

# 분산 클라우드 컴퓨팅을 위한 동적 자원 할당 기법

안태형\*, 김예나\*, 이수경°

## Dynamic Resource Allocation in Distributed Cloud Computing

TaeHyoungh Ahn\*, Yena Kim\*, SuKyoung Lee°

### 요약

분산 클라우드 컴퓨팅에서 자원 할당 알고리즘은 사용자 만족도와 서비스 수용 및 처리 능력과 밀접한 관련을 가지기 때문에 중요하다. 즉, 분산 클라우드에서는 서비스 처리를 위해 이용가능한 자원이 없을 때 발생하는 서비스 거부하는 사용자 만족도를 반감시킨다. 따라서 본 논문에서는 서비스 거부를 최소화하기 위하여 데이터센터 자원 상황을 고려한 자원 할당 알고리즘을 제안한다. 제안하는 알고리즘은 Q-Learning 기반의 자원 할당량 학습에 의해서 클라우드 데이터센터에서 최대 자원 할당량 만큼 할당을 할 수 있으면 자원 할당량이 증가하고 그렇지 못할 때는 자원 할당량이 감소하게 된다. 본 논문에서는 제안하는 알고리즘과 기존의 두 알고리즘을 평가하고 제안하는 알고리즘이 두 알고리즘 보다 낮은 서비스 거부율을 보임을 증명한다.

**Key Words** : Cloud Computing, Distributed Cloud, Q-Learning, Resource Allocation, Service Rejection

### ABSTRACT

A resource allocation algorithm has a high impact on user satisfaction as well as the ability to accommodate and process services in a distributed cloud computing. In other words, service rejections, which occur when datacenters have no enough resources, degrade the user satisfaction level. Therefore, in this paper, we propose a resource allocation algorithm considering the cloud domain's remaining resources to minimize the number of service rejections. The resource allocation rate based on Q-Learning increases when the remaining resources are sufficient to allocate the maximum allocation rate otherwise and avoids the service rejection. To demonstrate, We compare the proposed algorithm with two previous works and show that the proposed algorithm has the smaller number of the service rejections.

### I. 서론

분산 클라우드는 다수의 데이터센터를 지리적으로 분산시켜 사용자와 데이터센터간의 거리를 좁힘으로서 통신 비용과 지연 시간을 줄일 수 있는 장

점을 가진다<sup>[1,2]</sup>. 분산 클라우드에서 자원 할당 알고리즘은 사용자로부터 분산 클라우드로 전송되는 서비스 처리를 위한 데이터센터를 선택하기 때문에 사용자 만족도(User Satisfaction)와 클라우드 시스템의 서비스 수용 및 처리 능력과 밀접한 관련을

※ 이 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. NRF-2013R1A1A3005914)

