한국시뮬레이션학회 논문지
Vol. 15, No. 2, pp. 129-9 (2006. 12)

자원 신뢰성 측정을 통한 효율적인 그리드 자원 스케줄링 모델

박다혜[1†], · 이종식[1]

Efficient Grid Resource Scheduling Model with Resource Reliability Measurement

Da-Hye Park · Jong-Sik Lee

ABSTRACT

Grid computing has been appeared for solving large-scaled data which are not solved by a single computer. Grid computing is a new generation platform which connects geographically distributed heterogeneous resources. However, gathering heterogeneous distributed resources produces many difficult problems. Especially, to assure resource reliability is one of the most critical problems. So, we propose a grid resource scheduling model using grid resource reliability measurement. We evaluate resource reliability based on resource status data and apply it to the grid scheduing model in DEVSJAVA modeling and simulation. This paper evaluates parameters such as resource utilization, job loss and average turn-around time and estimates experiment results of our model in comparison with those of existing scheduling models such as a random scheduling model and a round-robin scheduling model. These experiment results showed that the resource reliability measurement scheduling model provides efficient resource allocation and stable job processing in comparison with a random scheduling model and a round-robin scheduling model.

Key words : Grid Computing, Computational Grid, Resource Reliabiliy, Grid Scheduling, DEVS modeling

요 약

그리드 컴퓨팅은 단일 컴퓨터로는 해결할 수 없는 대용량의 작업을 처리하기 위해 제안되었다. 그리드 컴퓨팅은 지리적으로 분산된 이기종 자원들을 상호 연결하여 대용량의 작업들을 처리하는 새로운 차세대 컴퓨팅이다. 그런데, 분산된 이기종의 자원들을 모을 때에 많은 어려운 문제들이 발생한다. 특히, 자원들의 신뢰성을 보장하는 것은 가장 심각한 문제 중에 하나이다. 그래서 우리는 그리드 자원의 신뢰성을 측정하여 자원을 할당하는 그리드 자원 신뢰성 측정 스케줄링 모델을 제안한다. 우리는 자원의 상태 정보를 기반으로 하여 자원 신뢰성을 측정하고, DEVSJAVA 모델링과 시뮬레이션 환경에서 그리드 시뮬레이션 모델에 그리드 자원 신뢰성 측정 방법을 적용하였다. 그리고, 이 논문은 스케줄링 모델들의 Utilization, Job loss, Throughput 그리고 Average Turn-around Time 같은 파라미터들을 측정하였고, 자원 신뢰성 측정을 이용한 그리드 자원 신뢰성 측정 스케줄링 모델의 실험 결과들을 기존의 스케줄링 모델들(랜덤 스케줄링 모델, 라운드 로빈 스케줄링 모델)과 비교하였다. 이 실험 결과들은 자원 신뢰성 측정 스케줄링 모델이 랜덤 스케줄링 모델과 라운드 로빈 스케줄링 모델에 비해 효율적인 자원 할당과 안정적인 작업 처리를 제공한다는 것을 보여준다.

주요어 : 그리드 컴퓨팅, 계산 그리드, 자원 신뢰성, 그리드 스케줄링, DEVS 모델링

1. 서 론

정보통신기술의 빠른 발전으로 컴퓨터의 성능은 꾸준히 향상되고 있다. 이와 동시에 컴퓨터가 처리할 자료의 크기도 점점 커지면서 이젠 하나의 컴퓨터로 처리하기 힘든 대용량의 자료들이 등장하게 되었는데 바로 이런 대용량의 자료들을 처리하기 위해 새롭게 등장한 기술이 바로

* 본 연구는 정보통신부 및 정보통신 연구진흥원의 대학 IT연구센터 육성 · 기원사업의 연구결과로 수행되었음.
2006년 6월 26일 접수, 2006년 10월 19일 채택
[1] 인하대학교 컴퓨터정보공학과
주 저 자 : 박다혜
교신저자 : 박다혜
E-mail; parkdh@inhaian.net

그리드 컴퓨팅[1,2]이다. 그리드 컴퓨팅은 지리적으로 분산된 고성능 컴퓨터, 첨단 실험 장비, 데이터베이스 등의 이기종 자원들을 고속 네트워크로 연동하여 대용량의 컴퓨팅 문제들을 해결하는 기술이다. 즉, 그리드 컴퓨팅은 가상조직의 중심에 있고 단지 컴퓨팅뿐만 아니라 네트워크상에 분산된 다양한 계산 자원이나 정보자원을 가상조직의 멤버가 하나의 가상 컴퓨터로서 이용하는 환경이 되는 것이다.

그리드 컴퓨팅은 응용분야에 따라 크게 세가지로 나눌 수 있다. 많은 자원을 연결하여 계산을 해결할 수 있는 계산 그리드(Computational Grid)[3], 원격지의 분산된 자료들을 통합하여 분석할 수 있게 해 주는 데이터 그리드(Data Grid), 그리고 분산 처리를 필요로 하는 어플리케이션을 위한 엑세스 그리드(Access Grid)로 나눌 수 있는데, 우리는 이 분야들 중 계산 그리드상에서의 문제에 대해 논의한다.

