

그리드 컴퓨팅 환경에서의 자원 관리를 위한 분산화된 브로커 기반 모델

마용범^{1†} · 이종식¹

Decentralized Broker-Based Model for Resource Management in Grid Computing Environment

Yong-Beom Ma · Jong-Sik Lee

ABSTRACT

Resource management in grid computing environment is essential for integration and interaction among heterogeneous resources. This paper discusses resource management methods of centralized and decentralized broker-based modeling for solving complex problems of resource management and presents design and development of the decentralized broker-based resource management modeling in grid computing environment. This model comprises a global resource broker and a local resource broker, and we derive reduction of communication and functional dispersion of job management using a local resource broker. The simulation experiment shows the improvement of resource utilization and average response time and proves that this model improves utilization of resources and replies to user requests promptly.

Key words : Grid computing environment, Resource management, DEVS modeling and simulation

요 약

그리드 컴퓨팅 환경에서 자원 관리는 이기종의 자원들 사이에서 통합과 상호작용을 위해 필수적이다. 이 논문은 자원 관리의 복잡한 문제들을 해결하기 위해 집중화된 브로커 기반과 분산화된 브로커 기반 모델링의 자원 관리 기법을 논하고 그리드 컴퓨팅 환경에서 분산화된 브로커 기반 자원 관리 모델링의 설계와 구현을 표현한다. 이 모델은 글로벌 자원 브로커와 로컬 자원 브로커로 이루어지고 우리는 통신의 감소와 로컬 자원 브로커를 사용한 작업 관리의 기능적인 분산을 이끌어낸다. 시뮬레이션 결과는 자원 활용도와 평균 응답 시간의 향상을 보여주고 이 모델이 사용자 요구에 즉각 응답하고 자원의 활용도를 향상시킨다는 것을 증명한다.

주요어 : 그리드 컴퓨팅 환경, 자원 관리, DEVS 모델링 및 시뮬레이션

1. 서 론

유비쿼터스 세상이 가능하려면, 어디를 가든 사물이나 행동을 인식하고, 신호를 처리하는 칩과 그 칩이 보내오는 데이터를 처리하는 처리기와 저장공간 등 사람과 사물, 그리고 작업을 실제로 수행하는 컴퓨터가 하나로 연결되어 있어야 한다. 이러한 컴퓨팅 환경을 구현함에 있어 하드웨어, 소프트웨어, 데이터는 물론 네트워크로 연결된 대

규모의 인적 자원들을 동적으로 자유롭게 관리하고 활용할 수 있도록 하는 그리드 컴퓨팅^[1]이 기반 기술로 각광받고 있다. 그리드 컴퓨팅은 휴먼 자원의 활용을 통한 효율성 증대, 이기종 자원의 통합, 데이터 접근의 용이성, 빠른 처리 속도 등의 장점으로 인해 의료 서비스, 금융서비스, 제조, 통신, 교육 등 다양한 분야에서 활용되고 있으며 많은 기업과 단체들의 연구가 이루어지고 있다.

그리드 컴퓨팅은 지리적으로 분산된 컴퓨팅 자원들을 활용하여 거대하고 고성능의 네트워크를 사용할 수 있게 하기 때문에, 모든 자원들을 관리하기 위한 방법이 필요하다. 그리드 컴퓨팅 환경에서 서로 다른 자원 소유주들이 서로 다른 정책 하에 그들의 자원을 관리하고 자원 사용자와 자원 공급자의 목적은 서로 다른 경우가 발생할 수 있다. 그러한 상황은 그리드 애플리케이션들이 종종

* 이 논문은 인하대학교의 지원에 의하여 연구되었음.

2006년 6월 26일 접수, 2006년 9월 2일 채택

^{1†} 인하대학교 컴퓨터공학부

주 저자 : 마용범

교신저자 : 마용범

E-mail; myb112@hanmail.net

자원들의 동시 할당을 요구한다는 사실에 의해 더욱 더 복잡하게 된다. 게다가 자원을 스케줄링하는 소프트웨어 하부 구조가 서로 다르기 때문에 매우 동적인 환경적 특성을 가진다. 따라서 그리드 컴퓨팅 환경에서는 거의 무한대에 이르는 이러한 다양한 자원들의 공유와 활용을 위해서 그리드 자원을 효율적으로 할당하고 처리하는 메커니즘^[5,10]이 필요하다.

이 논문은 그리드 컴퓨팅 환경에서의 자원 관리^[10]를 위한 분산화된 브로커 기반 모델을 제안한다. 우리는 성능 측정을 위해 DEVS 방법론^[2,3]을 적용한 이벤트 드리븐 방식으로 시뮬레이션 모델을 구성하고 실험하였다. 특히, 이 모델에서 자원의 관리와 스케줄링을 위해 그리드 자원 사용자와 그리드 자원 공급자 사이에 GRACE(Grid Architecture for Computational Economy)^[8]라고 불리는 경제학 구조 기반의 그리드 자원 관리 브로커^[6]를 구성한다. 또한, 그리드 자원 관리 시스템 성능을 향상시키기 위해 분산화 기법^[14]을 적용한 로컬 자원 브로커를 제안하고 이를 통해 사용자의 자원 사용에 대한 요청을 조정하고 그리드 자원 중개자의 기능을 분산시킨다. 또한, 기존의 시장형 경제 모델인 GRACE 기반 자원 관리 모델들과 달리 TAC SCM^[1]에서 에이전트의 하나인 MinneTAC^[12,13]의 두 가지 판매 전략을 그리드 자원 공급자의 자원 공급 전략에 적용하여 그리드 자원 브로커의 자원 선택을 돕는다. 위의 두 가지 알고리즘을 통해 기존 GRACE 기반 자원 관리 모델에서 그리드 자원 중개자의 거래 할당과 관리 기능을 분산시키고 사용자와 공급자 모두를 고려하는 자원 거래를 제공한다. 이것은 그리드 컴퓨팅 환경에서 자원의 활용을 높이고 거래를 활성화시키는 이점을 제공한다.

