

# 클라우드 컴퓨팅에서 다중 워크플로우 어플리케이션을 위한 비용 기반 랭크 스케줄링 알고리즘

최 경 근<sup>†</sup> · 이 봉 환<sup>††</sup>

## 요 약

클라우드 컴퓨팅은 자원 공유를 위한 새로운 컴퓨팅 패러다임이다. 클라우드 서비스를 위해 사용하는 다양한 어플리케이션들은 워크플로우들로 표현된다. 이러한 워크플로우 어플리케이션은 클라우드의 자원 또는 서비스들에 적절하게 할당되어야 한다. 본 논문에서는 클라우드 컴퓨팅 환경을 고려하여 다중 워크플로우 어플리케이션을 위한 새로운 스케줄링 알고리즘을 제안한다. 제안하는 알고리즘은 비용 기반 랭크 스케줄링 알고리즘으로서 다중 워크플로우 어플리케이션을 고려할 뿐만 아니라, 서비스 평가를 위한 다양한 QoS 메트릭을 고려한다. 실험결과에서 제안한 알고리즘은 다른 알고리즘들에 비해 평균 총 처리시간과 평균 가용성에서 향상된 결과를 보였다.

키워드 : 클라우드, 워크플로우, 태스크, 스케줄링, 비용

## Cost-Based Rank Scheduling Algorithm for Multiple Workflow Applications in Cloud Computing

Gyeong-Geun Choe<sup>†</sup> · Bong-Hwan Lee<sup>††</sup>

### ABSTRACT

Cloud computing is a new computing paradigm for sharing resources. Various applications used for cloud services are represented as workflows. These workflow applications must be appropriately allocated to resources or services in cloud. In this paper, a new scheduling algorithm is proposed for multiple workflow applications considering cloud computing environment. The cost-based rank scheduling algorithm considers not only multiple workflow applications, but various QoS metrics for evaluating services. Simulation results show that the proposed algorithm can improve the mean makespan and the availability significantly over two well-known algorithms.

Keywords : Cloud, Workflow, Task, Scheduling, Cost

### 1. 서 론

최근 인터넷 사용자의 증가와 대용량 어플리케이션의 증가로 인터넷 트래픽이 급증하고, 이를 처리하기 위한 서버 및 IT 자원의 수요도 증가함에 따라 효율적인 IT 자원 활용을 위한 기술들이 필요하게 되었다. 이를 위해 기업 또는 연구기관들이 비용절감을 위한 전략중 하나로 클라우드 컴퓨팅에 관심을 갖기 시작했다[1].

클라우드 컴퓨팅은 하드웨어, 플랫폼, 서비스들과 같은 가상화된 자원을 액세스 할 수 있는 구조로, 가상화 자원들은 동적으로 사용자 요구에 따라 할당 된다[2][3]. 클라우드 서비스 공급자는 필요한 만큼 사용자에게 자원을 빌려주고, 사용자는 사용한 만큼 비용을 지불한다[4]. 즉, 클라우드 컴퓨팅은 자원을 서로 공유하고 유휴자원을 효율적으로 활용하여 전체 자원 및 비용을 절감시킬 수 있는 장점을 갖고 있다. 이와 같이 클라우드 컴퓨팅은 수많은 어플리케이션을 위한 적합한 컴퓨팅 구조로 각광받고 있으며, 기업과 연구기관들에서 클라우드 컴퓨팅에 관한 많은 연구가 진행 중이다.

클라우드 컴퓨팅은 서버, 스토리지, 네트워크, 어플리케이션 등과 같은 컴퓨팅 자원과 비즈니스 자원을 포함하는 구조에서 적절한 자원을 선택하고 동적으로 구성한 후 워크로

※ 본 연구는 2010년도 교육과학기술부와 한국연구재단의 지역혁신인력양성사업 및 일반연구자지원사업으로 수행된 연구 연구결과임(No. 2010-0016574).

† 준 회 원 : 대전대학교 정보통신공학과 박사과정

†† 종신회원 : 대전대학교 정보통신공학과 교수(교신저자)

논문접수 : 2010년 12월 2일  
수정일 : 1차 2011년 1월 4일  
심사완료 : 2011년 1월 10일

드가 이 위에서 수행될 수 있도록 해야 한다[1][2]. 즉, 사용자들이 사용하는 어플리케이션들 데이터의 변형인 워크플로우들이 자원 또는 서비스에 적절하게 할당되어야한다.

워크플로우(Workflow)는 작업 절차를 통한 정보 또는 업무의 이동을 의미하며, 작업 흐름이라고도 부른다[5][6][15]. 이러한 워크플로우가 클라우드 컴퓨팅 환경에서 효율적으로 수행되기 위해서는 부분 작업들 또는 태스크들이 클라우드 자원 또는 서비스에 적절하게 할당되어야한다[6]. 즉, 워크플로우에 존재하는 각 태스크들의 효율적인 서비스 할당은 클라우드 컴퓨팅의 성능에 영향을 줄수 있다.

클라우드 컴퓨팅 환경은 여러 특징을 갖고 있다[3]. 첫째, 광범위한 확장성을 갖는다. 두 번째, 클라우드의 사용자들은 서로 다른 서비스 수준을 갖는다. 세 번째, 클라우드 서비스는 가상화를 통해 동적으로 설정된다.

