ISSN 2383-6318(Print) / ISSN 2383-6326(Online)
KIISE Transactions on Computing Practices, Vol. 23, No. 8, pp. 481-486, 2017. 8
https://doi.org/10.5626/KTCP.2017.23.8.481

하이브리드 클라우드 환경에서의 응용 특성 가중치를 고려한 자원 군집화 기법

(A Resource Clustering Method Considering Weight of Application Characteristic in Hybrid Cloud Environment)

오 유 리 † 김 윤 희 ††
(Yoori Oh) (Yoonhee Kim)

요 약 클라우드의 원하는 자원을 필요한 만큼만 사용하고 지불하는(Pay-per-use) 방식을 이용하여 과학 응용을 수행하고자 하는 과학자들이 늘어나는 추세이다. 그러나 다양한 특성으로 구성된 클라우드 자원으로 과학자들은 적절한 자원을 선택하는데 어려움을 겪는다. 이에 따라 자원의 효율적인 활용을 위하여 과학자가 실험하고자하는 응용의 특성에 따라 동적으로 자원을 분류하는 것이 필요하다. 본 연구에서는 하이브리드 클라우드 환경에서 응용의 특성을 반영한 자원 군집 분석 기법을 제안한다. 자원 군집 분석은 자기조직화지도 및 K-평균 알고리즘을 적용하여 유사한 자원을 군집화한다. 제안한 알고리즘을 통해 과학 응용의 특성을 반영한 유사 자원 군집을 형성하였음을 증명한다.

키워드: 하이브리드 클라우드, 자기조직화지도, 자원 분류, 응용 특성 가중치

Abstract There are many scientists who want to perform experiments in a cloud environment, and pay-per-use services allow scientists to pay only for cloud services that they need. However, it is difficult for scientists to select a suitable set of resources since those resources are comprised of various characteristics. Therefore, classification is needed to support the effective utilization of cloud resources. Thus, a dynamic resource clustering method is needed to reflect the characteristics of the application that scientists want to execute. This paper proposes a resource clustering analysis method that takes into account the characteristics of an application in a hybrid cloud environment. The resource clustering analysis applies a Self-Organizing Map and K-means algorithm to dynamically cluster similar resources. The results of the experiment indicate that the proposed method can classify a similar resource cluster by reflecting the application characteristics.

Keywords: hybrid cloud, self-organizing map, resource classification, application characteristic weight

· 이 논문은 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2017R1 A2B4005681)
· 이 논문은 제43회 동계학술발표회에서 '응용 특성 군집의 분석을 통한 클라우드 자원 분류 기법'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 숙명여자대학교 컴퓨터과학과
yoori0203@sookmyung.ac.kr
†† 종신회원 : 숙명여자대학교 컴퓨터과학과 교수
(Sookmyung Women's Univ.)
yulan@sookmyung.ac.kr
(Corresponding author임)

논문접수 : 2017년 2월 17일
(Received 17 February 2017)
논문수정 : 2017년 5월 24일
(Revised 24 May 2017)
심사완료 : 2017년 5월 25일
(Accepted 25 May 2017)

Copyright©2017 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.
정보과학회 컴퓨팅의 실제 논문지 제23권 제8호(2017. 8)