2. 연구 배경

2.1 그리드 자원 관리 모델

그리드 컴퓨팅 환경에서 사용자들은 동시다발적으로 자원 사용을 요청하고 주어진 기한과 비용 내에서 자원의 사용을 할 수 있기를 기대한다. 이러한 사용자의 요구는 매우 다양하고 그에 따라 자원의 공급과 요구의 통제를 통해 동적인 자원의 할당이 필요하게 되었다. 이에 따라 지금까지 개발된 그리드 자원 관리 모델은 계층형 모델과 시장형 모델로 분류된다. 계층형 모델은 그리드 컴퓨팅 환경에서 자원과 미들웨어의 관계를 계층적 구조로 표현하고 관리하는 데 효과적이다. 시장형 모델은 기존의 경제 모델을 그리드 자원 관리에 적용한 모델로서 자원 검색, 공유, 스케줄링 등을 제공하는 GRACE와 사용자 예

산과 작업 종료기한 등을 통해 자원의 검색과 할당을 실시하는 Nim rod/G^[9] 등이 있다. 분산화된 브로커 기반 모델은 그리드 자원을 수용하는데 적합한 계층형 구조를 제공하고 자발적이고 동적인 자원 할당을 실시하기 위해 그리드 자원 공급자, 소비자 그리고 그들을 중개하는 그리드 자원 중개자를 가진 GRACE구조를 기반으로 한다.

GRACE는 자원 관리에 경제적 원리를 도입하여 시장 기반 시스템의 잠재적인 이익 창출에 대한 이해를 돕고 의사결정 처리가 모든 사용자와 자원 소유자에 걸쳐 분산되어 있어 높은 확장성을 가진다. 그리드 컴퓨팅 환경에서의 서비스 접근 비용을 결정하기 위해 자원 거래를 사용한 다양한 GRACE 구조 기반의 자원 관리 모델들^[7,8]이 개발되었다. 대표적으로 상품 시장 모델, 매매 교섭 모델, 경매 모델, 이중 경매 모델 등이 있다. 상품 시장 모델에서는 자원 공급자가 가격을 공시하고 중개자를 통해 적합한 공급자와 거래가 이루어진다. 매매 교섭 모델에서는 상품 시장 모델과 반대로 소비자가 조건을 제시하고 중개자를 통해 적합한 공급자를 찾는다. 경매 모델은 위의 두 모델이 소비자나 공급자 한 쪽 측면만을 고려하는 단점을 해결한다. 거래는 소비자와 공급자 간의 입찰을 통해 이루어진다. 마지막으로 이중 경매 모델은 소비자와 공급자 사이에서 여러 번의 입찰을 통해 서로의 조건을 맞추어나가 최종적으로 양측 모두를 만족시키는 모델이다. 이러한 GRACE 기반의 그리드 자원 관리 방법은 다수의 소비자와 공급자 사이에 하나의 중개자를 포함한다. 따라서 집중화된 자원 관리 구조를 가지게 되고 이것은 거의 무한대에 이르는 자원을 관리하는 그리드 컴퓨팅 환경에서 자원 관리의 중요성과 관련이 있다. 이 논문에서 우리는 로컬 자원 브로커라는 분산화된 브로커를 통해 그리드 자원 중개자의 기능을 분산시킴으로써 자원 관리의 성능을 향상시킨다.

2.2 분산된 네트워크 관리 구조

분산된 환경에서 네트워크의 적절한 기능, 유지, 보안, 통제 그리고 데이터와 결합 관리를 보장하기 위해 네트워크 관리 시스템이 필요하다. 네트워크 관리 시스템은 분산되고 동적이고 유연성이 있는 구조를 갖기 위해 분산화와 위임의 기능을 적용하기도 한다. 분산화는 네트워크 관리에서의 다양한 기능을 효과적으로 분산시키는 역할을 하고 분산된 네트워크 관리 구조^[11]는 이를 위해 관리자, 서버, 관리 에이전트로 구성된다. 널리 알려진 네트워크 관리 구조들은 위치와 데이터베이스의 분할, 네트워크 관리자 사이의 정보 수집의 세분화에 따라 다르다. 그리

한 네트워크 관리 구조는 크게 중앙 집중화된 네트워크 관리, 계층형 네트워크 관리, 상호 네트워크 관리, 완전 분산된 네트워크 관리의 네 가지로 나누어진다. 이 논문에서는 네트워크 관리를 자원 관리에 응용하기 위해 중앙 집중화된 형태의 GRACE 구조와 완전 분산된 네트워크 관리 구조를 결합하여 기능 분산의 이점을 제공하는 분산화의 장점을 얻고 이를 통해 N:N의 관계에서도 더욱 활발한 자원 거래의 제공과 휴면 자원을 감소시키는데 중점을 둔다.

