

# 클라우드 컴퓨팅에서 효율적인 작업 스케줄링 알고리즘

최경근, 이봉환  
대전대학교 정보통신공학과  
e-mail: clove38@empal.com, blee@dju.kr

## An Efficient Task Scheduling Algorithm for Cloud Computing

Gyeong-Geun Choe, Bong-Hwan Lee  
Dept. of Information and Commun. Eng., Daejeon University

### 요 약

클라우드 컴퓨팅 환경에서 사용자들이 사용하는 다양한 어플리케이션은 워크플로우들로 표현된다. 이러한 구조에서의 클라우드 어플리케이션은 워크플로우의 각 작업에 따라 클라우드 서비스가 수행된다. 클라우드 서비스는 동시에 많은 사용자들의 어플리케이션인 다중 워크플로우가 발생되어 워크플로우 내의 작업들이 적절하게 서비스 되어야한다. 따라서, 본 논문에서는 클라우드 컴퓨팅을 고려한 다중 사용자의 워크플로우의 작업 스케줄링 기법을 제안한다.

### 1. 서론

최근 인터넷 사용자가 급증한 가운데, 사용자 및 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 기업들이 비용절감을 위한 전략중 하나로 클라우드 컴퓨팅에 관심을 갖기 시작했다[1]. 클라우드 컴퓨팅은 서버, 스토리지, 네트워크, 어플리케이션 등과 같은 컴퓨팅 자원과 비즈니스 자원을 포함하는 구조에서 적절한 자원을 선택하고 동적으로 구성한 후 워크로드가 이 위에서 수행될 수 있도록 해야 한다[2]. 즉, 사용자들이 사용하는 어플리케이션들의 데이터의 변형인 워크플로우들이 자원들에 적절하게 할당되어야한다.

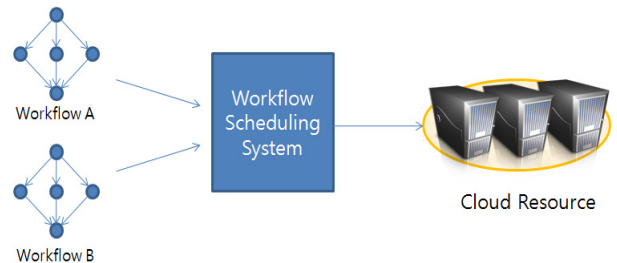
워크플로우(Workflow)는 작업 절차를 통한 정보 또는 업무의 이동을 의미하며, 작업 흐름이라고도 부른다[3]. 이러한 워크플로우가 클라우드 컴퓨팅 환경에서 효율적으로 수행되기 위해서는 부분 작업들이 클라우드 자원 또는 서비스에 적절하게 할당되어야한다[4].

따라서, 본 논문에서는 클라우드 컴퓨팅 환경을 고려한 효율적인 작업 스케줄링 알고리즘을 제안한다.

### 2. 문제정의

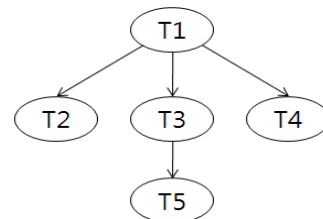
본 논문에서 제안한 스케줄링 알고리즘을 위해 필요한 시스템 환경, 워크플로우, 파라미터를 정의한다. 클라우드 컴퓨팅 시스템 환경은 그림 1과 같이 사용자의 워크플로우 어플리케이션이 스케줄링 시스템에 제출되면 스케줄링

시스템은 워크플로우 프로세싱을 위해 워크플로우의 작업들을 스케줄링한다. 스케줄링 된 워크플로우의 작업들은 적절한 자원을 할당받고 클라우드 서비스를 수행한다.



(그림 1) 워크플로우 스케줄링 시스템

워크플로우의 표현이 매우 다양하지만 가장 일반적이고 단순한 워크플로우 표현은 그림 2와 같이 비순환 그래프, DAG(Directed Acyclic Graph)를 사용한다[3].



