래스와 프로퍼티의 생성 및 수정을 GUI 환경으로 제공한다. 온톨로지 언어[10]인 RDF(Resource Description Framework)와 OWL(Web Ontology Language)을 모두 지원하며, 다른 온톨로지 툴로 확장이 용이하며 Java API 개발에 유연한 구조를 가진 장점이 있다.

보쌈(Bossam) 추론 엔진[11]은 한국전자통신연구원(ETRI)에서 국내 기술로 개발 한 RETE 기반의 Semantic Web 추론 엔진으로써 RDF 및 OWL로 표현 된 데이터를 규칙으로 생성하거나 추론을 실행하는데 용이하다. 보쌈 추론엔진[12]은 규칙 기반 추론 시스템으로써, 작은 프로그램의 크기로 규칙과 사실의 매칭에 대한 결론 도출 속도는 RETE 알고리즘을 이용하기 때문에 그 속도가 빠른 장점이 있다. 조건절에 논리합과 결론절에 논리곱을 허용해서 혼 논리의 표현력을 확장하여 간단하고 쉽게 규칙을 생성 할 수 있으며 클래스 및 상속 관계의 정의나 데이터 타입 속성을 정의하는 등, 프레임-슬롯의 객체지향 구조로 지식을 표현 할 수 있다. 규칙 내에서 자바 객체 메서드를 호출 할 수 있어서 간편하게 규칙 엔진과 어플리케이션간 연동이 쉬운 장점이 있다.

본 논문에서는 그리드 자원정보를 프로티지를 사용하여 온톨로지로 구축하였고, 보쌈 API를 이용해서 규칙을 생성한 후 추론 엔진을 거쳐 사용자에게 적합한 자원을 추론하여 제공한다.

III. 온톨로지 기반의 그리드 자원선택 시스템

그리드 사용자가 원하는 자원을 지능적이고 자동적으로 검색하는 기능을 제공하기 위해서 그리드 자원정보를 지식기반의 온톨로지로 구축하였다. 구축한 온톨로지를 바탕으로 그리드 자원정보와 사용자의 요구사항, 데이터의 특성에 따라 선호하는 자원으로 선택하는 자원을 규칙으로 정의한다. 우리는 규칙 기반으로 추론엔진을 거쳐 그리드 사용자에게 가장 적합한 자원 목록을 제공하기 위해서 사용자와 미들웨어 중간에 위치하여 지능적으로 적합한 자원을 검색 및 선택 할 수 있는 그림 1과 같은 온톨로지 기반의 그리드 자원선택 시스템(OGRSS : Ontology-based Grid Resource Selection System)을 제안한다.

3.1 데이터 추출 모듈(Data Extractor Module)

데이터 추출 모듈(Data Extractor Module)은 그리드 사용자가 처리하고자 하는 데이터를 전달 받아, 데이터의 특성과 사용자의 요구사항을 추출하여 중계부(Coordinator Module)로 전송하는 모듈이다. 사용자가 자원을 선택하고 요구사항과 데이터 속성을 결정하는 방법은 두 가지가 있다. 첫 번째 방법은 현재 처리하고자 하는 데이터와 요구하는 자원이 이미 이전에 처리했던 데이터의 속성, 선택했던 자원과 비슷하여 과거 선택자원 데이터베이스(Selection History

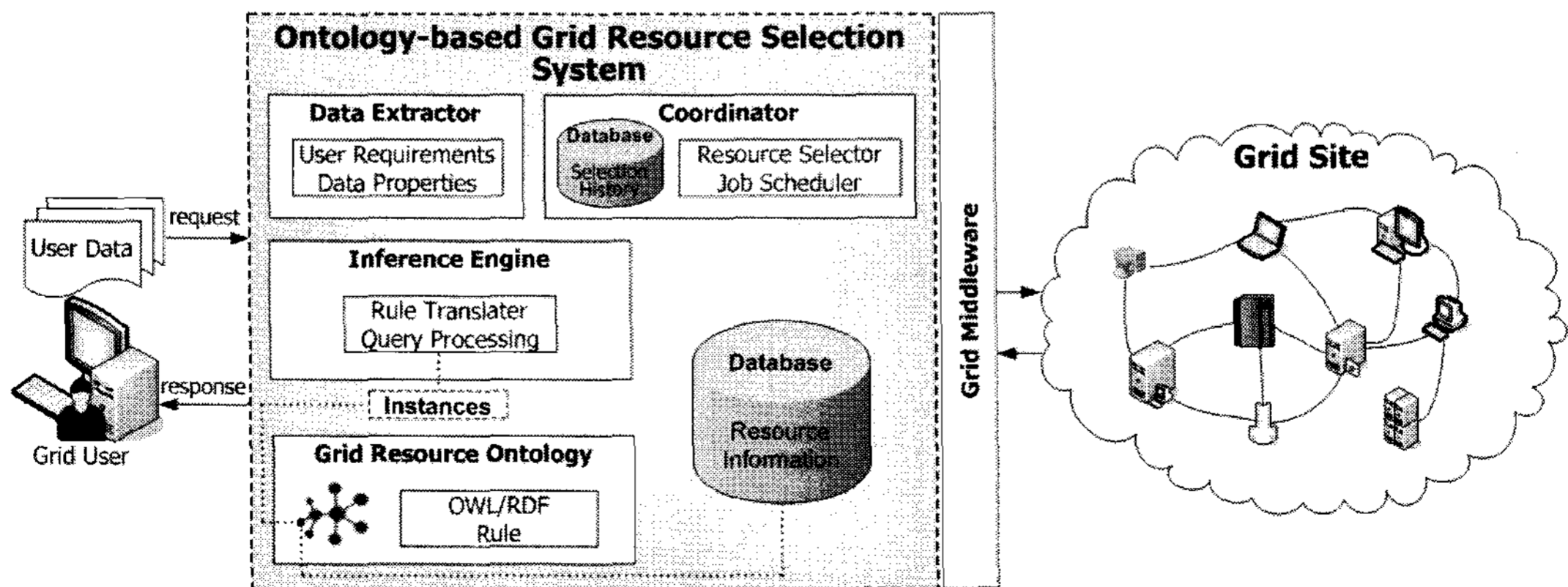


그림 1. 온톨로지 기반의 그리드 자원선택 시스템의 구조
Fig. 1. Ontology-based Grid Resource Selection System Architecture

DB)에 저장된 내용을 그대로 사용하거나 약간의 수정 과정을 거치는 방법이다. 이 경우 중계부는 규칙 기반의 추론 과정을 거치지 않고, 자원정보 데이터베이스(Resource Information DB)에서 선택하려는 자원이 가용한 상태인지 확인하고 가용한 상태일 경우 그 자원 목록으로 다음 작업을 진행한다. 만약 자원이 가용 상태가 아닐 경우 자원정보 데이터베이스로부터 선택되었던 자원 중 가장 비슷한 성능과 비용을 가지는 자원을 검색 및 선택하여 진행하는 방식이다.