중앙 집중화된 네트워크 관리 구조에서는 오직 하나의 중앙 네트워크 관리자가 관리를 지시하고, 네트워크의 상태를 결정하기 위해 네트워크 컴포넌트들에게 쿼리를 보낸다. 이 경우에 중앙 집중화된 데이터베이스는 네트워크 관리자와 함께 존재하고 각 네트워크 관리 에이전트들과 통신한다.

계층형 네트워크 관리 구조에서도 중앙의 네트워크 관리자가 존재하지만 하위 관리자를 돕는 역할을 한다. 하위 관리자는 중앙의 관리자의 기능 중 일부를 담당하게 되고 지속적으로 중앙 관리자와 각 네트워크 관리 에이전트들과 통신을 한다.

상호 네트워크 관리 구조에서는 네트워크 관리자들의 시기적절한 상호작용을 통해 네트워크의 서로 다른 도메인을 관리한다. 데이터베이스는 분할되어 있고 각 네트워크 관리자 사이에서 부분적으로 복제된다. 이 구조에서 네트워크 관리자는 그들의 도메인 내에 있는 네트워크 관리 에이전트와 통신한다.

완전 분산된 네트워크 관리 구조에서는 많은 네트워크 관리자들이 네트워크의 관리를 수행한다. 이 경우 데이터베이스는 고도로 분할되고 복제되며 각 네트워크 관리자는 각각 할당된 관리자 에이전트와 통신하고 다른 네트워크 관리자들과도 통신하게 된다.

2.3 DEVS 방법론

그리드 컴퓨팅 환경에서 자원은 수시로 상태가 변화하고 이러한 자원을 상태의 변화에 따라 효율적으로 할당하고 관리하는 것이 필요하다. 본 논문에서는 그리드 환경에서 수시로 변화하는 자원 요구를 표현하기 위해서 DEVS (Discrete Event system Specification) 방법론^[3]을 적용하였다. DEVS 방법론은 연속적인 시간상에서 사건을 발생시키는 이산 사건 시스템의 시뮬레이션을 위한 모델링 방법론으로서 시스템을 모듈화하고 계층화하여 분석하기 위한 수학적 방법을 제공한다. 따라서 그리드 컴퓨팅

환경에서 자원 거래가 이루어지는 것을 거래 요청과 처리, 완료 등의 사건 기반의 모델링하는 것이 가능하고 동시에 각 모델의 상태 변화에 따른 작업 처리가 가능하다.

DEVS 모델은 기본 모델(basic model)과 결합 모델(coupled model)로 분류된다. 기본 모델은 시스템의 행동을 모델링하기 위하여 사용되며 다음과 같이 구성된다.

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$$

X는 외부 입력 사건의 집합을 의미하고 Y는 출력 사건 집합을 의미하며 S는 순차적인 상태 집합을 의미한다. $\delta_{int}(S)$ 는 내부 전이 함수를 가리키고 $\delta_{ext}(Q, X^b)$ 는 외부 전이 함수를 가리킨다. 여기서, Q의 수학적 정의는 $Q = \{(s,e) \mid s \in S, 0 \leq e \leq ta(s)\}$ 와 같다. e는 최종 상태 전이까지의 경과시간을 나타낸다. X^b 는 X의 요소 집합이다. $\lambda(S)$ 는 외부 출력 사건이 발생하는 외부 함수이며, ta(S)는 시간 진행 함수이다.

기본 모델의 상태는 하나의 사건에서 다음 사건 발생까지 지속된다. 다음 사건의 시간 변수(tN)은 최종 사건 시간(tL)과 시간 진행 함수 ta(S)의 합이다. ta(S)는 시스템이 상태 S에 머무르는 동안의 시간 간격을 통합한 것이고 외부 사건이 발생하지 않을 때, 경과 시간(e)는 상태에서 머무르는 시간이다. 상태변수 시그마(σ)는 시스템이 상태 S에서 머무른 시간을 나타내며 ta(S) - e로 표현할 수 있다. 상태 전이 함수는 외부 전이 함수 $\delta_{ext}(S, e, X^b)$, 내부 전이 함수 $\delta_{int}(S)$ 로 구성된다. X^b 는 외부 입력 사건 집합 X의 요소들의 집합을 말한다. $\delta_{ext}(S, e, X^b)$ 는 단일/복합 입력들을 대상으로 한다. $\delta_{ext}(S, e, X^b)$ 는 약간의 입력들을 처리하고 현재 상태에서부터 다음 상태와 경과시간(e)를 기술한다. $\delta_{int}(S)$ 는 내부 사건들과 함께 다음 상태를 기술한다. 출력 함수 $\lambda(S)$ 는 현재 상태를 기반으로 하여 출력을 발생시킨다.

DEVS 방법론의 결합 모델은 각각의 컴포넌트 모델들을 상술하고 컴포넌트 모델간의 연결 관계와 상호작용을 표현한다. 연결 명세집합은 외부입력 연결 집합과 외부출력 연결 집합, 내부 연결 집합을 다룬다. 외부입력 연결 집합은 결합 모델의 입력 포트를 결합 모델 내 컴포넌트의 하나 이상의 입력 포트에 연결한다. 외부출력 연결 집합은 컴포넌트의 출력포트를 결합 모델의 하나 이상의 출력포트에 연결한다. 내부입력 연결 집합은 컴포넌트의 출력포트를 결합 모델 내 또 다른 컴포넌트의 입력 포트에 연결한다.