이러한 클라우드 컴퓨팅 환경의 특징을 가지고, 워크플로우 스케줄링 전략을 위해 새로운 특징을 고려한다. 첫째, 클라우드는 다중 사용자들에게 동시에 같은 서비스를 제공할 수 있어야 한다. 따라서 워크플로우 스케줄링은 다양한 QoS(Quality of Service) 파라미터를 가진 다중 워크플로우를 스케줄링 할 수 있어야 한다. 두 번째, 사용자들은 그들이 원할 때 자원 또는 서비스에 액세스 할 수 있어야 한다. 즉, 클라우드 서비스는 새로운 워크플로우를 시작할 준비가 되어 있어야 한다.

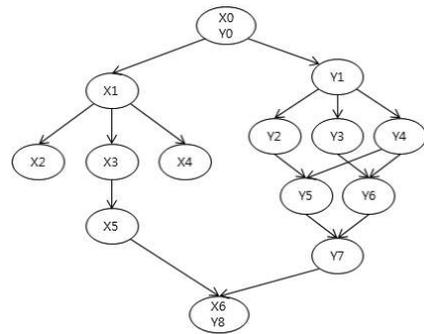
이러한 요구사항을 만족하기 위하여 본 논문에서는 클라우드 컴퓨팅 환경을 고려한 다중 QoS 파라미터를 가지고 다중 워크플로우를 위한 워크플로우 스케줄링 알고리즘을 제안한다.

## 2. 관련 연구

클라우드 컴퓨팅은 다중 사용자들에게 서비스를 제공해야 한다. 즉, 클라우드 서비스는 동시에 많은 사용자들의 어플리케이션인 다중 워크플로우를 위한 서비스를 제공해야 한다. 따라서, 다중 워크플로우 스케줄링을 고려한다.

워크플로우 스케줄링 문제는 그동안 주된 연구 토픽이었으며, 특히, 그리드 컴퓨팅 환경에서 단일 워크플로우의 최적 성능을 얻기 위한 목적으로 다양한 알고리즘들이 제안되었다. 이런 정적 또는 동적 알고리즘들은 많은 워크플로우 관리 시스템에 성공적으로 적용되었다[15].

Hönig은 다중 워크플로우에 대한 메타 스케줄러를 고안했다[7]. 이것은 전체적인 병렬성을 향상시키기 위해 다중 워크플로우를 단일 워크플로우로 분류하는 것을 제안했다. 하지만, 이런 노력들은 다른 시간에 다중 워크플로우에 들어오는 동적 워크로드를 다루는데 제한이 있다. 다중 워크플로우 스케줄링에 대한 위 접근에 대한 대안은 동시에 모든 워크플로우를 스케줄링 하는 것이다. 여기에는 많은 방법들이 있으나 일반적으로 네 가지로 분류할 수 있다[5]:



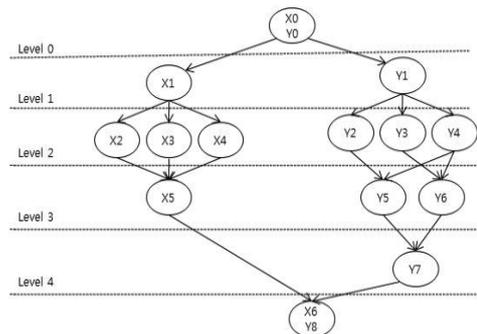
(그림 1) 복합 그래프

- 복합 워크플로우

모든 워크플로우의 상속자들이 새로운 입구 노드와 모든 워크플로우의 피상속자들이 새로운 출구 노드를 만들어 복합 그래프를 생성한다(그림 1).

- 레벨 기반 워크플로우

(그림 1)과 같은 방법으로 복합그래프를 생성한다. 하지만 새로운 그래프가 (그림 2)와 같이 레벨로 그룹화 된다. 스케줄링 태스크들은 레벨에 위치한다.



(그림 2) 레벨 기반 순서화 그래프

- 라운드 로빈 워크플로우

(그림 1)과 같은 방법으로 복합 워크플로우를 생성한다. 하지만, 스케줄링은 라운드로빈(Round-Robin) 방법으로 각 워크플로우의 태스크를 고려한다.

- 랭크 기반 워크플로우

이 접근의 개념은 각 워크플로우의 구조와 출구 노드들이 링크될 때 각 태스크 실행시간을 고려한다. 그래서 짧은 워크플로우의 출구 노드(즉, 스케줄링 되는 총 처리시간(Makespan)이 짧은 워크플로우)는 긴 워크플로우의 태스크로 링크된다.

위에서 언급한 다중 워크플로우 스케줄링에 대한 접근은 각각 그들만의 장점을 갖는다. 하지만 그 어떤 것도 클라우드 컴퓨팅 환경을 고려하지 않는다. 따라서 본 논문에서는 클라우드 컴퓨팅 환경을 고려한 다중 워크플로우 어플리케이션을 위한 스케줄링 알고리즘을 제안한다.

### 3. 문제정의

지금까지 워크플로우 스케줄링 알고리즘은 그리드 컴퓨팅 환경에서 제안되었다. 이 장에서는 두 가지 관점 즉, 휴리스틱 스케줄링 알고리즘과 평가 메트릭으로 분류해 문제점을 정의 한다.

#### 3.1 휴리스틱 스케줄링 알고리즘

이전의 워크플로우 스케줄링 알고리즘에서는 대부분의 경우에 정적 알고리즘이 동적 알고리즘보다 성능이 우수함을 보여준다[6][8][9][10][15]. 하지만, 정적 알고리즘들은 동적이고 광범위한 확장성을 갖는 클라우드 컴퓨팅에 적합하지 않다. 즉, 다중 워크플로우가 발생하는 클라우드 컴퓨팅 환경에 적합하지 않다. 또한 임의의 사용자들은 그들이 원할 때 서비스를 사용할 수 있어야 한다. 즉, 클라우드 서비스는 새로운 워크플로우를 시작할 준비가 되어 있어야 한다.

