

# 워크플로우 어플리케이션을 위한 저전력 가상 자원 할당 기법

이준영<sup>o</sup>, 강혜정, 김가희, 김윤희

숙명여자대학교 컴퓨터과학부

{pinekiss07, hjkang, ghk511, yulan}@sm.ac.kr

## 요 약

전 세계적으로 그린 컴퓨팅에 대한 관심이 높아지면서, 높은 컴퓨팅 수준을 보장하면서 데이터 센터의 전력 소모량을 줄이는 전력 고려 자원 관리 기법에 대한 연구가 활발히 진행 중에 있다. 하지만 기존 연구의 경우, 클라우드 환경에서 연구가 이루어지지 않았거나, 사용자의 요구조건은 만족하지 못하거나, 혹은 연구 대상이 되는 어플리케이션 유형에 대한 일반화가 부족하였다. 따라서 본 논문에서는 클라우드 컴퓨팅 환경에서 상호간 의존성이 존재하여 수행 순서를 지켜주어야 하는 서브 태스크들로 구성된 워크플로우 어플리케이션을 위한 가상 자원 할당 기법을 제안한다. 제안하는 프로비저닝 기법은 사용자의 데드라인 요구조건을 위반하지 않으면서 Auto-scaling 기술을 통해 사용자의 비용을 축소할 뿐만 아니라, Dynamic Voltage and Frequency Scaling (DVFS) 기술을 적용한 전력 소모량 최소화를 통해 플랫폼 관리자의 비용의 감소 또한 기대할 수 있다.

## 1. 서론

클라우드 컴퓨팅의 출현으로, 대규모 가상화 컴퓨팅과 데이터 센터는 IT 산업에서 중요한 요소로 자리잡게 되었다. 이로 인해 좀 더 높은 수준의 컴퓨팅이 가능하게 되었지만, 데이터 센터의 증가로 전력 소모량이 급증하였다. 데이터 센터의 컴퓨팅 머신들이 대기상태에 있더라도 소규모 도시가 소비하는 전력량에 달하는 많은 양의 전력을 필요로 하고, 뿐만 아니라 서버들을 적정 온도로 유지하기 위해서는 마찬가지로 비슷한 양의 전력으로 쿨링 시스템을 가동해야 한다. 이러한 요인들은 CO2 발생량을 증가시키고 현재 이슈가 되고 있는 지구 온난화를 가속화하게 된다. 그래서 고성능 컴퓨팅의 새로운 트렌드인 그린 컴퓨팅으로 이러한 문제를 해결하고자 한다. 그린 컴퓨팅은 높은 성능 수준 달성과 전력 소모량 절감을 통하여 전체 시스템의 효율을 최대화한다. 최근 활발히 연구되고 있는 저전력 기술 중 가장 대표적인 것으로는 Dynamic Voltage and Frequency Scaling (DVFS) 기술이 있다[1][2]. DVFS 기술은 시스템이나 시스템에 연결된 장치들이 동작하는 동안에는 소비 전력을 줄일 수 없는 기술의 한계를 극복하기 위해 제안된 것으로 머신의 동작 속도를 줄이는 대신 소비 전력에서 이득을 취하는 방식의 기술이다. DVFS 는 CPU 에서 소비되는 에너지 E 가 주파수  $f^2$ 에 비례한다는 점을 이용하여 CPU clock frequency 를 줄여 소비 전력을 줄인다.

Auto-scaling 기법은 클라우드 컴퓨팅의 특성을 활용하여 수행 중인 작업 및 자원의 지속적인 모니터링을 통해 자원 규모를 확장/축소한다. 이 기법을

기반으로 한 자원 프로비저닝에 대한 여러 가지 연구가 활발하게 진행되고 있는데, 사용자의 데드라인 요구 조건을 만족하면서 사용자 비용의 최소화까지 고려하는 스케줄링 기법에 대한 연구도 존재한다[3][4]. 더 나아가, 전력을 고려하는 가상 자원 할당에 대한 연구도 존재한다[5][6][7]. 특히, 선행 연구[7]에서는 클라우드 컴퓨팅 환경에서 서브 태스크간 의존성이 존재하지 않는 Bag of tasks 어플리케이션을 대상으로 자원 할당 자동화 기법을 이용한 동적 부하 관리에 대한 연구를 진행하였다. 이를 통해 사용자의 요구 조건을 충족하면서 클라우드 플랫폼 관리자의 비용을 절감하였지만, 서브 태스크들 간의 의존성이 없는 Bag of tasks 어플리케이션을 대상으로 하였으므로 연구 적용 대상에 대한 일반화가 부족하다고 할 수 있다.