계산 그리드에서는 대용량의 자료를 처리하기 위해 지리적으로 분산된 이기종의 자원들을 모아서 자료를 처리한다. 그런데 이런 지리적으로 분산된 이기종의 자원들을 모아서 자료를 처리할 때 많은 어려운 문제들이 발생한다. 그 중 가장 심각한 문제가 바로 자원의 비신뢰성이다. 계산 그리드 환경에서의 각 자원들이 자신의 담당 기능을 원활하게 수행할 수 있는 능력이 바로 자원 신뢰성이다. 그러나 분산되어 있는 이기종의 자원들간에 신뢰성을 확보하기란 어려운 문제이다. 그러므로 그리드 자원의 신뢰성을 보장해 줌으로써 그리드 컴퓨팅의 성능을 향상시킬 수 있고, 그리드 응용 프로그램의 이용률을 높일 수 있다. 그래서 이 논문에서는 자원 신뢰성을 측정하여 효율적으로 작업을 처리하는 그리드 자원 스케줄링 모델을 구성하였다.

논문의 구성은 다음과 같다. 2장에서는 이 논문의 관련 연구를 설명하고, 3장에서는 자원 신뢰성 측정 알고리즘을 제시하고, 자원 신뢰성 측정 알고리즘을 이용한 그리드 자원 스케줄링 모델을 소개한다. 그리고 4장에서는 DEVS 환경[4]에서 다른 기존 스케줄링 모델과의 비교 실험을 통해 자원 신뢰성 측정 알고리즘을 이용한 그리드 자원 스케줄링 모델의 효율성을 증명하고 마지막 5장에서는 결론을 맺는다.

2. 배경 이론 및 관련 연구

2.1 그리드 스케줄링

그리드 컴퓨팅이 발전하면서 그리드 스케줄링[5]의 필요성은 계속 증가하고 있다. 그리드 컴퓨팅 환경은 다양한 자원들로 구성되어 있고, 그리드 유저가 보낸 작업을 그 자원들에게 할당하여 빠른 처리 속도와 안정적인 작업 처리를 보장하기 위해서는 그리드 스케줄링이 필요하기 때문이다. 이러한 그리드 스케줄링은 크게 동적 스케줄링과 정적 스케줄링으로 나눌 수 있다.

동적 스케줄링(Dynamic Scheduling)은 자원의 상태가 유동적일 때 사용하는 방식으로 다시 시행 시점에 따라 온라인 방식과 배치 방식으로 나눌 수 있다. 온라인 방식은 유저로부터 들어온 작업을 도착 즉시 처리하는 방식으로 대표적으로 MCT(Minimum Completion Time), SA(Switching Algorithm) 등이 있다. 그러나 동적 스케줄링은 구현이 복잡하고 오버헤드가 발생할 수 있다.

정적 스케줄링(Static Scheduling)은 자원의 상태가 고정되어 있을 때 사용하는 방법으로 작업을 처리하기 전에 모든 자원과 파라미터들을 요구한다. 우리는 이러한 두 스케줄링의 단점들을 보완하여 그리드 자원의 상태 정보를 이용한 효율적인 그리드 스케줄링 모델을 구현하였다.

2.2 그리드 자원 신뢰성

그리드 컴퓨팅은 다양한 조직으로부터 형성된 동적인 가상 조직(Virtual Organization)으로서 자원을 공유하여 문제를 해결한다는 개념으로 분산되어 있는 이기종의 자원들 중 어떤 자원을 선택하여 일을 처리하는지에 따라 일의 처리 시간과 비용이 차이가 날 수 있다. 그러므로 대용량의 분산된 그리드 컴퓨팅에서 어떤 자원을 선택하여 일을 처리할지를 결정하는 것은 매우 중요한 문제이다.

이러한 문제를 해결하기 위해 경제 이론(economic theory)을 응용한 그리드 정책을 많이 사용하고 있는데, 응용되고 있는 대표적인 경제 모델[6]로는 상품 시장 모델, 매매 교섭 모델, 경매 모델 등이 있다. 그러나 이러한 경제 이론을 사용한 그리드 모델들은 자원 할당시에 신뢰성을 위한 정책을 포함하고 있지 않다. 그래서 최근에는 자원 신뢰성[7,8,9]을 보장하기 위한 많은 그리드 연구가 진행되고 있다.

신뢰성은 그 모델의 정확성이나 정밀성을 나타내며, 안정성(Stability), 일치성(consistency), 예측 가능성(Predictability)과 동일한 의미를 갖는다. 그리드 컴퓨팅 상에서 신뢰성을 보장한다는 것은 그리드 컴퓨팅 자원을 통해 일을 처리하는 소비자들이 좀더 빠르고 안정적으로 일을 처리해주는 자원을 보장 받는다는 것을 의미한다. 즉, 그리드 자원 신뢰성이란 각 자원들이 작업 처리시 자신의 능력을 최대로 발휘하여 빠르고 안정적인 작업 처리를 할 수 있는 것을 의미한다. 그러므로 그리드 자원 신뢰성을 보장해 줌으로써 그리드 컴퓨팅의 성능을 향상시킬 수 있고 그리드 응용 프로그램의 이용률

을 높일 수 있다.

