K-Means Clustering 알고리즘 기반 클라우드 동적 자원 관리 기법에 관한 연구

곽민기, 유헌창
고려대학교 컴퓨터정보통신대학원
dolphz@korea.ac.kr, yuhc@korea.ac.kr

A Study on Dynamic Resource Management Based on K-Means Clustering in Cloud Computing

Minki Kwak, Heonchang Yu
Graduate School of Computer & Information Technology, Korea University

요 약

글로벌 퍼블릭 클라우드 산업 규모는 매년 폭발적으로 성장하고 있으며 최근 COVID-19 등 비대면 문화 확산에 따라 지속 확장되고 있다. 클라우드 사업자는 유한한 인프라 자원으로 다수의 사용자에게 양질의 IT 서비스 제공을 위해 잉여 자원 할당을 최소화하는 것이 중요하다. 그러나 일반적인 퍼블릭 클라우드 환경에서는 정적 자원 할당 기법을 채택하고 있기 때문에 사용자의 주관적인 판단에 따라 잉여 자원의 발생은 필연적이다. 본 논문에서는 머신 러닝 기법 중 K-Means Clustering 알고리즘을 적용하여 클라우드 동적 자원 관리 기법을 제안한다. K-Means Clustering 기반으로 클라우드에 탑재된 각 Instance 의 자원 사용률 데이터를 분석하고, 분석 결과를 토대로 각 Instance 가 속한 Cluster 에 대하여 자원 최적화 작업을 수행한다. 이를 통해 전체 데이터센터 관점에서 잉여 자원의 발생을 최소화하면서도 SLA 수준 및 서비스 연속성을 보장한다.

1. 서론

클라우드 컴퓨팅이란 원격지 데이터센터에 설치된 서버로부터 인터넷을 통해 컴퓨팅 자원을 할당받아 IT 서비스를 제공받는 방식이다. 사용자는 시스템을 직접 구축하거나 관리할 필요가 없으며, 필요한 만큼 On-Demand 형태로 하드웨어 및 소프트웨어를 이용하고 그에 따른 비용만 지불하면 된다. 더 이상 시간과 장소에 구애받지 않고 언제 어디서든 IT 서비스를 이용할 수 있게 된 것이다.

최근 COVID-19 로 인한 비대면 문화 확산에 따라 가상 데스크톱 VDI 서비스 기반 재택근무 등 클라우드 전환이 가속화되고 있다. 글로벌 IT 컨설팅 업체인 가트너의 보고서[1] 에 따르면 2021 년 전 세계 퍼블릭 클라우드 시장은 전년 대비 23% 증가한 3,323 억 달러에 이를 것으로 보이며, 2022 년에는 3,975 억 달러 규모로 성장할 것이라고 전망하였다.

이러한 클라우드 산업의 급격한 성장에 따라 사업자는 다수의 사용자에게 양질의 IT 서비스 제공을 위해 효율적인 인프라 자원 관리가 필요하다. 사업자는 물리 인프라를 가상화하여 사용자에게 가상 머신 형태로 Instance 를 제공하며, 사업자가 보유한 물리 자원은 유한하기 때문에 각 서비스 요건에 따라 적절한 크기의 자원을 할당 및 관리하는 것이 중요하다.

그러나 일반적인 퍼블릭 클라우드 환경에서는 서비스 요건이 제대로 반영되지 않은 정적 자원 할당 기법[2] 이 채택되고 있다. 요건에 따라 최적화된 자원의 크기가 동적 할당되는 것이 아니라, 사업자에 의해 사전 정의된 Instance Type 중에서 사용자의 주관에 따라 가장 적절하다고 판단되는 자원의 크기를 고르는 형태이다. 이러한 판단은 개인의 주관에 따라 달라질 수 있으며 서비스 상용화 이전에 필요 자원의 크기를 정확히 예측하는 것은 어렵기 때문에 실제 사용량을 초과하는 잉여 자원의 발생은 필연적이다. 이로 인해 데이터센터 가용 자원이 부족해지면 자원이 고갈된 Instance 는 더 이상 추가 자원을 할당받을 수 없으며 서비스 장애가 발생된다. 또한, 사업자는 사용자와의 계약 사항에 명시한 SLA(Service Level Agreement) 불이행에 따라 금전적 손해가 발생된다[3].