본 논문에서는 선행 연구를 확장하여 클라우드 컴퓨팅 환경에서 서브 태스크간 의존성이 존재하는 워크플로우 어플리케이션을 대상으로 하여 자원 할당 자동화 기법을 이용하여 작업의 데드라인을 보장하는 동적인 부하 관리 기법을 제안한다. 사용자의 데드라인 요구 조건을 만족함과 동시에 DVFS 기술을 적용한 저전력 할당 기법을 통하여 플랫폼 관리자의 비용도 절감한다. 그리고 워크플로우 어플리케이션을 대상으로 연구를 진행하여 기존의 선행 연구보다 다양한 형태의 어플리케이션에 대해 기법을 적용할 수 있도록 하였다.

본 논문의 2 장에서는 관련 연구에 대해 다루고, 3 장에서 저전력 가상 자원 동적 할당 기법을 설명한다. 마지막 장에서는 결론을 맺는다.

## 2. 관련연구

클라우드 컴퓨팅 환경에서 가상화 기술의 도입으로 자원의 무한한 제공이 가능해지면서 무한한 자원들을 동적으로 할당할 수 있는 기법에 대한 연구가 활발하게 진행되고 있다. AWS(Amazon Web Service)[8]의 Auto-Scaling 기술은 자원 할당 자동화 서비스의 대표적인 예로, 사용자가 정의한 규칙을 기반으로 자원의 규모를 결정한다. 자원 규모의 확장과 축소는 하드웨어의 성능과 관련된 수치의 상한 값과 하한 값을 미리 정해 놓고 그 수만큼 가상 머신(Virtual Machine: VM)을 추가하거나 제거하는 방식으로 이루어진다. 그러나 이 방식은 데드라인과 같은 사용자의 요구사항을 고려하지 않는다는 단점이 있다.

Auto-scaling 기술을 적용하여 데드라인과 사용자 비용을 고려한 자원 할당 기법인 [3]연구는 이러한 단점을 보완한 방식으로, 작업의 데드라인 및 실제 클라우드 컴퓨팅 환경의 요소들을 고려해 사용자 비용 절감을 가능하게 한다. 하지만 클라우드 플랫폼 관리자의 비용은 고려하지 않았다. 이를 보완한 형태의 [5]연구는 작업의 데드라인을 고려하면서 전력 소모량까지 동시에 고려한 자원 할당 기법을 제안하였다. 이 기법은 사용자의 비용을 낮추고 데드라인 요구사항을 지켜 사용자 만족도를 높이는 것은 물론, 전력 사용량 감축을 통한 관리자 측면에서의 비용 또한 줄여준다. 이를 위해 가상 머신이 물리 머신(Physical Machine: PM)에 프로비저닝 될 때 발생하는 전력 소모량을 고려하고 가상 머신의 가격이 같을 경우 전력 소모량이 적은 자원을 할당해 준다. 이 기법은 태스크의 요구 성능에 따라 가상 머신을 제공하기 때문에 사용자에게 불필요한 비용 부담을 줄 수 있다는 단점이 있다. [5]연구의 단점을 보완하는 [7]연구는 클라우드 컴퓨팅 환경의 실제 제약 요소들과 전력 소모량을 함께 고려하여 사용자와 플랫폼 관리자, 둘 모두의 비용을 최소화하는 자원 할당 기법을 제안하였다. 그러나 이 기법은 태스크들간 상호 의존성이 존재하지 않는 Bag of tasks 어플리케이션을 대상으로 하였기 때문에, 주로 워크플로우 형태의 과학 어플리케이션에 기법을 적용할 수 없다는 한계점이 있다. 그러므로 이 연구는 기법을 적용할 수 있는 어플리케이션이 한정되어 있으므로 적용 대상의 일반화가 부족하다고 할 수 있다. [9]연구는 [7]연구의 단점인 Bag of tasks 어플리케이션으로 제한을 두지 않고 서브 태스크들 사이에 상호 의존성이 존재하는 워크플로우 형태의 어플리케이션을 대상으로 스케줄링 기법을 제안한다. 하지만, 클라우드 환경이 아닌 그리드 환경을 기반으로 하였다. 이를 보완한 [4]연구는 [9]연구와 달리 클라우드 환경에서 워크플로우 어플리케이션을 위한 자동화 자동 할당 기법을 제안하지만, 사용자의 비용만 고려할 뿐, 전력 소모량을 고려하지 않았다.

