

이주 효율성 향상을 위한 퍼지로지직 기반 우선순위 이주 모델

박민오 · 김재권 · 최정석 · 이종식*

Fuzzy logic-based Priority Live Migration Model for Efficiency

Min-Oh Park · Jae-Kwon Kim · Jeong-seok Choi · Jong-Sik Lee*

ABSTRACT

If the cloud computing environment is not sufficiently provide the required resources due to the number of virtual server to process the request, may cause a problem that the load applied to the specific server. Migration administrator receive the resources of each physical server for improving the efficiency of the virtual server that exists in the physical servers, and determines the migration destination based on the simulation results. But, there is more overhead predicting the future resource consumption of all the physical server to decide the migration destination through the simulation process in large and complex cloud computing environments. To solve this problem, we propose an improved prediction method with the simulation-based approach. The proposed method is a fuzzy-logic based priority model for VM migration. We design a proposed model with the DEVS formalism. And we also measure and compare a performance and migration count with existing simulation-based migration method. FPLM shows high utilization.

Key words : Cloud Computing, Live Migration, Server Virtualization, Distributed Processing

요약

클라우드 컴퓨팅 환경은 다수의 가상서버 처리요청으로 인해 필요 자원을 충분히 제공하지 못한 경우, 특정 서버에 부하가 걸리는 문제가 발생할 수 있다. 이주관리자는 물리서버 내에 존재하는 가상서버들의 이주 효율성 향상을 위해 각 물리서버의 자원 정보를 모니터링 시스템으로부터 전달받고, 시뮬레이션 결과 값을 토대로 이주 목적지 물리서버를 결정한다. 하지만 모든 물리서버의 미래 자원 사용량을 예측하여 시뮬레이션 과정을 거쳐 이주 목적지 물리서버를 결정하는 것은 소수의 서버 네트워크 컴퓨팅 환경보다 거대하고 복잡한 클라우드 컴퓨팅 환경에서는 오버헤드가 크다. 본 논문에서는 퍼지로지직 기반 이주 결정 모델(FPLM)을 제안하고 DEVS 형식론을 적용하여 이주 발생 횟수 및 성능을 비교 측정하였다. FPLM은 이주 발생 횟수 및 이주 목적지 결정 오버헤드를 감소시킴으로써 이주 발생으로 인한 물리서버 자원 사용 효율성을 증가시킨다.

주요어 : 클라우드 컴퓨팅, 자원이주, 서버 가상화, 분산처리

1. 서론

서버 가상화(Server Virtualization)는 물리서버 한 대

속에 여러 대의 가상서버를 구성하는 기술이다. 가상서버는 물리적인 단일 서버 내에서 하이퍼바이저 기법을 통해 일반 프로그램처럼 호스트 운영 체제에서 실행되는 가상 머신에서 구동 중인 서버를 지칭한다. 클라우드 컴퓨팅 환경에서는 서버 가상화를 이용해 수많은 사용자의 처리요청과 애플리케이션의 동작을 실시간으로 처리한다^[7]. 하지만 물리서버가 가진 처리능력보다 많은 다수의 가상서버 처리요청으로 인해 필요 자원을 충분히 제공하지 못한 경우, 특정 서버에 부하가 걸리는 문제가 발생한다. 이주관리자는 물리서버가 더 이상 QoS(Quality of Service)를 보장할 수 없는 상황이 발생하기 전에 물리서버 내에

* 이 논문은 방위사업청과 국방과학연구소의 지원으로 수행되었습니다(UD140022PD).

Received: 30 August 2015, **Revised:** 21 October 2015,
Accepted: 29 October 2015

*Corresponding Author: Jong-Sik Lee

E-mail: jslee@inha.ac.kr

Department of Computer and Information Engineering,
Inha University

존재하는 가상머신을 상대적으로 부하가 적은 다른 물리 서버로 이주시키는 방식(VM Live Migration)^[3]을 사용한다. 하지만 모든 물리서버의 미래 자원 사용량을 예측하여 시뮬레이션 과정을 거쳐 이주 목적지 물리서버를 결정하는 것은 소수의 서버 네트워크 컴퓨팅 환경보다 거대하고 복잡한 클라우드 컴퓨팅 환경에서는 오버헤드가 크다. 이와 관련하여 본 논문에서는 가상머신 이주 효율성 향상을 위해 퍼지로지 기반 이주 결정 모델(FPLM: Fuzzy logic based Priority Live migration Model)을 제안한다. FPLM은 CPU, 메모리, 네트워크 지연시간과 같은 물리 서버의 자원을 퍼지로직을 적용하여 효율적인 이주 목적지 결정이 가능하다.

본 논문의 구성은 다음과 같다. 2장에서 기존 연구에 대해 소개하고, 3장에서는 퍼지로지 기반 우선순위 이주 모델에 대해서 자세히 설명한다. 그리고 4장에서 DEVS (Discrete Event System Specification)형식론을 적용하여 퍼지로지 기반 우선순위 이주 결정 모델의 성능을 비교 측정하는 실험에 대한 설명 및 결과를 분석하며, 5장에서 결론을 맺는다.

2. 관련 연구

2.1 VM Live Migration

기존의 가상머신 이주 방식은 물리서버 내 가상서버에서 수행 중인 서비스 관련 애플리케이션의 모든 인스턴스를 중지한 후, 목적지 물리서버로 인스턴스 복제를 수행하기 때문에 서비스 중지로 인해 발생하는 비용이 존재한다. 가상머신 실시간 이주 기법(VM Live Migration)은 인스턴스 복제시 발생하는 문제점을 해결하였다. 가상서버에서 실행 중인 서비스가 물리서버 자원사용량의 임계치를 넘어서는 상황이 발생할 경우, 상대적으로 사용량이 적은 다른 물리서버로 서비스 중지 없이 스토리지 복제를 수행한다. 여기서 스토리지는 디스크 저장소가 아닌, 가상머신을 이주하기 위해 사용되는 메모리 저장 공간을 지칭한다.

