

에너지 소모량 최소화를 위한 가상머신과 사용자 태스크 배치 알고리즘

원 동욱*, 김 화성^o

Virtual Machine and User Task Placement Algorithm to Minimize Energy Consumption

Dong-wook Won*, Hwa-sung Kim^o

요 약

클라우드에서 가상 머신을 최적의 물리 머신에 배치하는 전략과 사용자 태스크를 최적의 가상 머신에 배치하는 전략은 데이터 센터의 효율적인 전력 관리에 상당한 영향을 미치기 때문에 중요하다. 기존 연구는 데이터센터의 전체 가동 중인 물리 머신의 개수를 최소화하도록 가상머신과 사용자 태스크를 물리 머신에 배치함으로써 낭비되는 유휴 전력을 줄였다. 하지만 기존 연구의 에너지 소모량 최적화 모델은 물리 머신의 가동시간이 아닌 특정 임의의 시간만큼 계산하기 때문에 정확도가 낮다. 또한, 최적화 알고리즘인 Best fit은 지역최적해에 수렴한다는 단점이 존재한다. 본 논문에서는 기존 연구보다 더 높은 정확도의 에너지 소모량 모델을 제안하였고, 지역 최적해 수렴의 위험에서 벗어나는 유전자 알고리즘을 수정한 최적화 알고리즘을 제안하였다. 실험은 CloudSim 시뮬레이터를 통해서 제안 방안과 기존 논문의 방안에 대해서 성능을 비교 평가하였다. 실험 결과 본 논문의 제안 방안이 기존 논문의 방안보다 에너지를 더 효율적으로 소모함을 확인하였다.

Key Words : Virtual machine placement, Bin packing, Cloud computing, User task placement, Energy consumption optimization

ABSTRACT

In the cloud system, the strategy for placing virtual machines on the optimal physical machine and the strategy for placing user tasks on the optimal virtual machine is important because it has a significant impact on energy consumption in the data center. Existing studies have reduced wasted idle power by minimizing the total number of running physical machines. However, the energy consumption optimization model of the existing research has low accuracy because it calculates for a certain small time rather than the makespan of a physical machine. In addition, an optimization algorithm of existing research has a disadvantage that it converges to a local optimal solution. In this paper, we proposed an energy consumption model with higher accuracy than existing energy consumption model. We also proposed an optimization algorithm modified from

* 본 연구는 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2019-0-00260, 초연결 공통 네트워킹 서비스 연구인프라 구축)

※ 이 논문은 2019년도 광운대학교 연구년에 의하여 연구되었음

• First Author : Chung-Ang University Department of Computer Science and Engineering, dwwon@uclab.re.kr, 학생회원

o Corresponding Author : Kwangwoon University Department of Electronics and Communications Engineering, hwkim@kw.ac.kr, 종신회원

논문번호 : 202209-211-B-RN, Received September 15, 2022; Revised November 22, 2022; Accepted December 10, 2022

a genetic algorithm that escapes the risk of local optimal solution convergence. The experiment was conducted through the CloudSim simulator. As a result, it was confirmed that the proposed method consumes energy more efficiently than the existing method.

I. 서 론

클라우드 컴퓨팅은 인터넷을 통해서 사용자에게 IT 하드웨어, 플랫폼, 인프라 같은 정보 자원을 제공한다. 사용자는 정보 자원을 사용한 만큼 클라우드 서비스 제공자에게 비용을 지불한다. 클라우드 서비스는 사업 초기 대규모 초기 투자 비용과 IT 인프라 유지 보수 비용을 경감시키므로 국내외 많은 기업들이 도입하고 있다^[1].

클라우드 서비스의 규모가 커지고 발전함에 따라 데이터센터에서 클라우드 컴퓨팅 환경을 유지하기 위한 전력 관리 문제가 대두되고 있다. 예를 들어 Citigroup의 데이터 센터는 전체 회사 인프라의 공간 비중 1%에 불과하지만, 전력 사용량은 전체 회사의 전력 소모 비용 중 약 25%를 차지한다^[2].

효율적인 전력 관리를 위해서 데이터센터의 전력 비용 절감 연구가 활발히 진행되고 있다. Meisner에 의하면 실제 서버 자원 활용률은 30% 이하이고 자원을 사용하지 않는 유휴 서버로 인해 낭비되는 전력은 데이터센터 최대 전력의 60% 정도를 차지한다고 한다^[3]. 그러므로 전력 소모를 줄이는 가장 좋은 방법은 데이터센터 내에 가동 중인 서버의 전체 개수를 최소화하여 전체 가동된 서버의 자원 활용률을 높이고 동시에 유휴 자원의 전력 낭비를 줄이는 것이다.

서버의 자원 활용률은 클라우드 컴퓨팅의 주요 기술인 가상화를 이용하여 높일 수 있다. 가상화란 컴퓨팅, 네트워크, 스토리지 같은 물리 자원을 여러 개의 논리 자원으로 분할하여 하나의 물리 머신에 여러 개의 가상머신을 생성하는 기술이다. 이렇게 생성된 가상머신에는 사용자 태스크를 위한 독립적인 실행 환경을 구축할 수 있어 한 대의 물리머신에 여러 실행 환경을 구축할 수 있다. 따라서 한 대의 물리머신에 다양한 종류의 사용자 태스크를 실행할 수 있게 되어 자원 활용도가 높아질 수 있고 시스템에 불필요하게 가동되는 물리머신의 수도 줄어 데이터 센터의 에너지 소모량 관점에서 효율적이다.

이런 맥락에서 가상 머신과 사용자 태스크 배치 문제는 중요하다. 가상머신 배치 문제란 가상 머신을 물리 머신에 배치할 때 물리머신의 다양한 특성을 고려하여 최적의 물리 머신에 배치하는 문제다. 사용자 태

스크 배치 문제란 사용자 태스크를 가상 머신에 배치할 때 가상머신의 다양한 특성을 고려하여 최적의 가상 머신에 배치하는 문제다. 이러한 가상머신과 사용자 태스크 배치 문제에 대한 적절한 해결 전략은 클라우드 시스템의 에너지 소모량 비용에 상당한 영향을 미치므로 중요하다. 따라서 본 논문은 클라우드 시스템에서 여러 사용자 태스크와 가상머신의 배치 요청이 접수 되었을 때 데이터 센터의 에너지 소모량이 최소화되도록 사용자 태스크와 가상머신을 배치하는 방법에 대해서 제안하고자 한다.

기존의 연구는 사용자 태스크 배치와 가상 머신 배치 문제를 각각 별도로 다루었다. 하지만 본 논문은 가상 머신과 사용자 태스크를 물리머신에 동시에 같이 배치하였고 이러한 방법이 기존의 연구에서 제안한 방법보다 에너지를 더 효율적으로 소모함을 실험을 통해서 확인했다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 기술하고, 3장에서는 기존의 연구 방안의 문제점을 개선한 본 논문의 제안 방안에 대해 기술한다. 4장에서는 제안 방안과 기존의 방안의 성능을 비교 평가하고, 5장에서는 결론에 대해 기술한다.

