

WGridSP: 그리드 컴퓨팅을 위한 웹 기반 스케줄링 플랫폼

강 오 한[†] · 강 상 성^{**}

요 약

본 논문에서는 웹 기반의 그리드 컴퓨팅 환경에서 시스템을 모델링하고 스케줄링 기법을 시뮬레이션 할 수 있는 그리드 스케줄링 플랫폼을 설계하고 구현하였다. 구현한 웹 기반 그리드 스케줄링 플랫폼은 자바 환경의 그리드 스케줄링 툴킷인 GridSim을 시뮬레이션 도구로 활용하였으며, 자원 모델링, 작업 모델링, 알고리즘 컴파일, 시뮬레이션, 성능분석을 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.

키워드 : 그리드, 스케줄링, 툴킷, 시뮬레이션, 성능분석, 자바

WGridSP: A Web-based Scheduling Platform for Grid Computing

Oh-Han Kang[†] · Sang-Seong Kang^{**}

ABSTRACT

In this paper, we designed and implemented a web-based grid scheduling platform(WGridSP), which can model a system and simulate scheduling scheme in grid computing. WGridSP used GridSim, a grid scheduling toolkit in java-environment, as a tool for simulation and is able to perform resource modeling, task modeling, algorithm compiling, simulation, and performance evaluation rapidly in web environment. WGridSP can be applied as a foundation for grid research and can be used to analyze the efficiency of scheduling algorithm.

Key Words : Grid, Scheduling, Toolkit, Simulation, Performance Evaluation, Java

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 어플리케이션들이 처리된다. 그리드 환경에서 시스템의 성능을 극대화하기 위해서는 사용자의 요구와 사용 가능한 자원의 성능을 파악하여 어플리케이션을 효과적으로 처리하도록 하는 스케줄러가 필요하다. 응용프로그램의 다양성과 자원의 이질성 등의 특성으로 인하여 그리드 환경에서는 어플리케이션에 대한 실행시간과 완료시간의 예측이 매우 어렵다. 이러한 그리드 시스템의 특성으로 인하여 시스템의 성능을 향상시키기 위해서는 그리드 연산 환경의 특성을 반영한 스케줄링 기법이 사용되어야 한다.

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 현 단계에서는 많은 어려움이 따른다. 이러한 문제를 해결하기 위해서 그리드 시스템과 사용자 요구조건, 어플리케이션 특성 등을 모델링하고 스케줄링 기법을 시뮬레이션할 수 있는 그리드 플랫폼이 연구되고 개발되어야 한다. 그리드 환경에서 처리되는 어플리케이션은 연산량, 사용자 요구, 통신 유형, 입출력 비율 측면에서 서로 다른 다양한 특성들을 가지고 있다. 이러한 상이한 특성을 갖는 어플리케이션은 서로 다른 스케줄링 알고리즘의 적용이 요구된다. 향후 그리드 시스템과 어플리케이션의 특성을 반영한 다양한 스케줄링 알고리즘들이 개발되어야 하며, 스케줄링 알고리즘의 성능을 분석하고 비교하기 위해서는 그리드 플랫폼 기반의 스케줄

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

† 종신회원: 안동대학교 컴퓨터교육과 교수

** 정 회 원: 안동대학교 교육대학원 컴퓨터교육전공

논문접수: 2006년 4월 13일, 심사완료: 2006년 8월 7일

링 툴이 개발되어야 한다.

