

화음 탐색법을 활용한 가상머신 재배포 연구

최재호^{1*}, 김장엽², 서영진¹, 김영현¹
¹국방기술품질원, ²광운대학교

Harmony Search for Virtual Machine Replacement

Jae-Ho Choi^{1*}, Jang-Yeop Kim², Young Jin Seo¹, Young-Hyun Kim¹

¹Defence Agency for Technology and Quality

²Kwang Woon University

요약 데이터센터는 서버, 스토리지, 네트워킹 기기 등을 운영하는 과정에서 냉각시설, 공조시설, 비상발전시설 등 많은 전력이 소비된다. 미국의 경우에는 2004년 데이터센터에서 소비하는 전력은 전체 전력 소비량의 1.8% 정도를 차지하기도 하였다. 데이터센터 산업은 큰 규모로 점진적으로 발전해왔으며, 향후에는 규모가 큰 하이퍼스케일 데이터센터의 수가 늘어날 것으로 전망되고 있다. 하지만 데이터센터의 서버 점유율을 조사해 본 결과, 평균 점유율이 15~20% 정도 밖에 되지 않는 등 서버가 효율적으로 사용되지 않는 문제가 존재하였다. 이러한 현상 및 문제점을 개선하고자 가상머신 마이그레이션 기능을 활용하여 가상머신 재배포 연구를 제안하고자 한다. 본 연구에서는 효과적인 가상머신 재배포를 위해 메타 휴리스틱 기법 중 하나인 화음 탐색법을 활용하였다. 유휴 서버 최대화를 목표로 하는 가상머신 재배포 문제를 설계하였으며 실험을 통해 풀이하였다. 본 연구는 가상머신 재배포를 통해 데이터센터 서버의 절전을 유도하여, 데이터센터의 운영비용을 절감하는 것을 목적으로 한다.

Abstract By operating servers, storage, and networking devices, Data centers consume a lot of power such as cooling facilities, air conditioning facilities, and emergency power facilities. In the United States, The power consumed by data centers accounted for 1.8% of total power consumption in 2004. The data center industry has evolved to a large scale, and the number of large hyper scale data centers is expected to grow in the future. However, as a result of examining the server share of the data center, There is a problem where the server is not used effectively such that the average occupancy rate is only about 15% to 20%. To solve this problem, we propose a Virtual Machine Reallocation research using virtual machine migration function. In this paper, we use meta-heuristic for effective virtual machine reallocation. The virtual machine reallocation problem with the goal of maximizing the idle server was designed and solved through experiments. This study aims to reducing the idle rate of data center servers and reducing power consumption simultaneously by solving problems.

Keywords : Meta Heuristic, Migration, Harmony Search, Visualization, Virtual Machine Reallocation

1. 서론

데이터센터에는 수많은 서버, 스토리지, 네트워킹 기
기들이 존재하며, 이들 서버가 배출하는 열기를 식히기
위한 냉각, 공조시설, 비상발전시설 등은 많은 전력을 소

비한다.

데이터센터가 소비하는 전력은 2005년에서 2010년까
지 24% 증가하여, 정보통신산업이 굴뚝없는 청정산업이
라는 말을 무색하게 하였다. 미국의 경우 데이터센터의
인프라가 소비한 전력량은 2014년 700억kWh로 총 전

본 논문은 2017년 대한민국 교육부와 한국연구재단의 지원을 받아 수행된 연구임(NRF-2017S1A5B8060156)

*Corresponding Author : Jae-Ho Choi(DTAQ)

Tel: +82-10-2731-1087 email: jaeho-choi@dtaq.re.kr

Received November 6, 2018

Revised (1st December 20, 2018, 2nd January 7, 2019)

Accepted February 1, 2019

Published February 28, 2019

력 소비량의 1.8%를 차지하였다. 국내의 경우, 2015년 산업 전체 전력 소비량 2,735억kWh중 약 1%인 26.5억 kWh를 국내 데이터센터가 소비하는 것으로 나타났다. 대형 데이터센터가 1년에 소비하는 전력은 약 6,000만 kWh로 5곳의 전력사용량이 경기 광주시(9만 가구 규모) 가정용 전력사용량과 맞먹는다. 목동에 있는 KT 데이터센터는 에너지 다소비건물 순위에서 1위를 차지하기도 하였다. 이에 데이터센터 사업자들은 전력소비량을 절감하기 위한 노력을 기울이고 있다[1-2].

데이터센터의 에너지 효율은 PUE(Power Usage Effectiveness, 전력소비효율)로 측정한다. PUE는 전체 전력량을 IT 장비가 소비하는 전력량으로 나눈 값을 의미하며, 1에 가까울수록 높은 효율을 뜻한다. 한국IT산업협회의 조사에 따르면, 해외 데이터센터 평균 PUE는 1.7수준이지만, 국내 평균은 2.66수준이었다. 특히 공공, 지자체의 PUE는 3.13으로 조사되었다. 해외 굴지의 기업들은 대규모 데이터센터를 운영할 때 외부의 차가운 공기와 냉각수를 이용하여 서버의 열을 식히는 외기공조 기법을 활용한다. 페이스북은 스웨덴 룰레오에 데이터센터를 구축하였으며, 마이크로소프트는 미국 캘리포니아 연안 해저에서 조력발전을 활용한 수중 데이터센터 구축을 추진 중이다. 이들의 보고서에 따르면 PUE가 1 초반대에 달한다고 한다[2].