두 번째 방법은 현재 사용자의 데이터 특성과 요구 사항이 이전에 선택했던 자원들과 달라서 새로운 자원들을 선택하는 것이다. 이때 사용자는 선택하고자 하는 자원에 대한 선호도와 지불할 수 있는 예산, 선택 할 자원의 개수, 종료 시점 등에 대한 정보를 입력하고, 입력된 정보는 정수 타입의 형태로 중계부에 전송된다. 사용자가 선택을 완료한 시점에는 과거 선택자원 데이터베이스에 사용자의 요구 사항과 데이터 특성 및 선택 자원 목록을 일정 개수 저장한다.

3.2 그리드 자원선택을 위한 온톨로지

그리드 사용자가 선호하는 자원과 데이터 특성에 따른 규칙 기반의 자원선택 시스템을 구축하기 위해서는 우선 그리드 자원정보를 온톨로지로 구축해야 한다. 우리는 기존의 관계형 데이터베이스에 저장된 자원정보를 바탕으로 프로티지(v3.3.1)를 이용하여 그리드 자원 온톨로지를 구축하였다. 자원정보 데이터베이스는 그리드 미들웨어와 연결되어 있어서 새로운 자원이 발견되면 등록하고, 자원들의 속성 값이 변경 되면 갱신하며 그리드 사이트에서 자원이 소멸 되었을 때 삭제하는 역할을 한다. 또한 현재 사용 중인 자원과 작업 처리를 위해 대기 중인 자원들에 대한 상태 정보를 가지고 있다.

우리는 OGRSS에서 자원을 그리드 컴퓨팅의 종류에 따라 나누었는데 계산 그리드(Computational Grid)를 위한 ComputationalResource와 데이터 그리드(Data Grid)를 위한 DataResource로 구분한다. 그림 2는 본 논문에서 구축한 그리드 자원의 온톨로지로서 클래스와 서브-클래스의 관계를 나타낸다.

본 논문에서는 계산 그리드 환경에서 처리하려는 데이터가 대용량으로 빠른 처리 속도를 요구하는 자원이기 때문에 ComputationResource 자원에 대해서 설명한다. 계산 그리드 자원 클래스인 ComputationalResource는 추상 클래스로써 DesktopComputer, Notebook, Server, Cluster 클래스를 서브 클래스로 갖는다. 이 서브 클래스들의 인스턴스는 ComputationalResource의 속성을 상속 받아 추가하거나 일부 선택하여 생성되는데, 이 때 인스턴스들의 속성 값은 구축되어 있는 기존의 관계형 데이터베이스에 저장된 속성

값을 로드하여 생성한다. 표 1은 ComputationalResource 클래스의 속성 이름에 대한 설명과 데이터 타입, 최소 값, 단위를 나타낸다.

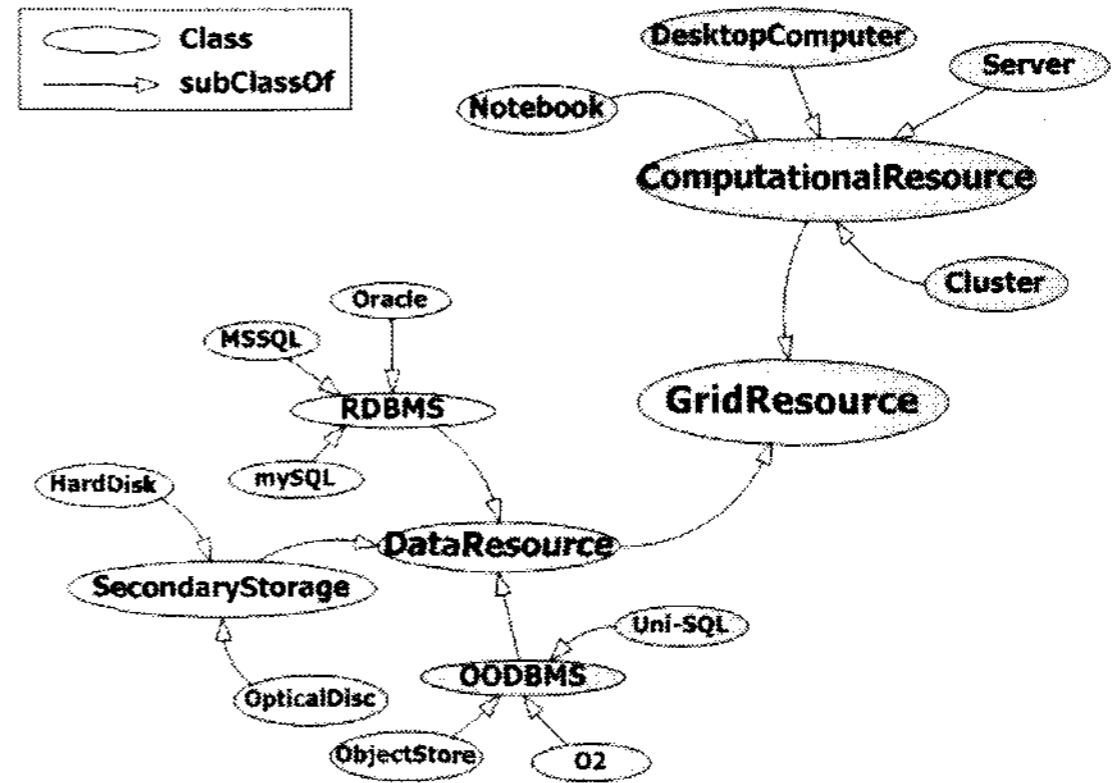


그림 2. 그리드 자원 온톨로지
Fig. 2. Grid Resource Ontology

표 1. ComputationalResource 클래스의 속성
Table 1. Property table : ComputationalResource Class

