

자원 가용성 기반 다중 경매 모델을 이용한 서비스 예약형 클라우드 자원 거래 시스템

이석우^{1†} · 김태영¹ · 이종식¹

Resource Availability-based Multi Auction Model for Cloud Service Reservation and Resource Brokering System

Seok Woo Lee · Tae Young Kim · Jong Sik Lee

ABSTRACT

A cloud computing is one of a parallel and distributed computing. The cloud computing provides some service for user with virtual resources. However, a user's service request does not show a time pattern. As a result, each resource also shows a different availability at the same time. This difference affects a quality of service (QoS) and a resource selection for users. Therefore, we propose the resource availability-based multi auction model for cloud service reservation and resource brokering system. The proposed system is to select the proper resource provider based on the users' request. The proposal adopts the multi phase of the auction to transact resources. The system evaluates the available factor of each resource on the auction phase, and finally reserves the service on the adaptive queue. The proposed model shows the better performance than other existing method.

Key words : Cloud Computing, Virtualized resources, Resource Trading Model, Multi-Auction Model

요약

클라우드 컴퓨팅은 분산 병렬 컴퓨팅의 일종으로 IT 자원을 가상화 하고 이를 사용자에게 제공한다. 그러나 사용자의 서비스 요청은 시간적 규칙성이 없으며, 이런 이유로 각 자원들은 가용성의 차이를 갖는다. 가용성의 차이는 클라우드 서비스 이용자의 QoS만족도 및 서비스 제공자 선택에 영향을 주게 된다. 본 논문에서는 클라우드 환경에서의 서비스 이용자의 요구에 따라 가상화된 IT자원 제공자의 효율적 선정을 위한 자원 가용성 기반 다중 경매 모델을 이용한 서비스 예약형 클라우드 자원 거래 시스템을 제안한다. 제안한 시스템에서는 서비스 이용자의 서비스 요청 발생 시 두 번의 경매 모델을 사용하는 다중경매 모델이며, 다중 경매 과정 중 단일 경매 단계에서 자원 가용성 평가 알고리즘 및 가변 큐를 이용한 서비스 예약방식을 적용하여 서비스 제공자의 자원 가용성을 평가하여 최적의 서비스 제공 자원을 찾는다. 제안모델은 QoS만족도 부분에서 앞서 작용한 자원 가용성 평가 알고리즘으로 인해 높은 성능을 보여주며, 작업 처리 시간활용도 면에서 QoS를 모두 고려한 상태로 안정적으로 더 많은 작업을 처리 할 수 있음을 입증하였다.

주요어 : 클라우드 컴퓨팅, 가상자원, 자원 거래모델, 다중 경매모델

1. 서론

클라우드 컴퓨팅은 IT자원을 사용자들에게 비용을 받고 제공하는 서비스로 인터넷 기술을 기반으로 가상화된 데이터 센터의 자원을 다수의 사용자들에게 다양한 서비스 형태로 공급하는 기술이며, 가상화 및 자동화, 백업, 스냅 샷(모니터링 및 기록 관리 등), 컴포넌트가 모두 제공되어야 클라우드 컴퓨팅이라 이야기할 수 있다(B. Hayes,

*이 논문은 인하대학교의 지원에 의하여 연구되었음.
접수일(2013년 3월 4일), 심사일(2013년 9월 2일),
게재 확정일(2013년 9월 17일)

¹⁾ 인하대학교 정보공학과

주 저 자 : 이석우

교신저자 : 이종식

E-mail; jslee@inha.ac.kr

2008; P. Mell, 2009). 클라우드 컴퓨팅 서비스는 통신기술을 매체로 언젠, 어디서나 서비스 이용자는 단말기만 있다면 서비스의 구매 및 이용이 가능한 기술이다. 또한 사용자의 요구에 따라 인프라(IaaS) 및 플랫폼(PaaS), 소프트웨어의 확장(SaaS)이 가능하며, 유연하게 적용이 가능하기 때문에 비용적인 측면에서 가격 절감 효과가 크다.

현재 국내에선 클라우드 컴퓨팅에 대한 다양한 연구가 진행되고 있으며, 컴퓨팅 기술의 흐름은 비즈니스(Business Driven Computing)를 중심으로 하는 서비스 클라우드 컴퓨팅 환경으로 변화 하고 있다.

클라우드 컴퓨팅 환경에서 가상화된 자원을 다수 보유한 서비스 제공자는 사용자의 서비스 이용 요청이 들어올 경우 자원 관리자를 이용하여 최적의 서비스 제공 자원을 선택 및 할당하게 되며, 또한 서비스 이용자는 서비스 준수 협약(SLA: Service Level Agreement)에 명시된 기준 내에서 서비스 제공자로 부터 자원 이용 서비스를 보장 받게 되며, 클라우드 환경 내에 저장 및 관리되고 있는 가상화된 자원들을 이용하여 자신에게 필요한 계산을 수행할 수 있다. 하지만 자원을 이용하기 위한 과정에서 서비스 제공자와 서비스 이용자 사이에는 자원 거래로 인한 비용적인 문제가 발생하게 된다. 서비스 이용 시 발생하는 비용적 문제는 서비스 이용자 측면에서 서비스 제공자 선택의 기준으로 작용하게 되며, 비용적 부분 외에도 서비스 이용자의 서비스 요청 시 서비스 제공자의 자원 가용성 타이밍에 따른 부분도 서비스 제공자 선택에 영향을 줄 수 있다. 이처럼 자원 거래 시 발생하는 비용적 문제 및 자원 가용성 문제를 포함한 최적의 서비스 제공자를 탐색하여 매칭 시키는 연구가 다양한 형태로 진행되어져 왔다. 이는 기존 분산 컴퓨팅 자원 거래와 비슷한 형태를 갖고 있으며, 기존 분산 컴퓨팅 자원거래 방식과 비슷한 관점에서 접근이 가능하다. 가장 대표적인 선행 연구로는 경매모델(U. Kant et al., 2005)을 이용한 자원 거래 시스템이 있으며, 세부적으로 단일 경매모델을 이용한 자원 거래 와 이중 경매 모델을 이용한 자원거래 모델이 존재한다(R. Buyya, 2002). 또한 국내에는 무선 접속 장치를 이용한 가상 무선 네트워크 제공 시스템 및 방법에 대하여 경매 모델을 이용하는 선행 시스템이 제안되어졌다(Lee et al., 2010). 그 외에도 GRIDS (Grid Computing and Distributed System)Lab에서 제안하는 시스템은 최근 급증하는 클라우드 시스템의 서비스 제공 방식에 대하여 경매 모델을 제안하였지만 이는 QoS 평가를 지원하지 않는 단점이 있었다(X Chu et al., 2007). 앞서 시행되어진 연구에 사용된 경매 모델들은 현 시점에서 클라우드

