

## 웹기반 그리드 스케줄링 플랫폼의 구현

강상성 강오한<sup>0</sup>

안동대학교 컴퓨터교육과

{edukang, ohkang<sup>0</sup>}@andong.ac.kr

### Implementation of Web-based Grid Scheduling Platform

SangSung Kang Ohhan Kang<sup>0</sup>

Dept. of Computer Education, Andong National University

#### 요 약

본 논문에서는 웹기반의 그리드 컴퓨팅 환경에서 시스템을 모델링하고 스케줄링 기법을 시뮬레이션 할 수 있는 그리드 스케줄링 플랫폼을 설계하고 구현하였다. 구현한 웹기반 그리드 스케줄링 플랫폼은 자바 환경의 그리드 스케줄링 툴킷인 GridSim을 시뮬레이션 도구로 활용하였으며, 자원 모델링, 작업 모델링, 알고리즘 컴파일, 시뮬레이션을 웹 환경에서 빠르게 수행할 수 있다. 구축한 스케줄링 플랫폼은 향후 그리드 연구에 기반구조로 활용될 수 있으며, 스케줄링 기법의 성능 분석을 위한 도구로 사용될 수 있다.

#### 1. 서 론

지역적으로 분산되어 있는 고성능의 시스템들을 하나로 묶어 사용하는 그리드 컴퓨팅이 차세대 병렬·분산 연산을 위한 새로운 패러다임으로 관심을 끌고 있다[1].

그리드 시스템을 위한 작업 스케줄링 알고리즘의 세부 동작과 성능을 분석하기 위해서는 실제로 구축된 그리드 시스템에 적용하여 로그 및 수행시간을 측정해야 한다. 그러나 그리드를 구성하는 자원들은 비교적 원거리에 위치해 있고, 서로 다른 여러 가지 특성을 가지며, 시간과 장소에 따라 자원을 통제할 수 없을 수도 있다. 또한 일반적인 그리드 환경을 위한 스케줄링 알고리즘의 성능을 검증하기 위해서는 다양한 그리드 환경에서의 결과를 필요로 하기 때문에 실제 그리드 환경을 이용한 알고리즘 분석은 현 단계에서는 많은 어려움이 따른다.

작업 스케줄링 알고리즘의 동작상태 및 성능을 분석하기 위해 그리드의 자원과 작업집합을 모델링하여 실제 시스템이 동작하는 것처럼 소프트웨어로 구현하여 결과를 시뮬레이션 하는 방법이 대안으로 많이 사용되고 있다. 시뮬레이션은 그리드 환경이 가지는 다양한 특성들을 쉽게 반영할 수 있고, 결과의 확인 및 분석이 용이한 장점이 있지만 실제 그리드 환경에서 영향을 미치는 변수를 빠뜨리게 되면 왜곡된 결과를 산출할 수도 있다. 일반적으로 그리드 스케줄링 시뮬레이션에서는 그리드 환경, 작업, 사용자 환경 등이 고려되어야 한다.

#### 2. 관련 연구

본 논문과 관련된 그리드 툴킷에 관한 연구는 미국에서 SDSC(San Diego Supercomputer Center)와 Virginia 대학의 GCG(Grid Computing Group)이 대표적이며, 호주에서는 Melbourne 대학의 GRIDS 연구실에서 연구를

활발히 추진하고 있다. SDSC에서는 CSAG(Concurrent Systems Architecture Group)과 같은 UC San Diego 대학의 연구 실험실과 공동으로 MicroGrid, TeraGrid 등의 툴킷을 개발하였다. 호주의 GRIDS 연구실에서는 GridSim을 개발하여 전 세계의 여러 기관에서 이를 연구에 활용하도록 지원하고 있으며 계속적으로 기능을 확장하고 있다. 국내에서는 KISTI를 중심으로 그리드에 관한 연구가 수행중이다.

기존의 그리드 스케줄링 툴킷을 활용하여 스케줄링 알고리즘을 시뮬레이션 하기 위해서 적절한 개발환경의 구축, 소스코드의 분석, 자원 모델링을 위한 프로그래밍, 작업집합 모델링을 위한 프로그래밍, 스케줄링 알고리즘 구현 등 여러 가지 작업을 필요로 한다. 이러한 작업들은 연구자들에게 중복작업을 유발하여 스케줄링 알고리즘 연구의 효율성을 저하시킨다.

