

碩 士 學 位 論 文

응용 특성을 고려한 동적 군집 분석 기반
자원 추천 기법

A Resource Recommendation Method Based on
Dynamic Cluster Analysis for Application Characteristics

淑明女子大學校 大學院

컴퓨터科學科 컴퓨터科學 專攻

吳 裕 利

碩 士 學 位 論 文

응용 특성을 고려한 동적 군집 분석 기반
자원 추천 기법

A Resource Recommendation Method Based on
Dynamic Cluster Analysis for Application Characteristics

淑明女子大學校 大學院

컴퓨터科學科 컴퓨터科學 專攻

吳 裕 利

碩 士 學 位 論 文

응용 특성을 고려한 동적 군집 분석 기반
자원 추천 기법

A Resource Recommendation Method Based on
Dynamic Cluster Analysis for Application Characteristics

指導教授 金 倫 希

이 論文을 理學 碩士 學位 論文으로 提出함

2016 年 12 月

淑明女子大學校 大學院

컴퓨터科學科 컴퓨터科學 專攻

吳 裕 利

吳裕利의 理學 碩士 學位 請求 論文을 認准함

응용 특성을 고려한 동적 군집 분석 기반
자원 추천 기법

A Resource Recommendation Method Based on
Dynamic Cluster Analysis for Application Characteristics

2016 年 12 月

審査委員長 教授 (印)

委 員 教授 (印)

委 員 教授 (印)

淑明女子大學校 大學院

목 차

목 차.....	i
표 목 차.....	iv
그림목차.....	v
국문요약.....	vii
I. 서 론.....	1
1. 연구 배경.....	1
2. 연구의 목표 및 내용.....	4
3. 논문의 구성.....	5
II. 관련 연구.....	6
1. 클라우드의 자원 명세 방법.....	6
2. 클라우드 자원 분류 기법.....	10
3. 클라우드 자원 추천 기법.....	11
4. 관련 연구들과 본 연구와의 비교.....	13

III. 자원 군집 분석 기반 추천 서비스.....	14
1. 자원 특성 명세.....	14
2. 자원 군집 분석 기반 추천 서비스 구조.....	16
3. 군집 분석.....	18
IV. 자원 군집 분석 기반 추천 알고리즘.....	22
1. 응용 특성을 반영한 자원 군집 분석.....	22
1) 자기조직화지도를 이용한 응용 특성 가중치 반영 모델 학습	23
2) Elbow method기반 K-평균 군집 분석을 이용한 군집 도출	25
2. 비용 효율적인 자원 추천 기법	28
V. 실험 및 결과.....	29
1. 대상 응용 특성	29
1) 항공우주 분야 : 비정상 유동해석 역학 시뮬레이션.....	29
2) 천문학 분야 : Mosaic 이미지 생성 엔진.....	30
2. 자원 군집분석 결과 분석.....	32
3. 비용효율적인 자원 추천 결과 분석.....	37
4. 군집화 성능 비교 분석.....	38
1) 군집 내의 실행시간 유사성 비교.....	39
2) 군집 내의 실행시간 및 가격 비교.....	40
3) 타 군집분석 기법과의 성능 비교....	43

VI. 결 론.....	46
참 고 문 헌.....	47
ABSTRACT.....	52

표 목 차

표 1.	과학 응용 분류 기준에 따른 응용 분류	2
표 2.	관련 연구의 논문과 본 연구와의 비교	13
표 3.	자원 특성 명세를 적용한 자원리스트 일부	15
표 4.	응용 특성 가중치 반영 모델 학습의 입력 뉴런의 일부	33
표 5.	응용 특성을 고려하지 않은 군집 분석 결과 일부	36
표 6.	CFD 응용의 특성을 반영한 군집 분석 결과 일부	36
표 7.	Montage 응용 특성을 반영한 군집 분석 결과 일부	36
표 8.	군집내의 비용효율적인 자원 추천 결과	37

그림 목 차

그림 1.	IEEE P2302: 인터클라우드 구조[8]	8
그림 2.	mOSAIC 온톨로지의 최상위 개념[9]	9
그림 3.	자원 군집 분석 기반 추천 서비스 구조	17
그림 4.	자기조직화지도 구조[26]	19
그림 5.	응용 특성을 반영한 자원 군집 분석 알고리즘	21
그림 6.	자기조직화지도를 이용한 응용 특성 가중치 반영 모델 학습 알고리즘	23
그림 7.	K-평균 군집분석 알고리즘	26
그림 8.	Shock tube 문제[29]	30
그림 9.	Mosaic 이미지 생성 엔진 수행 결과[31]	31
그림 10.	응용 특성을 고려하지 않은 군집 결과	34
그림 11.	CFD 응용 특성을 고려한 군집 결과	35
그림 12.	Montage 응용 특성을 고려한 군집 결과	35
그림 13.	CFD 응용의 군집 별 실행시간 측정 결과	39
그림 14.	Montage 응용의 군집 별 실행시간 측정 결과	40
그림 15.	CFD 응용 특성을 반영한 군집의 실행시간 및 가격 비교	41

그림 16.	Montage 응용 특성을 반영한 군집의 실행시간 및 가격 비교	42
그림 17.	CFD 응용 특성을 반영한 선행연구의 표준편차 비교	43
그림 18.	Montage 응용 특성을 반영한 선행연구의 표준편차 비교 ..	44

국 문 요 약

클라우드 컴퓨팅 기술의 발전으로 과학 응용을 수행하는 과학자들은 원하는 클라우드 자원을 필요한 만큼만 사용하고 지불하는(Pay-per-use) 방식을 이용하여 응용을 수행하고자 하는 과학자들이 늘어나고 있다. 그러나 다양한 특성으로 구성된 자원으로 과학자들은 적절한 자원을 선택하는데 어려움을 겪는다. 이에 따라 자원의 효율적인 활용을 위하여 분류가 필요하다. 한편, 자원에 대한 정적인 분류는 부정확한 결과를 도출할 수 있으며 과학자들은 다양한 실험 의도 및 요구사항을 제출하기 때문에 동적으로 자원의 분류를 변화시킬 필요가 있다. 따라서 과학자의 요구사항 및 응용 특성을 반영한 동적인 자원 분류가 필요하다. 또한 비용 효율적인 자원 추천 서비스가 필요하다.

본 연구에서는 하이브리드 클라우드 환경에서 응용의 특성을 반영한 자원 군집 분석 기법 및 비용 효율적인 자원 추천 기법을 제안한다. 자원 군집 분석은 자기조직화지도 및 K-평균 알고리즘을 적용하여 유사한 자원을 군집화한다. 또한 응용 특성을 고려한 자원 군집을 기반으로 효율성 척도를 적용하여 비용 효율적인 자원을 추천기법을 제안한다. 타 연구의 자원 분류 기법을 적용한 경우와 비교하여 제안하는 동적 자원 분류 기법의 우수성을 보인다. 제안한 알고리즘을 통해 과학 응용의 특성을 반영한 유사 자원 군집 형성 및 비용 효율적인 자원 추천이 가능하다.

주제어: 하이브리드 클라우드, 자기조직화지도, 군집분석, 동적 자원 분류

I. 서론

1. 연구 배경

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 [11], Microsoft Window Azure [12], KT Ucloud [13], Google Cloud Engine [14], NHN Entertainment [15] 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.

과학 응용은 런타임 분석 도구를 사용하여 I/O, memory, CPU와 같은 자원 사용량에 대한 작업 특성을 분석하고 해당 분석에 따라 분류할 수 있다. 분류 시, 작업 형태, 자원 사용 패턴, I/O 패턴, 메모리 접근 패턴에 따른 응용을 분류할 수 있으며 각 패턴에 따른 일반적인 응용 분류는 표 1과 같다. 응용의 작업 형태에 따라, Bag of Tasks, Workflow 응용으로 나뉠 수 있다. Bag of Tasks 응용은 task의 순서에 관계없이 정해진 작업을 수행한다. Workflow 응용은 task의 실행 순서가 정해져 있으며, 앞선 task의 실행이 끝나지 않으면 다음 task의 수행이 불가하다. 자원의 사용 패턴에 따라, CPU-intensive, I/O & Paging Intensive,

Network Intensive, Memory-intensive 응용으로 나뉠 수 있다. 또한 메모리에 접근하는 패턴에 따라 Sequential, Strided, Linear, Nearest neighbor 등으로 분류될 수 있다. I/O의 발생 빈도에 따라, I/O가 증가하다가 감소하는 형태의 Bell-shaped, 어느 순간 급증하는 Bursty, 일정한 I/O가 발생하는 Constant로 분류할 수 있다. 본 연구에서 언급하는 응용의 특성이란, 자원 사용 패턴에 따라 분류되는 응용 특성을 의미한다. 분류된 응용 특성에 따라 해당 응용 실행에 영향을 미치는 자원의 요소가 달라진다.

<표 1> 과학 응용 분류 기준에 따른 응용 분류

분류 기준	응용 분류
작업 형태[32]	Bag of Tasks, Workflow
자원 사용 패턴 [33]	CPU-intensive, I/O & Paging Intensive, Network Intensive, Memory-intensive
메모리 접근 패턴 [34]	Sequential, Strided, Linear, Nearest neighbor, 2D Spatially coherent, Scatter, Gather, Combined gather and scatter, Random
I/O 접근 패턴	Bell-shaped[35], Bursty[36], Constant[37]

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. 또한 자원

추천은 응용 실행에 영향을 크게 미치는 요소가 아닌 다른 요소로 인해 가격이 증가한 자원이 제외되어야 한다. 이에 따라, 비용 효율적인 자원을 추천하는 기법이 필요하다.

하이브리드 클라우드 환경에서 다양한 영역의 표준화를 위해 인터클라우드 프로젝트가 진행되고 있다[8]. mOSAIC 프로젝트[9]는 온톨로지를 사용하여 클라우드 자원을 명세한다. 이는 클라우드 서비스 제공자 사이의 자원에 대한 공통된 명세로 상호호환성을 제공해준다. 그러나 이는 실제 클라우드 서비스 제공자들이 제공하는 클라우드 자원의 특성을 모두 명세하지는 않았다.

다양한 클라우드 서비스 제공자들로 인해 클라우드 자원에 대한 분류 및 추천에 대한 연구가 진행되고 있다[17, 18, 19, 20, 21, 22]. 기계 학습 또는 통계적인 기법을 통해 자원을 분류한다. 또한 사용자의 요구사항에 알맞은 자원을 추천해준다. 본 연구와 유사한 연구인 연구 [20]은 통계적인 기법을 활용하여 응용 특성을 반영한 동적인 군집분석을 제안한다. 또한 대표자원으로 추천 서비스를 제공할 수 있다고 언급한다. 그러나 해당 연구에서 활용한 통계적인 기법은 클라우드 자원에 대한 유사도를 파악하는데 부족한 점이 있으며, 구체적인 추천 방법 및 결과를 제시하지 않는다. 따라서, 응용 특성을 반영한 유사 자원의 군집 형성을 기반으로 비용 효율적인 자원을 추천하는 기법이 필요하다.

2. 연구의 목표 및 내용

본 논문에서는 하이브리드 클라우드 환경에서 자원을 군집화하고 비용 효율적인 자원 추천 기법을 제안한다. 제안하는 알고리즘을 통해 응용의 특성을 반영한 유사 자원의 군집을 형성하며, 이를 기반으로 효율적인 자원 선택이 가능함을 보인다. 본 논문의 주요 연구 내용 및 방법은 다음과 같다.

첫째, 기존 자원 특성 명세를 확장하고 다양한 클라우드 서비스 제공자의 자원을 명세하였다.

둘째, 하이브리드 클라우드 환경에서 자기조직화지도 및 K-평균 알고리즘을 기반으로 응용의 특성을 고려하여 유사 자원 군집 분석 기법을 제시한다.

셋째, 응용의 특성에 가중치를 부여한 유사 자원 군집을 기반으로 해당 응용의 비용 효율적인 자원 추천 기법을 제시한다.