이와 관련하여 T.wood 등(2007)은 이주 대상 가상서버 및 목적지 물리서버 선정에 위한 자원 관리 기법에 대한 연구를 수행하였다. 특정 물리서버 자원 사용량이 증가하는 상황을 실시간 모니터링 기법과 자원 프로비저닝 기법을 이용해 탐지하고, 자원 사용량이 가장 큰 물리서버 내 존재하는 가상머신들 중에 자원 점유율이 가장 낮은 가상머신을 이주 대상으로 선정한다. 이 때 이주 대상 가상머신 및 목적지 물리서버 선정에 위해 각 가상머신

및 물리서버 자원 이용량을 아래의 수식으로 표현한다.

$$Volume = \frac{1}{1-cpu} \times \frac{1}{1-\neq t} \times \frac{1}{1-mem} \quad (1)$$

Eq. (1). The volume of VM in servers

Eq. (1)은 모니터링 시스템이 서버부하를 모니터링하기 위한 수식으로 사용할 수 있다. 부피(Volume) 값은 해당 서버의 CPU, 네트워크, 메모리 사용량을 모니터링하여 결정한다. 하지만, 부피값을 이용한 부하측정은 성능이 동일한 물리서버간의 가상머신 이주를 고려할 때 적합한 측정법이다.

서로 상이한 성능을 지닌 서버간 가상머신 이주가 이루어질 경우, 서버 자원 가용성 및 성능 측정 방법으로 부피값을 이용한 수식 외에 벡터 연산을 이용한 Vector Dot^[10] 방법이 더 적합하다. Vector Dot 방법은 물리서버의 CPU, 메모리, I/O 사용량을 각각 X, Y, Z 축으로 정의하고, 기하학적 접근법을 활용, 상대적인 성능을 지닌 물리서버간의 가상머신 배치 및 이주를 위한 상대적 부하량을 측정하는데 유용한 알고리즘이다.

2.2 시뮬레이션 기반 이주 알고리즘

최현식 등(2011)은 CPU 및 네트워크 대역폭의 주기성을 활용한 예측 방법을 통해 SLA(Service Level Agreement) 위반을 방지하는 연구^[2]를 수행하였다. 자기 회귀 분석 예측 기법으로 CPU 사용 요구량 정보를 참조하여 이주 대상 가상머신과 목적지 후보 물리서버간의 스토리지 복제 작업을 시뮬레이션 한다. 시뮬레이션 결과, 추가적인 스토리지 복제가 발생할 가능성이 가장 낮은 물리서버를 목적지 서버로 선정한다.

$$X_i^n = \phi_0 + \phi_1 X_i^{n-1} + \phi_2 X_i^{n-2} + \dots + \phi_\rho X_i^{n-\rho} + \epsilon_n \quad (2)$$

X_i^0, \dots, X_i^{n-1} : CPU utilization at time t

ϕ_0, \dots, ϕ_ρ : coefficient of X_i^n

ϵ_n : error at time t

Eq. (2). Auto regression analysis based prediction model

해당 방식에서 자기 회귀 분석의 성능은 시뮬레이션의 정확성을 결정하는 중요한 요소이다. 이 때 X_i^n 차수가 5보다 작을 경우, 예측 결과 값이 부정확하여 신뢰성이 떨어

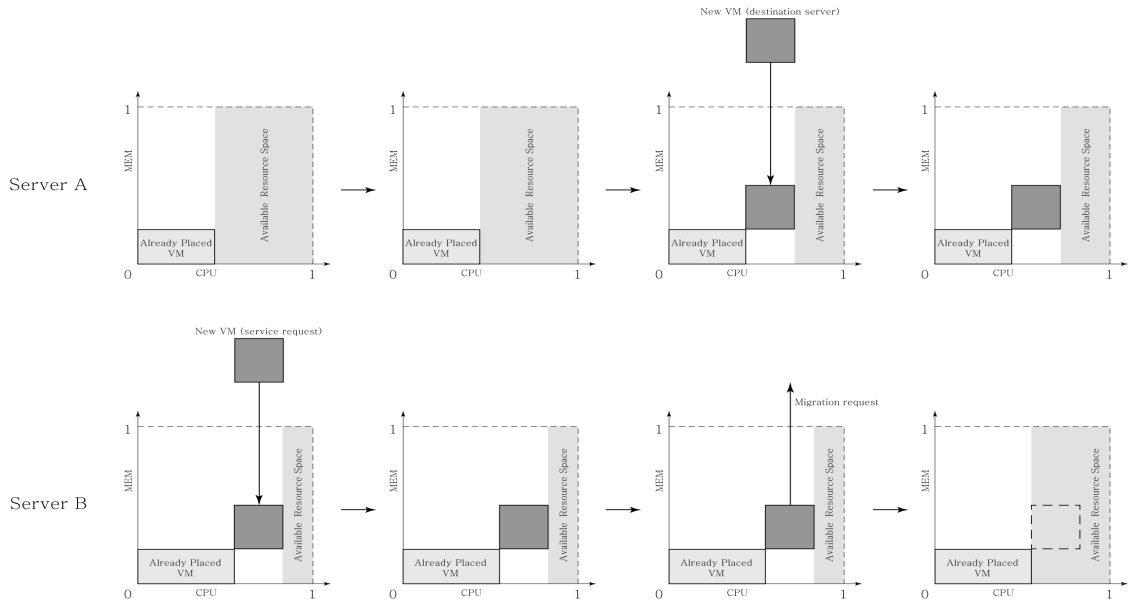


Fig. 1. Volume based VM Live Migration Concept

어질 수 있고 예측을 위한 Eq. (2)의 X_i^n 차수가 5 이상이 될 때, 신뢰 가능한 결과 값이 도출되나 오버헤드가 클 수 있다는 이슈가 존재 한다. 이주관리자가 관리하는 물리서버의 양과 모니터링 시스템의 성능에 의존하여 적절한 차수를 선택하는 것은 자원 사용량 예측 정확성을 결

정짓는 중요한 사항^[2]이다. 따라서, 데이터센터 내에 존재하는 모든 물리서버의 미래 자원 사용량을 예측 및 시뮬레이션 과정을 거쳐 이주 목적지 물리서버를 결정하는 것은 거대하고 복잡한 클라우드 컴퓨팅 환경에서는 효율적인 실시간 이주를 기대할 수 없다.