◆ 주저자 : 연세대학교 컴퓨터과학과 무선네트워킹 연구실, thahn@winet.yonsei.ac.kr, 학생회원

° 교신저자 : 연세대학교 컴퓨터과학과 무선네트워킹 연구실, sklee@cs.yonsei.ac.kr, 종신회원

\* 연세대학교 컴퓨터과학과 무선네트워킹 연구실, yenakim@winet.yonsei.ac.kr, 학생회원

논문번호 : KICS2013-04-173, 접수일자 : 2013년 4월 15일, 최종논문접수일자 : 2013년 7월 2일

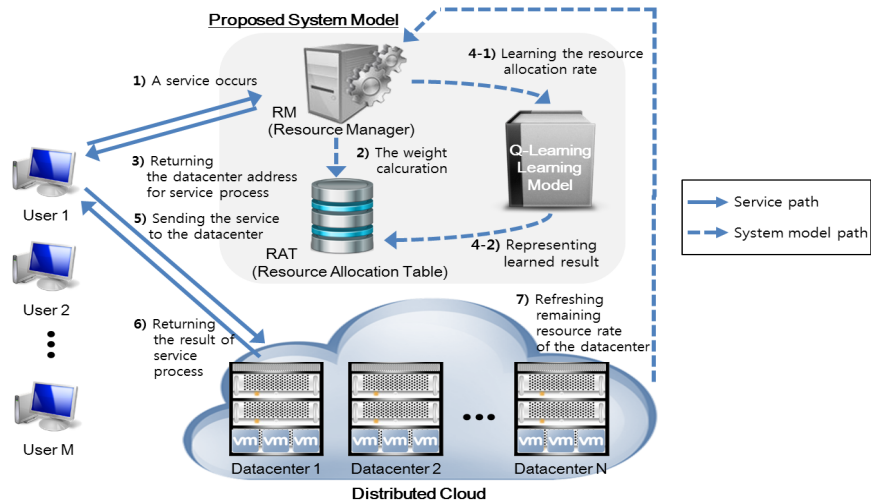


그림 1. 분산 클라우드 시스템을 위한 시스템 모델  
 Fig. 1. System model for a distributed cloud system.

가진다<sup>21</sup>. 즉, 서비스 처리를 위해 데이터센터에서 할당 할 수 있는 자원이 없어 발생하는 서비스 거부는 사용자 만족도를 반감시킨다<sup>31</sup>. 따라서 자원 할당 알고리즘은 클라우드 시스템에서 중요한 비중을 차지하기<sup>11</sup> 때문에 본 논문에서는 서비스 거부<sup>31</sup>를 최소화하기 위하여 Q-Learning<sup>41</sup>을 이용한 동적 자원 할당 알고리즘을 제안한다.

데이터센터는 다수의 VM(Virtual Machine)으로 구성되어 있고 VM은 클라우드 시스템에서 사용자에게 할당 할 수 있는 자원의 최소단위이다<sup>2,5,6,71</sup>(이하 자원이라 칭함). 데이터센터에 도착한 서비스 처리를 위하여 클라우드 시스템은 다수의 자원을 할당 할 수 있다.

앞서 제안된 논문들<sup>2,51</sup>은 최대 자원 할당량이 고정되어 있어 서비스 발생량이 많거나 적거나 상관없이 항상 고정된 최대 자원 할당량을 가진다. [5]에서는 서비스를 요청한 사용자와 가장 가까운 곳에 위치한 데이터센터가 서비스를 처리할 수 있으면 처리하고 그렇지 않으면 근처의 다른 데이터센터에게 전이(Transfer)한다. 이 방법은 서비스 발생이 집중되는 홈 데이터센터의 자원이 고갈된다면 추후 모든 요구에 추가적인 전이 비용을 발생시키게 된다. 따라서 본 논문에서는 데이터센터간의 전이 시간과 고정된 최대 자원 할당량을 없애기 위하여 RM(Resource Manager)과 RAT(Resource Allocation Table)를 제안한다. 마스터 서버 RM은 모든 Service Request를 받아 도메인에 할당 하고 RAT는 동적인 자원 할당량을 위하여 Q-Learning을 이용하여 학습된 자원 할당량을 유지한다. 그리고 서비스 처리를 위한 자원 할당 시

항상 고정된 자원 할당량을 가지는 Greedy 알고리즘과 [5]에서 제안하는 SMDP 기반의 결정 알고리즘의 일부분을 구현한 CD(Conservative Decision)와의 시뮬레이션을 통해 제안하는 알고리즘이 두 알고리즘보다 낮은 서비스 거부율을 보임을 증명한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 사용하는 시스템 모델을 제안하고 3장에서는 시뮬레이션을 통하여 알고리즘을 평가한다. 마지막으로, 4장에서는 본 논문의 결론을 맺는다.

## II. 분산 클라우드 시스템을 위한 시스템 모델

이번 장에서는 데이터센터의 가중치를 계산하여 서비스 처리에 적합한 데이터센터를 선택하는 절차와 학습된 자원 할당량으로 처리하는 과정을 설명한다. 또한, 본 논문에서 제안하는 RM과 RAT를 적용한 시스템 모델을 제안한다. 2.1장에서는 본 논문에서 제안하는 시스템 모델에 대하여 정의하고 2.2장에서는 Q-Learning 기반의 동적 자원 할당량 학습에 관하여 설명한다. 마지막으로, 2.3장에서는 데이터센터의 가중치 계산에 대해서 설명한다.

### 2.1. 시스템 모델

그림 1은 본 논문에서 제안하는 시스템 모델로서 서비스가 시스템에 도착했을 때 처리하는 순서를 나타낸다.