현재까지 그리드 환경에서 작업을 스케줄링하는 다양한 알고리즘들이 제안되었다[2-4]. 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 잘못 사용하게 되면 왜곡된 결과를 산출할 수도 있다. 일반적으로 그리드 스케줄링 시뮬레이션에서는 그리드 환경, 작업, 사용자 환경 등이 고려되어야 한다. 그리드 환경에서는 자원의 수, 각 자원의 프로세서 할당 전략, 각 자원의 통신망 전송 속도, 각 자원을 구성하는 노드(시스템)의 수, 각 노드의 프로세서 수, 각 노드의 프로세서 처리 능력 등이 고려되어야 한다. 작업과 관련된 내용으로는 처리할 작업의 수, 각 작업의 수행길이(프로세서 시간), 각 작업의 입력데이터 크기, 각 작업의 출력데이터 크기, 각 작업의 요구 프로세서 수, 각 작업의 선후 관계 등이 고려되어야 한다. 사용자와 관련된 내용으로는 사용자 수, 사용자의 통신망 속도 등이 고려되어야 한다.

본 논문에서는 그리드 컴퓨팅 환경에서 시스템을 모델링하고 스케줄링 기법을 시뮬레이션 할 수 있는 웹 기반의 그리드 스케줄링 플랫폼을 구현하였다. 구현한 스케줄링 플랫폼은 자원 모델링, 작업 모델링, 알고리즘 컴파일, 시뮬레이션, 성능분석을 웹 환경에서 빠르게 처리할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 그리드 컴퓨팅 환경에서 사용할 수 있는 스케줄링 툴킷의 특징을 소개하고, 3장에서는 본 논문에서 제안하는 스케줄링 플랫폼의 구현에 관하여 기술한다. 그리고 4장에서는 구현된 스케줄링 플랫폼에서의 적용 예시를 보여주며, 5장의 결론으로 끝을 맺는다.

2. 관련 연구

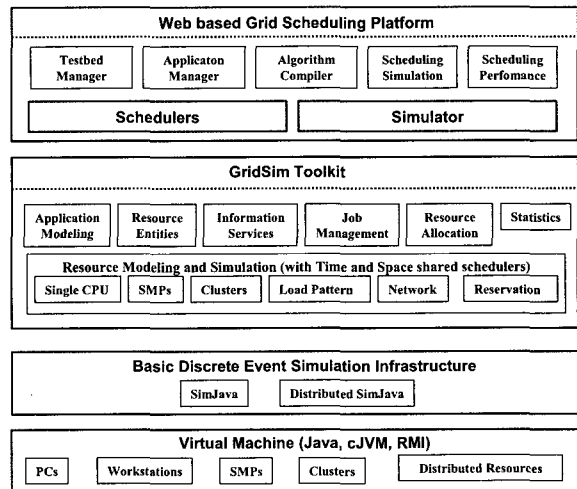
현재까지 그리드 컴퓨팅 환경에서 스케줄링 알고리즘을 시뮬레이션할 수 있는 다양한 툴킷들이 개발되었다[5-9]. 본 논문과 관련된 그리드 툴킷에 관한 연구는 미국에서 SDSC(San Diego Supercomputer Center)와 Virginia 대학의 GCG(Grid Computing Group)이 대표적이며, 호주에서는 Melbourne 대학의 GRIDS 연구실에서 연구를 활발히 추진하고 있다. SDSC에서는 CSAG(Concurrent Systems Architecture Group)과 같은 UC San Diego 대학의 연구 실험실과 공동으로 MicroGrid[5], SimGrid[6], TeraGrid 등의 툴킷을 개발하였다. 호주의 GRIDS 연구실에서는 GridSim[7]을 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 알고