넷째, 제안하는 알고리즘에 대한 실험을 통해 성능을 검증한다.

3. 논문의 구성

본 논문의 구성은 다음과 같다. 2장에서는 클라우드 자원 명세 방법에 대한 연구 및 관련 표준화와 다양한 기법을 이용한 자원 분류 기법, 자원 추천 기법에 대한 관련 연구를 소개하고 본 연구와 관련 연구들을 비교 분석한다. 3장에서는 하이브리드 클라우드 컴퓨팅 환경에서의 자원 특성을 명세하고, 자원 군집 분석 기반 추천 서비스 구조 및 군집 분석에 대해 소개한다. 4장에서는 응용 특성을 반영한 자원 군집분석 및 비용 효율적인 자원 추천 기법에 대해 소개한다. 5장에서는 제안하는 알고리즘의 성능을 평가하기 위해 실험을 진행하고 그 결과를 분석한다. 마지막으로 6장에서는 본 연구의 결론을 맺는다.

II. 관련 연구

본 장에서는 클라우드 자원 명세 방법에 대한 연구 및 관련 표준화에 대해 설명한다. 또한 다양한 기법을 이용한 자원 분류 기법 및 자원 추천 기법에 대한 관련 연구를 소개하고, 마지막으로 본 연구와 관련 연구들을 비교 분석한다.

1. 클라우드의 자원 명세 방법

클라우드 컴퓨팅은 IT 비용 절감 효과 및 자원 이용의 유용성 등의 장점을 중심으로 각광받는 서비스로 부상하였다. 클라우드의 활성화에 따라 다양한 용어가 등장했다. 클라우드 서비스 운용 형태에 따라 공용, 사설, 하이브리드 클라우드로 구분된다. 공용 클라우드는 불특정 다수를 대상으로 하는 서비스로 여러 서비스 사용자가 이용하는 형태이다. 사설 클라우드는 기업 및 기관 내부에 클라우드 서비스 환경을 구성하여 내부자에게 제한적으로 서비스를 제공하는 형태이다. 하이브리드 클라우드는 공용 클라우드와 사설 클라우드를 결합한 형태이다. 공유를 원하지 않은 일부 데이터 및 서비스에 대해 프라이빗 정책을 설정하여 서비스를 제공한다[1].

하이브리드 클라우드 환경에서 수 많은 클라우드 서비스 제공자들로 인해 자원에 대한 서로 다른 명세를 해결하고자, 다양한 연구가 진행되고 있다. Yan[2]는 IaaS를 사용하는 것이 하이브리드 클라우드 환경에서 사용자가 대부분 관리의 복잡성을 직면한다고 언급한다. 이에 해당 연구

에서는 Monsoon 클라우드 관리 시스템을 제안하며, 이는 다양한 공용 클라우드 및 사설 클라우드 제공자로부터 클라우드 인프라를 관리하는 포털과 인터페이스를 제공한다. 제안하는 시스템의 자원 명세부분은 Compute, Storage, Network에 대한 것이 전부이다. 이는 메모리 및 클라우드 서비스 제공자 등의 정보가 필요하여, 실제 자원에 대한 명세로는 한계가 있다.

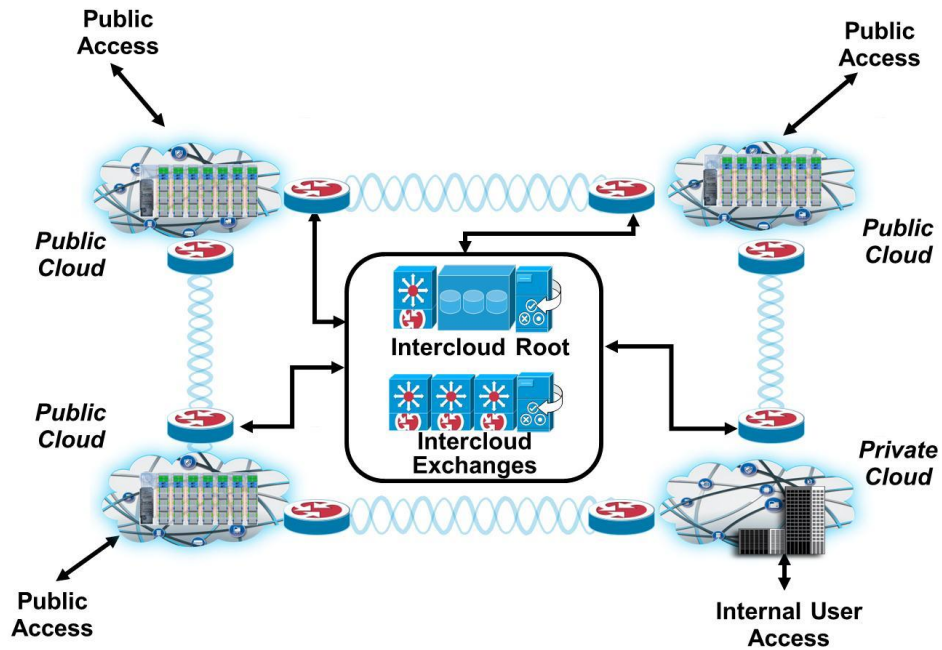
Di Martino[3]은 다양한 프로젝트 및 조직이 클라우드 분야에서 표준을 지정하는데 역할을 하고 있지만, 클라우드 자원에 대한 확장성 있는 명세가 고려되고 있지 않은 점을 언급했다. 해당 연구에서는 초기 인터넷을 개발할 때 정의된 확장성 있는 구조 및 클라우드 온톨로지를 정의할 시멘틱 웹의 개념을 사용한다. 그러나 명확한 명세에 대해 제시하지 않고, 명세에 대한 설계 및 개념을 증명하는 정도에만 그친 수준이다.

하이브리드 클라우드에 대한 서비스 확산이 가속화되고 있는 반면 아직도 표준화된 시스템을 갖추어 도입하는데 어려움이 존재하며, 관련 시스템 및 서비스 구축, 제공 시 특정 기술 및 서비스 종속 심화도 우려되고 있다[4]. 인터클라우드 표준이란 다양한 클라우드 서비스 제공자의 클라우드를 연결/연계하여 복수 서비스 제공자 클라우드간의 자원과 서비스 연동을 위한 표준을 말한다. 인터클라우드 표준은 하이브리드 클라우드의 유연한 사용을 가능하도록 한다. 이는 ‘클라우드의 클라우드’란 뜻으로 다양한 클라우드 자원과 서비스를 확장하여 사용하는 것을 의미한다. 사용자에게 하나의 클라우드 환경이 되기 위해서는 둘 이상의 클라우드 서비스 제공자 간의 상호호환성이 있어야 한다[5].

Nodehi[6]은 IaaS 클라우드 서비스를 위한 통합된 인터클라우드 모델을 제안한다. 이는 다른 클라우드간의 동적인 스케줄링이 가능하도록 하는 모델을 제시했다. SLA(Service Level Agreement) 및 Qos(Quality

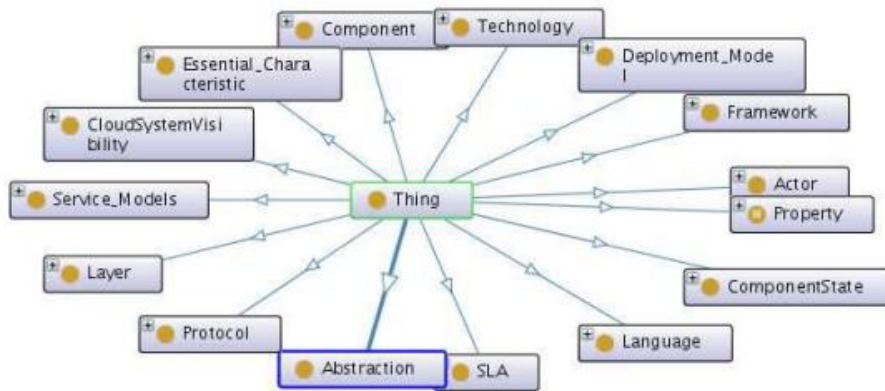
of Service)를 만족하기 위한 스케줄링에 초점을 맞춰 연구를 진행하였고, 클라우드 자원 특성에 대한 명세는 CPU, Storage, Memory 정도로 되어있다. 실제 클라우드 서비스 제공자가 제공하는 자원 특성에 대해서는 추가적인 정의가 필요하다.

Grozev[7]은 인터클라우드 용어에 대해 분류한다. 인터클라우드는 다양한 클라우드를 사용함으로써 전반적으로 더 나은 QoS, 신뢰성, 비용 효율성 등의 이득을 얻는 것을 목표로 한다. 해당 연구에서는 학술적, 산업적으로 개발된 20개 클라우드 연계 프로젝트에 대해 소개한다. 그중 인터클라우드 프로젝트[8]는 핵심이 되는 프로젝트이다. 그림 1은 인터클라우드 프로젝트[8]에서 제안한 인터클라우드의 구조이다. 인터클라우드 루트(Intercloud Root), 교환국(Exchange), 클라우드, 게이트웨이로 구성되어 있으며 다양한 클라우드 자원을 제약 없이 사용할 수 있다.



[그림 1] IEEE P2302: 인터클라우드 구조[8]

클라우드 자원의 통합된 명세를 위해 인터클라우드 프로젝트에서 표준화 프로젝트를 진행하고 있다. 이에 많은 연구들은 인터클라우드 프로젝트의 표준화 프로젝트의 내용을 활용하여 연구를 진행하고 있다. 인터클라우드 표준화 프로젝트 중, mOSAIC 프로젝트[9]는 온톨로지를 정의하여 클라우드 서비스에 대한 통합된 접근 방식을 제안하는 것을 목표로 한다. 해당 프로젝트는 다양한 클라우드 제공자가 사용하는 용어, 다른 용어로 불리는 클라우드 시스템에 대한 표준에 대해 mOSAIC 온톨로지를 정의한다.



[그림 2] mOSAIC 온톨로지의 최상위 개념[9]

그림 2는 mOSAIC 온톨로지의 최상위 개념을 보여주며 주요 개념들을 나타내고 있다. Component 클래스는 mOSAIC 온톨로지의 메인 클래스이며, 모든 클라우드 요소(자원, 서비스, 인프라 등)를 서브클래스로 갖는다. Property 서브클래스는 클라우드 자원의 특성을 명세할 때 필요한 모든 요소들을 포함한다. Computing Non Functional Property는 CPU와 Memory과 관련된 속성으로 나뉜다. 이 속성들은 OCCI[10] 표준을 따른다. mOSAIC 프로젝트[9]는 mOSAIC 온톨로지를 통해 클라우드 자원의 명세를 표준화하고 상호호환성을 제공한다. 그러나 클라우드 서비스 제공자들이 제공하는 자원의 특성을 모두 표현하기에는 부족하다.

클라우드 자원에 대해 다양한 명세 방법이 있으나 각각 개별적인 명세로 인해 내용을 통합하여 파악하는데 어려움이 있다. 이에 자원에 대해 총체적으로 파악하는데 효율적인 명세가 필요하다.

2. 클라우드 자원 분류 기법

다양한 클라우드 컴퓨팅 서비스 제공 회사(Amazon EC2 [11], Microsoft Window Azure [12], KT Ucloud [13], Google Cloud Engine [14], NHN Entertainment [15])는 각 회사의 방침대로 수 많은 서로 다른 성능의 가상 자원을 제공한다. 이에 따라 클라우드 자원을 분류하는 기법에 대한 연구들이 진행되고 있다.

다양한 속성 및 성능을 가진 클라우드 자원의 편리한 관리를 위해 자원의 요소를 고려하여 요약하거나 분류하는 것이 필요하다. 기계 학습 및 통계적인 기법을 사용하여 다량의 데이터를 요약 및 분류할 수 있다. 이에 따라 자원에 대해 군집화를 적용하는 분류에 대한 연구가 진행 중이다[17, 18, 19, 20].