국내의 경우에는 강원도 춘천이 데이터센터를 운영하기에 적합한 최적의 자연환경을 가지고 있는 것으로 평가받고 있다. 연평균 기온이 다른 지역보다 1~2도 낮으며, 춘천 소양강댐 냉수를 활용하여 전력 사용량을 줄일 수 있다. 강원도 춘천에는 네이버, 더존비즈온이 데이터센터를 건립하였으며, 삼성SDS 또한 데이터센터를 설립할 예정이다.

자연환경을 활용하는 것은 데이터센터를 효율적으로 운영할 수 있는 최적의 대안이지만, 국내 데이터센터는 높은 토지매입·유지비용에도 불구하고 수도권에 집중되어 있는 특징이 있다. 국내의 경우에는 상면면적 500㎡ 이상 규모를 데이터센터로 정의하고 있는데, 이를 기준으로 하면 2017년 국내에 있는 데이터센터는 민간이 48개, 정부·지자체가 43개로 총 91개다. 이 중 민간의 경우 39개(81.3%)의 데이터센터가 수도권에 집중되어 있으며, 정부·지자체의 데이터센터 또한 16개(37.2%)의 데이터센터가 수도권에 건립되어 있다. 서비스 지연 등의 최소화를 위해 사용자가 밀집해 있는 대도시나 인근 지역

에 입지하고 있는 것이다[3].

데이터센터는 규모가 클수록 운영에 유리(Bigger is better)한 특징을 가지고 있다. 전 세계 데이터센터의 개수는 2017년에 860만 개로 최대치를 기록할 것으로 관측되었지만, 그 이후에는 느리게 감소하기 시작할 것으로 예측되고 있다. 이는 소규모의 데이터센터들은 큰 규모의 메가 데이터센터(하이퍼스케일 데이터센터)로 통합되어 운영하려는 추세를 보이기 때문이다. IDC는 하이퍼스케일 데이터센터를 5,000대의 서버, 1만 평방피트의 면적으로 정의하고 있다. 최근 조사에 따르면 전 세계에 400곳에 가까운 하이퍼스케일 데이터센터가 있다고 한다. 앞으로 하이퍼스케일 데이터센터의 수는 늘어날 것으로 전망되고 있지만, 현재까지도 데이터센터의 운영은 주먹구구식으로 이루어지고 있는 부분이 많이 존재한다 [4].

구글 데이터센터에 있는 일부 서버 5,000대를 6개월간 조사한 결과, 평균 CPU 점유율이 15~20% 정도 밖에 되지 않았다. 서버가 일단 가동이 시작되면 폐기할 때까지 처음 구성된 상태 그대로 운영되는 경우가 많기도 하며, 공급 업체가 어플리케이션을 격리하여 실행해야 한다는 요구사항도 비효율적인 서버 운영의 원인 중 하나이다[5].

그동안 서버 분야의 기술들은 비약적인 발전을 이루었다. 특히 하나의 서버를 여러 개의 가상머신(Virtual Machine, 이하 VM)으로 나누어 사용할 수 있는 가상화 기술은 복잡성 감소, 비용 절감, 유연성, 민첩성 증가 등의 효과를 기대할 수 있었다. 하지만 자원은 충분함에도 서비스 수준 협약(Service Level Agreement, SLA) 위반이 빈번하게 발생하고, 비효율적인 VM 배치로 인한 리소스 사용률 저하, VM수 증가로 인한 기민한 대응 불가, 불필요한 서버 가동으로 인한 데이터센터 전력 및 예산 낭비 등의 기대효과 대비 낮은 실적을 보였다[6-7].

데이터센터의 서버들은 하이퍼바이저로 관리하는데, 대부분의 하이퍼바이저들은 VM을 서버에서 다른 서버로 옮길 수 있는 마이그레이션 기능을 지원한다. 본 연구에서는 VM 마이그레이션 기능을 활용하여, VM을 재배치하여 효율적인 데이터센터 운영에 기여하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 가상화 기술 및 VM 재배치에 대한 기존 연구를 서술하였다. 3장에서는 VM 재배치에 관한 알고리즘을 제안하였고, 4장에서 실험을 진행하였다. 5장에서는 결론 및 향후 연구

를 제시하였다.

2. 기존 연구

2.1 VM 재배치 문제

서버에 설치되어 있는 VM을 마이그레이션하여, 데이터센터의 운영 효율성을 높이는 문제는 VM 재배치 문제(Virtual Machine Reallocation, VMR) 혹은 VM 통합 문제(Virtual Machine Consolidation, VMC) 등이 있다. 본 논문에서는 VM 재배치 문제(이하 VMR)로 기술한다.

앞서 설명한 가상화 기술은 Fig.1.과 같이 하나의 서버에 하나의 운영체제(Operation System)를 설치하여 사용했던 전통적인 구조와는 달리 하나의 서버에 가상화 전용 소프트웨어인 하이퍼바이저를 설치하여, 여러 운영체제로 분할하여 사용할 수 있는 기술을 뜻한다.