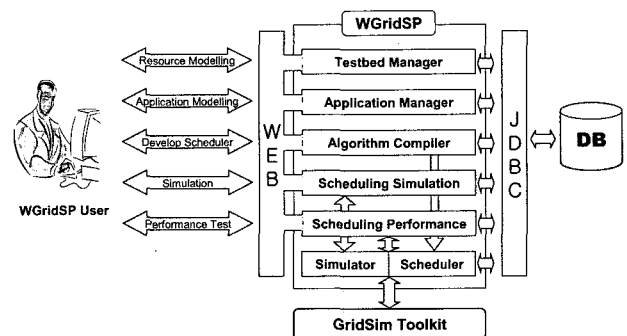
리즘을 시뮬레이션 하기 위해서 적절한 개발환경의 구축, 소스코드의 분석, 자원 모델링을 위한 프로그래밍, 작업집합 모델링을 위한 프로그래밍, 스케줄링 알고리즘 구현 등 여러 가지 작업을 필요로 한다. 이러한 작업들은 연구자들에게 중복작업을 유발하여 스케줄링 알고리즘 연구의 효율성을 저하시킨다. 호주의 GRIDS 연구실에서는 GUI 기반의 모델링 도구인 Visual Modeler(VM)[10]를 개발하여 역시 그들이 개발한 GridSim을 위한 자원모델링과 어플리케이션 모델링 소스코드를 쉽게 작성할 수 있도록 하였다. 그러나 VM[10]은 모델링 과정만을 다소 용이하게 할 뿐 자원 및 어플리케이션 정보의 저장과 관리, 시뮬레이션, 성능분석 등의 기능은 지원하지 않고 있다.

3. 웹 기반 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[11] 패키지를 바탕으로 함으로써 실용도와 안정성이 높은 그리드 스케줄링 툴킷이다. WGridSP는 사용자와 GridSim 사이



(그림 1) WGridSP와 GridSim과의 관계



(그림 2) WGridSP의 구조

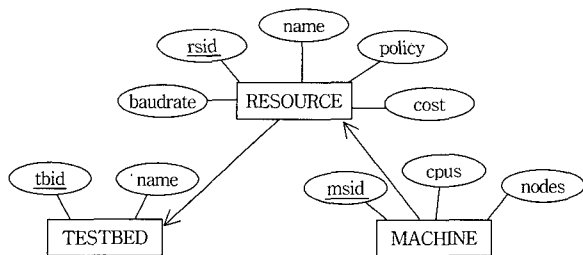
에서 웹을 매개로 한 인터페이스 역할을 함과 동시에 자원 모델링, 어플리케이션 모델링 자료와 스케줄링 알고리즘을 데이터베이스에 관리하여 재사용이 가능하도록 한다. (그림 1)은 WGridSP와 GridSim 툴킷과의 관계를 나타낸 것이며, (그림 2)는 5개의 기능단위로 구성된 WGridSP의 구조를 나타낸 것이다.

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 저장되고 시뮬레이션 및 성능분석에 사용된다. 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 비용을 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(CPUs): 머신이 가지는 프로세서의 수를 나타낸다. 프로세서의 수가 많을수록 동시에 처리할 수 있는 작업의 수가 많아진다.

(그림 3)은 테스트베드 관리자를 위한 데이터베이스 구조를 E-R(Entity-Relation) 다이어그램으로 나타낸 것이다.



(그림 3) 테스트베드 관리자 DB 구조

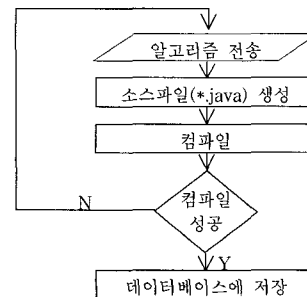
3.2 어플리케이션 관리자

어플리케이션 관리자는 각 어플리케이션을 구성하고 있는 작업들의 모델링을 지원한다. 사용자가 모델링한 어플리케이션 정보는 데이터베이스에 저장되어 시뮬레이션에 사용된다. 각 어플리케이션은 하나 이상의 작업(task)으로 구성되어 있으며 각 작업은 다음과 같은 속성을 가지게 된다.

- 길이(length): 작업의 CPU 요구시간으로 이 값이 클수록 프로세서에서의 처리시간이 길어진다.
- 입력데이터 크기(input size): 작업을 처리하기 위해 필요한 입력데이터(실행코드 포함)의 크기를 말한다. 스케줄러에 의해 할당된 자원에 전송해야 하므로 작업을 자원까지 전달하는데 소요되는 시간에 영향을 준다.
- 출력데이터 크기(output size): 자원이 작업의 처리를 완료하고 산출하는 데이터의 크기로서 자원에서부터 스케줄러까지 결과를 전송하는데 소요되는 시간에 영향을 준다.