Property Name	Description	Data Type	Min. Value	Unit
id	자원 식별자	Integer	0	---
owner	자원의 소유자	String	---	---
organization	자원이 속한 소속기관	String	---	---
location	자원의 위치	String	---	---
OperatingSystem Type	OS의 종류	String	---	---
CPUArchitecture Type	CPU의 구조	String	---	---
CPU Speed	CPU 최대 클럭 속도	Float	700	Megahertz
cache	cache 메모리의 크기	Float	256	KiloByte
memorySize	주 기억장치의 크기	Float	32	MegaByte
diskStorageSize	보조 기억장치의 크기	Float	10	GigaByte
topology	네트워크의 구성 형태	String	---	---
bandwidth	네트워크 최대 대역폭	Float	---	MegaBPS
cost	자원의 사용 금액	Integer	1	Griddollar

3.3 추론을 통한 자원선택

본 논문에서 제안하는 OGRSS에서 자원선택 방법은 다음과 같은 과정을 거친다. 그리드 자원 온톨로지로 생성한 OWL 파일을 JAVA API로 импорт 한다. ComputationalResource 클래스의 서브 클래스들 즉, 계산 그리드 자원의

인스턴스들은 자원정보 데이터베이스에 저장된 속성 값을 프로퍼티 값으로 생성한다. 사용자가 선호하는 자원들에 대한 데이터 값과 데이터 특성 값을 입력 받아 데이터 추출 모듈이 정수 형태로 중계부에게 전송한다. 중계부의 자원선택 모듈(Resource Selection Module)이 데이터 값을 전달 받아 정의된 룰을 포함하고 추론 엔진을 통해 전체 자원을 정제하고 매칭되는 자원을 사용자가 요구하는 개수만큼 선택한다.

동일한 성능을 가진 자원이 여러 개 선택 되었을 경우 사용자의 요구사항을 바탕으로 정의된 룰에 의해 필요한 만큼만 재 선택한다. 만약 사용자가 원하는 개수의 자원을 취득하지 못했을 경우에는 OGRSS가 선택 한 자원을 사용자에게 보여주고 그 자원으로 선택 할 것인지에 대한 질의를 실행한다. 이후 사용자는 OGRSS가 선택 한 자원을 사용할 것인지에 대해 결정하거나 사용 할 일부 자원 목록을 결정한다. 또는 자신이 원하는 자원의 선호도와 요구사항 정보를 조정 및 수정하여 전송하는 과정을 거친다. 이렇게 최종적으로 사용자가 선택한 자원을 확인 및 승인하는 과정을 거친 후, 중계부가 선택된 자원 목록을 과거 선택자원 데이터베이스에 저장한다. 표 2는 SWRL(Semantic Web Rule Language) 형태로 나타낸 자원선택을 위해 본 논문에서 정의한 규칙과 설명을 나타낸 몇 가지 예이다.

3.4 중계부(Coordinator Module)

중계부는 자원선택 및 작업 스케줄링을 담당하는 모듈이다. 자원선택 모듈(Resource Selector)은 데이터 추출 모듈로부터 사용자의 요구사항과 데이터 특성 정보를 전달 받아 자원정보 데이터베이스에 저장된 속성 값을 바탕으로 그리드 자원 온톨로지로 인스턴스들을 생성한다. 생성된 인스턴스들로 규칙이 포함된 추론엔진을 통하여 질의 및 응답을 받고, 사용자에게 선택 자원을 최종 승인할 것인지에 대한 과정을 반복한다. 중계부가 선택한 자원이 사용자에게 의해 확인 및 승인이 되면 사용자마다 과거 선택자원 데이터베이스에 일정 수 이상을 저장하거나 오래된 정보는 삭제한다.

작업처리 스케줄러(Job Scheduler)는 사용자의 데이터를 선택된 자원에 분배 및 할당하는 역할을 한다. 본 논문에서 사용한 스케줄링 기법은 선택한 자원에 일정한 순서로 작업을 분배하는 Round-robin(RR) 방식과 이전부터 현재까지의 신뢰성 값을 측정 및 저장해서 가장 높은 신뢰성을 갖는 자원에 할당하는 Reliability Measurement(RM)[13]를 사용하였다. 중계부는 스케줄링 기법에 따라 분배 및 할당된 작업의 결과들을 취합하여 결과 파일을 생성해서 사용자에게 응답하는 역할도 한다.

표 2. 그리드 자원선택을 위한 규칙
Table 2. Rules for Grid Resource Selection

Rule	Description
1	DesktopComputer(?x) ^ CPUSpeed(?x, ?y) ^ (?y >= 3000.0) ^ organization(?x, ?z) ^ (?z = usrOrg) → 동일기관고속컴퓨터(?x) 빠른 처리 속도와 동일한 기관에 속한 자원을 원한다.
2	Server(?x) ^ bandwidth(?x, ?y) ^ (?y >= 100.0) ^ diskStorageSize(?x, ?z) ^ (?z > 5000.0) → 대용량고속서버(?x) 고속 네트워크 환경에서 대용량의 데이터 처리가 가능한 자원을 원한다.
3	Server(?x) ^ CPUSpeed(?x, ?y) ^ (?y >= 3400.0) ^ cost(?x, ?z) ^ (?z > 15) ^ diskStorageSize(?x, ?k) ^ (?k < 10) → 작은데이터계산위주서버(?x) 자원이 비싸더라도 연산에 필요한 데이터 크기는 작고 처리 속도가 빠른 작은 자원을 원한다.
4	Notebook(?x) ^ location(?x, ?y) ^ (?y = usrLoc) ^ cost(?x, ?z) ^ (?z < 5) → 저가노트북(?x) 처리해야 할 속도와 크기는 상관없고, 사용자와 동일한 지역에 있는 가격이 저렴한 자원을 원한다.
5	Cluster(?x) ^ numCPU(?x, ?y) ^ (?y >= 5) ^ CPUSpeed(?x, ?z) ^ (?z > 3000.0) ^ OperatingSystemType(?x, ?k) ^ (?k = WinFamily) ^ diskStorageSize(?x, ?h) ^ (?h >= 1000.0) ^ bandwidth(?x, ?s) ^ (?s >= 100.0) → 고성능대용량고속클러스터(?x) 가용 CPU의 개수가 5개 이상이고, OS가 윈도우 계열이며 고속 네트워크 환경의 대용량의 처리가 가능한 자원을 원한다.
·	·
·	·
·	·

IV. 실험 및 결과 분석

우리는 온톨로지 기반의 그리드 자원선택 시스템이 효율적이며 지능적으로 자원을 선택하는 것을 평가하기 위해 DEVS[14] 형식론을 적용해서 모델링하고 시뮬레이션 하여 기존 모델인 Condor-G 와 Nimrod-G를 비교하였다. 자원 선택 모델 별로 제공하는 자원선택 방법으로 10개의 자원을 각기 선택한 후 앞서 언급한 RR, RM 두 가지 스케줄링 기법을 이용하여 선택한 자원에게 작업을 분배함으로써 작업 처리율, 작업 손실, 자원 활용률, 평균 작업처리 시간을 측정하였다.