II. 배경지식 및 관련 연구

2.1에서는 가상머신 배치 문제에 대해서 기술하고, 2.2에서는 사용자 태스크 배치 문제에 대해서 기술한다. 기존의 논문은 사용자 태스크 배치 문제와 가상머신 배치 문제 각각을 별도로 다루었지만, 본 논문은 사용자 태스크와 가상 머신과 사용자 태스크를 동시에 물리머신에 배치하는 것을 고려하였다.

2.1 가상머신 배치 문제

가상머신 배치 문제란 가상 머신을 물리 머신에 배치할 때 다양한 특성을 고려하여 최적의 물리 머신에 배치하는 것이다. Beloglazov가 제안한 Modified Best Fit Decreasing (MBFD)^[4]은 OpenStack Neat에 구현된 알고리즘으로 현재까지 state-of-the-art로 취급되고 있다. 동작 과정은 먼저 클라우드 시스템에서 가동되고 있는 모든 물리 머신에 대해서 평균적인 CPU 활용도를 기준으로 내림 차순 정렬 한다. 그 다음 가

상 머신이 요구하는 자원량의 조건을 만족함과 동시에 가장 적은 CPU 자원량을 가진 물리 머신을 찾아서 배치한다. MBFD 알고리즘의 장점은 구현이 간단하며 가동 중인 물리머신의 전체 개수가 최소화되도록 가상머신을 배치하기 때문에 유휴 전력의 낭비를 줄인다. 하지만 MBFD는 데이터 센터에 존재하는 모든 물리머신의 종류가 같은 상황을 동질적인 상황을 가정하여 이질적인 물리머신으로 구성된 현실의 데이터센터 상황에서는 효율적으로 동작하지 않는 문제점이 존재했다.

Beloglazov가 제안한 PABFD^[5]는 이질적인 물리머신으로 구성된 데이터센터 상황의 문제를 고려하기 위해 사전에 각 물리머신의 전력과 자원 활용도 간의 선형 관계를 가정하여 물리머신의 전력 소모량을 모델링하였다. 이러한 전력 소모량 모델을 특정 시간 간격만큼 적분함으로써 물리머신의 에너지 소모량을 모델링하였고 이러한 에너지 소모량 모델을 이용해 현재 가상 머신이 특정 물리머신에 배치됐을 때 소모할 에너지량을 예측했다. 그다음 가장 적은 양의 에너지를 소모하는 물리머신을 찾는 Best fit 방식으로 가상 머신을 배치한다.

하지만, PABFD는 에너지 소모량 최적화 모델과 최적화 알고리즘에 대해서 다음의 네 가지 문제점이 존재했다. 첫 번째로 PABFD의 에너지 모델은 물리머신의 자원 활용도와 전력 간의 관계를 인간이 사전에 분석해서 직접 수동으로 모델링해야 하는 번거로움이 존재했다. 본 논문에서 제안하는 에너지 모델은 인간이 직접 모델링하지 않고도 머신러닝의 선형 회귀 모델을 통해 데이터를 기반으로 자동으로 에너지 소모량을 모델링한다. 두 번째로 PABFD의 기존 에너지 모델은 물리머신의 파워 모델을 특정 시간만큼 적분할 때 특정 시간이 가상머신과 사용자 태스크를 배치하기 직전부터 실행을 완료하기까지 경과된 시간이 아니라 배치한 직전부터 직후까지의 짧은 시간이므로 에너지 소모량 모델의 정확도가 부족했다. 반면, 본 논문은 물리머신의 보유 자원과 물리 머신에 걸린 부하를 나타내는 배치된 가상머신 및 태스크의 자원 요구사항에 관련된 데이터를 기반으로 학습된 회귀 모델을 통해 물리머신의 작동 시간을 유추하여 에너지 소모량을 더 정확하게 계산한다. 세 번째로 기존 에너지 소모량 모델에서 고려하지 않는 static power, start up power, shutdown power를 고려하여 더 정교하게 모델링하였다. 네 번째 문제로 최적화 알고리즘으로 탐욕스러운 Best fit 방식을 사용하였기 때문에 지역 최적해에 수렴해 전역 최적해 도달하지 않는다는 문

제점이 존재했다.

Kalka Dubey는 기존의 PABFD 에너지 소모량 모델에 시뮬레이티드 어닐링(Simulated Annealing) 최적화 알고리즘을 적용했다^[6]. 시뮬레이티드 어닐링 알고리즘은 목적함수가 복잡한 비선형 최적화 문제에서 근사 해를 찾는 알고리즘이며 지역 최적해의 수렴에서 확률적으로 벗어난다는 장점이 있기 때문에 기존의 그리디한 최적화 알고리즘 Best fit 보다 더 나은 성능을 보인다. 하지만 시뮬레이티드 어닐링은 해를 1개 단위로 탐색하는 단점이 있다.

유전자 알고리즘은 해를 1개가 아닌 집단 단위로 탐색하기 때문에 시뮬레이티드 어닐링 알고리즘 보다 전역 해를 찾을 확률이 더 높다. 그래서 본 논문은 기존의 유전자 알고리즘을 가상 머신과 사용자 태스크 배치문제에 맞게 수정하여 적용하였다.

2.2 사용자 태스크 배치 문제

사용자 태스크 배치 문제란 사용자 태스크를 가상 머신에 배치할 때 다양한 특성을 고려하여 최적의 가상 머신에 배치하는 것이다. 선행 연구로 제안된 First Come First Serve (FCFS)^[7]은 요청된 사용자 태스크들의 대기 시간이 긴 순서대로 차례대로 가상머신에 배치한다. Shortest Task First (STF)^[8]는 사전에 클라우드 시스템에 실행을 위해 제출된 사용자 태스크의 리스트와 현재 가동 중인 가상머신의 리스트를 입력으로 받고, 사용자 태스크 리스트를 실행시간을 기준으로 정렬이 한다. 그리고 가장 짧은 실행시간을 가진 태스크부터 먼저 가상머신에 배치한다. 반대로 가장 오랜 실행시간을 가진 태스크를 가상머신에 배치하는 Largest Task First (LTF)^[9]방식도 제안되었다. 또한, 본 논문에서 시뮬레이터로 사용하는 CloudSim^[10]에서는 가장 기본적인 사용자 태스크 배치 방식으로 First fit 알고리즘이 구현되어 있다. First fit 알고리즘은 사용자 태스크의 자원을 담을 수 있는 가상 머신 리스트를 차례대로 탐색하다 가장 첫 번째로 발견한 가상 머신에 사용자 태스크를 배치한다.

하지만 위의 논문들은 사용자 태스크 배치만 고려하고 가상 머신 배치 문제를 고려하지 않았다. 가상머신과 사용자 태스크를 동시에 배치하는 것은 데이터 센터의 에너지 효율성 관점에서 상당히 중요하기 때문에 본 논문에서는 두 문제를 동시에 고려하였다.

