

클라우드 컴퓨팅에서 Analytic hierarchy process를 활용한 작업 스케줄링 기법

김정원*

A Job Scheduling Scheme based on Analytic Hierarchy Process in Cloud Computing

Jeong-Won Kim*

요약

클라우드 컴퓨팅을 구성하는 자원은 이질적인 자원으로 구성되므로 다양한 요구사항의 작업을 이질적인 자원에 할당하는 것은 중요한 문제이다. 이는 다수의 판단 기준하에 다수의 선택 문제로 정의할 수 있는데 본 연구에서는 AHP (Analytic hierarchy process) 기법을 통해 이 문제를 해결하고자 한다. 본 연구에서는 작업의 특성을 고려한 우선순위기반 스케줄링 기법을 제안하는데 1단계에서는 작업의 중요도에 의해 우선순위를 부여하고 2단계에서는 응답시간, 시스템 이용률, 그리고 비용을 판단기준으로 하여 AHP알고리즘에 의해 최적의 자원에 작업을 할당한다. 제안하는 알고리즘의 분석과 실험을 통해 제안 기법의 효율성을 제시하고자 한다.

▶ Keywords : 작업 스케줄링, 클라우드 컴퓨팅, AHP, 우선순위

Abstract

As the resources of cloud computing are essentially heterogeneous and jobs have various characteristics, resource allocation to jobs is one of important problems. We define this issue as a multi-criteria decision-making problem. This paper proposes a priority-based job scheduling algorithm based on analytic hierarchy process (AHP). On the first step, jobs are classified based on their preferences. On the second step, response time, system utilization, and load becomes decision criteria based on the AHP algorithm. Jobs are allocated to adequate resources through their priorities that are calculated by the AHP algorithm. Through analysis and experiment of the proposed algorithm, we are to confirm that the scheme can schedule jobs as well as utilize its

• 제1저자 : 김정원
• 투고일 : 2013. 5. 7, 심사일 : 2013. 7. 9, 게재확정일 : 2013. 8. 6
* 신라대학교 컴퓨터공학과(Dept. of Computer Engineering, Silla University)

resource efficiently.

▶ Keywords : job scheduling, cloud computing, AHP, priority.

I. 서 론

분산 컴퓨팅 환경에서 다수의 작업을 스케줄링하는 알고리즘들이 제안되었는데 이들 알고리즘의 목적은 고성능과 시스템 이용률을 최대화 하는 것이다[1,2]. 클라우드 컴퓨팅도 일종의 분산 컴퓨팅 환경으로 작업 스케줄링이 해결해야 할 중요한 문제들 중의 하나이다 [3,4]. 이 작업 스케줄링 문제는 그리드 컴퓨팅의 주된 연구 주제로 일반적으로 단일 워크플로우에서 최적의 성능을 얻기 위해 정적 또는 동적 알고리즘들이 제안되었는데 주로 스케줄링 완료시간 보장, 대기시간 감소 등의 기법들이다. 이 기법들은 주로 최선 방식(best-effort)으로서 일정 수준의 자원의 신속한 할당과 어느 정도의 자원이 워크플로우에 전용될 수 있는가가 주된 관심이지만 클라우드에서는 다른 관점의 접근이 필요하다 [5, 6, 7].

클라우드 컴퓨팅은 광범위한 확장성을 가지고 있고 사용자가 요구하는 서비스의 수준이 다양하며 가상화를 통해 서비스 환경이 동적으로 설정되므로 그리드 컴퓨팅에서의 스케줄링과는 다른 관점이 요구된다. 예를 들면 데이터 처리 위주의 작업의 경우는 스토리지와 데이터 전송 비용이 시간에 따라 기하급수적으로 증가되는 것을 고려해야한다.