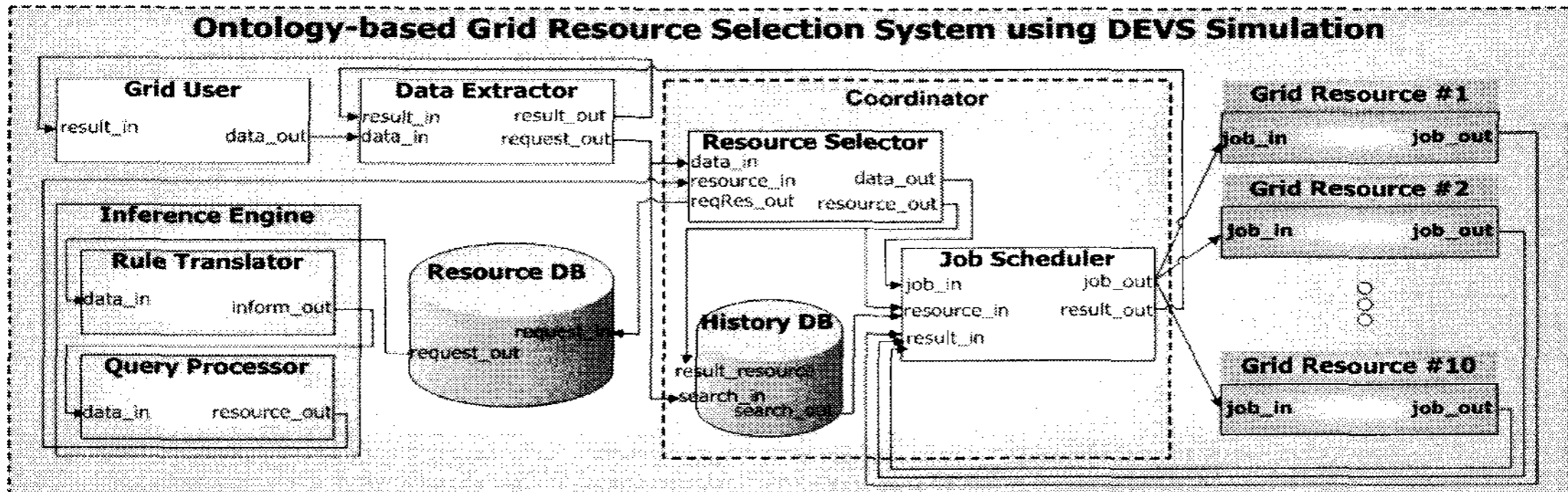


그림 3. DEVS로 시뮬레이션한 OGRSS 모델
Fig. 3. OGRSS Model using DEVS Simulation

4.1 온톨로지 기반의 그리드 자원선택 시스템 시뮬레이션

우리는 성능 평가를 위해 DEVS[14] 형식론을 적용한 시뮬레이션 모델을 그림 3과 같이 6개의 컴포넌트로 나타내었다.

그림 3에서 Grid User는 그리드 컴퓨팅을 이용하는 사용자로서 처리하고자 하는 데이터를 OGRSS로 보낸다. Data Extractor 모듈은 그리드 사용자의 자원사용 요청 메시지를 받아, 처리된 결과를 송신할 때까지 세션 연결 등 통신을 담당하며 사용자의 데이터가 어떤 특징을 갖는지 정수 형태로 변환하여 중계부로 전송한다. 또한 History DB에서 현재 요구사항과 데이터 특성이 비슷한 자원을 요구했는지 확인하고, 일치하거나 비슷한 자원을 발견하면 Job Scheduler에 해당 자원 목록을 전송한다. Coordinator 모듈은 사용자의 데이터를 전달받아 입력 큐에 저장하고 있다. 자원이 선택되면 그리드 자원에게 분배하는 역할을 한다. Resource Selector 모듈은 Data Extractor 모듈로부터 사용자 데이터의 특성과 요구사항 정보를 입력 받아, 자원을 새로 선택해야 할 경우 Resource DB로 자원을 요청하여 최종 선택 자원 목록을 전달 받는다. 이 때 최종 자원 선택 목록은 History DB에 저장하고, Job Scheduler로 자원 목록 정보를 전송한다. Job Scheduler는 전달 받은 선택 자원 목록으로 그리드 자원을 구성하고, RR과 RM 스케줄링 기법으로 작업을 배분한다. 또한 그리드 자원으로부터 처리가 완료된 자원을 취합하여 결과 파일로 만들고 Data Extractor 모듈로 전송하는 역할을 한다. Resource DB는 기존의 그리드 미들웨어와 연동되어 자원에 대한 정보 값들을 데이터베이스에 등록 및 갱신, 삭제하는 역할을 하며 Resource Selector로부터 자원 요청 메시지를 받으면 자원에 대한 정보들을 추론 엔진에 전송하는 역할을 한다. 추론 엔진은 보쌈 추론 엔진을 사용하였으며 Resource DB

에서 전달 받은 속성 값들로 보쌈 API를 이용하여 인스턴스들을 생성한다. 추론 엔진을 통하여 본 논문에서 정의한 규칙으로 사용자가 요구하는 자원에 응답하여 선택 한 자원 목록을 Resource Selector에 전송하는 역할을 한다.