3. 분산화된 브로커 기반 모델

이 논문은 GRACE 기반의 자원 거래 구조를 모델링하고 로컬 자원 브로커 기반의 분산된 자원 관리를 자원 거래에 적용한다. 자원 거래를 실행하기 위해 이 논문은 시간의 흐름에 따른 자원 거래를 보여주고 분산화된 브로커 기반의 자원 거래 모델을 구현한다. 분산화된 브로커 기반 모델은 지역별로 위치한 로컬 자원 브로커와 이들과 통신하는 하나의 글로벌 자원 브로커를 포함한다. 또한, 다양한 구매 성향을 가진 자원 사용자들과 서로 다른 자원 공급 전략을 가진 자원 공급자들을 포함하고 자원 거래는 자원 사용자의 자원 요청이 브로커를 거쳐 공급자에게 전달되고 공급자로부터 사용자에게 전달되는 것을 의미한다.

이 장에서 우리는 분산화된 브로커 기반의 자원 관리 모델의 구조를 살펴보고 중요한 구성원이 되는 글로벌 자원 브로커와 로컬 자원 브로커의 기능을 설명한다. 또한, 그리드 자원 공급자의 자원 공급 알고리즘을 제안하고 분산화된 브로커 기반의 자원 관리에 적용한다.

3.1 분산화된 브로커 기반의 자원 관리 모델 구조

그리드 컴퓨팅 환경의 시장형 자원 관리 모델에서 자원 거래는 자원 사용자와 자원 공급자 사이에 위치한 그리드 자원 중개자를 통해 이루어진다. 자원 사용자들은 지역별로 분산되어 있고 서로 다른 구매 성향을 가지며 자원 공급자는 공급 전략에 따라 구분된다. 자원 사용자들은 지속적으로 자원의 사용을 요청하고 중앙의 그리드 자원 중개자에게 가격, 요구량, 만기일, 요청일, 타입 등의 정보를 포함한 주문 정보를 전달한다. 이 정보는 그리드 자원 브로커의 중개를 통해 적절한 자원 공급자에게 전달된다. 그리드 자원 브로커는 적절한 자원 공급자를 선택하기 위해 자원 공급자의 상태를 모니터링하고 사용자로부터 받은 주문 정보를 기반으로 적절한 자원 공급자와 연결해준다. 그리고 자원 공급자는 그리드 자원 중개자로부터 받은 주문에 따라 자원 사용자에게 자원을 공급하고 그에 따른 비용을 지불 받는다.

분산화된 브로커 기반 모델은 기존의 그리드 자원 중개자의 역할을 하는 글로벌 자원 브로커를 통해 자원 거래가 이루어진다. 자원 사용자와 공급자는 각각 구매 성향과 공급 전략에 따라 매우 다양할 수 있다. 그림 1은 분산화된 브로커 기반 모델의 구조를 보여준다. 그리드 자원 사용자들은 지속적으로 자원의 사용을 요청하고 주문 정보에 따라 세 개의 그룹으로 구분된다. 기존의 시장형

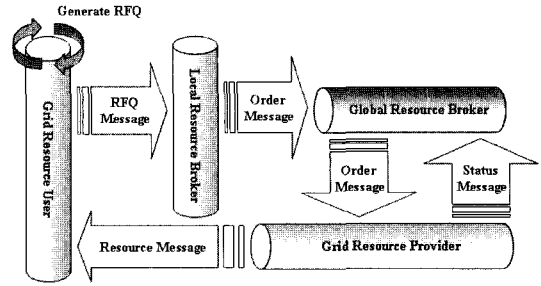


그림 1. 분산화된 브로커 기반 모델의 기본 구조

자원 관리 모델과 다르게 이 모델은 자원 사용자와 글로벌 자원 브로커 사이에 지역별로 로컬 자원 브로커를 둔다. 로컬 자원 브로커는 각 지역 내 사용자들의 자원 사용에 대한 요청을 조정하는 역할을 하고 이를 위해 지역 내 사용자들과 통신을 하게 된다. 우리는 이것을 지역 내 통신이라 부른다. 그룹 내에서 각 사용자들의 자원 사용에 대한 요청이 많을수록 글로벌 자원 브로커와의 통신이 잦아지게 되고 우리는 이것을 지역 외 통신이라 부른다. 글로벌 자원 브로커와 로컬 자원 브로커가 담당하는 역할은 3.2장에서 자세하게 살펴본다.

3.2 글로벌 자원 브로커와 로컬 자원 브로커

기존의 시장형 자원 관리 모델에서 그리드 자원 중개자는 자원 사용자와 자원 공급자 사이에서 중개인의 역할을 한다. 일반적으로는 자원 발견, 자원 선택, 주문 스케줄링과 모니터링 등을 담당한다. 자원 발견을 위해 그리드 자원 중개자는 모델 내 모든 공급자와 연결되어 있고 자원 상태 정보를 지속적으로 모니터링한다. 실제로 모든 공급자들이 사용자의 요구를 만족시킬 수는 없기 때문에 사용자의 요구와 적절하게 맞는 공급자를 선택하는 것이 필요하다. 자원 사용에 따른 비용과 시간, 요구량 등을 포함하는 사용자의 요구 확인을 위해 자원 선택이 이루어지고 자원 공급자와 거래한다. 주문 스케줄링은 거래 실행의 비용을 최소화 시키는 것을 목적으로 하고 자원 발견, 자원 선택, 그리고 거래의 할당을 담당한다. 그리드 컴퓨팅 환경은 동적인 특성을 가지기 때문에 우리는 그러한 환경 하에서 자원의 상태 변화를 모니터링하기 위해 지속적으로 갱신된 공급자의 상태 정보를 받는다.