본 논문에서는 앞서 언급한 관련연구들의 단점

인 적용 대상의 일반화 부족을 보완하여 사용자의 데드라인 요구사항을 만족하면서 사용자와 클라우드 플랫폼 관리자, 모두의 비용을 축소할 수 있는 워크플로우 형태의 어플리케이션을 대상으로 한 저전력 자원 할당 기법을 제안한다.

## 3. 저전력 동적 가상 자원 할당 기법

### 3.1 서비스 구조

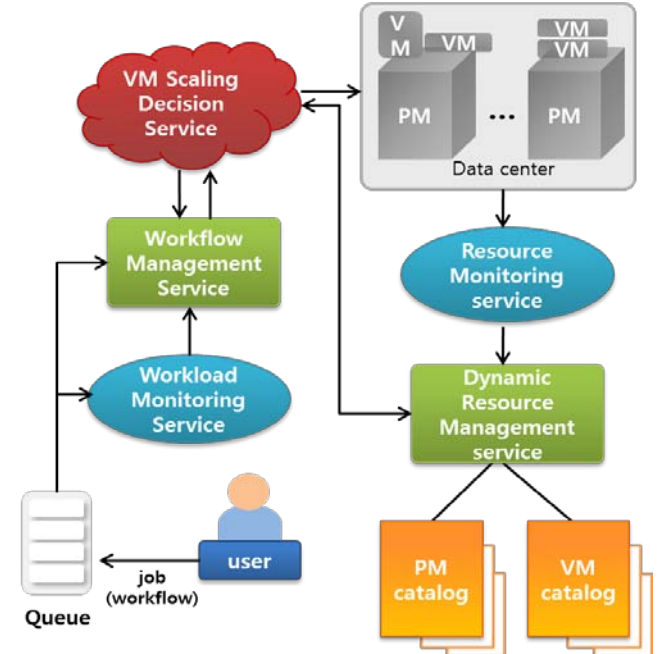


그림 1. 서비스 구조도

본 논문에서 제안하는 클라우드 자원의 동적 할당 자동화 기법을 지원하는 프레임워크의 구조는 그림 1 과 같으며, 이는 선행연구[7]의 구조도를 추가 및 수정하여 구성하였다.

**VM Scaling Decision Service(VMDS 서비스)** - 주변 서비스 모듈들로부터 상호작용으로 얻은 정보를 동적인 자원 할당 자동화에 이용하는 핵심 서비스 모듈이다. 가상 머신의 프로비저닝 및 작업 스케줄링을 수행한다.

**Workflow Management Service(워크플로우 관리 서비스)** - Queue 에 대기중인 작업들의 각 요구 수행 시간 및 요구 자원 성능 정보를 관리한다. 그리고 저장된 작업 정보를 이용하여 DAG(Directed Acyclic Graph)형태로 정리된 작업의 critical path 를 결정하고, 우선 순위에 따라 ‘클러스터링’ 작업을 거쳐 각 서브 태스크들의 몇 개의 클러스터로 묶어준다. 또한, 각 서브 태스크들의 개별 데드라인을 설정하는 서비스를 제공한다. 그리고 VMSD 서비스에서 결정된 자원 할당 스케줄에 따라 작업을 자원에 제출하는 서비스를 제공한다.

