

# 유전 알고리즘을 활용한 워크플로우 스케줄링 기법에 관한 연구

장현규\*, 주경노\*, 윤찬현\*

\*한국과학기술원 전기및전자공학과

e-mail : {h.g\_jang, eu8198, chyoun}@kaist.ac.kr

## Performance Evaluation of the Workflow Scheduling with Using Genetic Algorithm

Hyun-gyu Jang\*, Kyung-no Joo\*, Chan Hyun Youn\*

\*Dept. of Electrical Electronic Engineering, KAIST

### 요 약

계놈을 위한 Bioinformatics 분석을 위해 순서가 있는 작업(Task)의 집합인 워크플로우를 특정 데드라인 이내로 수행할 것을 요청하면, 워크플로우 관리자는 데드라인을 지켜주면서 비용이 최소가 되는 방향으로 워크플로우를 수행하여야 한다. 본 논문은 클라우드 환경에서 유전알고리즘을 적용하여 이 워크플로우를 효율적으로 스케줄링하는 기법을 분석하고자 한다.

### 1. 서론

지구물리학, 천문학, 유전자정보학(Bioinformatics) 등의 분야에서는 다양한 이종(Heterogeneous) 데이터들을 다룬다. 그러한 데이터들을 처리하기 위한 복잡한 실험들은 상당한 컴퓨팅 인프라를 필요로 하게 되고, 그에 따른 비용도 많아지게 된다. 이러한 문제를 해결하기 위해 클라우드 컴퓨팅을 이용하는 사용자들이 많아지게 되었고, 이 클라우드 컴퓨팅 자원을 최소한의 비용으로 특정 데드라인을 지키면서 사용하기 위해 수많은 워크플로우 스케줄링 알고리즘들이 연구되고 있다. 여기서 과학 워크플로우(Scientific Workflow)는 의존성(Dependency)을 가지는 데이터들을 처리하기 위한 일련의 작업(Task)들을 실행하는 자동화된 과학적 프로세스를 말한다 [1].

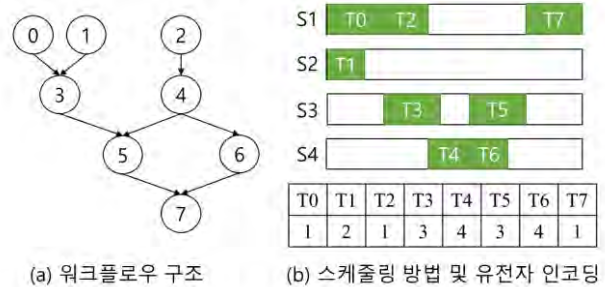
위에 언급된 여러 분야 중에서도 유전자정보학에서는 유전자에 대한 데이터들을 처리하기 위한 여러 가지 알고리즘들이 연구되고 있다. 본 논문에서는 유전 알고리즘을 활용하여 계놈 워크플로우를 효율적으로 스케줄링하는 기법을 분석할 것이다.

### 2. 유전 알고리즘과 워크플로우 스케줄링

유전 알고리즘(Genetic Algorithm)은 자연 환경의 유전적 진화 과정에 기반한 최적화 기법이다. 유전 알고리즘은 임의의 유전자 집단을 만든 후, 선택 - 교차 - 변이의 과정을 거쳐 다양한 유전자들로 변화시킨다. 살아남은 유전자들은 다음 세대를 이루게 되며, 위의 과정이 반복된다. 세대를 거치면서 생존에 적합한 유전자들이 살아남으며, 결국 최적해에 근접한다는 것이 유전알고리즘의 기본 원리이다 [2].

워크플로우 스케줄링은 워크플로우 내의 작업들을 적절한 클라우드 자원과 매핑(Mapping)시키는 문제이다. 비용(Cost), 실행 시간(Makespan) 등 고려할 변수가 많기 때문에 NP-Hard 문제로 알려져 있다 [3]. 워크플로우 스케줄링 문제를 해결하기 위해 유전 알고리즘을 적용할 수 있다. 유전자가 주어진 워크플로우를 스케줄링하는 정보를 담고 있을 때, 유전 알고리즘을 통해 주어진 적합함수(Fitness function)를 최소화시키는 스케줄링 방법을 찾아낼 수 있다.