1. 서 론

최근 클라우드 컴퓨팅 기술은 인터넷만 연결되어 있다면 사용자가 필요로 하는 자원을 시간과 공간의 제약 없이 원하는 만큼 빌려 쓸 수 있다는 특성으로 각광을 받고 있다. 이러한 클라우드 컴퓨팅 서비스를 제공하는 회사들은 Amazon EC2[1], Microsoft Window Azure [2], KT Ucloud[3], Google Cloud Engine[4], NHN Entertainment[5] 등으로 다양하다. 그러나 각 회사들은 제공하는 자원의 표기법이 서로 상이하여 사용자는 본인에게 적합한 자원을 선택하는데 이를 조사하고 비교하는데 많은 시간과 노력을 소비해야 한다. 따라서 클라우드 서비스 간에 상호호환성을 제공해주는 하이브리드 클라우드 환경에서 사용자가 클라우드 컴퓨팅 서비스 제공회사에 관계없이 자원을 선택하기 위해 자원 명세의 표준화된 표기법이 필요하다. 클라우드 자원은 다양한 특성으로 구성되기 때문에 과학 응용을 실험하고자 하는 과학자들 입장에서 응용의 특성에 따라 적합한 클라우드 자원을 선택하는 것이 어렵다. 따라서 적합한 자원 선택 등 효율적인 자원 활용을 위하여 자원에 대한 분류가 필요하다. 한편, 과학자들은 실험에 따라 다양한 실험의도 및 요구사항을 제출하기 때문에 자원에 대해 정적인 분류법을 적용하는 것은 불명확한 분류 결과를 도출할 수 있다. 따라서 응용의 특성에 따라 관점을 달리하여 상황에 맞게 동적으로 자원의 분류를 변화시킬 필요가 있다. 동적인 자원 분류는 사용자 요구사항을 충족시키며 정확하게 필요로 하는 자원추천이 가능하다.

본 논문에서는 하이브리드 클라우드 환경에서 응용 특성 가중치를 고려한 자원 군집화 기법을 제안한다. 제안하는 알고리즘을 통해 응용의 특성을 반영한 유사 자원의 군집을 형성함을 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 클라우드 자원의 분류에 대한 관련 연구를 소개하고 3장에서는 자원 군집 분석 알고리즘을 제안한다. 4장에서는 제안하는 알고리즘의 성능을 평가하기 위해 실험을 진행하고 그 결과를 분석한다. 마지막으로 5장에서는 본 연구의 결론을 맺는다.

2. 관련 연구

본 장에서는 클라우드 자원에 대한 다양한 분류에 대한 관련연구를 소개한다.

다양한 클라우드 컴퓨팅 서비스 제공 회사는 각 회사의 방침대로 수 많은 서로 다른 성능의 가상 자원을 제공한다. 이에 따라 클라우드 자원을 분류하는 기법에 대한 연구들이 진행되고 있다. Chavan[6]는 가상자원을 군집화 할 때, K-평균 클러스터링 기법을 적용하고 가상자원에 할당된 메모리를 고려하여 군집을 형성한다. 해당 연구에서 군집화는 가상 자원의 재구성 및 스케줄링에 이용된다. 이는 메모리만을 고려하여 군집을 형성하여, 자원의 다양한 속성을 고려하지 못한다. Wu[7]는 클라우드 서비스 자원을 베이지안 분류 알고리즘을 이용하여 분류한다. 자원의 여러 가지 요소의 유사성을 계산하여 분류한다. 특히, 특정 요소가 분류에 영향을 크게 주기 위하여 가중치를 고려한 베이지안 식을 이용하여 특징의 유사성을 계산한다. 그러나 위 연구에서는 실행하고자 하는 응용에 대한 고려가 없으며 다양한 자원 자체에 대한 분류를 했다. 안윤선[8]은 제안하는 연구와 유사한 연구로 인터클라우드 환경에서 응용의 특성을 고려하여 자원을 동적으로 분류하는 기법을 제안한다. 해당 연구에서 적용하는 k-modes 알고리즘은 해당 특성의 값이 같은지의 여부를 확인하여 유사도를 계산한다. 이는 해당 속성의 값이 같을 때만 유사도에 기여할 수 있기 때문에 비슷한 속성 값에 대한 반영이 어려워 두 자원간의 유사도를 파악하는데 부족하다.

3. 응용 특성 가중치를 고려한 자원 군집 분석 서비스

본 장에서는 응용 특성 가중치를 고려한 자원 군집 서비스 구조, 알고리즘에 적용한 기법, 자원 군집 분석 알고리즘에 대해 소개한다.