본 연구에서는 작업의 다양한 특성과 자원의 이질성을 동시에 고려하는 스케줄링 알고리즘을 제안한다. 서비스 제공자는 시스템의 효율성이 최우선의 과제이고 사용자는 만족스러운 서비스 수준을 요구할 것이다. 이 문제는 일종의 다중 기준의 다중 선택 문제(MCDM: multi-criteria decision-making)로 정의된다. 이 MCDM을 본 연구에서는 AHP 기법을 도입하여 해결하고자 하며 제안하는 알고리즘은 두 단계로 구분하여 실행되는데 먼저 작업의 중요도에 의해 우선순위를 분류하는 단계와 분류된 작업을 자원에 할당하는 단계이다. 편의상 자원은 가상머신(VM: virtual machine)으로 정의한다.

알고리즘의 첫 번째 단계에서는 심각한 사업상의 손실과 휴유증을 초래할 수 있는 경우는 자원을 전용(resource provisioning)하고 소규모의 사용자 불편과 미미한 기능적 장애를 초래할 수 있는 경우는 최선(best-effort) 방식으로 분류한다.

두 번째 단계에서는 시스템의 효율을 최대화할 수 있도록 작업을 가상머신에 할당하는 단계이다. 클라우드 환경에서는 수많은 노드가 존재하고 노드의 자원은 불가피하게 신뢰성이 낮고 이질적이다. 또한 작업은 응답성, 비용, 시스템의 부하 측면에서 다양한 QoS를 가진다. 이 문제의 효율적인 해결을 위해 다양한 결정 기준에 대한 대안의 우선순위를 결정할 수 있는 AHP 알고리즘을 적용하여 작업을 실행할 가상머신을 결정한다.

본 논문에서는 클라우드 컴퓨팅에서의 다양한 관련 연구를 살펴보고 제안하는 알고리즘을 구체적으로 소개한다. 또한 시뮬레이션을 통해 본 연구에서 제안하는 기법을 검증하고 타 기법 대비 효율성을 분석한다.

II. 관련 연구

클라우드 컴퓨팅에서 작업 스케줄링은 배치 모드의 휴리스틱기법과 온라인 모드의 휴리스틱기법으로 분류될 수 있다. 배치 모드에 포함되는 것은 FCFS, RR, Min-Min and Max-Min 알고리즘이 있고 온라인 모드는 작업이 도착하자마자 스케줄되는데 클라우드 환경이 이질적이고 각 프로세서의 성능이 다양하므로 온라인 모드가 적합하며 MFTF(Most fit task scheduling) 알고리즘이 대표적이다. 다음은 본 연구와 관련된 선행 연구를 분석한 것이다.

Jia Yu 등은 비용기반 스케줄링 알고리즘을 제안하였는데 테드라인을 만족하면서도 실행 비용의 최소화를 고려하고자 하였다 [8]. Jia Yu 와 Rajkumar Buyya는 결과의 비용을 만족시키면서 실행시간의 최소화를 얻고자하는 스케줄링 기법을 제안하였는데 스케줄링 최적화 문제를 해결하기 위하여 유전 알고리즘을 적용하였고 그리드환경에서 실험 결과를 제시하였다 [9].

P. Padala는 작업이 요구하는 QoS를 만족시키고 자원 이용도를 높이기 위하여 응용들 사이에 자원 공유를 조정하고자하는 알고리즘을 제안하였다 [10]. Zhifeng Yu와 Weisong Shi는 다중 워크플로우를 위해 계획기반 알고리즘을 제안하였는데 워크플로우 간의 순위를 미리 결정하고 이를 기반으로 스케줄 대상을 선정하는 방식이다 [11]. Meng Xu

는 요구사항이 중첩되는 다중 QoS와 다중 워크플로를 지원하는 스케줄링 알고리즘을 제안하였는데 논문에서는 이 기법을 통해 스케줄링 접근 비율을 향상시킬 수 있음을 보이고 있다 [12]. J. Kosinska는 클라우드 자원내에서 실행되는 응용프로그램의 신뢰성과 확장성을 향상시키고자 클라우드 위상에 대한 다양한 형태를 제안하였다 [13].