한편, 동적 알고리즘은 각 태스크가 실행할 준비가 되었을 때 만 스케줄링 결정을 하기 때문에 다중 워크플로우를 지원할 수 있다. 하지만, 대부분의 동적 알고리즘은 다양한 QoS 파라미터를 고려하지 않는다. 즉, 실행 시간 또는 비용 같은 단일 QoS 파라미터를 고려한다.

다중 워크플로우 스케줄링을 위한 이러한 접근들의 대안적 접근은 동시에 모든 워크플로우를 스케줄링 하는 것이다 [5]. 여기에는 많은 방법들이 있다. 특히 2장에서 랭크 기반 워크플로우 개념은 Zhifeng Yu와 Weisong Shi[11]에 의해서 제안되었는데, 제안한 접근은 planner-guided 알고리즘으로서 다중 워크플로우를 고려했다. 하지만, 그러한 접근은 클라우드 서비스 또는 자원이 충분하지 않을 경우, 불필요한 처리시간 또는 더 많은 시스템 가용성을 갖는다.

따라서, 본 논문에서는 제안하는 다중 워크플로우 스케줄링 기법을 비교 평가하기 위해 Zhifeng Yu 와 Weisong Shi[11]가 제안한 planner-guided 전략을 고려한다.

#### 3.2 평가 메트릭

앞서 설명한 워크플로우 스케줄링 알고리즘 관점 이외에, 사용자와 서비스 공급자 측면에서의 클라우드 서비스 평가를 위한 QoS 메트릭을 고려한다.

클라우드 컴퓨팅에서 서비스 로드는 사용자 서비스 요청에 따라 동적으로 변하기 때문에 다중 사용자가 존재하며, 그들이 서비스를 받고 싶을 때마다 동시에 서비스를 요청할 수 있다.

클라우드 컴퓨팅 환경에서 서비스 평가는 사용자와 서비스 공급자에 따라 다르다[12]. 사용자는 일반적으로 QoS 메트릭에서 처리율(Throughput) 보다 응답시간(Response Time)과 총 처리시간(Makespan)에 관심이 있다. 또한, 서비스 공급자는 비용(Cost), 이용률(Utilization) 및 가용성

(Availability) 등에 관심이 있다[12].

따라서, 각 워크플로우를 동시에 발생시켜, 사용자와 공급자 측면에서의 평균 총 처리시간(Mean Makespan)과 서비스 공급자 측면에서의 평균 가용성(Mean Availability)을 고려한다. 가용성은 컴퓨터 시스템 설계에 중요한 메트릭 중에 하나이며, 본 논문에서 제안하는 워크플로우 스케줄링 시스템의 효율성을 측정하는 메트릭이 된다.

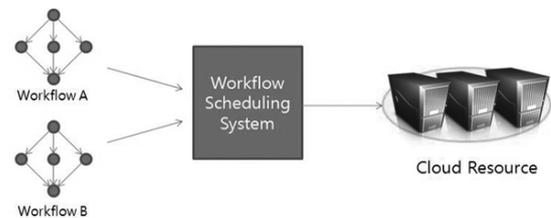
제안하는 알고리즘은 다중 워크플로우를 적용하며, 사용자와 서비스 공급자 두 측면에서 워크플로우 스케줄링 서비스를 평가한다. 따라서, 제안하는 알고리즘은 각 워크플로우의 성능을 고려할 뿐만 아니라, 서비스가 즉각적으로 사용할 준비가 되어 있는 지를 평가하는 QoS 메트릭을 고려한다.

## 4. 시스템 개요 및 워크플로우 어플리케이션

본 논문에서 제안한 스케줄링 알고리즘을 위해 필요한 워크플로우 스케줄링 시스템과 워크플로우 어플리케이션을 정의한다.

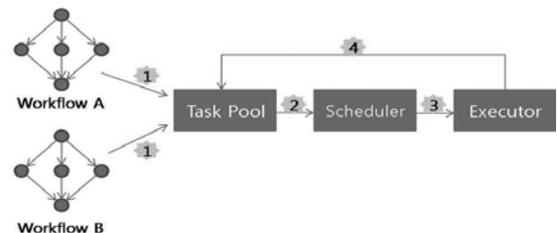
### 4.1 워크플로우 스케줄링 시스템

클라우드 컴퓨팅 시스템 환경은 (그림 3)과 같이 사용자의 워크플로우 어플리케이션이 스케줄링 시스템에 제출되면 워크플로우의 태스크들을 스케줄링 한다. 스케줄링 된 워크플로우의 태스크들은 적절한 자원을 할당받고 클라우드 서비스를 수행한다.



(그림 3) 워크플로우 스케줄링

각 워크플로우의 태스크들을 동적으로 스케줄링 하여 서비스 할당 결정을 최적화하기 위해, 제안하는 스케줄링 시스템은 3가지 요소, 즉, 태스크 풀(Task Pool), 스케줄러(Scheduler) 및 실행기(Executor)들로 구성된다. (그림 4)는 워크플로우 스케줄링 시스템을 나타낸 것이다.