3.1 응용 특성 가중치를 고려한 자원 군집 서비스 구조

본 연구에서는 응용 특성 가중치를 고려한 자원 군집 서비스 구조를 제안한다. 하이브리드 클라우드 환경에서 응용의 특성을 반영하여 자원의 군집을 형성한다.

사용자가 실행하고자 하는 응용을 제출하였을 때, 해당 응용의 특성을 기반으로 자원 군집을 형성한다. 자원은 하이브리드 클라우드 자원의 명세를 바탕으로 군집 분석을 진행한다. 실행하고자 하는 응용의 특성을 반영하여 군집이 형성된다.

자원 군집 분석 서비스는 개인적으로 구축하여 사용하는 사설 클라우드와 클라우드 컴퓨팅 서비스를 제공하는 공용 클라우드의 자원을 모두 고려하는 하이브리드 클라우드 자원에 대해 군집을 형성한다. 이 때, 사용자가 제출한 응용의 특성을 반영하여 유사한 자원끼리 군집을 형성하도록 한다.

3.2 자기조직화지도 및 K-평균 군집 분석

자기조직화지도는 대뇌피질을 모델화한 인공신경망의 일종이다[9]. 이는 자율학습으로 훈련되며 고차원의 데이터를 일반적으로 2차원인 저차원의 지도를 생성한다. 이 지도는 입력공간에서 주어진 훈련 샘플에 대한 이산적인 표현을 나타내며, 입력 공간에 대한 위상 속성을 보존한다.

elbow method[10]는 군집분석 시, 군집의 개수를 결정하는데 이용하는 방법이다. 형성되는 군집의 개수에 따라 각 경우에 군집의 중심과 해당 군집에 속한 요소 간의 거리 합을 도출한다. 해당 값이 급격히 감소하다 군집의 개수가 증가하더라도 많이 감소하지 않은 시점이 적정한 군집의 개수로 선정한다.

K - 평균 군집분석 알고리즘[11]은 주어진 데이터를 K 개의 군집으로 묶는 알고리즘으로 각 군집과 거리 차이의 분산을 최소화 하는 방식이다.

3.3 응용 특성을 고려한 자원 군집분석

응용 특성을 반영한 자원 군집 분석은 크게 두 단계로 진행된다. 먼저 자기조직화지도를 이용하여 사용자가 제출한 응용의 실행에 영향을 미치는 특성에 가중치를 부여하여 군집을 형성한다. 그리고 앞선 결과를 이용하여 K-means 군집분석을 진행한다. 이 때, elbow method를 이용하여 적정 군집개수 k를 결정한다.

알고리즘 1은 자기조직화지도[9]를 이용한 모델 학습 알고리즘을 나타낸다. 알고리즘을 시작하기에 앞서, 자기조직화지도[9]를 수행하기 위해서는 출력 뉴런(output neuron)의 지도(map) 크기를 설정해야 한다. 또한 학습 횟수(Iter)와 응용 특성 가중치 우선순위(α)과 학습률(β)을 설정한다.

Algorithm 1. training model reflecting application characteristic weight using self-organizing map

t = current iteration

Iter = max iteration number

$X_i(t)$ = i th input neuron at iteration t

$W_{ij}(t)$ = weight between i th input neuron and j th output neuron

1) t <- 0

2) $W_{ij}(0)$ <- random number, for all i and j

3) Input neuron <- input vector

4) **While**(t != Iter)

5) **Calculate** d_j for all output neuron

$$d_j = \sum_{i=0}^{N-1} k_i (X_i(t) - W_{ij}(t))^2, k_i = \begin{cases} 1, & i \neq characteristic \\ \alpha, & i = characteristic \end{cases}$$