**Dynamic Resource Management Service(동적 자원 관리 서비스)** - 물리 머신과 가상 머신을 모두

관리하는 서비스로서, VMSD 서비스에서 결정된 자원 할당 스케줄링에 따라 가상 머신을 추가하고 제거한다.

**Workload Monitoring Service**(작업 부하 모니터링 서비스) – Queue 에 존재하는 아직 스케줄링 되지 않은 작업 부하를 모니터링하고, 작업 부하 정보를 워크플로우 관리 서비스에 전달하는 서비스를 제공한다.

**Resource Monitoring Service**(자원 모니터링 서비스) – 물리 머신과 가상 머신을 포함하는 자원의 실시간 상태를 모니터링하고, 모니터링한 정보는 동적 자원 관리 서비스에 전달한다.

**Queue** – 자원 할당이 될 때까지 사용자가 제출한 작업은 Queue 에 저장된다.

**PM Catalog**(물리 머신 카탈로그) – 물리 머신의 정보를 제공한다. 전력 소모량 측정에 필요한 물리 머신의 성능 정보 및 주파수 정보를 가지고 있다.

**VM Catalog**(가상 머신 카탈로그) – 클라우드 플랫폼에서 제공하는 가상 머신 정보를 지니고 있으며, 가상 머신 유형별로 성능과 가격에 대한 정보를 제공한다.

### 3.2 서비스 구조 시나리오

본 논문에서 제안한 자원 동적 할당 자동화 기법은 다음과 같은 시나리오로 수행되며, 이 시나리오의 선행연구[7]의 시나리오를 추가 및 수정하여 구성하였다.

- 1) 사용자가 Queue 로 워크플로우 형태의 작업을 제출한다.
- 2) 작업 모니터링 서비스가 작업의 제출을 워크플로우 관리 서비스에 알린다.
- 3) 워크플로우 관리 서비스는 작업 데드라인, 그에 따른 작업 별 자원 성능 요구(MIPS, Hz)와 같은 작업 정보를 넘겨준다.
- 4) 자원 할당 시간 간격이 돌아오면, VMSD 서비스가 자원 할당을 수행하는데, 작업 관리 서비스로부터 현재 대기중인 모든 작업 정보를 얻고 동적 자원 관리 서비스로부터 물리 머신과 가상 머신의 정보를 얻는다.
- 5) 자원 할당 결정을 시작하기 전에, DAG 형태로 작업의 태스크들의 수행 순서가 결정되어 있으므로, 그에 따라 critical path 를 결정한다. 그 외의 태스크들도 앞서 정해진 우선 순위에 따라 여러 개의 클러스터로 묶는다. 단, 같은 클러스터 내의 태스크들은 같은 자원에 할당된다.
- 6) 사용자가 요구한 작업 데드라인을 고려하여 각 클러스터 내의 태스크들의 개별 데드라인을 할당하여 태스크들의 수행 순서 및 수행 시간을 지킬 수 있도록 유도한다.
- 7) 각 가상 머신 유형별로 작업 부하를 계산하고 필요한 가상 머신의 개수를 확인한다.

- 8) 물리 머신의 확장/축소 여부를 결정하기 전에 이미 실행 중인 가상 머신들이 합병 가능한지 확인하고 합병을 수행한다. 단, 합병 수행 시 critical path 의 태스크들의 수행순서와 데드라인을 위반하지 않아야 한다.
- 9) 합병이 완료된 상태에서 가상 머신이 유형별로 물리 머신에 할당되었을 경우의 전력 소모량을 고려하여 확장할 머신과 제거할 머신을 결정한다.
- 10) 마지막으로 워크플로우 관리 서비스가 각 클러스터의 태스크들을 지정된 가상 머신에 할당한다.
- 11) 위의 1)~10) 과정은 각 자원 할당 시간 간격마다 반복되어 동적 자원 할당이 이루어진다.