본연구와 유사한 연구는 속성레벨이 자원이고 대안이 작업인 모델에서 작업의 우선순위를 결정하기 위해 AHP기법을 사용하였는데 [14] 본 연구는 AHP 모델의 기본 취지에 적합하게 속성레벨이 작업의 특성인 응답성, 비용, 시스템 부하로 두고 대안을 가상머신으로 두는 모델이다. 작업이 시스템에 들어오면 QoS에 의해 1차적으로 우선순위가 나누어지고, AHP에 의해 작업의 특성에 따라 최적의 가상머신에 작업을 배치한다.

III. AHP에 기반한 작업 스케줄링

AHP 우선순위에 기반한 작업 스케줄링은 2단계로 진행된다. 첫 번째 단계에서는 작업의 중요도에 따라 우선순위가 높은 작업을 먼저 스케줄링하는 단계이고 두 번째 단계는 작업을 어느 가상머신에 할당할 것인가를 결정하는 단계로 AHP 모델을 적용한다.

3.1 작업의 QoS에 기반한 우선순위 분류

작업은 크게 Resource Provisioning과 Best-effort로 분류하는데 Resource Provisioning은 요구자원을 미리 확보하는 경우이고 Best-effort는 작업의 QoS에 최대한 맞추어 스케줄링하는 경우이다. Resource Provisioning의 경우 사용자가 비용을 많이 지불하거나 중요도가 높은 경우에 해당하며 만약 스케줄링이 실패할 경우 심각한 손실을 초래하는 경우가 레벨 1, 상당한 휴유증을 유발하는 경우는 레벨 2로 분류한다. Best-effort는 저렴한 비용을 지불한 사용자나 중요도가 낮은 경우에 해당하며 소규모 사용자의 불편을 초래하는 경우는 레벨 3, 미미한 기능적 장애를 초래한 경우는 레벨 4로 분류한다. 레벨 1부터 4의 순서로 사용료 지급이 높아질 것을 예상되므로 레벨 1부터 스케줄되고 동일 레벨의 경우는 FCFS방식으로 처리한다. 또한 정해진 순서는 변경 불가능하며 비선점 방식으로 스케줄링한다.

3.2 AHP에 기반한 자원(VM) 할당

작업이 스케줄러에 입력되면 어떤 가상머신에 할당할 것인가

를 결정하게 되는데 각 가상 머신의 자원 이용도와 작업의 특성은 다양하기 때문에 적합한 VM의 선택은 중요한 문제이다.

클라우드 스케줄러는 선택 가능한 VM, 즉 대안(alternatives)들 중 하나에 작업을 할당한다. 이때 VM의 선택시 기준(attributes)이 존재하는데 본 연구에서는 기준에 가중치를 부여하고 AHP 알고리즘이 대안들 중에서 최선 방식으로 VM을 선택한다. 이 AHP의 장점은 다양한 대안들을 고려하고 기준에 가장 적합한 대안을 선택하는 것이다.