컴퓨팅내의 QoS 평가를 지원하지 않으며, 이를 보완할 필요가 있으며, 이를 보완하기 위하여 QoS 평가가 가능한 자원 거래 모델을 제안한다. 제안 모델은 경매 모델을 이용하여 자원을 거래하며 중간에 QoS평가 모듈을 배치하여 가용성을 평가하며, 가변 큐를 배치하여 서비스 제공률을 상승시킨다.

본 논문의 구성은 2장에서는 관련연구, 3장에서는 제안모델인 자원 가용성 기반 다중 옥션 모델을 이용한 클라우드 자원 거래 시스템에 대하여 기술한다. 4장에서는 본 논문에 대한 실험 및 평가를 실시하며, 5장에서 결론을 논의한다.

2. 관련연구

2.1 시장 경제 원리를 반영한 경매모델

경매 모델에는 단일 경매 모델, 이중 경매 모델이 있으며(R. Buyya, 2002), 이중 경매 모델은 서비스 이용자와 서비스 제공자 양측을 모두를 만족시킬 수 있는 거래 모델로서 상호 입찰방식을 사용하는 특징을 갖고 있다(L. Joita, 2004). 일반적으로 서비스 이용자는 낮은 가격으로 자원을 거래 받길 원하며, 서비스 제공자는 높은 가격으로 자원이 거래되길 바란다. 이런 상황에서 서비스 이용자와 서비스 제공자는 자신이 보유한 가격 탄력성을 바탕으로 서비스 이용자는 가격을 높여서 입찰하고, 서비스 제공자는 가격을 낮춰서 새로운 가격을 갱신 및 반복을 통하여 서로의 가격이 시장 균형에 맞을 경우 거래가 성사된다. 이것은 현실 시장 거래 원리와 가장 흡사한 형태를 갖는다.

본 논문에서는 세 번째 거래 단계에서 이중 경매 모델의 상호 입찰 방식을 도입하여 서비스 제공자와 서비스 이용자 사이에 가격협정이 가능할 수 있도록 구성한다.

2.2 클라우드 컴퓨팅 환경에서의 자원거래 모델

클라우드 컴퓨팅 환경에서의 자원거래의 특징은 서로 다른 지역에 존재하는 물리자원에 가상화 기술을 적용하여 협업할 수 있도록 하는 기술이다. 하지만 이는 서비스 종속된 벤더 내에서만 사용이 가능하며, 서비스 제공자들의 협업을 허락하지 않고 있다. 또한 QoS에 대한 평가역시 지원되지 않는 것이 현실이다(Han et al., 2009). 위와 같은 환경에서 거래 모델들은 서비스 제공자와 이용자 사이에 자원을 단순하게 거래하는 것을 벗어나 서비스 이용자의 자원 요구사항을 파악하고 이를 정확하게 선별하여 제공할 수 있는 시스템을 필요로 하며, 이를 해결하기 위

해 클라우드 마켓 컴퓨팅을 위한 리소스 추천 시스템(Han et al., 2009) 및 클라우드 환경에서의 자원거래를 위한 경매 모델(Lee et al., 2010) 등이 존재한다. 하지만 클라우드 환경에서의 경매 모델은 QoS 평가 기능을 지원하지 않으며 이를 지원하는 모델의 연구가 필요하다(Han et al., 2009).

2.3 Amazon Web Service – EC2

Amazon의 클라우드 컴퓨팅 서비스는 1995년 분산 컴퓨팅 시스템에서 발달했다. 이것은 AWS (Amazon Web Service)로 칭해졌으며, 현 아마존 클라우드 컴퓨팅 시스템의 플랫폼을 구성하는 초석이 되었다. 현 시점의 아마존의 클라우드 컴퓨팅 시스템은 대표적으로 EC2(가상화된 컴퓨팅 자원) 서비스와 S3(대용량 스토리지 서비스) 서비스로 나뉘지며, 본 논문의 관련 연구에서는 서비스 이용자에게 가상 자원으로서 할당되며, 사용자가 비용을 지불하여 사용하는 거래 모델인 EC2에 대해서만 다루도록 한다.

EC2 모델은 자신이 사용한 시간만큼의 서비스 비용을 지불한다. 서비스 제공자와 이용자 사이에 가격협상이 단절된 일반적인 자원 보유자 중심의 거래 방식을 사용하고 있으며, 이런 이유로 양측 가격 협의가 불가능한 단점을 보유하고 있다. 본 논문에서는 EC2 모델을 비교 모델로 활용한다.

2.4 사용자 요구기반의 그리드 거래 관리 모델

사용자 요구기반의 그리드 거래 관리 모델(User Demand-based Grid Trade Management Model) (Ma, 2009)은 그리드 컴퓨팅 환경에서의 자원거래 모델로서, 많은 양의 상품을 주문하는 상황이 발생할 경우 서비스 제공자는 판매가 보다 낮은 가격을 제시하며, 반대로 적은 양의 상품 주문 상황이 발생할 경우 상품의 비용을 증가시키는 상업 시장의 거래 원리를 적용하였다. 실제로 시스템에서는 제품의 서비스 이용자의 자원거래 요구량이 복수일 경우 거래량을 이용한 거래 전략을 통하여 낮은 비용으로 서비스 제공자와 가격 협의를 실시하며, 반대로 요구 자원이 단일인 경우 자원의 가격을 높게 책정하여 경매를 실시하는 방식으로 구성되어 있으며, 그리드 브로커란 시스템을 통하여 서비스 이용자와 서비스 제공자 사이에서 서비스 이용자의 거래 요청 성향에 따라 공급자를 선정함으로써 비용적 측면과 요구량 측면을 고려하도록 설계되어 있다. 해당 모델은 비용적 측면을 제외한 어떤 부분도 고려하지 않는 단점을 보여주고 있다.

