클라우드 컴퓨팅 환경에서 신뢰성 기반 적응적 스케줄링 기법

조인석† · 유헌창††

요 약

클라우드 컴퓨팅은 인터넷 혹은 인트라넷 기반의 대규모 컴퓨팅 자원을 가상화하여 사용자가 원하는 서비스를 언제 어디서든 제공하도록 하는 컴퓨팅 패러다임이다. 이러한 클라우드 컴퓨팅은 시스템 환경 자체가 대규모의 데이터를 처리하며, 다중 사용자 접속 환경 기반이어서 시스템의 신뢰성이 중요한 요소이다. 본 논문에서는 클라우드 환경에서 발생하는 문제(사용자의 요구사항 변경, 자원 결함 발생 등)를 해결하기 위해 시스템 환경 내부의 자원 변화에 대처할 수 있고 결함 포용적인 신뢰성 기반 적응적 스케줄링 기법을 제안한다. 이 기법의 타당성을 검증하기 위해 CloudSim 시뮬레이션 환경에서 실험하였다.

주제어 : 클라우드 컴퓨팅, 신뢰성, 적응적 스케줄링 기법

Adaptive Scheduling Technique Based on Reliability in Cloud Compuing Environment

Inseock Cho† · Heonchang Yu††

ABSTRACT

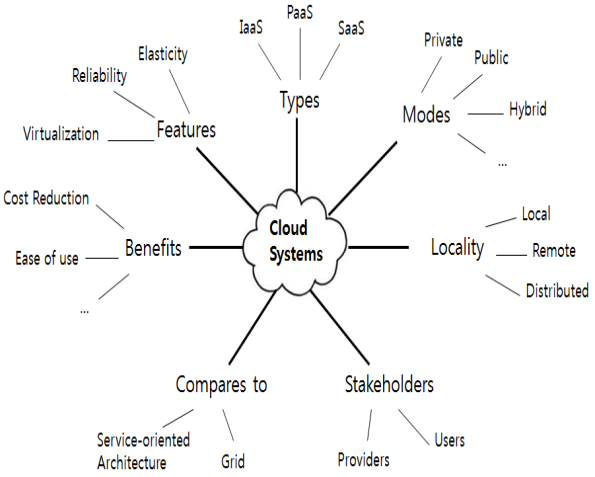
Cloud computing is a computing paradigm that provides user's services anywhere, anytime in a virtualized form composed of large computing resources based on internet or intranet. In Cloud computing environments, reliability of system is impact factor because many applications handle large data. In this paper, we propose an adaptive scheduling technique based on reliability with fault tolerance that manages resource variable and resolves problems(change of user's requirement, failure occurrence) in Cloud computing environment. Futhermore, we verified the performance of the proposed scheduling through experiments in CloudSim Simulation.

Keywords : Cloud Computing, Reliability, Adaptive Scheduling

† 종신회원: 고려대학교 컴퓨터교육과 박사과정
†† 종신회원: 고려대학교 컴퓨터교육과 교수(교신저자)
논문접수: 2010년 12월 27일, 심사완료: 2011년 01월 26일

1. 서 론

수십 년에 걸쳐 진행되어온 컴퓨팅 기술 진화 과정의 결과로 클라우드 컴퓨팅의 개념이 나타났다. 클라우드 컴퓨팅에 관한 다양한 정의들[1][2][3]이 있지만, 그리드 컴퓨팅[4][5]의 발전된 형태로 대규모 컴퓨팅 자원을 사용자들이 보다 쉽게 사용할 수 있도록 하는 컴퓨팅 환경을 뜻한다. 즉, 사용자는 언제 어디서나 원하는 서비스를 Pay-as-you-go(사용한 만큼 지불하는 방식)으로 이용한다. 또한 서비스의 품질 측면은 사용자와 서비스 제공자 간의 SLA(Service Level Agreement)를 통해 보장하고 있다. 이를 위해 인터넷이나 인트라넷을 통해 분산되어 있는 대규모 자원들을 하나의 컴퓨팅 자원처럼 보이게 해주는 가상화 기술도 사용한다.



<그림 1> 클라우드 시스템

이로 인해 클라우드 컴퓨팅은 효율성, 유연성, 규모의 경제를 통한 낮은 가격, 컴퓨팅 자원 소유에 대한 필요성 감소 등 다양한 장점을 제공하지만, 아직까지 해결해야 될 문제 또한 많은 것이 현실이다. 클라우드 컴퓨팅 서비스에서 해결해야 하는 큰 문제들 중 하나가 신뢰성이다.