3.3 스케줄링 알고리즘 관리

WGridSP는 스케줄링 알고리즘의 핵심 부분만을 필요로 한다. 알고리즘에 필요한 그리드 정보를 수집하는 소스코드는 컴파일 직전 자동 생성되며, 컴파일 결과를 사용자에게 전달하여 오류를 수정할 수 있게 한다. (그림 4)는 스케줄링 알고리즘의 컴파일 및 저장과정을 나타낸 것이다.



(그림 4) 컴파일 및 저장 과정

3.4 스케줄링 시뮬레이션

스케줄링 시뮬레이션을 위하여 필요한 테스트베드, 어플리케이션, 알고리즘을 선택하면 시뮬레이션 결과를 웹브라우저로 확인할 수 있다. 각 작업이 어떤 자원에 어느 시점에 전달되고 실행, 반환되는지 확인함으로써 알고리즘의 세부동작 상황을 분석할 수 있다. (그림 5)는 시뮬레이션 기능의 핵심 클래스인 시뮬레이터의 세부동작을 Pseudo 코드로 나타낸 것이다.

3.5 스케줄링 성능분석

스케줄링 성능분석은 하나 이상의 스케줄링 알고리즘 선택하고 성능분석에 사용할 테스트베드, 작업의 속성 범위를 선택하면 작업의 수에 따라 작업완료시간의 변화를 차트로 확인할 수 있다. 각 작업의 수행길이, 입출력 크기 등은 지

정한 범위 안에서 균일분포 또는 정규분포에서 난수를 발생하고, 정확한 측정값을 위해 지정한 반복횟수에 따른 평균값을 구해서 최종수행시간을 산출한다. (그림 6)은 성능분석 모듈의 처리과정을 Pseudo 코드로 나타낸 것이다.

```

Load resource and task informations from database;
GridSim initialize;
for i:=1 to ResourceList.size do
  Create a new resource;
  for j:=1 to MachineList[i].size do
    Create a new machine;
    for k:=1 to MachineList[i].cpus do
      Create a new processor;
      Add the processor to created machine;
    endfor
    Add the machine to created resource;
  end for
endfor
for i:=1 to TaskList.size do
  Create a new task;
endfor
Create the scheduler object requested by user;
Start GridSim simulation;
Send the simulation result to user;
    
```

(그림 5) 시뮬레이션 알고리즘

```

Create Resources with loaded resource information from database;
makespan_sum = 0;
for i:=TaskNum_Min to TaskNum_Max step TaskNum_Step do
  for loop:=1 to NoOfSimulation do
    Create a new resource;
    for j:=1 to i do
      length := random(from Length_Min to Length_Max);
      in := random(from Input_Min to Input_Max);
      out := random(from Output_Min to Output_Max);
      Create a new task with length, in, out;
    endfor
    Start GridSim Simulation;
    time := result of simulation;
    makespan_sum := makespan_sum + time
  end for
  makespan = makespan / NoOfSimulation;
  Draw the chart with makespan;
endfor
    
```

(그림 6) 성능분석 알고리즘

4. WGridSP의 특징 및 적용 예시

본 논문에서는 그리드 컴퓨팅 환경에서 시스템을 모델링하고 스케줄링 기법을 시뮬레이션 할 수 있는 웹 기반의 그리드 스케줄링 플랫폼(Web-based Grid Scheduling Platform: WGridSP)을 제안하고 구현하였다. 웹 기반의 그리드 스케줄링 알고리즘을 구현함으로써 그리드 컴퓨팅 환경과 함께 스케줄링 알고리즘의 모델링과 시뮬레이션이 가능하다. 본 논문에서 구현한 WGridSP는 테스트베드 관리자, 어플리케이션 관리자, 알고리즘 컴파일러, 스케줄링 시뮬레이션, 성능분석 모듈로 구성된다. WGridSP는 자원 모델링, 작업 모델링, 알고리즘 컴파일, 시뮬레이션, 성능분석을 웹 환경에서 빠르게 처리할 수 있다. 따라서 연구자들에게 알고리즘의 구현