Chavan [17]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. 형성된 군집을 이용하여 향상된 확장성으로 자원에 대해 고가용성을 보장한다.

Whaiduzzaman [18]는 클라우드 구매자가 각자의 필요에 따라 적절한 서비스를 선택할 수 있어야 하며, 다양한 접근법이 제안되었다고 언급하고 있다. 그 중 MCDA(Multicriteria Decision Analysis)는 사용자가 이용 가능한 클라우드 서비스들 중 선택이 가능하게 하는 것을 의미한다. 해당 연구에서는 MCDA를 카테고리, 유형, 방법 등으로 분류하고 각각에

대해 수행 방법을 설명하고 있다.

Wu [19]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 분류했다. 또한 분류 시, 병렬화로 효율성을 높이는데 목표를 두고 있다.

안윤선 [20]은 제안하는 연구와 유사한 연구로 인터클라우드 환경에서 응용의 특성을 고려하여 자원을 동적으로 분류하는 기법을 제안한다. 이는 통계적인 기법인 k-modes 알고리즘을 이용하여 응용의 특성에 따라 가중치를 부여하고 군집을 형성한다. 하지만 k-modes 알고리즘은 해당 특성의 값이 같은 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 부족한 점이 있다.

3. 클라우드 자원 추천 기법

클라우드 서비스 제공 회사는 클라우드 자원의 각 속성에 대해 정형된 성능을 고정하여 가상 자원을 제공한다. KT ucloud [13] 및 Google Cloud Engine [14]의 경우, 자원의 이름에 highmem, highcpu, High-memory 와 같이 자원의 특성을 나타내기도 한다. 그러나 각 자원이 정확하게 어떠한 특성을 얼마나 제공하는지는 직접 홈페이지의 설명을 확인하고 다른 회사의 서비스와 비교하는 과정이 필요하다. 다양한 클라우드 서비스에

대해 클라우드 서비스 사용자는 자신이 실행하고자 하는 응용에 적합한 가상 자원을 선택하는데 어려움을 겪을 수 있다. 따라서, 사용자에게 적절한 클라우드를 추천하는 기법에 대한 연구들이 진행 중이다.

Jung [21]은 사용자들은 자신이 수행할 워크로드에 대해 비용 및 성능 예측이 어렵다는 점을 언급하고 있다. 이에 해당 연구에서는 CloudAdvisor로 부르는 클라우드 추천 플랫폼을 제안한다. 제안하는 플랫폼은 사용자의 워크로드에 대해 예산, 예상 성능, 에너지 효율과 같은 선호를 기반으로 클라우드 자원을 추천한다. 그리고 클라우드 사용자가 제안된 금액과 성능을 다른 클라우드 제안과 비교할 수 있다. 하지만 가상자원 설정을 위하여 CPU, 메모리, 디스크, 네트워크 용량의 속성에 대해 VMType을 small, medium, large의 3가지 값 중에 선택하므로 가상자원의 속성에 대해 다양한 고려를 하지 못한다.

Yan [22]은 기업들이 자신의 사업 요구사항에 대해 적절한 클라우드 자원 선택과 회사 방침이 점점 복잡해지고 중요해졌다. 이에 하이브리드 클라우드 관리 플랫폼의 상단에 체계적인 프레임워크를 제안한다. 이는 기업에게 사업 요구사항, 회사 방침, 표준 등에 따라 자동적으로 클라우드 서비스를 추천하고 선택하는 기능을 제공한다.

안윤선[20]은 클라우드 환경에서 응용의 특성을 반영하여 유사 자원을 군집화 한 후, 각 군집에서 대표 자원을 추출한다. 대표자원은 군집 내의 자원의 특성의 평균값에 근거하여 가장 가까운 자원이 선택된다. 대표자원을 활용하여 자원 추천 서비스를 제공할 수 있다고 언급하며 시스템 구조에 대한 제안을 했지만 구체적인 추천 방법이나 결과를 언급하지 않았다.

4. 관련 연구들과 본 연구와의 비교

표 2은 다양한 기법을 사용한 자원 분류 기법에 대한 연구와 본 연구를 비교한 것이다. 각 연구에서 적용한 분류 기법과 응용 특성 고려 군집화에 대해 비교하였다. 또한 자원 추천 여부를 비교하였다. Whaiduzzaman [18]은 자원에 대해 분류 시, 응용의 특성을 반영하지 않고, 자원 추천 서비스를 제공하지 않는다. Wu [19]는 특정 요소에 대한 가중치를 고려하고, 응용에 대한 특성을 고려하지 않는다. 또한 자원에 대한 추천서비스를 제공하지 않는다. 본 연구와 가장 유사한 안윤선[20]는 통계적인 기법을 이용하여 응용의 특성을 고려하여 자원 군집을 형성했다. 그러나 적용한 분류 기법에 비슷한 속성을 반영하기 어려우며, 구체적인 추천 기법이나 결과를 제시하지 않았다. 따라서, 본 연구에서는 자원의 유사한 속성을 반영하여 군집화의 정확도를 높이고, 비용효율적인 자원 추천 기법을 제안한다.

<표 2> 관련 연구의 논문과 본 연구와의 비교

	Whaiduzzaman [18]	Wu [19]	안윤선 [20]	본 연구
분류 기법	다기준 의사결 정방법	베이지안 분 류 알고리즘	k-modes 알고리즘	자기조직화 지도 및 k- 평균 클러스 터링
응용 특성 고려 군집화	X	△	○	○
자원 추천 여부	X	X	△	○

III. 자원 군집 분석 기반 추천 서비스

본 장에서는 하이브리드 클라우드 컴퓨팅 환경에서의 자원 특성을 명세하고, 자원 군집 분석 기반 추천 서비스 및 군집 분석에 대해 소개한다.

1. 자원 특성 명세

클라우드 컴퓨팅 서비스 제공 회사들의 다양한 자원 표기 방식을 통합 관리하기 위하여 자원의 특성을 명세 한다. 연구 [20]에서 활용한 자원 특성 명세를 활용하여 자원 특성을 명세하였다. 연구[20]에서는 mOSAIC 프로젝트[9]의 mOSAIC 온톨로지를 활용 및 확장하여 다양한 클라우드 자원에 대해 일관된 자원의 특성을 명세하였다.

자원 특성 명세를 통해 클라우드 컴퓨팅 서비스 제공회사의 공통된 자원의 특성을 정의하여 서비스 제공 회사에 관계없이 자원의 특성을 이용하여 자원을 선택할 수 있다.

본 연구에서는 사설 클라우드인 Openstack[16]과 공용 클라우드 제공 회사(Amazon EC2 [11], Microsoft Window Azure [12], KT Ucloud [13], Google Cloud Engine [14], NHN Entertainment [15])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 명세하

였다. 연구 [20]의 자원 특성 명세를 기반으로 Drive Type을 추가로 명세하였고, Google Cloud Engine[14], NHN Entertainment[15]의 자원 정보를 추가하였다. 표 3은 6개의 클라우드 자원 제공 회사의 97개의 클라우드 자원의 자원 명세 중 일부를 나타낸다. 표의 첫 행은 명세한 자원 특성들로 구성되어 있다. 클라우드 자원의 특성은 자원 이름(Resource Name), 서비스 제공 회사명(Provider), 메모리 크기(Memory Size, GB), CPU 처리속도(CPU flops), 네트워크 대역폭(Network Bandwidth), CPU(vCPU), 스토리지(Storage, GB), 가격(단위: \$0.01), 드라이브 종류(HDD 또는 SSD)로 구성되어 있다. 두 번째 행부터는 각 클라우드 자원에 대해 해당하는 자원 특성 정보가 추가되어 있다.

<표 3> 자원 특성 명세를 적용한 자원리스트 일부

Resource Name	Provider	Memory size (GB)	Cpu flops	Network Bandwidth	vCPU	Storage(GB)	Cost(단위: \$0.01)	Drive type
N-m2.small	NHN Entertainment	2	2.4	5	1	30	4	HDD
m4.xlarge	Amazon EC2	16	2.4	8	4	30	33.1	HDD
A0	Microsoft Azure	0.75	2.4	2	1	20	2	HDD
A3	Microsoft Azure	7	2.4	8	4	285	20.2	HDD
F8	Microsoft Azure	16	2.4	8	8	128	40.1	SSD

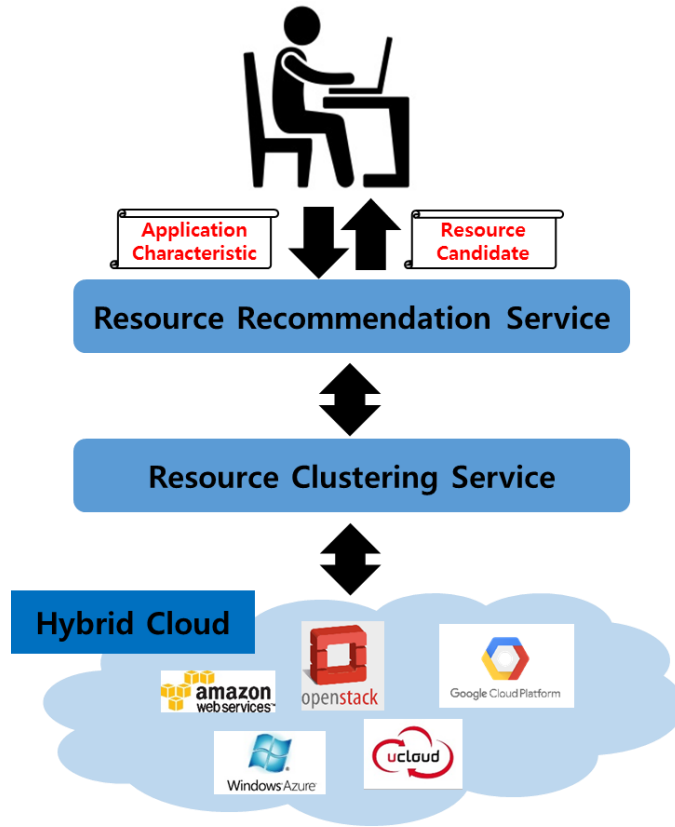
2. 자원 군집 분석 기반 추천 서비스 구조

그림 3은 본 연구에서 제안하는 자원 군집 분석 기반 추천 서비스 구조를 나타낸다. 하이브리드 클라우드 환경에서 응용의 특성을 반영하여 자원의 군집을 형성하고, 이를 기반으로 자원을 추천해주는 서비스를 제공한다.

사용자가 실행하고자 하는 응용을 제출하였을 때, 해당 응용의 특성을 기반으로 자원 군집을 형성한다. 자원은 하이브리드 클라우드 자원의 명세를 바탕으로 군집분석을 진행한다. 실행하고자 하는 응용의 특성을 반영하여 군집이 형성된다. 형성된 군집에서 비용 효율적인 자원을 추천하여 사용자에게 제공하게 된다.

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. 또한, 결과를 자원 추천 서비스에 제공한다.

자원 추천 서비스는 응용의 특성을 기반으로 형성된 군집에 따라 비용 효율적인 자원을 선택한다. 선택된 자원은 사용자에게 추천 자원으로 제공된다.