III. 제안 방안

3.1에서는 본 논문에서 수정한 유전자 알고리즘에

대해서 기술하고 3.2에서는 본 논문에서 제안한 에너지 소모량 모델에 대해서 기술한다. 3.3에서는 가상머신과 사용자 배치 문제 영역에 더 특화된 해 부호화 방법에 대해서 설명한다. 이어서 3.4에서는 초기화 방법, 3.5에서는 선택 연산, 3.6에서는 교배 연산, 3.7에서는 돌연변이 연산에 대해서 설명한다.

3.1 유전자 알고리즘

유전자 알고리즘^[11]은 최적화 문제에서 가능한 해를 염색체라는 자료구조로 부호화하고 이러한 염색체의 집단 군을 만들어 일정 세대 동안 관리한다. 유전자 알고리즘은 처음에 이러한 염색체를 초기화한다. 이후에 생존에 더 적합한 염색체들을 적합도 함수를 이용하여 선별한다. 여기서 적합도 함수란 일반적인 최적화 문제에서 목적함수와 유사한 역할을 한다. 염색체의 적합도 값이 클수록 문제 해결 능력이 높은 것으로 간주하여 재생산을 위한 선택될 확률이 높다. 재생산이란 염색체들끼리 교배, 돌연변이 연산을 통해서 이전 세대의 염색체보다 더 나은 염색체를 만들어내는 것을 의미한다. 교배 연산은 지역 탐색을 수행하고 돌연변이는 전역 탐색을 수행한다. 선택부터 돌연변이 연산까지 수행되는 과정을 한 세대로 부른다. 세대를

거쳐 재생산 과정은 계속해서 무한으로 반복될 수 있으므로 다음 세대로 넘어가기 전에 반복 종료 조건이 있다. 이때 종료 조건을 달성하지 못한다면 재생산된 염색체는 다음 세대에 전달되며 진화가 이루어진다. 만약 종료 조건을 달성한 경우 진화를 멈추고 해가 반환이 된다. 지금까지 설명한 유전자 알고리즘의 흐름을 정리하면 다음의 그림 1과 같다.

3.2 제안하는 에너지 소모량 모델

선형 회귀 분석은 관측된 데이터를 기반으로 독립 변수가 종속 변수에 미치는 영향을 파악하고 특정 독립 변수의 값에 상응하는 종속 변수의 값을 예측하는 모델을 산출하는 통계 기반의 데이터 분석 기법이다. 사례로 난폭 운전과 운전자에 의한 교통사고 총 건수 사이의 상관관계를 분석하거나 부모의 키와 자식의 키에 미치는 영향의 관계를 예측하는 데 사용된다.

주어진 데이터 $\{E_{(i,j)}, x_{(i,1)}, x_{(i,2)}, \dots, x_{(i,n)}\}_{i=1}^m$ 에 대해서 사전에 독립변수 $E_{(i,j)}$ 와 종속 변수 $x_{(i,1)}, x_{(i,2)}, \dots, x_{(i,n)}$ 간의 선형 관계를 가정하고 모델링한다. 이 분석 모델은 수식 (1)으로 표현되며 이러한 방정식을 선형 회귀 모형(Linear Regression Model)이라고 한다.

$$E_{(i,j)} = f_1(x_{(i,1)}, x_{(i,2)}, \dots, x_{(i,n)}) = \beta_0 + \beta_1 x_{(i,1)}^2 + \dots + \beta_n x_{(i,n)}^2 + \beta_{n+1} x_{(i,1)} x_{(i,2)} + \dots + \beta_k x_{(i,n-1)} x_{(i,n)} + \epsilon_{(i,j)} \quad (1)$$

위 수식 (1)에서 $E_{(i,j)}$ 는 종속 변수로 특정 물리머신 j의 에너지 소모량을 뜻하며 예측하고자 하는 결과 값에 해당한다. 입력 데이터로 표현되는 독립변수 $x_{(i,1)}, x_{(i,2)}, \dots, x_{(i,n)}$ 로 부터 종속변수인 에너지소모량 $E_{(i,j)}$ 간의 선형 상관 관계를 분석함으로써 그 예측 결과 값을 산출하는 모델을 설계할 수 있다.

$f_1(x_{(i,1)}, x_{(i,2)}, \dots, x_{(i,n)})$ 은 독립 변수들 간의 1차원 교호작용으로 생성된 변수들의 선형조합이다. 여기서, n는 선형 회귀 분석을 위한 입력 데이터의 개수이다. 에너지 소모량과 관련된 총 19개의 독립 변수를 <표 1>과 같이 선정하였다. 독립변수는 물리머신의 특성과 물리머신에 배치된 가상머신의 특성과 가상머신에 배치된 사용자 태스크의 특성으로 총 세 부분으로 구성되어 있고 선형 회귀 모델의 학습을 위해서 정답 값으로 이런 상황을 가정했을 때의 물리머신이 소모하는 에너지양이 주어진다.

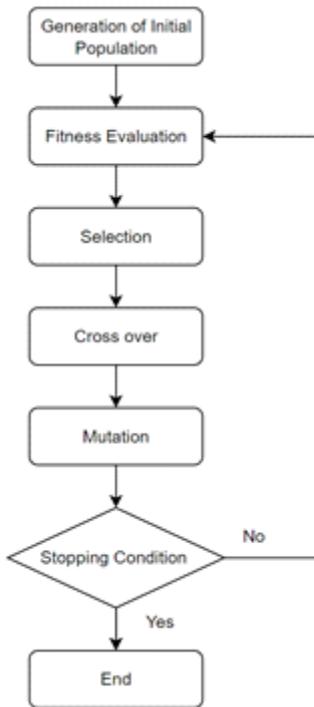


그림 1. 유전자 알고리즘 순서도
Fig. 1. Genetic algorithm flow chart

수식 (1) 에서 계수 $\beta_0, \beta_1, \dots, \beta_k$ 를 선형회귀 모형의 모수 (parameter) 라고 한다. 하지만 모수의 참값을 정확하게 알 수가 없다. 따라서 본 논문에서는 모수 $\beta_0, \beta_1, \dots, \beta_k$ 를 점 추정한 예측 모델 $\hat{E}_{(i,j)}$ 을 가정하며 수식으로는 (2)와 같다. $\hat{E}_{(i,j)}$ 는 특정 물리머신 j에 관련된 i번째 독립변수 데이터 집합이 주어졌을 때의 에너지 소모량 예측값을 의미한다.

$$\begin{aligned} \hat{E}_{(i,j)} &= \hat{f}_1(x_{(i,1)}, x_{(i,2)}, \dots, x_{(i,n)}) \\ &= b_0 + b_1 x_{(i,1)}^2 + \dots + b_n x_{(i,n)}^2 + b_{n+1} x_{(i,1)} x_{(i,2)} \\ &+ \dots + b_k x_{(i,n-1)} x_{(i,n)} \end{aligned} \quad (2)$$