(그림 4) 워크플로우 스케줄링 시스템

태스크 풀은 동적인 클라우드 서비스를 위해 다중 워크플로우를 관리한다. 태스크들이 다른 워크플로우들에 속할 지라도, 실행기는 태스크 풀에 있는 태스크들에 제안하는 스케줄링 알고리즘을 적용할 수 있다[11]. 따라서, 태스크 풀은 클라우드에 동적으로 제출되는 다중 워크플로우들을 스케줄링 할 수 있다.

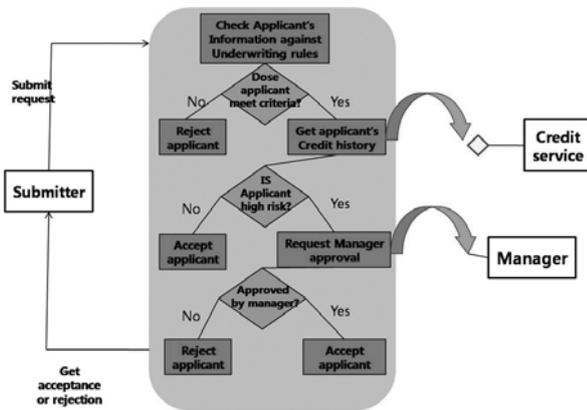
스케줄러는 제안하는 다중 워크플로우 스케줄링 전략에 따라 큐에 있는 태스크들을 계산한다.

실행기는 큐에 있는 태스크들을 실행하기 위해 최적의 자원 또는 서비스를 선택한다. 임의의 태스크가 성공적으로 서비스가 완료되면, 실행기는 태스크 풀에 태스크 완료 상태를 공지한다. 그러면, 태스크 풀은 그에 상응하는 태스크를 태스크 풀에서 제거한다. 이러한 3가지 요소들 사이에서 지속적이고 동적인 통신에 의해 협업이 수행된다.

(그림 4)의 스케줄링 시스템의 순서는 다음과 같다. 첫 번째, 워크플로우가 스케줄링 시스템에 제출된다. 두 번째, 태스크 풀은 워크플로우의 각 태스크들의 속성(비용, 처리율, 실행시간 등)들을 계산한다. 세 번째, 스케줄러는 전략에 따라 스케줄링 한다. 네 번째, 실행기는 스케줄링 순서에 따라 최적의 자원을 선택하여 서비스를 수행하고 태스크 풀에 태스크 서비스 완료상태 공지한다.

4.2 워크플로우 어플리케이션

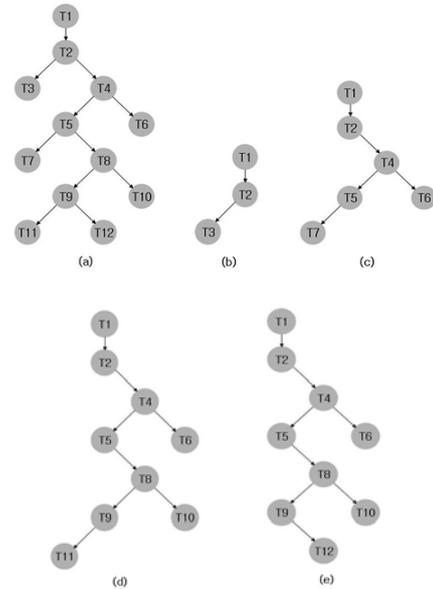
제안하는 스케줄링 알고리즘의 평가를 위해 실제 워크플로우 어플리케이션을 구성하였다.



(그림 5) 워크플로우 어플리케이션 모델: 자동차 보험 증명서 요청

보험 회사를 위한 워크플로우 어플리케이션을 생성하는 경우를 가정한다. 또한, 이 보험회사의 시스템은 전체적으로 클라우드 컴퓨팅 환경으로 구성되어 있다고 가정한다. (그림 5)는 자동차 보험 증명서를 요청하는 어플리케이션을 보여준다[13].

제출자가 새 증명서 요청을 전송하면서 프로세스가 시작된다. 이 제출자는 콜센터의 직원, 현장의 보험 대리인 또는 인터넷을 통해 요청하는 고객 중 한 사람이 된다. 새 요청



(그림 6) 자동차 보험 증명서 요청 워크플로우: (a) 전체 워크플로우, (b) 시나리오 1, (c) 시나리오 2, (d) 시나리오 3, (e) 시나리오 4

이 도착하면 이 워크플로우의 새 인스턴스가 만들어진다. 워크플로우는 요청 정보를 회사의 증명서 발행 규칙과 비교확인하면서 시작된다. 신청자가 회사의 기준에 못 미친다면, 신청은 거절된다. 기준에 도달한다면, 워크플로우는 외부 신용 서비스 기관에게 신청자의 신용 경력을 요청한다. 만족스러운 신용점수는 즉시 승인이 되지만, 신용점수가 나쁘고 높은 위험의 신청자는 관리자의 승인이 필요하게 된다. 관리자가 승인하면 신청이 받아들여지지만 그렇지 않다면 신청이 거부된다.

(그림 6)의 (a)는 (그림 5)의 전체적인 워크플로우 그래프를 나타낸 것이다.