(그림 2) DAG(Directed Acyclic Graph)의 예

DAG는 노드와 에지로 구성되는데, 노드는 계산결과로 표현되고 에지는 노드사이의 우선권을 표현한다. 각 DAG는 단일 입구 노드와 단일 출구 노드를 갖는다. 각 DAG에는 노드가 실행될 수 있고 노드들에 대한 다양한 시간이 필요한 클라우드 자원 또는 클라우드 서비스의 셋(set)이 존재한다. 클라우드 서비스는 한 번에 하나의 노드만을 실행할 수 있으며, 노드는 상위 노드의 모든 실행이 끝날 때까지 실행될 수 없다[4][5]. 본 논문에서 DAG는 워크플로우로, 노드는 작업으로 기술되며, 한 번에 하나의 작업만이 클라우드 서비스 자원을 사용할 수 있다고 가정한다. 또한 에지는 작업 사이의 통신 비용을 고려하지 않고, 작업 사이의 우선권을 표현한다.

워크플로우 스케줄링 알고리즘을 위해 각 작업의 랭크 값을 계산한다.

$$rank(Task_i) = \bar{r}_i \quad (1)$$

$$rank(Task_i) = \bar{r}_i + \max_{v_j \in succ(v_i)} (Task(v_j)) \quad ,$$

$$Task(v_i) = rank(Task_i), \quad i = 1, \dots, n \quad (2)$$

식(1)과 같이 워크플로우의 각 작업의 예상 실행시간은  $\bar{r}_i$ 로 표현되며,  $Task_i$ 의 예상 실행 시간이다. 식(2)에서  $succ(v_i)$ 는 작업  $v_j$ 의 하위 작업의 집합이며,  $Task(v_i)$ 는 하위 작업들의 랭크 값이다.

본 논문에서 클라우드 컴퓨팅 환경에 적합한 동적인 환경을 고려하기 위해 워크플로우 스케줄링 시스템에서 작업 저장소인 작업 풀을 사용한다. 이것은 워크플로우 스케줄링 시스템은 서비스가 가능할 때 만 작업 풀에 저장되며 이 작업 풀에서 동적으로 도착하는 다중 워크플로우 어플리케이션들을 스케줄한다. 또한 작업 풀에 저장되기 전에 작업의 지연시간 및 예상 실행 시간의 추정치를 반영한  $\bar{r}_i$  값이 저장되고 이를 스케줄링 전략에 사용한다.

### 3. 제안된 스케줄링 알고리즘

본 논문에서 제안하는 알고리즘을 위해 작업의 랭크를 정하기 위한 우선순위 결정 단계와 결정된 우선순위로 작업을 서비스에 할당하는 선택 단계로 구성한다. 우선순위 결정 단계에서는 워크플로우의 시작 작업부터 마지막 작업까지의 모든 작업의 우선순위를 결정한다. 서비스 할당 단계에서는 기존의 연구에서 사용된 삽입기반 정책을 사용하여 각 클라우드 서비스에 작업을 할당한다.

#### 3.1 작업 우선순위 결정 단계

제안된 알고리즘을 위해 우선 비용을 계산하기 위한 처리율(throughput)은 예상 실행시간  $\bar{r}_i$ 를 이용하여 계산되며, 식 (4)와 같다.

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

$$Tput(Task_i) = 1 / \bar{r}_i, \quad i = 1, \dots, n \quad (4)$$

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

#### 3.2 서비스 선택 단계(자원 선택 단계)

제안된 알고리즘은 모든 준비된 워크플로우 내의 작업을 각각의 속성에 따라 정렬하며, 다음과 같이 워크플로우 스케줄링 시스템에서 스케줄링 된다.

- 1) 작업 풀: 새로운 워크플로우가 도착하면 워크플로우 별 작업이 제출된다. 그 다음 모든 준비된 작업 별로 예상 시간  $\bar{r}_i$ 을 계산한 후 작업 풀의 큐에 준비 작업을 삽입한다.
- 2) 스케줄러: 이용할 수 있는 서비스가 존재하며 작업의 정보가 계산되어 작업 풀에서 기다리고 있을 때 마다 스케줄러는 각 알고리즘별 우선순위로 재계산하여 모든 작업을 정렬하며 이것을 반복적으로 수행한다.
- 3) 작업 완료 공지: 작업이 성공적으로 완료되면 작업 완료 상태를 작업 풀에 공지한다.