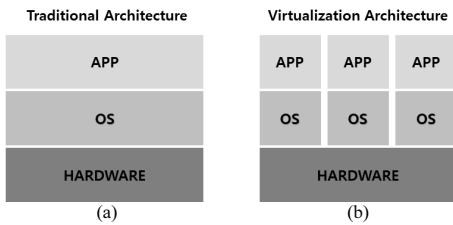


Fig. 1. Difference between traditional architecture and virtualization architecture

이때 하나의 운영체제 단위를 VM이라고 하며, VM은 서버의 일정 CPU와 메모리를 할당받아 운영된다. 일반적으로 하이퍼바이저는 실행 중인 VM을 물리적 서버 간에 이동할 수 있는 마이그레이션 기능을 지원한다. 이때 VM은 이동하는 동안에도 계속해서 작동하며, 사용자 애플리케이션도 계속해서 실행된다. 마이그레이션 기능은 VM의 요구사항을 만족할 수 있을 만큼 대상 서버의 CPU 용량과 메모리 용량이 충분해야 하는 조건이 존재한다. Fig.2.는 VM 재배치 문제의 예시로 마이그레이션 기능을 활용하여 VM을 다른 서버로 옮긴 뒤, 기존 서버의 전원을 중단시켜 전력 사용량을 최소화하는 것을 목적으로 한다[8].

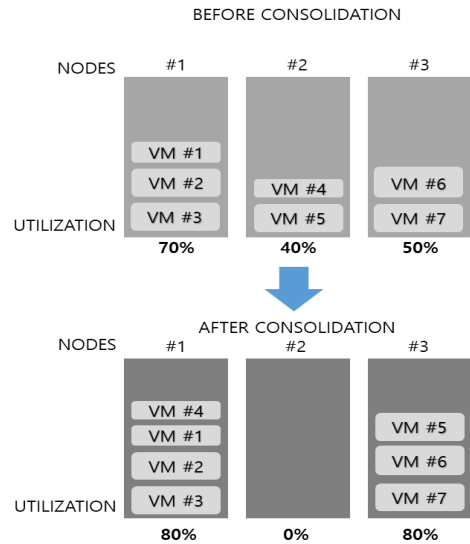


Fig. 2. Example of Virtual Machine Reallocation

VM 재배치 문제는 기본적으로 상자 채우기 문제(Bin Packing Problem, 이하 BPP)와 동일하다. BPP는 n 개의 아이템은 무게, 크기 등이 정해져 있고, 최대 용량이 정해진 가방에 n 개의 아이템을 최대한 채워넣는 문제로서 대표적인 NP-Hard 문제이다. BPP는 가방이 비어있는 상태에서 시작하는 반면, VMR는 VM이 채워진 상태에서 재할당한다는 차이점이 존재한다. Fig.3.은 BPP와 VMR를 표현한 그림으로 BPP의 경우 아이템의 가로, 세로 길이의 합이 각각 x, y 축을 나타내며, VMR의 경우 VM의 CPU와 Memory의 합이 각각 x, y 축을 나타낸다[9].

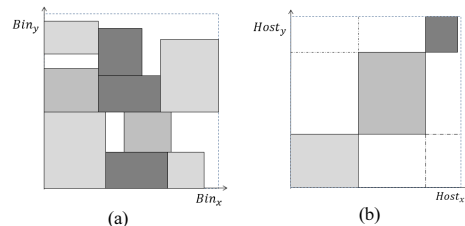


Fig. 3. Representation of (a) 2-D Bin Packing Problem and (b) Virtual Machine Reallocation

2.2 관련 연구

VMR은 크게 어떤 서버(Physical Machine, 이하 PM)를 마이그레이션 할건지 선택하는 Source PM Selection

문제, PM의 어떤 VM을 마이그레이션할 건지 선택하는 VM Selection 문제, 선택된 VM을 어떤 PM으로 마이그레이션 시킬지에 대한 Destination PM Selection 문제도 이루어져 있다.

목적함수는 에너지 소비 최소화(minimize Energy Consumption), 네트워크 트래픽 최소화(minimize Network Traffic), 수익 최대화(maximize Economical Revenue), 성능 최대화(maximize Performance), 자원 활용 최대화(maximize Resource Utilization) 등이 있다. 이 중에서 에너지 소비 최소화와 성능 최대화, 자원 활용 최대화는 비슷한 주제에 속하는데, PM의 수를 최소화하거나 연구자가 직접 전력량을 산출하여, 이를 최소화하는 문제가 가장 많았다.

VMR 연구는 휴리스틱을 활용한 연구가 가장 많은 비중을 차지한다. 대표적인 알고리즘으로 감소순 최적합(Best Fit Decreasing, BFD), 최적합(Best Fit, BF), 최초 적합(First Fit, FF), 감소순 최초적합(First Fit Decreasing, FFD) 등의 기본 알고리즘을 활용한 연구가 일반적이며, 탐욕 알고리즘(Greedy Algorithm)을 활용한 연구 또한 활발하게 이루어졌다. BPP의 경우 FFD 알고리즘을 사용할 경우 최적해의 약 88%를 보장(Performance Guarantee)한다고 증명된 바 있기 때문에 간단한 휴리스틱 기법을 활용하더라도 어느 정도 높은 성능을 얻을 수 있다[10-19].