분산화된 브로커 기반 모델에서 글로벌 자원 브로커는 기존의 그리드 자원 중개자와 유사하게 자원 사용자와 자원 공급자 사이에서 중개인의 역할을 하지만 그 기능은 조금 다르다. 기존의 시장형 자원 관리 모델에서는 자원 발견과 선택, 주문 스케줄링과 모니터링이 모두 중앙의

그리드 자원 중개자에 의해 수행된다. 그러나, 분산화된 자원 브로커에서는 로컬 자원 브로커가 주문 스케줄링의 기능을 수행한다. 분산화된 자원 브로커에서 글로벌 자원 브로커는 각 로컬 자원 브로커에 의해 조정된 주문을 받고 자원의 상태 변화를 모니터링하기 위해 지속적으로 갱신된 공급자의 상태 정보를 받는다.

로컬 자원 브로커는 그룹화되어 전달되는 사용자의 주문 정보를 평균을 내어 글로벌 자원 브로커에게 주문한다. 이것은 지역 외 통신을 위한 사용자와 글로벌 자원 브로커간 주문 정보의 전달을 감소시킨다. 또한, 공급자와 글로벌 자원 브로커간의 주문횟수를 줄이고 주문량은 비슷하게 유지시킨다. 자원 공급자는 한 번에 보다 많은 양의 자원 거래를 실행하게 되고 한 번의 주문으로 여러 사용자들에게 자원을 공급할 수 있게 된다. 이것은 공급자에게 자원 활용도를 향상시킬 수 있다는 이점을 제공하고 중개자에게 보다 많은 주문을 처리할 수 있다는 이점을 제공한다.

3.3 자원 공급 알고리즘

우리는 그리드 자원 공급자의 자원 공급 전략 중 하나로 요청된 주문으로부터 기대되는 이익인 Profit이 최대화되는 거래를 우선적으로 선택하는 MaxEProfit^[12,13] 전략을 적용한다. Profit은 $Profit = price - cost$ 로부터 구할 수 있고 요청된 거래에서 그리드 자원 사용자가 제시하는 자원의 사용에 따른 가격(price)에서 실제 시장에서 형성된 비용(cost)을 뺀 자원 하나당 이익을 말한다. 비용은 그리드 자원 공급자가 자원을 생산하는데 필요한 생산 원가를 의미한다. 따라서 가격이 변동됨에 따라 이익이 변하게 되고 비용은 모든 자원 공급자 사이에서 동일하다. 그리드 자원 브로커는 자원당 이익이 최대가 되는 거래를 MaxEProfit 전략을 가진 그리드 자원 공급자에게 주문하기 위해 그리드 자원 사용자의 거래 요청이 이루어질 때마다 Profit을 측정한다. TAC SCM에서는 주문으로부터 기대되는 이익을 계산하고 가격을 결정하기 위해 식 (1)^[12]을 사용한다. (1) 식에서 제안가, 요구량, 거래승인 확률, 리드 타임, 예약가, 페널티, 상품 타입 등으로 이루어진 고객 주문을 받을 기대 확률인 $P(order)$ ^[12]를 설정한다.

$$E[Profit] = Profit \times P(order) \quad (1)$$

그리드 자원 공급자의 두 번째 자원 공급 전략으로 거래 요청 가격을 거의 고려하지 않고 주문량이 최대인 거래를 우선적으로 선택하는 DemandDriven^[12,13] 전략을

적용한다. 주문량은 각 거래에서 그리드 사용자가 제시하는 자원의 요구량과 같으며 그리드 공급자는 이러한 사용자의 요구량만큼 충분한 자원을 공급할 수 있다고 가정한다. 따라서 그리드 자원 브로커의 각 거래의 주문량은 그리드 사용자의 요구량과 항상 같다.

4. 실험 및 결과 분석

우리는 그리드 컴퓨팅 환경에서 자원 거래 모델의 성능을 평가하기 위해 거래 수의 변화량, 응답 시간의 변화량, 그리고 처리시간 활용도를 측정하였다. 또한 메시지 이벤트 트리븐 방식으로 동작하는 DEVJAVA^[2]를 사용하여 그림 2와 같이 분산화된 브로커 기반 시뮬레이션 모델을 구성하고 신뢰성 있는 실험을 위해 약 10000개의 자원 요청 데이터를 추출하였다. 실험에서 기존의 시장형 자원 관리 모델은 집중화된 브로커 기반 모델이라고 표현하였다.