그러나 그리드 환경하에서 많은 이기종 자원들의 신뢰성을 보장하는 것은 매우 어려운 일이다. 왜냐하면 그리드 컴퓨팅 내의 자원들은 서로 다른 정책을 갖고 있는 분산된 자원들이기 때문이다. 그래서 그리드 자원 신뢰성을 보장하기 위해서는 모든 그리드 자원들에 적용될 수 있는 정책이 필요하다.

현재 그리드 자원 신뢰성을 위한 연구에서 사용하고 있는 정책으로는 Trust Mechanism[10,11]이 있다. Trust Mechanism은 자원에 credit을 부여하고, 작업 처리시 자원을 부여된 credit을 이용하여 선택하는 방법이다. 그러나 이 방법은 자원 제공자가 credit을 속일 수 있는 문제가 발생할 수 있다. 그래서 우리는 각 자원에서 발생하는 상태 데이터를 바탕으로 자원의 신뢰성을 평가하고, 이를 그리드 자원 스케줄링에 적용하여 효율적이고 안정적인 작업 처리를 보장하였다.

2.3 DEVS Modeling

DEVS(Discrete Event System Specification)형식론이란 연속적인 시간상에서 발생하는 이산 사건 모델을 위해 정의된 모델링 방법론이다. 이는 계층적이고 모듈화된 이산 사건 모델들을 위해 정의된 이론으로 시스템이 일반적으로 갖는 특성들을 정의하여 시스템 모델링을 할 수 있는 기반을 제공한다. 따라서 동적인 특성을 가진 그리드 컴퓨팅 시스템의 추상화와 모델링에 적합하다. 또한, 단순한 시간 측정이 아닌 연속적인 시간흐름의 제어를 통해 보다 효율적인 연속 시간선상의 시스템 설계가 가능하다.

DEVS 형식론은 크게 기본 모델과 결합 모델로 나눌 수 있는데, 먼저 기본 모델은 시간에 따른 시스템의 동작을 표현하며, 그 구성은 다음과 같다.

$$M = <X, S, Y, \delta int, \delta ext, \lambda, ta>$$

여기서 X는 입력 사건 집합을 S는 순차적인 상태들의 집합을 Y는 출력 사건 집합을 의미한다. 그리고 δint는 내부 상태 전이 함수를 의미하고, δext는 외부 상태 전이 함수를 나타낸다. 또한, λ는 외부 출력 사건이 발생하는 출력 함수이며, ta는 시간 진행 함수이다.

그 다음 결합 모델은 시스템의 구성 요소 간 상호 작용을 표현하기 위한 모델로서 기본 모델이나 결합 모델로 기술된 구성 요소 모델을 연결하여 만든 모델이다. 아래는 결합 모델의 구성을 나타낸다.

$$N = <X, Y, D, \{Md \mid d \in D\}, EIC, EOC, IC, Select>$$

여기서 X는 입력 사건 집합을 Y는 출력 사건 집합을 D는 구성 요소 이름 집합을 의미한다. 또한, Md는 모델의 구성 요소를 나타내며, EIC는 외부 입력 관계를 나타내고, EOC는 외부 출력 관계를 나타내며, IC는 내부 입출력 관계를 나타낸다. 본 논문에서는 수시로 변화하는 그리드 환경에서 작업 처리를 표현하기 위해서 DEVS (Discrete Event system Specification) 형식론을 적용하였다.

3. 자원 신뢰성 측정 스케줄링 모델

이 절에서는 그리드 자원 스케줄링에서 사용한 자원 신뢰성 측정 알고리즘을 설명하고, 이를 기반으로 한 그리드 자원 신뢰성 측정 스케줄링 모델의 구성을 살펴본다.

3.1 그리드 자원 신뢰성 측정 알고리즘

그리드 자원 신뢰성을 보장하기 위해 우리는 자원의 과거 상태 데이터를 모아서 신뢰성을 측정하였다. 자원의 상태는 할당된 작업의 수에 따라 측정되는데, 이에 따라 총 3가지 상태로 나누었다. 자원의 상태는 High-Status, Middle-Status 그리고 Low-Status로 나뉘어진다. High-Status는 세가지 상태 중 가장 높고 안정적인 서비스를 제공할 수 있는 상태를 나타내며, Middle-Status는 High-Status보다는 낮은 서비스를 제공하지만 Low-Status보다는 높은 서비스를 제공하며, 작업 처리시 지연 시간이 발생할 수 있다. 마지막으로, Low-Status는 자원의 상태가 가장 좋지 않을 때 발생하며 할당된 작업이 제대로 처리될지 보장받을 수 없게 된다.

이렇게 세가지 상태로 자원의 상태를 평가하고 각각의 상태 데이터들을 모아서 각 자원의 과거 상태 데이터들을 구성한다. 과거 상태 데이터는 그림 1과 같이 총 3가지로 나눌 수 있다.