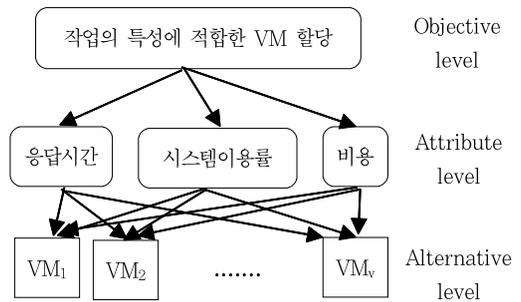


그림 1. VM할당을 위한 AHP 모델
Fig. 1. AHP Model for VM Allocation

그림 1의 모델에서 objective level은 작업을 어느 가상머신에 배치할 것인가를 결정하는 것이다. 이를 해결하기 위한 판단기준, 즉 Attributes level은 응답시간, 시스템이용률, 그리고 비용이다. 그리고 대안, 즉 Alternative level은 클라우드를 구성하는 가상머신이 된다. 각 판단기준들은 중요도를 가지는데 본 연구에서는 표 1에서 보듯이 9단계로 정하고 9가 가장 높은 우선순위가 된다. 2, 4, 6, 8은 중간정도의 우선순위를 의미하며 역수는 중요도의 반대 개념이다. 이 우선순위값은 작업의 응답시간, 시스템이용율, 비용의 선호도 결정과 각 가상머신의 현재 응답시간, 시스템이용율, 비용에 대한 자원의 결정에 적용된다.

표 1. 중요도의 단계
Table 1. Level of importance

| 중요도 | 정의 |
|------------|------------|
| 1 | 통상적임 |
| 3 | 약간 중요함 |
| 5 | 중요함 |
| 7 | 매우 중요함 |
| 9 | 극히 중요함 |
| 2, 4, 6, 8 | 위 중요도의 중간값 |
| 역수 | 중요도의 반대 값 |

그림 2는 AHP 모델에 의해 VM을 선택하는 단계를 나타내고 있는데 먼저 결정 기준을 정의하고 모든 판단기준에 대한 쌍비교 행렬을 계산한다. 이 행렬에서 가중치를 구하여 일관성을 조사한다. 이 값(Consistent Index : CI)이 0.1보다 작으면 모든 가상 머신에 대한 득점을 계산하여 최고 득점의 VM에 작업을 할당한다. 다음은 구체적인 과정을 설명한다.

본 연구에서는 작업의 집합은 $\delta = \{J_1, J_2, \dots, J_m\}$ 이고, 판단 기준은 $\zeta = \{C_1, C_2, \dots, C_n\}$, 가상 머신의 집합은 $\Omega = \{VM_1, VM_2, \dots, VM_v\}$ 이라고 가정한다.

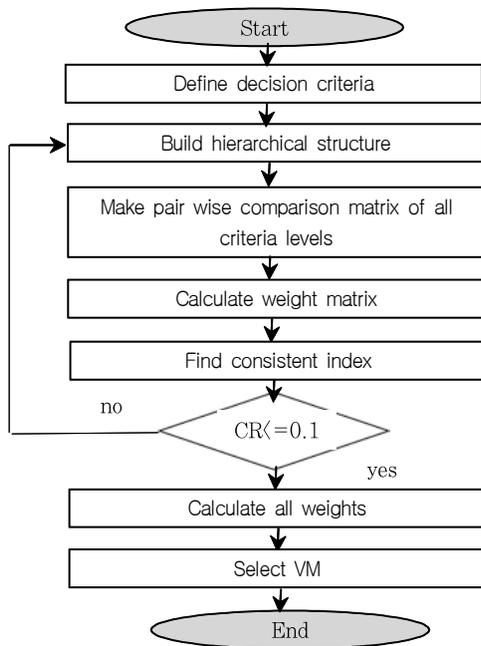


그림 2. AHP 모델에 의한 VM선택 단계
Fig. 2. VM selection based on AHP model

3.2.1 objective level과 attribute level에서 가중치 벡터와 CI 계산

먼저 Objective level과 Attribute Level 간의 쌍비교 행렬은 작업의 응답성, 시스템이용률, 비용 등의 특성에 의해 결정되는데 이것을 행렬 M으로 가정한다. 행렬 M은 판단기준의 $n \times n$ 행렬이고 (i,j) 성분의 선호도가 5라면 (j,i) 성분의 선호도는 1/5이 될 것이다. 행렬 M의 임의 비교를 통해 $\omega^c = \{\omega_1, \omega_2, \dots, \omega_n\}$ 과 같은 가중치 벡터를 얻을 수 있다. M의 쌍비교 행렬은 식 (1)과 같이 표현된다.