2.3 Storage Copy Scheme

클라우드 컴퓨팅 환경에서 해당 부하를 동적으로 재배치시키는 이주 기법은 스토리지 복제와 가상머신의 이주의 순서에 따라 선 복제 모델(pre-copy)^[5]과 후 복제(post-copy)^[5] 모델로 분류된다. 물리서버 A에서 물리서버 B로 이주 요청에 의해 스토리지 복제가 먼저 이루어질 때, write 작업이 발생할 경우 두 스토리지간 동기화를 위해 데이터 블록을 재전송하는 과정이 필요하다. 원본 스토리지에 write 작업이 발생한 블록이 많을수록 데이터 블록의 수는 증가하며 결과적으로 이주 소요 시간에 많은 영향을 끼친다. 반면 스토리지보다 가상머신을 먼저 이주하는 경우 데이터가 없는 스토리지가 가상머신과 연동된다. 존재하지 않는 데이터의 read 작업 요청 시 물리서버 A에서 일시적인 스토리지 복제과정이 필요하므로 read 작업의 양이 이주 시간에 영향을 미친다^[1]. 따라서 서비스 중지 없이 이주를 실시할 경우 이주 기법의 종류에 관계없이 서비스 요청에 의해 발생하는 오버헤드가 존재하며 해당 요청이 빈번할 경우 QoS를 보장할 수 없다.

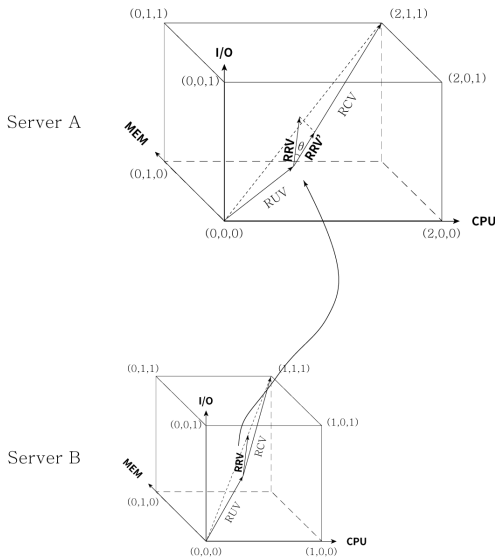


Fig. 2. Vector Dot based relative VM placement

2.4 Fuzzy logic

데이터센터 내에 존재하는 물리서버 컴퓨팅 자원은 서로 상이하며 수 천 대의 서버 컴퓨터가 가상화 기술을 통해 운영된다. 원활한 서비스 제공을 위해 데이터센터 이주 관리자는 이주 상황이 발생할 경우 다른 물리서버로 가상서버를 이주해야 한다. 어떤 물리서버의 컴퓨팅 자원을 할당할지 신속하게 결정하는 것은 SLA 보장을 위한 중요한 요소이다. 해당 요소는 데이터센터 내 가용 물리서버 컴퓨팅 자원의 탐색 속도에 따라 결정된다. 이 때 모든 물리서버에 존재하는 할당 가능한 컴퓨팅 자원량을 탐색할 경우 오버헤드가 크다는 문제점이 존재한다. 하지만 클러스터 단위의 탐색을 할 경우는 상대적으로 이주 요청에 적합하고 할당 가능한 컴퓨팅 자원량을 지닌 후보 목적지 물리서버를 결정하기 수월하다. 이 때 수식으로 정의하거나 표현하기 어려운 인간의 언어, 사고를 수학적으로 다루는 퍼지 이론을 적용하여 각 물리서버의 CPU 속도, 네트워크 대역폭, 메모리 크기에 따른 물리서버 자원을 어떤 클러스터에 편입시킬 것인지 결정한다. 물리서버의 성능을 결정하는 요소를 퍼지화를 통해 퍼지입력으로 변환하고, 제어규칙에 의거하여 퍼지추론과정을 거친 후, 최종적으로 역퍼지화를 통해 데이터센터 물리서버들의 상이성에 대한 정보를 관리한다.

3. 퍼지로직 기반 우선순위 이주 결정 모델

본 장에서는 가상머신 이주 상황이 발생할 경우, 오버헤드를 예측기반 시뮬레이션 방식보다 감소시키기 위한 FPLM에 대하여 제안한다. FPLM은 시뮬레이션 기반 이주 알고리즘^[2]의 한계점을 보완하고 데이터센터가 지니는 특성을 고려하여 이주 목적지 서버를 시뮬레이션 기반 이주 방식(Simulation-based Live Migration)^[2]보다 효율적으로 결정할 수 있다.

3.1 데이터센터를 구성하는 물리서버의 특성

대형 클라우드 서비스를 제공하는 데이터센터는 여러 해를 거쳐 운용되었기 때문에 서버 노후화, 기재고장 발견 등으로 인한 물리서버 교체가 빈번하다. 그래서 데이터센터는 다양한 성능과 특성을 가진 서버 컴퓨터로 구성되어있다. 데이터센터 내 존재하는 각 물리서버의 CPU 속도, 네트워크 대역폭, 메모리 크기가 교체시기별 혹은 기능별 그 성능이 매우 상이한 점은 서버 컴퓨팅 자원을 필요로 하는 클라우드 서비스가 사용자 요청 수준에 따라 해당 서비스를 수행하기 적합한 물리서버를 찾는 작업과

연관성이 있다. 또한 컴퓨팅 성능이 상대적으로 낮은 물리서버에 존재하는 가상머신이 해당 물리서버의 임계치 이상 요구하는 서비스의 컴퓨팅 자원 할당을 요청받았을 경우, 다른 물리서버로의 가상서버 이주는 불가피하다. 따라서 사용자 요청 수준에 적합한 가상서버 할당을 위해 데이터센터 내 물리서버의 특성 및 성능과 요청 서비스를 프로파일링하여 관리하면 사용자 요청 수준에 적합한 물리서버에 이주 작업을 요청시 빠르게 탐색할 수 있다.