본 실험에서 그리드 자원들은 무작위로 위치정보, 소속기관, CPU 속도, 메모리 크기, 디스크 크기, 소속기관, 네트워크 대역폭, 비용 등 표 1과 같은 형태로 생성하여 자원정보 데이터베이스에 저장하였다. OGRSS의 자원선택 기법은 본 논문에서 제안한 표 2와 같은 규칙을 미리 생성하여 사용자가 처리하고자 하는 데이터가 테라바이트(terabyte)급 이상의 실시간으로 처리를 요하는 고성능의 컴퓨터로써, 네트워크 속도가 빠르며 그 중, 비용이 가장 적게 드는 자원 목록을 선택하였다. Condor-G는 자원의 조건을 상세히 명세 할 수는 있지만, 하나의 자원만 검색하는 기능을 가지므로 일정 성능 이상이 되는 자원을 선택했다. Nimrod-G는 계산속도 중심의 CPU 성능이 일정 성능 이상인 자원을 선택하였다. 이렇게 각기 다른 선택 방법으로 10개의 자원을 선택하여 그리드 자원을 구성한 후, 중계부가 RR, RM 스케줄링 방식으로 작업을 분배하여 일정 DEVS 시간 동안 성능을 측정한다.

4.2 실험 결과

첫 번째 실험은 빠른 작업을 요구하는 작업에 대하여 단위 시간당 처리된 작업의 개수를 나타내는 작업 처리율을 측정하였다. 그림 4, 5는 RR, RM의 다른 스케줄링 방식으로 작업을 분배하였을 때 Condor-G, Nimrod-G, OGRSS로 선택한 자원들의 작업 처리율 변화를 나타낸다. Condor-G의 경우 RR, RM 순서대로 0.3581, 0.3777 이었고, Nimrod-G는 0.3541, 0.3839 이었으며 OGRSS는 0.3954, 0.4526를 나타냈다. 두 가지 스케줄링 기법 모두에서 OGRSS로 선택한 자원들로 작업을 분배하였을 때 더 좋은 처리율을 보였다. 이

는 다양한 형태로 존재하는 이기종의 그리드 컴퓨팅 환경에서 보다 효율적인 작업을 처리할 수 있도록 더 좋은 성능을 가진 자원을 적당하게 선택함을 알 수 있다.

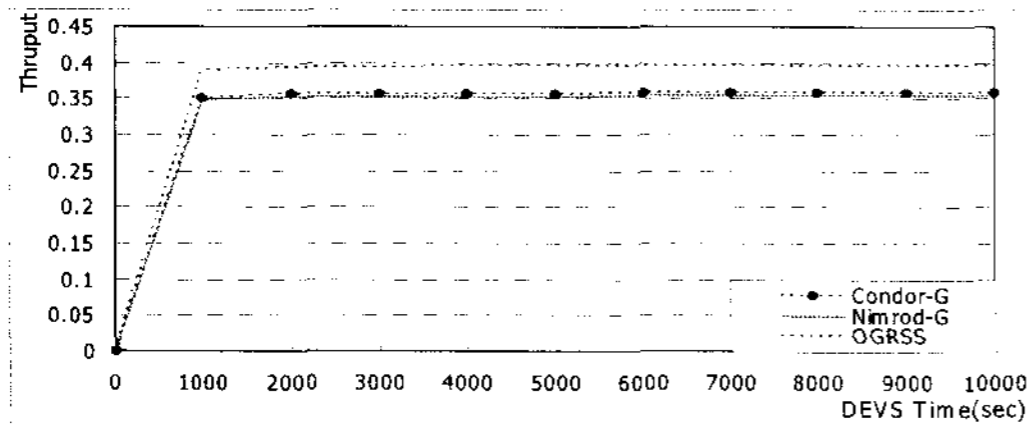


그림 4. RR 스케줄링의 작업 처리율
Fig. 4. Throughput of RR Scheduling

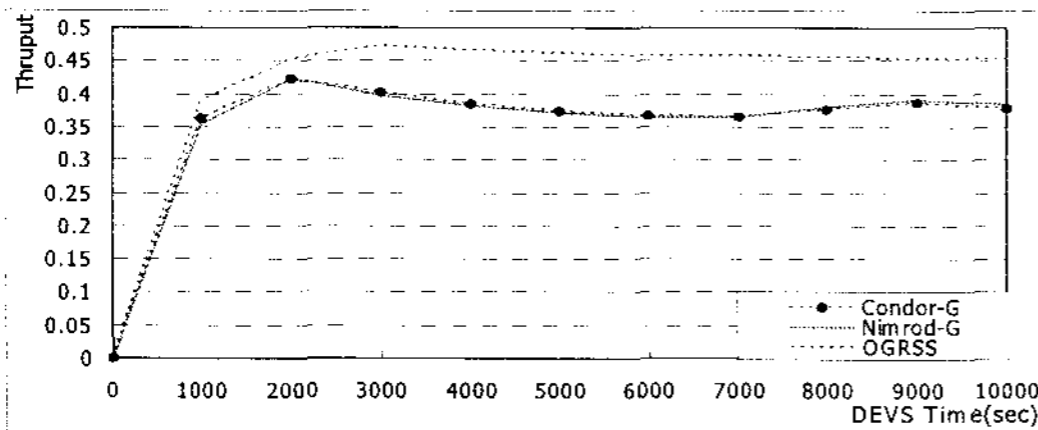


그림 5. RM 스케줄링의 작업 처리율
Fig. 5. Throughput of RM Scheduling

두 번째 실험은 시뮬레이션 시간 동안 발생한 작업 손실을 측정하였다. 그림 6, 7은 Condor-G, Nimrod-G, OGRSS로 선택한 자원들이 RR, RM 스케줄링 기법을 적용하였을 때 발생한 작업 손실 개수이다. 그림에서 알 수 있듯이 Condor-G의 경우 RR, RM 순서로 발생한 작업 손실의 개수는 1419, 1223을 기록했으며, Nimrod-G는 1459, 1161개를 기록했다. OGRSS로 선택한 자원들은 이에 비하여 현저하게 적은 1046, 474의 작업 손실을 기록하였으며, 이는 매우 다양한 형태로 존재하는 그리드 컴퓨팅 환경에서 OGRSS 기법으로 선택한 자원들이 보다 더 안정적으로 작업을 처리할 수 있도록 성능이 좋은 자원을 선택할 수 있음을 알 수 있다.

