

# 클라우드 프로비저닝 서비스를 위한 퍼지 로직 기반의 자원 평가 방법

김재권<sup>1</sup> · 이종식<sup>1†</sup>

## Fuzzy Logic-driven Virtual Machine Resource Evaluation Method for Cloud Provisioning Service

Jae-Kwon Kim · Jong-Sik Lee

### ABSTRACT

Cloud computing is one of the distributed computing environments and utilizes several computing resources. Cloud environment uses a virtual machine to process a requested job. To balance a workload and process a job rapidly, cloud environment uses a provisioning technique and assigns a task with a status of virtual machine. However, a scheduling method for cloud computing requires a definition of virtual machine availabilities, which have an obscure meaning. In this paper, we propose Fuzzy logic driven Virtual machine Provisioning scheduling using Resource Evaluation(FVPRE). FVPRE analyzes a state of every virtual machine and actualizes a value of resource availability. Thus FVPRE provides an efficient provisioning scheduling with a precise evaluation of resource availability. FVPRE shows a high throughput and utilization for job processing on cloud environments.

**Key words** : Cloud Service, Virtual Machine Provisioning, Job Scheduling, Fuzzy Logic, FVPRE

### 요약

클라우드 환경은 여러 개의 컴퓨팅 자원들을 이용하는 분산 컴퓨팅 환경의 일종으로 가상머신을 이용 하여 작업을 처리한다. 클라우드 환경은 작업 요청에 따르는 부하분산과 빠른 작업 처리를 위한 프로비저닝 기술을 이용하여 가상머신의 상태에 따라 작업을 할당 한다. 하지만, 클라우드 환경의 작업 스케줄링을 위해서는 가상머신의 성능에 따르는 애매모호한 상태에 대한 가용성의 정의가 필요하다. 본 논문에서는 클라우드 환경의 프로비저닝 스케줄링을 위해 퍼지 로직 기반의 자원평가를 이용한 가상머신 프로비저닝 스케줄링(FVPRE: Fuzzy logic driven Virtual machine Provisioning scheduling using Resource Evaluation)을 제안한다. FVPRE는 각 가상머신의 정의하기 어려운 성능의 상태를 분석하여 자원 가용성에 대한 값을 구체화 하여 정확한 자원의 가용성 평가를 통해 효율적인 프로비저닝 스케줄링이 가능하다. FVPRE는 클라우드 환경의 작업 처리에 대해 높은 처리율과 활용율을 보인다.

**주요어** : 클라우드 서비스, 가상머신 프로비저닝, 작업 스케줄링, 퍼지 로직, FVPRE

## 1. 서론

세계적으로 데이터의 폭발적인 증가와 이를 처리하기 위한 방법이 이슈가 되고 있으며, 이에 클라우드 컴퓨팅

은 가상화 환경을 이용하여 리소스 인프라를 구성하여 컴퓨팅 환경의 소모비용을 절감하고 확장성, 유연성을 높이며 자원을 통합 관리하는 것으로 QoS(Quality of Service) 제공에 큰 주목을 받고 있다(Rodrigo 외 2011).

클라우드 컴퓨팅은 크게 SaaS(Software as a Service), PaaS(Platform as a Service), IaaS(Infrastructure as a Service)로 구분이 되며, SaaS는 사용자에게 콘텐츠 서비스 제공에 목적이 있으며, PaaS는 서비스의 요청에 대해 어떻게 처리할지에 대한 목적이 있다. 그리고 IaaS는 데이터 센터와 같은 물리적 자원을 활용하여 가상화 환경을 구축하여 작업 처리에 목적을 두고 있다.

\*이 논문은 인하대학교의 지원에 의하여 연구되었음.  
접수일(2012년 7월 3일), 심사일(2012년 9월 5일),  
게재 확정일(2012년 9월 25일)

<sup>1)</sup> 인하대학교 정보공학과

주 저 자: 김재권

교신저자: 이종식

E-mail; jaekwonkorea@naver.com, jslee@inha.ac.kr

가상화는 하나 이상의 물리적 서버를 다수의 가상머신들이 공유를 가능하게 하여 자원 활용 율을 향상시킬 수 있으며, 이를 통해 운영비용의 절감으로 클러스터의 전체 성능이 향상이 가능하다(Daniel 외 2009). 하지만, 가상화 환경에서는 클러스터로 구축된 물리적 서버 중 특정 서버에 자원 이용이 편중 될 수 있으며 이는 전체 성능의 저하를 가져오게 된다(Cherkasova 외 2005).