메타휴리스틱을 활용하여 최적해를 도출하기 위한 연구도 활발하게 이루어졌다. Y. GAO의 4인은 낭비되는 서버의 리소스와 전력을 줄이는데 다목적함수 문제에 적합한 개미 군집 최적화를 제시하였고, 이를 유전 알고리즘(Genetic Algorithm)과 최대-최소 개미 시스템(Max-Min Ant System)과 비교실험을 진행하였다[20]. C.C.T. Mark 외 2인은 VM의 리소스 수요를 예측하고, VM 배치를 최적화하여 비용을 최소화하는 진화 알고리즘(Evolutionary Algorithm)을 제시하였다[21]. A.C.Adamuthe 외 2인은 유전 알고리즘을 VM 배치에 적합하게 디자인하여, FFD와 성능을 비교하였고[22], G. Wu의 3인은 로드 밸런싱과 이익을 최대화하고, 리소스의 낭비를 최소화하는 문제를 유전알고리즘과 그의 확장 알고리즘인 NSGA, NSGA-II를 활용하여 풀이하였다[23]. J. Xu의 2인은 유전 알고리즘을 활용하여 총 리소스 낭비와 전력 소비 및 열 분산 비용을 최소화하는 다목적 함수 문제에 적합한 VM 배치를 제안하였다[24].

Y.Wu 외 2인은 시뮬레이티드 어닐링을 활용하여, VM 배치 문제를 풀이하였고[25], T. Ferreto의 2인은 지정된 최대 시간에 마이그레이션이 완료됨을 보장하는 타부 서치 알고리즘을 제안하였다[26].

3. 문제정의

본 연구에서는 VM 재배치 문제를 통해 유휴 서버를 최대화하는 것을 목표로 한다. 유휴 서버 최대화는 서버를 최대한 끄(released)을 의미하며, 가동되고 있는 서버 사용을 최소화하는 것과 동일한 의미를 갖는다. 가상머신 재배치 문제 풀이는 메타휴리스틱 기법 중에 하나인 화음 탐색법(Harmony Search, 이하 HS)을 활용하였다.

3.1 화음 탐색법

화음 탐색법은 즉흥 음악 연주자들이 화음을 맞춰나가는 과정을 모방한 메타 휴리스틱 알고리즘으로 기존 메타 휴리스틱 알고리즘에 비해 간단하지만, 최적해 탐색에 있어 우수한 성능을 자랑한다.

화음 탐색법은 3가지 연산자(operator)를 활용하여, 불협화음에서 아름다운 협화음으로 발전해 나간다. 연산자는 기존에 연주했던 음원을 그대로 추출하는 HMC(Harmony Considering), 기존에 연주했던 음원을 추출하여, 미세한 조절을 하는 PA(Pitch Adjusting), 임의의 음을 연주하는 Randomization으로 구성된다. 화음 탐색법은 하나의 해를 표현하는 화음 메모리(Harmony Memory, HM)가 화음 메모리 사이즈(Harmony Memory Size, HMS)만큼 존재하는 개체 기반의 메타 휴리스틱(P-Meta Heuristic) 이다.

VM 재배치 문제에서 해 표현 방식은 Fig.4와 같다. 각 배열의 인덱스는 VM의 인덱스를 뜻하며, 인덱스의 값은 VM이 속해있는 PM의 인덱스를 의미한다.

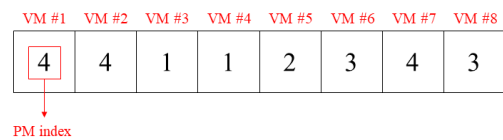


Fig. 4. Representation of VMR Solution (HM)

3.2 고려사항

CPU 점유율과 서버의 전력 소비량은 상관관계가 있기 때문에, CPU 점유율을 높이는 것은 지양해야 한다. 하지만 여러 대의 서버를 관리하는 관점에서 전체의 전력 소비량을 줄일 수 있는 가장 좋은 방법은 서버를 최대한 절전하는 것이다. 이에 본 연구에서는 VM 배치를 통해 서버를 최대한 절전하는 것을 목적으로 하는 알고리즘을 제안하고자 한다.

PM에는 여러 개의 VM이 할당되어 있다. VM은 PM의 CPU 일부와 메모리(RAM) 일부를 점유하고 있으며, PM에 할당된 모든 VM의 CPU와 메모리를 각각 합하면, PM의 CPU와 메모리의 값과 동일하다고 가정한다 (수식 1, 2). 실제 환경에서는 PM이 VM 할당 외에도 다른 작업을 수행하기 때문에 동일하지 않다.

$$Pc_i = \frac{\sum_{j \in S_i} Vc_j}{c_i} \quad (1), \quad Pm_i = \frac{\sum_{j \in S_i} Vm_j}{m_i} \quad (2)$$

VM을 PM에 마이그레이션하기 위해 고려되는 변수는 CPU 값과 메모리 값이다. 마이그레이션할 PM에 VM의 CPU 할당값과 메모리값 만큼의 여유 공간이 있다면, 마이그레이션이 가능하다고 판단한다. 여유 공간이 있는 경우에는 CPU와 메모리 값에 가중치를 두어 점수로 산정(수식 3)하여, source PM과 destination PM을 선택하는 기준으로 활용한다.

$$Score(S_i) = \alpha \times Pc_i + (1 - \alpha) \times Pm_i \quad (3)$$

또한, PM의 경우에는 CPU 값과 메모리 값이 특정값을 초과하지 않는 조건을 추가하였다. PM의 경우 과부하가 발생하여 서비스가 중단되는 경우, 막대한 피해로 이어질 수 있기 때문에 운영을 위한 임계값(threshold)을 지정한다. 임계값의 기준은 환경마다 상이하지만, 기존 연구들을 기준으로 본 연구에서는 70%로 설정하였다.