$$M_{i,j} = \begin{cases} M_{i,j} = \frac{w_i}{w_j}, & i \neq j, (i, j \leq n) \\ 1, & i = j \end{cases} \quad \text{수식 (1)}$$

다음은 행렬 M의 가중치 벡터 $\omega^c = \{\omega_1, \omega_2, \dots, \omega_n\}$ 를 구하는 것인데 이 과정은 식(2)를 계산하여 얻게 된다.

$$M \times \omega^c = \lambda_{\max} \times \omega^c \quad \text{수식 (2)}$$

λ_{\max} 는 행렬 M의 고유값이고 고유벡터인 ω^c 로 표현된다. AHP 알고리즘의 핵심적인 부분은 가중치 벡터를 계산하는 것이다. 식(2)의 구체적인 표현은 식(3)이 된다.

$$\begin{bmatrix} w_1 & \dots & w_1 \\ w_1 & \dots & w_n \\ \vdots & \ddots & \vdots \\ w_n & \dots & w_n \\ w_1 & \dots & w_n \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \lambda_{\max} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \quad \text{수식 (3)}$$

이 행렬 M이 일관성이 있다면 λ_{\max} 값은 n에 수렴하게 된다. 이 일관성 비율을 식(4)로 계산하였다. 식(4)에서 RI는 random index이고 이 값은 비교행렬의 우선순위에 의해 임의적으로 계산되어진다 [15]. 표 2는 계산되어진 RI 값을 나타낸다. CR값이 0.1이하이면 작으면 행렬 M은 일관성이 있음을 나타낸다 [15].

$$CR = \frac{CI}{RI}, \text{ where } CI = \frac{\lambda_{\max} - n}{n - 1} \quad \text{수식 (4)}$$

표 2. RI(Random index)
Table 2. RI(Random index)

| n | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|------|-----|------|------|------|------|------|------|
| RI | 0.58 | 0.9 | 1.12 | 1.24 | 1.32 | 1.41 | 1.45 | 1.49 |

우선순위 벡터를 계산하는 다양한 방법들이 존재한다. 본 논문에서는 식(2)를 계산하기 위해서 반복적 기법을 사용하는데 일관성 문제를 걱정할 필요 없이 우선순위 벡터를 계산할 수 있기 때문이다.

3.2.2 attribute level과 alternative level 사이의 가중치행렬과 일관성의 계산

알고리즘의 다음 단계는 각 판단기준에 대한 가상머신의 중요도와 일관성을 계산하는 것이다. 식(1)과 같이 쌍비교 행

렬을 계산하고 각 VM에 대한 판단기준의 중요도 벡터와 일관성 값을 계산한다. 각 VM에 대한 중요도 벡터를 $\omega^{vm} = \{\omega_1, \omega_2, \dots, \omega_v\}$ 라고 가정하면 ω^{vm} 은 행의 수가 가상머신의 개수 v 이고 열의 수는 판단기준 n 이 된다. 따라서 3.2.1과 3.2.2에서 구한 중요도 행렬을 각각 곱하면 각 가상머신의 특징을 얻을 수 있다.

$$rank = \omega^{vm} \times \omega^c, (\omega^{vm} = v \text{ by } n, \omega^c = n \text{ by } 1) \quad \text{수식 (5)}$$

식 (5)의 rank는 v 개의 요소를 가지므로 이중 최대값을 가지는 가상머신이 작업을 수행하게 된다. 알고리즘의 복잡도를 계산하면 식 (6)과 같다.