과거 상태 데이터에는 H_{high}, H_{middle}, 그리고 H_{low}가 있다. H_{high}는 자원이 High-Status에 머물러 있던 시간들을 모아놓은 데이터이고, H_{middle}은 자원이 Middle-Status에

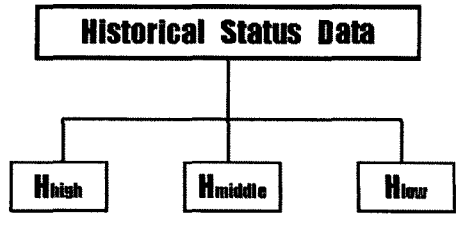

그림 1. 과거 상태 데이터

머물러 있던 시간들을 모아놓은 데이터이며, 마지막으로 H_{low}는 자원이 Low-Status에 머물러 있던 시간들을 모아 놓은 데이터이다. 이 세 가지 과거 상태 데이터들을 기반으로 그림 2의 자원 신뢰성 측정 계산식을 이용하여 각각의 그리드 자원들의 신뢰성을 측정한다.

계산 그리드에서는 서로 다른 성능을 가진 다양한 자원들이 존재하게 된다. 그러므로 여러 자원 중 성능이 좋은 자원들에게 자원을 할당 하는 것이 무엇보다도 안정적이고 빠른 작업 처리를 위해서는 반드시 필요하게 되는데, 위의 그림 2의 자원 신뢰성 측정 계산식을 이용하면 각 자원들의 성능을 측정하는 것이 용이해진다. 계산 그리드 상에 존재하는 자원들은 다양한 성능을 가지고 있기 때문에 각 자원들이 세 가지 상태에 머물러 있던 시간들도 차이가 발생한다. 특히, 그 세 가지 상태 중 가장 높은 성능을 제공하는 High-Status 상태에 머물러 있던 시간들이 많을 수록 그 자원의 성능이 높다는 것을 알 수 있다. 그러므로 각 자원들의 세 가지 상태중 가장 높은 성능을 발휘하는 High-Status 상태에 머물러 있던 데이터가 각 자원의 신뢰성을 측정하는데 가장 중요한 요소가 될 수 있기 때문에 H_{high} 값을 수식의 분자에 위치시켜 자원의 신뢰성을 측정하였다. 위의 자원 신뢰성 측정 계산식으로 각 자원의 신뢰성을 측정한다. 그 결과 가장 높은 값을 갖는 자원은 High-Status 상태에 머물러 있던 시간이 많다는 것을 의미한다. 또한 그 자원에 작업을 할당하면 빠르고 안정적인 작업 처리가 가능하다는 것을 의미하기 때문에 신뢰성이 가장 높다는 것을 의미한다. 그래서 우리는 가장 높은 신뢰성 측정 값을 보이는 자원에 작업을 할당함으로써 그리드 컴퓨팅 환경에서 안정적이고 빠른 작업 처리를 보장할 수 있게 된다.

3.2 자원 신뢰성 측정 스케줄링 모델

그리드 자원 신뢰성 측정 알고리즘을 사용한 그리드 자원 스케줄링의 기본 구성은 아래 그림 3와 같다.

기본적으로 그리드 유저, 코디네이터, 자원 그리고 측정기 등 총 4개의 컴포넌트로 구성되어 있다. 그리고 각 컴포넌트가 하는 일은 다음과 같다.

그리드 유저는 일정한 간격으로 코디네이터에게 작업을 보낸다. 코디네이터는 그리드 유저로부터 받은 작업을 자원에 할당한다. 그리고 코디네이터는 각 자원들의 측정기로부터 받은 자원 상태 정보를 기반으로 신뢰성 측정 수식을 이용하여 신뢰성을 측정하고, 가장 빠르고 안정적인 작업 처리를 할 수 있는 자원, 즉, 자원 신뢰성 측정 수식 결과가 가장 높은 자원을 선택하여 작업을 보낸다. 자원들은 코디네이터로부터 받은 작업을 처리하는데 이 때 각 자원들은 한번에 하나씩 작업을 처리할 수 있다. 그리고 작업 처리시 코디네이터로부터 받은 작업은 자신의 큐에 보관하고, 그 큐에 보관된 작업들의 개수를 기반으로 하여 자원의 상태를 측정한다. 이 자원의 상태를 측정하는 것은 각 자원들과 연결된 측정기가 측정을 하는데, 이 측정기는 해당 자원의 상태를 해당 자원의 큐에 보관된 작업의 개수에 따라 측정하여, 상태 정보가 바뀔 때마다 코디네이터에게 자원의 상태 정보를 전달한다.

그리드 자원 신뢰성 측정 알고리즘을 사용함으로써 성능이 좋은 자원들에게 자원을 할당하게 되어, 자원의 처리를 효율적으로 할 수 있게 된다. 즉, 다양한 성능을 가진 자원들이 존재하는 그리드 컴퓨팅 환경에서 빠른 작업 처리를 보장하는 자원을 선택하여 작업을 처리함으로써 안정적이고 빠른 작업 처리를 보장할 수 있다.

$$\text{Resource Reliability} = \frac{H_{high}}{H_{high} + H_{middle} + H_{low}}$$

그림 2. 자원 신뢰성 측정 계산식

4. 실험 및 결과 분석

4.1 시뮬레이션 방법

우리는 앞 절에서 설명한 자원 신뢰성 측정 계산식을 이용하여 시뮬레이션 모델을 구현하였다. 이 시뮬레이션 모델의 효율성과 안정성을 증명하기 위해 DEVSJAVA 시뮬레이션 모델링을 사용하였다. 그리고 DEVS 환경에서의 자원 신뢰성 측정 스케줄링 모델의 구성은 아래 그림 4와 같다.