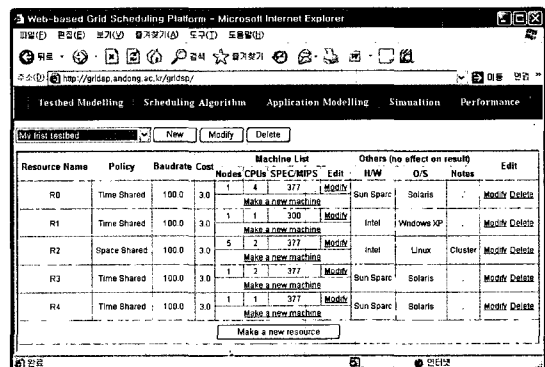
<표 1> GridSim과 WGridSP의 비교

| | GridSim | WGridSP |
|----------------------|------------------|---------------------------------|
| 컴파일/실행 환경 구축 | 필수 | 필요 없음 |
| 틀킷 구조 분석 | 필수 | 필요 없음 |
| 자원 모델링 어플리케이션 모델링 | 도구 형태 | 텍스트 에디터 자바 소스 |
| | 컴파일 | 웹에서 컴파일 |
| 스케줄링 알고리즘 구현 | 도구 형태 | 텍스트 에디터 자바 소스 |
| | 컴파일 | 웹에서 컴파일 |
| 시뮬레이션 | 실행 결과 | 웹에서 실행 콘솔 및 로그파일 |
| | 도구 | 텍스트 에디터 |
| 성능분석 | 실행 변수지정 결과 | 웹에서 실행 소스코드로 작성 콘솔 및 로그파일 |
| | | |

과 결과 분석에만 집중할 수 있는 환경을 제공한다. 본 논문에서 개발한 WGridSP를 자바 기반의 대표적 스케줄링 툴킷인 GridSim[7]과 비교하면 <표 1>에서와 같은 다양한 특징을 가지고 있다. WGridSP를 사용하여 자원 및 작업을 모델링하고 스케줄링 알고리즘을 적용하여 스케줄링의 동작 상황 및 성능을 분석하는 예를 보면 다음과 같다.

4.1 테스트베드 모델링

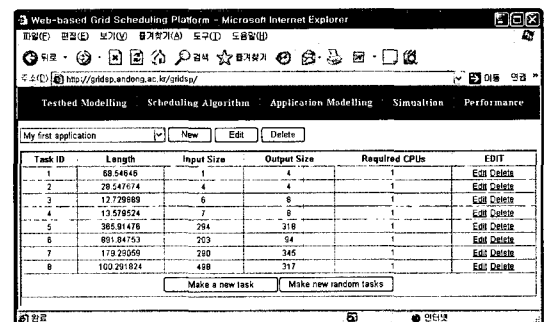
시뮬레이션을 위한 새로운 테스트베드를 생성하고 테스트 베드에 속한 자원들을 작성한다. 이 예시에서는 다섯 개의 자원을 작성하였고 세부 사항은 (그림 7)과 같다.



(그림 7) 테스트베드 모델링의 예

4.2 어플리케이션 모델링

어플리케이션 모델링을 할 때 처리해야할 작업 목록의 각 변수를 사용자가 직접 지정할 수 있고 범위를 지정한 후 난

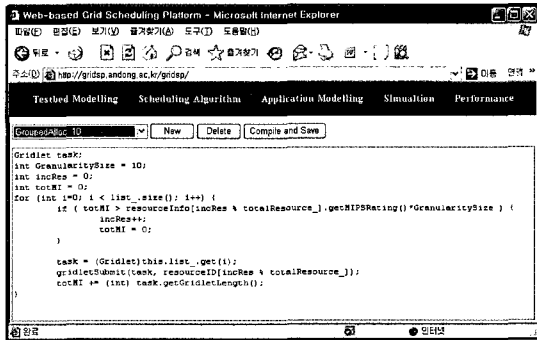


(그림 8) 어플리케이션 모델링의 예

수를 이용하여 생성할 수도 있다. (그림 8)은 8개 작업(task)의 변수를 난수를 이용하여 작성한 것이다.