- 1) 모든 사용자로부터 발생하는 서비스는 마스터 서버 RM에 도착한다.
- 2) RM에서는 데이터센터의 자원 할당량과 사용자부터 각 데이터센터까지의 거리를 고려하여

데이터센터의 가중치  $w(s,a)$ 를 계산한다.

- 3) 2)에서 계산된 데이터센터들의 가중치 중 가장 큰 값인  $\max w(s,a)$ 가 서비스 처리에 가장 적합한 데이터센터이므로 해당 데이터센터의 주소와 자원 할당량을 사용자에게 반환한다. 할당 가능한 자원이 없을 때는 서비스 거부 발생한다.
- 4) 데이터센터에서는 서비스 처리 후 서비스 처리에 할당된 자원량과 데이터센터에 남아 있는 자원량을 Q-Learning에 적용하여 데이터센터들의 자원 할당량을 유지하고 있는 RAT의 자원 할당량 값을 학습한다.
- 5) 사용자는 전달 받은 데이터센터의 주소와 자원 할당량을 서비스와 함께 전달 받은 데이터센터의 주소로 전달하여 처리를 요청한다.
- 6) 데이터센터에서는 전달 받은 자원 할당량 만큼의 자원을 할당하여 서비스를 처리하고 결과를 사용자에게 반환한다.
- 7) 데이터센터에서 서비스 처리를 위해 할당된 자원은 처리가 끝난 후 다시 가용할 수 있는 상태로 변경되었으므로 RM에게 알린다.

본 논문에서는 RM과 RAT를 제안하여 고정된 최대 자원 할당량과 데이터센터간의 전이 시간을 없애도록 한다. 추가적인 전이 시간을 없애기 위하여 마스터 서버 RM은 모든 서비스를 받아 데이터센터에 할당 하고 동적인 자원 할당량을 위하여 RAT는 Q-Learning을 이용하여 학습된 자원 할당량을 유지한다.

RAT는 클라우드 시스템에 존재하는 각 데이터센터에서 서비스 처리를 위해 할당 할 수 있는 자원 할당량을 유지한다. RAT에 유지되는 각 데이터센터의 자원 할당량은 서비스 처리 후 Q-Learning을 이용하여 데이터센터 자원 상황을 고려한 자원 할당량으로 업데이트된다(이하 데이터센터의 자원 할당량은 RAT에 유지되는 데이터센터의 자원 할당량을 칭함). 즉, 데이터센터에서 최대 자원 할당량 만큼 할당을 할 수 있으면 자원 할당량이 증가하고 그렇지 못할 때는 자원 할당량이 감소시키는 방향으로 학습시키는 것을 말한다. 자원 할당량 학습에 관해서는 2.2 장에서 자세히 설명한다.

클라우드 시스템에 존재하는 데이터센터들의 자원 상황을 알고 있는 마스터 서버 RM은 모든 사용자로부터 서비스를 받아서 처리에 적합한 데이터센터를 선택하는 역할을 한다. 데이터센터 선택 시 RM은 데이터센터들의 자원 할당량과 서비스가 발생한 사용자

부터 데이터센터까지의 거리를 고려하여 데이터센터의 가중치,  $w(s,a)$ 를 구한다. 서비스 처리를 위한 데이터센터 선택 알고리즘에 관해서는 2.3장에서 자세히 설명한다.

## 2.2. Q-Learning 기반의 자원 할당량 학습

Q-Learning<sup>[4]</sup>은 강화 학습(Reinforcement Learning)에 속하는 대표적인 알고리즘 중에 하나로 모바일 통신 등 다양한 분야의 자원 할당 알고리즘에 적용되어 이미 많은 연구에 의해서 효율성은 증명되었다<sup>[8-11]</sup>. 클라우드 컴퓨팅 분야에서도 Q-Learning을 적용하여 사용자가 스스로 서비스 처리에 필요한 자원량을 학습하는 분산 학습 알고리즘이<sup>[12]</sup> 제안되었다. 하지만 마스터 서버를 활용하여 데이터센터의 자원 할당량을 학습하는 알고리즘은 아직 연구되지 않았다. 따라서, 본 논문에서는 마스터 서버 RM에서 Q-Learning을 사용하여 RAT에서 유지하는 데이터센터의 자원 할당량을 데이터센터의 남아 있는 자원 상황에 맞게 학습시킨다.