4. 실험 설계 및 결과

4.1 실험 데이터

본 연구에서 VM 배치에 활용할 데이터는 기존연구를 참고하여, 난수 생성하였다. 실험 데이터는 총 4개의 Class로 구성되어있으며, PM과 VM의 사양은 Table 1.

에 기재된 값을 참고하여 생성하였다. 실험 데이터는 PM과 VM의 CPU 값과 메모리 값을 각각 난수로 생성한다. Class 1의 VM 데이터 생성을 예시로 보면, CPU 값은 1.0과 2.0사이의 값을 균등한 확률로 난수 생성한다. 난수 생성 항목이 별표로 표시된 Class 3, 4의 경우는 특정값이 높은 확률로 난수 생성함을 의미한다. 별표로 표시된 값이 아닌 값들이 선택될 확률은 $(1-P_m)/\text{number of possible values}$ 로 이때 P_m 값은 0.9로 설정하였다. Class 3의 경우를 예로들면, PM의 CPU 값은 {4,5,7,8}이 $(1-0.9)/5 = 0.02$ 확률로 생성되고, 6이 0.92확률로 생성된다. 또한 각 Class별로 VM을 25, 50, 75, 100, 125, 150개 생성하여, 상이한 크기로 실험을 진행하였다.

Table 1. Test data for the experiment

Class	PM		VM	
	Ci	Mi	VCj	VMj
1	6	4	[1.0, 2.0]	[0.5, 1.5]
2	10	6	[1.0, 2.2]	[0.5, 1.5]
3	{4,5,6*,7,8}	{3,4*,5}	[1.0, 2.0]	[0.5, 1.5]
4	{8,9,10*,11,12}	{5,6*,7}	[1.0, 2.2]	[0.5, 1.5]

4.2 실험 설계

VM 재배치 실험은 Performance Guarantee가 증명된 휴리스틱 알고리즘인 FFD와 화음 탐색법의 성능을 비교하였다. FFD는 내림차순을 한 뒤, 앞에서부터 채울 수 있는 공간을 채워나가는 알고리즘으로 의사코드(Pseudo Code)는 Table. 2와 같다.

Table 2. First Fit Decreasing for VMR

```

First Fit Decreasing
Input : S, V, N
S ← initialize Mapping ()
VM Score ← calculate VM Score(V)
sorted VM ← decrease sort VMs by Score
foreach vm ∈ sorted VM
  for I ← 1 to |S|
    success ←check If Migration Possible (vm, Si)
    if(success)
      then add VM to S
      break(for loop)
  end for
end foreach
Output : number of Nodes used, number of released Nodes
    
```

화음 탐색법은 반복을 진행하면서 해를 개선한다. 첫 번째 난수를 생성하여 HMC 연산을 수행할지, Random 연산을 수행할지 결정하고, 만약 HMC 연산을 수행하게 되면, 추가 난수를 생성하여 PA 연산을 수행할지를 판단하고, 그렇지 않으면 HMC를 수행한다. 해가 개선되면 최적해(best harmony)에 저장되며, 반복실험을 마친 뒤에는 VM 배치 정보를 토대로 VM이 어떤 PM으로 이동하고, 그로 인해 어떤 PM이 종료되는지에 대한 결과값이 출력된다. 실험의 반복 횟수는 매개변수 값에 따라 결정된다. Table 3.은 화음 탐색법의 의사코드이다.

Table 3. Pseudo Code of Harmony Search

Harmony Search	
Begin	
Define Objective Function $f(\mathbf{X})$, $\mathbf{X} = (x_1, x_2, \dots, x_n)$	
Define HMCR, PAR, BW and other parameters	
Generate harmony memory with random harmonies	
While $i < \text{number of iterations}$	
While $j < \text{number of variables}$	
IF (Rand1 < HMCR)	
choose a value from HM for the variable j	
IF (Rand2 < PAR)	
adjust the value by adding a certain amount	
End IF	
Else IF choose random value	
End IF	
End While	
Accept the new harmony if better	
End While	
Find the current best solution	
End	
Output : number of Nodes used, number of released Nodes	

4.3 실험 조건

화음 탐색법은 PM 절전 최대화를 목적으로 총 10,000번의 반복을 30회 실시하여, 평균값, 최소값, 최대값을 도출한다. 화음 탐색법의 매개변수는 HM은 VM의 수와 동일하게 설정하였고, HMCR은 0.9, PAR은 0.7로 설정하였다.

실험 환경은 Table 4.와 같이 Windows OS, Intel Core i7(2.67GHz), 16GB RAM의 사양에서 진행하였다.