각 유전자가 스케줄링 정보를 담고 있어야 하는데 기존의 유전 알고리즘에서 사용하던 Bit encoding 방식은 이를 나타내기에 정보가 부족하다. 따라서 본 논문에서는 Permutation encoding 기법을 사용하였다. 유전자의 i 번째 요소는 i 번째 작업이 어떤 VM에 배치되었는가를 나타낸다. 그림 1은 워크플로우 스케줄링을 위한 유전자 인코딩 방법을 나타낸다. 그림 1(a)의 워크플로우 구조가 그림 1(b) 상단과 같이 배치된 경우, 그림 1(b) 아래와 같이 유전자로 나타낼 수 있다.



(그림 1) 유전자 인코딩 방법

그림 1 과 같은 방법으로 인코딩된 유전자의 적합도를 알기 위해서는 문제에 맞는 적합함수를 정의해야 한다. 본 논문에서는 워크플로우 스케줄링의 적합도는 비용이 낮을수록 좋다고 생각하였다. 하지만, 스케줄링의 수행 시간(Makespan)은 반드시 사용자가 정한 데드라인을 만족해야 하기 때문에 만족하지 못한 경우에 대한 벌금(Penalty)를 추가로 고려하였다. 수식 1 은 유전자 ch 에 대한 적합함수를 나타낸다. ms(ch) 는 유전자 ch 의 수행 시간(Makespan)을 의미하며, DL 은 사용자가 원하는 수행 데드라인이다. cost(ch) 는 유전자 ch 의 방법으로 스케줄링할 때의 총 비용이다.

$$f(ch) = \max(ms(ch) - DL, 0) + cost(ch) \quad \text{수식 1}$$

유전자 풀에서 적합함수의 값이 낮은 유전자를 선택하는 과정이 필요한데, 선택 과정은 가장 적합도가 높았던 유전자를 다음 세대에 보존하는 Elitism 방식과 Fitness 함수에 따라 유전자를 임의로 선택하는 Roulette wheel 방식을 사용하였다. 또한, 유전자의 다양성을 넓혀 해가 Local minima 에 빠지는 것을 막기 위해 교차(Crossover)와 변이(Mutation) 기법을 추가하였다. 교차는 임의의 Single point 에서 두 유전자의 내용을 뒤바꾸는 Single-point crossover 방식을 사용하였으며, 변이는 유전자 내에서 임의의 두 점을 잡아 할당된 VM 을 교환하는 Swapping 방식을 적용하였다.

### 3. 실험 환경

유전자에 대한 정보들을 처리하는 작업(Task)들의 실행 시간(Execution Time)을 알아내기 위해 BWA (Burrows-Wheeler Aligner) 어플리케이션을 이용하였다 [4]. PM(Physical Machine)에서 VM(Virtual Machine)을 3 가지로 구분하여 생성한 후, 사설(Private) 클라우드 환경에서 실험하였다. 표 1 은 생성한 VM 종류에 따라 각각의 성능과 비용을 보여준다. VM 을 생성하는 환경인 PM 의 사양은 Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz (8-core)이다.

<표 1> VM 성능

VM 종류	VCPU(#)	Mem(GB)	비용
Small	1	2	2\$/h
Medium	2	4	3\$/h
Large	4	8	5\$/h

원래 사설 클라우드 환경에서는 비용이 없어야 되지만, 실험에서 임의로 정해준 이유는 대부분의 사용자는 공용(Public) 클라우드를 이용하기 때문에 그에 맞춰서 결과를 측정하기 위함이다.

<표 2> 시뮬레이션 변수(Parameter)