Q-Learning은 시간  $t$ 의 상태( $s$ , State)에서 행동( $a$ , Action)를 취하였을 때 받는 보상( $r(s,a)$ , Reward)을 바탕으로 최적의 가치함수(Value Function)  $Q^*(s,a)$ 를 예측하기 위하여  $Q(s,a)$ 를 다음과 같이 업데이트한다. 학습속도를 결정하는 학습률(Learning Rate)과 미래 행위에 대한 보상의 할인 계수(Discount Factor)는 각각  $\alpha$ ,  $\gamma$ 로 나타낸다.

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha * [r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_{t+1}, a_{t+1})] \quad (1)$$

본 논문에서 제안하는 자원 할당 알고리즘에서는 Q-Learning을 클라우드 시스템에 적용하기 위하여 다음과 같이 상태, 행동 그리고 보상을 정의한다. 사용자로부터 발생하는 서비스 처리를 위해 시스템에서 할당 할 수 있는 자원의 최소 단위는 VM이다.

- 1) 상태( $s$ ) : 상태  $s$ 는 클라우드 시스템에 존재하는 데이터센터들의 남아 있는 자원량으로 정의된다.  $n$ 은 클라우드에 존재하는 총 데이터센터 수를 말하며  $n \in \{1, 2, 3, \dots, N\}$ 으로 표현되고  $d_n$ 은  $n$ 번째 데이터센터의 남아 있는 자원량으로서 상태  $s = \langle d_1, d_2, \dots, d_N \rangle$ 로 정의된다.
- 2) 행동( $a$ ) : 상태  $s$ 에서 서비스에 대한 행동  $a$ 는 *Accept*, *Reject*가 행해 질 수 있으며 행동집합(Action Set)은  $a(s)$ 로 나타낼 수 있다. *Accept*( $a(s)=n$ )는  $n$ 번째 데이터센터에서 서비스

표 1. 시뮬레이션 변수<sup>[5]</sup>  
Table 1. Simulation parameter<sup>[5]</sup>.

Parameter	Value	Parameter	Value
T	50	$\alpha_i$	5
$\epsilon$	1	$\alpha_d$	2
$\mu$	6.0	$\alpha_e$	-1
$\alpha$	0.1	$\gamma$	0.1

처리를 하는 것을 말하고  $Reject(a(s)=0)$ 는 데이터 센터들에서 처리할 수 있는 자원이 없어 서비스 거부가 발생하는 것을 뜻한다. 따라서 행동 공간 (Action Space)  $Act=0,1,2,\dots,N$ 이다.

- 3) 보상( $r(s,a)$ ) : 보상  $r(s,a)$ 는 상태  $s$ 에서 행동  $a$ 를 하여 산출된 결과를 이용하여 구해진다. 즉, 보상은 상태  $s$ 가  $\langle d_1, d_2, \dots, d_N \rangle$ 와 같이 구성되어 있을 때 서비스 처리를 위해 선택된  $a$ 번째 데이터센터에서 서비스 처리에 할당된 자원량( $g(s,a)$ )과 데이터센터에 남아 있는 자원량( $C_{sa}$ )을 고려하여 서비스가 할당된 데이터센터의 자원 할당량 증감, 감소, 전체 데이터센터의 자원 할당량 감소 중 어느 것을 할 지 결정한다.

상태  $s$ 에서 서비스 처리를 위해  $a$ 번째 데이터센터에서 할당한 실제 자원량과 상태  $s$ 에서 RAT에 유지되는  $a$ 번째 데이터센터의 자원 할당량이 같은 경우( $g(s,a)=C_{sa}$ )에는 데이터센터에 자원의 여유가 있어 할당 할 수 있는 자원 할당량  $C_{sa}$ 만큼을 할당했기 때문에 자원 할당량의 증가를 초래하는 값,  $\sigma_i$ 를 보상으로 결정한다. 데이터센터의 남아 있는 자원량이  $C_{sa}$ 만큼의 자원이 없는 경우( $g(s,a)<C_{sa}$ ) 또는 데이터센터에 남는 자원이 없는 경우( $O_{sa}=0$ )에는 자원이 부족해  $C_{sa}$ 만큼 자원 할당을 하지 못했기에 자원 할당량을 감소시키기 위하여 보상으로  $\alpha_d$ 를 할당한다. 서비스 거부가 발생한 경우 ( $a(s)=0$ )에는 서비스 거부 발생을 줄이고 보다 많은 서비스를 받아들이기 위하여 모든 데이터센터의 자원 할당량을 줄이기 위한 값,  $\alpha_e$ 을 보상으로 결정한다.  $g(s,a)$ 를 결정하는 방법에 관해서는 2.3장에서 설명한다. 따라서  $r(s,a)$ 는 다음과 같이 나타낼 수 있다.