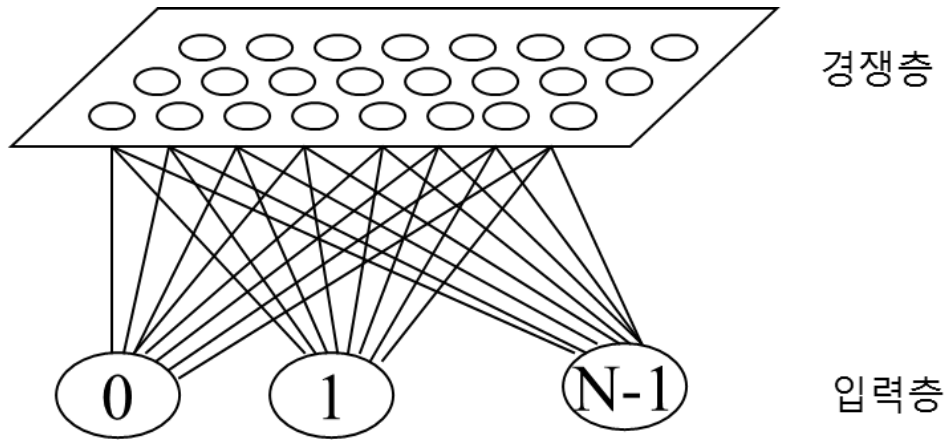


[그림 3] 자원 군집 분석 기반 추천 서비스 구조

3. 군집 분석

군집분석[23]은 비지도학습(Unsupervised learning)[24]의 일종으로, 라벨 데이터 없이 주어진 데이터들의 특성을 고려해 분류하는 것이다. 즉, 샘플들에 대해 지식 없이 유사도에 근거하여 군집들을 구분한다. 패턴 공간에 주어진 유한 개의 패턴들이 서로 가깝게 모여 무리를 이루고 있는 패턴 집단을 군집이라고 하며, 무리 지어 나가는 처리과정을 군집화라 한다. 군집간의 유사도를 평가하기 위해 여러 가지의 거리 측정 함수를 사용하며, 그 예로는 Euclidean distance를 사용한다. 같은 군집이라는 것은 비슷한 특성을 가진 데이터들의 집단이다. 반대로 데이터의 특성이 다르면 다른 군집에 속해야 한다.

자기조직화지도(Self-Organizing Map; SOM)[25]는 대뇌피질의 시각피질을 모델화한 인공신경망의 일종이다. 이는 비교사 학습에 의한 클러스터링 방법의 하나이며 자기조직화맵이라고도 불린다. 그림 4는 자기조직화지도의 구조를 보여주며, 2개의 층으로 이루어져 있는 것을 확인할 수 있다. 첫 번째 층은 입력 층이고 두 번째 층은 경쟁 층으로 2차원 격자로 되어있다. 모든 연결은 첫 번째 층에서 두 번째 층의 방향으로 되어 있으며, 두 번째 층은 완전 연결(fully connected) 되어 있다. 이 지도는 입력공간에 대한 위상 속성을 보존하려는 성격을 가진다. 또한 이는 입력 벡터를 훈련 집합에서 일치되도록 가중치를 조정한다. 자기조직화지도는 고차원으로 표현된 데이터를 저 차원으로 변환하여 저장하는데 유용하다.



[그림 4] 자기조직화지도 구조[26]

K-평균 알고리즘[27]은 분할적 군집분석의 가장 널리 알려진 기법 중 하나이다. 이는 샘플 데이터를 입력 받은 K 개의 군집으로 분류한다. 이때, 군집을 나누는 과정은 거리 기반 그룹간 비 유사도와 같은 비용 함수를 최소화하는 방식으로 이루어진다. 이 과정에서 같은 군집 내 데이터끼리는 유사도가 증가하고 다른 그룹에 있는 데이터의 유사도는 감소하게 된다. K-평균 알고리즘은 각 군집의 중심과 군집 내의 데이터와의 거리의 제곱 합을 비용 함수로 정하고, 이 함수 값을 최소화하는 것을 목표로 데이터 소속 군집을 업데이트 해줌으로써 군집화를 수행한다. 이는 컴퓨터 비전, 지질통계학, 천문학 등 광범위한 분야에 적용될 수 있다. K-평균 알고리즘에서는 군집 개수인 K를 지정해주어야 하는 단점이 있다. K-값에 따라 군집 결과가 달라지며, 성능에 영향을 미친다.

Elbow method[28]는 군집의 개수를 설정할 때 이용하는 방법론이다. 클러스터의 수를 순차적으로 추가하면서 결과를 모니터링 한다. 만약 하나의 군집이 추가되었을 때 이전보다 더 나은 결과를 나타내지 않는다면, 이전의 군집의 수를 구하고자 하는 군집의 수로 설정한다.

본 연구에서는 다양한 속성을 가진 자원을 군집화 하기 위해 자기조직화지도[25]를 이용하여 1차적인 군집 분석을 수행한다. 그리고 이를 바탕으로 요약된 데이터를 입력 뉴런으로 설정하여 K-평균 알고리즘[27]을 통한 2차 군집 분석을 수행하여 클라우드 자원을 분류한다.

IV. 자원 군집 분석 기반 추천 알고리즘

본 장에서는 응용 특성을 반영한 자원 군집분석 및 비용 효율적인 자원 추천 기법에 대해 소개한다. 응용 특성을 반영한 자원 군집분석은 자기조직화지도[25]를 이용한 응용 특성 가중치 반영 모델 학습, Elbow method 기반 K-평균 군집 분석을 이용한 군집 도출에 대해 제안한다. 마지막으로 비용 효율적인 자원 추천 기법에 대해 설명한다.

1. 응용 특성을 반영한 자원 군집 분석

```
Application Characteristics  $AC = \{CPU, Memory, Network, \dots\}$   
1) Submit Application  $App$   
2) Identify the Most Important Application Characteristics  $AC$   
3) Set  $InputData = \{AC, Resource List RL\}$   
4) Group of Output Nodes  $SOM\ model \leftarrow$  SOM training( $InputData$ )  
5) Cluster Lists  $CL \leftarrow$  K-means clustering( $K, SOM\ model$ ) based  
on elbow method.
```

[그림 5] 응용 특성을 반영한 자원 군집 분석 알고리즘

그림 5은 응용의 특성을 반영하여 다양한 클라우드 서비스 제공 회사 및 사설 클라우드 자원을 군집화하는 알고리즘이다. 응용 특성 집합은 응용을 실행하는데 영향을 미치는 자원의 요소로 CPU, Memory, Network 등의 원소로 구성되어 있다.

$$\text{Application Characteristic AC} = \{\text{CPU, Memory, Network, ...}\} \quad (1)$$

사용자는 수행하고자 하는 응용(*App*)을 제출한다(line 1). 제출된 응용의 실행에 가장 큰 영향을 미치는 특성(*AC*)을 찾는다(line 2). 자기조직화지도 훈련을 위한 입력데이터(*InputData*)를 준비한다(line 3). 입력 데이터는 응용 특성(*AC*)와 3.1 자원 특성 명세의 <표 3> 자원 명세를 적용한 자원 리스트(*RL*)로 이루어져 있다.

$$\text{InputData} = \{\text{AC, Resource List RL}\} \quad (2)$$

자기조직화지도 훈련(SOM training)을 이용하여 자기조직화지도 모델(*SOM model*)을 도출한다(line 4). 입력데이터(*InputData*)를 입력뉴런으로 사용하여 출력 뉴런의 그룹으로 자기조직화지도 모델(*SOM model*)이 도출된다. Elbow method를 기반으로 K-평균 군집분석(K-means clustering)을 이용하여 형성된 군집 결과(*CL*)를 출력한다(line 5).

응용 특성을 반영한 자원 군집 분석은 크게 두 단계로 진행된다. 첫 번째 단계에서는 자기조직화지도를 이용하여 자원을 경쟁층에 맵핑한다. 이 때, 사용자가 제출한 응용의 실행에 영향을 미치는 특성에 가중치를 부여하여 군집을 형성한다. 두 번째 단계에서는 첫 번째 단계의 결과를 K-means 군집분석을 진행한다. 이 때, elbow method를 이용하여 적정 군집개수 *k*를 결정한다.

1) 자기조직화지도를 이용한 응용 특성 가중치 반영 모델 학습

그림 6은 자기조직화지도[25]를 이용한 모델 학습 알고리즘을 나타낸다. 알고리즘을 시작하기에 앞서, 자기조직화지도[25]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.

```
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)  $t \leftarrow 0$ 
2)  $W_{ij}(0) \leftarrow$  random number, for all i and j
3) Input neuron  $\leftarrow$  input vector
4) While( $t \neq$  Iter)
5)   Calculate  $d_j$  for all output neuron
      
$$d_j = \sum_{i=0}^{N-1} k_i (X_i(t) - W_{ij}(t))^2, \quad k_i = \begin{cases} 1, & i \neq \text{characteristic} \\ \alpha, & i = \text{characteristic} \end{cases}$$

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
```

[그림 6] 자기조직화지도를 이용한 응용 특성 가중치 반영 모델 학습 알고리즘

현재 반복 횟수를 0으로 초기화 한다(line 1). 또한 모든 연결가중치 벡터($W_{ij}(0)$)를 랜덤 값으로 초기화 한다(line 2). 새로운 입력 벡터를 입력 뉴런으로 제시한다(line 3). 입력벡터는 자원리스트(RL) (그림 5의 line 3)로 구성되어 정규화 과정을 거친 후, 입력 뉴런으로 제시된다. 최대 반복횟수까지 반복한다(line 4). N 은 입력 벡터의 속성 개수이며, 각 속성에 대해 입력뉴런($X_i(t)$)과 연결 가중치 벡터($W_{ij}(t)$) 사이의 거리 제곱 합을 도출한다(line 5). 이 때, k_i 는 응용의 특성 (AC) (그림 5의 line 2)에 대한 가중치로 응용 실행에 영향을 미치는 요소에 가중치를 곱하여 도출한다. 사용자가 제출한 응용에 가중치를 부여 해야 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. 현재 반복횟수를 증가시키고(line 8) 지정된 최대반복횟수까지 반복하여 학습을 진행한다.

2) Elbow method 기반 K-평균 군집 분석을 이용한 군집 도출

1)의 자기조직화지도[25]를 이용한 학습된 모델을 이용하여 K-평균 군집분석[27]을 진행한다. 1)에서 학습된 결과 모델은 출력 뉴런이 각각 연결가중치를 갖는다. 이를 K-평균 군집분석[27]의 입력 값으로 이용한다. 이 때, 군집의 개수를 추가하며 elbow method[28]를 수행하여 적정 군집개수 K를 선정한다.

그림 7 는 K-평균 군집분석[27]의 알고리즘을 나타낸다. 초기에 생성할 군집의 개수(k)가 주어진다. 주어진 군집 개수대로 군집 중앙점(c_i)을 랜덤으로 초기화한다(line 1-3). 각 데이터에 대해 군집 중앙점과 최소거리를 갖는 군집을 라벨($l(d_i)$)로 할당한다(line 4-6). *changed* 변수는 false로 초기화하며(line 7) false로 변경될 때까지 다음을 반복 수행한다. 앞서 각 데이터에 할당된 라벨에 따라 새롭게 형성된 군집에 대해 군집 중앙점을 업데이트한다(line 9-11). 각 데이터에 대해 군집 중앙점과 최소거리를 갖는 군집을 임시라벨(*minDist*)로 할당한다(line 13). 임시라벨(*minDist*)과 이전의 라벨($l(d_i)$)이 같지 않으면(line 14) 라벨을 업데이트하고(line 15) *changed* 변수를 true로 변경한다(line 16).

```

k = number of clusters
D = { $d_1, d_2, \dots, d_n$ }, set of data
C = { $c_1, c_2, \dots, c_k$ }, set of cluster centroids
L = { $l(d) \mid d = 1, 2, \dots, n$ }, set of cluster labels of D
1) for each  $c_i \in C$ 
2)    $c_i \leftarrow d_j \in D$ 
3) endfor
4) for each  $d_i \in D$ 
5)    $l(d_i) \leftarrow j$  with minimum distance( $c_j, d_i$ ),  $j \in \{1 \dots k\}$ 
6) endfor
7) changed  $\leftarrow$  false
8) repeat
9)   for each  $c_i \in C$ 
10)    UpdateCluster( $c_i$ )
11)  endfor
12)  for each  $d_i \in D$ 
13)     $minDist \leftarrow j$  with minimum distance( $c_j, d_i$ ),  $j \in \{1 \dots k\}$ 
14)    if  $minDist \neq l(d_i)$  then
15)       $l(d_i) \leftarrow minDist$ 
16)      changed  $\leftarrow$  true
17)    endif
18)  endfor
19) until changed == false

```

[그림 7] K-평균 군집분석 알고리즘

K-평균 군집분석[27]을 진행 시, 적정 군집개수를 도출하기 위하여 elbow method[28]를 수행한다. Elbow method[28]는 군집의 개수에 따라 각각 군집 중심과 해당 군집에 속한 요소간의 거리의 제곱 합(wcss; within cluster sum of squares)을 계산한다.

$$WCSS = \sum_{i=1}^K \sum_{x \in S_i} \text{dist}(x - c_i)^2 \quad (1)$$

수식 1은 n 개의 데이터(x_1, x_2, \dots, x_n) 집합이 주어졌을 때, K 개의 군집 $S = \{S_1, S_2, \dots, S_k\}$ 으로 분류하는 것을 의미한다. 이때 c_i 는 집합 S_i 의 중심점이며 각 군집 별 중심으로부터 군집 내 데이터의 제곱 합 도출한다.