프로파일링을 통한 서버 자원 점유율 관리는 전문 모니터링 시스템에 의해 이루어진다. 물리서버 특성을 기준으로 프로파일링을 할 경우 물리서버는 해당 물리서버와 유사한 특성을 지닌 집단끼리 묶어서 관리가 이루어질 수 있다. 유사성을 기준으로 프로파일링을 수행하기 전과 비교했을 때, 프로파일링을 통한 관리 방법 이전에는 이주 상황 발생시 이주 목적지 물리서버 선정을 위해 전문 모니터링 시스템에 의해서 모니터링이 이루어지고, 이주 관리자가 해당 모니터링 시스템을 통해 클라우드 시스템의 정보를 가져다가 어떤 호스트로 요청받은 가상서버 이주를 수행해야하는지 판단한다. 이 때 전체 클라우드 시스템 관점에서 볼 경우, 수백~수천 대에 이르는 가상화된 호스트들을 모니터링하는 것이 아닌, 요청받은 가상머신 크기와 유사한 가용성을 지닌 물리서버가 프로파일링된 목적지 물리서버를 탐색한다. 해당 방법을 적용하면 유사성을 지닌 물리서버에서 목적지 물리서버를 탐색하면 되기 때문에 모니터링 시스템이 목적지 물리서버 결정을 위해 탐색하는 물리서버의 개수는 감소한다.

3.2 이주 대상 가상서버의 목적지 물리서버 결정 방법

시뮬레이션 기반 이주방식은 이주 대상 가상머신과 목적지 물리서버를 선정하기 위해 물리서버들의 CPU 사용률을 시뮬레이션하여 이주 정책을 도출한다. 각 물리서버에 존재하는 센서는 시뮬레이션을 위한 현재 CPU 사용률을 모니터링 시스템에게 제공한다. 해당 정보를 토대로 이주상황이 언제 발생할지 예측하며, 예측된 값에 근거하여 이주 정책을 결정한다. FPLM은 시뮬레이션 기반 이주 방식을 보완하기 위해 퍼지로직을 도입한다. 퍼지로직은 데이터센터 내 상이한 물리서버 특징을 비슷한 속성끼리 묶어주는 역할을 한다. 이 때 물리서버들의 어떠한 속성을 퍼지로직에 적용할 것인지 결정하느냐에 따라 이주의 효율성이 결정된다.

이주 정책 관점에서 살펴보면 가상서버의 이주 소요시간은 두 물리서버간 스토리지 복제 속도에 비례한다. 따

Algorithm 1. ClusterClassifier

```

1:  $S \leftarrow$  VM operating Physical Server in Cloud System
2:  $CPU \leftarrow$   $S$ 's CPU speed
3:  $MEM \leftarrow$   $S$ 's memory size
4:  $NET \leftarrow$   $S$ 's network bandwidth
5: while All  $S$  are not classified do
6:   if  $fuzzy(CPU, MEM, NET) = VPR$  then
7:     assign  $S$  to  $VPRC$ 
8:   else if  $fuzzy(CPU, MEM, NET) = GD$  then
9:     assign  $S$  to  $GDC$ 
10:  else if  $fuzzy(CPU, MEM, NET) = MD$  then
11:    assign  $S$  to  $MDC$ 
12:  else if  $fuzzy(CPU, MEM, NET) = PR$  then
13:    assign  $S$  to  $PRC$ 
14:  else  $fuzzy(CPU, MEM, NET) = VPR$  then
15:    assign  $S$  to  $VPRC$ 
16: end while

```

라서 스토리지 복제 수행에 가장 큰 영향을 미치는 값은 물리서버의 CPU 속도, 네트워크 대역폭, 메모리 크기이다. Algorithm 1은 이주 목적지 서버들의 클러스터화를 위한 알고리즘이다. 퍼지로직을 적용하여 얻은 결과값으로 데이터센터 내 물리서버들을 클러스터로 분류하고, 물리서버가 속하는 클러스터 내에 이주 요청이 발생할 경우, 동일 클러스터 안에 할당 가능한 컴퓨팅 자원을 보유한 물리서버를 탐색하여 이주를 수행한다.

Algorithm 2. GetDestinationServerPlacement (VM)

```

1:  $M \leftarrow$  migration requested Physical Server
2:  $C \leftarrow$  cluster which is belong to  $M$ 
3:  $V \leftarrow$  volume of VM that operated by  $M$ 
4:  $DestinationServerInfo \leftarrow \phi$ 
5: Introduce a new Physical Server;
6: while Destination Server of VM is not placed do
7:   for all Physical Server in  $C$  do
8:      $available\_resource\_space \leftarrow$  remain capacity in  $C$ 
9:     if ( $available\_resource\_space \geq V$ )
10:      return  $DestinationServerInfo$ 
11:   else
12:     /* Choose the neighbor cluster of  $C$  */
13:      $C \leftarrow neighbor(C)$ ;
14:      $V \leftarrow VectorDot(V)$ ;
15:      $GetDestinationServerPlacement(VM)$ ;
16: end while

```

Algorithm 2는 목적지 물리서버를 선택하는 알고리즘이다. 이주 요청시, 모니터링 시스템은 동일 클러스터 내

가용 물리서버 정보를 이주관리자에게 전달한다. 동일 클러스터 내 가용서버가 없을 경우, Vector Dot 기법을 이용하여 상대적인 크기값을 계산 후, 다른 클러스터 내 가용서버를 탐색한다. 이 때, 요청 클러스터를 참조하여 다음으로 밀접한 클러스터를 탐색한다. 요청한 가용 서버가 있으면, 이주관리자에게 클러스터 정보 전달하게 되고, 아닐 경우는 재탐색을 재귀적으로 수행하게 된다.