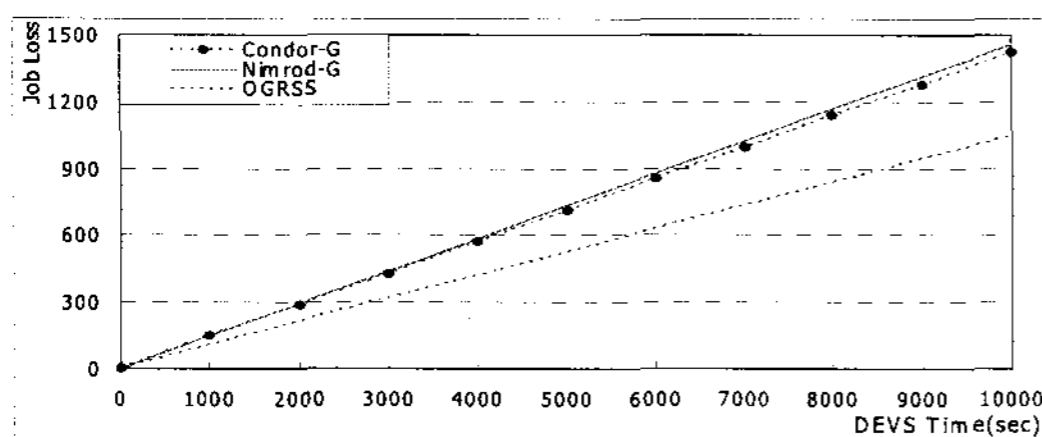


그림 6. RR 스케줄링의 작업 손실
Fig. 6. Job Loss of RR Scheduling

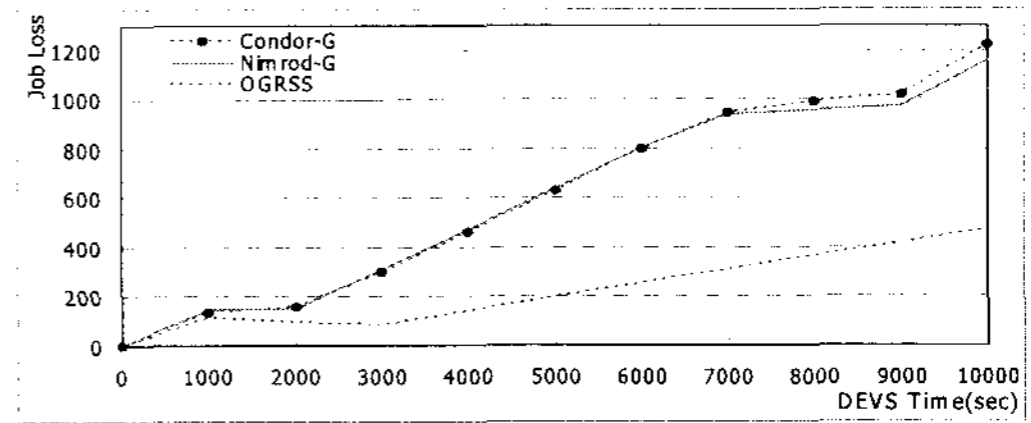


그림 7. RM 스케줄링의 작업 손실
Fig. 7. Job Loss of RM Scheduling

세 번째 실험은 자원의 활용률을 측정하였다. 자원의 활용률[13]은 시뮬레이션 시간동안 선택된 자원이 가동된 비율을 나타낸 지표로써 빠른 작업 처리가 가능한 자원을 선택하여 처리하고 있음을 나타낸다.

그림 8, 9는 RR, RM 스케줄링 기법에 따른 자원의 활용률을 나타낸 그림으로써, Condor-G는 71.64, 75.60이었고, Nimrod-G는 70.86, 76.81이었으며 OGRSS는 79.08, 90.53를 기록했다. 주목할 만 한 점은 OGRSS로 선택한 자원으로 RM 스케줄링 기법으로 작업을 분배 하였을 때 90% 이상의 높은 자원 활용률을 기록했다는 것이다. 이는 OGRSS로 선택한 자원과 적합한 스케줄링 기법을 적용하면 자원의 활용률을 극대화시킬 수 있음을 보여준다. 작업 처리율과 마찬가지로 OGRSS로 선택한 자원들에 작업을 분배 하였을 경우, 다른 두 가지 방법으로 선택한 자원들 보다 모두 높은 수치의 자원 활용률을 보임으로써 그리드 자원을 보다 더 효율적으로 사용함을 나타낸다.