### 3.3 전력 소모량 계산

본 절에서는 자원 규모 확장/축소를 결정할 때 전력 소모량을 고려하는 방법과 전력 소모량을 계산하는 방법에 대해서 설명한다. 주파수  $f$ 를 기반으로 한 물리 머신의 전력 소모량 계산은 [5]연구를 참고하였으며, 이는 ‘Power Model’ 식

$$E = \alpha \cdot t \cdot S^2 \quad (1)$$

을 사용하여 계산한다. 이 때,  $\alpha$ 는 전력 소모량 계산을 위한 계수로 1을 대입하여 사용하며,  $t$ 는 작업이 처리되는 수행 시간을 의미한다.  $f_{max}$ 를 물리 머신의 최대 주파수라고 하면,  $S$ 는  $f_{max}/f$ 로 표현할 수 있으며, 상대적 프로세서 속도를 나타낸다.

본 논문에서 전력 소모량을 고려하여 자원 할당을 하기 위해서 DVFS 기술을 사용한다. DVFS 기술은 머신이 동작하는 동안에도 자유롭게 머신의 주파수를 조정하여 전체 전력 소모량을 줄인다. 이와 같은 기술을 적용하여 적절한 주파수의 물리 머신에 결정된 스케줄링에 따라 가상 머신을 할당한다.

### 3.4 전력 소모량 비교

본 절에서는 전력 소모량을 고려하지 않고 물리 머신의 성능을 최대 사용하는 기법과 가상 머신 합병을 수행하기 전의 static PCH 기법과 본 논문에서 제안하는 가상 머신 합병을 적용한 기법을 사용하였을 때의 전력 소모량을 비교한다. 전력 소모량은 식(1)을 사용하여 계산하였다.

세 기법 모두 1200Hz를 갖는 하나의 물리 머신이 존재한다고 가정하였으며, 자원 할당을 필요로 하는 워크플로우 형태의 한 개의 작업  $J$ 는  $t_1, t_2, t_3, \dots, t_{10}$  총 10 개의 서브 태스크들로 이루어진다. 작업  $J$ 는 다음 그림 2와 같이 DAG 형태로 표현할 수 있으며, 그림 2의 워크플로우 작업 샘플은 [9]연구의 샘플을 참고하여 변형 및 축소하여 구성하였다. [9]연구의 static PCH기법에 따라 우선 순위를 고려

하여 Critical path를 결정하였고, ‘클러스터링’ 작업을 거쳐 3 개의 클러스터로 묶어 주었다. 사용자가 요구하는 데드라인은 145sec라고 가정하고, 각 태스크는  $(P_r, R_r)$ 로 표현한다. 각각 요구하는 자원 성능  $P_r$  과 해당 성능의 자원에서 태스크를 수행하였을 때의 수행시간(Response Time)  $R_r$ 로 나타낸다. 그림 2의 각 태스크의 밑에  $P_r$ 과  $R_r$ 을 표시하였다. 따라서 작업J의 태스크들을  $cls_1, cls_2, cls_3$  총 3 개의 클러스터로 묶었다고 하였을 때,  $cls_1=\{t_1, t_2, t_5, t_7, t_8, t_9, t_{10}\}$ ,  $cls_2=\{t_3\}$ ,  $cls_3=\{t_4, t_6\}$  으로 나타낼 수 있다.