본 논문에서는 사용자 요구기반의 그리드 거래 관리 모델을 이용하여 제안모델과 성능비교를 실시한다.

2.5 자원관리

클라우드 서비스를 제공하기 위해서는 인프라 성능 및 제공자의 수익을 예상하여야 하며, 이를 바탕으로 설계를 해야 하며, 이를 통하여 이익을 향상시킬 수 있어야 한다 (Ma et al., 2011). 또한 클라우드 서비스는 자원 요청에 따른 서비스 이용자의 요구 발생 서비스 제공에 따른 부하가 가장 적은 자원이 선택되어야 하며, 데이터 센터는 다수의 가상머신으로 구성이 가능하다(Mousumi et al., 2011).

본 논문에서는 가상 자원에 거래를 제안하는 Auctioneer 모듈에 작업을 각각에 큐로 분산하고 초과되는 작업에 대해서는 가변 큐 기법을 이용하여 서비스 제공에 따른 실패를 저하 메커니즘을 제시하려한다.

3. 자원 가용성 기반 다중 경매 모델을 이용한 클라우드 자원 거래 시스템

3장에서는 시장 지향형 모델을 바탕으로 클라우드 환경에서 활용 가능한 자원 가용성 기반 다중 경매 모델을 이용한 클라우드 자원 거래 시스템을 자세히 설명한다.

서비스 이용자는 Web Portal을 이용하여 최초 서비스 시작을 실시하며, 시스템 구동방식은 Fig. 1과 같다.

3.1 경쟁 입찰을 통한 자원 후보군 선정(Single Auction)

자원 이용을 희망하는 서비스 이용자는 자신의 요구사항을 자원 중개자에게 전달하게 된다. 이때 전달하는 서

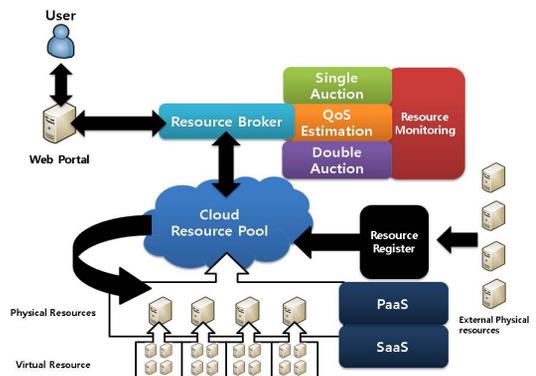


Fig. 1. System Architecture

Table 1. Properties of static information for transfer to service provider

Division	Property name	Description
Service users	St	Desired service start time
	Sd	Desired service deadline
	Si	Service type
	Sf	Desired service costs
	Lf	Service cost limit
	Hrp	desired rate for using service
Service Providers	VM ID	Virtual machine identifier
	Ps	Types of services available
	Et	Estimated time to perform job
	Ed	Estimated time to job termination
	Sci	Services ratio
	Rp	Rate services

Table 2. Cost of service provider's delivery service

Type of jobs	price
1. mail / message	22 ~ 135/h
2. Streaming service	180 ~ 500/h
3. Scientific computing services	1000 ~ 3500/h

Table 3. PCE allowance

Task (N)	Origin Price	PCE (%)
1	22 ~ 135/H	20 ~ 80
2	180 ~ 500/H	30 ~ 80
3	1000 ~ 3500/H	30 ~ 80

비스 중개자에게 전달되는 정적 정보 속성은 Table 1의 내용을 참조한다.

자원 중개자는 1단계서 받은 서비스 이용자의 정보 중 희망 서비스 이용 비용부분을 참조하며, 경매에 참여한 서비스 제공자들은 서비스 이용자의 최대 가격 값을 바탕으로 자신의 기대 이윤이 일정 수치 이상일 경우 경매에 참여하며, *ServiceUserMaxPrice*는 식 (1)과 같다.

$$ServiceUserMaxPrice = Task(N) \times SimulationTime \times PCE(\%) \quad (1)$$

식 (1)의 Task (N)은 임의로 할당받는 요청 작업의 시간당 이용가능 최대 비용 값이며, Simulation Time은 서비스 이용 시간, PCE (Preferred cost elasticity)는 희망 비용 탄력성을 의미한다. Table 3은 희망 비용 탄력성 (PCE)에 대한 허용범위 정보이다. PCE는 요청 작업에 따라 20~80%범위를 갖게 되며, PCE의 값이 최소 20%, 30% 밑으로 이루어질 경우 서비스 제공자의 마진을 보장할 수 없기 때문에 제공자의 비용 효율이 떨어지게 된다. 또한 80% 이상으로 범위가 넘어갈 경우 서비스 이용자의 비용효율 감소로 이어지기 때문에 Table 3과 같은 범위를 책정한다. 다음은 서비스 이용자의 최대 보유금액 유도방식이다. 예를 들어 Task (3)의 최대 서비스 이용 가격이 3500일 경우 Task (3)을 이용하는 서비스 이용자의 최대 보유 금액은 3500에 대한 30~80%범위의 가격을 서비스 이용 최대 희망 금액으로 보유하게 된다.

$$EM = Task(N) \times SimulationTime \times PCE(\%) \quad (2)$$

다음은 서비스 제공자의 기대 이익이다. 서비스 제공자의 기대 이익은 식 (2)와 같다. 거래 기대 마진(Expected Margin, EM)은 서비스 제공자의 최소 거래 희망 가격이다. 서비스 제공자는 각각의 Task (N) 별로 20%~30%의 고정적인 기대 마진을 갖는다. 고정적 거래 기대 마진 (EM) 값 보다 서비스 이용자의 최소 희망 금액이 같거나 클 경우 자원 후보군에 속하게 되며, 다음 단계인 자원 가용성 평가 알고리즘의 적용을 받게 된다.