3.3 퍼지로지 기반 우선순위 결정 모델

데이터센터 내에 존재하는 상이한 각 물리서버의 CPU 속도, 네트워크 대역폭, 메모리 크기를 각각 입력 변수 X , Y , Z 로 정의하고 퍼지로지 출력 값을 물리서버 성능(P)으로 정의한다. CPU 속도는 각 물리서버가 소유한 CPU 클럭 속도 레벨을 나타낸다. 메모리는 스토리지 복제를 실행할 때 작업을 저장하기 위해 사용가능한 크기 레벨이며 네트워크 대역폭은 각 물리서버가 하나의 패킷을 다른 스토리지로 전송하는 대역폭 레벨을 나타낸다. 입, 출력 변수의 구성은 Table 1과 같다.

이주 목적지 물리서버들의 성능 출력 값을 추론하기 위한 방식은 min-max 합성법^[6]을 이용한다.

$$X = \{x|x \text{ is Server CPU speed in DC}\}$$

$$Y = \{y|y \text{ is Server Network bandwidth in DC}\}$$

$$Z = \{z|z \text{ is Server Memory size in DC}\}$$

$$\forall (x,y) \in X \times Y, \quad \forall (x,y) \in Y \times Z$$

$$\mu_{S \cdot R}(x,z) = \min_y [\max(\mu_R(x,y), \mu_S(y,z))] \quad (3)$$

Eq. (3). Min-max composition for server selection

Fuzzy Logic 입/출력 변수들의 멤버십 함수는 다음과 같다.

Table 1. Fuzzy Input/Output variables

I/O	variables	Fuzzy variables				
Input X	CPU speed	VSL	SL	MD	FA	VFA
Input Y	network bandwidth	VSL	SL	MD	FA	VFA
Input Z	Memory size	VSM	SM	MD	LR	VLR
Output P	Server Computing power	VPR	PR	MD	GD	VGD

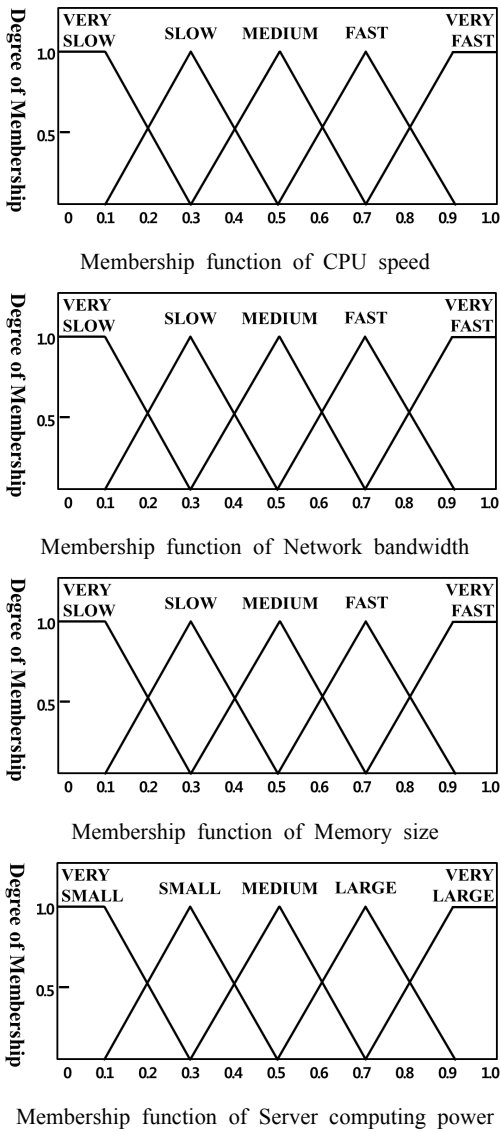


Fig. 3. Membership function of CPU speed, Network bandwidth, Memory size and Server computing power

각 서버의 CPU 속도, 네트워크 대역폭, 메모리 크기 정도를 나타내는 언어변수는 Table 2를 참고한다. 언어변수를 대표하는 값을 각각 0.1, 0.3, 0.5, 0.7, 0.9로 정의한다.

퍼지 추론 규칙에 따라 이주 목적지 물리서버 성능을 결정한다. 퍼지 추론 규칙은 Table 3과 같다.

퍼지 출력값의 역퍼지화 과정은 무게중심법을 사용한

Table 2. Fuzzy linguistic variables

linguistic variables	Fuzzy numbers
very slow or very Small	0.1
slow or small	0.3
medium	0.5
fast or large	0.7
very fast or very large	0.9

Table 3. Fuzzy Rule

I/O	INPUT			OUTPUT
	X	Y	Z	P
Rule 1	VSL	VSL	VSM	VPR
Rule 2			SM	PR
Rule 3			MD	PR
Rule 4			LR	MD
Rule 5			VLR	MD
Rule 6		SL	VSM	VPR
Rule 7			SM	PR
Rule 8			MD	MD
Rule 9			LR	MD
Rule 10			VLR	MD
...				
Rule 125	VFA	VFA	VLR	VGD

다. 무게중심법에 의한 역퍼지화 방법은 Eq. (4)와 같다.

$$p_0 = \frac{\sum_{i=0}^n \mu_s(p_i) \cdot p_i}{\sum_{i=0}^n \mu_s(p_i)} \quad (4)$$

$\mu_s(p_i)$: membership of defuzzification values p_i

p_i : defuzzification values

Eq. (4). Center of gravity method for server selection

FPLM은 퍼지로그를 통해 물리서버의 컴퓨팅 자원 수준을 판단한 후, 모니터링 시스템은 같은 속성을 갖는 물리서버를 여러 개 모아서 하나의 클러스터로 관리한다.