본 논문에서 개발한 웹기반 그리드 스케줄링 플랫폼(이하:WGridSP)은 자원 모델링, 작업 모델링, 알고리즘 컴파일, 시뮬레이션을 웹 환경에서 빠르게 시행할 수 있다. 따라서 연구자들에게 알고리즘의 구현과 결과 분석에만 집중할 수 있는 환경을 제공한다. 자바 기반의 대표적 스케줄링 툴킷인 GridSim[2]과 본 논문에서 개발한 WGridSP를 비교하면 [표 1]과 같다.

#### 3. 웹기반 그리드 스케줄링 플랫폼(WGridSP)의 구조

WGridSP는 자바 환경의 그리드 스케줄링 툴킷인 GridSim을 시뮬레이션 도구로 활용하였다. GridSim은 글로벌 그리드 환경에서의 다양한 변수들을 대부분 채택하고 있으며 분산환경의 시뮬레이션에 많이 사용되고 있는 simjava[3] 패키지를 바탕으로 함으로써 실용도와 안정성이 높은 그리드 스케줄링 툴킷이다. WGridSP는 사용자와 GridSim 사이에서 웹을 매개로 한 인터페이스 역할

을 함과 동시에 자원 모델링, 어플리케이션 모델링 자료와 스케줄링 알고리즘을 데이터베이스에 관리하여 재사용이 가능토록 하고 있다. WGridSP는 [그림 1]과 같이 4개의 기능단위로 구성되어 있다.

표 1. GridSim과 WGridSP의 비교

	GridSim	WGridSP
컴파일/실행 환경 구축	필수	필요 없음
툴킷 구조 분석	필수	필요 없음
자원 모델링	도구	웹브라우저
어플리케이션 모델링	형태	자바 소스
	컴파일	웹에서 컴파일
스케줄링 알고리즘 구현	도구	텍스트 에디터
	형태	자바 소스
시뮬레이션	컴파일	웹에서 컴파일
	실행	웹에서 실행
	결과	콘솔 및 로그파일
		웹페이지

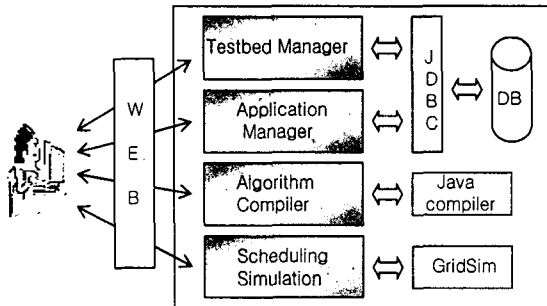


그림 1. WGridSP의 구조

### 3.1 테스트베드 관리자(Testbed Manager)

테스트베드 관리자는 사용자에게 자원 모델링을 쉽고 빠르게 할 수 있도록 지원한다. 특히, 사용자가 모델링한 자원 정보는 테스트베드별로 관리되어 데이터베이스에 저장된다. 테스트베드는 하나 이상의 자원으로, 각 자원은 하나 이상의 머신으로, 각 머신은 하나 이상의 프로세서를 가진다. 일반적으로 두 개 이상의 머신으로 구성된 자원은 클러스터 시스템으로 볼 수 있다. WGridSP의 자원에 지정되는 속성들은 다음과 같다.

- 프로세서 할당 전략(policy): 자원이 소유하고 있는 프로세서의 수보다 처리를 기다리는 작업의 수가 많은 경우 프로세서를 어떻게 할당할 것인가를 결정하는 속성이다. WGridSP에서는 특정 프로세서에 두 개 이상의 작업을 할당하여 시분할 방식으로 처리하는 TimeShared 방식과 먼저 할당받은 작업을 종료한 후 대기 중인 작업을 처리하는 SpaceShared 방식 중 하나를 선택할 수 있다.
- 전송 속도(baudrate): 해당 자원이 네트워크에 연결된 전송속도를 말한다. 작업의 처리에 필요한 입력 데이터의 크기는 처리 후 반환하는 출력 데이터의 크기와 함께 전송시간에 영향을 준다.
- 비용(cost): 해당 자원의 사용비용을 정의해 두면 비용

을 고려한 스케줄링 알고리즘에 활용할 수 있다.