4.3 알고리즘 컴파일

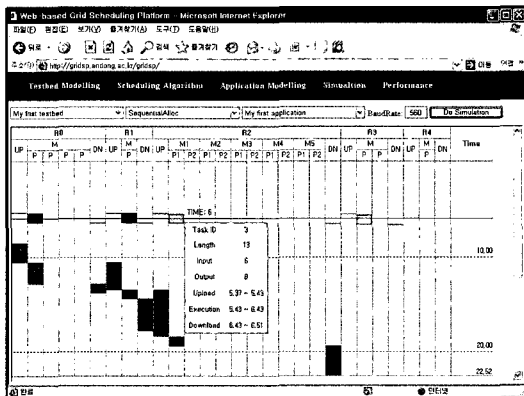
(그림 9)에 나타난 스케줄링 알고리즘은 Granularity 크기를 10으로 설정한 Grouping-Based Scheduling[12]을 WGridSP 환경에 맞게 적용한 것이다.



(그림 9) 스케줄링 알고리즘의 컴파일

4.4 스케줄링 시뮬레이션

위에서 작성한 테스트베드, 어플리케이션, 알고리즘을 선택하고 시뮬레이션 실행하면 (그림 10)과 같이 각 작업의 배정 결과와 실행 결과가 나타난다. [My first testbed]에 [My first application]을 SequentialAlloc 스케줄링 알고리즘으로 적용할 때 모든 작업이 종료되는데 소요된 시간은 22.52로 나타났다. 각 작업이 자원에 전달되어 수행되고 결과를 반환하는 과정이 (그림 10)과 같이 시간대별로 차트로 나타난다.

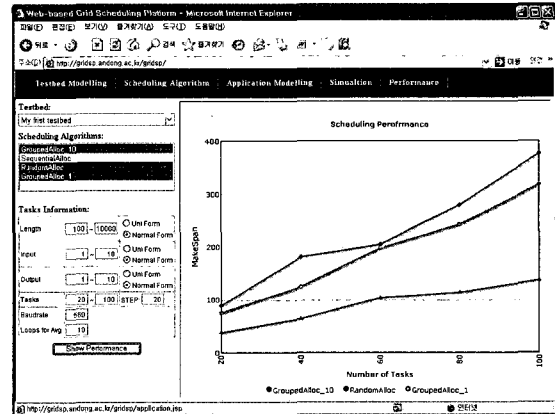


(그림 10) 시뮬레이션의 결과

4.5 스케줄링 알고리즘의 성능 비교

이 절에서는 WGridSP에 스케줄링 알고리즘을 구현하고 그에 따른 성능을 알아보았다. (그림 11)은 앞에서 모델링한 테스트베드인 [My first testbed]에서 세 개의 스케줄링 알고리즘을 구현한 후 성능을 비교한 결과이다. 그래프에서 X축은 작업의 수, Y축은 모든 작업을 완료하는데 소요된 시간을 나타낸다. 성능분석에 사용되는 작업의 길이, 입력의 크기, 출

력의 크기는 모두 정규분포를 따르도록 설정하였으며, 10회씩 시뮬레이션한 결과의 평균을 그래프로 나타낸 것이다. 분석 결과는 Granularity 크기를 10으로 설정한 Grouping-Based Scheduling[12]이 가장 우수한 것으로 나타났다.