3.2 자원 가용성 기반 자원 평가 알고리즘

이 단계에서는 서비스 이용자의 요구사항 정보 및 서비스 제공자의 가상머신 가용성 상태를 기반으로 평가를 실시한다. 자원 가용성 평가 알고리즘은 Fig. 2와 같다. 최적 자원 선정 알고리즘은 자원 중개자로 부터 전달받은 서비스 이용자의 희망 서비스 시작시간(St), 희망 서비스 마감 시간(Sd), 서비스 유형 구분 정보(Si), 희망 서비스 제공률(Hrp) 값과 서비스 제공자로부터 전달 받는 제공가능 서비스 유형(Ps), 작업 수행 예상시간(Et), 작업 종료 예상시간(Ed), 서비스 제공률(Rp) 값을 이용하여 알고리즘에 대입시킨다. 만약 복수개의 가상 자원이 선정될 경우 가상 자원 수만큼 알고리즘의 반복을 실시한다.

다음 Table 4는 서비스 이용자가 포함하게 되는 희망 서비스 시작 시간과 희망 서비스 마감 시간의 허용범위이다.

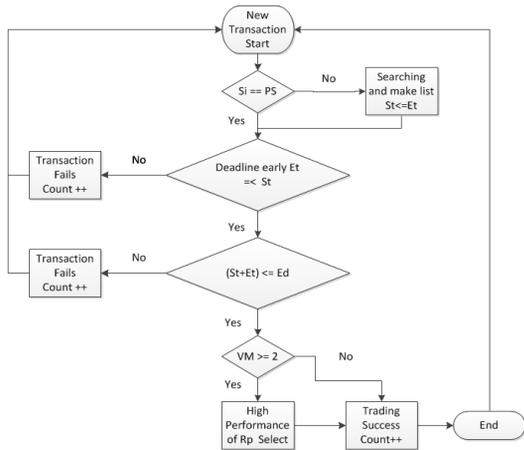


Fig. 2. Resource availability evaluation algorithm

Table 4. Deadline allowance of service user

Deadline early Et	Division	Deadline rate ED
-t	St	+t
-t	Sd55	+t

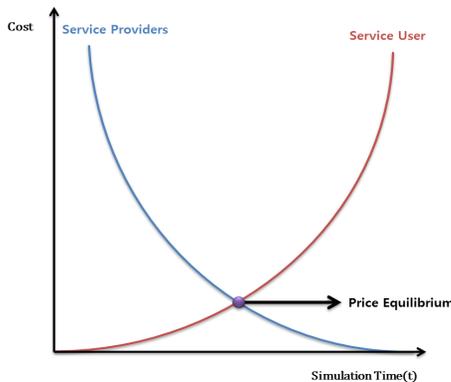


Fig. 3. Agreed price graph of dual auction model

서비스 제공자의 작업수행가능 시간과 마감 시간이 위 값의 범위를 벗어날 경우 해당 후보군은 마감 시간을 지킬 수 없다 판단하여 후보군에서 제외시킨다. Table 4의 Deadline early는 서비스 시작시간(St)의 경우 서비스 이용자의 희망 시작 시간보다 요청 작업이 빠르게 시작될 것을 의미하며, Deadline late는 서비스 이용자의 요청 작업이 희망 시작 시간보다 작업이 늦게 시작되는 경우를 의미한다. 또한 서비스 마감 시간(Sd)에 적용될 경우 Deadline early는 서비스 이용자의 희망 종료 시간보다 빠르게 끝날 것을 의미하며, Deadline late 경우 서비스

이용자의 희망 종료 시간보다 늦게 끝날 것을 의미한다.

네 번째 단계에서는 서비스 자원 후보군들에 대하여 서비스 제공률을 바탕으로 판단한다. 비슷한 스펙의 자원들이 복수개로 후보에 있을 경우에 서비스 제공률이 높은 자원을 선택하게 된다. 서비스 이용자의 희망 서비스 제공률 허용 범위는(Hrp - 15Mbyte)/Mbyte까지 허용한다. 오버된 제공률은 서비스 이용자에게 이득이 되므로 고려하지 않는다. 서술한 내용과 같이 자원 가용성 평가 알고리즘은 사용자의 요구조건과 서비스 제공자의 조건을 정의된 자원 가용성 평가 알고리즘에 적용하여 기준을 충족시키지 못하는 자원을 제거해나가는 방법으로 가장 최상의 조건을 가진 자원을 선택할 수 있는 수단을 제공한다.

3.3 이중거래 조건 협상 방식(Double Auction)

Fig. 3는 이중 경매 모델의 가격 합의점 도출을 나타낸 그래프이다. 서비스 제공자와 서비스 이용자는 Simulation Time 동안 서로간의 가격 합의점 도출을 위해 20번의 제안 횟수 내에 가격 탄력성을 바탕으로 서로간의 가격을 조절하여 가격 합의점을 찾아가는 단계이다. 지나치게 긴 가격 협정 시간은 시스템의 성능 저하를 초래할 수 있기 때문에 요청 작업별 가격 협정 제한 횟수는 20번의 제한을 둔다. 서비스 이용자는 저렴하게 이용하는 것을 목적으로 하기 때문에 최대 단가의 20% 가격을 최초 입찰가로 설정하고 이후 실시되는 재입찰에서는 각각의 가격 탄력성에 맞추어 입찰을 실시한다. 또한 서비스 이용자의 20% 경매 입찰 가격은 서비스 제공자의 최소 마진 보장을 위해서 20%로 고정한다. 서비스 제공자의 가격 탄력성을 이용한 입찰 식은 식 (3)과 같다.