Table 4. Experiment environment

Item	Description
Objective Function	Number of released Nodes
Parameter	HM = # of VM, HMCR = 0.9, PAR = 0.7
Iteration	10,000
Repetition	30
OS	Windows 10, 64bit
CPU	Intel Core i7 (2.67GHz)
Memory	16GB RAM
IDE	Visual Studio 2017
Programming Language	C# 6.0

4.4 실험 결과

Table 5.에는 실험 데이터별 FFD와 화음탐색법의 결과값을 기재하였다. FFD의 경우는 확률 기반의 알고리즘이 아니기 때문에 특정 데이터에 대한 해는 일정하게 도출되며, Solution 항목은 해당 실험 데이터에 대한 지역해(local optimal)를 뜻한다. 반면 화음 탐색법은 확률 기반의 알고리즘이기 때문에 매 실험마다 결과값이 상이하다. 10,000번의 반복실험을 총 30회 실시하여, 도출된 결과값 중 최고값(Best), 평균값(Average), 최악값(Worst)을 기재하였다. 화음탐색법은 각 클래스 별로 VM이 25개, 50개, 75개인 경우 반복 실험에서 비교적 높은 확률로 지역 최적해에 도달하였다.

목적함수인 유휴 PM의 결과값을 비교한다면, 화음 탐색법이 FFD보다 모든 클래스에서 좋은 결과를 보였다. FFD대비 화음 탐색법의 유휴 PM 개수는 적게는 14%(Class 4, VM Size = 125), 크게는 약 53%(Class 2, VM Size = 100) 정도 개선되었다. 이는 전반적으로 화음탐색법이 FFD보다 효과적이라고 해석 가능하다.

기존 연구에서는 효율적인 마이그레이션을 측정하고자, 마이그레이션 효율성(Migration Efficiency)를 정의하였으며, 이를 구하는 공식은 수식 4와 같다[27].

Migration Efficiency =

$$\frac{\text{Number of nodes}}{\text{Number of migrations}} \times 100\% \quad (4)$$

마이그레이션 효율성은 유휴 전환된 PM을 총 마이그레이션 횟수로 나눈 값으로, 최소한의 마이그레이션으로 PM을 유휴 전환하는 정도를 의미한다. 만약 3번의 마이그레이션으로 3개의 PM이 유휴 전환된다면 마이그레이션 효율성은 100%가 된다. Table 6.은 FFD와 화음 탐색법의 마이그레이션 효율성 값을 기재하였다. FFD의 마

Table 5. Result of the experiment(# of released Nodes)

Class	VM Size	First Fit Decreasing		Harmony Search			
		Solutions	Computation Times(sec)	Best	Average	Worst	Computation Times(sec)
1	25	3	2.24	4	4	4	5.39
	50	8	3.18	10	10	10	12.8
	75	11	5.09	14	13.6	13	21.71
	100	16	6.63	21	19.13	17	37.94
	125	21	8.91	25	22.87	21	51.47
	150	26	10.31	32	30.13	27	60.24
2	25	4	2.02	5	4.4	4	4.89
	50	9	3.29	11	10.8	10	11.2
	75	11	4.08	14	13	12	20.7
	100	15	6.93	23	20.63	17	34.64
	125	20	8.54	26	23.66	22	49.53
	150	26	10.54	35	31.13	26	55.43
3	25	3	2.37	4	4	4	6.17
	50	8	3.68	11	10.6	10	13.25
	75	10	5.21	14	12.83	11	27.22
	100	14	6.65	19	16.77	15	39.99
	125	21	9.02	26	24.27	23	52.62
	150	27	10.63	31	30	29	65.47
4	25	4	1.85	6	5.8	5	6.06
	50	8	3.21	10	9.7	9	13.04
	75	12	4.23	16	14.06	13	26.88
	100	15	7.17	21	19.67	18	38.13
	125	22	9.39	25	24.37	24	49.44
	150	27	10.86	34	30.1	27	63.21

Table 6. Result of the Migration Efficiency

Class	VM Size	First Fit Decreasing			Harmony Search (Best)		
		Solutions	Number of Migrations	Migration Efficiency	Solutions	Number of Migrations	Migration Efficiency
1	25	3	18	16.67%	4	6	66.67%
	50	8	49	16.33%	10	14	71.43%
	75	11	71	15.49%	14	21	66.67%
	100	16	97	16.49%	21	31	67.74%
	125	21	121	17.36%	25	47	53.19%
	150	26	148	17.57%	32	59	54.24%
2	25	4	23	17.39%	5	8	62.50%
	50	9	55	16.36%	11	17	64.71%
	75	11	71	15.49%	14	23	60.87%
	100	15	98	15.31%	23	39	58.97%
	125	20	118	16.95%	26	45	57.78%
	150	26	151	17.22%	35	66	53.03%
3	25	3	17	17.65%	4	6	66.67%
	50	8	48	16.67%	11	18	61.11%
	75	10	71	14.08%	14	26	53.85%
	100	14	89	15.73%	19	36	52.78%
	125	21	117	17.95%	26	47	55.32%
	150	27	146	18.49%	31	57	54.39%
4	25	4	22	18.18%	6	8	75.00%
	50	8	50	16.00%	10	14	71.43%
	75	12	71	16.90%	16	23	69.57%
	100	15	91	16.48%	21	32	65.63%
	125	22	124	17.74%	25	39	64.10%
	150	27	157	17.20%	34	57	59.65%

이그레이션 효율성은 15%~18% 정도인 반면, 화음 탐색법은 마이그레이션 효율성은 50~75% 수준으로 더 높은 효율성을 보였다.