- 시나리오 1: 새로운 요청이 도착하면 이 워크플로우의 새 인스턴스가 만들어진다. 워크플로우는 요청 정보를 회사의 증명서 발행 규칙과 비교확인하면서 시작된다. 신청자가 회사의 기준과 다르기 때문에 신청은 거절된다.
- 시나리오 2: 신청자가 회사의 기준에 도달한다. 따라서 워크플로우는 외부 신용 서비스 기관에게 신청자의 신용 경력을 요청한다. 만족스러운 신용점수로 즉시 승인이 된다. 즉, 리스크가 높지 않기 때문에 승인된다.
- 시나리오 3: 신청자가 회사의 기준에 맞지 않기 때문에, 신청은 거절된다. 기준에 도달한다면, 워크플로우는 외부 신용 서비스 기관에게 신청자의 신용 경력을 요청한다. 신용점수가 나쁘고 높은 위험의 신청자는 관리자의 승인이 필요하게 된다. 따라서 관리자는 신용점수가 너무 나빠 승인을 거부한다.
- 시나리오 4: 신청자가 회사의 기준에 도달하여 워크플로우는 외부 신용 서비스 기관에게 신청자의 신용 경력을

요청한다. 신용점수가 나쁘고 높은 위험의 신청자는 관리자의 승인이 필요하게 된다. 따라서 관리자는 신용점수는 비교적 나쁘지 않기 때문에 요청에 승인한다.

## 5. 제안하는 알고리즘

이번 장에서는 제안하는 알고리즘을 위해 태스크의 랭크를 정하기 위한 우선 순위화 단계와 결정된 우선순위로 태스크를 서비스에 할당하는 서비스 선택 단계로 구성한다. 우선 순위화 단계에서는 워크플로우의 시작 태스크부터 마지막 태스크까지의 모든 태스크의 우선순위를 결정한다. 서비스 할당 단계에서는 각 클라우드 서비스에 태스크를 할당한다.

### 5.1 태스크 우선 순위화

제안하는 다중 워크플로우 스케줄링 알고리즘을 비교 평가하기 위해 Zhifeng Yu와 Weisong Shi[11]가 제안한 랭크 기반 알고리즘인 Planner-Guided 전략의 랭크 공식을 고려하여 본 논문에 적용할 변수로 수정한다.

$$rank(Task_i) = \bar{\tau}_i + \max_{Task_j \in succ(Task_i)} (\overline{c(i, j)} + rank(Task_j)),$$

$$i = 1, \dots, n \quad (\text{식 1})$$

$succ(Task_i)$ 는  $Task_i$ 의 자식 태스크의 집합이며  $\overline{c(i, j)}$ 는 에지의 평균 통신비용이고,  $\bar{\tau}_i$ 는 태스크  $Task_i$ 의 추정 실행 시간이다. Zhifeng Yu와 Weisong Shi[11]가 제안한 Planner-Guided 전략은 클러스터 환경에서 정의되었다. 즉, 본 논문에서는 클라우드 컴퓨팅 환경을 고려하기 때문에, 에지의 평균 통신비용인  $\overline{c(i, j)}$ 를 고려하지 않으며, 워크플로우 태스크 사이의 우선순위만을 고려한다. 따라서, 식 1을 식 2로 수정한다.

$$rank(Task_i) = \bar{\tau}_i + \max_{Task_j \in succ(Task_i)} rank(Task_j),$$

$$i = 1, \dots, n \quad (\text{식 2})$$

제안하는 비용기반 랭크스케줄링 (Cost-based Rank Scheduling, CRS) 알고리즘은 다중 워크플로우 스케줄링을 위한 것으로 동적 알고리즘이다. 따라서, 스케줄러에서 사용할 비용 값을 계산하기 위한 식을 정의한다. 식 3처럼 처리율을 이용하는데 처리율은 태스크의 예상 실행 시간  $\bar{\tau}_i$ 를 이용하여 계산된다.

$$Cost(Task_i) = Length(Task_i) \times Tput(Task_i),$$

$$Tput(Task_i) = 1 / \bar{\tau}_i, \quad i = 1, \dots, n \quad (\text{식 3})$$

여기서 처리율  $Tput(Task_i)$ 는 각 태스크 별 처리율을 나타내며,  $Length(Task_i)$ 는 각 태스크 별 크기를 나타낸다. 워크플로우가 스케줄링 시스템에 제출되면 각 워크플로우의 시작 태스크부터 마지막 태스크까지 모두 우선순위를 결정한다.

### 5.2 서비스 선택

제안하는 알고리즘은 식 3에서 정의된 특성에 따라서 모든 준비된 태스크들을 스케줄링 하여 자원 또는 서비스에 할당한다. 따라서 다음과 같이 서비스 선택 단계를 설명한다.