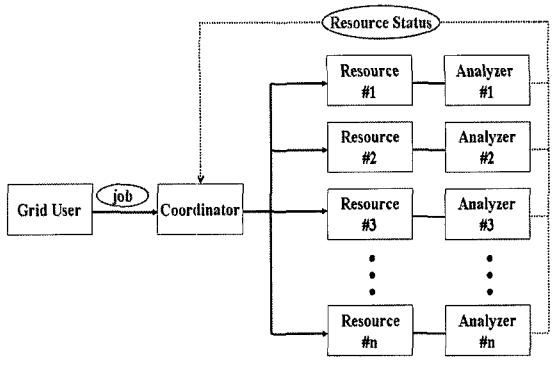

그림 3. 자원 신뢰성 측정 스케줄링 모델의 구성

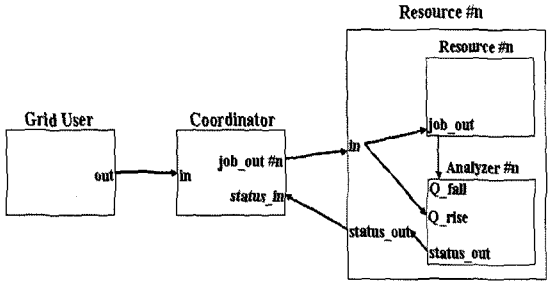

그림 4. DEVS 환경에서의 자원 신뢰성 측정 스케줄링 모델

자원 신뢰성 측정 스케줄링 모델은 기본적으로 일정한 간격으로 작업을 발생시키는 그리드 유저, 그리드 유저로부터 작업을 받아 각 자원들의 측정기로부터 받은 상태 정보를 기반으로 하여 신뢰성을 측정하고 각 자원들에게 작업을 할당하는 코디네이터, 코디네이터로부터 작업을 받아 자신의 처리 시간동안 작업을 처리하는 자원, 그리고 자원의 상태를 측정하여 코디네이터에게 상태 정보를 보내주는 측정기로 구성되어 있다. 하나의 코디네이터에 복수 개의 자원들이 연결되어 있어 각 자원들의 상태 정보를 이용하여 각 자원들의 신뢰성을 측정하고, 측정값이 높은 자원에 작업을 할당한다.

그리드 자원 스케줄링 모델의 유스케이스 다이어그램은 아래 그림 5와 같다. 그리드 유저는 그리드 서비스, 즉, 처리가 필요한 작업들을 코디네이터에게 전달한다. 코디네이터는 전달 받은 작업을 자원에게 할당하는데, 할당하기 전에 각 자원의 신뢰성을 측정하여 가장 높은 신뢰성을 갖는 자원에게 작업을 할당한다. 자원은 할당 받은 작업을 자신의 처리 시간(Processing Time)동안 실행하고, 만약 작업을 처리하는 동안 다른 작업이 들어 왔을 때는 그 작업을 자신의 큐(Queue)에 보관한다. 그리고 자신의 큐 안에 기다리고 있는 작업의 수를 측정기에게 전달한다.

측정기는 자원으로부터 받은 정보를 기준으로 상태를 측정하고, 상태가 바뀌었을 때 마다 코디네이터에게 그 상태에 머물러 있던 시간을 전달한다. 그리고 그 전달 받은 상태 정보를 통해 코디네이터는 다시 자원의 신뢰성을 측정한다.

우리는 자원 신뢰성 측정 모델의 유용성을 증명하기 위해 랜덤 스케줄링 모델과 라운드 로빈 스케줄링 모델을 추가로 구현하여 같은 실험 환경에서 기록된 실험 결과를 비교하였다.

우리는 총 5개의 자원들을 구성하였고, 총 10000개의 작업을 발생시켜 각 모델들의 작업 처리 결과들을 비교하

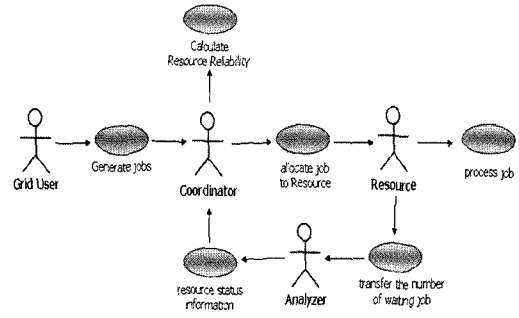

그림 5. 자원 신뢰성 측정 스케줄링 모델의 유스케이스 다이어그램

표 1. 각 자원들의 성능

Resource	Processing time	High Status	Middle Status	Low Status
Resource 1	3	QS < 10	10<=QS<20	QS >= 20
Resource 2	20	QS < 7	7<=QS<12	QS >= 12
Resource 3	2	QS < 10	10<=QS<20	QS >= 20
Resource 4	30	QS < 3	3<=QS<6	QS >= 6
Resource 5	25	QS < 4	4<=QS<8	QS >= 8