### 4. 성능 분석 및 평가

생성된 워크플로우를 이용하여 제안한 알고리즘과 기존의 알고리즘인 RANK HF, RANK HBD와 성능 비교 평가를 한다. RANK HF와 RANK HBD 알고리즘은 식(2)를 이용하여 랭크 값을 결정한다[6].

#### 4.1 실험 환경

본 논문에서 제안하는 스케줄링 알고리즘의 성능 측정을 위해 클라우드 컴퓨팅 환경은 10개의 서비스를 가지며, 모든 서비스는 동시에 하나의 작업을 실행할 수 있다고 가정한다. 모의실험에서 5, 10, 15, 20, 25개의 워크플로우 어플리케이션을 동시에 발생시켰다.

워크플로우 크기는 임의로 랜덤하게 3개에 12개의 작업으로 구성된 워크플로우 그룹을 사용한다. 또한 워크플로우의 작업 크기는 50~1000 단위 만큼 생성하였다. 실험 횟수는 각 알고리즘 별 5회를 실시하여 평균 실행 시간을 측정하였다.

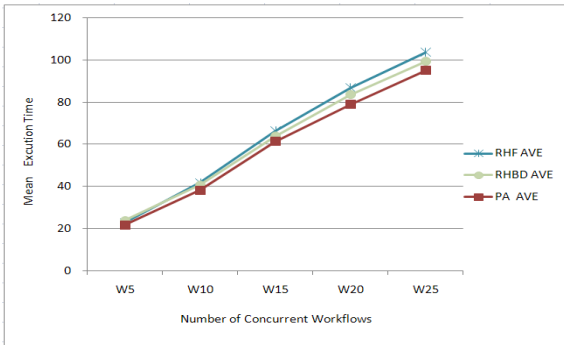
#### 4.2 실험 결과

표 1은 각 워크플로우 수에 따른 실행 시간을 보여준다. 각 태스크 별, 워크플로우 수에 따라 계산하였으며 같은 작업 수를 가지고 각 알고리즘 별 실행 시간을 계산하였다.

<표 1> 평균 실행시간

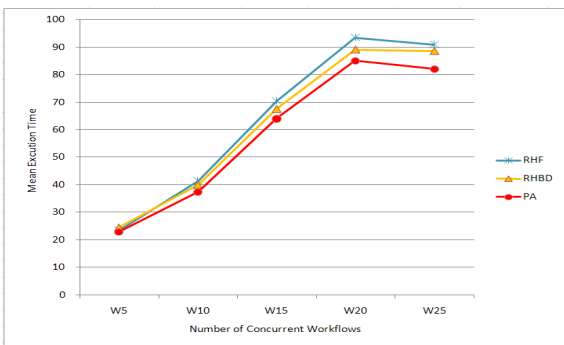
Algorithm	Test#	Task#	W5	Test#	Task#	W10	Test#	Task#	W15	Test#	Task#	W20	Test#	Task#	W25
RANK HF	Test 1	24	17	Test 1	60	40	Test 1	105	70	Test 1	138	86	Test 1	186	120
	Test 2	33	27	Test 2	54	41	Test 2	81	53	Test 2	132	83	Test 2	171	110
	Test 3	30	23	Test 3	57	40	Test 3	111	77	Test 3	153	103	Test 3	165	101
	Test 4	33	24	Test 4	66	43	Test 4	105	64	Test 4	129	84	Test 4	138	81
	Test 5	36	26	Test 5	69	45	Test 5	102	67	Test 5	126	78	Test 5	171	106
<b>RHF AVE</b>			<b>22.75</b>			<b>41.80</b>			<b>66.20</b>			<b>86.80</b>			<b>103.60</b>
RANK HYBD	Test 1	24	18	Test 1	60	40	Test 1	105	69	Test 1	138	83	Test 1	186	114
	Test 2	33	26	Test 2	54	40	Test 2	81	53	Test 2	132	81	Test 2	171	105
	Test 3	30	23	Test 3	57	38	Test 3	111	72	Test 3	153	98	Test 3	165	98
	Test 4	33	26	Test 4	66	42	Test 4	105	63	Test 4	129	80	Test 4	138	79
	Test 5	36	27	Test 5	69	44	Test 5	102	63	Test 5	126	76	Test 5	171	101
<b>RHBD AVE</b>			<b>24.00</b>			<b>40.80</b>			<b>64.00</b>			<b>83.60</b>			<b>99.40</b>
Proposed Algorithm	Test 1	24	14	Test 1	60	37	Test 1	105	66	Test 1	138	79	Test 1	186	111
	Test 2	33	24	Test 2	54	41	Test 2	81	51	Test 2	132	75	Test 2	171	100
	Test 3	30	22	Test 3	57	36	Test 3	111	70	Test 3	153	94	Test 3	165	93
	Test 4	33	24	Test 4	66	39	Test 4	105	58	Test 4	129	76	Test 4	138	71
	Test 5	36	25	Test 5	69	42	Test 5	102	62	Test 5	126	71	Test 5	171	100
<b>PA AVE</b>			<b>21.80</b>			<b>38.25</b>			<b>61.40</b>			<b>79.00</b>			<b>95.00</b>