- 워크플로우 제출: 임의의 새로운 워크플로우가 워크플로우 스케줄링 시스템에 도착하면, 워크플로우들은 태스크 풀에 제출된다. 모든 태스크들의 추정 실행 시간 및 비용 같은 특성을 계산 한 후, 태스크들은 태스크 풀의 큐에 저장된다. 큐에 있는 첫 번째 태스크가 스케줄러에 제출된다. 실행기가 태스크의 완료 공지가 있을 때마다, 태스크 풀은 임의의 자식 태스크들이 준비하고 있는지 아닌지를 검사한 다음 스케줄러에 제출된다. 태스크의 특성들 즉, 예상 실행 시간, 비용, 지연시간 같은 정보는 태스크와 함께 제출된다(처음 대기상태 태스크는 0으로 세팅).
- 태스크 스케줄링: 이용할 수 있는 서비스가 존재하며 태스크의 정보가 계산되어 태스크 풀에서 기다리고 있을 때 마다, 스케줄러는 각 알고리즘별 우선순위로 재계산하여 모든 태스크를 정렬하며 이것을 반복적으로 수행한다. 제안하는 스케줄링 알고리즘은 스케줄러에서 모든 태스크들의 계산된 비용을 가지고 스케줄링 한다. 스케줄러에서 다음과 같이 반복 수행하며, 실행중인 태스크는 1로 세팅된다.
  - (1) 태스크 풀의 큐에 있는 첫 번째 태스크를 제거한다.
  - (2) 가장 낮은 비용을 가진 태스크들을 스케줄링 한다. 만일 비용 값이 같다면, 더 높은 지연 시간 값을 가진 태스크가 우선적으로 스케줄링 된다.
- 서비스 선택: 실행기는 어떤 서비스가 비어 있는지 확인한다. 적절한 서비스에 태스크를 할당한다. 만일 태스크를 완료할 수 있는 서비스가 존재하지 않는다면 다른 라운드 즉 서비스로 태스크를 삽입한다. 모든 서비스는 동시에 하나의 태스크 만을 실행한다.
- 서비스 완료 공지: 서비스가 성공적으로 완료되면 실행기는 서비스 완료 상태를 태스크 풀에 공지하고, 그에 상응하는 태스크들을 태스크 풀에서 제거한다. 즉, 완료된 태스크는 2로 세팅하여 태스크 풀에 반환되고 태스크 풀은 해당 태스크를 제거한다.

## 6. 성능 분석 및 평가

본 논문에서 제안하는 알고리즘의 성능을 평가하기 위해, [11]의 두 종류의 Planner-Guided 스케줄링 알고리즘과 비

교평가 한다. [11]에서 제안한 두 알고리즘은 단일 워크플로우 스케줄링 알고리즘으로 알려진 RANK\_HF (Highest Rank First)와 하이브리드 우선순위 알고리즘 RANK\_HYBD(Rank Hybrid) 이다.

RANK\_HYBD는 태스크 풀에 있는 태스크가 다중 워크플로우인지를 검사한다. 만일 단일 워크플로우이면 RANK\_HYBD는 RANK\_HF와 동일하다. 하지만, 다중 워크플로우가 검사 된다면, 태스크 풀에서 식 2와 같이 랭크 값을 정한 후 스케줄러에게 보내진다. 스케줄러에서는 랭킹이 낮은 태스크로 정렬된 후 실행기에게 보내진다.

제안한 알고리즘은 태스크 풀에서 식 3과 같이 각 태스크의 속성들을 계산하여 스케줄러에 보내진다. 스케줄러는 비용이 낮은 우선순위로 스케줄링 하여 실행기에게 보내진다. 또한, 만약 같은 비용 값이 존재하면, 들어온 순서 즉, 지연시간 값이 큰 태스크를 우선적으로 스케줄링 된다. 지연시간은 워크플로우 제출 시 미리 계산된다.

6.1 실험 환경

클라우드 환경은 20개의 서비스 구성되며, 모든 서비스는 동시에 하나의 태스크를 실행할 수 있다. 10명에서 100명의 사용자가 클라우드 환경을 이용하는 것으로 가정하여 모의 실험에서는 10, 15, 20, 30, 50, 80, 100개의 워크플로우 어플리케이션을 동시에 발생시켰다.

워크플로우 어플리케이션은 3.2절에서 설명한 것처럼 워크플로우로 표현된다. 워크플로우를 랜덤하게 구성하며 여러 워크플로우 그래프 요소들로 구성된다. 시나리오에 따라 랜덤하게 4가지 종류의 워크플로우를 발생시킨다.

워크플로우 크기는 임의로 랜덤하게 3개에 9개의 태스크로 구성된 워크플로우 그룹을 사용한다. 또한 워크플로우의 태스크 크기는 50~1000 단위만큼 생성하였으며 도착간격은 포아송 분포를 따라 워크플로우 스케줄링 시스템에 도착한다. 모든 클라우드 서비스는 동시에 하나의 태스크를 실행할 수 있으며, 태스크의 크기가 100단위가 서비스를 수행하는데 걸리는 시간을 1초, 1000단위는 10초로 설정하였다.

각 알고리즘 별 테스트는 약 100회를 실시하였다. 즉, 최대 워크플로우 100개에 약 1200 태스크를 가진 워크플로우의 태스크를 가지고 각각의 알고리즘을 적용 테스트를 수행하였다.

6.2 실험 결과

제안하는 CRS 알고리즘과 RANK\_HF, RANK\_HYBD 알고리즘은 랜덤하게 생성된 각 시나리오별로 비교 평가한다. 각 워크플로우 또는 시나리오 별로 동시에 생성되면 각 워크플로우 별 평균 총 처리시간과 평균 가용성을 계산한다.

6.2.1 평균 총 처리시간

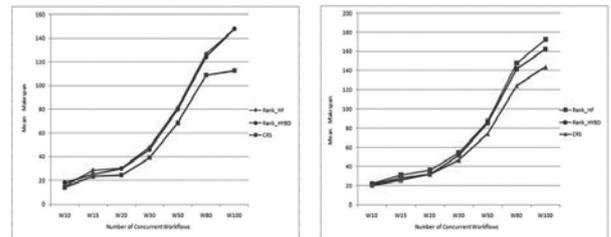
평균 총 처리시간은 워크플로우 어플리케이션이 시스템에

들어와 스케줄링 되어 서비스를 받고 실행이 완료된 총 처리시간 또는 실행시간을 의미한다.