6) **Select** j with minimum d_j

7) **Update** j and j' s neighbor neuron' s weight vector

$$W_{ij}(t+1) = W_{ij}(t) + \beta(X_i(t) - W_{ij}(t))$$

8) t++

9) **EndWhile**

현재 반복 횟수 및 모든 연결가중치 벡터(Wij(0))를 초기화 한다(line 1-2). 입력벡터는 자원리스트로 구성되어 정규화 과정을 거친 후, 입력 뉴런으로 제시된다. N은 입력 벡터의 속성 개수이며, 각 속성에 대해 입력뉴런(Xi(t))과 연결 가중치 벡터(Wij(t)) 사이의 거리 제곱 합을 도출한다(line 5). 이 때, ki 는 응용의 특성 (AC)에 대한 가중치로 응용 실행에 영향을 미치는 요소에 가중치를 곱하여 도출한다. 사용자가 제출한 응용에 가중치를 부여해야 하는 요소가 여러 개일 경우, 이는 우선순위(α)에 따라 가중치를 다르게 부여할 수 있다. 도출된 거리 제곱 합 중 가장 거리가 짧은 승자뉴런을 선택한다(line 6). 그 승자뉴런과 이웃한 출력 뉴런에 연결된 가중치들을 갱신한다(line 7). 학습률(β)은 0과 1 사이의 값을 가지며 연결될 가중치를 갱신하는 정도를 결정한다.

알고리즘1의 학습된 결과 모델을 이용하여 K-평균 군집분석[11]을 진행한다. 학습된 결과 모델은 출력 뉴런이 각각 연결가중치를 갖는다. 이를 K-평균 군집분석[11]의 입력 값으로 이용하다. 이 때, 군집의 개수를 추가하며 elbow method[10]를 수행하여 적정 군집개수 K를 선정한다. Elbow method[10]는 군집의 개수에 따라 각각 군집 중심과 해당 군집에 속한 요소간의 거리의 제곱 합(wcss; within cluster sum of squares)을 계산한다. 군집 개수가 추가되더라도 wcss 값이 크게 감소하지 않은 경우, 이전의 군집의 개수가 적정 군집 개수로 선택된다.

4. 실 험

본 연구에서 알고리즘1. 자기조직화지도를 이용한 응용 특성 가중치 반영 모델 학습을 진행 시, 64개의 출력 뉴런(8*8)을 설정하였으며, 응용의 특성에 대한 가중치는 2로 설정하였다. 또한 출력뉴런에 연결된 가중치를 갱신 시 사용되는 학습률은 0.05로 설정하고, 0.01까지 선형적으로 감소한다. 지정된 학습 횟수는 500번으로 설정하였다. Elbow method 기반 K-평균 군집 분석을 이용한 군집 도출 부분에서 이전 군집개수와 wcss 값 차이가 10미만은 경우를 적정 군집 개수로 선정하였다.

본 연구에서는 사설 클라우드인 Openstack[12]과 공용 클라우드 제공 회사(Amazon EC2[1], Microsoft Window Azure[2], KT Ucloud[3], Google Cloud Engine [4], NHN Entertainment[5])의 가상 자원에 대해 자원 특성 정의를 적용하여 명세하였다. Openstack의 5개 자원, Amazon EC2의 24개 자원, Microsoft Window Azure의 25개 자원, KT Ucloud의 18개 자원, Google Cloud Engine의 12개 자원, NHN Entertainment의 13개 자원으로 총 97개의 클라우드 가상 자원을 명세하였다. 97개의 클라우드 자원들을 3.3 응용 특성을 반영한 자원 군집 분석에 따라 분류했다. 각 출력 뉴런은 응용의 특성을 반영함에 따라 다른 값을 갖게 되어, 해당 특성에 따라 각각 다른 구성의 클라우드 자원 군집이 형성된다.

제안하는 기법의 우수성을 확인하기 위하여 항공우주 분야의 비정상 유동해석 역할 시뮬레이션(CFD[13])와 천문학 분야의 Mosaic 이미지 생성 엔진(Montage[14])를 대상으로 실험을 진행했다. CFD의 경우, 응용 수행을 위해 격자의 크기, target time, 반복횟수를 설정할 수 있다. 본 실험에서 2KB의 격자를 사용했으며, 입력 파일에서 Target time을 0.5, 최대 반복횟수를 1000001로 설정했다. 또한 CPU 집약적인 응용으로 CPU를 응용 특성으로 설정하였다. Montage의 경우, shrink 단계에서 planes의 크기(15, 10, 5)에 따라 중간/최종 생성데이터의 크기가 달라진다. 본 실험에서는 15 planes를 사용하여 실험을 진행하였다. 또한 메모리 집약적인 응용으로 메모리를 응용의 특성으로 설정하였다.