관리 도중 새로운 가상서버 요청 작업을 할당받으면 가용성이 가장 높은 물리서버를 선택하게 된다. 또한 실시간 이주 요청 발생 시 데이터센터 내 존재하는 물리서버들의 각 자원 상황을 탐색할 때, 이주 요청한 물리서버의 특성과 유사한 클러스터를 찾아 해당 클러스터 내에 존재하는 물리서버들의 성능을 평가하고 해당 서버의 자원사용량을 고려하여 목적지 물리서버를 선정한다.

4. 실험 및 결과 분석

본 장에서는 FPLM의 성능을 평가하기 위해 2가지 실험을 진행하여 이를 분석하였다. 첫째로 동일한 크기의 VM 요청을 받는 클라우드 가상화 환경을 구성한 후, 이주 횟수(Number of Live Migration times) 측정 실험을 진행하였다. 둘째로 제안 모델의 이주효율성을 검증하기 위해 이주 요청 처리율을 측정하여 비교하였다. 가상의 환경을 구성하고 시뮬레이션을 진행하여 결과를 분석한다.

4.1 이주 횟수 감소 효과 여부 검증

클라우드 시스템은 이주 발생시 가용 자원 할당을 통해 자원 관리의 탄력성을 가져야 한다. 이 장에서는 본 논문에서 제안하는 FPLM의 성능을 판단하는 중요한 요소 중 하나인 이주 횟수 측정을 통해 다른 비교 모델과의 차이를 입증하고자 한다.

4.1.1 실험 환경 구성 및 설계

FPLM의 성능을 평가하기 위해 DEVS 형식론^[9]을 적용하여 실험 테스트 환경을 모델링한다. 실험 시나리오를 구성하기 위해서 클라우드 환경에서의 가상머신 이주를

고려한 클라우드 시스템을 구성한다. VM Request는 VM 요청 작업을 생성하고 클라우드 시스템에 해당 작업을 보내는 모듈로서, 사용자 VM 요청 작업을 모델링 한 것이다. 요청 작업은 클라우드 시스템의 자원할당 프로비저닝 알고리즘에 따라 적절한 물리서버로 VM을 할당한다. 두번째로, Migration Manager는 클라우드 시스템 내에서 VM 이주 작업 요청을 받으면, FPLM, SMA, RRS 방식으로 물리서버에게 VM 이주 요청 작업을 할당하는 모듈이며, 작업을 할당하기 위해 Monitoring System의 정보를 이용한다. 이주 작업을 완료하게 되면 Transducer로 작업이 완료 되었다는 신호를 알리게 되며 이주 횟수, VM 요청 처리율, 평균 처리시간을 평가하게 된다. 각 모듈에서 VM 요청과 같은 메시지 전달은 Job Processing Message로, 계산이나 참조를 위해 상태를 전송하는 Status Transport Message로 통신하게 된다. 10개의 물리서버로 구성된 3개의 클러스터를 구성하였으며, 사용자는 100개의 VM을 요청한다. 시뮬레이션 기반 이주 방식의 모델링은 예측구간의 크기가 50ms인 환경에서의 자기 회귀 분석 예측 기법^[8]을 참고하여 모델링한다. 위와 같이 테스트 환경을 구성하였으며, 다음으로 클라우드 시스템에 가해지는 가상머신 요청 시나리오를 Table 4와 같이 구성하여 실험한다.

비교를 위한 실험 모델은 총 2가지이며, 첫 번째로는, 회귀분석 예측 기법을 통해 미래 자원 사용량을 측정하여 시뮬레이션한 값을 근거로 이주 목적지를 결정하는 시뮬레이션 기반 알고리즘(A Simulation-based Migration Algorithm : SMA)(Choi 외, 2011) 두 번째는, 시분할 시스템을 위해 설계된 선점형 스케줄링의 하나로서, 가상머신들 사이에 우선순위를 두지 않고, 순서대로 할당하는 방식의 라운드 로빈 스케줄링(Round Robin Scheduling : RRS)이다.

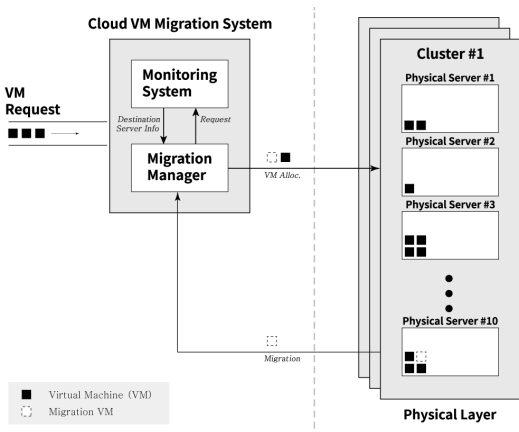


Fig. 4. Cloud VM Migration Times Test Bed

Table 4. VM Request Scenario

time	VM Request	Cumulative
0	10	10
100	10	20
200	10	30
300	10	40
400	10	50
500	10	60
600	10	70
700	10	80
800	10	90
900	10	100

가상머신 이주 테스트 환경에서 FPLM, SMA, RRS 방식에 따른 이주 발생 횟수(Number of Live Migration)를 측정한다. 이주 발생 횟수는 관측시간동안 발생하는 이주 요청의 총합으로 Eq. (5)와 같이 정의한다.

$$\text{Number of Live Migration} = \sum_{i=0}^n LM(\text{Server}_i)$$

$$LM(\text{Server}_i) = \sum_{t=0}^{\text{observation time}} \text{Live Migration}_t \quad (5)$$

n = Total number of Servers in Datacenter

t = Executed live migration at time t

Eq. (5). Number of Live Migration

4.1.2 결과 및 분석

4.1에서 설정한 테스트 환경을 가지고 지속적으로 사용자 VM 요청 작업이 이루어지는 가운데, FPLM 방식을 적용하면 이주 횟수 감소가 이루어지는지 판단하기 위해 DEVS 시뮬레이터를 실행시킨 후 관찰하였다.