$$O(n^{2.38} + n \times v^{2.38} + d \times n^{2.38}) \quad \text{수식 (6)}$$

식(6)에서 n 은 판단기준의 수이고 v 는 가상머신의 수이며 d 는 일관성 검사에서 탈락된 회수를 의미한다. n by n 행렬의 곱셈의 복잡도는 $O(n^3)$ 이지만 지금까지 알려진 가장 빠른 알고리즘은 $O(n^{2.38})$ 이다 [16]. 따라서 제안하는 알고리즘의 최악시간은 $O(n^3)$ 이고 최적의 시간복잡도는 $n^{2.38}$ 또는 $v^{2.38}$ 이 될 것이다. 그러나 보통 판단기준 n 보다는 가상머신의 수 v 가 높으므로 식 (7)과 같이 $n \times v^{2.38}$ 에 의해 시간 복잡도가 결정된다.

$$O([\max(v, n)]^{2.38} \times q), \text{ if } v > n, q = n, \text{ else } q = d \quad \text{수식 (7)}$$

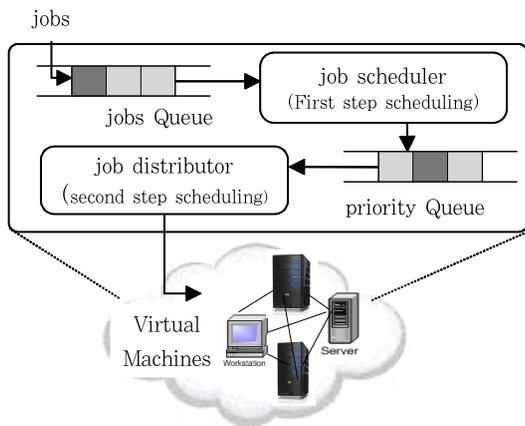


그림 3. 시뮬레이션 환경
Fig. 3. Simulation environment

IV. 실험

제안기법의 성능을 평가하기 위하여 그림 3과 같은 실험 환경을 구축하였다. 실험의 요소는 작업, 스케줄링 기법, 그리고 VM으로 구성된다. 작업의 QoS 파라미터는 도착 시간 분포, 우선순위, 자원 요구량으로 구성되고, 스케줄링의 파라미터는 RR, First fit, AHP이다. RR은 작업을 가상머신에 순서대로 할당하는 것이다. First fit은 가상머신을 시스템 이용률 순서대로 정렬하여 가장 여유가 있는 가상머신에 작업을 할당하는 방법이다.

작업의 도착 시간 분포는 포아송 분포이고 중요도는 1에서 9까지 정규분포를 따른다. 각 작업의 판단기준 값은 1에서 5까지 랜덤 변수를 생성하여 할당하였다. 스케줄링 파라미터는 기존 연구와 비교를 통해 제안 기법의 성능을 평가하는 것이 원칙이지만 기존 기법을 구현하는 것이 현실적으로 불가능하므로 RR, First fit 기법과 비교를 하였다. VM은 이질적 성능을 고려하여 5개의 VM으로 구성되고 각 VM은 최대 5배의 성능 차를 가지는 것으로 가정하였다. 각 VM의 응답시간은 1에서 5까지의 정수이고, 1이 가장 성능이 좋은 시스템이다. 시스템 이용률은 각 VM이 작업 큐에서 서비스하고 있는 작업의 개수에 의해 계산되어진다. 비용은 시스템 구성에 따라 차이가 많으므로 실험의 간단성을 위해 최대 세 배까지로 정하였다. 본 실험은 매트랩과 시뮬링크를 사용하여 수행되었다.

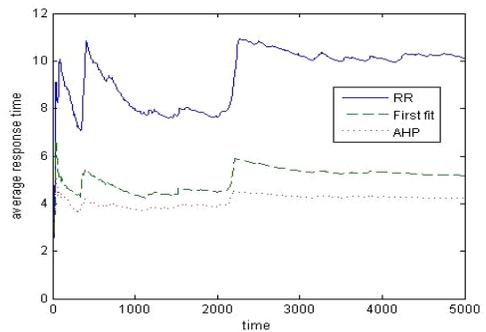


그림 4. 평균 응답 시간
Fig. 4. Average response time

그림 4는 단위 시간에 큐잉된 작업들의 평균 응답시간을 시간의 변화에 따라 측정한 것이다. 그림 4의 실험은 사용자의 관점에서 본 논문이 제안하는 기법의 효율성을 확인하는 것이다. 작업은 시뮬레이션의 파라미터에 의해 지속적으로 생성이 되며 응답시간은 매초 마다 큐잉된 작업들이 가상머신을