클라우드 컴퓨팅에서는 시스템 상의 미미한 오류에도 심각한 경제적인 문제를 발생하거나 고객의 신뢰를 잃어버릴 수 있기 때문에 시스템이 항상 가용하도록 관리해야 하고, 다수의 고객을 대상으로 서비스를 제공하기 때문에 부하를 견디고 서비스를 안정적으로 제공해야 한다. 최근 아마존의 S3 서비스가 장애를 일으킨 예를 보면 서비스 안정성과 신뢰성이 클라우드 컴퓨팅에서 얼마나 큰 문제인지 알 수 있다.

이와 같이 클라우드 컴퓨팅에서 높은 가용성, 신뢰성을 제공하기 위해서 시스템 내부의 자원들의 상태 정보 관리 뿐만 아니라 자원의 결함이 발생하더라도 복구할 수 있는 방법이 필요하다.

이를 위해 본 논문에서는 클라우드 컴퓨팅 환경에서 발생하는 문제 (사용자 요구사항 변경, QoS 변경, 자원 결함 발생 등)를 해결하기 위해 시스템 환경 내부의 자원 변화에 대처할 수 있고 결함 포용적인 신뢰성 기반 적응적 스케줄링 기법을 제안한다. 이 기법의 타당성을 검증하기 위해 CloudSim 시뮬레이션 환경에서 실험하였다.

본 논문의 구성은 다음과 같다. 2장에서 클라우드 컴퓨팅 환경에 대한 관련 연구를 살펴본다. 3장에서는 제안된 시스템 구성과 기반이 되는 환경을 설명하고 본 논문에서 제안한 신뢰성 기반 적응적 스케줄링 기법에 대해 자세히 다루었다. 4장에서는 모의 실험을 통해 본 논문에서 제안한 스케줄링 기법의 타당성을 검증한다. 5장에서는 결론 및 향후 연구 과제에 대해 논의한다.

2. 관련 연구

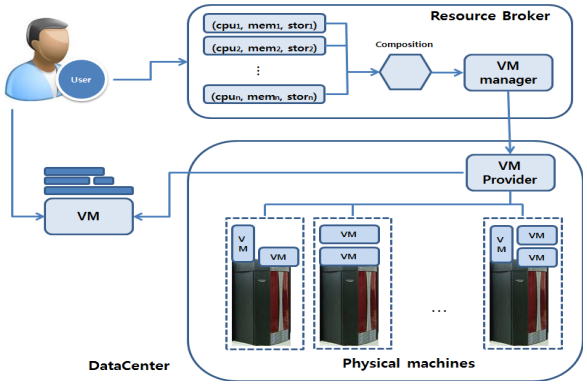
2.1 클라우드 컴퓨팅 환경

현재 Google EC2[6], Microsoft Asure[7]와 같은 기업들이 클라우드 컴퓨팅 분야를 이끌고 있고, 비즈니스적 관점에서 상용 서비스를 목적으로 하고 있기 때문에 클러스터 서버 등의 클라우드 컴퓨팅의 인프라 구축에 고성능 및 고비용의 서버들을 이용하고 있다. 그러나 이것은 클라우드 컴퓨팅을 위한 서버들을 구축하는데 적지 않은 비용이 소요되며, 클라우드 컴퓨팅의 발전에 걸림돌이 될 수 있다.

이에 별도로 클러스터 서버를 구입하지 않고도 기존의 데스크톱 PC 들을 연결하여 클라우드 컴퓨팅을 위한 인프라를 구축하는 방법이 필요하다. 이 데스크톱 PC들은 P2P 방식으로 연결될 수 있으며 클라우드 컴퓨팅 서비스를 제공하기 위한 자원이 된다. 또한 데스크톱 PC들을 클라우드 컴

퓨팅 자원으로 이용하려면 이들을 논리적으로 하나로 묶어줄 수 있는 방법이 필요하며, 이것이 바로 가상 파일 시스템[8]이다.

본 연구에서는 <그림 2>와 같이 클라우드 컴퓨팅의 인프라 구축을 위해 클러스터 서버가 아닌 기존의 데스크톱 PC들을 P2P 방식으로 연결하고, 가상의 단일 뷰를 제공함으로써 대용량 데이터를 저장하고 처리할 수 있는 기반이 되는 가상 파일 시스템을 이용하여 클라우드 컴퓨팅 환경을 구축하였다.