모형을 완성하기 위해서 전체 관찰된 학습 데이터 집합 $\{E_{(i,j)}, x_{(i,1)}, x_{(i,2)}, \dots, x_{(i,n)}\}_{i=1}^m$ 을 토대로 해당 종속변수에 상응하는 관측된 에너지 소모량값과 예측 모델의 측정값 간의 오차 제곱 합을 최소화하는 모수의 점추정치 $b_0, b_1, b_2, \dots, b_k$ 를 찾는다. 이러한 방식을 최소제곱법이라고 하며 오차항 $e_{(i,j)}$ 는 실제 관측된 값 $E_{(i,j)}$ 과 추정된 값 $\hat{E}_{(i,j)}$ 의 차이이며 수식 (3)으로 표현된다.

$$e_{(i,j)} = E_{(i,j)} - \hat{E}_{(i,j)} \quad (3)$$

전체 m개의 데이터에 대해서 특정 물리머신 j의 오차 제곱 합 SSE(Sum of Squares Error)인 $EnergyConsumptionSSE_j$ 는 다음의 수식 (4)으로 표현된다.

$$EnergyConsumptionSSE_j = \sum_{i=1}^m e_{(i,j)}^2 \quad (4)$$

따라서 최소 제곱법을 이용하여 선형 회귀 모델은 각각의 물리머신 j 마다 $EnergyConsumptionSSE_j$ 를 최소화하는 예측 모델의 회귀 계수 모수 b_0, b_1, \dots, b_k 를 찾음으로써 에너지 소모량 예측 모델이 완성된다.

특정 물리머신 j에 관련된 i번째 독립변수 데이터 집합이 주어졌을 때의 에너지소모량 예측값인 $\hat{E}_{(i,j)}$ 를 전체 데이터 센터에 존재하는 물리머신의 개수인 |PM| 만큼 계산해서 전체 합산하면 데이터 센터의 모든 물리머신이 소모한 에너지양인 $\hat{E}_{(i,DC)}$ 가 계산된다.

표 1. 에너지 소모량 예측을 위한 독립변수
Table 1. Independent variables for predicting energy consumption

Entity	Type	Variable	Feature	Description
PM	Power	x_1	Static Power	Leakage power
		x_2	Dynamic Power	Power consumed by using PM
		x_3	Startup Power	Power consumed to start first time
		x_4	Shutdown Power	Power consumed to turn off PM
	Resource	x_5	Total Number of cores	Total number of cores the PM has
		x_6	Total MIPS (Million Instructions per Second)	Sum of the MIPS of all cores
		x_7	Memory	Amount of memory
		x_8	Storage	Amount of storage
		x_9	Band Width	Amount of bandwidth
Deployed VMs	Request-ed Resource	x_{10}	Number of cores	Total number of cores
		x_{11}	Total MIPS	Total MIPS requested by VMs
		x_{12}	Memory	Total memory requested by VMs
		x_{13}	Storage	Total storage requested by VMs
		x_{14}	Band Width	Total bandwidth requested by VMs
Deployed user tasks	Request-ed Resource	x_{15}	Number of cores	Total number of cores
		x_{16}	Total MIPS	Total MIPS requested by tasks
		x_{17}	Memory	Total memory requested by tasks
		x_{18}	Storage	Total storage requested by tasks
		x_{19}	Band Width	Total bandwidth requested by tasks

이 전체 데이터 센터의 총 에너지 소모량 $\hat{E}_{(i,DC)}$ 가 최적화 알고리즘의 목적함수가 되며 수식으로는 (5)와 같이 표현된다.

$$\hat{E}_{(i,DC)} = \sum_{j=1}^{PM} \hat{E}_{(i,j)} \quad (5)$$

본 논문은 F1 Score로 학습된 모델의 성능을 분석했다. F1 Score는 재현도와 정밀도의 조화평균으로 만든 지표로서, 학습된 회귀 모형의 전반적인 성능을 측정하는 데 널리 이용된다¹³⁾. 보통 F1 Score는 0.75 이상의 값이 산출되면 의미 있는 모형으로 판명한다. 본 논문에서는 종속변수인 에너지 소모량에 대해서 <표 1> 대로 독립변수를 설정하고 CloudSim의 5,964 개의 관찰값에 대해서 regression 모형을 학습한 결과 0.881이 나왔다.

3.3 해 부호화

유전자 알고리즘에서 문제의 해는 특정 자료구조로 부호화되며 본 논문에서는 두 가지 부호화 방식을 제안한다.

3.3.1 물리 머신과 가상머신 식별자 기반의 정수 부호화

해는 정수 배열로 표현된다. 이때 배열의 전체 길이는 배치 요청을 위해 제출된 가상머신 리스트 크기 M과 배치 요청을 위해 제출된 태스크 리스트 크기 N-1의 합인 M+N-1이다. 이 배열의 인덱스 0부터 M-1까지에 해당하는 셀이 갖는 값의 범위는 0부터 현재 데이터 센터에 존재하는 물리머신 리스트 크기 L-1 까지다. 배열의 인덱스 0부터 M-1까지의 특정 셀의 값은 식별자가 i인 가상머신이 식별자가 j인 물리머신에 배치된다는 것을 의미한다. 인덱스 M 부터 M+N-1까지 배열의 셀이 갖는 값의 범위는 0부터 배치 요청을 위해 제출된 가상머신 리스트 크기 M-1 까지다. 인덱스 M 부터 M+N-1까지 배열의 셀이 갖는 값은 식별자가 p인 사용자 태스크가 식별자가 k인 가상머신에 배치된다는 것을 의미한다.

3.3.2 물리머신과 가상머신의 여유 CPU 자원량 순위 기반의 정수 부호화

지역 탐색하는 교배 연산의 특성을 고려해서 문제 공간(Problem Space)에서 탐색 공간(Searching Space)의 지역성을 잘 반영하는 것이 매우 중요하다. 문제 공간에서 지역적으로 조금 이동한 만큼 탐색 공간에서도 지역적으로 조금 이동해야 한다. 그렇지 않고 같이 문제 공간에서 지역적으로 조금 이동했는데 탐색 공간에서 많이 이동하면 지역 탐색하는 교배 연산의 의미가 저해된다. 이러한 문제를 해밍 절벽 문제

라고 한다.

직관적으로 문제 공간에서 가까운 지식을 포함하기 위해서 사전에 시스템에 존재하는 모든 물리머신을 여유 CPU 자원량이 많은 순으로 정렬하고, 배치 요청된 모든 가상머신도 요청 CPU자원량이 많은 순으로 정렬했다. 3.1과 동일하게 해는 정수 배열로 표현된다. 배열의 전체 길이는 배치 요청을 위해 제출된 가상머신 리스트 크기 M과 배치 요청을 위해 제출된 태스크 리스트 크기 N-1 의 합이다. 이 배열의 인덱스 0 부터 M-1까지에 해당하는 셀이 갖는 값의 범위는 0 부터 현재 데이터 센터에 존재하는 물리머신 리스트 크기 L-1 까지다. 이 배열의 인덱스 M 부터 M+N-1 까지 배열의 셀이 갖는 값의 범위는 0 부터 배치 요청을 위해 제출된 가상머신 리스트 크기 M-1까지다.

