

그리드 컴퓨팅을 위한 웹기반 동적스케줄링 시뮬레이션 플랫폼 1)

강상성⁰ 강오한
안동대학교 컴퓨터교육과
{edukang⁰, ohkang}@andong.ac.kr

Web-based Dynamic-scheduling Simulation Platform for Grid Computing

Sangseong Kang⁰, Ohhan Kang
Dept. of Computer Education, Andong National University

요 약

본 논문에서는 그리드 컴퓨팅 환경을 위한 동적 스케줄링 알고리즘을 웹상에서 시뮬레이션 할 수 있는 그리드 스케줄링 시뮬레이션 플랫폼을 설계하고 구현하였다. 구현한 웹기반 동적스케줄링 시뮬레이션 플랫폼은 자바 환경의 그리드 스케줄링 툴킷인 GridSim과 GridBroker를 수정하여 시뮬레이션 도구로 활용하였으며, 자원 모델링, 작업 모델링, 알고리즘 컴파일을 웹상에서 수행할 수 있다. 특히, 실시간 자원정보를 활용하는 동적 스케줄링 알고리즘을 지원함으로써 실제 그리드의 특성을 최대한 반영하였다. 구축한 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 환경에서는 상이한 성능을 갖는 다양한 자원들이 지역적으로 분산되어 인터넷으로 연결되어 있으며, 대용량의 연산능력이 요구되는 응용프로그램들이 처리된다. 그리드 환경에서 시스템의 성능을 극대화하기 위해서는 사용자의 요구와 사용 가능한 자원의 성능, 그리고 동적 정보인 현재 부하를 고려하여 응용프로그램을 효과적으로 처리하도록 하는 스케줄러가 필요하다.

그리드 시스템을 위한 작업 스케줄링 알고리즘의 세부 동작과 성능을 분석하기 위해서는 실제로 구축된 그리드 시스템에 적용하여 로그를 분석하고 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 현재 단계에서는 많은 어려움이 따른다.

이러한 문제를 해결하기 위해서 그리드 시스템의 자원, 사용자, 작업 등을 모델링하고 스케줄링 기법의 시뮬레이션이 가능한 그리드 플랫폼이 연구되고 개발되어야 한다. 특히, 그리드 환경은 여러 사용자가 임의의 시간에 처리를 요구할 수 있는 환경이므로 요구 시점의 자원 특성을 반영할 수 있는 동적 스케줄링 기법을 지원할 수 있어야 한다.

2. 관련 연구

현재까지 그리드 컴퓨팅 환경에서 스케줄링 알고리즘을 시뮬레이션할 수 있는 다양한 툴킷들이 개발되었다. 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. 본 논문에서 활용한 GridSim[2]은 호주의 GRIDS 연구실에서 개발하여 전 세계의 여러 기관에서 이를 연구에 활용하도록 지원하고 있으며 계속적으로 기능을 확장하고 있다.

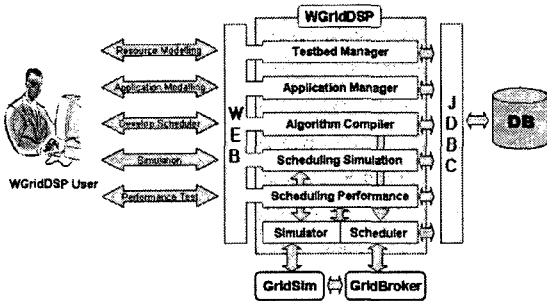
GridSim을 비롯한 기존의 그리드 스케줄링 툴킷은 그리드 환경의 가상적인 구현을 쉽게 할 수 있으나 자원 모델링, 사용자 모델링, 작업 모델링, 스케줄링 구현 및 분석에 필요한 각종 변수의 조작을 위해 프로그램 소스코드를 직접 작성해야 하고 생성된 결과를 다시 가공하여 차트 등으로 변환해야 한다. GRIDS 연구실에서는 GUI 기반의 모델링 도구인 Visual Modeler(VM)[3]를 개발하여 그들이 개발한 GridSim[2]을 위한 자원모델링과 어플리케이션 모델링을 위한 소스코드를 쉽게 작성할 수 있도록 하였다. 그러나 이 툴(tool)은 모델링 과정만을 다소 용이하게 할 뿐 자원 및 어플리케이션 정보의 저장과 관리, 시뮬레이션, 성능분석 등의 기능은 지원하지 않고 있다.