본 논문에서는 클라우드 컴퓨팅 환경에서 잉여 자원의 발생을 최소화하면서도 SLA 를 보장하기 위해 머신 러닝 기반 동적 자원 관리 기법을 제안한다. 머신 러닝 기법 중 K-Means Clustering 을 적용하여 클

라우드에 탑재된 각 Instance 의 자원 사용률 데이터를 분석하고, 분석 결과를 토대로 각 Cluster 별로 추가 자원 할당 또는 일부 자원 회수 등 최적화 작업을 수행한다. 여기서 최적화란 자원의 크기를 변경하는 작업이므로 일반적으로 Instance 삭제 및 재 생성 과정이 수반된다. 따라서 제안하는 기법에서는 현재 Instance 의 상태를 체크하여, 서비스 업데이트 목적의 일시 정지 혹은 서비스 종료 후 재 생성 과정에서 자원 최적화 결과를 반영한다. 이를 통해 SLA 수준 및 서비스 연속성을 보장한다.

2. 관련 연구

2.1 Instance Provisioning

클라우드 사업자는 데이터센터에 보유한 물리 인프라를 가상화하여 사용자에게 가상 머신 형태로 Instance 를 제공한다. 사업자는 <표 1> 과 같이 CPU, Memory 자원의 크기에 따라 Instance Type 들을 사전 정의하고 있으며, 사용자는 서비스 성격에 따라 적절한 Type 을 선택하여 Instance 를 생성한다. 이러한 가상 머신 자원 할당 및 생성 과정을 Instance Provisioning[3] 이라고 한다.

이 때 Instance 에 할당된 자원의 크기가 클수록 더 많은 비용이 발생된다. 따라서 Provisioning 단계에서 필요한 자원만큼만 최적의 Instance Type 을 선택하여 잉여 자원을 최소화하는 것이 중요하다.

2.2 정적 자원 할당 기법

일반적으로 클라우드 사업자가 제공하는 Instance Type 들은 시간이 지나더라도 사전 정의된 크기에서 변하지 않는다[2]. 즉, 사용자는 항상 <표 1> 과 같이 정형화된 Instance Type 중에서 자신의 서비스가 필요한 자원의 크기를 예측하여 가장 적절하다고 판단되는 Type 을 선택해야 한다. 결국 서비스 특성에 최적화된 자원의 크기가 동적 할당되는 것이 아니라, 최초 Provisioning 단계에서 결정된 Instance 의 크기를 변경 없이 정적으로 사용하는 기법이다. 이러한 정적 할당 기법은 전체 데이터센터 관점에서 필연적인 잉여 자원의 발생을 초래하므로 동적 자원 관리 기법의 적용이 필요하다.

2.3 K-Means Clustering

K-Means Clustering 은 머신 러닝 분야 중 비지도 학습에 해당되므로 입력 데이터만 주어지고 정답은 주어지지 않는다. 정답이 주어지지 않았을 때 주어진 입력 데이터 간 유사도를 파악하여, 비슷한 데이터끼리 K 개의 Cluster 로 구분하는 알고리즘이다[4]. 그 결과 K 개 Cluster 및 각 Cluster 의 평균 값을 의미하는 K 개의 중심점(Centroid)이 도출된다.

이러한 K-Means Clustering 알고리즘은 기업의 고객 군 분석을 통한 맞춤형 마케팅 전략 수립, 포털 사이트 뉴스 기사의 주제별 그룹화, 기후 데이터 Clustering 을 통한 날씨 예측 등 다양한 분야에서 활용되고 있다.

<표 1> AWS(Amazon Web Services) Instance Types 예시

Instance Type	vCPU(Core)	Memory(GB)
Nano	2	0.5
Micro	2	1
Small	2	2
Medium	2	4
Large	2	8
XLarge	4	16
2XLarge	8	32

2.4 관련 연구 동향

백현지 외 2 명은 가상 데스크톱 VDI 서비스 제공 시 개인의 작업 유형에 따른 Memory Load 를 K-Means Clustering 분석하여 사용자 맞춤형 자원 할당 시스템[5] 을 제안하였다. 하나의 Instance 에서 수행되는 여러 종류의 프로세스에 대해 시간 경과에 따른 Memory Load 데이터를 수집하고, K-Means Clustering 을 통해 Memory 사용량에 따라 Low, Medium, High Performance 의 Job 으로 구분하였다. 이러한 분석 결과를 토대로 3 가지 자원 할당 정책을 제시하여, 사용자가 원하는 작업 유형(Low, Medium, High)에 따라 Instance Type 을 취사선택하도록 하였다.