(그림 11) 성능분석 결과

5. 결론

본 논문에서는 자바 환경의 그리드 스케줄링 툴킷인 GridSim을 시뮬레이션 도구로 활용하여 자원 모델링, 작업 모델링, 알고리즘 컴파일, 시뮬레이션, 성능분석을 웹 환경에서 빠르게 수행할 수 있는 웹 기반의 그리드 스케줄링 플랫폼(WGridSP)을 구현하였다. WGridSP는 웹 환경에서 그리드 스케줄링을 시뮬레이션 할 수 있어서 많은 시간과 노력을 절약할 수 있다. 향후에는 성능분석 기능을 보완하여 Y축으로는 연산비용, X축으로는 작업의 이질성, 자원의 이질성, 통신속도 등 다양한 변수를 선택하여 성능을 비교분석할 수 있게 할 예정이다.

참고 문헌

- [1] Foster and C. Kesselman, "The Grid: Blueprint for a Future Computing Infrastructure," Morgan Kaufmann Publishers, USA, 1999.
- [2] Rajkumar Buyya, "Economic-based Distributed Resource Management and Scheduling for Grid Computing," Ph. D. Thesis, Monash University, Melbourne, Australia, 2002.
- [3] Srikumar Venugopal and Rajkumar Buyya, A Deadline and Budget Constrained Scheduling Algorithm for eScience Applications on Data Grids, 6th International Conference on Algorithms and Architectures for Parallel Processing, pp. 60-72, 2005.
- [4] L. Lie, J. Zhan, and L. Li, A Runtime Scheduling Approach with Respect to Job Parallelism for Computational Grid, Pro. of 3rd International Conference on Grid and Cooperative Computing, pp.261-268, 2004.

[5] H. Song, et. al., "The MicroGrid: A Scientific Tool for Modeling Computational Grids," Proc. of IEEE Supercomputing (SC 2000), Nov., 2000.

[6] H. Casanova, "Simgrid: A Toolkit for the Simulation of Application Scheduling," Proc. of the 1st IEEE/ACM International Symposium on Cluster Computing and the Grid(CCGrid 2001), May, 2001.

[7] 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.

[8] K. Aida, et. al., "Performance Evaluation Model for Scheduling in a Global Computing System". The International Journal of High Performance Computing Applications, Vol.14, No.3, Saga Publications, 2000.

[9] Michael Walker. "A Framework for Scheduling Data-Parallel Applications in Grid Systems", MS Thesis, University of Virginia, 2001.

[10] 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.

[11] F. Howell and R. McNab, "SimJava: A Discrete Event Simulation Package for Java with Applications in Computer Systems Modelling", Proc. of 1st Int. Conference on Web-based Modelling and Simulation, Society for Computer Simulation, 1998.

[12] Nithiapidary Muthuvelu, Junyang Liu, Nay Lin Soe, Srikumar Venugopal, Anthony Sulistio and Rajkumar Buyya, "A Dynamic Job Grouping-Based Scheduling for Deploying Applications with Fine-Grained Tasks on Global Grids", Australian Workshop on Grid Computing and e-Research (AusGrid2005), Newcastle, Australia. Conferences and Practice in Information Technology, Vol.44, 2005.



강 오 한

e-mail : ohkang@andong.ac.kr

1982년 경북대학교 전자계열 전산모듈 (학사)

1984년 한국과학기술원 전산학과 (공학석사)

1992년 한국과학기술원 전산학과 (공학박사)

1984년~1994년 (주)큐닉스컴퓨터 선임/책임연구원

1994년~현재 안동대학교 컴퓨터교육과 교수

관심분야: 그리드 컴퓨팅, 태스크 스케줄링, OVPN 등



강 상 성

e-mail : edukang@andong.ac.kr

1998년 안동대학교 컴퓨터교육과(학사)

2001년 안동대학교 교육대학원 교육공학전공(이학석사)

2001년~현재 안동대학교 강사

관심분야: 그리드 컴퓨팅, 스케줄링 등