군집 개수가 증가함에 따라 wcss 값은 감소하게 된다. 군집의 개수가 추가되더라도 wcss 값이 크게 감소하지 않은 경우, 이전의 군집의 개수가 적정 군집 개수로 선택된다(1).

2. 비용 효율적인 자원 추천 기법

군집 분석을 통해 자원들은 응용의 특성을 반영하여 유사한 특성을 가진 자원끼리 군집을 형성하게 된다. 형성된 군집은 유사한 성능을 나타내지만 비용은 다양하다. 따라서, 응용 실행에 영향을 크게 미치는 요소가 아닌 다른 요소로 인해 비용이 증가한 자원 선택을 막기 위해 다음과 같이 Efficiency 를 정의하였다. 가격이 0인 경우에는 해당 자원의 Efficiency를 ∞ 로 정의한다.

$$\text{Efficiency} = \frac{\prod \text{application characteristics}}{\text{cost}} \quad (2)$$

Efficiency 값을 통해 응용 특성이 자원의 가격에 미치는 영향을 수치화 할 수 있으며, 해당 값이 클수록 비용 효율적인 자원을 추천 할 수 있다.

V. 실험 및 결과

본 장에서는 제안하는 알고리즘의 성능을 평가하기 위해 실험을 진행하고 그 결과를 분석했다. 먼저 대상 응용 특성에 대해 설명하고, 자원 군집분석 결과에 대해 분석한다. 마지막으로, 진행한 실험에 대해 설명하고 결과를 분석한다.

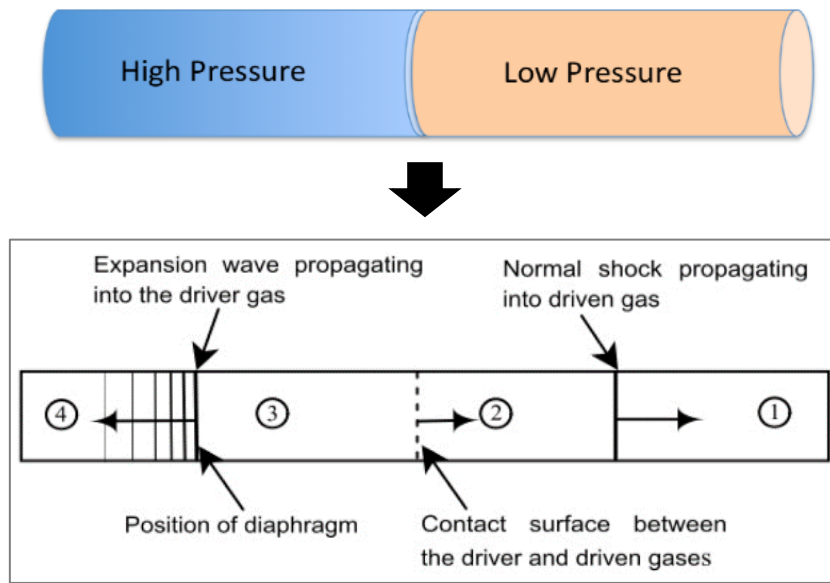
1. 대상 응용 특성

제안하는 기법의 우수성을 확인하기 위하여 서로 다른 2가지의 과학 응용을 활용하여 실험을 진행했다. 항공우주 분야의 비정상 유동해석 역학 시뮬레이션(Computational fluid dynamics; CFD)[29]과 천문학 분야의 Mosaic 이미지 생성 엔진(Montage GALFA)[30]을 대상으로 실험을 진행했다.

1) 항공우주 분야: 비정상 유동해석 역학 시뮬레이션

비정상 유동해석 역학 시뮬레이션(CFD)중 2차원 Euler 비정상 유동해석 프로그램을 대상으로 수행하였다. 해당 프로그램에서 수행하는 문제는 2-D shock tube problem으로 그림 8과 같이 원형 tube에 가운데 격막을 기준으로 고압관과 저압관으로 나뉘어져 있다. 격막을 제거 했을 때, 압력 차에 의해 고압관에서부터 저압관으로 비정상 유동장(충격파: shock, 접촉면: contact, 팽창파: expansion wave)가 발생한다. 프로그

램에서는 특정 Target time을 설정하고, 해당 시간의 비정상 유동을 해석하여 예측한 결과 output file을 생성한다[31]. 따라서, CFD의 응용 특성은 CPU 집약적이며 자원의 CPU에 따라 실행에 영향을 받는다. 또한 본 실험에서는 2KB의 격자를 사용했으며, 입력 파일에서 Target time을 0.5, 최대 반복횟수를 1000001로 설정하여 실험을 진행하였다.

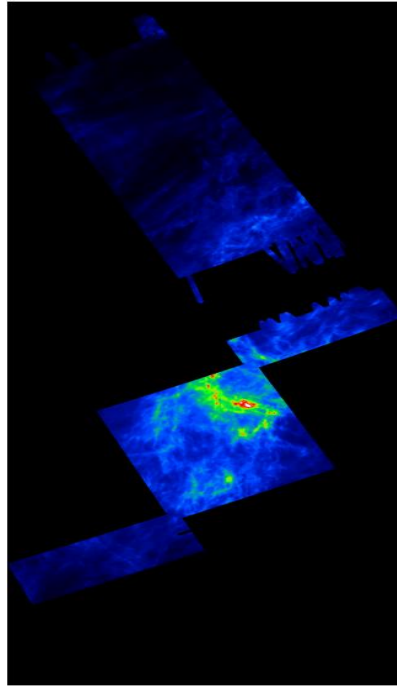


[그림 8] Shock tube 문제[29]

2) 천문학 분야: Mosaic 이미지 생성 엔진

천문학 분야의 Mosaic 이미지 생성 엔진(Montage GALFA)은 대용량의 input 데이터를 사용하여 수많은 중간데이터와 하나의 output 데이터를 생성한다. 해당 응용은 27.5GB의 입력 데이터를 사용하며, 중간 생성 데이터 및 최종 데이터를 생성하기 때문에 데이터 집약적인 응용으로 분류된다. shrink 단계에서 planes의 크기(15, 10, 5)에 따라 데이터가 달라지기 때문에 중간/최종 생성되는 데이터의 사이즈가 달라진다. Montage

응용은 특정 모듈에서 이전에 생성된 이미지 테이블과 헤더 파일을 활용하여 새로운 mosaic FITS 이미지를 생성한다. 이때 급격한 메모리 사용을 요구한다. 따라서, Montage의 응용 특성은 메모리 집약적이며 자원의 메모리에 따라 실행에 영향을 받는다. 또한 본 실험에서는 shrink 단계에서 15 planes를 사용하여 실험을 진행하였다.



[그림 9] Mosaic 이미지 생성 엔진 수행 결과[31]

본 연구에서 대상이 되는 CFD 응용은 CPU 집약적인 응용 특성을 가지며 응용 실행에 영향을 미치는 자원 요소로 CPU를 설정하였다. 또한 Montage 응용은 메모리 집약적인 응용으로 메모리를 실행에 영향을 미치는 자원 요소로 설정하였다.

2. 자원 군집분석 결과 분석

본 연구에서 4.1.1 자기조직화지도를 이용한 응용 특성 가중치 반영 모델 학습을 진행할 때, 64개의 출력뉴런(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 학습횟수는 500번으로 설정하였다. 4.1.2 Elbow method 기반 K-평균 군집 분석을 이용한 군집 도출 부분에서 군집개수가 증가할 때, 이전 군집개수와 wcss 값 차이가 10미만은 경우를 적정 군집 개수로 선정하였다.

그림 10, 11, 12는 97개의 클라우드 자원들을 4.1 응용 특성을 반영한 자원 군집 분석에 따라 분류된 군집을 나타낸 결과이다. 이는 64개의 출력 뉴런(8*8)에 대해 각각 속하는 군집을 그림으로 표현한 것이다. 각 뉴런의 숫자는 뉴런의 번호이다. 각 출력 뉴런은 응용의 특성을 반영함에 따라 다른 값을 갖게 되어, 해당 특성에 따라 각각 다른 구성의 클라우드 자원 군집이 형성된다. 표 4는 4.1.1 자기조직화지도를 이용한 응용 특성 가중치 반영 모델 학습의 입력 뉴런의 일부이다. 자원들은 표 4와 같이 정규화되어 저장된다. 자원들은 훈련을 통해 출력 뉴런에 맵핑된다. 응용을 고려하지 않은 경우, 표 4의 자원들은 모두 다른 뉴런으로 할당되며, K-평균 군집분석을 통해 {Standard7, Standard8, N-m2.xlarge, n1-highcpu-8}이 군집 1로, {n1-standard-8, n1-highmem-8}이 군집 5로, {N-c2.large}이 군집 6으로 분류된다. CFD 응용을 고려한 경우, {n1-standard-8, n1-highmem-8} 25번 뉴런, {n1-highcpu-8}은 26번 뉴런, {Standard7, Standard8, N-m2.xlarge, N-c2.large}은 27번 뉴런에 맵핑된다. 또한 K-평균 군집분석을 통해 모든 뉴런은 군집 2의 하나의 군집으로 분류된다. 응용을 고려하지 않은 경우에 비해, CFD응용의 특성을

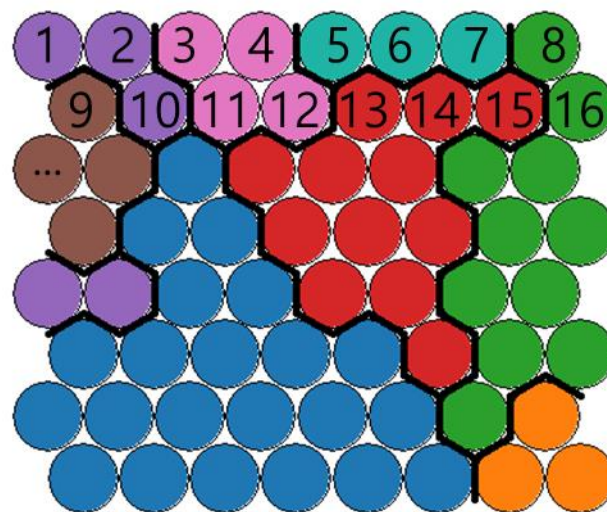
고려하여, vCPU 값이 같은 자원이 하나로 묶인 것을 알 수 있다. Montage 응용 특성을 고려한 경우, {Standard7, Standard8}은 13번 뉴런, {N-m2.xlarge, N-c2.large}은 5번 뉴런, {n1-standard-8}는 1번 뉴런, {n1-highmem-8}는 9번 뉴런, {n1-highcpu-8}는 6번 뉴런으로 맵핑된다. 이는 {Standard7, Standard8, N-m2.xlarge, N-c2.large, n1-highcpu-8}가 군집 2, {n1-standard-8}가 군집 1, {n1-highmem-8}가 군집 3으로 분류된다. 이는 Memory 속성에 가중치를 부여했기 때문에 군집을 형성시, 메모리의 값이 유사함에 따라 군집을 형성하는 결과를 보인다.