4.1 응용 특성 가중치를 부여한 군집 분석 결과

그림 1은 자원의 특성에 가중치를 부여하지 않고 군집을 형성한 결과를 나타낸다. 그림 2는 계산집약적인 CFD 응용에 대해 군집을 형성하였으며 이는 vCPU에 가중치를 부여 하고 군집을 형성하였다. 그림 3은 데이터 집약적인 Montage 응용에 대해 군집을 형성하였으며 이는 memory에 가중치를 부여하고 군집을 형성하였다. 각각의 자기조직화지도 훈련에 의해 형성된 경쟁층은 다른 출력 벡터를 가지게 된다. 따라서 이에 따라 K- 평균 군집분석 기법을 사용 하여 군집을 형성하였으므로 각기 다른 개수의 군집을 형성하였다. 가중치를 부여하지 않은 경우는 8개의 군집을, CFD 응용을 대상으로 군집을 형성한 경우는 6개의 군집을, Montage 응용을 대상으로 군집을 형성한 경우는 7개의 군집을 형성하였다. Amazon EC2의 r3.2xlarge 자원을 예를 들어, 이는 가중치를 부여하지 않은 경우, {m4.4xlarge, c4.4xlarge, c3.4xlarge, F16}와 하나의 군집으로 형성되었다. 또한 CPU에 가중치를 부여하는 경우, {m4.2xlarge, m4.4xlarge, m3.2xlarge, c4.4xlarge, c3.4xlarge, r3.2xlarge, F16}와 하나의 군집을 형성하였으며, memory에 가중치를 부여한 경우, {m4.2xlarge, m3.2xlarge, r3.2xlarge, F8, High-memory6}와 하나의 군집을 형성하였다. 응용의 특징에 따라 형성되는 자원의 군집이 다르며, 이는 어떠한 요소에 가중치를 부여하는지에 따라 동적으로 군집이 형성된다. 군집분석은 응용의 특성을 반영하여 유사한 속성을 가진 자원을 군집으로 형성하여 결과를 도출한다.

4.2 군집 내의 실행시간 유사성 비교

두 가지 응용(CFD, Montage)의 군집 내 실행시간 유사성 비교를 위해 Amazon EC2[1]의 3개 자원(m4.xlarge, m4.2xlarge, m3.2xlarge)과 Microsoft Window Azure[2]의 2개 자원(A3, F8), Openstack[14]의 2개 자원(m1.medium, m1.large)에서 CFD 응용과 Montage 응용을 반복 수행하여 평균 실행시간을 측정하였다.

CFD 응용의 경우, 그림 4와 같이 응용 특성을 반영한 자원 군집분석 결과에 따라 실제 수행한 위의 자원들은 3개의 군집으로 분류된다. 군집2는 A3, m4.xlarge 자원이 속하며 각 자원의 평균 실행시간은 140초, 153초이다. 군집 2의 평균 실행시간은 146.5초이며, 최소값은 140초, 최대값은 153초이다. 군집 3은 m1.large, m1.medium 자원으로 구성되었으며 각 자원의 평균 실행시간은 85초, 122초이다. 군집 3의 평균 실행시간은 103.5초이며, 최대값 및 최소값은 각각 122초, 85초이다. 군집 5는 m3.2xlarge, F8, m4.2xlarge 자원으로 구성되어 있으며, 각 자원의 평균 실행시간은 45초, 76초, 70초이다. 군집 5의 경우는 평균 실행시간이 63.67초이며 최소값 및 최대값은 45초, 76초이다. 같은 군집에 속해 있는 자원들은 유사한 실행시간을 나타내고, 각 군집의 평균 실행시간이 달라 유사한 자원끼리 군집을 형성한 것으