<그림 2> 가상 파일 시스템을 이용한 클라우드 컴퓨팅 환경 모델

2.2 스케줄링 기법에 대한 관련 연구

과학기술 분야에서 대규모의 문제를 해결하기 위해서 대규모 분산 컴퓨팅 기술이 이용되고 있다. 일반적인 분산 컴퓨터는 이기종의 컴퓨팅 노드로 이루어져 있기 때문에, 분산 컴퓨터의 자원을 효율적으로 사용하기 위해 성능이 우수한 스케줄링 알고리즘이 필요하다. 이기종 분산 컴퓨터를 위한 스케줄링 문제를 해결하기 위해서 기존 연구에서는 다양한 스케줄링 알고리즘들[9][10][11][12][13][14]을 제안하고 있다.

<표 1> 기존 스케줄링 기법과의 비교

기준 / 논문	비용	성능	결함	환경 구성
[11][12][13]	고려 안함	고려함	고려 안함	고려 안함
[14]	고려 안함	고려함	고려함	고려 안함
[15][16]	고려함	고려 안함	고려함	단일기종
본 논문	고려함	고려함	고려함	이기종

대규모 응용을 위해 많은 현존하는 자원 할당 및 태스크 스케줄링 정책들은 중매(Matchmaking)에 초점을 뒀으나 효율적인 할당 정책을 연구하지는 않았다. 워크플로우 기반 할당 정책들[9][10][11]은 중매 알고리즘보다 성능에 더 초점을 맞추었다. 그러나 이런 알고리즘들의 목적은 응용의 실행시간을 최소화하는 것이지 실행 비용은 신경쓰지 않았다.

Ramakrishnan[12] 에서는 그리드와 클라우드 환경에서 개별 작업들의 실행 성공 확률을 증가시키기 위해 워크플로우들을 복제하는 방법을 사용하여 결함 포용적인 워크플로우 스케줄링 기법을 제시하였다. 그러나 이 기법은 실행 시간은 다소 줄일 수 있지만, 기반 환경에 대한 비용을 고려하지 않았다. [13][14]에서는 자원 할당 비용과 제약 사항에 초점을 두었지만, 이기종의 자원 사항에 제한을 두지 못하였다. 본 연구에서는 자원 할당 비용과 성능의 두 가지를 모두 고려하였다.

본 논문에서 이질적인 분산 컴퓨터 환경에서 동작하는 새로운 신뢰성 기반 적응적 스케줄링 알고리즘을 제안하였으며 시뮬레이션을 통하여 기존의 스케줄링 알고리즘보다 성능이 뛰어남을 보였다.

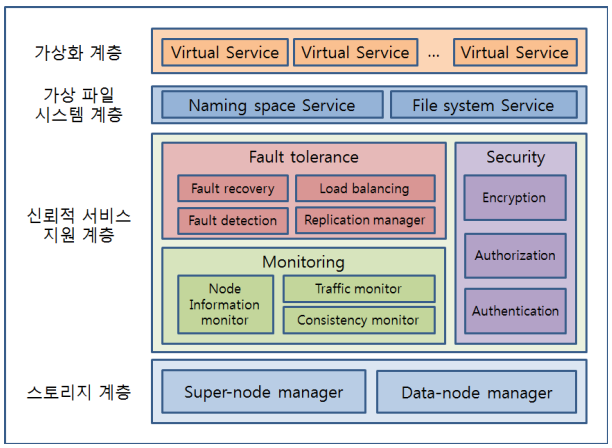
3. 신뢰성 기반 적응적 스케줄링 기법

현재 대부분의 클라우드 컴퓨팅 시스템은 대규모 클러스터 서버로 구성되어 있다. 그리드 컴퓨팅도 초기에는 슈퍼 컴퓨터들의 집합으로 구성되었으나 최근에 모바일 기기의 자원도 활용하도록 하는 모바일 그리드 컴퓨팅 환경으로 발전해가고 있는 것처럼, 클라우드 컴퓨팅 시스템도 현재의 서버들의 집합에서 일반 데스크톱 PC들의 자원도 활용하고 더 나아가서는 모바일 기기들의 자원도 활용하도록 컴퓨팅 환경이 변화되어 갈 것이다. 본 논문에서는 데스크톱 PC들을 클라우드 컴퓨팅 환경에 포함시켜 자원을 관리하는 방법까지 연구하였고, 추후 모바일 기기들의 자원 활용에 관하여 연구를 진행할 것이다.