Fig. 5는 물리서버에 전달되는 부하의 크기에 따른 이주 발생 횟수를 관찰한 결과이다. 311sec에 FPLM, SMA, RRS 모두 가상머신 이주가 발생하기 시작하여, FPLM은 약 376sec의 시간 간격으로, 이주가 발생함을 보였다. 결과 그래프를 살펴보면, 클라우드 시스템 환경에서 최초로 가상머신 이주가 발생하는 시간이 모두 동일한데, 이것은 VM request 모듈에서 동일한 크기의 VM을 보냈고, 동일한 VM 프로비저닝 알고리즘에 의한 자원할당을 했다고 할 수 있다. SMA와 RRS는 각각 322sec, 127sec 주기를

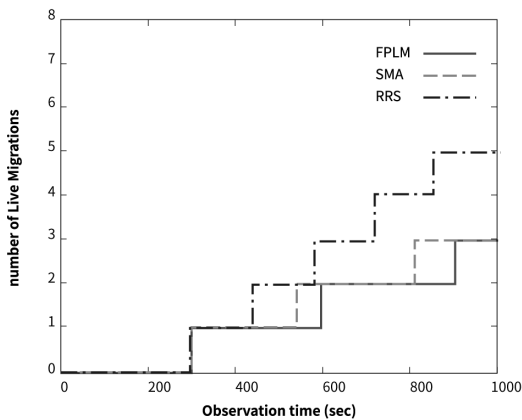


Fig. 5. Number of Live Migration times

보이는데, 동일한 클라우드 시스템 테스트 환경에서 동일한 간격으로 특정 크기의 VM을 요청하는 실험을 진행한 결과이기 때문에 시간 간격은 Migration Manager의 이주 목적지 선정 방법에 의한 차이로 인식할 수 있다. 총 이주 횟수는 FPLM과 SMA는 총 3회로 시뮬레이션 기반 이주 방식과 동일한 이주 횟수를 보이지만, 이는 실험을 위한 관측시간을 1000sec로 제한하였기 때문으로 보인다.

4.2 이주 효율성 검증

본 장에서는 제안하는 가상머신 이주 목적지 선정 기법의 이주 효율성을 검증하기 위해 FPLM, SMA, RRS의 이주관리자가 가상머신 이주 요청을 받는 시점 T_r , 가상머신 이주가 완료되는 시점 T_d 을 10회 측정하여 FPLM, SMA, RRS의 이주 효율성을 비교분석한다.

4.2.1 실험 환경 구성 및 설계

FPLM의 이주 효율성을 평가하기 위해 DEVS 형식론^[9]을 적용하여 실험 테스트 환경을 모델링한다. 실험 시나리오를 구성하기 위해서 클라우드 환경에서의 가상머신 이주를 고려한 클라우드 시스템을 구성한다. 4.2.1 실험에서는 새로운 VM 요청 없이 단일 물리서버에서 임의의 가상머신 이주요청을 한다. 총 10회의 이주 요청 시나리오를 구성하였으며, 이주 요청 시나리오는 Table 5와 같이 구성하여 실험한다. 이주효율은 RRS를 적용한 클라우드 시스템에서 목적지 서버로 이주하는데 걸리는 시간과 비교했을 때, 얼마나 빠른지 나타낸 것으로 T_r , T_d 값을 측정한 후 Eq. (6)으로 도출한다.

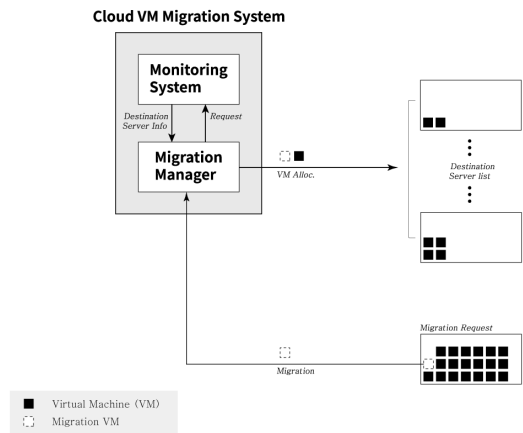


Fig. 6. Cloud VM Migration Efficiency Test Bed

Table 5. VM Live Migration Request Scenario

time	CPU	MEM	I/O
1	10	15	12
2	15	13	21
3	17	15	9
3	22	17	11
4	35	18	16
5	42	16	18
6	21	18	22
7	27	19	16
8	32	18	33
9	23	22	24
10	9	21	21

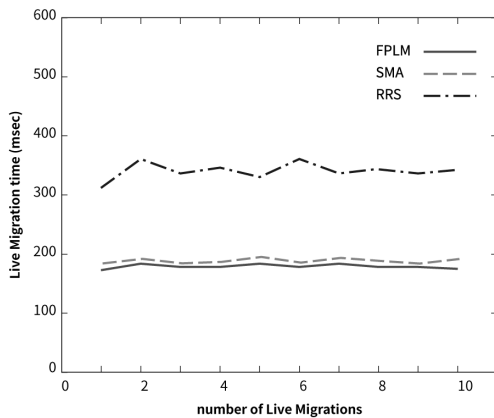
$$E = \frac{T - |(t_d - t_r) - T|}{T} \times 100 \quad (6)$$

T = RRS Migration Time

Eq. (6). VM Live Migration Efficiency

4.2.2 결과 및 분석

Fig. 7은 이주관리자가 이주요청을 받은 순간부터 가상머신 이주가 완료 될 때까지의 시간을 측정된 그래프이다. 그래프의 가로축은 가상머신 이주 발생 횟수를 나타내며, 세로축은 가상머신 이주에 소비된 시간을 보여준다. 이주요청 시나리오에 따라 가상머신 이주는 총 10번 일어났기 때문에 이주발생 횟수 역시 10번이 된다. FPLM,