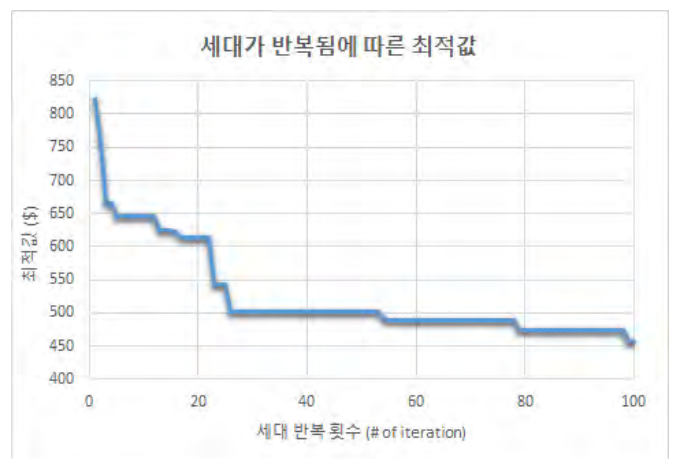
변수	값
VM 종류	3

작업(Task) 수	100
초기 집단 생성	100
교차 기법	Single-Point
변이 기법	Swapping
초기 집단 크기	100
반복 횟수	100
교차 확률	0.5
변이 확률	0.01

표 2 는 실험을 위해 필요한 Parameters 와 그 값을 보여준다. VM 은 Small, Medium, Large 각각 1 개씩 사용한다. 선택(Selection)에는 Roulette Wheel Selection, Rank Selection, Steady-state Selection, Elitism 등이 있다. 표 2 에는 따로 표기하지 않았지만, Roulette Wheel 과 Elitism 기법을 사용하였다. 이에 대한 설명은 앞서 언급했지만 보충하자면, 집단의 각 개체들의 Fitness 값이 우수할수록 다음 세대로 넘어갈 확률을 높게 하면서 가장 적합도가 높은 유전자를 다음 세대에 보존함으로써 최적값을 찾기 위한 반복 횟수를 줄일 수 있다. 교차 기법은 100 개의 개체 집단에서 50 쌍을 만들고 한 쌍끼리 유전체를 교환하게 될 때, 어느 부분을 교환할지에 대한 Parameter 이다. Single-Point 는 한 지점을 정한 후, 한 쌍끼리 앞부분이나 뒷부분을 서로 교환하는 것이다. 변이 기법은 한 개체 내에서 유전체를 어떻게 변형시킬 것인지에 대한 Parameter 이다. Swapping 은 한 개체 내에서 두 부분의 위치를 바꾸는 것이다. 교차와 변이는 확률을 정해주어서 처리한다. 왜냐하면 모든 집단의 개체들이 교차와 변이를 한다면 전체 실행 시간이 늘어나게 되고, 우수한 개체를 찾기가 힘들어지게 된다.

### 4. 실험 결과

그림 2 는 표에 따라 Parameter 들을 설정하여 유전 알고리즘을 적용한 결과이다. 그림 2 는 , 그림 3 은 유전 알고리즘에서 세대가 반복됨에 따른 최적값의 추이이다.



(그림 2) 세대가 반복됨에 따른 최적값

## 5. 결론

세대가 반복되면서 유전 알고리즘을 수행한 결과가 최적값에 점점 수렴하는 것을 확인할 수 있다. 30 세대 이전까지는 가파르게 감소하다가 이 후부터 천천히 감소한다. Parameter 와 작업(Task)의 특성에 따라 수렴하는 속도가 달라질 수 있지만, 반복 횟수를 늘린다면 최적값에 수렴하는 것을 확인할 수 있다. 아직 워크플로우에 활용되는 다른 알고리즘과 성능 비교를 하지 못했지만, 차후 알고리즘을 개선한 후 다른 알고리즘과의 비교를 진행할 예정이다.

### Acknowledgement

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업[10038768, 유전체 분석용 슈퍼컴퓨팅 시스템 개발]과 2012 년도 정부(교육과학기술부)의 재원으로 한국연구재단 - 클라우드 Collaboration 기술 사업과 BK21 사업의 일환[No. 2012-0006425]으로 수행하였음.

### 참고문헌

- [1] J. Yu and R. Buyya, "A taxonomy of scientific workflow systems for grid computing.", SIGMOD Record, vol. 34, no. 3, pp. 44-49, 2005
- [2] Lovejit Singh and Sarbjit Singh, "A Genetic Algorithm for Scheduling Workflow Applications in Unreliable Cloud Environment.", Second international Conference, SNDS 2014
- [3] Pinedo, M.L., "Scheduling: Theory, Algorithms, and Systems", Springer, 2008
- [4] Charlotte Herzeel, Pascal Costanza, Thomas J. Ashby and Roel Wuyts, "Performance Analysis of BWA Alignment", ExaScience life lab, November 5, 2013