3. 클라우드 동적 자원 관리 기법

본 논문에서 제안하는 클라우드 동적 자원 관리 기법은 각 Instance 의 자원 사용률 데이터를 입력으로 받아 K-Means Clustering 분석하고, 각 Cluster 마다 자원 최적화에 대한 의사 결정을 내리며, 서비스 상태에 따라 실제 최적화를 실행하는 기술이다. 최초 Provisioning 단계에서 결정된 Instance 자원의 크기를 변경 없이 정적으로 사용하는 것이 아니라, 알고리즘의 학습에 따라 자원 최적화 과정을 거쳐 동적으로 관리한다. 따라서 전체 데이터센터 관점에서 잉여 자원의 발생을 최소화할 수 있다. 3.1 과 3.2 절에서는 입력 데이터 분석 및 자원 최적화 의사결정 과정을 상세히 기술하며, 3.3 과 3.4 절에서는 구체적인 파라미터 및 알고리즘을 제시한다.

3.1 입력 데이터 형식

클라우드 인프라에 탑재된 전체 Instance 의 개수를 n 개로 가정한다. 입력 데이터는 <표 2> 와 같이

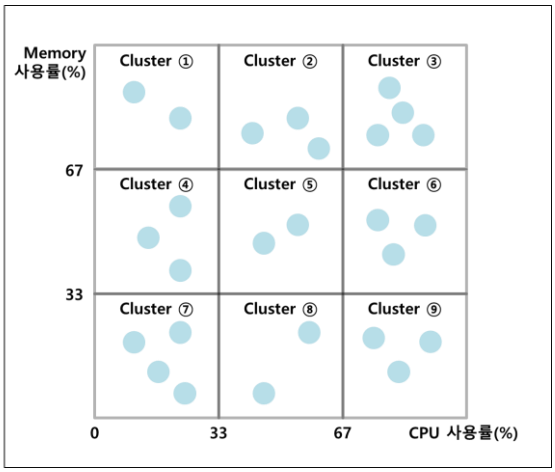
n 개 Instance 들의 할당된 자원의 크기 및 평균 자원 사용률 데이터이다. 이 중 평균 CPU, Memory 사용률 데이터에 대해 K-Means Clustering 을 적용하여 분석한다. 분석 결과에 따라 vCPU Core, Memory Size 등 추가 자원 증설 또는 일부 자원 회수에 대한 의사 결정을 내린다.

<표 2> 동적 자원 관리 기법의 입력 데이터 형식

Instance 번호	vCPU (Core)	Memory Size(GB)	CPU 사용률(%)	Memory 사용률(%)
1 번	2	2	74	17
2 번	2	4	39	72
…	…	…	…	…
(n-1)번	4	16	88	82
n 번	2	8	21	32

3.2 K-Means Clustering (K=9)

각 Instance 의 CPU 사용률 및 Memory 사용률 데이터를 2 차원 좌표 평면 위에 사영하여 유클리디안 거리가 가까운 것끼리 K-Means Clustering 을 적용한다. 이 때 K 값은 (그림 1) 와 같이 9 로 설정하여 Clustering 을 수행하며, 각 Cluster 마다 ① ~ ⑨ 의 번호를 붙인다. 편의 상 X 축은 CPU 사용률, Y 축은 Memory 사용률로 사영한다.



(그림 1) K=9 인 K-Means Clustering 적용 결과

알고리즘의 최종 목표는 각 Cluster ① ~ ⑨ 에 분포된 Instance 들의 자원 사용률을 Cluster ⑤ 와 같은 자원 사용률을 보이도록 추가 자원을 할당하거나 일부 자원을 회수하는 것이다. 이를 위해 <표 3> 과 같이 각 Cluster 별로 자원 크기 조정에 대한 Action 을 의사 결정한다. 만약 CPU, Memory 등 자원 사용률이 33% 미만인 경우 일부 자원 회수(0.5 배) 를 통해 33 ~ 66% 수준(2 배)으로 최적화한다. 반대로 사용률이 67%를 초과하는 경우 추가 자원 할당(1.5 배)을 통해 33 ~ 66% 수준(0.67 배)으로 최적화한다. 마지막으로 사용률이 33 ~ 66%인 경우에는 현재 할당된 자원의 크기를 유지한다.

실제 Instance 의 자원 조정 절차는 즉시 이루어지는 형태는 아니며, 사용자가 서비스 업데이트 목적으로 Instance 를 일시 정지하거나 서비스 종료 후 재이용 시 자원 최적화가 이루어진다.