<표 4> 응용 특성 가중치 반영 모델 학습의 입력 뉴런의 일부

자원 이름	Provider	Memory	Cpu flops	Network Bandwidth	vCPU	Storage	Cost	Drive type
Standard7	0.36	-0.54	-0.56	-0.48	0.56	0.07	-0.20	-0.77
Standard8	0.36	-0.07	-0.56	-0.48	0.56	0.07	0.28	-0.77
N-m2.xlarge	1.05	-0.07	-0.56	-0.48	0.56	-0.70	0.07	-0.77
N-c2.large	1.05	-0.54	-0.56	-0.48	0.56	-0.70	-0.34	-0.77
n1-standard-8	1.74	0.74	0.13	-0.48	0.56	-0.70	-0.01	0.54
n1-highmem-8	1.74	2.04	0.13	-0.48	0.56	-0.70	0.26	0.54
n1-highcpu-8	1.74	-0.59	0.13	-0.48	0.56	-0.70	-0.26	0.54

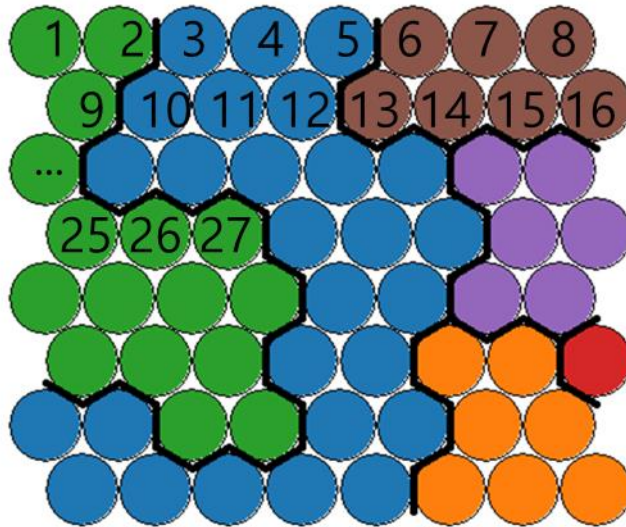
응용의 특성을 고려하지 않고 군집을 형성한 결과 그림 10과 같이 8개의 군집으로 분류되었다. 응용의 특성을 고려하지 않고, 모든 속성을 같은 비율로 반영하여 군집을 형성했을 때, 클라우드 자원 회사가 같을수록 같은 군집으로 형성되는 것을 확인할 수 있다. CFD 응용의 경우, vCPU를 응용의 특성으로 고려하여 vCPU가 유사할수록 자원이 같은 군집을 형성하며 그림 11과 같이 6개의 군집을 형성했다. Montage 응용의 경우에도 Memorysize에 가중치를 부여하여 그림 12와 같이 7개의 군집을 형성했다.

Clusters



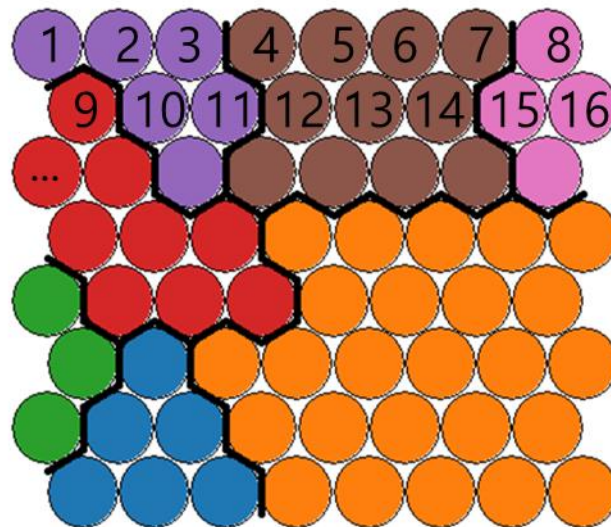
[그림 10] 응용 특성을 고려하지 않은 군집 결과

Clusters



[그림 11] CFD 응용 특성을 고려한 군집 결과

Clusters



[그림 12] Montage 응용 특성을 고려한 군집 결과

표 5, 6, 7는 응용 특성을 고려하지 않은 경우와 응용의 특성을 반영하여 군집을 분석한 결과의 자원 일부를 표로 나타낸다. 예를 들어, 표 6의 CFD 응용의 특성을 반영한 경우에 대해 살펴보자. vCPU에 가중치를 부여했으므로, 표 3의 자원에 대해 4.1 응용 특성을 반영한 자원 군집 분석을 적용했을 때, vCPU가 유사한 자원끼리 같은 군집을 형성하게 된다. 자원 A0와 N-n2.small은 자원 명세 일부인 표 3에서 확인할 수 있듯이, Memory Size가 각각 0.75, 2이다. 또한 Cost는 각각 2, 4의 값을 갖는다. 이와 같이 응용 실행에 영향을 미치는 특성(vCPU)이 아닌 다른 특성(Memory Size, Cost)이 유사하지 않더라도 응용 특성(vCPU)이 1로 동일하기 때문에 군집 1로 형성된 것을 알 수 있다.

<표 5> 응용 특성을 고려하지 않은 군집 분석 결과 일부

군집 4	D11 v2	군집 6	Standard9
	D12 v2		Standard10
	D2		Standard11
	D3		Standard12

<표 6> CFD 응용의 특성을 반영한 군집 분석 결과 일부

군집 1	A0	군집 5	m4.2xlarge
	n1-standard-1		m3.2xlarge
	n1-standard-2		r3.2xlarge
	N-n2.tiny		F8
	N-n2.small		High-memory6

<표 7> Montage 응용 특성을 반영한 군집 분석 결과 일부

군집 4	A7	군집 5	m3.2xlarge
	A8		c4.4xlarge
	D13 v2		c3.4xlarge
	D13		F16

3. 비용 효율적인 자원 추천 결과 분석

응용의 특성에 따라 군집이 다양하게 형성되는 것을 확인하였다. 형성된 군집에서 4.2의 비용 효율적인 자원 추천 기법을 적용하여 각 군집에서 추천된 자원을 도출했다.

예를 들어, CFD 응용의 경우, CPU가 응용의 특성이다. 군집 1번의 경우, A0 자원은 CPU는 1, cost는 2의 값을 갖는다. 같은 군집에 있는 N-m2.small 자원은 CPU는 1, cost는 4의 값을 갖는다. 이 때, 각각 A0의 Efficiency는 0.5, N-m2.small의 Efficiency는 0.25로 A0의 Efficiency가 큰 것을 알 수 있다. 이는 응용 실행에 영향을 미치는 요소와 가격을 활용하여 비용 효율적인 추천이 가능함을 알 수 있다.

<표 8> 군집내의 비용효율적인 자원 추천 결과

	군집번호	추천 자원
CFD 응용	1	A0
	2	N-c2.large
	3	m1.xlarge
	4	N-c2.xlarge
	5	F8
	6	A4
Montage 응용	1	n1-standard-8
	2	n1-highmem-2
	3	n1-highmem-8
	4	D12 v2
	5	r3.2xlarge
	6	r3.xlarge
	7	m1.xlarge

4. 군집화 성능 비교 분석

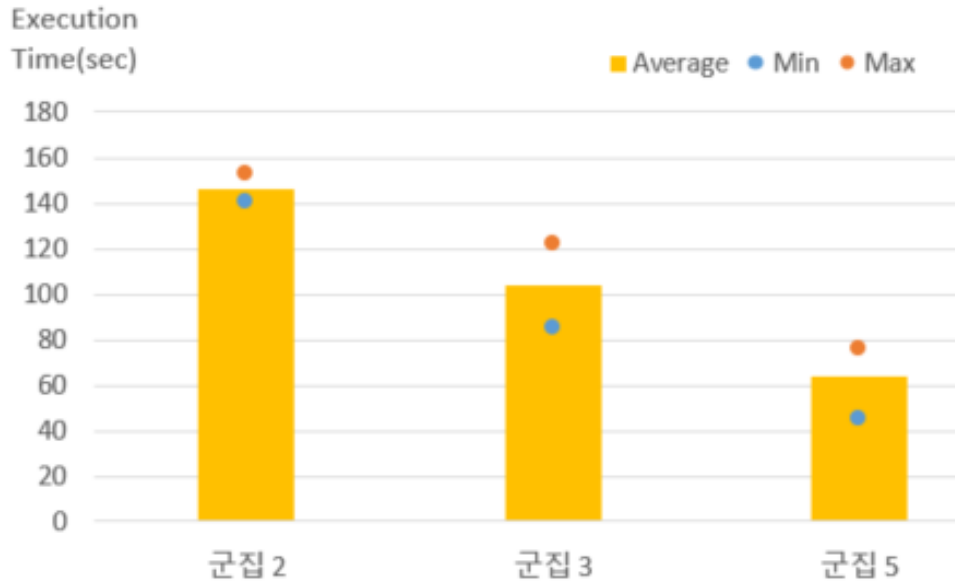
1) 군집 내의 실행시간 유사성 비교

두 가지 응용(CFD, Montage)의 군집 내 실행시간 유사성 비교를 위해 Amazon EC2[11]의 3개 자원(m4.xlarge, m4.2xlarge, m3.2xlarge)과 Microsoft Window Azure[12]의 2개 자원(A3, F8), Openstack[16]의 2개 자원(m1.medium, m1.large)에서 CFD 응용과 Montage 응용을 반복 수행하여 평균 실행시간을 측정하였다. 각 군집의 평균 실행시간을 추정하기 위하여, 응용의 특정 군집 자원을 반복 수행하여 표준편차를 도출하였다. CFD 응용의 경우, 군집 5의 자원을 5회 반복 수행하여, 표준편차 16을 얻었다. 추정오차 8초 이내임을 90% 확신하기 위하여 각 자원에 대해 응용을 5회 반복 수행하였다[38]. Montage 응용의 경우, 군집 6의 자원을 5회 반복 수행하여, 표준편차 23을 얻었다. 추정오차 12초 이내임을 90% 확신하기 위하여 각 자원에 대해 응용을 5회 반복 수행하였다[38].

응용 특성을 반영한 자원 군집분석 결과에 따라 실제 수행한 위의 자원들은 3개의 군집으로 분류된다. CFD 응용의 특성을 반영하여 군집을 분류한 경우, m4.xlarge, A3은 군집 2, m1.medium, m1.large는 군집 3, m4.2xlarge, F8, m3.2xlarge은 군집 5로 분류된다. 그림 13은 각 군집별 작업 실행시간의 평균치는 막대그래프로, 최대 및 최소값은 점으로 표시하였다.

군집 2의 평균 실행시간은 146.5초이며, 최소값은 140초, 최대값은 153초이다. 군집 3의 평균 실행시간은 103.5초이며, 최대값 및 최소값은 각각 122초, 85초이다. 군집 5의 경우는 평균 실행시간이 63.67초이며 최소값 및 최대값은 45초, 76초이다. 같은 군집에 속해 있는 자원들은 유사한 실행시간을 나타내고, 각 군집의 평균실행시간이 달라 유사한 자

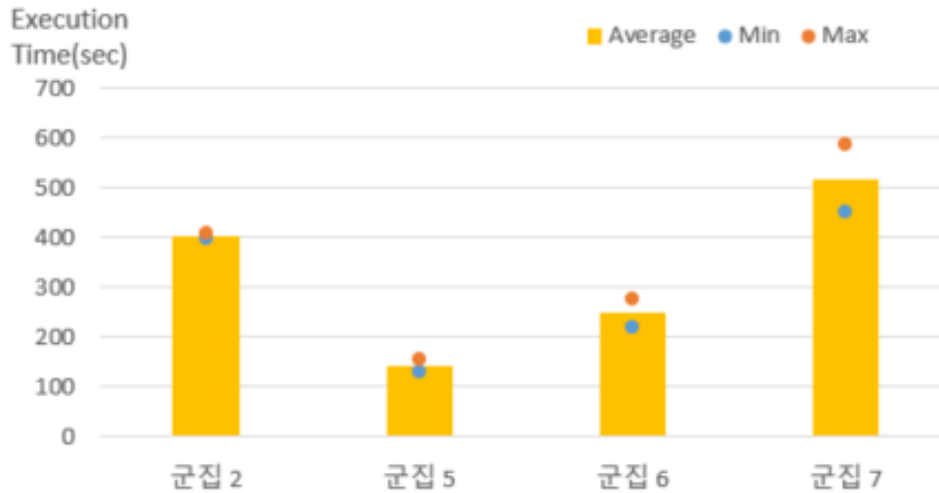
원끼리 군집을 형성한 것으로 볼 수 있다.