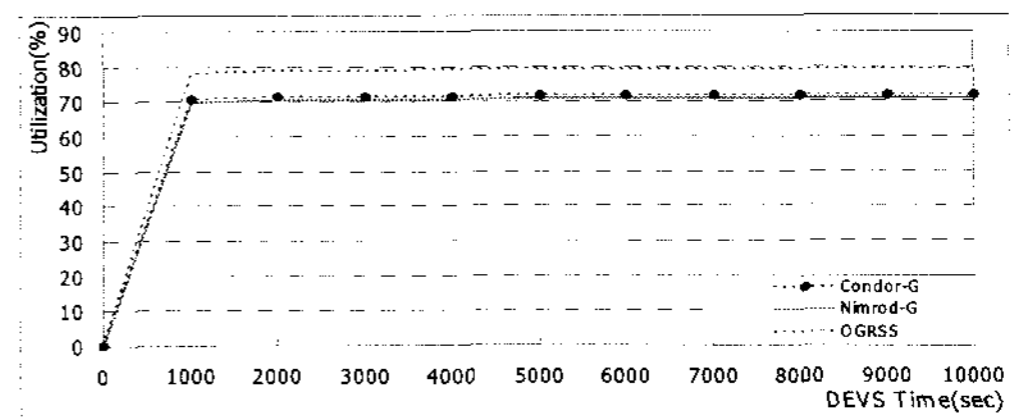


그림 8. RR 스케줄링의 자원 활용률
Fig. 8. Utilization of RR Scheduling

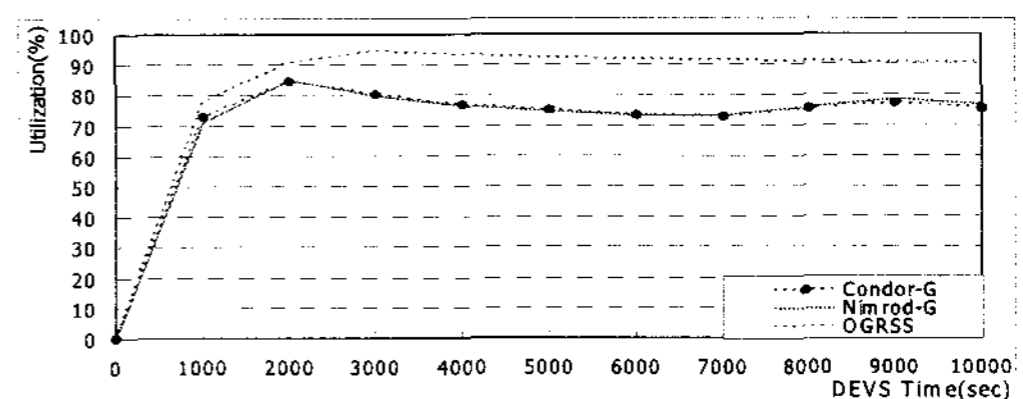


그림 9. RM 스케줄링의 자원 활용률
Fig. 9. Utilization of RM Scheduling

마지막 실험은 작업처리에 소요되는 평균 시간 값을 측정하였다. 평균 작업처리 시간은 요청한 작업이 처리 완료되어 반환되는 시간의 평균값이다. 그림 10, 11은 RR, RM 스케줄링 기법에 따른 평균 작업 소요시간으로써 Condor-G는 548.28, 1115.53이고, Nimrod-G는 567.78, 1119.92이며 OGRSS는 497.88, 440.05이었다. Condor-G와 Nimrod-G의 경우 RM의 평균 작업처리 시간이 RR에 비하여 2배가량 높게 나타났다. 이는 RR이 선택한 자원들의 저장 능력에 상관없이 순차적으로 분배하여 손실되는 작업이 많은 데 비하여, RM은 높은 신뢰성을 기록한 자원에게 할당함으로써 그림 7과 같이 적은 양의 작업 손실을 나타냈고, 이는 소요 시간이 작업 큐에 머무는 시간을 더하므로 전체적인 평균 작업 처리 시간도 증가한 것이다. 이에 비하여 OGRSS로 선택한 자원은 RM에서도 RR보다 더 낮은 평균 작업처리 시간을 기록했다. 이는 RR에서 손실된 작업을 제외하고 계산된 평균 작업처리 시간 보다 작업 손실 없이도 처리 완료된 평균 처리 시간이 더 빠르다는 것을 알 수 있다. 즉, 작업 손실은 더 적고 평균 작업처리 시간이 더 빠르다는 것을 알 수 있다.

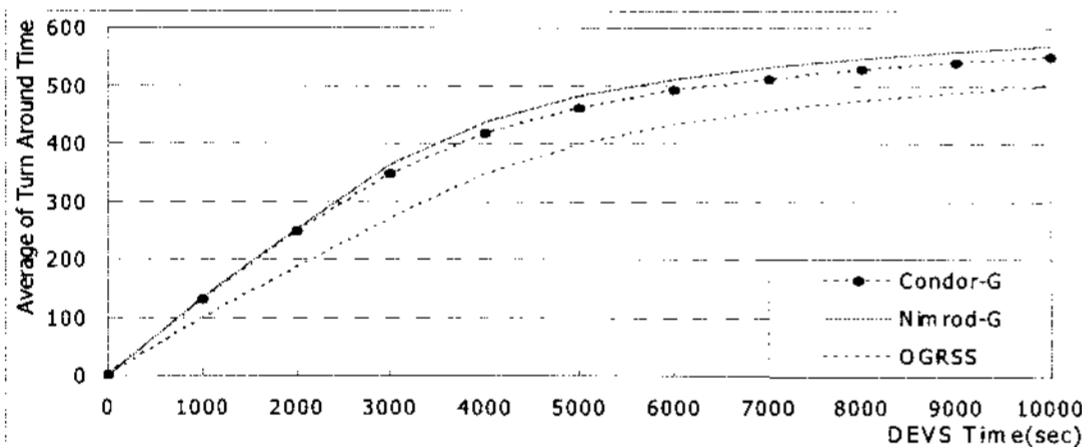


그림 10. RR 스케줄링의 평균 작업 처리시간
Fig. 10. Average of Turn Around Time of RR Scheduling

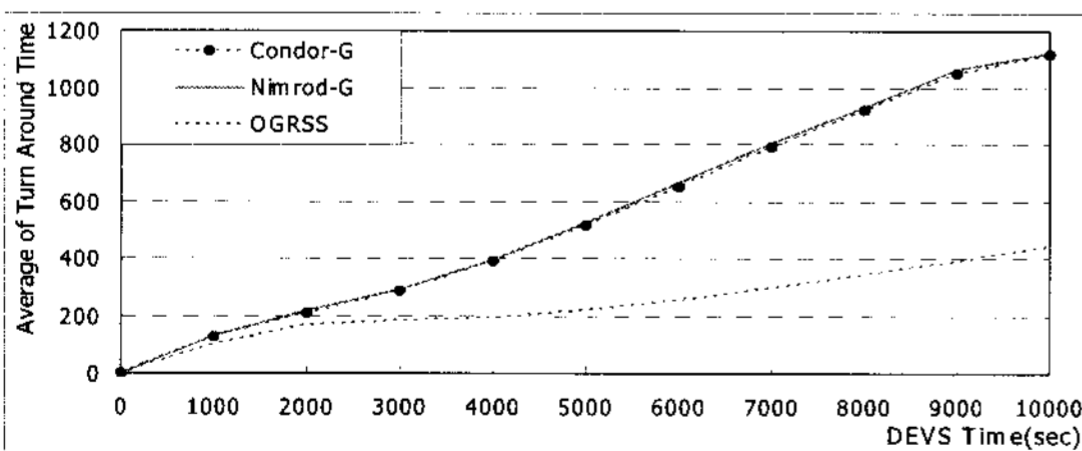


그림 11. RM 스케줄링의 평균 작업 처리시간
Fig. 11. Average of Turn Around Time of RM Scheduling

우리는 지금까지 Condor-G, Nimrod-G, OGRSS로 선택한 자원들로 RR, RM 스케줄링 기법을 적용하여 작업 처리율, 작업 손실, 자원 활용률, 평균 작업처리 시간을 측정하였다. 스케줄링 기법으로 보았을 때 RR 기법 보다 RM 스케

줄링 기법이 더 나은 성능을 기록하였고, 자원선택 기법으로는 Condor-G와 Nimrod-G가 비슷한 성능을 보였으며 본 논문에서 제안한 OGRSS가 두 개의 자원선택 방법 보다 더 나은 성능을 보였다. 이는 본 논문에서 제안한 시스템이 이기종의 다양한 자원들로 이루어진 그리드 컴퓨팅 환경에서 고성능의 그리드 자원을 요구할 때, 보다 더 적합한 자원들을 효율적으로 선택 할 수 있으며 더 나은 시스템 성능을 제공할 수 있음을 증명한다.