위와 같이 FFD와 화음탐색법을 마이그레이션 효과와 효율성 측면에서 비교했을 때, 화음탐색법이 우세한 결과를 보였지만 화음 탐색법의 경우에는 VM 사이즈가 커질수록, FFD의 실행시간의 증가폭이 크지 않은 반면, 화음 탐색법은 실행시간이 다소 증가하는 모습을 볼 수 있었다. 하지만 VM 사이즈가 가장 큰 경우의 실행시간은 1분 내외로 VM 마이그레이션이 빈번하게 이루어지는 작업이 아님을 감안한다면, 마이그레이션 계획을 도출하는데 걸리는 실행시간은 수용 가능한 시간으로 판단된다.

5. 결론 및 향후 연구

본 연구는 메타 휴리스틱 기법 중 하나인 화음 탐색법을 활용한 가상머신 재배치를 통해 데이터센터 서버의 절전을 유도하여, 데이터센터의 운영비용을 절감하는 것을 목적으로 하였다. 기존 연구에서 제시된 실험 데이터를 활용하여, 본 연구에서 제안하는 화음탐색법과 FFD 알고리즘의 비교 실험을 진행하였다. 실험 결과는 FFD 알고리즘 보다 화음 탐색법이 유휴 PM 수가 더 높게 나왔으며, 마이그레이션 대비 유휴 PM 대수를 비교하는 마이그레이션 효율성 측면에서도 화음 탐색법이 더 높은 결과를 보임으로써 화음 탐색법의 우수성을 검증하였다.

본 연구를 통해 가상머신 재배치 안이 도출될 수 있지만, 실제 환경에서 가상머신 마이그레이션을 하기 위해서는 고려해야할 부분이 다수 존재한다. 라이브 마이그레이션 기능을 통해 마이그레이션을 하는 동안에도 어플리케이션은 지속적으로 실행 가능 하지만 미세하게 다운타임이 발생하기 때문에 보수적인 데이터센터 관리자의 경우 안정성을 이유로 패치를 진행하거나, 유지보수, 서버 점검 시간을 별도로 두어 마이그레이션을 진행한다. 또한 CPU, 메모리 용량이 큰 가상머신의 경우에는 마이그레이션 실행시간이 늘어난다. 상기의 이유 등으로 마이그레이션은 시간과 안정성을 고려해야하기 때문에 효율적인 운영을 위해서는 빈번하게 실행하는 것을 지양해야한다.

본 연구는 특정 시점의 PM과 VM의 점유율 정보를

기반으로 가상머신 재배치를 진행하였다. 하지만 PM과 VM의 CPU와 메모리는 실시간으로 점유율이 변동하는 특징을 가지고 있다. PM의 CPU 점유율이 임계치를 초과하여, VM을 옮기고자 할 때, VM의 CPU 점유율을 현 시점뿐만 아니라 미래 시점까지 고려하여 마이그레이션을 계획할 필요가 있다. 가령 VM의 CPU 점유율을 현 시점만 고려하여, 마이그레이션을 했는데, VM의 CPU 점유율이 현 시점보다 더 높아진다면, 또 다시 마이그레이션을 해야하는 비효율적인 상황이 발생할 수 있다. 이러한 비효율적인 상황에 직면하지 않기 위해서는 PM의 워크로드 밸런스를 고려해야 한다. 서버의 워크로드 패턴이 완만하게 유지된다면, 마이그레이션을 할 필요가 없어진다. 서로의 패턴을 최대한 상쇄시키는 VM 조합들을 생성하여, PM에 배치한다면 마이그레이션 빈도가 줄어들 것이다. 향후 연구에서 VM의 워크로드 패턴을 예측하고, VM 워크로드 밸런스를 고려한 최적 조합을 도출한다면, 보다 현실적인 환경을 고려할 수 있을 것으로 사료된다.

References

- [1] J. H. Bang, "The data center is 'Electric Eating Hippo' & When you are on the Internet," The Hankyoreh, 2015.06.
- [2] Y. J. Bae, "Korea is a data center battleground & Reasons for global companies coming in one after another", Chosun NewsPress, 2017.10.
- [3] M. J. KIM, "Gangwon-do Data Center Status and Future Tasks", The Bank of Korea, 2018.04.
- [4] Data Center Knowledge, "Research: There are Now Close to 400 Hyper-Scale Data Centers in the World", 2017.12.
- [5] W.Vogels, "Beyond Server Consolidation", ACM Queue, 2008.01.-02.
- [6] S. Crosby, and D.Brown, "Virtualization reality", ACM Queue, 2006.12.
- [7] P. Timalena, "A Study of The impact of Virtualization on Computer Networks", Master's thesis, 2013.
- [8] Redhat, "Virtualization Deployment and Administration Guide"
- [9] Eli M. Dow, "Decomposed multi-objective bin-packing for virtual machine consolidation", PeerJ Computer Science, 2016.
- [10] S.-H. Wang, P. P.-W. Huang, C. H.-P. Wen, and L.-C. Wang, "Eqvmp: Energy-efficient and qos-aware virtual machine placement for software defined datacenter networks," in Information Networking (ICOIN), 2014