3.1.1과는 다르게 배열의 인덱스 0 부터 M-1까지의 특정 셀의 값은 식별자 i인 가상머신은 여유 CPU 자원량 순위 j인 물리머신에 배치된다는 것을 의미한다. 배열의 인덱스 M부터 M+N-1까지 배열의 셀이 갖는 값은 식별자가 p인 사용자 태스크가 여유 CPU 자원량 순위가 k인 가상머신에 배치된다는 것을 의미한다.

3.4 초기화

초기화 과정은 다음과 같다. 먼저 군을 생성하고 각 군에 가능한 초기 해를 생성한다. 초기 해를 생성할 때 가상머신을 임의의 물리머신에 배치하고 사용자 태스크를 임의의 가상머신에 배치한다. 이러한 해의 문제는 제약조건이 위반될 수 있다는 점이 있다. 빈역할 하는 물리머신이나 가상머신의 자원 수용 용량을 초과하는지 체크하고 만약 그렇다면 First fit을 기반으로 솔루션을 수정해서 재배치해주고, 그렇지 않다면 해를 군에 추가하고 위와 같은 과정을 사용자가 임의로 지정한 해가 구성된 군의 크기만큼 반복한다.

3.5 선택

앞서 초기화된 군에서 재생산 연산을 적용할 부모 2개를 선택해야 한다. 본 논문에서는 바이너리 토너먼트 선택 방법을 이용하였다. 바이너리 토너먼트 선택 과정은 먼저 2개의 솔루션을 임의로 선택하고 임의의 [0,1] 범위의 난수를 발생시킨다. 이때 난수 값이 특정 한계값보다 작으면 두 염색체 중 적합도가 높은 것을 부모로 선정하고 그렇지 않다면 적합도가 낮은 것을 부모로 선정한다.

3.6 교배

유전자 알고리즘의 교배 연산으로 시뮬레이티드 바

이더리 교배 (Simulated Binary Crossover) 연산을 사용하였다¹²⁾. 3.5 에서 선택한 두 개의 부모 개체들로부터 두 개의 자식 해를 특정 확률 분포를 기반으로 생성한다. 생성하는 방법은 수식으로 (6)과 같다.

$$\begin{aligned} c_1 &= \bar{x} + 0.5\bar{\alpha}(p_2 - p_1) \\ c_2 &= \bar{x} + 0.5\bar{\alpha}(p_2 - p_1) \\ by, \bar{x} &= 0.5(p_2 + p_1), p_2 > p_1 \end{aligned} \quad (6)$$

p_1 과 p_2 는 선택 된 두 부모 염색체 배열의 동일한 특정 인덱스에 해당하는 셀의 값을 의미하고, 이때 p_2 는 p_1 보다 크다고 가정한다. c_1 과 c_2 는 자식 염색체 배열의 동일한 특정 인덱스에 해당하는 셀의 값이며 특정 인덱스는 p_1 과 p_2 에 해당하는 부모 염색체 배열의 인덱스와 같다.

$\bar{\alpha}$ 는 spreading factor라고 한다. $\bar{\alpha}$ 가 1보다 작을 때 축소 교배 (Contracting Crossover) 효과가 발생하며 p_1 과 p_2 지점 사이에 c_1 과 c_2 가 결정된다. $\bar{\alpha}$ 가 1보다 클 때 확장 교배 (Expanding Crossover)로 불리며 p_1 과 p_2 지점 바깥에 c_1 과 c_2 가 결정된다. $\bar{\alpha}$ 가 1과 같을 정적 교배 (Station-ary Crossover)로 불리며 p_1 과 p_2 와 같은 지점에 c_1 과 c_2 가 결정된다. $\bar{\alpha}$ 는 다음과 같은 확률분포를 따르는 랜덤변수 α 의 특정 값이다.

$$c(\alpha) = 0.5(n+1) \frac{1}{\alpha^{(n+2)}}, \alpha > 1 \quad (7)$$

$$c(\alpha) = 0.5(n+1)\alpha^n, \alpha \leq 1 \quad (8)$$

랜덤 변수 α 의 확률분포의 주요 특징으로 n 이 커질수록 자식의 해는 부모의 값과 가까운 해를 더 높은 확률로 많이 생산된다. n 이 무한대로 커지면 정적 교배 결과로 자식의 해는 부모의 해와 동일해진다. 대부분의 논문에서는 n 은 2와 5사이로 하며 본 논문에서 n 을 2로 설정했다.

3.7 돌연 변이

본 논문에서의 돌연 변이 연산에서 제안한 다항 돌연 변이 (Polynomial Mutation) 연산¹²⁾을 사용했다. 부모의 해로부터 돌연변이 연산자가 적용된 자식 해는 다음과 같은 과정으로 계산된다.

- 1) 구간 [0,1]에서 유니폼 분포를 따르는 랜덤 변수에 대해서 임의의 값 r 을 생성한다.
- 2) 다음의 수식 (9)에 따라서 δ 를 계산한다.

$$\delta = \begin{cases} 2r^{\frac{1}{q+1}} - 1, & \text{if } r < 0.5 \\ 1 - (2-2r)^{\frac{1}{q+1}}, & \text{if } r \geq 0.5 \end{cases} \quad (9)$$

- 3) 부모의 해 P로부터 돌연 변이된 자식 해 C는 (10) 수식으로 계산된다. Δ 는 프로그래머가 임의로 정의하는 숫자이며 본 논문에서는 값을 1.5 로 설정하였다.

$$C = P + \delta \times \Delta \quad (10)$$

IV. 실험

본 장에서는 시뮬레이션은 CloudSim¹⁰⁾시뮬레이터를 이용하여 실험을 진행하였다. 실험의 워크 로드는 PlanetLab 시스템 기반에서 구축된 실제 워크 로드를 기반으로 진행되었다. 실험의 시나리오는 두 가지로 나뉘서, 데이터 센터에 존재하는 모든 물리 머신의 종류가 같은 시나리오와 모든 물리 머신의 종류가 다른 시나리오로 진행했다. 기존 논문에 제시된 MBFD와 PABFD, 기존 에너지 소모량 최적화 모델과 시뮬레이터드어널링을 결합한 방식의 알고리즘을 구현하여 본 논문의 성능비교 대상으로 사용하였다. 그리고 각 시나리오에 대해서 가상머신의 개수를 늘려가면서 각 비교 알고리즘의 에너지 소모량 결과 순위가 동일한지 확인을 하기 위한 실험을 진행하였고 실행 시간에 대해서도 비교하고 평가를 진행하였다.

4.1 워크로드 트레이스

PlanetLab은 전세계적인 연구자들이 네트워크 서비스 관련 연구를 쉽게 진행할 수 있도록 만들어진 네트워크 클라우드 기반의 테스트베드 시스템이다. CoMon은 PlanetLab 위에 구축된 모니터링 시스템이다. 워크로드 데이터는 전 세계에 걸쳐 500개 이상의 지역에 1000대 이상의 가상머신에 의해서 5분 간격으로 수집한 CPU 자원 활용도 데이터가 담겨 있다. 2010년 3월에서 4월까지 임의의 10일을 선정해서 데이터는 수집되었다. 본 논문은 이러한 10일 치의 PlanetLab Workload Trace 데이터를 이용해서 제안 알고리즘과 기존 알고리즘의 성능을 비교하였다.