3.1 시스템 모델

본 논문에서는 클라우드 컴퓨팅을 위한 인프라

를 구축하는 방법으로 데스크톱 PC들을 P2P 방식으로 연결하여 데이터 센터를 구축하고 클라우드 컴퓨팅 서비스를 제공하기 위한 자원으로 활용하였다. 이때, 데스크톱 PC들은 자율적이며 독립적인 자원으로, 각자 이질적인 컴퓨팅 환경을 가진다. 이를 위해 논리적으로 하나의 시스템으로 보여주기 위해 데스크톱 PC들의 자원을 하나의 단일 자원처럼 보여주는 가상 파일 시스템 부분과 클라우드 컴퓨팅 서비스에서 반드시 필요한 신뢰성과 보안 문제를 해결하기 위한 모듈을 추가하였다. 각각의 모듈은 모니터링(monitoring) 모듈, 결함 포용(fault tolerance) 모듈, 보안(security) 모듈 등 3부분으로 나누며, 각각의 역할은 다음과 같다.



<그림 3> 시스템 모델

* 모니터링 모듈 : 전체 시스템 상태와 각 서버의 상태를 주기적으로 체크하여 시스템 일관성을 유지할 수 있도록 하며, 한 서버에 미리 설정된 값보다 높은 트래픽이 발생할 경우 결함 발생 가능성이 높으므로 결함 포용 모듈에 알려준다.
* 결함 포용 모듈 : 주기적으로 서버 기록을 명시한 것을 바탕으로 필요한 정보 리스트를 백업하고, 한 서버에 트래픽이 많이 발생할 경우 부하 균형을 이루도록 다른 서버에 정보 리스트를 옮긴다. 작업의 복제 개수도 시스템 상태에 맞게 정한다.
* 보안 모듈 : 사용자 등급과 권한에 맞는 클라우드 자원을 제공한다.

3.2 신뢰성을 위한 결함 포용 정책

본 논문에서 신뢰성을 위한 결함 포용 정책은 결함 예방 정책과 결함 복구 정책으로 나누어진다. 결함 예방 정책에서 각 네트워크에 속하는 데스크톱 PC들은 상호 자원들의 정보 리스트를 공유하고 자신이 속한 네트워크의 PC들 중 가장 성능이 좋은 PC에서 정보 리스트를 보낸다. 결함 복구 정책에서 네트워크를 이탈한 PC들의 작업을 현재 네트워크에서 가장 성능이 좋은 PC의 정보 리스트가 최우선, 차우선 자원 매핑 정보를 확인하여 작업을 계속 유지시켜 주도록 한다. 이 때 체크포인트(checkpoint) 기법을 사용하여 진행 중인 작업의 복구 시점까지 작업을 복구시킨다.

3.3 적응적 스케줄링 기법

작업의 유형은 작업에 필요한 자원의 종류와 비율에 따라 계산 집약적 작업과 데이터 집약적 작업으로 분류하였다. 자원의 종류는 3가지 (cpu, memory, storage)로 정하였으며, 각 컴퓨팅 자원들 간의 네트워크의 대역폭(bandwidth)과 지연시간(latency)은 동일하다고 가정하였다.

본 논문에서 제안한 기법은 PROMETHE[15] 방법을 이용하였다. criteria를 각 자원들로 설정하였으며, <표 2>와 같이 자원들 간의 관계에 가중치(Weight)를 부여하였고, 각 자원들은 그룹(Group)으로 구성하여 자원 내의 정보를 관리하였다. 하나의 VM(virtual machine)의 총 가중치(W_{total})는 식 (1)과 같이 각 자원들 간의 관계의 합이며, 범위는 0과 1사이의 값으로 정해진다. 그룹 내의 각 자원에 대한 이용 가능 수치($R_{i_{AVAIL}}$)는 식 (2)와 같이 각 자원의 최대 사용 가능량($R_{i_{MAX}}$)과 현재 사용량($R_{i_{USAGE}}$)을 비교하여 나타내었다. 이를 바탕으로 그룹 내의 각 자원들의 순위를 정하였다.