<표 3> 각 Cluster 별 자원 최적화 Action 의사 결정

Cluster	Action
Cluster ①	- vCPU Core 0.5 배 (사용률 2 배) - Memory Size 1.5 배 (사용률 0.67 배)
Cluster ②	- vCPU Core 유지 (사용률 변화 없음) - Memory Size 1.5 배 (사용률 0.67 배)
Cluster ③	- vCPU Core 1.5 배 (사용률 0.67 배) - Memory Size 1.5 배 (사용률 0.67 배)
Cluster ④	- vCPU Core 0.5 배 (사용률 2 배) - Memory Size 유지 (사용률 변화 없음)
Cluster ⑤	- vCPU Core 유지 (사용률 변화 없음) - Memory Size 유지 (사용률 변화 없음)
Cluster ⑥	- vCPU Core 1.5 배 (사용률 0.67 배) - Memory Size 유지 (사용률 변화 없음)
Cluster ⑦	- vCPU Core 0.5 배 (사용률 2 배) - Memory Size 0.5 배 (사용률 2 배)
Cluster ⑧	- vCPU Core 유지 (사용률 변화 없음) - Memory Size 0.5 배 (사용률 2 배)
Cluster ⑨	- vCPU Core 1.5 배 (사용률 0.67 배) - Memory Size 0.5 배 (사용률 2 배)

3.3 파라미터

알고리즘에서 사용하는 8 개 파라미터를 <표 4> 와 같이 정의한다. Instance VM_i 가 Provisioning 단계에서 할당받은 CPU, Memory 자원의 크기를 각각 C_i, M_i 라 하고, 평균 CPU, Memory 사용률을 각각 CU_i, MU_i 라 한다. 먼저 입력 데이터인 CU_i, MU_i 값에 대해 K=9 로 하여 K-Means Clustering 을 적용하고, 분석 결과에 따라 VM_i 의 최적화된 CPU, Memory 자원의 크기를 구하여 $NewC_i$ 와 $NewM_i$ 변수에 저장한다.

<표 4> 동적 자원 관리 기법의 8 개 파라미터 정의

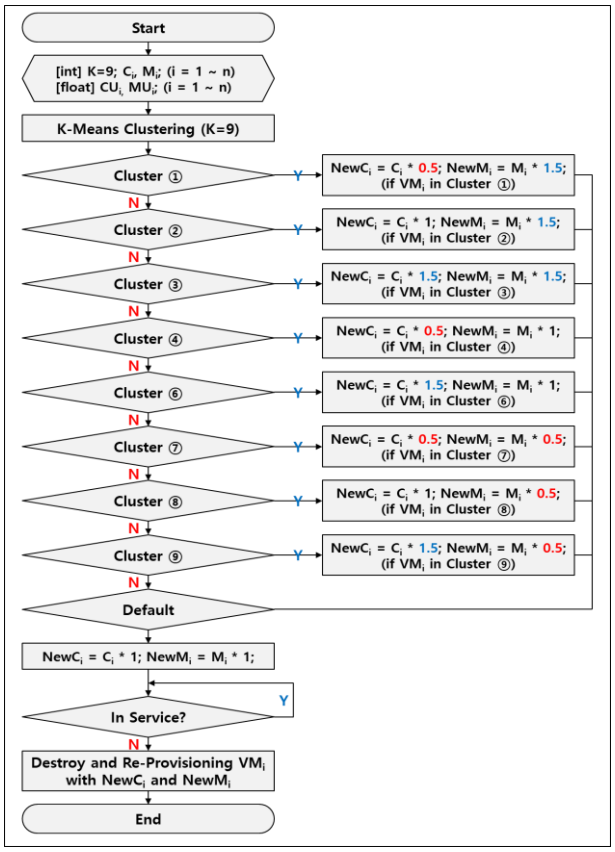
변수	타입	설명
K	int	K-Means Clustering 적용 시 Cluster 의 개수이며 9 의 값으로 고정
VM_i	object	클라우드 인프라에 탑재된 전체 n 개의 Instance 들 (i = 1, 2, …, n)
C_i	int	VM_i 가 할당받은 CPU 자원의 크기
M_i	int	VM_i 가 할당받은 Memory 자원의 크기
CU_i	float	VM_i 의 평균 CPU Usage
MU_i	float	VM_i 의 평균 Memory Usage
$NewC_i$	int	VM_i 의 최적화된 CPU 자원의 크기
$NewM_i$	int	VM_i 의 최적화된 Memory 자원의 크기

3.4 K-Means Clustering 알고리즘 기반 클라우드 동적 자원 관리 기법

K-Means Clustering 알고리즘 기반 클라우드 동적 자원 관리 기법을 (그림 2)와 같이 기술한다. 먼저 입력 데이터는 앞서 정의한 것과 같이 5 개 파라미터가 주어진다. K 값은 Cluster 의 개수를 의미하여 9 로 초기화한다. C_i, M_i 값은 VM_i 가 할당받은 CPU, Memory 자원의 크기를 의미하며 CU_i, MU_i 값은 VM_i 의 평균 CPU, Memory 사용률을 의미한다.