였다. 그리고 그리드 컴퓨팅에는 다양한 자원들이 존재할 수 있기 때문에 5개의 자원들의 성능을 서로 다르게 구성하였는데 그 구성은 위 표 1에서 보여진다. 우리는 각 자원의 큐에 있는 작업의 수에 따라 상태를 측정하고 각 상태에 머물러 있던 시간을 측정기에서 측정하여 코디네이터에게 그 값을 전달한다. 코디네이터는 그 전달된 값을 기반으로 하여 앞 절에서 제안한 자원 신뢰성 측정 수식을 이용하여 각 자원의 신뢰성을 측정한다. 그리고 이때, 각 자원의 큐 사이즈 기준 또한 자원의 성능이 다르기 때문에 서로 다르게 된다. 위 표의 Resource 1을 보면 큐 사이즈가 10 미만 이면 High Status를 의미하며, 10이상 20 미만의 값을 가지면 Middle Status를 의미하고, 20이상이면 Low Status를 의미한다. 각각 이 High Status, Middle Status, Low Status에 머물러 있던 시간을 코디네이터에게 전달하여 코디네이터는 이 전달된 수치를 이용하여 신뢰성을 측정하고 높은 신뢰성 수치를 기록한 자원에게 작업을 할당한다.

4.2 시뮬레이션 결과

첫 번째 실험은 스케줄링 모델들의 Utilization을 측정하였다. Utilization은 아래의 수식을 사용하여 측정한다.

$$Utilization = \frac{(\sum_{i=1}^{n} R_n \cdot JS_n) \times PT_n}{ST}$$

여기서 n은 총 자원의 개수를 가리키고, R_n은 n번째 자원을 의미하며 JS_n은 n번째 자원의 작업처리 개수를 나타낸다. 그리고 PT_n은 n번째 자원의 처리 시간을 의미하며, ST는 시뮬레이션 시간(Simulation Time)을 의미한다.

이 수식을 이용하여 Utilization을 측정한 결과는 그림 6과 같다.

이 결과를 보면 신뢰성 측정을 통한 그리드 자원 스케줄링 모델이 다른 두 랜덤 스케줄링 모델과 라운드 로빈 스케줄링 모델에 비해 Utilization이 높게 나타나는걸 볼 수 있다. 신뢰성 측정 스케줄링 모델은 86.5%의 Utilization을 보인 반면 랜덤 스케줄링 모델과 라운드 로빈 스케줄링 모델은 약 79%의 Utilization을 기록하였다. 이는 신뢰성 측정 스케줄링 모델이 다른 두 스케줄링 모델에 비해 빠른 작업 처리가 가능한 자원을 선택하여 작업을 처리하고 있음을 의미하며, 단위 시간당 자원들의 작업 처리량이 높다는 것을 증명해준다. 그러므로 각 자원의 신뢰성을 측정하여 작업을 할당해줌으로써 그리드 컴퓨팅에서의 효율적인 작업 처리가 가능할 수 있음을 보여준다.

두 번째 실험은 세 스케줄링 모델의 Throughput을 측정하였다. 그림 7은 세 스케줄링 모델의 throughput을 비교한 그림이다. 먼저 신뢰성 측정 스케줄링 모델은 높은 throughput을 보인 반면 랜덤 스케줄링 모델과 라운드 로빈 스케줄링 모델은 신뢰성 측정 스케줄링 모델보다 50% 낮은 throughput을 나타내고 있다. 이는 신뢰성 측정 스케줄링 모델이 주어진 자원의 신뢰성을 측정함으로써 다양한 자원들로 구성되어 있는 그리드 컴퓨팅 환경에서 빠른 작업 처리가 가능한 자원을 선택하여 작업을 처리하고 있음을 증명하는 것이다. 그러므로 그리드 컴퓨팅 환경에서 효율적인 작업 처리를 보장해 줄 수 있다.

세 번째 실험은 각 스케줄링 모델들의 Job Loss를 측정하였다. 이 실험의 결과는 그림 8에 나타나 있다. 랜덤 스케줄링 모델과 라운드 로빈 스케줄링 모델이 각각 약 47.5%와 47%의 Job Loss 율을 보인 반면 신뢰성 측정 스케줄링 모델은 약 34%의 Job Loss 율을 기록하였다. 이는 우리가 제안한 신뢰성 측정 스케줄링 모델이 다른 두 스케줄링 모델에 비해 안정적으로 작업을 처리해 주고 있음을 증명하는 결과이다. 즉, 다양한 자원들이 존재하는 그리드 컴퓨팅상에서 안정적으로 작업을 처리해 주는 자원을 선택함으로써 그리드 컴퓨팅의 신뢰성을 보장해준다.