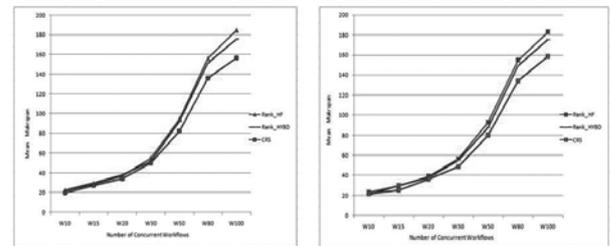
(그림 7)과 (그림 8)은 시나리오 별로 평균 총 처리시간을 보여준다. 특히, 각 시나리오의 워크플로우가 80개일 때 평균 총 처리시간이 차이가 많음을 보여준다.

RANK\_HF, RANK\_HYBD 알고리즘은 워크플로우 수가 증가 할수록 그 차이는 작지만, RANK\_HYBD가 RANK\_HF보다 평균적으로 처리시간이 작다는 것을 알 수 있으며, CRS 알고리즘은 두 알고리즘 RANK\_HF, RANK\_HYBD보다 워크플로우 수가 증가할수록 평균 총 처리시간이 작다는 것을 알 수 있다.

특히, CRS 알고리즘은 RANK\_HF 보다 평균 14%, RANK\_HYBD 보다는 10% 향상된 결과를 나타내었다. 워크플로우 수가 증가하면 할수록 그 차이는 더 증가함을 볼 수 있다.



(그림 7) Mean Makespan: (a)시나리오 1, (b)시나리오 2



(그림 8) Mean Makespan: (c)시나리오 3, (d)시나리오 4

6.2.2 평균 가용성

평균 가용성은 워크플로우가 클라우드 시스템에서 자원, 또는 서비스를 액세스 할 수 있는 평균 가용성을 의미한다. 본 논문에서는 워크플로우 어플리케이션이 동시에 여러 개 발생되기 때문에 워크플로우 수가 증가할수록 서비스 받기 위한 경쟁은 심화되며, 그 경쟁은 피할 수 없다. 즉, 서비스 받기 위한 경쟁이 심화될수록 클라우드 서비스 가용성은 하락할 것이다. 따라서, 가용성 메트릭은 오늘날 시스템 설계에 중요한 요소가 되기 때문에 서비스 공급자 측면에서 관심 있는 평가 메트릭이 된다[12][14].

가용성과 같은 평가 메트릭은 총 처리시간 메트릭을 보완할 수 있는 메트릭 정보라 할 수 있다. 가용성은 클라우드

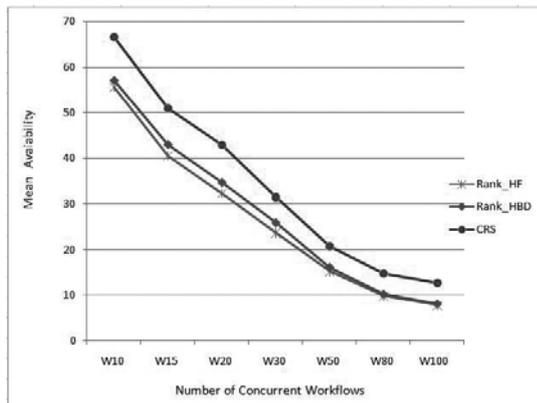
서비스가 즉각 사용할 준비가 되었는지를 판단하는 품질 평가이다. 가용성은 서비스를 이용할 확률로 나타낸다. 가용성이 큰 값들은 서비스가 항상 사용할 준비가 되었다는 것을 의미하며, 더 작은 값들은 특정 시간에 서비스를 이용할 수 있을지 불확실하다는 것을 의미한다[16]. 따라서 모의실험에서 다음과 같이 가용성을 계산한다.

$$Availability = \frac{UpTime}{Period\ of\ Time} = \frac{Mean\ Service\ Time}{\sum_{i=1}^n Task\ RTime},$$

$i = 1, \dots, n, \quad i \text{ is Task number of Workflow}$  (식 4)

여기서 UpTime의 의미는 태스크들이 정상적으로 실행 중인 상태를 의미한다. 즉, 태스크가 클라우드 서비스를 받는 시간을 의미한다. Period of Time은 실행시간으로, 이 시간은 태스크들의 총 실행 시간 합계에서 태스크들의 지연 시간 중복을 제외한 시간으로 설정하였다.

(그림 9)는 RANK\_HF, RANK\_HYBD 및 CRS 알고리즘의 모의실험 결과 동시에 발생하는 워크플로우 수에 따라 가용성은 전체적으로 하락하는 것을 볼 수 있다. 특히, W20이 발생된 구간을 보면 서비스가 20개이고 워크플로우가 20개인 경우 CRS 알고리즘만이 가용성이 높게 나타나고 있다. CRS 알고리즘이 전체 워크플로우 발생 수에 따라 태스크들의 서비스 가용성이 RANK\_HF와 RANK\_HYBD 알고리즘 보다 크다는 것을 볼 수 있다. 특히 제안하는 CRS 알고리즘이 RANK\_HF 보다 약 평균 32%, RANK\_HYBD보다 27% 정도 서비스 가용성이 높다는 것을 알 수 있다.



(그림 9) 평균 가용성

## 7. 결론

본 논문에서는 클라우드 컴퓨팅 환경을 고려한 다중 워크플로우 스케줄링 문제에 비용 기반 랭크 알고리즘을 적용하는 구체적인 방법을 제시하였다. 즉, 다중 사용자들을 위한 다중 워크플로우와 다양한 QoS 파라미터를 사용하여 워크

플로우 스케줄링 시스템과 스케줄링 알고리즘을 제안했다. 또한, 이러한 클라우드 컴퓨팅 환경에서 워크플로우 스케줄링을 위해 실제 워크플로우 어플리케이션을 모델링하여 워크플로우를 시나리오 별로 구성하였다.