4.1 분산화된 브로커 기반 시뮬레이션 모델 구성

분산화된 브로커 기반 시뮬레이션 모델은 그림 2와 같이 네 가지 컴포넌트로 구성된다. 그리드 사용자(Grid User)는 각 지역별로 서로 다른 구매 성향을 가진다. 우리는 DEVS 모델링의 특성인 각 모델의 동적인 특성을 반영하여 그리드 사용자를 구매 성향에 따라 A, B, C로 구분하여 모델링한다. 지역 내에는 성향 별로 많은 사용자가 존재하고 이들은 구매 성향에 따라 그룹화되어 자원 사용을 요청한다. 이들은 자원의 사용 요청을 위해 지역 내 로컬 브로커에게 주문 정보가 저장된 Request Message를 보낸다. 이것은 주기적으로 반복되며 그리드 사용자의 상태 변화를 통해 표현이 가능하다. 또한, Request Message는 지역 내에서 전달되는 메시지로써 지역 내 통신을 나타낸다. 로컬 자원 브로커(Local Resource Broker)는 그

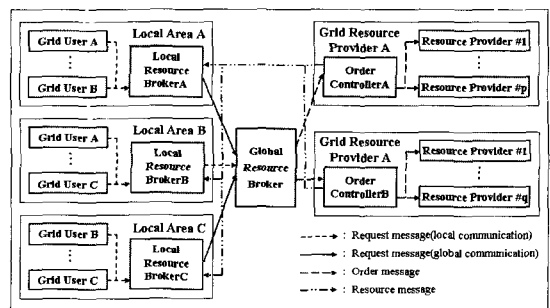


그림 2. 분산화된 브로커 기반의 시뮬레이션 모델 구성

리드 자원 사용자의 자원 요청을 조정하여 글로벌 자원 브로커에게 전달하는 컴포넌트로서 조정된 주문 정보가 저장된 Request Message를 글로벌 자원 브로커에게 전달한다. 이것은 그리드 사용자로부터 자원 사용이 요청될 때마다 각 지역의 외부에 존재하는 글로벌 자원 브로커와 통신하며 이것은 지역 외 통신을 나타낸다. 글로벌 자원 브로커(Global Resource Broker)는 로컬 브로커와 그리드 자원 공급자 사이에서 중개인 역할을 수행하는 컴포넌트로 사용자로부터 요청된 주문을 적절한 그리드 자원 공급자에게 전달한다. 이 컴포넌트에서는 주문을 받을 때마다 자원 모니터링과 선택을 기반으로 적합한 그리드 자원 공급자를 찾아 해당 공급자에게 주문을 전달한다. 또한, 주문 전달이 완료되면 다음 주문을 로컬 자원 브로커로부터 받을 때까지 기다리는 상태가 된다. 그리드 자원 공급자(Grid Resource Provider)는 글로벌 자원 브로커로부터 주문 정보를 받아 사용자에게 자원 사용 정보를 Resource Message에 저장하여 전달한다. 그리드 자원 공급자는 전략에 따라 구분된 그룹 별로 각각의 주문 컨트롤러(Order Controller)를 가지고 있으며 이들에 의해 각각의 공급자에게 주문 정보가 전달된다. 그리드 자원 공급자는 글로벌 자원 브로커로부터 주문이 전달될 때마다 해당 주문에 대한 처리를 하고 다음 주문을 기다리는 상태로 대기한다.

4.2 실험 결과 분석

4.2.1 실험 1: 거래 횟수의 변화율

모델 성능의 평가 척도 중 하나는 거래 횟수의 변화율이고 각 모델 내 거래 횟수의 변화율은 그림 3에서 주어진다. 그림 3에서 거래 횟수의 변화율은 1000 DEVS 시간마다 자원 사용자와 자원 공급자 사이에서 요청된 거래 당 처리된 거래를 나타내며 집중화된 브로커 기반 모델은 약 10%~12%의 변화율을 보이고 분산화된 브로커 기반

모델은 약 11%~13%의 변화율을 보인다. 이것은 두 모델이 단위 시간당 각각의 변화율만큼 거래 횟수가 증가한다는 것을 의미한다. 또한, 분산화된 브로커 기반 모델이 집중화된 브로커 기반 모델보다 단위 시간 당 거래 수의 변화율이 대체적으로 높다. 단위 시간 당 거래 횟수가 많은 것은 그리드 자원 사용자가 요청한 거래가 많이 성립되었다는 것을 의미한다. 이것은 거래 수의 변화율과 연관되며 분산화된 브로커 기반 모델이 구성원들 간 많은 거래의 기회 제공을 통해 보다 많은 자원의 활용을 가능하게 한다는 것을 보여준다. 우리는 단위 시간 당 거래 수의 변화율을 측정하기 위해 식 (2)를 사용한다. 식 (2)에서 TN (Trade Number)은 실험에서 단위 시간당 완료된 거래의 수를 나타내고 ct (current time)는 현재의 단위 시간을 나타내고 pt (previous time)는 이전 단위 시간을 나타낸다. ct 와 pt 의 간격은 단위 시간을 의미하며 실험에서는 1000으로 설정하였다.

거래 횟수의 변화율

$$= (TN_{ct} - TN_{pt}) \times 100(\%) \quad (2)$$

4.2.2 실험 2: 자원 활용도

세 번째 평가 척도는 자원 활용도이고 각 모델 내 자원 활용도는 그림 4에서 주어진다. 자원 활용도는 완료된 거래의 자원 활용도를 나타내고 집중화된 브로커 기반 모델은 약 39%~45%의 자원 활용도를 보이고 분산화된 브로커 기반 모델은 약 41%~52%의 자원 활용도를 보인다. 이것은 분산화된 브로커 기반 모델이 자원 활용도가 더 높다는 것을 의미한다. 또한, 분산화된 브로커 기반 모델이 집중화된 브로커 기반 모델보다 자원 활용도가 높다. 높은 자원 활용도는 자원 공급자가 동일한 시간 내에 더