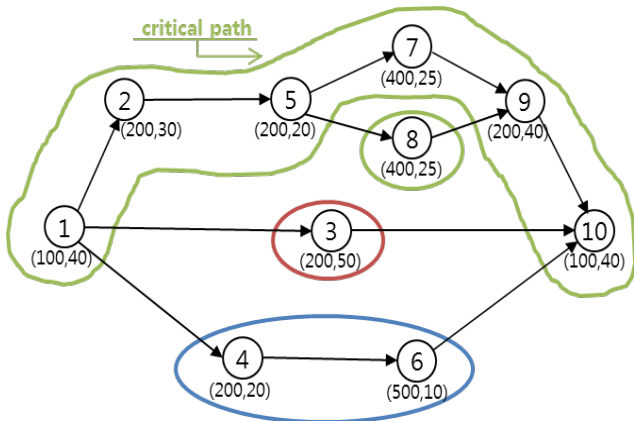
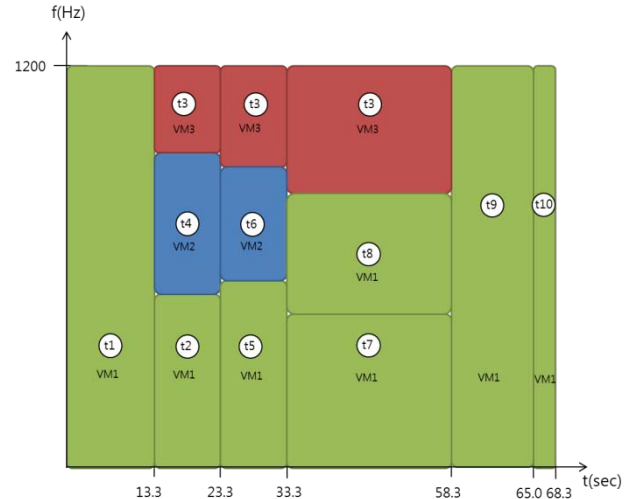


그림 2. 작업 J의 Sample DAG

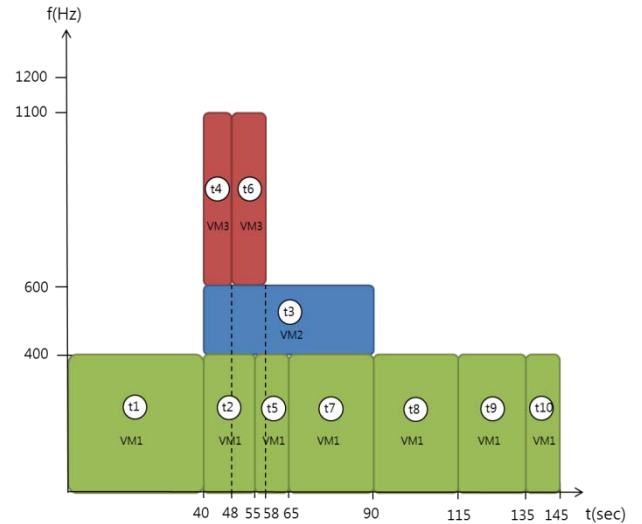
그림 3 은 작업 J 에 대한 가상 머신의 할당 방법을 그래프로 나타낸 것이다. 그림 3(a)는 태스크 당 하나의 가상 머신을 할당하고, 물리 머신의 성능을 최대치인 1200Hz 를 사용하였다. 그 결과, 수행 시간은 단축되었으나 전력 소모량이 68.3 으로 매우 크다. 반면에 static PCH 기법을 사용한 그림 3(b)의 경우, 수행 시간은 그림 3(a)에 비해 길어졌으나, 전력 소모량이 20.62 로 훨씬 감소되었다. 즉, static PCH 기법을 적용하였을 때가 물리 머신 성능을 최대로 사용하여 작업을 수행했을 때보다 데드라인을 만족하면서 더욱 에너지 효율적임을 알 수 있다.

그림 3(c)는 본 논문에서 제안한 기법을 적용한 것으로, static PCH 기법을 확장하여 가상 머신 합병(Consolidation)을 추가로 수행하여 데드라인 만족과 작업의 서브 태스크들의 수행 순서 및 의존성을 지키는 수준에서 전력 소모량을 절감시켰다. 그림 3(b)에서는 VM<sub>1</sub>, VM<sub>2</sub>, VM<sub>3</sub> 총 가상 머신을 3 개를 사용하여 작업을 수행하였지만, 그림 3(c)에서는 VM<sub>2</sub> 와 VM<sub>3</sub>를 합병하여 사용하는 가상 머신의 수를 줄였다. 두 개의 가상 머신을 합병하는 과정에서 t<sub>4</sub>와 t<sub>6</sub> 두 태스크의 처리가 t<sub>10</sub>의 처리가 시작되기 전에 완료되어, 서브 태스크들 간 의존성과 수행 순서를 위반하지 않으면서 합병이 이루어지는 것을 확인할 수 있다. 이렇게 합병된 가상 머신들을 수행할 때의 전력 소모량을 계산하면 18.75 이며, 이는 (a), (b), (c) 세 가지 경우 중에서 가장 적은 전력 소모량을 기록한다. 이를 통하여 사용자의 요구 조건을 만족하면서 사용자의 비용을 줄일 뿐만 아니라, 전력 소모