- International Conference on. IEEE, pp. 220-225, 2014.
- [11] T. Ferreto, C. A. De Rose, and H.-U. Heiss, "Maximum migration time guarantees in dynamic server consolidation for virtualized data centers," in Euro-Par 2011 Parallel Processing. Springer, pp. 443 - 454, 2011.
- [12] D. Dong and J. Herbert, "Energy efficient vm placement supported by data analytic service," in Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on. IEEE, pp. 648 - 655, 2013
- [13] X. Zhang, Q. Yue, and Z. He, "Dynamic energy-efficient virtual machine placement optimization for virtualized clouds," in Proceedings of the 2013 International Conference on Electrical and Information Technologies for Rail Transportation (EITRT2013)-Volume II. Springer, pp. 439 - 448, 2014.
- [14] W. Shi and B. Hong, "Towards profitable virtual machine placement in the data center," in Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on. IEEE, pp. 138 - 145, 2011.
- [15] I. Hwang and M. Pedram, "Hierarchical virtual machine consolidation in a cloud computing system," in Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on. IEEE, pp. 196 - 203, 2013.
- [16] H. Jin, D. Pan, J. Xu, and N. Pissinou, "Efficient vm placement with multiple deterministic and stochastic resources in data centers," in Global Communications Conference (GLOBECOM), 2012 IEEE. IEEE, pp. 2505 - 2510, 2012.
- [17] W. Li, J. Tordsson, and E. Elmroth, "Virtual machine placement for predictable and time-constrained peak loads," in Economics of Grids, Clouds, Systems, and Services. Springer, pp. 120 - 134, 2012.
- [18] O. Biran, A. Corradi, M. Fanelli, L. Foschini, A. Nus, D. Raz, and E. Silvera, "A stable network-aware vm placement for cloud systems," in Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012). IEEE Computer Society, pp. 498 - 506, 2012.
- [19] W. Wang, H. Chen, and X. Chen, "An availability-aware virtual machine placement approach for dynamic scaling of cloud applications," Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on. IEEE, pp. 509 - 516, 2012.
- [20] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing," Journal of Computer and System Sciences, vol. 79, no. 8, pp. 1230 - 1242, 2013.
- [21] C. C. T. Mark, D. Niyato, and T. Chen-Khong, "Evolutionary optimal virtual machine placement and demand forecaster for cloud computing," in Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on. IEEE, pp. 348 - 355, 2011.
- [22] A. C. Adamuthe, R. M. Pandharpatte, and G. T. Thampi, "Multiobjective virtual machine placement in cloud environment," in Cloud & Ubiquitous Computing & Emerging Technologies (CUBE), 2013 International Conference on. IEEE, pp. 8 - 13, 2013.
- [23] G. Wu, M. Tang, Y.-C. Tian, and W. Li, "Energy-efficient virtual machine placement in data centers by genetic algorithm," in Neural Information Processing. Springer, pp. 315 - 323, 2012.
- [24] J. Xu and J. A. Fortes, "Multi-objective virtual machine placement in virtualized data center environments," in Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom). IEEE, pp. 179 - 188, 2010.
- [25] Y. Wu, M. Tang, and W. Fraser, "A simulated annealing algorithm for energy efficient virtual machine placement," in Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on. IEEE, pp. 1245-1250, 2012.
- [26] T. Ferreto, C. A. De Rose, and H.-U. Heiss, "Maximum migration time guarantees in dynamic server consolidation for virtualized data centers," in Euro-Par 2011 Parallel Processing. Springer, pp. 443 - 454, 2011.
- [27] A. Murtazaev, SY. Oh, "Sercon: Server Consolidation Algorithm using Live Migration of Virtual Machines for Green Computing", IETE Technical Review, 28:3, 212-231, 2014.

최 재 호(Jae-Ho Choi) [정회원]



- 2015년 8월 : 연세대학교 산업공학과 (공학석사)
- 2015년 9월 ~ 현재 : 국방기술품질원 항공센터 연구원

<관심분야>
 품질경영시스템, 산업공학, 서비스사이언스

김 장 엽(Jang-Yeop Kim) [정회원]



- 2012년 2월 : 연세대학교 산업공학과 (공학석사)
- 2017년 2월 : 연세대학교 산업공학과 (공학박사)
- 2017년 3월 ~ 8월 : 연세대학교 공학연구원 박사후연구원
- 2017년 9월 ~ 현재 : 광운대학교 방위사업연구소 선임연구원
- 2018년 3월 ~ 현재 : 한국시물레이션학회 교육이사

<관심분야>
 OR, M&S, SCM, 방산수출

서 영 진(Young Jin Seo)

[정회원]



- 2018년 2월 : 경상대학교 정보과학과(공학석사)
- 2014년 8월 ~ 현재 : 국방기술품질원 연구원

<관심분야>

항공전자, 국방 소프트웨어

김 영 현(Young-Hyun Kim)

[정회원]



- 2012년 2월 : 부산대학교 재료공학부 졸업
- 2015년 9월 ~ 현재 : 국방기술품질원 연구원

<관심분야>

품질경영, 품질인증