$$r(s,a) = \begin{cases} \sigma_i, & g(s,a) = C_{sa} \\ \sigma_d, & g(s,a) < C_{sa} \text{ or } O_{sa} = 0 \\ \sigma_e, & a(s) = 0 \end{cases} \quad (2)$$

### 2.3. 데이터센터 가중치 계산

본 논문에서 제안하는 마스터 서버 RM에서는

사용자로부터 발생한 서비스를 처리하는데 적합한 데이터센터를 찾기 위하여 거리적인 요소 뿐만 아니라 할당 가능한 자원 할당량을 고려하여 각 데이터센터의 가중치  $w(s,a)$ 를 계산한다.  $w(s,a)$ 는 상태  $s=\langle d_1, d_2, \dots, d_N \rangle$ 에서  $a$ 번째 데이터센터의 가중치를 계산한다. 서비스가 RM에 도착했을 때 서비스 처리를 위한 데이터센터들에 자원이 남아 있지 않을 때는 ( $Reject, a(s)=0$ ) 별도의 서비스 처리 비용이 필요하지 않기 때문에  $w(s,a)=0$ 이 된다. 데이터센터들에 서비스 처리를 위한 자원이 있으면 ( $Accept, a(s) \in n$ ) 각 데이터센터의 가중치  $w(s,a)$ 를 다음과 같이 계산하여 서비스 처리에 효율적인 데이터센터를 선택한다. 데이터센터는 새로운 서비스를 받아들임에 따라 수익 T를 얻게 되고 서비스를 전달하기 위한  $\Theta_d \epsilon$ 의 전이 비용과 실제 자원 할당량  $g(s,a)$ 만큼의 자원을 사용하여  $\Theta_d \epsilon / g(s,a) \mu$ 의 자원 사용 비용이 든다.  $1/g(s,a) \mu$ 는  $g(s,a)$ 개의 자원 사용 시간을 의미하고  $\mu$ 는 단위 시간 동안 하나의 자원이 서비스를 처리하는 량을 뜻한다.  $\epsilon$ 은 시간 당 비용을 말하며 서비스의 수익 T와 단위가 같다. 서비스 처리를 위해 할당한 실제 자원량  $g(s,a)$ 는 상태  $s$ 에서  $a$ 번째 데이터센터에 남아 있는 자원량  $O_{sa}$ 가 상태  $s$ 에서 RAT에서 유지하는  $a$ 번째 도메인의 자원 할당량  $C_{sa}$ 보다 같거나 크면 ( $O_{sa} \geq C_{sa}$ )  $C_{sa}$ 만큼 할당을 할 수 있기에  $C_{sa}$ 로 결정되고 그렇지 않으면 ( $O_{sa} < C_{sa}$ )  $O_{sa}$ 로 결정된다. 상태  $s$ 에서  $a$ 번째 데이터센터의  $O_{sa}$ 는  $O_{s'a} = O_{sa} - g(s,a)$ 와 같이 정의된다.  $s'$ 는 상태  $s$ 의 다음 상태를 나타낸다. 서비스 전이 시간  $\Theta_d = \Theta_i + \Theta_e$ 로서 서비스가 발생한 사용자로부터 RM까지의 전이 시간  $\Theta_i$ , 데이터센터까지의 전이 시간  $\Theta_e$ 의 합으로 나타낸다.