네 번째 실험은 세 스케줄링 모델의 Average Turn Around Time을 측정하여 비교하였다. 그림 9는 Average Turn

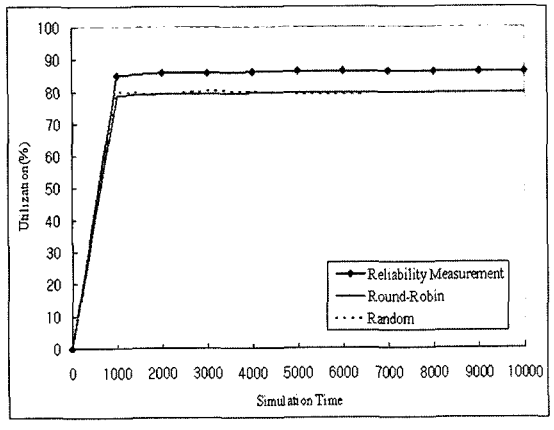

그림 6. 스케줄링 모델들의 Utilization

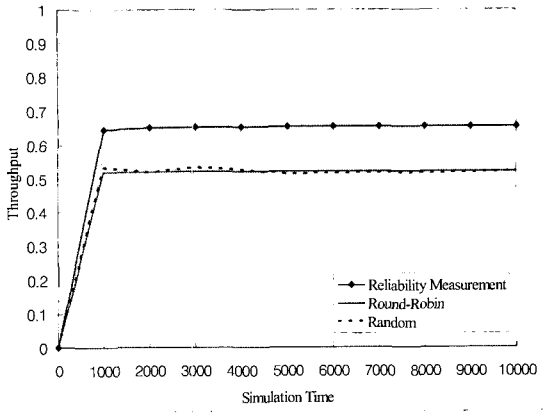

그림 7. 스케줄링 모델들의 Throughput

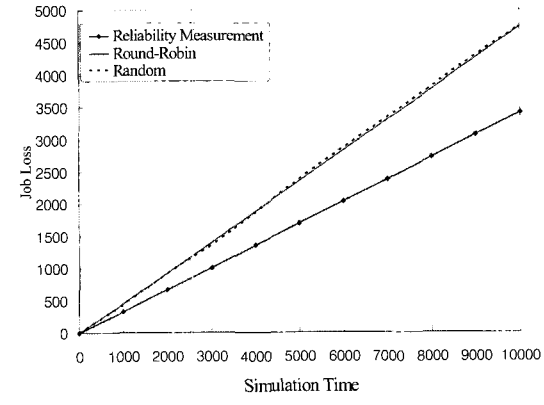

그림 8. 스케줄링 모델들의 Job Loss

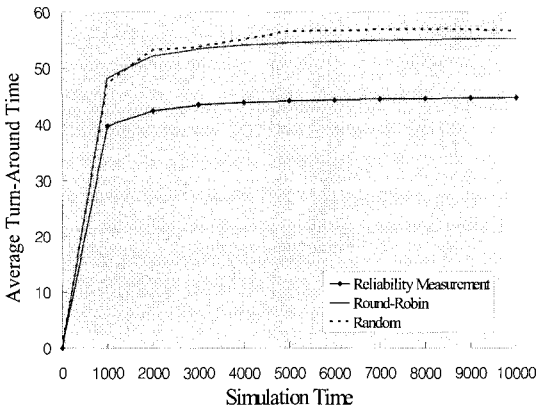

그림 9. 스케줄링 모델들의 Average Turn Around Time

Around Time을 비교한 결과를 보여준다. 신뢰성 측정 스케줄링 모델이 다른 두 스케줄링 모델에 비해 낮은 Average Turn Around Time을 기록하고 있다. 랜덤 스케줄링 모델이 약 56.7, 라운드 로빈 스케줄링 모델이 약 55의 Turn Around Time을 기록한 반면 신뢰성 측정 스케줄링 모델은 44.5의 낮은 Turn Around Time을 기록하였다. 이는 우리가 제안한 신뢰성 측정 스케줄링 모델이 다른 두 스케줄링 모델들에 비해 빠른 작업 처리를 보장해주고 있음을 증명하는 결과이다. 이 결과는 자원 신뢰성 측정 스케줄링 모델을 이용하면 그리드 컴퓨팅 상에서의 자원들 중 빠르게 작업을 처리해주는 자원들을 선택하여 작업을 처리 할 수 있음을 보여준다.

5. 결 론

그리드 컴퓨팅은 이기종의 자원들로 구성되어 있다. 그리고 이 이기종 자원들을 이용하여 슈퍼 컴퓨팅으로 처리해야 할 대용량의 작업들을 처리하는 새로운 컴퓨팅 기술이 바로 그리드 컴퓨팅이다. 그런데 그리드 컴퓨팅 환경의 이용률과 활용도를 높이기 위해서는 자원의 신뢰성을 보장하는 것이 중요하다. 그리드 자원 신뢰성이란 계산 그리드 상의 각 자원들이 주어진 환경 아래에서 자신의 담당 기능을 원활하게 수행할 수 있는 능력을 일컫는데 분산된 이기종의 자원들 간에 신뢰성을 확보하기란 어려운 문제이다. 그래서 그리드 자원의 신뢰성을 보장하는 방법으로 다양한 연구가 진행되고 있는데 우리는 자원 신뢰성 측정 알고리즘을 이용한 그리드 자원 스케줄링 모델을 제안하였다.