<표 2> 관련 자원들의 그룹(G_i)과 가중치(W_i)

자원	R_1 (cpu)	R_2 (memory)	R_3 (storage)
자원 그룹 내 정보	G_1	G_2	G_3
자원들 간의 관계	W_1	W_2	W_3

$$W_{total} = \sum_{j=1}^{k} W_j \ (0 \le W_{total} \le 1) \quad \cdots\cdots\cdots \quad (1)$$

$$R_{i_{AVAIL}} = \frac{R_{i_{USAGE}}}{R_{i_{MAX}}} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (2)$$

본 논문에서 제안한 신뢰성 기반 적응적 스케줄링 기법은 크게 3부분으로 나눌 수 있다. 첫째, 사용자가 작업을 요청하였을 때 해당 작업에 맞는 자원을 찾아주는 자원 선택 부분이다. 둘째, 실제로 서비스를 수행하기 위해 해당 작업과 자원을 매핑 시켜주는 부분이다, 셋째, 작업 수행 중 결함이 발생하였을 때 문제를 해결시켜 주는 부분이다.

3.3.1 자원 선택

본 논문에서 사용한 자원은 단일 자원이 아닌 다중 자원(cpu, memory, storage)을 기반으로 했기에 Multicriteria Problems[16]으로 설정하여 <표 3>과 같이 그룹(G_i)과 VM(X_i) 간 평가 테이블(Evaluation table)을 구성하였다. 이 평가 테이블을 참조하여 식 (3)에서 가장 적합한 자원을 선택하였다. X는 선택 가능한 VM들의 유한 집합들{X_1, X_2, ..., X_n}이며, G는 평가 기준이 되는 자원 그룹 내 정보들의 집합{$G_1(.)$, $G_2(.)$, $G_3(.)$}이다.

<표 3> 그룹-VM 평가 테이블

X	$G_1(.)$	$G_2(.)$	$G_3(.)$
X_1	$G_1(X_1)$	$G_2(X_1)$	$G_3(X_1)$
X_2	$G_1(X_2)$	$G_2(X_2)$	$G_3(X_2)$
...	...	...	...
X_n	$G_1(X_n)$	$G_2(X_n)$	$G_3(X_n)$

$$\max\{G_1(x), G_2(x), G_3(x) | x \in X\} \quad \cdots\cdots\cdots\cdots (3)$$

자원 관리자(Resource Manager)가 클라우드 컴퓨팅 내의 자원 정보를 리스트 형태로 관리하고 있어서, 사용자가 클라우드 환경에 작업을 요청하면 해당 작업을 분할한 후 작업을 수행할 수 있는 가용 자원 리스트에서 분할된 작업을 수행할 수 있는 가용 자원들을 선택한다. 이 때, 식 (3)을 이용하여 가용 자원들 중 수치가 가장 높은 값을 가지는 자원이 우선 배정된다.

3.3.2 작업과 자원의 매핑

현재 수행 중인 작업 정보는 각각의 컴퓨팅 자원이 정보 리스트로 관리하고 있으며, 작업에 따른 자원 정보 리스트에서 우선 순위를 매겨 컴퓨팅 자원을 매핑 시켜 주도록 하였다.

이를 위해 자원을 선택 시 현재 가용한 자원의 이용률 측면을 고려하여 작업을 매핑시키도록 하기 위해 개별 자원 이용률을 식 (4)를 이용하여 측정한다. 이 때, 각 자원들의 평가 기준은 사용자가 제시한 비용이다. 개별 자원에 대한 사용 요금을 바탕으로 자원을 선택한다.

$$\max\{c(a), m(a), s(a) | a \in A\} \quad \cdots\cdots\cdots\cdots (4)$$

A : 가능한 경우들 (a_1, a_2, ... ,a_n)의 유한 집합

c, m, s : 각각 cpu, mem, storage 평가 기준의 집합

작업과 자원의 매핑 부분과 노드 선택 부분은 식 (5)와 같이 decision maker[16]를 활용하였다. $\{G_i(.), P_i(a,b)\}$을 $G_i(.)$에 대한 일반화된 기준(generalized criterion)으로 설정하여, 사용자가 요구한 비용 내에서 가장 성능이 좋은 자원에게 우선 작업을 할당시켰다.

$$P_i(a,b) = F_i[d_i(a,b)] \ \forall a,b \in A \quad \cdots\cdots\cdots\cdots (5)$$
$$d_i(a,b) = g_i(a) - g_i(b), 0 \le P_i(a,b) \le a$$