본 논문에서 개발한 웹기반 동적 스케줄링 플랫폼(이하: WGridDSP)은 자원 모델링, 작업 모델링, 알고리즘 컴파일, 시뮬레이션, 성능분석을 웹 환경에서 빠르게 시행할 수 있도록 하였다. 또한 GridBroker를 활용하여 여러 사용자가 임의의 시간에 작업처리를 요청하는 상황에서 자원의 동적 특성을 고려한 스케줄링이 가능하도록 하였다. 이는 각 사용자와 자원이 여러 지역과 조직에 속해있는 그리드 환경을 목표로 하는 스케줄링에 반드시 요구되는 사항이다.

1) 본 연구는 정보통신부의 2005년 IT학술기초연구(No. B1220-0501-0048) 지원사업에 의하여 수행되었음.

3. 웹기반 동적스케줄링 시뮬레이션 플랫폼(WGridDSP)의 구조

WGridDSP는 자바 환경의 그리드 스케줄링 툴킷인 GridSim을 일부 수정하여 그리드 환경을 위한 시뮬레이션 도구로 활용하였다. GridSim은 글로벌 그리드 환경에서의 다양한 변수들을 대부분 채택하고 있으며 분산환경의 시뮬레이션에 많이 활용되고 있는 SimJava 패키지를 사용함으로써 실용도와 안정성이 높은 그리드 스케줄링 툴킷이다. WGridDSP는 사용자와 GridSim 사이에서 웹을 매개로 한 인터페이스 역할을 함과 동시에 자원, 사용자, 작업 등의 모델링 자료와 스케줄링 알고리즘을 데이터베이스에 관리하여 재사용이 가능하도록 한다. WGridDSP의 구조는 [그림 1]과 같이 사용자, GridSim, DB 등과 연계되어 있다.

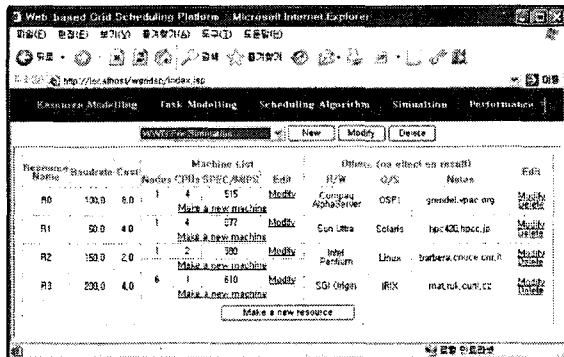


[그림 1] WGridDSP의 구조

4. WGridDSP의 기능 및 적용 예시

4.1 자원 모델링

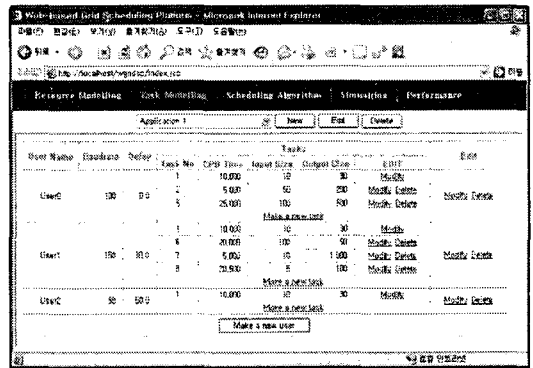
자원모델링은 실제계의 그리드 시스템을 가정하여 스케줄링 시뮬레이션을 하기 위한 환경을 정의하는 것이다. 자원모델링은 개별 테스트베드 단위로 저장/수정/삭제가 가능하다. [그림 2]는 R0~R3 네 개의 자원으로 이루어진 테스트베드를 정의한 것이다. 자원 R0는 통신속도 100, 초당 비용 8, 515MIPS의 성능을 지닌 프로세서 4개를 장착한 시스템이다. R3는 노드의 수가 6개인 클러스터 시스템을 가정한 것이다.