떠나는 경과시간을 측정할 것이다. First fit, AHP는 응답시간이 조금씩 증가하지만 큰 변화는 보이지 않으나 RR 기법은 두 기법보다 높은 응답 시간을 보이고 있다. RR기법이 VM의 부하에 관계없이 작업을 순서대로 할당하기 때문이다. 논문에서 제안하고 있는 AHP기법이 First fit 보다 응답시간이 빠른 것은 First fit이 처음 발견되는 VM에게만 지속적으로 작업을 할당하기 때문에 큐의 대기시간이 증가하기 때문으로 분석된다.

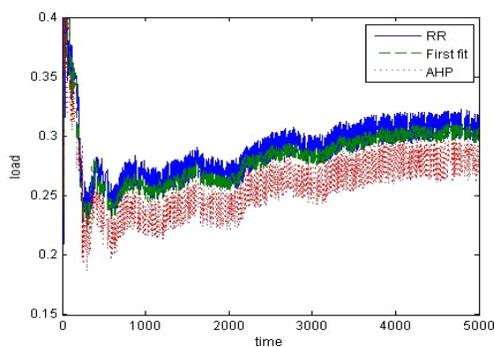


그림 5. VM의 부하(자원 이용률)
Fig. 5. Load of VM(resource utilization)

그림 5는 단위시간에서 모든 VM들의 자원 이용률, 즉 부하도를 측정할 것이다. 그림에서 load값 0은 모든 VM이 성능 대비 동일한 부하도를 유지하고 있는 것을 의미하고 1은 부하의 불균형이 최대임을 의미한다. 이 실험은 서비스 제공자 관점에서 제안하는 기법의 효율성을 확인하는 것이다. 그림에서 load값이 커질수록 특정 VM에 부하가 집중되고 있음을 의미하는데 실험에서 VM의 성능 차이는 최대 5배가 되도록 설정하였다. 세 기법 모두 실험 초기에는 부하가 특정 VM에 집중되므로 부하 불균형을 보이지만 실험이 진행될수록 완만해진다. 그림에서 RR기법은 VM의 성능 차이에 관계없이 작업을 할당하므로 성능이 낮은 VM은 부하가 높게 발생되므로 전체적인 부하의 편차가 크게 나타난다. 반면 First fit, AHP는 스케줄 시점에서 부하가 낮은 VM에게 작업을 할당하므로 부하가 균등하게 유지된다. 또한 AHP는 최적의 VM을 선정하므로 First fit에 비해 낮은 부하를 나타낸 것으로 분석된다.

V. 결론

이질적인 자원으로 구성된 클라우드 컴퓨팅 환경에서 다양

한 QoS를 가진 작업을 처리해야 하는 작업 스케줄링은 중요한 문제이다. 본 연구에서는 이 문제를 다중 기준 다중 선택으로 정의하고 AHP 모델에 기반하여 2단계로 작업을 스케줄링하는 알고리즘을 제시하였다. 첫 번째 단계에서는 작업의 특성을 반영하기 위해 자원을 전용해야 하는 작업과 최선으로 자원을 할당하는 작업으로 구분하여 우선순위를 정한다. 2단계에서는 응답성, 비용, 시스템 부하를 판단 기준으로 하여 AHP 모델을 적용하여 최적의 VM에 작업을 할당한다. 실험 결과 작업의 요구조건을 만족하면서도 시스템의 이용률을 최대화할 수 있음을 확인하였다. 본 연구는 사용자관점과 서비스제공자 관점 등 다양한 판단기준을 모두 고려하여 작업을 자원에 할당하는 기법을 제안하였고 실험을 통해 긍정적인 결과를 확인할 수 있었다. 향후 과제는 알고리즘의 복잡도를 최대한 줄이고 실제 시스템에 적용하는 것이다.