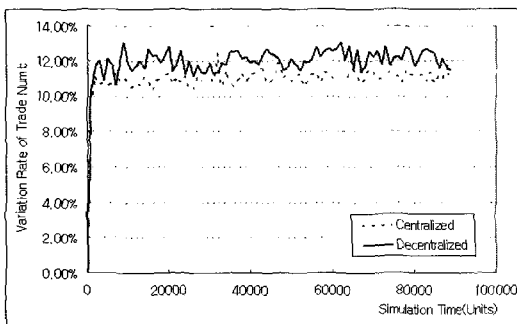


그림 3. 거래 횟수의 변화율

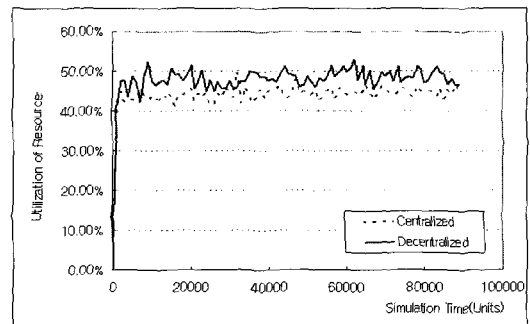


그림 4. 자원 활용도

많은 자원을 사용했다는 것을 의미하고 이것은 휴면 자원이 감소한다는 것을 나타낸다. 따라서 그리드 사용자들에게 많은 자원을 활용할 수 있게 하고 그에 따라 그리드 사용자의 대기 시간이나 지연 시간을 줄이는 것을 가능하게 한다. 우리는 실험에서 단위 시간당 자원 활용도를 측정하기 위해 식 (3)을 사용한다. 식 (3)에서 NPT (Number of Processed Trades)는 단위 시간 내에 완료된 총 거래의 수를 나타내고 NGT (Number of Generated Trades)는 단위 시간 내에 그리드 사용자로부터 발생된 총 거래의 수를 나타낸다. t 는 실험에서의 시간 단위를 나타내고 매 1000 시뮬레이션 시간으로 설정하였다.

$$\text{자원 활용도} = \frac{NPT_t}{NGT_t} \times 100(\%) \quad (3)$$

4.2.3 실험 3: 평균 처리 시간

세 번째 평가 척도는 평균 처리 시간이고 각 모델 내 평균 처리 시간은 그림 5에서 주어진다. 평균 처리 시간은 단위 시간당 완료된 거래의 평균 처리 시간을 나타내고 집중화된 브로커 기반 모델과 분산화된 브로커 기반 모델 모두 지속적으로 증가한다. 거래가 이루어지기 위해서는 이전 거래가 종료되기를 기다려야 하는데 그 때 발생하는 대기와 지연으로 인해 단위 시간당 평균 처리 시간은 지속적으로 증가한다. 그림 5의 평균 처리 시간은 하나의 거래가 완료될 때마다 해당 거래의 응답 시간을 나타내고 확실히 분산화된 브로커 기반 모델이 집중화된 브로커 기반 모델보다 빠르게 처리되는 것을 알 수 있다. 처리 시간은 거래가 완료된 시간과 요청된 시간을 통해 계산할 수 있고 낮은 평균 처리 시간은 거래의 요청부터 완료까지 거래의 지연이 많이 발생하지 않았다는 것을 의

미한다. 이것은 그리드 사용자의 거래에 대한 대기 시간과 지연 시간이 적다는 것을 의미하고 이것은 보다 빠른 거래의 체결을 통해 거래의 활성화를 제공한다는 것을 보여준다. 우리는 실험에서 평균 처리 시간을 측정하기 위해 식 (4)를 사용한다. 식 (4)에서 RT (Response Time)는 하나의 거래가 완료된 총 시간 즉, 소비자로부터 요청된 자원 사용에 대한 요구가 브로커를 통해 공급자에게 전달되어 공급자로부터 가용한 자원을 할당받고 비용을 지불하는 데 걸리는 총 시간을 의미한다. t 는 실험에서의 시간 단위를 나타내고 매 1000 시뮬레이션 시간으로 설정하였다.

$$\text{평균 처리 시간} = \text{AVG}(\sum RT_t) \quad (4)$$

5. 결 론

그리드 컴퓨팅 환경 하에서 자원은 지역적으로 분산되고 서로 다른 정책을 가진 다양한 자원 소유주가 이것들을 소유한다. 이러한 자원들의 공유와 활용을 위해 자원을 분배하고 관리하는 것은 매우 복잡한 문제이다. 따라서 이 논문에서 우리는 그리드 컴퓨팅 환경에서 자원 거래를 위해 집중화된 브로커 기반과 분산화된 브로커 기반의 자원 관리에 대해 언급했다. 그리고 GRACE 기반의 일반적인 자원 거래 구조에 분산된 구조를 적용한 분산화된 브로커 기반의 자원 관리 모델을 제안하였다.