$$\begin{aligned} \text{Service Providers Bidprice} = \\ \text{Previous Bidding Price Of SP} \\ - \text{Random}(PE \text{ of SP}) \end{aligned} \quad (3)$$

식 (3)의 SP는 서비스 제공자(Service Providers)를 의미하며, Random (PEofSP)는 서비스 제공자의 임의의 가격 탄력성(Price Elasticity of Service Providers; PEOfSP)을 의미한다. 탄력성의 범위는 Table 5의 서비스 이용자의 탄력성과 동일하다. 가격 탄력성(Price Elasticity; PE)는 Table 3의 Origin Price의 서비스 제공 최소값에 대하여 Task (1)은 50%, Task (2)는 30%, Task (3)은 20%범위를 갖게 되며 범위 책정 근거는 각 범위 이상 상의 범위를 책정할 경우 서비스 이용자의 비용적 효율이 저하되기 때문에 위와 같이 설정한다.

Table 5. The bidding price that includes price elasticity of service user

Task (N)	Service User PE
Task (1)	$Task (1)MinPrice \times PCE + PE$
Task (2)	$Task (2)MinPrice \times PCE + PE$
Task (3)	$Task (3)MinPrice \times PCE + PE$

다음은 가격 탄력성 범위(PE)의 설정 방법을 이야기한다. 임의의 수를 출력하는 함수를 이용하여 0~1까지의 double type의 값을 리턴 받고 서비스 이용자와 서비스 제공자의 최대 가격 탄력성을 곱 연산을 실시하여 정수형의 가격 탄력성 범위를 생성한다. 서비스 이용자는 서비스 이용 최소 금액에 임의의 PCE (20~80%) 범위를 곱 연산하여 얻은 금액에 대하여 가격 탄력성(PE)를 합 연산하여 가격을 산출하며, 최종적으로 Table 5와 같다. Table 5의 PE는 가격 탄력성을 의미하며, Table 3의 작업 종류에 따라 작업 1은 0~10%, 작업 2는 0~50%, 작업 3은 0~100%의 가격 탄력성을 갖는다. 작업 1은 비용의 범위가 좁기 때문에 큰 가격 탄력성을 보유할 경우 가격 조절의 의미가 없기 때문에 위와 같은 가격에 따른 탄력성의 범위를 갖는다.

다음은 최종 합의된 입찰가 도출 방식이다. 서비스 이용자와 서비스 제공자의 입찰가가 합의 될 경우 최종 금액을 산출하며, 서비스 이용자는 가상머신을 할당받게 된다. 이때 지불하게 되는 최종 금액은 식 (4)와 같다.

$$ResultPricie = \frac{(LastBiddingPrice\ Of\ SU + LastBiddingPrice\ Of\ SP)}{2} \tag{4}$$

이와 같이 앞서 자원 가용성 평가 알고리즘의 자원 선택을 통해 선정된 제공자와 클라우드 사용자가 조건을 맞춰나가기 위해 협상하는 과정을 추가하여 양측의 만족스러운 서비스 제공 및 이용을 기대할 수 있다.

3.4 가변 큐를 이용한 우선순위의 서비스 예약 방식의 자원 탐색

본 제안 모델은 서비스 요구자의 희망 서비스 시작 시간(St) 및 희망 마감시간(Et)에 의한 서비스 제공물 저하를 방지하기 위해서 Auctioneer내에 Fig. 4와 같은 큐를 이용한 작업 분산 방식을 도입한다.

해당 모듈은 서비스 이용자의 희망이용 시간이 빠른 Task를 중심으로 각 Task의 St값을 기준으로 서비스 이용 시간을 각각의 큐를 바탕으로 구분 짓고 특정 큐에 작

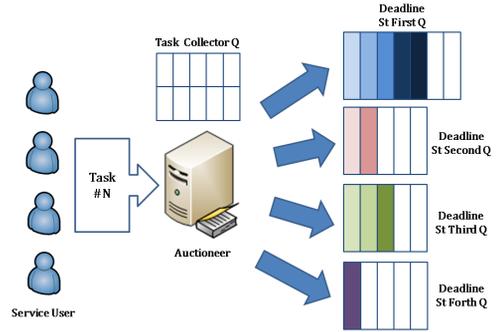


Fig. 4. Sorting jobs using Q

Table 6. Q Value Range

Q name	Q Value Range		
First Q	0	<=	a
Second Q	a+1	<=	β
Third Q	β+1	<=	γ
Forth Q	γ+1	<=	δ

업이 집중 될 경우 해당 큐의 길이를 가변적으로 조정할 수 있도록 한다. 또한 특정 St값을 요구하는 큐가 가변적으로 증가할 경우 Auctioneer는 요구 QoS를 의미하는 Si 값 다음으로 서비스 제공자의 Et를 우선적으로 자원을 탐색 및 제공하게 된다. Auctioneer 내부에 존재하는 큐는 Fig. 4와 같다. Fig. 4는 큐를 활용한 태스크 정렬 개념으로서, Fig. 4에서 Deadline St first의 우선순위가 가장 높으며, 차례로 Second, Third, Forth 우선순위를 할당 받는다. Auctioneer는 작업을 전송 받으면 Task Collector Q에 작업을 전송하며, 해당 Q의 #1~#n 까지 작업을 탐색하여, 각 작업의 St값에 따라 우선순위가 각 다른 큐로 몇 개의 작업이 이동하게 될지 파악하며, 어떤 큐에 작업이 집중적으로 할당될지 합산하며, Auctioneer는 작업이 집중되는 Q의 길이를 가변적으로 증가시킨다. Task Collector Q에 25개의 작업이 전송되고 St first Q로 이동되는 작업이 총 7개일 경우 2개의 Task가 초과되기 때문에 2개의 큐를 증가시킨다. 이렇게 큐를 가변적으로 함으로써, 우선순위가 높은 작업들을 집중적으로 처리 할 수 있는 장점이 있으며, 이는 서비스 이용자의 요구인 Deadline St에 따른 거래 실패율을 저하시킬 수 있다. Task를 각 큐에 배치하기 위한 조건은 Table 6과 같다.