[그림 2] 자원 모델링의 예

4.2 어플리케이션 모델링

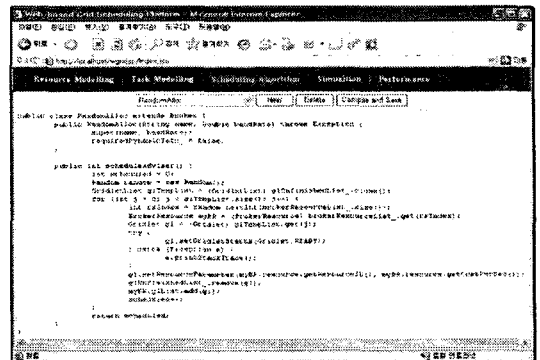
어플리케이션 모델링은 시뮬레이션 기능만을 위한 것이다. 명확한 특성을 가진 소수의 사용자와 그에 속한 작업을 정의하게 된다. [그림 3]은 User0~User2 3명의 사용자에게 각각 3개, 4개, 1개씩의 작업이 속해있는 어플리케이션을 나타내고 있다. User0은 통신속도 100에 연결되어 있으며 지연시간이 0이므로 시뮬레이션 시작 직후 작업을 수행할 수 있다. User1과 User2는 각각 통신속도 150, 50에 연결되어 있으며, 시뮬레이션 시작 후 30과 50 이후부터 스케줄링 알고리즘에 따라 그들이 소유한 작업에 자원이 배정된다.



[그림 3] 어플리케이션 모델링의 예

4.3 알고리즘 컴파일

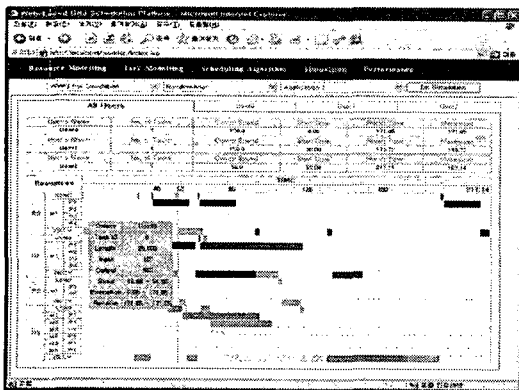
알고리즘 컴파일러는 개발자가 작성한 자바 소스 코드를 컴파일하는 역할을 수행한다. 모든 스케줄링 알고리즘은 같이 Broker 클래스를 상속받게 된다. Broker 클래스에는 scheduleAdviser() 메소드가 abstract로 지정되어 있으므로 scheduleAdviser() 메소드는 반드시 정의해야 하며, 이 메소드에 핵심 알고리즘을 작성할 수 있다. [그림 4]는 gUnfinishedList에 있는 작업들을 무작위로 선택한 자원에 배정하는 간단한 알고리즘이다. 무작위로 배정하므로 자원의 동적 정보는 필요로 하지 않는다. 따라서 생성자 함수에서 requiredDynaInfo 값을 false로 지정하였다.



[그림 4] 알고리즘 작성의 예

4.4 시뮬레이션

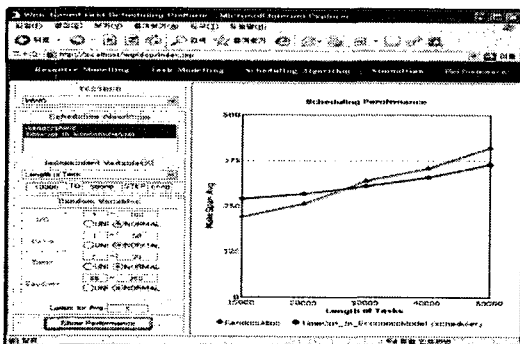
[그림 5]는 'WWG For Simulation'이라는 테스트베드에 'Application 1'이라는 어플리케이션을 무작위로 배정하는 알고리즘을 적용하여 시뮬레이션한 결과이다. 시뮬레이션 결과는 모든 사용자의 작업을 한 번에 볼 수도 있으며 특정 사용자에 속한 작업만을 볼 수도 있다. 그림에서 회색(흐린색) 차트로 표시된 부분은 사용자가 그리드의 자원정보를 요청하고 받는 경우를 나타내고 컬러(진한색)로 표시된 부분이 특정 자원을 자원에 전송하여 실행하고 결과를 반환받는 과정을 나타낸다. 각 차트에 마우스 포인터를 올리면 그림과 같이 작업의 자세한 수행시점을 확인할 수 있다.