본 논문의 목표는 그리드 환경에서의 자원 할당 및 관리 문제의 해결이다. 이를 위해 분산화된 브로커 기반 모델에서는 로컬 자원 브로커라는 지역별 브로커를 도입하고 시장경제의 원리를 통한 배분 방식을 적용한다. 이를 통해 자원 거래를 위한 지역 외 통신을 줄이고 거래 요청에 대한 응답 속도를 향상시킨다. 지역 외 통신의 감소는 사용자 거래 요청이 동일한 지역 내 통신에 의해 발생되더라도 더 많은 공급자에게 주문을 요청할 수 있도록 돕는다. 또한, 응답 속도의 향상은 사용자로부터 요청된 거래가 보다 빨리 완료될 수 있도록 돕는다. 이것은 일정한 시간 내 더 많은 거래가 일어날 수 있다는 것을 의미하므로 자원의 활용을 높이고 거래를 활성화시키는 효과를 제공한다. 실험 결과는 분산화된 브로커 기반 모델이 사용자의 거래 요청에 보다 빠르게 응답 및 처리가 가능하고 자원의 활용을 높인다는 것을 보여준다. 이것은 분산화된 브로커 기반 모델이 그리드 컴퓨팅 환경에서 자원을 할당하고 관리하는데 매우 효과적이라는 것을 증명한다.

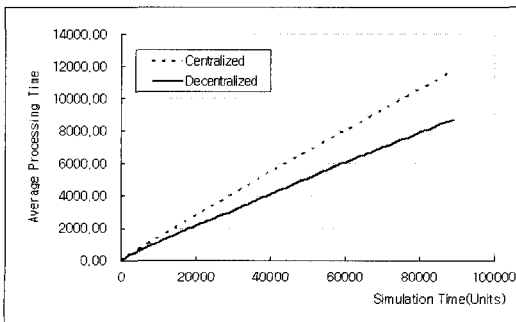
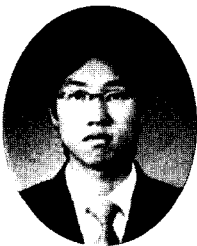


그림 5. 평균 처리 시간

참 고 문 헌

1. Arunachalam, R., Sadeh, N., Eriksson, J., Finne, N., and Janson(2004), The Supply Chain Management Game for the Trading Agent Competition 2004, CMU-CS-04-107, ISRI tech. report.
2. B.P. Zeigler, et al.(1996), DEVS Framework for Modeling, Simulation, Analysis and Design of Hybrid Systems in Hybrid II, Lecture Notes in CS, Vol. 3045 Springer-Verlag, Berlin, pp. 529-551.
3. B.P. Zeigler, et al.(1997), The DEVS Environment for High-Performance Modeling and Simulation, IEEE C S & E, Vol. 4, No. 3, pp. 61-71.
4. F. Berman, G. Fox and T. Hey(2003), Grid computing : making the global infrastructure a reality, J. Wiley, New York.
5. Ian Foster, Carl Kesselman(2003), The Grid2 : Blueprint for a New Computing Infrastructure, Morgan Kaufmann, November, pp. 259-265.
6. Moreno R. A.(2002), Job Scheduling and Resource Management Techniques in Dynamic Grid Environments, Lecture Notes in Computer Science Vol. 2970.
7. R. Buyya, D. Abramson, and J. Giddy(2000), An Economy Driven Resource Management Architecture for Global Computational Power Grids, The 7th International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2000), pp. 26-29.
8. R. Buyya(2002), A market Paradigm for Distributed Resource Management and Scheduling for Service Oriented Grid Computing, PhD Thesis.
9. R. Buyya, D. Abramson, and J. Giddy, "The Nimrod-G Grid Resource Broker Software", <http://www.csse.monash.edu.au/~davida/nimrod/>
10. R. Wolski, J. Plank, J. Brevik, and T. Bryan(2001), Analyzing Market-based Resource Allocation Strategies for the Computational Grid, International Journal of High performance Computing Applications, Sage Publications, Vol. 15, No. 3, pp. 258-281.
11. Sahai A. and Morin C.(1998), Towards Distributed and Dynamic Network Management, proceedings of the IEEE/IFIP Network Operation and Management Symposium, New Orleans, LA, USA.
12. W. Ketter, et al(2004), Analysis and Design of Supply-Driven Strategies in TAC SCM, Workshop on Trading Agent Design and Analysis(AAMAS'04), Columbia University, New York.
13. W. Ketter, et al(2004), MinneTAC sales strategies for supply chain TAC, In Proc. of the Third Int'l Conf. on Autonomous Agents and Multi-Agent Systems, page submitted, New York.
14. Yavnai A.(1994), Distributed Decentralized Architecture for Autonomous Cooperative Operation of Multiple Agent System", Proceedings of the IEEE 1994 Symposium on Autonomous Underwater Vehicle Technology, pp. 61-67.



마 용 범 (myb112@hanmail.net)

2005년 인하대학교 컴퓨터공학부 학사
 2007년 인하대학교 컴퓨터공학부 석사
 2007년~현재 인하대학교 컴퓨터공학부 박사과정

관심분야 : 그리드 컴퓨팅, 모델링&시뮬레이션



이 종 식 (jslee@inha.ac.kr)

1993년 인하대학교 전자공학과 학사
 1995년 인하대학교 전자공학과 석사
 2001년 애리조나대 컴퓨터공학과 박사
 2001년~2002년 캘리포니아 주립대학교 전기·컴퓨터공학과 전임강사
 2002년~2003년 클리블랜드 주립대학교 전기·컴퓨터공학과 조교수
 2003년~2006년 인하대학교 컴퓨터공학부 조교수
 2006년~현재 인하대학교 컴퓨터공학부 부교수

관심분야 : 소프트웨어공학, 시스템 모델링&시뮬레이션, 그리드 컴퓨팅