위 설명에 의해  $w(s,a)$ 는 다음과 같이 계산된다.

$$w(s,a) = \begin{cases} T - \theta_d \epsilon - \frac{\epsilon}{g(s,a) \mu}, & a(s) \in n \\ 0, & a(s) = 0 \end{cases} \quad (3)$$

식 (3)을 이용하여 클라우드 시스템에 존재하는 모든 데이터센터들의 가중치를 계산한 후 최적의 가중치  $w^*(s,a)$ 를 구한다.  $w^*(s,a)$ 는 서비스에 가장 적합한 데이터센터로서 클라우드 시스템에서 가중치가 가장 큰 데이터센터  $w^*(s,a) = \max_{a \in n} w(s,a)$ 로 나타낸다.

## III. 시뮬레이션

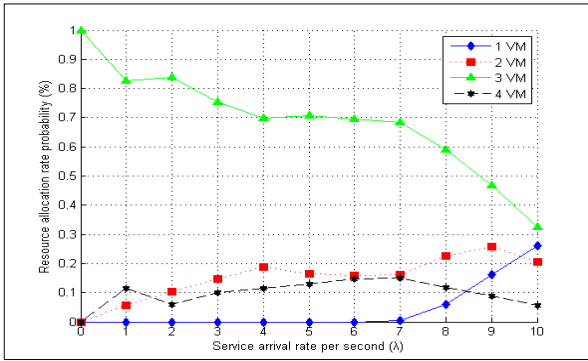


그림 2. Service Request 발생량에 따른 자원 할당량 비율  
 Fig. 2. Resource allocation rate probability under various service arrival rates.

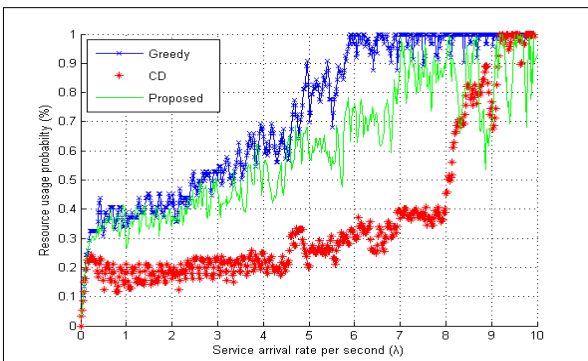


그림 3. Service Request 발생량에 따른 자원 사용률  
 Fig. 3. Resource usage probability under various service arrival rates.

본 장에서는 Java 기반으로 작성된 시뮬레이터를 사용하여 분산 클라우드 환경에서 제안하는 알고리즘과 Greedy, CD(Conservative Decision) 알고리즘 간의 자원 할당 성능을 평가한다. Greedy 알고리즘은 자원 할당 시 항상 할당 할 수 있는 고정된 자원 할당량을 할당하고 CD는 서비스가 발생한 위치에서 가장 가까운 거리에 있는 데이터센터에 서비스를 전이 하여 처리하도록 한다. CD는 서비스 처리를 위한 자원 할당에 관해서는 적은 자원 할당량으로 서비스를 처리하지만 데이터센터에서 서비스 처리를 위한 자원이 없을 시에는 다른 가까운 데이터센터로 전달을 한다.

시뮬레이션에서 사용되는 변수들간의 관계를 파악하기 위하여 값을 변경하면서 30000초 동안 다양한 시뮬레이션을 했다. 시뮬레이션에서 사용된 클라우드 시스템은 8개의 데이터센터로 구성되고 각 데이터센터는 10, 30, 15, 20, 35, 20, 25, 30개의 자원으로 구성된다. 초 당 서비스 발생량  $\lambda$ 는 Poisson 분포를 따르며 시뮬레이션 시 시간의 흐름에 따라 발생률을 높

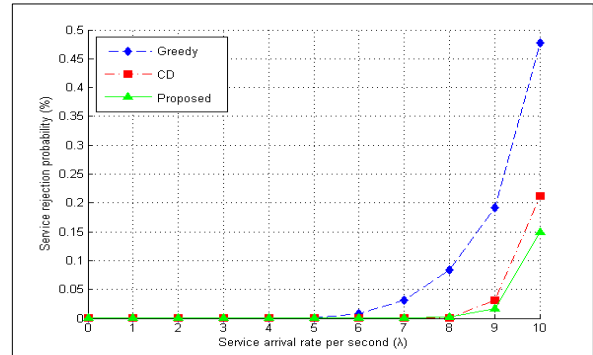


그림 4. Service Request 발생량에 따른 Service Request 거부율  
 Fig. 4. Service reject probability under various service request arrival rates.