우리는 그리드 자원의 신뢰성을 측정하기 위해 각 자원의 과거 상태 정보를 이용하였다. 각 자원의 상태를 총 3가지로 나누어 자원의 상태를 측정하였고, 각 상태에 머물러 있던 시간을 측정하여 과거 상태 정보를 측정하였다. 이 측정된 과거 상태 정보를 이용하여 각 자원의 신뢰성을 계산하였다. 각 자원의 신뢰성은 그리드 자원 신뢰성 측정 알고리즘을 사용하여 계산하였다. 자원 신뢰성 측정 알고리즘은 가장 높은 결과 값을 갖는 자원이 신뢰성이 높다는 것을 의미하고, 우리는 그 자원에 작업을 할당하여 효율적이고 안정적인 작업 처리를 보장하였다.

우리는 그리드 자원 신뢰성 측정 스케줄링 모델의 유용성과 효율성을 증명하기 위해 DEVSJAVA 모델링 환경에서 실험을 하였다. 그리고 제안한 모델의 결과를 다른 기존 스케줄링 모델들(랜덤 스케줄링 모델, 라운드 로빈 스케줄링 모델)과 비교하여 총 4개의 파라미터 값을 측정하였다. 이 비교 실험을 통해 우리가 제안한 신뢰성 측정 스케줄링 모델이 기존 랜덤 스케줄링 모델과 라운드 로빈 스케줄링 모델과 비교하여 빠르고 안정적인 작업 처리를 보장해 줄 수 있음을 보였다. 이것은 신뢰성 측정 스케줄링 모델이 그리드 컴퓨팅 환경에서 자원을 할당하고 관리하는데 매우 안정적이고 효율적이라는 것을 증명하는 결과이다.

하지만 우리가 제안한 신뢰성 측정 스케줄링 모델은 다량의 자원들이 존재하는 그리드 컴퓨팅 환경에서 자원의 상태 정보를 보관하고 관리하는데 오버헤드가 발생할 수 있다. 이 문제점을 해결하기 위해 하나의 코디네이터만으로 상태 정보를 관리하고 신뢰성을 측정하지 않고, 추후 실험을 통해 일정 범위 내의 일정 개수의 자원들을 관리하는 코디네이터를 여러 개 배치함으로써 오버헤드를 줄일 수 있는지를 실험할 예정이며, 이 논문에서 비교 실험한 랜덤 스케줄링 모델과 라운드 로빈 스케줄링 모델 이외의 다른 그리드 스케줄링 모델(Trust Mechanism Model)들과 비교 실험을 진행할 예정이다.

참 고 문 헌

1. Foster and C. Kesselman: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers (1998)
2. F. Berman, G. Fox and T. Hey: Grid computing: making the global infrastructure a reality. J. Wiley. New York (2003)
3. Xiu-Chuan Wu, Liang Hu, Jiu-Bin Ju: ACGWPRS: the

active computational grid framework. Machine Learning and Cybernetics, 2003 International Conference on Volume 2,(2003) 994-999
4. Zeigler, B.P., et al: The DEVS Environment for High -Performance Modeling and Simulation. IEEE CS & E, Vol. 4, No3 (1997) 61-71
5. Carsten Ernemann, Volker Hamscher, Uwe Schwiegelshohn, Ramin Yahyapour and Achim Streit: On Advantages of Grid Computing for Parallel Job Scheduling. 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'02), May 21-24, 2002, Berlin, Germany
6. R. Buyya: Economic-based Distributed Resource Management and Scheduling for Grid Computing. Available at http://www.buyya.com/thesis/ (April 2002)
7. Liu, C., Yang, L., Foster, I., Angulo, D.: Design and evaluation of a resource selection framework for grid applications. Proceedings of the 11th IEEE International Symposium on High-Performance Distributed Computing (HPDC 11) (2002)
8. Li, C., Xiao, N., Yang, X.: Application availability measurement in computational grid. Proceedings of the 2nd workshop on Grid and Cooperative Computing (GCC2003), Springer LNCS 3032 (2003) 151-154
9. Xuanhua Shi, Hai Jin, Weizhong Qiang, and Deqing Zou: Reliability Analysis for Grid Computing. Proceedings of the 3rd workshop on Grid and Cooperative Computing (GCC 2004), Springer LNCS 3251 (2004) 787-790
10. S Kamvar, EigenRep: Reputation Management in P2P Networks. Technical Report, SCCM-02-16, Stanford University (2002)
11. Junzhou Luo, Peng Ji, Xiaozhi Wang, and Ye Zhu: A Novel Method of QoS Based Resource Management and Trust Based Task Scheduling. CSCWD 2004, LNCS 3168 (2005) 21-32

박 다 혜 (parkdh@inhaian.net)
2005 인하대학교 컴퓨터공학부 학사
2005～현재 인하대학교 컴퓨터정보공학과 석사과정

관심분야 : 시스템 모델링 및 시뮬레이션, 그리드 컴퓨팅

이 종 식 (E-mail 기재 요망)
1993 인하대학교 전자공학과 학사
1995 인하대학교 전자공학과 석사
2001 애리조나대 컴퓨터공학과 박사
2001～2002 캘리포니아 주립대학교 전기·컴퓨터공학과 전임강사
2002～2003 클리블랜드 주립대학교 전기·컴퓨터공학과 조교수
2003～현재 인하대학교 컴퓨터공학부 조교수

관심분야 : 소프트웨어공학, 시스템 모델링 및 시뮬레이션, 그리드 컴퓨팅