[그림 13] CFD 응용의 군집 별 실행시간 측정 결과

Montage 응용의 특성을 반영하여 군집을 분류한 경우, A3는 군집 2, m3.2xlarge, m4.2xlarge는 군집 5, m4.xlarge, F8은 군집 6, m1.medium, m1.large은 군집 7로 분류된다. 그림 14는 각 군집 별 작업 실행시간의 평균치는 막대그래프로, 최대값 및 최소값은 점으로 표시하였다.

군집 2의 평균 실행시간은 401초이며 최소값 및 최대값은 398초, 406초이다. 이는 A3 자원을 5번 반복 수행하여 얻은 결과이다. 군집 5의 평균 실행시간은 141.5초이며 최대값은 153초, 최소값은 130초이다. 군집 6의 평균 실행시간은 247.5초이며 최소값 및 최대값은 각각 220초, 275초이다. 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.



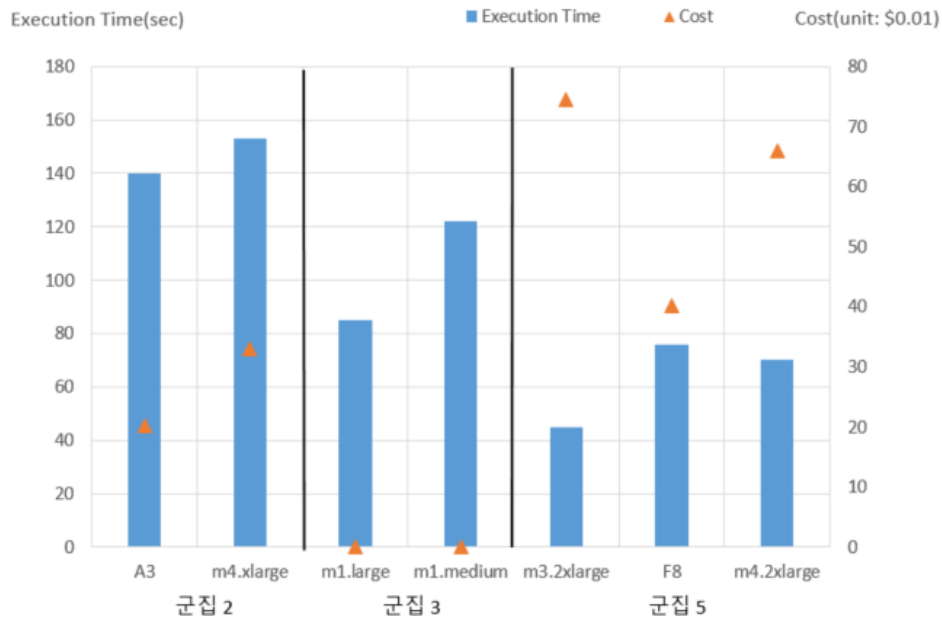
[그림 14] Montage 응용의 군집 별 실행시간 측정 결과

2) 군집 내의 실행시간 및 가격 비교

앞서 군집 내의 실행시간의 유사성을 검증하는 실험에서 사용된 자원들에 대해 각 군집 내의 자원의 실행시간과 가격을 비교하였다. 유사한 자원으로 형성된 군집 내에서 응용의 특성에 영향을 크게 미치는 요소의 성능을 비슷하지만 그 외의 다른 요소들의 성능은 각 자원마다 다르다. 같은 군집 내에서 유사한 실행시간을 갖지만 다양한 가격으로 비용 효율적인 추천기법이 가능함을 볼 수 있다.

그림 15은 CFD 응용의 군집 별 실행시간 및 가격을 비교한 결과이다. 가격은 \$0.01를 기준으로 설정하였다. 군집 2의 경우, 실행시간이 더 긴 자원(m4.xlarge)의 가격이 높은 것을 볼 수 있다. 이는 각 자원이 제공하는 메모리크기로 인한 가격 차이임을 확인 할 수 있다. 응용 실행

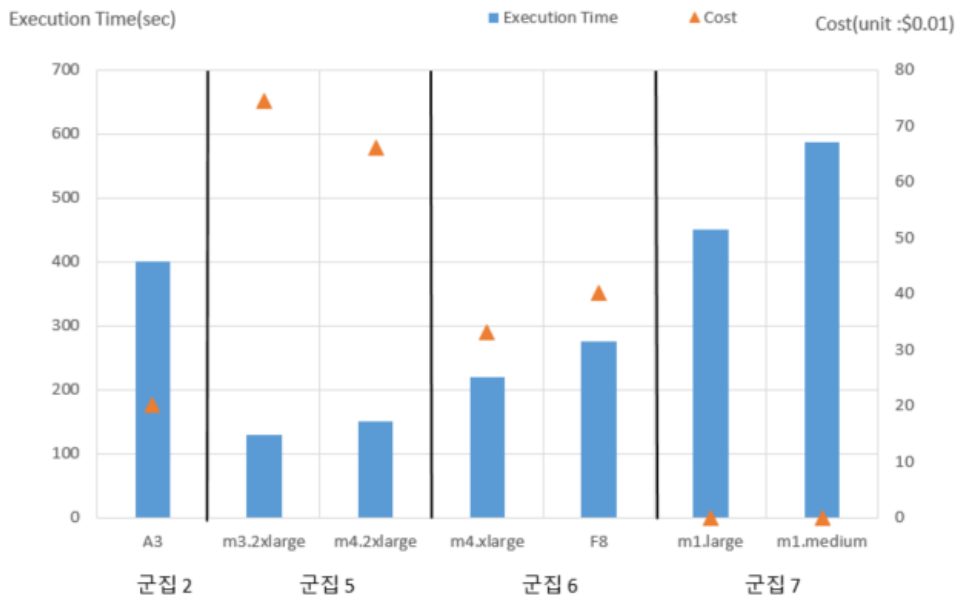
에 영향을 크게 미치지 않은 메모리 크기를 비교했을 때, A3와 m4.xlarge 자원은 각각 7GB와 16GB의 메모리를 제공한다. 이러한 요인으로 응용의 실행시간이 길지만 가격도 비싼 경우를 확인 할 수 있다. 군집 3의 경우는 두 자원 모두 사설 클라우드인 Openstack의 자원임으로 가격은 0이다. 군집 4의 경우는 가장 실행시간이 짧은 자원(m3.2xlarge)은 가격이 가장 높은 것으로 나타난다. 그리고 나머지 두 자원(F8, m4.2xlarge)은 실행시간이 유사하나 제공하는 메모리의 크기는 각각 16GB, 32GB이다. 따라서, 실행시간이 유사하나 응용의 실행에 영향을 미치는 특성 외의 요소로 가격이 달라지는 것을 확인하였다.



[그림 15] CFD 응용 특성을 반영한 군집의 실행시간 및 가격 비교

그림 16은 Montage 응용의 군집 별 실행시간 및 가격을 비교한 결과이다. 가격은 \$0.01를 기준으로 설정하였다. 군집 5의 경우, 일반적인 경우와 유사하게 실행시간이 길면, 가격이 저렴한 것을 볼 수 있다. 군

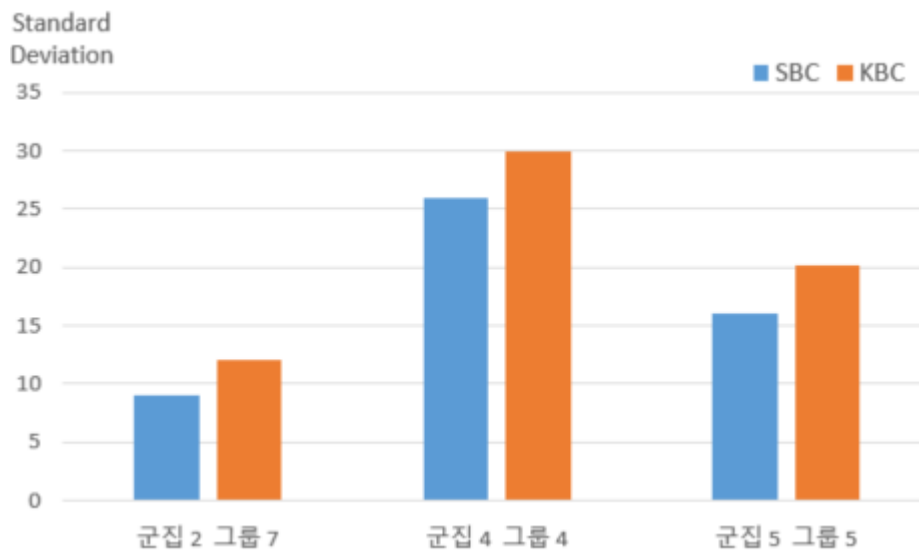
집 6의 경우는 자원 F8의 경우, 실행시간이 길지만 m4.xlarge 보다 가격이 더 비싼 것을 알 수 있다. 이는 m4.xlarge와 F8 각각 자원이 제공하는 CPU가 4, 8 이기 때문이다. Montage 응용 실행에는 CPU가 큰 영향을 미치지 않으므로 실행시간이 길지만 오히려 가격이 비싼 자원이 존재할 수 있다. 군집 7의 경우는 사실 클라우드 Openstack의 가상자원으로 가격은 0이다.



[그림 16] Montage 응용 특성을 반영한 군집의 실행시간 및 가격 비교

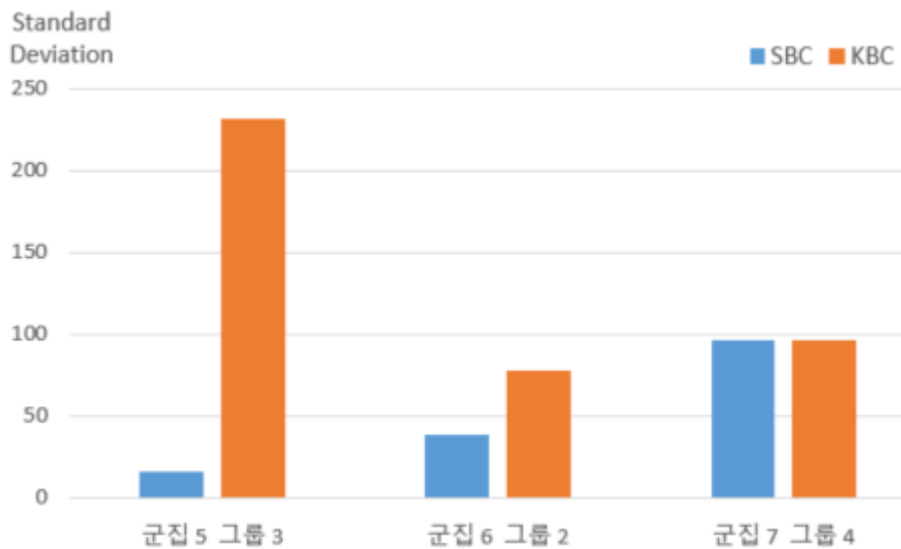
3) 타 군집 분석 기법과의 성능 비교

본 연구와 유사하게 과학 응용의 특성을 반영하여 클라우드 자원 군집을 도출한 선행연구[20]에 대해 비교실험을 진행했다. 표 3의 자원에 대해 CFD응용 특성으로는 vCPU를 설정하여 가중치를 부여하였다. Montage 응용 특성으로는 MemorySize를 설정하였다. 선행연구[20]에서 제안한 k-modes 기법을 적용하여 군집을 형성했다. 형성된 자원군집 중, 1)군집 내의 실행시간 유사성 비교 실험에서 사용한 자원이 포함된 그룹을 중심으로 자원을 샘플링하여 실행시간을 측정하였다. 본 연구에서 제안하는 기법을 이용하여 형성한 군집과 선행연구[20]의 기법을 이용한 군집결과가 다르다. 따라서, 1) 군집 내의 실행시간 유사성 비교 실험에서 사용한 군집의 자원과 가장 많은 공통자원이 포함된 그룹을 유사 그룹으로 지정하였다.