- 프로세서 처리 능력: 작업을 처리하는 능력을 나타내는 것으로 MIPS 단위로 설정한다. 작업의 길이와 함께 처리시간에 영향을 준다.
- 프로세서 수(CPU): 머신이 가지는 프로세서의 수를 나타낸다. 프로세서의 수가 많을수록 동시에 처리할 수 있는 작업의 수가 많아진다.

### 3.2 어플리케이션 관리자

어플리케이션 관리자는 각 어플리케이션을 구성하고 있는 작업들의 모델링을 지원한다. 모델링한 작업들은 모두 그리스 시스템의 처리를 기다리는 것으로 GridSim Toolkit과 WGridSP에서는 Gridlet이라고 한다. 각 어플리케이션은 하나 이상의 Gridlet으로 구성되어 있으며 Gridlet은 다음과 같은 속성을 가지게 된다.

- 길이(length): Gridlet의 CPU 요구시간으로 이 값이 클수록 프로세서에서의 처리시간이 길어진다.
- 입력데이터 크기(input size): Gridlet을 처리하기 위해 필요한 입력데이터의 크기를 말한다. 스케줄러에 의해 할당된 자원에 전송해야 하므로 Gridlet을 자원까지 전달하는데 소요되는 시간에 영향을 준다.
- 출력데이터 크기(output size): 자원이 Gridlet 처리를 완료하고 산출하는 데이터의 크기로서 자원으로 부터 스케줄러까지 결과를 전송하는데 소요되는 시간에 영향을 준다.

### 3.3 스케줄링 알고리즘 관리

WGridSP는 스케줄링 알고리즘의 핵심 부문만을 필요로 한다. 알고리즘에 필요한 그리드 정보를 수집하는 소스코드는 컴파일 직전 자동 생성되며 컴파일 결과를 사용자에게 전달하여 오류를 수정할 수 있게 한다.

### 3.4 스케줄링 시뮬레이션

스케줄링 시뮬레이션을 위하여 필요한 테스트베드, 어플리케이션, 알고리즘을 선택하면 시뮬레이션 결과를 웹브라우저로 확인할 수 있다. Gridlet이 어떤 자원에 어느 시점에 배정되는지와 종료시점을 확인함으로써 알고리즘의 세부동작 상황을 분석할 수 있다.

### 4. 시뮬레이션 결과

본 논문에서 개발한 WGridSP를 사용하여 그리드 스케줄링을 시뮬레이션하는 예를 보면 다음과 같다.

#### 4.1 테스트베드 모델링

시뮬레이션을 위한 새로운 테스트베드를 생성하고 테스트베드에 속한 자원들을 작성한다. 이 예시에서는 두 개의 자원을 작성하였고 세부 사항은 [그림 2]와 같다.

#### 4.2 어플리케이션 모델링

처리해야할 작업 목록인 Gridlet의 각 변수를 사용자가 직접 지정할 수 있고 범위를 지정하고 난수를 이용하

여 생성할 수도 있다. [그림 3]은 10개의 Gridlet의 변수를 난수를 이용하여 작성한 것이다.