이면서 진행하였다. 자원 할당량의 경우, [5]과 같이 세 알고리즘 모두 3개의 자원 할당량(VM)을 기본으로 하고  $\Theta_s$ 와  $\Theta_c$ 는 각각 2 ms, 30 ms로 설정하였다. 시뮬레이션 시 사용된 파라미터들은 [5]을 기반으로 값을 설정하였고 표 1과 같다.

그림 2는 제안하는 알고리즘의 서비스 발생량에 따른 자원 할당량 비율을 보여주며 학습에 의해서 자원 할당량이 어떻게 변화하는지를 나타낸다. 데이터센터 자원 상황을 고려한 자원 할당량의 학습에 의해서 서비스 발생량이 적을 때는 3 VM, 4 VM이 할당될 확률이 높고 발생량이 많아질수록 데이터센터의 자원이 부족하게 되어 1 VM, 2 VM이 할당될 확률이 높아지는 것을 볼 수 있다.

그림 3은 세 알고리즘들의 서비스 발생량에 따른 자원 사용률을 보여준다. Greedy의 경우에는 모든 서비스 처리 시 고정된 자원 할당량 때문에 높은 자원 사용률을 보인다. CD는 많은 서비스가 시스템에 도착하면 낮은 자원 할당량을 유지하기 때문에  $\lambda=7$  이전에는 자원 사용률이 0.2 ~ 0.4 사이에서 완만한 상승률을 보인다. 하지만 서비스 발생량이 많아질수록 가까운 데이터센터의 자원은 고갈되고 다른 데이터센터로 서비스를 전이 시간이 증가하게 된다. 따라서 서비스 전이 시간에 의해 처리 시간은 증가하게 되고 자원 사용률은 증가하므로써  $\lambda=7$  이후로 급격한 자원 사용률의 증가를 나타낸다. 반면, 제안하는 알고리즘의 경우에는 CD와는 달리 마스터 서버 RM에 의해서 서비스 처리를 위한 데이터센터를 선택하기 때문에 서비스 전이 비용이 발생하지 않는다. 또한 데이터센터 자원 상황을 고려한 자원 할당량 학습에 의해서 그림 2에서 보이는 것처럼 서비스 발생량이 적을 때는 3 VM, 4 VM이 할당될 확률이, 서비스 발생량이 많을 때는 1 VM, 2 VM이 할당될 확률이 높기 때문에 효

율적으로 높은 자원 사용률을 보인다. 하지만 데이터 센터의 자원 상황을 고려하지 않고 항상 고정된 할당량을 가지는 Greedy에 비해서는 적다.

그림 4에서 나타나는 서비스 거부율은 그림 3에서 설명한 세 알고리즘들의 자원 사용률과 밀접한 관계를 가지고 있다. Greedy의 경우에는 고정된 자원 할당량 때문에 높은 거부율을 보인다. CD는 서비스 처리에 낮은 자원 할당량에 의해서 Greedy 보다 낮은 거부율을 보이지만 전이 시간 때문에 제안하는 알고리즘 보다 높은 거부율을 보인다. 제안하는 알고리즘은 데이터센터 자원 상황을 고려한 자원 할당량 학습에 의해서 낮은 서비스 거부율을 보인다.

위 설명에 의해 제안된 알고리즘은 데이터센터에 남아 있는 자원 상황을 고려하여 효율적으로 자원을 사용할 뿐만 아니라 서비스 거부율 측면에서도 Greedy, CD 보다 각각 69%, 30% 좋은 결과를 보여 준다. 이러한 결과가 나온 이유는 CD와는 달리 서비스 전이 시간이 없고 동적인 상황에서 데이터센터의 자원 상황을 고려한 자원 할당량 학습을 활용하여 자원 할당을 하기 때문이다. 따라서 서비스가 많이 발생할 때는 자원 할당량이 감소하고 적을 때는 자원 할당량이 증가하게 된다.