[그림 17] CFD 응용 특성을 반영한 선행연구의 표준편차 비교

그림 17은 CFD 응용을 각 자원에 대해 5회 실행시간을 측정하여 유사 그룹에 대해 군집의 표준편차와 그룹의 표준편차를 비교한 결과이다. 본 연구에서 제안하는 군집기법을 SBC(Som based Clustering)이라고 명시하고, 연구[20]에서 제안하는 군집기법을 KBC(K-modes based Clustering)이라 명시하였다. 그룹 4는 m1.medium, m1.large, m4.2xlarge 자원으로, 그룹 5는 m3.2xlarge, N-m2.xlarge, N-c2.large 자원으로, 그룹 7은 A3, F4로 구성되었다. 군집 2와 그룹 7은 A3을 공통 자원으로, 군집 4와 그룹 4는 m1.medium, m1.large를 공통자원으로, 군집 5와 그룹 5는 m3.2xlarge를 공통자원으로 갖는다. 따라서 각 군집과 그룹을 유사그룹으로 지정하였다. 각 군집 및 그룹을 구성하는 자원에 대해 CFD 응용을 수행하여 실행시간을 측정하고, 이에 대한 표준편차를 도출했다. 모든 경우에 대해 SBC의 기법을 적용한 경우, 군집 내의 표준편차가 작은 것을 알 수 있다. 특히, 군집 5와 그룹 5의 표준편차는 각각 16, 20.2로 차이가 가장 크다.



[그림 18] Montage 응용 특성을 반영한 선행연구의 표준편차 비교

그림 18은 Montage 응용을 각 자원에 대해 5회 반복 수행하여 선행연구[20]와 표준편차 비교 실험 결과이다. 그룹 2는 m4.xlarge, N-c2.xlarge, D13 자원으로, 그룹 3은 m4.2xlarge, N-r2.small 자원으로, 그룹 4는 m1.medium, m1.large 자원으로 구성되어 있다. 군집 5와 그룹 3은 m4.2xlarge, 군집 6과 그룹 2는 m4.xlarge, 군집 7과 그룹 4는 m1.medium, m1.large를 공통자원으로 갖는다. 따라서 각 군집과 그룹을 유사그룹으로 지정하였다. 각 군집 및 그룹을 구성하는 자원에 대해 Montage 응용을 수행하여 실행시간을 측정하고, 이에 대한 표준편차를 도출했다. 군집 5와 그룹 3의 경우, 표준편차의 차이가 215로 가장 큰 차이를 보였다. 군집 7과 그룹 4는 구성하는 자원이 같아, 표준편차가 96으로 같은 값을 나타낸다.

유사 군집과 그룹을 구성하는 자원에 대해 응용 실행시간을 측정한 결과, 본 연구에서 제안하는 방법인 SBC의 표준편차가 선행연구[20]에서 제안하는 KBC보다 작은 것을 확인하였다. 이를 통해, 본 연구에서 제안하는 자원 군집 분석 기법은 응용 특성을 반영하여 유사 자원에 대해 정확한 군집을 형성하는 것을 알 수 있다.

VI. 결 론

본 연구에서는 하이브리드 클라우드 환경에서 응용의 특성을 반영한 동적 자원 군집 분석 기법 및 비용 효율적인 자원 추천 기법을 제안한다. 자원 군집 분석은 자기조직화지도 및 K-평균 알고리즘을 적용하여 유사한 자원을 군집화한다. 또한 응용 특성 기반 자원 군집을 기반으로 효율성 척도를 적용하여 비용 효율적인 자원을 추천기법을 제안한다.

제안하는 알고리즘을 기반으로 실제 다양한 클라우드 서비스 제공자의 자원을 이용해 응용을 수행하고 군집 내의 자원들의 실행시간은 유사하고, 자원의 가격은 유사하지 않아, 비용효율적인 자원 추천이 유의미함을 보였다. 또한 타 군집 분석 기법을 이용하여 응용을 수행한 경우와 성능을 비교하여 제안하는 동적 자원 분류 기법의 우수성을 보였다.

향후 연구로는 클라우드 자원 환경을 추가, 확장하고 제안하는 알고리즘을 보완하여 적용할 것이다. 또한 응용에 대해 다양한 특성 및 우선순위를 부여하여 다중 특성에 대해 효과적인 자원 군집 분석 및 추천 알고리즘으로 확장하고자 한다.

참 고 문 헌

- [1] “최근 클라우드 컴퓨팅 서비스 동향” , NET Term: 인터넷 & 시큐리티 이슈, pp. 21-25 June. 2012.
- [2] Yan, Shixing, et al. "Infrastructure management of hybrid cloud for enterprise users." Systems and Virtualization Management (SVM), 2011 5th International DMTF Academic Alliance Workshop on. IEEE, 2011.
- [3] Di Martino, B., et al. "Towards an Ontology-Based Intercloud Resource Catalogue--The IEEE P2302 Intercloud Approach for a Semantic Resource Exchange." Cloud Engineering (IC2E), 2015 IEEE International Conference on. IEEE, 2015.
- [4] 이강찬, “클라우드 컴퓨팅 표준화 동향 및 전략” , Internet and Information Security, 제 3권 제 3호, pp. 03-17, 2012
- [5] Bernstein, David, et al. "Blueprint for the intercloud-protocols and formats for cloud computing interoperability." Internet and Web Applications and Services, 2009. ICIW'09. Fourth International Conference on. IEEE, 2009.
- [6] Nodehi, Tahereh, Sudeep Ghimire, and Ricardo Jardim-Goncalves. "Toward a unified intercloud interoperability conceptual model for IaaS cloud service." Model-Driven Engineering and Software Development (MODELSWARD), 2014 2nd International Conference on. IEEE, 2014.

- [7] Grozev, Nikolay, and Rajkumar Buyya. "Inter-Cloud architectures and application brokering: taxonomy and survey." *Software: Practice and Experience* 44.3 (2014): 369-390.
- [8] Intercloud project, <http://www.intercloudtestbed.org/>
- [9] Moscato, Francesco, et al. "An analysis of mosaic ontology for cloud resources annotation." *Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on*. IEEE, 2011.
- [10] Open Grid Forum: Open Cloud Computing Interface (OCCI): <http://forge.ogf.org/sf/projects/occi-wg>.
- [11] Amazon EC2, <http://aws.amazon.com/>
- [12] Microsoft Window Azure, <http://www.windowsazure.com/>
- [13] KT Ucloud, <http://ucloudbiz.olleh.com/>
- [14] Google Cloud Engine, <https://cloud.google.com/>
- [15] NHN Entertainment, <http://cloud.toast.com/>
- [16] OpenStack, <https://www.openstack.org/>
- [17] 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.
- [18] Whaiduzzaman, Md, et al. "Cloud service selection using multicriteria decision analysis." *The Scientific World Journal* 2014

(2014).

[19] Wu, Qingtao, et al. "A cloud service resource classification strategy based on feature similarity." *Journal of Networks* 9.11 (2014): 2987-2993.

[20] 안윤선, 김윤희, "군집분석을 이용한 하이브리드 클라우드 컴퓨팅 환경에서의 시맨틱 클라우드 자원 추천 서비스 기법", *정보처리학회논문지: 컴퓨터 및 통신시스템*, 제4권 제9호, pp. 283-288, 2015년 9월

[21] Jung, Gueyoung, et al. "Cloudadvisor: A recommendation-as-a-service platform for cloud configuration and pricing." 2013 IEEE Ninth World Congress on Services. IEEE, 2013.

[22] Yan, Shixing, et al. "Cloud service recommendation and selection for enterprises." 2012 8th international conference on network and service management (cnsm) and 2012 workshop on systems virtualization management (svm). IEEE, 2012.

[23] 군집분석, https://en.wikipedia.org/wiki/Cluster_analysis

[24] 비지도학습, Hinton, Geoffrey E., and Terrence Joseph Sejnowski. *Unsupervised learning: foundations of neural computation*. MIT press, 1999.

[25] 자기조직화지도, Kohonen, Teuvo, and Timo Honkela. "Kohonen network." *Scholarpedia* 2.1 (2007): 1568.

[26] 자기조직화지도 구조, *신경망(Neural Networks)*, 단국대학교 컴퓨터공학과, 이복주.

- [27] K-평균 알고리즘, MacQueen, James. "Some methods for classification and analysis of multivariate observations." Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Vol. 1. No. 14. 1967.
- [28] Elbow method, [https://en.wikipedia.org/wiki/Elbow_method_\(clustering\)](https://en.wikipedia.org/wiki/Elbow_method_(clustering))
- [29] 비정상 유동 해석 역학 시뮬레이션(CFD), 서울대학교 공력 시뮬레이션&설계 연구실.
- [30] 천문학 분야의 Mosaic 이미지 생성 엔진(Montage GALFA), <http://montage.ipac.caltech.edu/docs/cubemosaicstutorial.html>
- [31] Mosaic 이미지 생성 엔진 수행 결과, <https://montageblog.wordpress.com/>
- [32] Iosup, Alexandru, and Dick Epema. "Grid computing workloads." IEEE Internet Computing 15.2 (2011): 19-26.
- [33] Zhang, Jian, and Renato J. Figueiredo. "Application classification through monitoring and learning of resource consumption patterns." Proceedings 20th IEEE International Parallel & Distributed Processing Symposium. IEEE, 2006.
- [34] Memory access pattern, https://en.wikipedia.org/wiki/Memory_access_pattern
- [35] Comparing Decoupled I/O Kernels versus Real Traces in the I/O Analysis of the HACC Scientific Application on Large-Scale Systems by Sean McDaniel

[36] Performance and Profiling of the LAMMPS Code on HPCx Fiona J. L. Reid and Lorna A. Smith

[37] libhashckpt: Hash-based Incremental Checkpointing Using GPU's Kurt B. Ferreira^{1,3}, Rolf Riesen², Ron Brighwell¹, Patrick Bridges³, and Dorian Arnold³

[38] 이훈영. "이훈영교수의 연구조사방법론, 도서출판 청람." (2008).

ABSTRACT

A Resource Recommendation Method Based on Dynamic Cluster Analysis for Application Characteristics

Yoori Oh

Department of Computer Science

The Graduate School

Sookmyung Women' s University

With the development of cloud computing technology, there are many scientists who want to perform their experiments in cloud environment. Due to Pay-per-use method, scientists can only pay for cloud services that they needed. However, it is difficult for scientists to select suitable resources since resources are composed of various characteristics. Therefore, classification is needed to support effective utilization of cloud resources. Static classification of resource can derive inaccurate result. And scientists submit various experiment intentions and requirements.

Thus, dynamic resource clustering method is needed for reflecting application characteristics and scientist' s requirement. And cost-effective resource recommendation service is also needed.

In this paper, resource clustering analysis considering application characteristics and cost-effective recommendation method in hybrid cloud environment are proposed. Resource clustering analysis applies Self-Organizing Map and K-means algorithm to cluster similar resources dynamically. Also cost-effective resource recommendation method applies efficiency metric based on application-aware resource clustering. In the experiment, performance is verified by comparing the proposed clustering method with other study' s resource classification method. Results show that the proposed method can classify similar resource cluster reflecting application characteristics and recommend cost-effective resources.

Key words: hybrid cloud, self-organizing map, cluster analysis, dynamic resource clustering