**Fig. 7.** VM Live Migration time

SMA, RRS의 가상머신 이주 소요시간 차이는 모니터링 시스템으로부터 받은 목적지 서버 정보를 이주관리자가 해석하여 가상머신 호스트를 이주하는데, 출발지와 목적지 물리서버간 성능차이로 인한 대기 손실 시간과 목적지 서버 예측을 위한 공간 복잡도의 차이에서 발생하였다. RRS를 적용한 시스템은 항상 모든 물리서버를 순차적으로 탐색하여 이주관리자에게 전달을 하기 때문에 이주 소요 시간 역시 평균 357ms(303ms~417ms)의 큰 소요 시간을 보인다. 반면 FPLM을 적용한 시스템은 이주 발생 시 최고 186ms(175ms~186ms)의 비교적 적은 소요 시간을 보여준다. 줄어든 탐색 공간의 크기, 그리고 가상머신 이주 요청 물리서버와 유사한 성능의 목적지 물리서버로의 이주로 인하여 결과적으로 가상머신 이주에 걸리는 시간이 RRS와 비교 65%, SMA와 비교 3% 정도 단축되었다. FPLM은 주기적으로 발생하는 자원 사용량 예측 오버헤드를 탐색규모를 감소시킴으로써 줄였기 때문에 SMA보다 효율적이다.

5. 결론

본 논문에서는 클라우드 컴퓨팅 환경에서 이주 효율성 향상 및 목적지 물리서버 결정 시 발생하는 오버헤드 감소를 위해 FPLM을 제안하였다. 그리고 가상머신 이주 상황이 발생하였을 때 제한된 시간 내에 이주 발생 횟수를 RRS, SMA 방식과 비교 측정하는 실험을 통해 FPLM이 이주 발생 횟수 및 목적지 서버 결정을 위한 오버헤드를 감소시켜 이주효율성을 향상시킬 수 있음을 실험을 통해 증명하였다. 하지만 실제 데이터센터가 아닌 이주 효율성 측정과 검증을 위해 모델링된 데이터센터 환경에서의 가상머신 실시간 이주를 하였고 때문에 한계가 존재한다. 따라서 향후에는 사용자의 가상머신 요청이 변화하는 상황에서의 가상머신 실시간 이주 기법 및 효율성과 관련된 연구를 진행할 계획이다.

References

1. Chang, J.H., Lee, W.J., Jeon, C.H., "Performance Evaluation of WAN Storage Migration Scheme for Cloud Computing Environment", Journal of The Korea Society of Computer and Information, Vol. 17, No 5, pp. 1-7, 2012.
2. Choi, H.S., Ko, Y.R., Park, S.Y., "A Simulation-based Migration Algorithm Minimizing the Number of Migrations in Server Virtualization Environments", KIISE, pp. 159-

- 169, 2011.
3. Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I., Warfield, A., "Live Migration of Virtual Machines" In Proceedings of Second Symposium Networked Systems Design and Implementation, NSDI, pp. 273-286, 2005.
 4. Liu, H., Xu, C.Z., Jin, H., Gong, J., Liao, X., "Performance and Energy Modeling for Live Migration of Virtual Machines", HPDC, pp.1-11, 2011.
 5. Ma, F., Liu, F., "Live virtual machine migration based on improved pre-copy approach", IEEE International Conference on Software Engineering and Service Sciences (ICSES), pp. 230-233, 2010.
 6. Mamdani, E.H., Assilian, S., "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller", International Journal of Man-Machine Studies, Vol. 7, pp. 1-13, 1974.
 7. NIPA, "Web service development guide in cloud computing environment", pp. 16-17, 2013.
 8. Zari, M., Saiedian, H., Naeem, M., "Understanding and reducing web delays", IEEE, pp. 31-37, 2001.
 9. Zeigler, B.P., Praehofer, H., Kim, T.G., Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems, 2nd Edition, Academic Press, pp. 76-96, 2000.
 10. Mayank, M., Anirudha, S., On Theory of VM Placement : Anomalies in Existing Methodologies and Their Mitigation Using a Novel Vector Based Approach, IEEE, pp. 277-278, 2011.
 11. Greenberg, A., Hamilton, J.R., Jain, N., et al. "VL2: a scalable and flexibledata center network", ACM SIGCOMM Computer Communication Review, Vol. 39, pp. 56-58, 2009.



박 민 오 (park-min-o@nate.com)

2013 인하대학교 컴퓨터정보공학과 학사
2013~인하대학교 컴퓨터정보공학과 석사과정

관심분야 : 소프트웨어공학, 클라우드 컴퓨팅, 모델링&시뮬레이션



김 재 권 (jaekwonkorea@naver.com)

2011 가천의과대학교 정보처리과 학사
2013 인하대학교 컴퓨터정보공학과 석사
2013~인하대학교 컴퓨터정보공학과 박사과정

관심분야 : 클라우드 컴퓨팅, 인공지능, 모델링&시뮬레이션



최 정 석 (jeongseokchoi.korea@gmail.com)

2015 인하대학교 컴퓨터정보공학과 학사
2015~인하대학교 컴퓨터정보공학과 통합과정

관심분야 : 클라우드 컴퓨팅, 모델링&시뮬레이션



이 중 식 (jslee@inha.ac.kr)

1993 인하대학교 전자공학과 학사
1995 인하대학교 전자공학과 석사
2001 미국 애리조나대 전기·컴퓨터공학과 박사
2001~2002 캘리포니아 주립대학교 전기·컴퓨터공학과 전임강사
2002~2003 클리블랜드 주립대학교 전기·컴퓨터공학과 조교수
2003~인하대학교 컴퓨터정보공학과 교수

관심분야 : 소프트웨어공학, 모델링&시뮬레이션