4.2 데이터센터의 물리노드 종류가 동질적인 상황
 데이터 센터에서 모든 물리머신들의 종류가 동일한
 상황을 가정하고 실험을 진행하였다. 실험을 구성한
 데이터센터의 전반적인 환경은 표 2와 같다.

가상 머신이 100개인 상황에서 에너지 소모량에 대
 한 실험 결과는 그림 2와 같다. 물리머신이 동질적인
 상황에서 기존의 에너지 소모량 모델의 특별한 장점
 은 없기 때문에 MBFD는 PABFD와 실험 결과 동일
 하다. 이어서 기존 에너지 모델에 시뮬레이티드 어닐
 링 최적화 알고리즘을 결합한 방식이 실험 결과
 MBFD와 PABFD보다 에너지 소모량과 관점에서 우
 수했다. 그 이유는 태스크와 가상머신을 같이 최적 배
 치를 했기 때문이고 또한 지역 최적해에 수렴하는
 Best fit과는 다르게 시뮬레이티드 어닐링 방식은 지
 역 최적해의 수렴에 벗어날 수 있기 때문이다. 이어서
 제안 에너지 모델과 기존 에너지 모델을 비교하기 위
 해서 같은 최적화 알고리즘인 시뮬레이티드 어닐링으
 로 실험을 진행하였다. 실험 결과 제안 에너지 모델에
 시뮬레이티드 어닐링 최적화 알고리즘을 결합한 방식
 이 기존 에너지 모델에 시뮬레이티드 어닐링 최적화
 알고리즘보다 에너지를 효율적으로 소모했기 때문
 이다. 이어서 제안 에너지 모델이 기존 에너지 모
 델 보다 더 우수함을 알 수 있다. 이어서 제안 에너지

표 2. 동질적인 시나리오에서 실험 파라미터와 환경 설정
 Table 2. Homogeneous scenario parameters and
 configuration

Parameters	Configuration
PM types	HP pro Liant ML110 G4 (2 x 1800 MIPS)
Number of PMs	800
Number of VMs	100, 300, 500, 700
VM types (25% of each VM type)	2500MIPS, 2000MIPS, 1500MIPS, 1000MIPS
Workload	PlanetLab (10 days of traces)

모델에 시뮬레이티드 어닐링 최적화 알고리즘 대신
 유전자 최적화 알고리즘을 사용하였더니 데이터센터
 의 에너지를 더 적게 소모하였다. 유전자 최적화 알고
 리즘은 부호화 과정을 3.2.1의 방식과 3.2.1의 해밍 절
 벽 문제를 개선한 3.2.2의 방식으로 각각 진행하였다.
 실험 결과 3.1.2.1에서 기술한 물리머신과 가상머신
 식별자 기반의 정수 부호화 방식의 부호화 보다
 3.1.2.2 에서 기술한 물리머신과 가상머신의 여유
 CPU 자원량 정렬 기반의 정수 부호화 방식이 에너지를
 효율적으로 소모하였다. 마지막으로 가상머신을
 100개, 300개, 500개, 700개로 늘려도 순위는 변동되

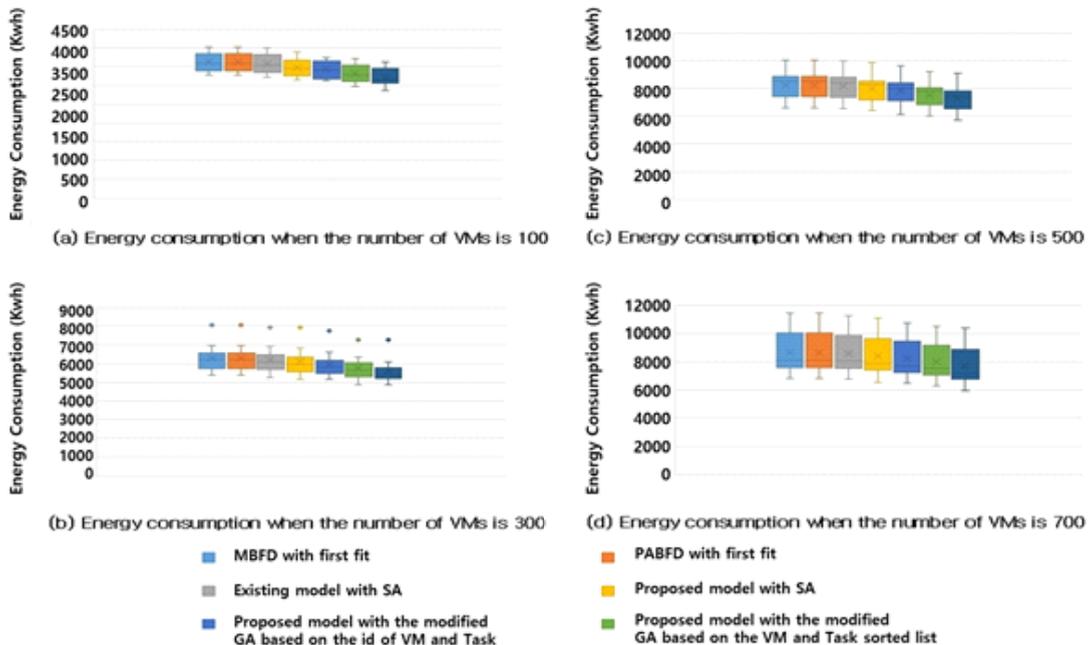


그림 2. 데이터 센터가 동질적인 상황에서 가상 머신 개수가 100, 300, 500, 700개 일 때 에너지 소모량
 Fig. 2. Energy consumption when the number of VMs is 100, 300, 500, or 700 in a homogeneous data center

지 않았다.

4.3 데이터센터의 물리노드 종류가 이질적인 상황

데이터 센터에서 물리머신들의 종류가 4가지로 다른 상황을 가정하고 실험을 진행하였다. 실험을 구성한 데이터센터의 전반적인 환경은 표 3과 같다.

가상 머신이 100개인 상황에서 에너지 소모량을 실험한 결과는 그림 3의 (a)이다. 실험 결과, MBFD와는 달리 PABFD는 물리 머신의 다양한 특성을 고려해서 최적화 에너지 모델을 정의했기 때문에 에너지를 더 적게 소모함을 확인할 수 있었다. 이때 MBFD와 PABFD는 가상 머신 배치 알고리즘이며 태스크 배치 알고리즘으로는 First fit을 사용하였다. 이어서 가상 머신과 태스크 배치를 같이 고려하며 기존 에너지 모델에 시뮬레이티드 어닐링 최적화 알고리즘을 결합한 방식이 PABFD보다 에너지 소모량이 적었다. 그 이유는 데이터 센터의 에너지 소모량 관점에서 태스크와 가상머신을 같이 최적 배치를 하며 지역 최적해에 수렴하는 First fit 과는 다르게 시뮬레이티드 어닐링은 지역 최적해의 수렴이 벗어날 수 있기 때문이다. 이러한 기존 에너지 모델에 시뮬레이티드 어닐링 알고리즘을 결합한 방식과 본 논문에서 제안한 에너지 모델에 시뮬레이티드 어닐링 알고리즘을 결합한 방식을 비교하였다. 실험 결과 같은 종류의 최적화 알고리즘