Fig. 5는 Task Collector Q를 구체화 한 것으로 큐를 가변적으로 증가시키는 알고리즘을 포함하고 있다. 본 알고리즘은 작업(Task)을 서비스 이용자의 요구 Deadline

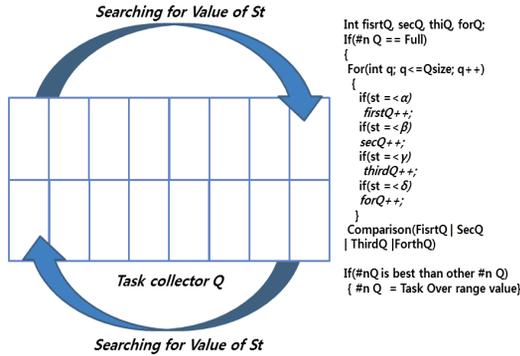


Fig. 5. Job checking algorithm using St value

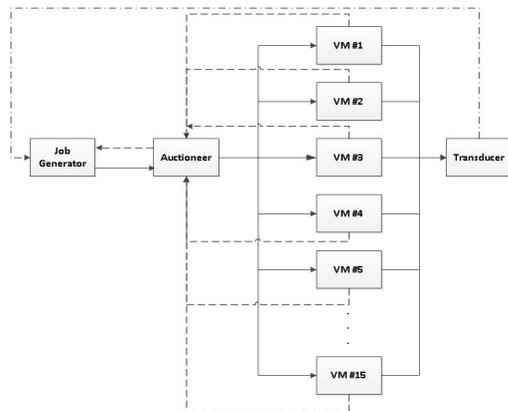


Fig. 6. Cloud test environment

St 값에 따라 Task Collector Q를 통해 제한된 숫자의 Task들을 전송하며, Task Collector Q가 가득 찰 경우 탐색을 통해서 St 값을 파악하고 분류하는 작업을 실시하여 각각의 큐로 전송한다. 또한 이렇게 분류된 Task들은 각각의 우선순위에 따라 각각의 큐에 저장되며, 특정 큐에 집중되는 초과 작업에 따른 가변 큐의 증가량은 Deadline St #n Q (Q의 이름)에 초과되는 작업의 수와 같다. 각 큐들의 작업들은 St 값이 작은 작업부터 경매 단계로 진입하게 된다. 이때 각 작업의 처리 시간이 St (시작시간)+Pt (처리시간)이 서비스 제공자의 Et (종료시간)보다 클 경우 해당 작업은 서비스 이용자의 희망 처리 시간을 맞출 수 없는 것으로 판단하여, Loss Task가 된다.

4. 실험

본 논문에서 제안하는 RMAM (Resource availability-based Multi Auction Model)의 효과를 입증하기 위해서

DEVS 형식론(B. Zeigler, 1997)을 적용하여, 가상화된 클라우드 환경을 구성하며, 성능을 측정하기 위해 거래 성공률, QoS 만족도 및 거래 당 처리 시간 활용도를 측정한다. 비교 모델로는 아마존 EC2기반의 고정비용 체계를 사용하는 거래 모델 AWS Single Auction Model (Hesam Izakian, 2010), 가용성 평가 없이 단순히 비용기반 경매만을 사용하는 모델 CDA (Continuous double Auction Model)(Hesam Izakian, 2010) 및 사용자 요구사항에 따라 자원 거래자를 선정하는 UDGTM (User Demand-based Grid Trade Management Model) (Ma, 2006)을 비교 모델로 사용한다.

4.1 테스트 환경 구성

본 논문에서 제안하는 RMAM의 성능 측정을 위해서 DEVS 형식론 기반으로 Fig. 6과 같은 실험 테스트 환경을 모델링 한다. Job Generator와 Auctioneer와 n개의 가상머신들 그리고 transducer로 구성되며, 총 15개의 가상자원으로 구성된다. 실험 시나리오를 구성하기 위해서 첫 번째 아톰릭 모델인 Job Generator는 작업을 생성하고 작업을 보내는 모듈이며, Auctioneer는 자원 중개자를 모델링한 것이며, 내부에는 최적 거래자원 탐색 시스템인 자원 가용성 평가 알고리즘 및 가변 큐를 활용한 작업 정렬 기능을 포함하고 있다. VM #n은 자원 풀 내에 존재하는 가상 자원들을 모델링한 것이다.

본 모델의 구동방식은 3의 자원 가용성 기반 다중 경매 모델을 이용한 클라우드 자원 거래 시스템에서 설명한 내용과 동일하다.

4.2 서비스 제공 모델별 QoS 만족도

두 번째 실험에서는 비교 모델들과 제안 모델의 QoS 만족도 비교를 통하여 성능을 평가해보도록 한다. Fig. 7은 비교 모델들과 제안모델의 QoS 만족도를 거래량에 따라 그래프로 표시한 것이다. QoS 만족도는 Deadline준수를 포함한 사용자의 요구사항을 얼마나 만족시키며 거래를 성사시켰느냐를 나타낸 것으로서, 서비스 요구 1만 건에 대하여 내부적으로 QoS 만족도 요구사항에 해당되는 가격, 희망 작업 시작 시간, 희망 작업 종료시간 요소가 포함되며 이것을 지켰을 경우 QoS 만족도를 만족시킨 것으로 판단하며 별도로 카운트하여 식 (5)를 이용하여 계산한다. 본 실험 부분에서는 거래 실패로 인해 발생하는 정보는 제외하였다. 거래 실패요인은 서비스를 받지 못한 것으로서 QoS 만족도를 판단 할 수 없기 때문이다. 본 실험에서 제안 모델은 비교 모델들 QoS만족도 면에서 높은