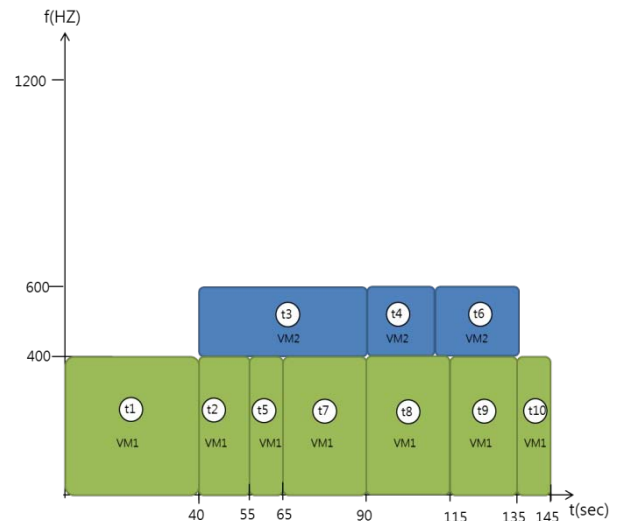
량을 감소시켜 클라우드 플랫폼 관리자의 비용 축소 또한 기대할 수 있다.



(a) 물리 머신의 성능을 최대로 사용하는 경우



(b) Static PCH[8] 기법을 적용하는 경우



(c) 본 논문의 기법을 적용하는 경우

그림 3. 가상 머신의 할당 방법

### 3.5 에너지 효율적인 물리 머신

본 절에서는 물리 머신의 성능 지표인 주파수와 작업 수행 시간 및 에너지, 3 가지 변수간의 관계를 살펴보고, 3 가지 변수를 이용하여 그래프를 제시하여 작업 수행에 적합한 최적의 물리 머신의 성능 및 조건을 예상한다.

그림 3의 그래프들을 분석하였을 때, 물리 머신의 전체 주파수가 낮아질수록 작업 수행 완료시간이 훨씬 더 길어지므로, 작업들의 수행시간  $t$ 와 주파수  $f$ 는 반비례 관계에 있다는 것을 알 수 있다. 즉, 물리 머신의 성능이 좋아질수록, 작업의 수행 완료 시간이 매우 짧아지는 것을 의미한다. 그리고 [1]연구에 따르면, 에너지  $E$ 는 주파수  $f^2$ 에 비례한다. 이 3 개의 변수를 하나의 그래프로 나타내면 아래의 그림 4와 같다.