참고문헌

- [1] Dongwon Jeong, "A Standard Reference Model for Semantic Interoperability in Cloud Computing," Journal of The Korea Society of Computer and Information, pp.71-80, 2012.
- [2] WonJoo Lee, YongIk Yoon, "A Design and Implementation of N-Screen Emulator Based on Cloud," Journal of The Korea Society of Computer and Information, pp.11-18, 2013.
- [3] Monir Abdullah, Mohamed Othman, "Optimal Workload Allocation Model for Scheduling Divisible Data Grid Applications," Future Generation Computer Systems 26, pp.971-978, 2010.
- [4] Amin Shokripour, Mohamed Othman, "New Method for Scheduling Heterogeneous Multi-Installment Systems," Future, Generation Computer Systems 28, pp.1205-1216, 2012.
- [5] Ewa Deelman, "Grids and Clouds: Making Workflow applications Work in Heterogeneous Distributed Environments," The International Journal of High Performance Computing Applications, Vol24, No.3. Fall 2010.
- [6] Jablonski, S. and C. Bussler, "Workflow Management Systems: Modeling, Architecture and Implementation," Thomsom Press, 1996.

- [7] Jia Yu, Rajkumar Buyya and Chen Khong Tham, "Cost-based Scheduling of Scientific Workflow Applications on Utility Grids," In 1st IEEE International Conference on e-Science and Grid Computing, Melbourne, Australia, Dec. 5-8, 2005.
- [8] Jia Yu, Rajkumar Buyya and Chen Khong Tham, "Cost-based Scheduling of Scientific Workflow Applications on Utility Grids," In 1st IEEE International Conference on e-Science and Grid Computing, Melbourne, Australia, Dec. pp.5-8, 2005.
- [9] Jia Yu and Rajkumar Buyya, "A Budget Constrained Scheduling of Workflow Applications on Utility Grids using Genetic Algorithms," Proceedings of the 15th IEEE International symposium on High Performance Distributed Computing, June 19-23, 2006.
- [10] P. Padala, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant and K. Salem, "Adaptive control of virtualized resources in utility computing environments," SIGOPS Oper. Syst. Rev., vol.41, pp.289-302, 2007.
- [11] Zhifeng Yu and Weisong Shi, "A Planner- Guided Scheduling Strategy for Multiple Workflow Applications," International Conference on Parallel Processing - Workshops, pp.1-8, 2008.
- [12] Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi, "A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing," 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications, pp.629-634, 2009.
- [13] J. Kosinska, J. Kosinski, K. Zielinski, "The Concept of Application Clustering in Cloud Computing Environments", 2010.
- [14] Shamsollah Ghanbaria, Mohamed Othman, A Priority based Job Scheduling Algorithm in Cloud Computing, International Conference on Advances Science and Contemporary Engineering, pp.778-785, 2012.
- [15] T.L.Saaty, Decision Making for Leaders: The Analytical Hierarchy Process for Decisions in a Complex World, Pittsburgh: RWS Publications, 2000.
- [16] Coppersmith, Don, Winograd, Shmuel, "Matrix multiplication via arithmetic progressions," Journal of Symbolic Computation, 1990.

저 자 소 개



김 정 원

1995년 : 부산대학교
전자계산학과(학사)

1997년 : 부산대학교 대학원
전자계산학과(석사)

2000년 : 부산대학교 대학원
전자계산학과(박사)

2000년~2001년 : 기술신용보증기금
기술평가역(차장)

2002년~현재 : 신라대학교
컴퓨터정보공학부 교수

관심분야 : 내장형시스템,
멀티미디어, 운영체제

Email : jwkim@silla.ac.kr