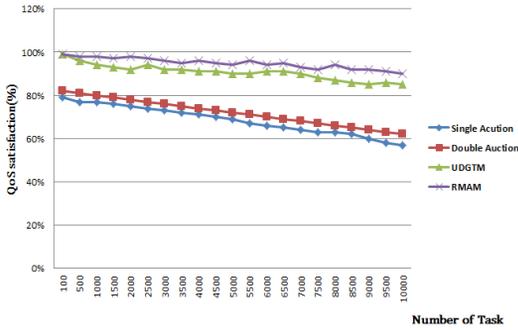


Fig. 7. The QoS satisfaction over time(UDGTM : User Demand-based Grid Trade Management Model; RMAM : Resource availability-based Multi Auction Model)

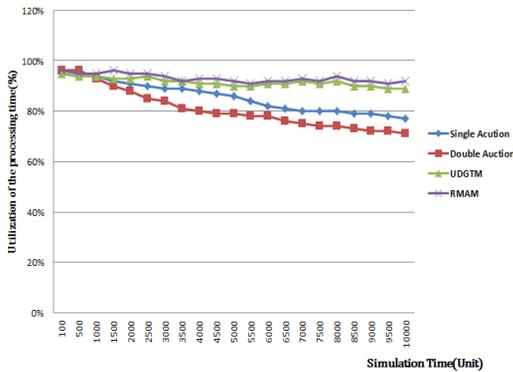


Fig. 8. Utilization of the processing time(UDGTM : User Demand-based Grid Trade Management Model; RMAM : Resource availability-based Multi Auction Model)

성능을 보여주고 있다. 이것은 내장되어있는 자원 가용성 평가 알고리즘(최적 자원 선정 알고리즘)에 의한 결과로서 서비스 이용자의 요구사항에만 맞춰 자원 제공자를 선정하기 때문에 나타난 결과이다.

$$QoS\ Satisfaction = \frac{Number\ Of\ Trading\ Success\ (T)}{Run\ From\ The\ Task\ Of\ Optimal\ Resource\ (J)} (\%) \quad (5)$$

4.3 각 거래별 처리 시간의 활용도

Fig. 8은 각 거래별 처리시간의 활용도를 시간에 따라 나타낸 그래프로서, 비교 모델 및 제안모델들의 성능 비교를 보여주고 있다. 그래프에 대하여 제안 모델은 비교 모델들의 비하여 우수한 성능을 보여주고 있다. 실험의

측정을 위하여 식 (6) (Ma, 2006)을 사용한다.

$$Utilization\ of\ Preocessing\ Time = AVG\left(\frac{\sum Execute\ Time_i}{\sum (Finished\ Time_i - Requested\ Time_i)}\right) \quad (6)$$

식 (6)의 *ExecuteTime*은 실험에서의 각 작업 처리에 따른 처리 시간(*SimulationTime*)을 나타내며, *FinishedTime*은 거래가 종료된 시간을 나타낸다. 또한 *RequestedTime*은 서비스 이용자의 거래 요청 시간을 나타낸다.

Fig. 8을 통하여 살펴볼 때 단일 경매 모델과 이중 경매 모델은 거래 초반에 높은 활용도를 보여주고 있다. 하지만 시간의 변화에 따라 지속적인 성능 저하를 보여주고 있다. 하지만 또 다른 비교 모델인 UDGTM은 제안 모델과 함께 꾸준한 성능 유지를 보여주고 있다. 두 모델의 성능 분석을 위해 식 (7)을 사용한다.

$$Increasing\ Rate\ of\ Utilization = \frac{AVG(\sum Each\ Trading\ Model)}{AVG(\sum Single\ Auction)} \times 100\% \quad (7)$$

본 식은 단일 거래 모델을 기준으로 성능을 비교를 실시한다. 단일 거래 모델의 평균 성능을 기준으로 계산하며 각각의 비교 모델들의 평균 성능을 통하여 계산을 실시한다. 그 결과 각각의 모델들은 단일 거래 모델 기준으로 이중 거래 모델은 115%의 증가된 선을 보여준다. UDGTM의 경우 129%의 성능 증가를 보여주며 제안 모델은 앞선 모델들 보다 더 높은 136%의 성능 증가를 보여준다. 이러한 결과는 본 제안모델이 보다 안정적으로 거래 처리에서 많은 이점을 보여줄 수 있음을 입증한다.

4.4 거래 성공률

Fig. 9는 본 제안 모델 및 비교 모델들의 거래 성공률을 그래프로 나타낸 것이며, 평가에 사용된 식은 식 (8)와 같다. 각 모델들은 자신의 거래를 처리 할 수 없거나 가격 협정에 실패할 경우 거래 실패로 처리하게 된다. 제안 모델은 타 모델들에 비해 높은 성능을 보여준다.

이것은 Auctioneer에 존재하는 큐를 활용한 작업 정렬 및 예약 기능을 통하여 작업 처리 시간이 적합한 자원들을 지속적으로 탐색하고 이를 해당 자원에 배치함으로써, 거래 *Deadline*에 따른 거래 실패율을 최소화 할 수 있었다.

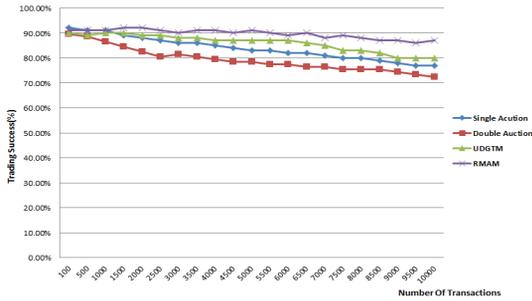


Fig. 9. Trading success(UDGTM : User Demand-based Grid Trade Management Model; RMAM : Resource availability-based Multi Auction Model)

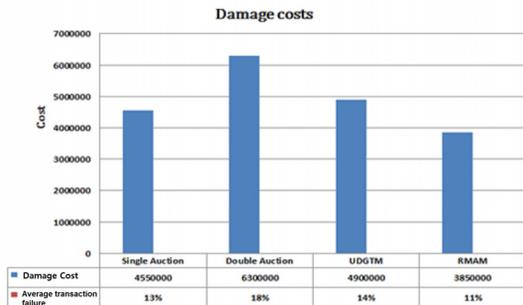


Fig. 10. Damage Cost

$$\text{Transaction failure rate} = \frac{\text{Number of successful operation}}{\text{Total resource allocation request}} (\%) \quad (8)$$

또한 Fig. 10은 거래 성공률을 평균적으로 계산하여 평균 실패율에 따른 비용 손해를 계산하여 나타낸 그래프이다. 비용은 무작위로 생성되는 Task 비용으로 인해, 공평성을 위하여 과학연산 서비스의 최대 지불 비용으로 계산하며 식 (8)을 사용한다. 제안 모델은 가장 낮은 손해비용을 보이고 있으며, 타 모델들보다 다양한 방향으로 성능이 우수함을 입증하였다.