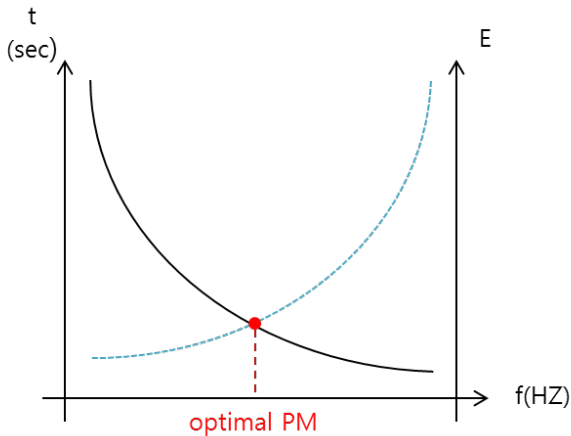


그림 4. 에너지 효율적인 물리 머신 예상 그래프

그림 3(a)의 경우, 물리 머신의 성능을 최대로 사용하였을 때 수행 시간이 매우 짧지만 전력 소모량이 매우 크다. 그에 반해, 그림 3(c)의 경우, 물리 머신의 성능의 절반만 사용할 때 수행 시간이 매우 늘어난다. 이를 통해 사용자가 요구한 데드라인 만족을 위해 적절한 작업 수행 시간을 보장하고, 동시에 클라우드 관리자의 비용을 줄일 수 있는 에너지 효율적인 저전력 물리 머신을 찾는 것이 중요하다는 것을 알 수 있다. 그림 4에서 두 그래프의 교차하는 지점에 해당하는 주파수 수치의 물리 머신을 작업 수행에 사용할 경우, 적절한 수행 시간을 만족하는 저전력 스케줄링이 가능함을 예상할 수 있다.

## 4. 결론 및 향후 과제

본 논문에서는 클라우드 환경에서 워크플로우 어플리케이션을 위한 저전력 가상 자원 할당화 기법을 연구한다. Bag of tasks로 한정된 어플리케이션을 대상으로 연구한 [7]논문을 확장하여 상호의존성을 가지는 다양한 특성의 서브 태스크들로 이루어진 워크플로우를 대상으로 전력을 고려한 가상 자

원 할당 기법을 제안한다. 이를 통해 사용자가 요구한 데드라인을 위반하지 않으면서 사용자의 자원 사용 비용을 최소화하고, 그와 동시에 물리 머신의 전력 소모량을 고려하여 클라우드 플랫폼 관리자의 비용을 최소화한다.

향후에는 본 논문에서 제안한 기법을 다양하고 더 복잡한 워크플로우 어플리케이션을 대상으로 시뮬레이션 및 실험을 진행한다. 시뮬레이션 결과를 통하여 정확한 수치를 얻고 이를 이용하여 그래프 등 통계 자료를 도출해내어 본 논문에서 제안한 가상 자원 할당 기법의 타당성 및 효율성을 증명한다.

## 5. 참고 문헌

- [1] 유종훈, 허승주, 홍성수, “DVFS에 기반한 안드로이드 스마트폰의 CPU 소모 전력 절감 기법의 한계”, 정보과학회지, 제 30 권 제 7 호, pp. 9-16, 2012년 7월.
- [2] G. Laszewski, L. Wang, A. J. Younge and X. He, “Power-Aware Scheduling of Virtual Machines in DVFS-enabled Clusters,” *Cluster Computing and Workshops*, pp. 1-10, Aug. 2009.
- [3] M. Mao, J. Li and M. Humphrey, “Cloud Auto-scaling with Deadline and Budget Constraints,” *Grid Computing, 2010 11<sup>th</sup> IEEE/ACM International Conference on*, pp. 41-48, Oct. 2010.
- [4] M. Mao and M. Humphrey, “Auto-Scaling to Minimize Cost and Meet Application Deadlines in Cloud Workflows,” in *Proc. 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, 2011.
- [5] K. Kim, A. Beloglazov and R. Buyya, “Power-Aware Provisioning of Virtual Machines for Real-Time Cloud Services,” *Concurrency and Computation: Practice and Experience*, vol. 23, no. 13, pp. 1491-1505, 2011.
- [6] N. A. Mehdi, A. Mamat and A. Amer, “Minimum Completion Time for Power-Aware Scheduling in Cloud Computing,” *Developments in E-systems Engineering (DeSE)*, pp.484-489, Dec. 2011.
- [7] 강혜정, 박숙영, 김윤희, “어플리케이션 데드라인 조건을 만족하는 저전력 자원 할당 자동화”, 2012 통신망 운용관리 학술대회(KNOM), 2012년 5월.
- [8] Amazon EC2, <http://aws.amazon.com/ec2/>
- [9] L. F. Bittencourt and E. R. Maderia, “A Performance-oriented Adaptive Scheduler for Dependent Tasks on Grids,” *Concurrency and Computation: Practice and Experience*, vol. 20, pp. 1029-1049, 2008.