작업에 따른 자원 우선 순위 선택 알고리즘은 <그림 4>와 같다.

```
1: do order-of-priority algorithm (job, resource)
2:   for each G_i ∈ Resource(cpu, mem, str)
3:     W_total ← Σ_{j=1}^{k} W_{cpu_j} + Σ_{j=1}^{k} W_{mem_j} + Σ_{j=1}^{k} W_{str_j}
4:   end for
5:   for each R_i ∈ VM
6:     R_{i_AVAIL} ← R_{i_USAGE} / R_{i_MAX}
7:   end for
8:   align ascending order G(R_{i_AVAIL})
```

```
Si ← call getServerList()
for each Ji ∈ requestJobList(Si), Ti ∈ taskList(Ji)
    Ji split into Ti(Ji)
end for
for each Ti ∈ taskList(Ji), Ri ∈ resourceList(Gi)
    matching Pi(Ti, Ri)
end for
end do
```

<그림 4> 우선 순위 선택 알고리즘

3.3.3 결함 포용 정책

클라우드 컴퓨팅 환경에서 작업 수행시 사용자의 QoS(Quality of Service)를 보장하기 위해서 자원 제공자의 자원 정보 값($R_{i_{AVAIL}}$)에 대한 결함 발생 비율(f)과 신뢰도(γ)를 기반으로 작업의 복제 개수(p)를 정하고 분배 비율을 식 (6)으로 정의하였다. 작업의 최대 복제 개수를 3으로 설정하였고, 이는 복제 개수가 3을 초과할 경우 복제로 인한 오버헤드가 작업 수행 시간에 미치는 영향이 커져서 성능이 많이 떨어지기 때문이다.

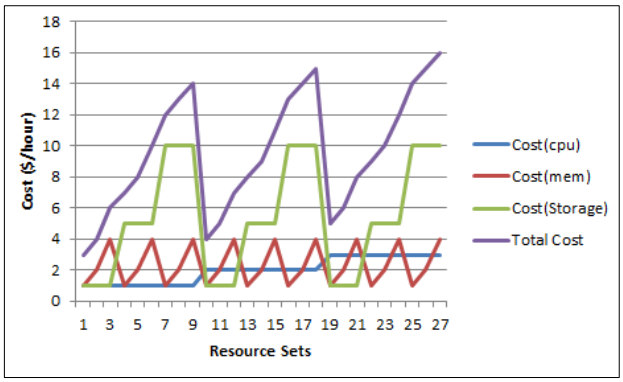
$$\gamma_i = \frac{\gamma_{i-1}}{(1-f)} \quad \begin{cases} \gamma > 0.9 (p=1) \\ \gamma \le 0.2 (p=3) \\ 0.2 < \gamma \le 0.9 (p=2) \end{cases} \quad \cdots\cdots\cdots\cdots (6)$$

$$(\gamma_0 = R_{i_{AVAIL}})$$

4. 성능 평가

성능 평가를 위해 VMware 7.0.1 기반 Ubuntu 10.10에서 CloudSim[17]을 사용하였다. CloudSim에서는 물리적 컴퓨팅 서버를 Host로 표현하며, 클라우드 환경을 구성하기 위해 데이터 센터(Data Center)는 하나 이상의 Host들로 이루어져 있다. 사용자는 데이터 센터의 구성 여부와는 상관없이 가상 장치(VM: Virtual Machine)을 이용하여 서비스를 사용하며, 실제 작업을 수행하는 역할은 Cloudlet(cloud task unit)이 담당한다.

실험 시스템 구성 환경은 CloudSim 시뮬레이션에서 단일 클라우드 시스템 환경으로 10개의 Desktop PC들로 구성되고 각 PC들의 자원(CPU, mem, storage)은 (1, 1, 10)부터 (3, 4, 100)으로 27가지 경우로 구분하였으며, 동일한 네트워크 집단으로 구성되어 있고, 안전한 네트워크 상태(자원 결함 발생 외 네트워크 결함을 발생하지 않음)라고 가정하였다. 각 Host 별로 VM은 최대 3개까지 가능하도록 설정하였다.