본 논문에서 제안하는 알고리즘은 다중 워크플로우 어플리케이션을 위한 비용 기반 랭크 스케줄링 알고리즘으로서, 태스크 크기와 추정 실행시간(처리율)으로 계산된 비용을 이용하여 다중 워크플로우 스케줄링 방법을 제시하였다.

제안하는 알고리즘의 성능 평가를 위해 본 논문에서는 기존의 다중 워크플로우 스케줄링 기법인 Planner-Guided 기법의 RANK\_HF와 RANK\_HYBD 스케줄링 알고리즘을 고려하였다. 또한, 이러한 워크플로우 스케줄링 알고리즘의 비교 평가를 위한 QoS 메트릭으로, 서비스 공급자와 사용자 모두에게 중요한 평균 총 처리시간(Mean Makespan)과 서비스 공급자 측면에서 클라우드 서비스의 평균 가용성(Mean Availability)을 측정하였다.

실험 결과 제안한 CRS 알고리즘이 평균 총 처리 시간과 평균 가용성 면에서 RANK\_HF 및 RANK\_HYBD 알고리즘보다 더 좋은 평가 결과를 보였다.

향후 연구과제로 다양한 워크플로우 어플리케이션 모델 적용을 위한 연구와 시스템의 신뢰성을 위한 다양한 QoS 파라미터를 측정하는 연구가 요구된다.

## 참고 문헌

- [1] 강영준, 박세권, 류승완, “클라우드 컴퓨팅 기술 동향”, IT 기획시리즈 차세대 컴퓨팅(2) 정보통신 진흥원 주간기술동향 통권 1433호, pp29-38, 2010.2.17.
- [2] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic, “Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility”, Future Generation Computer Systems, Elsevier Science, Amsterdam, June 2009, Volume 25, Number 6, pp. 599-616.
- [3] Ian Foster, Yong Zhao, Ioan Raicu and Shiyong Lu, “Cloud Computing and Grid Computing 360-Degree Compared”, Grid Computing Environments Workshop 2008(GCE '08).
- [4] L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner, “A Break in the Clouds: Towards a Cloud Definition”, vol 39, pp.50-55, Jan. 2009.
- [5] H. Zhao and R. Sakellariou, “Scheduling multiple dags onto heterogeneous systems”, in Proceedings of the 15th Heterogeneous Computing Workshop (HCW), Rhodes Island, Greece, April 2006.
- [6] J. Yu and R. Buyya, “Workflow Scheduling Algorithms for Grid Computing”, Metaheuristics for Scheduling in Distributed Computing Environments, F. X. a. A. Abraham, ed., Springer, 2008.

[7] U. Höning and W. Schifmann, "A meta-algorithm for scheduling multiple dags in homogeneous system environments", in Proceedings of the 18th International Conference on Parallel and Distributed Computing and Systems (PDCS'06). IEEE, 2006.

[8] M. Wicczorek, R. Prodan, and T. Fahringer, "Scheduling of Scientific Workflows in the ASKALON Grid Environment", ACM SIGMOD Record, 34(3):56-62, Sept. 2005.

[9] J. Blythe et al., "Task scheduling strategies for workflow-based applications in grids", in Proceedings of the IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05), Cardiff, UK, 2005.

[10] H. Topcuoglu, S. Hariri, and M. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing", IEEE Transactions on Parallel and Distribution Systems, vol. 13, no. 3, pp. 260 - 74, 2002.

[11] Zhifeng Yu and Weisong Shi, "A Planner-Guided Scheduling Strategy for Multiple Workflow Applications", icppw, pp.1-8, International Conference on Parallel Processing - Workshops, 2008.

[12] K. Xiong and H. Perros, "SLA-based resource allocation in cluster computing systems", In Proceedings of the IEEE IPDPS, 2008.

[13] David Chappell, Chappell & Associates, "Introduction Windows Workflow Foundation", Microsoft, September 2007.

[14] G. Malewicz, A. Rosenberg, and M. Yurkewych, "Toward a theory for scheduling dags in internet-based computing", IEEE Transactions on Computers, vol. 55, no. 6, pp. 757 - 68, 2006.

[15] Jablonski, S. and C. Bussler, "Workflow Management Systems: Modeling, Architecture, and Implementation", Thomson Press, 1996.

[16] H-L. Truong, R. Samborski, T. Fahringer, "Towards a Framework for Monitoring and Analyzing QoS Metrics of Grid Services", Proceedings of the Second IEEE International Conference on e-Science and Grid Computing. May, 2006.



**최 경 근**

e-mail : clove38@empal.com

1998년 목원대학교 수학과 졸업(학사)

2001년 동국대학교 통계학과(이학석사)

2003년 동국대학교 정보통신공학과  
(박사수료)

2011년 대전대학교 정보통신공학과  
(공학박사)

관심분야 : 그린 IT, Cloud Computing, 네트워크 보안



**이 봉 환**

e-mail : blee@dju.kr

1985년 서강대학교 전자공학과 졸업(학사)

1987년 연세대학교 전자공학과(공학석사)

1993년 Texas A&M 대학교 전기 및 컴  
퓨터공학과(공학박사)

1995년 3월~현 재 대전대학교 정보통신  
공학과 교수

관심분야 : Cloud Computing, 네트워크 보안, u-Health System