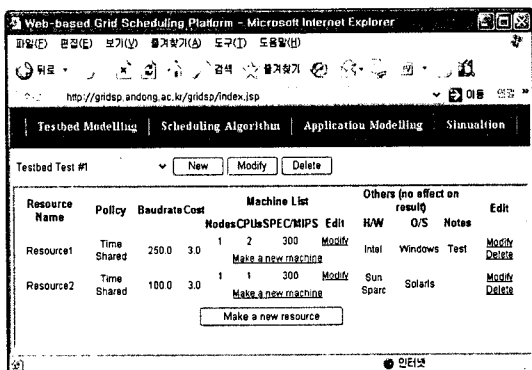


그림 2. 테스트베드 모델링의 예

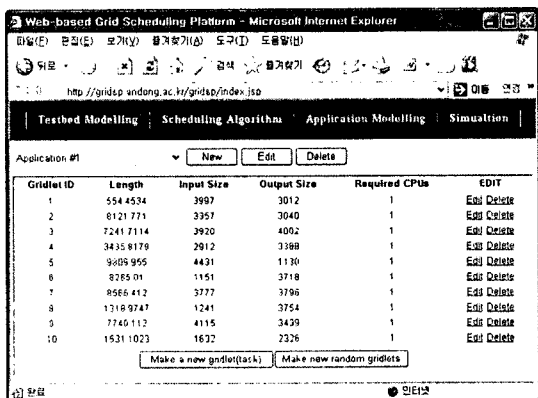


그림 3. 어플리케이션 모델링의 예

#### 4.3 알고리즘 컴파일

[그림 4]의 스케줄링 알고리즘은 현재 Gridlet을 무작위로 선정된 자원에게 배정하는 간단한 소스코드로 구성되어 있다.

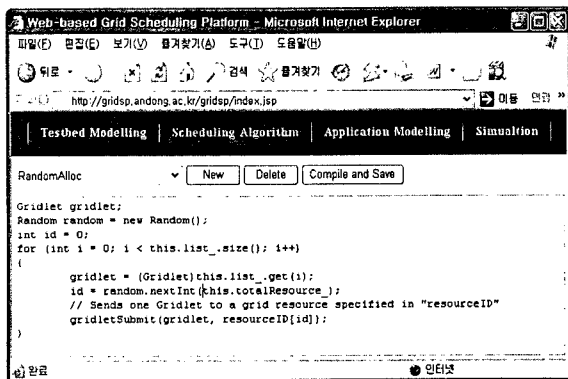


그림 4. 스케줄링 알고리즘의 컴파일

#### 4.4 스케줄링 시뮬레이션

위에서 작성한 테스트베드, 어플리케이션, 알고리즘을 선택하고 시뮬레이션 실행하면 [그림 5]와 같이 각 Gridlet의 배정 결과와 실행 결과가 나타난다. 예시에서 모든 작업이 종료되는데 소요된 시간은 210.652로 나타났다.

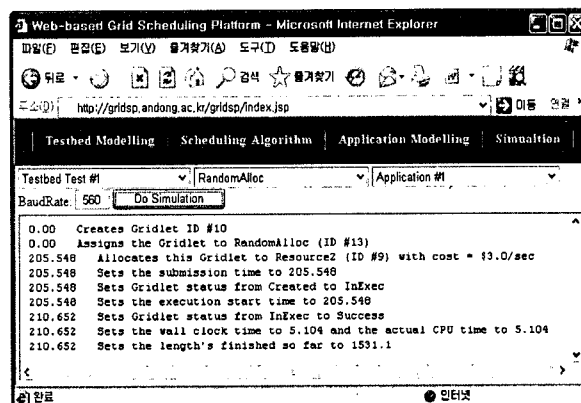


그림 5. 시뮬레이션의 결과

#### 5. 결론

본 논문에서는 자바 환경의 그리드 스케줄링 툴킷인 GridSim을 시뮬레이션 도구로 활용하여 자원 모델링, 작업 모델링, 알고리즘 컴파일, 시뮬레이션을 웹 환경에서 빠르게 수행할 수 있는 웹환경 그리드 스케줄링 플랫폼(WGridSP)을 구현하였다. WGridSP는 웹 환경에서 그리드 스케줄링을 시뮬레이션 할 수 있기에 많은 시간과 노력을 절약할 수 있으나 단일 사용자를 지원한다는 제한이 있다. 향후에는 simjava를 수정하여 다수의 사용자가 동시에 시뮬레이션할 수 있도록 기능을 보완할 예정이다.

#### 참고 문헌

- [1] Foster and C. Kesselman, "The Grid: Blueprint for a Future Computing Infrastructure," Morgan Kaufmann Publishers, USA, 1999.
- [2] R. Buyya, and M. Murshed, "GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing," The Journal of Concurrency and Computation: Practice and Experience(CCPE), Vol.14, pp. 1175-1220, 2002.
- [3] F. Howell and R. McNab, "SimJava: A Discrete Event Simulation Package for Java with Applications in Computer Systems Modelling," Proc. of 1st International Conference on Web-based Modelling and Simulation, Society for Computer Simulation, 1998.