그림 3의 그래프를 보면 워크플로우 수가 증가할수록 RANK HF와 RANK HBD 보다 제안하는 알고리즘 (PA:Proposed Algorithm)의 평균 실행 시간이 더 작다는 것을 알 수 있다. RANK HF 보다 평균 8%, RANK HBD 보다 평균 6% 평균 실행 시간이 작다.



(그림 3) 평균 실행 시간

표 1을 보면 각 워크플로우 별 작업 수는 랜덤하게 발생된다. 특히 작업 3과 4에서 W20 과 W25 결과만을 보면 작업 수가 153에서 165로, 129에서 138로 증가되었지만 실행시간은 감소되었다. 이것은 작업의 크기에 따라 실행 시간이 차이가 나게 되는 것이다(그림 4).



(그림 4) 테스트 3과 4에서 평균 실행시간

따라서 RANK HF와 RANK HBD는 계산 비용 즉, 실행 시간을 갖고 순위를 정하지만 실제 워크플로우 성능에 영향을 미치는 파라미터는 작업 크기가 성능에 영향을 준다. 따라서 제안된 알고리즘이 작업 크기와 실행시간(처리율)에 따른 순위 결정은 적절하다 할 수 있다.

## 5. 결론

본 논문에서는 클라우드 컴퓨팅 환경을 고려한 워크플로우 스케줄링 문제에 실행시간과 비용을 적용하는 구체적인 방법을 제시하였다. 이 방법은 기존의 단일 워크플로우 알고리즘인 RANK HF와 클러스터 환경에서의 다중 워크플로우 알고리즘인 RANK HBD의 스케줄링 방법을 써오므로 작업의 실행 시간뿐만 아니라 비용 같은 다른 파라미터를 사용하여 더욱 좋은 스케줄링 알고리즘을 찾을 수 있도록 한다.

제안된 알고리즘과 RANK HF, RANK HBD 알고리즘의 모의실험 결과 RANK HF, RANK HBD, 제안된 알고리즘 순으로 평균 실행 시간이 감소함을 알 수 있다.

## Acknowledgement

이 논문은 2010년도 교육과학기술부와 한국연구재단의 지역혁신인력양성사업과 일반연구자지원사업으로 수행된 연구 결과임 (No. 2010-0016574).

## 참고문헌

- [1] 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.
- [2] Ian Foster, Yong Zhao, Ioan Raicu and Shiyong Lu, "Cloud Computing and Grid Computing 360-Degree Compared", Grid Computing Environments Workshop 2008(GCE '08).
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] K. Xiong and H. Perros, "SLA-based resource allocation in cluster computing systems," In Proceedings of the IEEE IPDPS, 2008.