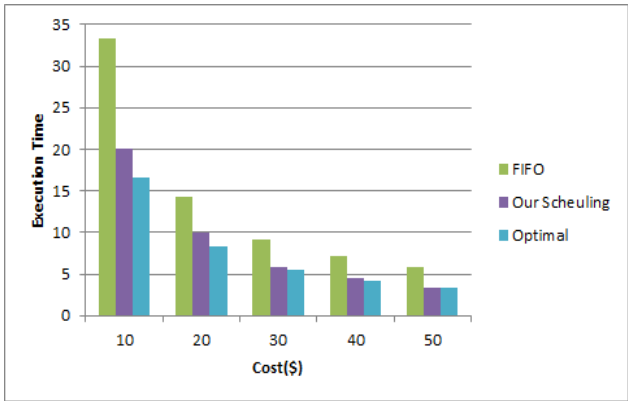
<그림 5> 자원 집합에 따른 요금

<그림 5>는 각 자원 집합들에 대한 각 자원의 개별 비용과 총 비용을 나타낸 것이다. CPU는 기준 시간당 1G 사용 시 $1, memory는 기준 시간당 1G 사용 시 $1, storage는 기준 시간 10G 사용 $1로 정하였으며 bandwidth는 동일하게 정하였다. 예를 들어 기준 시간당 사용 자원이 (CPU, mem, storage) = (1, 1, 10) 일 경우 총 자원 소비 요금은 $3이고, 자원 당 소비 요금은 1+1+1/3 = 1 로 $1가 된다.

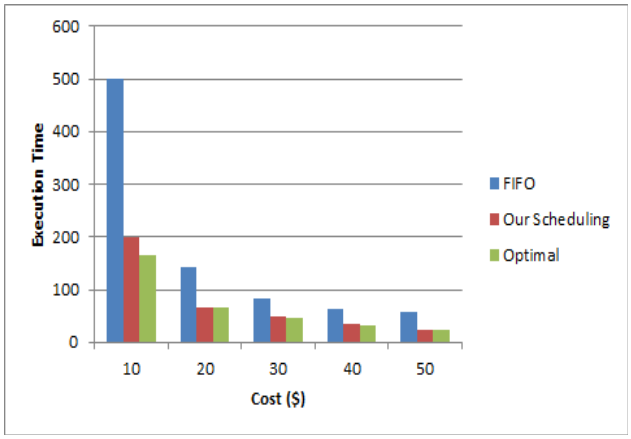
본 논문에서 실험한 내용은 크게 2가지로 나누어 볼 수 있다. <그림 6>은 사용자 요금 설정에 따른 계산 집약적 작업 처리 시간을 나타내며, <그림 7>은 사용자 요금 설정에 따른 데이터 집약적 작업 처리 시간을 나타낸다. 두 실험 모두 사용자 요금 정책에 따른 우선 순위 자원 선택-매칭 알고리즘을 활용한 실험으로 사용자 요금이 작게 설정되었을 때 큰 효과를 보였다. 또한, <표 4>에서 보듯이 실험 결과에서는 계산 집약적 작업보다 데이터 집약적 작업이 실행 시간 비율이 더 많이 줄어들었으며, 이는 클라우드 컴퓨팅 환경에서는 데이터 집약적 작업이 더 적합하다는 것을 증명한다.

<표 4> Improvement Ratio

cost	10	20	30	40	50
IR(com)	1.67	1.43	1.55	1.57	1.76
IR(data)	2.5	2.14	1.67	1.81	2.35



<그림 6> 사용자 요금 설정에 따른 계산 집약적 작업 처리 시간



<그림 7> 사용자 요금 설정에 따른 데이터 집약적 작업 처리 시간

실험 결과를 통해 본 논문에서 제안한 알고리즘이 일정한 사용자 요금 변화에 상관없이 신뢰성 있는 자원 매칭 알고리즘이라는 것을 증명하였다.

5. 결 론

대규모 데이터를 처리하는 클라우드 컴퓨팅은 다중 사용자 접속 환경 기반이어서 시스템의 신뢰성이 중요한 요소이다. 본 논문에서는 클라우드 시스템 환경 내부의 자원 변화에 대처할 수 있는 결함 포용적인 신뢰성 기반 적응적 스케줄링 기법을 제안하고 실험을 통한 타당성을 증명하였다.

본 논문에서 제안한 신뢰성 기반 적응적 스케줄링 기법은 클라우드 컴퓨팅 환경뿐만 아니라 그리드 컴퓨팅 환경에서도 사용이 가능하며, 시스템 상황에 맞춰 최적의 사용 환경을 제공하는 것을 목표로 하고 있다. 따라서, 추후 시스템 내부 자원 변화 뿐만 아니라 시스템 환경 자체 변화에 영향을 받지 아니하도록 설계하여, 그리드 환경에 이식이 가능하도록 연구할 것이다. 그리고 자원을 제공하는 컴퓨팅 환경도 모바일 자원도 포함하여 연구를 진행할 것이다.

또한, 본 논문에서는 타당성을 검증하기 위해 CloudSim 시뮬레이션 환경에서 실험을 하였지만, 추후 실제 클라우드 컴퓨팅 서비스를 제공하는 사이트에서 실험을 통해 시뮬레이션 환경과 실제 클라우드 컴퓨팅 환경과의 차이점을 비교 분석할 것이다.