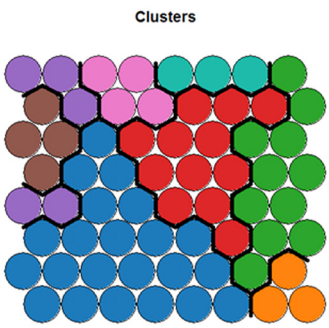

그림 1 가중치를 부여하지 않고 형성된 군집 결과

Fig. 1 Result of the clusters without assigning the weight

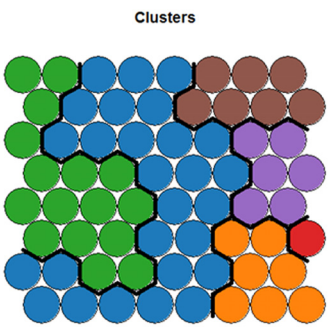

그림 2 CFD 응용에 대해 형성된 자원의 군집 결과

Fig. 2 Result of the clusters for the CFD application

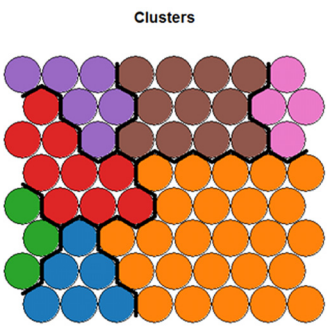

그림 3 Montage 응용에 대해 형성된 자원의 군집 결과

Fig. 3 Result of the clusters for the Montage application

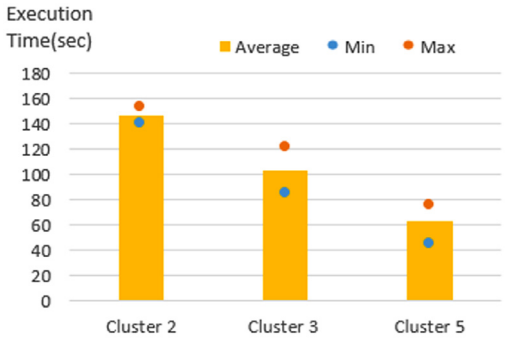

그림 4 CFD 응용의 군집 별 실행시간 측정 결과
Fig. 4 Result for the execution time of the CFD application

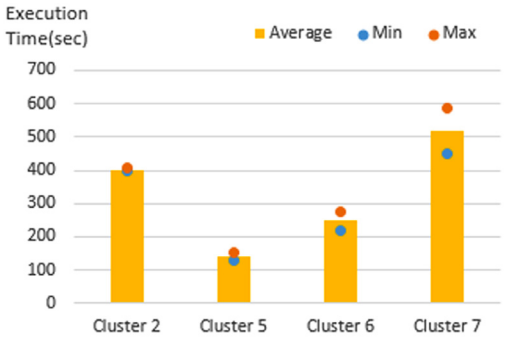

그림 5 Montage 응용의 군집 별 실행시간 측정 결과
Fig. 5 Result for the execution time of the Montage application

로 볼 수 있다.

Montage 응용의 경우, 그림 5와 같이 응용 특성을 반영한 자원 군집분석 결과에 따라 실제 수행한 자원들은 4개의 군집으로 분류된다. 군집 2의 평균 실행시간은 401초이며 최소값 및 최대값은 398초, 406초이다. 이는 A3 자원을 5번 반복 수행하여 얻은 결과이다. 군집 5는 m3.2xlarge, m4.2xlarge로 구성되어 있으며, 각 자원의 평균 실행시간은 130초, 153초이다. 군집 5의 평균 실행시간은 141.5초이며 최대값은 153초, 최소값은 130초이다. 군집 6은 m4.xlarge, F8자원으로 구성되어 있으며, 각 자원의 평균 실행시간은 220초, 275초이다. 군집 6의 평균 실행시간은 247.5초이며 최소값 및 최대값은 각각 220초, 275초이다. 군집 7은 m1.large, m1.medium으로 구성되어 있으며 각 자원의 평균 실행시간은 450초, 586초 이다. 군집 7의 평균 실행시간은 518초이며, 최소값은 450초, 최대값은 586초이다. 해당 실험에서도 같은 군집 내의 자원의 실행시간 측정을 통해 유사한 자원끼리 군집을 형성한 것을 확인할 수 있다.