표 3. 이질적인 시나리오에서 실험 파라미터와 환경 설정
Table 3. Heterogeneous scenario parameters and configuration

Parameters	Configuration
PM types	HP pro Liant ML110 G4 (2 x 1800 MIPS) HP pro Liant ML110 G5 (2 x 2660 MIPS) Quad-core IBM Xeon (4 x 3067 MIPS) Dual-core HP ProLiant (4 x 2933 MIPS)
Number of PMs	800 (25% of each PM type)
Number of VMs	100, 300, 500, 700
VM types (25% of each VM type)	2500MIPS, 2000MIPS, 1500MIPS, 1000MIPS
Workload	PlanetLab (10 dias of traces)

을 사용했기 때문에 본 논문의 에너지 소모량 최적화 모델이 기존의 에너지 소모량 최적화 모델보다 더 우수함을 알 수 있었다. 이어서 제안 에너지 모델에 시뮬레이티드 어닐링 최적화 알고리즘 대신 유전자 최적화 알고리즘을 사용하였더니 더 에너지를 효율적으로 소모하였다. 유전자 알고리즘에서 3.2.1 에서 기술

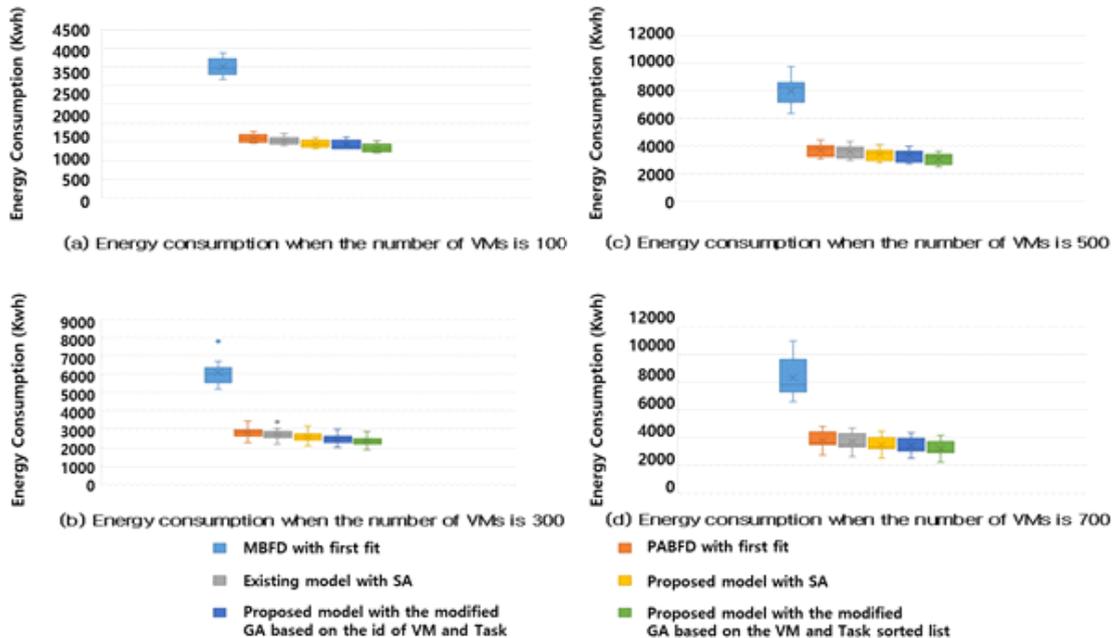


그림 3. 데이터 센터가 이질적인 상황에서 가상 머신 개수가 100, 300, 500, 700개 일 때 에너지 소모량
Fig. 3. Energy consumption when the number of VMs is 100, 300, 500, or 700 in a heterogeneous data center

한 물리머신과 가상머신 식별자 기반의 정수 부호화 방식의 부호화 보다 3.2.2 에서 기술한 물리머신과 가상머신의 여유 CPU 자원량 정렬 기반의 정수 부호화 방식이 더 에너지 소모량 관점에서 효율적이다. 마지막으로 가상머신을 100개, 300개, 500개, 700개로 늘려도 실험 결과에서 순위는 변동되지 않았음을 확인했다.

4.4 제안 알고리즘과 기존 알고리즘의 실행시간 비교 분석

표 4는 4.2 절에서 데이터센터의 물리노드가 동질적인 상황에서의 VM 개수가 100개인 상황을 가정하고 PlanetLab 워크로드를 10번의 실험을 진행하여 실행 시간을 평균 낸 결과이다. 실험 결과 PABFD는 기존의 MBFD에 에너지 소모량 모델의 계산 시간이 추가되어 MBFD보다 실행 시간이 더 오래걸렸다. 이어서 기존 에너지 모델에 시뮬레이티드 어닐링 방식도 최적화한 알고리즘 자체의 복잡성 때문에 MBFD와 PABFD보다 더 오랜 실행 시간이 걸렸다. 본 논문에서 제안하는 에너지 소모량 모델은 선형 회귀 모델 기반으로 추론하기 때문에 기존의 에너지 소모량 모델보다 계산하는 비용이 크다는 단점이 있다. 따라서 본 논문의 에너지 소모량 모델에 시뮬레이티드 어닐링을 접목한 방식이 기존 에너지 모델에 시뮬레이티드 어닐링을 접목한 방식보다 더 오랜 실행 시간이 걸렸다. 그리고 제안 에너지 소모량 모델에 유전자 알고리즘을 접목한 방식은 교배, 돌연변이 같은 복잡한 탐색 연산 때문에 제안하는 에너지 소모량 모델에 시플레

이티드 어닐링 알고리즘을 접목한 방식 보다 실행 시간이 오래 걸렸다. 3.2.1에서 기술한 물리머신과 가상머신 식별자 기반의 정수 부호화방식의 부호화방식에서 해밍절벽 문제를 개선한 3.2.2에서 기술한 물리머신과 가상머신의 여유 CPU 자원량 정렬 기반의 정수 부호화방식이 에너지 소모량 관점에서 효율적이었지만 실행 시간도 단축되어 우수함을 확인할 수 있었다.

V. 결 론

본 논문에서는 기존 에너지 소모량 모델의 단점을 개선한 새로운 에너지 소모량 모델을 제안하였다. 제안한 에너지 소모량 최적화 모델에 적용 가능한 최적화 알고리즘으로 기존의 유전자 알고리즘을 수정한 최적화 알고리즘을 제안하였다. 제안한 최적화 모델과 알고리즘과 기존의 최적화 모델과 알고리즘의 성능을 비교하기 위해서 Cloudsim 시뮬레이터 환경에서 실험을 진행하였다. 비교 알고리즘으로 기존 논문인 MBFD 방식과 PABFD, 기존의 에너지 모델에 시뮬레이티드 어닐링 알고리즘을 결합한 방식, 본 논문에서 제안한 에너지 모델에 시뮬레이티드 어닐링 알고리즘을 결합한 방식과 유전자 알고리즘을 결합한 방식을 선정했다.