5. 결 과

본 논문은 클라우드 환경에서 발생하는 자원 거래에 대하여 DEVS 방법론을 적용한 자원 가용성 기반 다중 경매 모델을 이용한 클라우드 자원 거래 시스템을 제안하였다. 제안 모델은 실험을 통하여 거래 성공률, QoS 만족도, 작업 처리 시간의 활용률 및 거래 실패율에 따른 손해비용을 평가 하였으며, QoS 만족도와 작업 처리 시

간, 거래 성공률 및 손해비용에 대해서 높은 성능을 보여 주었음을 입증하였다.

References

1. Yong Bum Ma, Jong Sik Lee, (2006), "User Demand - based Grid Trade Management Model", *The Korea Society for Simulation*, Vol. 15, No. 3, pp. 11-21.
2. Won-Yeol Lee, Eun-Gyeong Baek, Seong-Chun Lee (2009), "Virtual wireless network offer system and method of the radio interconnection unit base", Korea Patent No. 10-2012-0051451, 20120522.
3. Hun Sik Ju (2010), "Trends and Viewpoint in Technology of Cloud Computing", *Korea Society For Internet Information* Vol. 11, Issue 4, pp. 39-47.
4. Seung-Min Han, Eui-Nam Huh, Changwoo youn (2009), "Efficient Resource Recommendation System for Cloud Market Computing", *Korea Society For Internet Information* Vol. 11, Issue 3, pp. 121-129.
5. B. Hayes (2008), "Cloud computing", *Communications of the ACM*, vol 51, Issue 7, pp. 9-11.
6. B. Zeigler, Y. Moon, D. Kim and G. Ball (1997), The DEVS environment for high performance modeling and simulation, *Computational Science and Engineering*, IEEE CS&E, pp. 61-71.
7. Hesam Izakian, Ajith Abraham and Behrouz Tork Ladani (2010), An auction method for resource allocation in computational grids, *Future Generation computer System*, Vol. 26, Issue 2, pp. 228-235.
8. Ma, Y.B., Jang, S.H. and Lee, J.S. (2011), "Ontology-based Resource Management for Cloud Computing", *The 3rd Asian Conference on Intelligent Information and Database System (ACIIDS) 2011*, Deagu, Korea, pp. 343-352.
9. Mousumi Paul, Debabrata Samanta and Goutam Sanyal (2011), "Dynamic job Scheduling in Cloud Computing based on horizontal load", *International Journal of Computer Technology and Applications*, Vol. 2, Issue 5, pp. 1552-1556.
10. P. Mell and T. Grance, "The NIST Definition of Cloud Computing", *National Institute of Standards and Technology*, Information Technology Laboratory, Version 15.
11. P. Patel, A. Ranabahu, and A. Sheth (2009), "Service Level Agreement in Cloud Computing", *Cloud Workshops at OOPSLA09*, Available at: <http://knoesis.wright.edu/aboutus/visitors/summer2009/PatelReport.pdf>
12. R. Buyya (2002), "Grid Economy: A Market Paradigm

- for Distributed Resource Management and Scheduling for Service Oriented Grid Computing, PhD Thesis, Monash University, Australia.
13. R. Buyya, Che Shin Yeo and S. Venugopal (2008), "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Service as Computing Utilities," *High Performance Computing and Communication's*, HPCC '08. 10th IEEE International Conference on, pp. 513.
 14. U. Kant, D. Grosu (2005), "Double auction protocols for resource allocation in grids", *Information Technology: Coding and Computing*, 2005 ITCC 2005. International Conference on, Vol. 1, pp. 366-371.
 15. V. Hamscher, U. Schwiegelshohn (2001), a. Streit and R. Yahuapour: Evaluation of Job-Scheduling Strategies for Grid Computing, *Lectuer Notes in Computer Science* Volume 1971, pp. 191-202.
 16. X. Chu, K. Nadiminti, C. jin, S. Venugopal, and R. Buyya., "Next-Generation Enterprise Grid Plaform for e-Science and e-Business Applications", *IEEE International Conference on e-Science and Grid Computing*, Dec 2007.



이 석 우 (int3nd3r@gmail.com)

2010 한남대학교 컴퓨터공학과 학사
2011~현재 인하대학교 컴퓨터공학과 석사 과정

관심분야 : 모델링&시뮬레이션, 클라우드 컴퓨팅



김 태 영 (silverwild@gmail.com)

2007 인하대학교 컴퓨터공학부 학사
2009 인하대학교 정보공학과 석사
2009~현재 인하대학교 컴퓨터정보공학과 박사과정

관심분야 : 시스템 모델링&시뮬레이션, 분산처리



이 종 식 (jslee@inha.ac.kr)

1993 인하대학교 전자공학과 학사
1995 인하대학교 전자공학과 석사
2001 미국 애리조나대 전기·컴퓨터공학과 박사
2001~2002 캘리포니아 주립대학교 전기·컴퓨터공학과 전임강사
2002~2003 클리블랜드 주립대학교 전기·컴퓨터공학과 조교수
2003~현재 인하대학교 컴퓨터정보공학부 교수

관심분야 : 소프트웨어공학, 모델링&시뮬레이션