5. 결론 및 향후 연구

본 연구에서는 하이브리드 클라우드 환경에서 응용의 특성을 반영한 자원 군집 분석 기법을 제안한다. 자원 군집 분석은 자기조직화지도 및 K-평균 알고리즘을 적용하여 유사한 자원을 군집화한다.

제안하는 알고리즘을 기반으로 실제 다양한 클라우드 서비스 제공자의 자원을 이용해 응용을 수행하고 군집 내의 자원들의 실행시간은 유사함을 보였다. 향후 연구로는 클라우드 자원 환경을 추가, 확장하고 제안하는 알고리즘을 보완하여 적용할 것이다.

References

[1] Amazon EC2, [Online]. Available: http://aws.amazon.com/
[2] Microsoft Window Azure, [Online]. Available: http://www.windowsazure. com/
[3] KT Ucloud, [Online]. Available: http://ucloudbiz.olleh.com/
[4] Google Cloud Engine, [Online]. Available: https://cloud.google.com/
[5] NHN Entertainment, [Online]. Available: http://cloud.toast.com/
[6] Chavan, Vinay, and Parag Ravikant Kaveri, "Clustered virtual machines for higher availability of resources with improved scalability in cloud computing," *Networks & Soft Computing (ICNSC), 2014 First International Conference on. IEEE*, 2014.
[7] Wu, Qingtao, et al., "A cloud service resource classification strategy based on feature similarity," *Journal of Networks 9.11*, 2987-2993, 2014.
[8] Ahn, Younsun, and Yoonhee Kim, "Semantic Cloud Resource Recommendation Using Cluster Analysis in Hybrid Cloud Computing Environment," *KIPS Transactions on Computer and Communication Systems*, Vol. 4, No. 9, 283-288, 2015.
[9] Self-Organizing Map, Kohonen, Teuvo, and Timo Honkela, "Kohonen network," Scholarpedia 2.1 1568, 2007.
[10] Elbow method, [Online]. Available: https://en.wikipedia.org/wiki/Elbow_method_(clustering)
[11] K-means clustering, MacQueen, James, "Some methods for classification and analysis of multivariate observations," *Proc. of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1, No. 14, 1967.
[12] OpenStack, [Online]. Available: https://www.openstack.org/.
[13] CFD(Computational fluid dynamics), Seoul National University, Aerodynamic Simulation & Design Lab.

[14] Montage GALFA, [Online]. Available: http://montage.ipac.caltech.edu/docs/cubemosaicstutorial.html

오 유 리

2015년 숙명여자대학교 수학과 졸업(학사). 2017년 숙명여자대학교 컴퓨터과학과 졸업(석사). 2017년~현재 한국특허정보원 재직. 관심분야는 클라우드 컴퓨팅, 지능형 시스템

김 윤 희

1991년 숙명여자대학교 전산학과 졸업(학사). 1996년 Syracuse University 전산학과 졸업(석사). 2000년 Syracuse University 전산학과 졸업(박사). 1991년~1994년 한국전자통신연구원 연구원 2000년~2001년 Rochester Institute of Technology 컴퓨터공학과 조교수. 2001년~2016년 숙명여자대학교 컴퓨터과학부 교수. 2017년~현재 숙명여자대학교 소프트웨어학부 교수. 관심분야는 클라우드 컴퓨팅, 워크로우 제어, 그리드/클라우드 관리