다음으로 입력 데이터 중 CU_i, MU_i 값에 대해 K-Means Clustering 을 통해 분석한다. K 값은 9 로 초기화했기 때문에 전체 n 개의 Instance 들은 2 차원 좌표 평면 상에서 유클리디안 거리가 가까운 것끼리 총 9 개의 Cluster 로 구분된다. 각 Cluster 에 대하여 Switch-Case 제어를 통해 적절한 Action 을 수행한다. 여기서 Action 이란 3.2 절에서 설명한 것과 같이 CPU, Memory 자원 사용률이 속하는 범위에 따라 추가 자원을 할당(1.5 배) 하거나, 일부 자원을 회수(0.5 배) 하여 최적화하는 작업이다. 이러한 작업의 결과로 C_i, M_i 값에 대한 조정이 이루어지며, 조정된 값은 각각 $NewC_i$, $NewM_i$ 값에 저장된다.

마지막으로 각 Instance 의 서비스 상태를 체크한다. 만약 서비스가 Out of Service 상태라면 기존 Instance 를 삭제하고 최적화 완료된 $NewC_i$, $NewM_i$ 의 자원을 할당하여 새로운 Instance 로 재 생성한다. 반대로 서비스가 In Service 상태라면 Out of Service 가 될 때까지 대기 상태를 유지한다.



(그림 2) K-Means Clustering 알고리즘 기반 클라우드 동적 자원 관리 기법

4. 결론 및 향후 계획

본 논문에서 제안하는 기법의 궁극적인 목표는 전체 데이터센터 관점에서 잉여 자원을 최소화하면서도 사용자와 약속한 SLA 수준을 보장하여 안정적으로 서비스하는 것이다. 먼저 K-Means Clustering 을 기반으로 클라우드에 탑재된 모든 Instance 들의 자원 사용률 데이터를 분석했고, 다음으로 각 Instance 가 속한 Cluster 에 대하여 자원 최적화에 대한 의사 결정을 수행했으며, 마지막으로 서비스 일시 정지 또는 종료 여부 확인 후 Instance 의 자원 최적화를 수행하여 SLA 수준 및 서비스 연속성을 보장하였다.

향후 계획으로는 3 장에서 설계한 기법에 대해 실제 구현 및 성능 평가를 진행한다. 이론적으로는 산술 계산에 의해 일부 자원 회수(0.5 배) 시 사용률 증가(2 배), 추가 자원 할당(1.5 배) 시 사용률 감소(0.67 배) 하는 것으로 가정했지만, 실제 테스트 시 vCPU Core 간 병렬 처리 등으로 이론과 다른 결과가 나타날 수 있다. 제안하는 기법의 구현 및 성능 평가를 통해 이론 값과 실측 값을 비교 분석한다.

참고문헌

[1] “Gartner Forecasts Worldwide Public Cloud End-User Spending to Grow 23% in 2021”, Gartner Press Release Newsroom, April 21, 2021, https://www.gartner.com/en/newsroom/press-releases/2021-04-21-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-grow-23-percent-in-2021

[2] 최성민, 송성진, 유헌창, 정광식, 박지수, “클라우드 환경에서 도커의 동적 자원 할당 구현 및 효과 분석”, 한국정보처리학회 추계학술발표대회 논문집 제 22 권 제 2 호, pp.140-143. 2015

[3] 최영호, 임유진, 박재성, “클라우드 컴퓨팅 환경에서 강화학습 기반 자원할당 기법”, 한국통신학회 논문지 제 40 권 제 4 호, 2015, pp.653-658

[4] 장민서, 오수진, 김응모, “TF-IDF 를 활용한 K-Means 기반의 효율적인 대용량 기사 처리 및 요약 알고리즘”, 한국정보처리학회 춘계학술발표대회 논문집 제 25 권 제 1 호, pp.271-274, 2018

[5] 백현지, 김용현, 허의남, "컨테이너 기반 VDI 시스템에서 워크로드 패턴 기반 자원 할당 방법 연구", 한국정보처리학회 추계학술발표대회 논문집 제 24 권 제 2 호, pp.24-26, 2017