비교 결과 물리머신이 동질적인 상황에서 MBFD는 PABFD와 실험 결과 동일하다. 하지만 이질적인 상황에서 PABFD는 다양한 물리머신의 특성을 고려하여 에너지를 모델링하여 MBFD보다 더 효율적으로 에너지를 소모함을 알 수 있었다. 그리고 이질적인 상황과 동질적인 상황 모두 기존 에너지 모델에 시플레이티드 어닐링 최적화 알고리즘을 결합한 방식이 MBFD와 PABFD 보다 에너지 소모량 관점에서 우수했다. 그 이유는 태스크와 가상머신을 같이 최적 배치를 하고 지역 최적해 수렴에 벗어나기 때문이다. 또한, 제안 에너지 모델과 기존 에너지 모델을 비교하기 위해서 같은 모델에 시플레이티드 어닐링 최적화 알고리즘을 결합해서 비교하였더니 제안 에너지 모델에 시플레이티드 어닐링 최적화 알고리즘을 결합한 방식이 더 에너지를 효율적으로 소모함을 알 수 있었다. 따라서 본 논문에서 제안하는 에너지 소모량 모델이 기존의 에너지 소모량 모델보다 더 정확하게 모델링 된 것을 알 수 있었다. 제안 에너지 모델에 시플레이티드 어닐링 최적화 알고리즘 대신 유전자 최적화 알고리즘을 사용하였더니 성능이 개선되었다. 또한, 시플레이티드 어닐링 알고리즘 대신에 유전자 알고리즘을 사용하여 실험을 진행하였고, 유전자 알고리즘을 두

표 4. 기존 연구의 알고리즘과 제안 알고리즘의 평균 실행시간 비교
Table 4. Comparison of the average execution time of the existing algorithm and the proposed algorithm

Algorithms	Average Running Time (ms)
MBFD with first fit	9.6
PABFD with first fit	10.02
Existing model with SA	48.23
Proposed model with SA	87.08
Proposed model with the modified GA based on the id of VM and Task	413.76
Proposed model with the modified GA based on the sorted list of VM and Task	121.7

가지 3.2.1과 3.2.2에서 기술한 부호화 방식으로 나눠 실험을 진행했다. 실험 결과 3.1.2.2 에서 기술한 물리 머신과 가상머신의 여유 CPU 자원량 정렬 기반의 정수 부호화 방식은 3.1.2.1에서 기술한 물리머신과 가상머신 식별자 기반의 정수 부호화 방식의 단점인 해밍 절벽 문제를 개선했기 때문에 에너지 소모량 관점에서 효율적이고 실행 시간도 단축되고 에너지 소모량도 더 효율적임을 확인했다.

하지만 본 논문에서 제안하는 방안이 PABFD에 비해서 에너지 소모량이 적지만 실행 시간은 더 오래 걸린다. 이러한 단점으로 만약 실제 산업에서 클라우드의 운영자가 실시간으로 알고리즘을 동작하는데 소요된 시간이 사용자 서비스 운영에 지장을 주어 용인할 수 없는 수준이 된다면 문제가 될 수 있다. 이러한 문제점에 대한 후후 보완 방법으로 유전자 알고리즘의 실행 속도를 개선할 수 있는 방법이 필요하다. 후속 연구로 유전자 알고리즘을 병렬로 계산하여 실행시간을 개선하는 방법에 관해서 연구하고자 한다.

References

- [1] 이주영, “클라우드 컴퓨팅의 특징 및 사업자별 제공 서비스 현황,” *IIPR*, vol. 22, no. 6, Apr. 2010.
- [2] S. L. Sams, *Discovering hidden costs in your data centre - a CFO perspective (2011)*, Retrieved Jul. 22, 2014, from ibm.com/services/siteandfacilities.
- [3] D. Meisner, B. T. Gold, and T. F. Wenisch, “PowerNap: Eliminating server idle power,” *ASPLOS*, vol. 37, no. 1, pp. 205-216, 2009. (<https://doi.org/10.1145/2528521.1508269>)
- [4] A. Beloglazov and R. Buyya, “Open-stack neat: A framework for dynamic and energy efficient consolidation of virtual machines in openstack clouds,” in *Concurr. Comput. Pract. Exp.*, 2014. (<https://doi.org/10.1002/cpe.3314>)
- [5] A. Beloglazov and R. Buyya, “Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers” in *Concurr. Comput. Pract. Exp.*, 2012. (<https://doi.org/10.1002/cpe.1867>)
- [6] K. Dubey, S. C. Sharma, and A. A. Nasr, “A simulated annealing based energy efficient VM placement policy in cloud computing,” *IC-ETITE*, pp. 1-5, 2020. (https://doi.org/10.1109/ic-ETITE47_903.2020.119)
- [7] L. Huang, H. Chen, and T. Hu, “Survey on resource allocation policy and job scheduling algorithms of cloud computing,” *JSW*, vol. 8, no. 2, pp. 480-487, 2013. (<https://doi.org/10.4304/jsw8.2.480-487>)
- [8] R. den Bossche, K. Vanmechelen, and J. Broeckhove, “Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads in Cloud Computing,” *IEEE 3rd Int. Conf. Cloud Comput.*, pp. 228-235, 2010. (<https://doi.org/10.1109/CLOUD.2010.58>)
- [9] R. Kaur and S. Kinger, “Analysis of job scheduling algorithms in cloud computing,” *Int. J. Comput. Trends Technol.*, vol. 9, no. 7, pp. 379-386, 2014. (<https://doi.org/10.14445/22312803/IJCTT-V9P169>)
- [10] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, “CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” *Softw. Pract. Exp.*, vol. 41, no. 1, pp. 23-50, 2011. (<https://doi.org/10.1002/spe.995>)
- [11] M. Gen, *Genetic Algorithms & Eng. Optimization*, John Wiley & Sons, 2000.
- [12] K. Deb and R. B. Agrawal, “Simulated binary crossover for continuous search space,” *Complex Syst.*, vol. 9, no. 2, pp. 115-148, 1995.
- [13] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Inf. Process. Manag.*, vol. 45, pp. 427-437, 2009. (<https://doi.org/10.1016/j.ipm.2009.03.002>)

원 등 옥 (Dong-wook Won)



2020년 2월 : 광운대학교 전자
통신공학과
2020년 8월 : 광운대학교 전자
공학과 석사
2023년 3월~현재 : 중앙대학교
컴퓨터공학과 박사과정

<관심분야> 클라우드 컴퓨팅, 분산시스템
[OCID:0009-0002-1781-1419]

김 화 성 (Hwa-sung Kim)



1981년 2월 : 고려대학교 전자
공학과
1983년 2월 : 고려대학교 전자
공학과 석사
1996년 : Lehigh Univ. 전산학
박사

1984년 3월~2000년 2월 : ETRI 책임 연구원
2000년 3월~현재 : 광운대학교 전자통신공학과 교수
<관심분야> 분산 및 병렬 컴퓨팅, 가상화 기술
[OCID:0000-0001-5893-5691]