#### IV. 결 론

본 논문에서는 전이 시간과 고정된 자원 할당량을 제거하기 위하여 RM과 RAT를 제안하였다. 전이 시간을 없애기 위하여 마스터 서버 RM은 사용자로부터 발생하는 서비스를 받아 데이터센터에 할당했고 서비스 처리를 위한 동적인 자원 할당량을 위하여 RAT에서는 데이터센터 자원 상황을 고려한 Q-Learning 기반의 자원 할당량 학습을 하는 자원 할당 알고리즘을 제안하였다. 또한 시뮬레이션을 통하여 제안하는 알고리즘의 자원 할당량이 서비스 발생량에 따라서 동적으로 데이터센터의 자원상황을 고려하여 변하는 것을 확인하였고 제안한 알고리즘이 Greedy 알고리즘과 CD보다 낮은 서비스 거부율을 보임을 증명하였다.

#### References

[1] P. T. Endo, A. V. P. de Almeida Palhares, N. N. Pereira, G. E. Goncalves, D. Sadok, J. Kelner, B. Melander, and J.-E. Mangs, "Resource allocation for distributed cloud:

concepts and research challenges," *IEEE Network*, vol. 25, no. 4, pp. 42-46, Jul.-Aug. 2011.

[2] M. Alicherry and T. V. Lakshman, "Network aware resource allocation in distributed clouds," in *Proc. IEEE INFOCOM*, pp. 963-971, Orlando, U.S.A., Mar. 2012.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50-58, Apr. 2010.

[4] C. J. C. H. Watkins and P. Dayan, "Q-Learning," *Machine Learning*, vol. 8, pp. 279-292, Jan. 1992.

[5] H. Liang, L. X. Cai, D. Huang, X. Shen, and D. Peng, "An SMDP-based service model for interdomain resource allocation in mobile cloud networks," *IEEE Trans. Veh. Tech.*, vol. 61, no. 5, pp. 2222-2232, Jun. 2012.

[6] S. T. Maguluri, R. Srikant, and L. Ying, "Stochastic models of load balancing and scheduling in cloud computing clusters," in *Proc. IEEE INFOCOM*, pp. 702-710, Orlando, U.S.A., Mar. 2012.

[7] H. Kim, W. Kim, and Y. kim, "A pattern-based prediction model for dynamic resource provisioning in cloud environment," *KSII Trans. Internet Inform. Syst.*, vol. 5, no. 10, pp. 1712-1732, Oct. 2011.

[8] J. Nie and S. Haykin, "A Q-learning-based dynamic channel assignment technique for mobile communication systems," *IEEE Trans. Veh. Tech.*, vol. 48, no. 5, pp. 1676-1687, Sep. 1999.

[9] Y. Chen, C. Chang, and F. Ren, "Q-learning-based multirate transmission control scheme for RRM in multimedia WCDMA systems," *IEEE Trans. Veh. Tech.*, vol. 53, no. 1, pp. 38-48, Jan. 2004.

[10] L. Hu, X.-L. Che, and S.-Q. Zheng, "Online system for grid resource monitoring and machine learning-based prediction," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 1,

pp. 134-145, Jan. 2012.

- [11] Y. J. Kim, "Modified Q-learning for intelligent system," *J. Korea Inst. Commun. Soc. (KICS)*, vol. 33, no. 2, pp. 82-87, Feb. 2008.
- [12] J. Rao, X. Bu, C.Z. Xu, and K. Wang, "A distributed self-learning approach for elastic provisioning of virtualized cloud resources," in *Proc. IEEE MASCOTS*, pp. 45-54, Singapore, Jul. 2011.

안 태 형 (TaeHyoung Ahn)



2012년 2월 광운대학교 컴퓨터 소프트웨어학과 졸업  
2012년 3월~현재 연세대학교 컴퓨터과학과 석사과정  
<관심분야> 클라우드 컴퓨팅

김 예 나 (Yena Kim)



2010년 2월 단국대학교 컴퓨터 과학과 졸업  
2010년 3월~현재 연세대학교 컴퓨터과학과 박사과정  
<관심분야> Wireless Network, Vehicular-Ad-hoc Networking, 클라우드 컴퓨

팅

이 수 경 (SuKyoung Lee)



2005년 9월~현재 연세대학교 컴퓨터과학과 교수  
<관심분야> 이종망통합 (4G, WLAN), Mobility Management, Vehicular-Ad-hoc Network-working, Optical Network