V. 결론 및 향후 과제

다양한 형태로 존재하는 자원 중에서 사용자의 요구사항과 데이터 특성에 따라 적합한 자원을 선택하기 위해서는 기존의 자원선택 방법인 Condor-G와 Nimrod-G로는 자원 정보관계형 데이터베이스에 저장되어있어서 구조상 그 한계가 있다. 본 논문은 그리드 컴퓨팅 환경에서 계산 그리드 자원들을 온톨로지로 표현하고, 기존의 관계형 데이터베이스에 저장된 자원 정보들을 바탕으로 인스턴스들을 생성한 후, 규칙을 정의하고 자원 정보에 의미를 부여하고 추론 엔진을 통하여 대용량의 분산 작업 처리를 위해 사용자들에게 적합한 자원을 선택하는 온톨로지 기반의 그리드 자원선택 시스템을 제안하였다.

실험 결과를 통하여 본 논문에서 제안한 온톨로지 기반의 그리드 자원선택 시스템이 기존의 자원선택 시스템보다 더 높은 작업처리율과 자원 활용률을 보이고, 낮은 작업 손실, 짧은 평균 작업처리 시간을 제공함으로써 사용자가 대용량의 고속 컴퓨팅 자원을 선택하는데 더 효과적이라는 것을 알 수 있다. 더불어 그리드 자원을 선택함에 있어서 다양한 사용자의 요구사항과, 데이터 특성, 자원 정보에 보다 더 세분화 된 규칙을 적용함과 동시에 적합한 스케줄링 기법을 적용하면 사용자의 만족도를 개선시킬 수 있으며 그리드 컴퓨팅 전반에 걸쳐 시스템 성능 상승효과를 기대 할 수 있다. 향후 연구에서는 네트워크의 혼잡도와 CPU 가용률 측정을 통하여 그리드 자원정보 온톨로지를 보다 더 세분화하고, 사용자의 요구사항이 잘 반영된 규칙을 정의하여 자원선택의 적합성 및 정확성을 개선시킬 수 있는 그리드 자원선택 시스템을 개발할 예정이다.

참고문헌

[1] 정준원, 정호영, 김종남, 임동혁, 김형주. "RDF 기반의 온톨로지 처리시스템", 정보과학회논문지, 제11권 제4

호, pp.381-392, 2005. 8

[2] 박다혜, 이종식. "계산 그리드 컴퓨팅에서의 자원 성능 측정을 통한 그리드 스케줄링 모델", 한국컴퓨터정보학회논문지, 제11권, 제5호, pp.87-94, 2006. 11.

[3] Rajkumar Buyya, David Abramson, and Spikumar Venugopal. "The Grid Economy", Proceedings of the IEEE, VOL.93, NO.3, 2005.

[4] 이준돈, 정윤미, 길아라, 윤현주. "그리드 컴퓨팅 환경에서 효율적인 자원 활용을 위한 성능 계량 모델 및 자원 선택 알고리즘 제안", 한국정보과학회 가을 학술발표논문집, Vol.30, No.2, pp.466-468, 2003. 10.

[5] A.M. Pernas, M.A.R Dantas. "Using Ontology for Description of Grid Resources", Proceedings of the 19th International Symposium on HPCS'05, 2005.

[6] 조수현, 김영학. "계산 그리드 상에서 프로그램의 특성을 반영한 작업 프로세스 수의 결정에 관한 연구", 한국컴퓨터정보학회논문지, 제11권, 제1호, pp.71-85, 2006. 3.

[7] 노남수, 홍필두, 장택수, 이용우. "Grid환경기반 자원 선택에 관한 연구", 한국컴퓨터종합학술대회 논문집, Vol.32, No.1(A), pp.49-51, 2005. 7.

[8] 한선우. "온톨로지 기반의 조선설계 지식 통합 및 매핑", 석사학위논문, 인하대학교, 2006.

[9] Protégé available at "<http://protege.stanford.edu/>"

[10] 윤보현, 서창호. "시맨틱웹을 위한 효율적인 온톨로지 객체 모델", 한국컴퓨터정보학회논문지, 제11권, 제2호, pp.7-13, 2006. 5.

[11] Bossam Inference Engine available at "<http://bossam.wordpress.com/>"

[12] 최정화, 박영택. "온톨로지 추론 기술 동향", 정보과학회지, 제24권 제12호, pp.47-55, 2006. 12.

[13] 박다혜, 이종식. "자원 신뢰성 측정을 통한 효율적인 그리드 자원 스케줄링 모델", 한국시뮬레이션학회 논문지, Vol.15, No.2. pp.129-136, 2006. 12.

[14] B.P. Zeigler, et al. "DEVS Framework for Modeling, Simulation, Analysis and Design of Hybrid systems in Hybrid II", Lecture Notes in CS, Vol.3045 Springer-Verlag, Berlin, 1997.

저자소개



노창현
 2006 수원대학교 인터넷 정보공학과 학사
 2006.8~현재 인하대학교 정보공학과 컴퓨터정보공학 석사과정
 관심분야 : 그리드 컴퓨팅, 온톨로지, 시스템 모델링 및 시뮬레이션, 임베디드 시스템



장성호
 2004 용인대학교 컴퓨터 정보공학과 학사
 2006 인하대학교 컴퓨터 정보공학과 석사
 2006.3~현재 인하대학교 정보공학과 컴퓨터정보공학 박사과정
 관심분야 : 그리드 컴퓨팅, RFID, 웹서비스, 소프트웨어 모델링



김태영
 2007 인하대학교 컴퓨터공학부 학사
 2007.3~현재 인하대학교 정보공학과 컴퓨터정보공학 석사과정
 관심분야 : 분산처리, 시스템 모델링 및 시뮬레이션



이종식
 1993 인하대학교 전자공학과 학사
 1995 인하대학교 전자공학과 석사
 2001 미국 애리조나대 전기·컴퓨터 공학과 박사
 2001~2002 캘리포니아 주립대학교 전기·컴퓨터공학과 전임강사
 2002~2003 클리블랜드 주립대학교 전기·컴퓨터공학과 조교수
 2003~2006 인하대학교 컴퓨터공학부 조교수
 2006~현재 인하대학교 컴퓨터공학부 부교수
 관심분야: 시스템 모델링 및 시뮬레이션, 그리드 컴퓨팅