[그림 5] 시뮬레이션의 예

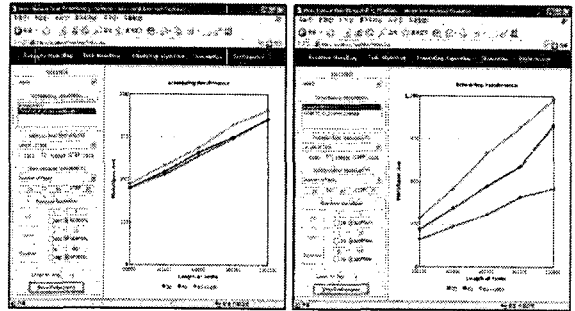
2.5 성능분석

[그림 6]은 두 개의 스케줄링 알고리즘의 성능을 비교한 것이다. 'RandomAlloc'은 작업을 무작위로 자원에 배정하는 방식이며, 'TimeOpt_In_EconomicModel' 알고리즘은 EconomicModel[4]의 시간최적화 알고리즘만을 WGridDSP 환경에 맞게 수정한 것이다. 두 알고리즘을 비교하면 작업의 길이가 약 25,000 이하에서는 무작위 배정 방식의 알고리즘이 우수하지만 그 이상에서는 반대 현상을 나타낸다. 이는 작업의 길이가 짧은 경우 자원의 동적 정보를 필요로 하는 'TimeOpt_In_EconomicModel'에 오버헤드의 영향이 크기 때문이다.



[그림 6] 두 알고리즘의 성능의 비교분석

[그림 7]은 각 사용자가 가지는 작업의 수를 20, 40, 60 세 부분으로 나누어서 작업의 길이 변화에 따른 완료시간을 나타낸 것이다. 'TimeOpt_In_EconomicModel'이 'RandomAlloc' 보다 작업의 수에 따른 영향을 작게 받고 전체적인 수행시간도 짧은 것으로 나타났다.



[그림 7] 작업의 수에 따른 성능 분석

5. 결론

WGridDSP는 웹 환경에서 동적인 그리드 스케줄링을 시뮬레이션 할 수 있어서 많은 시간과 노력을 절약할 수 있다. 그러나 기존의 알고리즘을 지원할 수 있도록 기능들을 구현함으로써 새로운 알고리즘의 개발과정에 예상치 못한 기능상의 제약점이 따를 수 있다. 향후 WGridDSP를 활용하여 다양한 알고리즘을 개발하면서 여러 가지 알고리즘 기법들을 수용하도록 기능을 추가할 예정이다.

그리드 시스템에서는 다양한 분야의 응용프로그램들이 처리될 수 있으므로 그리드 시스템의 성능을 극대화하기 위해서 본 논문에서 연구한 그리드 플랫폼을 사용하여 다양한 실험을 수행할 수 있다. WGridDSP는 향후 국내에서 수행할 그리드 관련 연구에 기반 구조로 활용될 수 있으며, 특히 응용프로그램의 스케줄링 기법 연구에 필요한 표준 톨로 사용될 수 있을 것으로 기대된다.

6. 참고문헌

- [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, Vol. 14, pp. 1175-1220, 2002.
- [3] Anthony Sulistio, Chee Shin Yeo, and Rajkumar Buyya, "Visual Modeler for Grid Modeling and Simulation (GridSim) Toolkit", ICCS 2003, LNCS 2659, pp 1123-1132, 2003.
- [4] Rajkumar Buyya, "Economic-based Distributed Resource Management and Scheduling for Grid Computing," Ph. D. Thesis, Monash University, Melbourne, Australia, 2002.