참 고 문 헌

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zahariam(2009). Above the clouds: A Berkeley view of Cloud computing, || Technical report UCB/EECS-2009-28, Electrical Engineering and Computer Sciences, University of California at Berkeley, USA, February 2009.

[2] Rajkumar Buyya., et al(2008). "Market-oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications.

[3] M. Miller(2008). Cloud Computing: Web-based Applications that Change the Way You Work and Collaborate Online, QUE.

[4] I.Foster, Y.Zhao, I.Raicu, S.Lu(2008). "Cloud computing and Grid computing 360-degree compared", Grid Computing Environments Workshop.

[5] Foster, I. (2008). 'Cloud, Grid, what's in a name?' - "http://ianfoster.typepad.com/blog/2008/08/cloud-grid-what.html"

[6] Amazon Elastic Compute Cloud(EC2), "http://www.amazon.com/ec2/"
[7] Windows Azure, "http://www.microsoft.com/windowsazure/"
[8] Using the Parallel Virtual File System "http://www.parl.clemson.edu/pvfs"
[9] Jim Blythe, Sonal Jain, Ewa Deelman, Yolanda Gil, Karan Vahi, Anirban Mandal, and Ken Kennedy(2005). Task Scheduling Strategies for Workflowbased Applications in Grids. In International Symposium on Cluster Computing and the Grid (CCGrid'05), pp. 759 - 767.
[10] Wei Guo, Weiqiang Sun, Weisheng Hu, and Yaohui Jin(2006). Resource Allocation Strategies for Data-Intensive Workflow-Based Applications in Optical Grids. In 10th IEEE Singapore International Conference on Communication systems (IEEE ICCS 2006), pp. 1 - 5, October 2006.
[11] Anirban Mandal, Ken Kennedy, Charles Koelbel, Gabriel Marin, John Mellor-Crummey, Bo Liu, and Lennart Johnsson(2005). Scheduling strategies for mapping application workflows onto the grid. In 14th IEEE International Symposium on High Performance Distributed Computing (HPDC'05), pp. 125 - 134, Washington, DC, USA, IEEE Computer Society.
[12] Lavanya Ramakrishnan, Daniel Nurmi, Anirban Mandal, Charles K., Dennis G., T.M Huang, Yang-Seok Kee, Graziano O., Kiran T., Rich W., Asim Y., and Dmitri Zagorodnov(2009). VGrADS: Enabling e-Science Workflows on Grids and Clouds with Fault Tolerance. In International Conference for High Performance Computing, Networking, Storage and Analysis (SC09), November 2009.
[13] Pinar Senkul and Ismail H. Toroslu(2005). An architecture for workflow scheduling under resource allocation constraints. Information Systems, 30(5):399 - 422.
[14] Zhijiao Xiao, Huiyou Chang, and Yang Yi(2007). Optimization of Workflow Resources Allocation with Cost Constraint, pp. 647 - 656. Springer Berlin / Heidelberg, August 2007.
[15] B. Mareschal(1987). "Aide a la Decision Multicritere: Developpements Recents des Methodes PROMETHEE," Cahiers du Centre d'Etudes en Recherche Operationelle, pp. 175 - 241.
[16] J.-P. Brans, J. Figueira, S. Greco, and M. Ehrgott(2005). Multiple Criteria Decision Analysis:State of the Art Surveys. Springer New York.
[17] Rodrigo N. Calheiros, R. Ranjan, César A. F. De Rose and R. Buyya(2009). CloudSim : A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services, ICPP2009.

조 인 석

2006 고려대학교 컴퓨터교육과(이학사)
2008 고려대학교 컴퓨터교육과(이학석사)
2009~현재 고려대학교 컴퓨터교육과 박사과정
관심분야: 그리드 컴퓨팅, 클라우드 컴퓨팅, 분산 시스템, 결함포용 시스템, 컴퓨터교육
E-Mail: violet@korea.ac.kr

유 헌 창

1989 고려대학교 이과대학 전산과학과(이학사)
1991 고려대학교 대학원 전산과학과(이학석사)
1994 고려대학교 대학원 전산과학과 (이학박사)
1998~현재 고려대학교 컴퓨터교육과 교수
2006~2010 한국컴퓨터교육학회 부회장
관심분야: 그리드 컴퓨팅, 분산시스템, 결함포용시스템, u-learning
E-Mail: yuhc@korea.ac.kr