이를 해결하기 위해서는 가상머신에 대해 전체 성능을 저하하는 요인을 분석하여, 다음 작업에 대해 특정 자원에 편중을 줄일 수 있는 리소스 프로비저닝이 필요하다. 리소스 프로비저닝은 PaaS에 구성되어 있는 중요 모듈로서, 자원 리소스들에 균등한 부하분산이 가능하기 때문에 성능 저하문제를 줄일 수 있다(Ye 외, 2009). 하지만, 가상머신의 프로비저닝을 위해서는 각 가상머신의 CPU, RAM, Net I/O의 상태에 맞는 프로비저닝을 해야 하지만 그 기준이 불명확하다. 따라서 가상머신의 성능에 맞는 부하 분산을 판단해야 하며, 판단된 부하 분산에 따라 다음 작업을 요청받으면, 부하가 적은 가상머신에게 작업을 할당해야한다. 즉, 클라우드 환경의 작업 스케줄링을 위해서는 가상머신의 성능에 따르는 애매모호한 상태에 대한 구체적인 가용성의 평가가 필요하다(Shaout 외, 1998). 이를 해결하기 위한 퍼지 로직(Stuart 외, 1995)은 애매모호한 데이터에 대해 추론이 가능한 대표적인 기법이며, 불명확한 데이터에 대한 의사결정에 많은 장점이 있다.

본 논문에서는 클라우드 환경의 프로비저닝 스케줄링을 위해 퍼지 로직 기반의 자원평가를 이용한 가상머신 프로비저닝 스케줄링(FVPRE: Fuzzy logic driven Virtual machine Provisioning scheduling using Resource Evaluation)을 제안한다. 퍼지 로직은 각 가상머신의 정의하기 어려운 성능의 상태를 분석하여 자원 가용성에 대한 값을 구체화하여 보다 정확한 자원의 가용성 평가를 통해 효율적인 프로비저닝 스케줄링이 가능하다.

본 논문의 구성은 제 2장에서는 관련연구로서 자원관리 스케줄링과 퍼지로직에 대해 기술한다. 제 3장은 퍼지 로직 기반의 프로비저닝 스케줄링을 기술한다. 제 4장은 제안하는 기법의 실험 및 결과를 기술하며, 제 5장은 결론에 대해 기술한다.

## 2. 관련연구

### 2.1 자원관리 스케줄링

클라우드 서비스를 제공하기 전에 인프라의 성능과 제 공자의 수익을 예측하고 설계함으로써 자원 인프라의 집

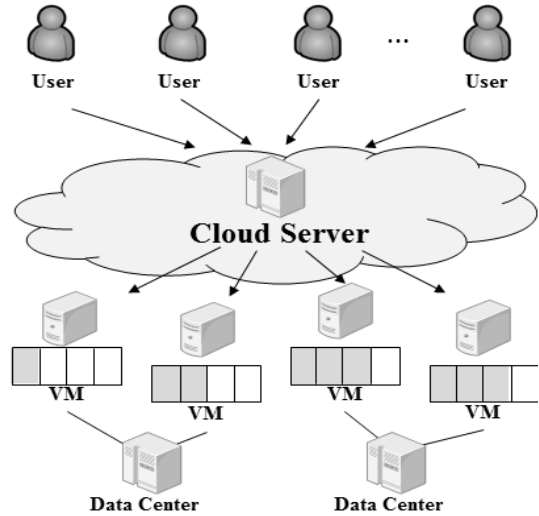


Fig. 1. Cloud Scheduling structure

근 비용을 최적화하여 이익을 향상시킬 수 있어야 한다 (Ma 외, 2011).

클라우드 서비스의 자원관리 스케줄링의 구조는 Fig. 1과 같이 사용자가 서비스를 요청하면, 클라우드 서버에서는 작업의 개수가 가장 적은 가용성이 좋은 가상머신 혹은 자원을 선택하여 작업을 수행해야한다. 또한 데이터 센터는 가상머신을 구성하고 있으며, 한 개의 데이터 센터가 복수개의 가상머신을 가지거나, 여러 개의 데이터 센터가 한 개의 가상머신으로 구성될 수도 있다(Mousumi 외, 2011).

클라우드 스케줄링에 따른 클라우드 서비스를 위해서는 이론적이나 막연한 측정이 아닌 시뮬레이션을 통해 자원 관리 상황을 예측하는 것이 필요하다(Rajkumar 외, 2009). 클라우드 자원 관리를 위해서는 자원 할당 기법이 필요하며, 자원 할당기법에 대한 연구가 진행 중이다.

첫 번째로, 자원의 신뢰성 측정 알고리즘(Resource Reliability Measurement Algorithm : RRMA)(Park 외, 2007)은 자원들의 신뢰성을 측정하여 작업을 할당하는 스케줄링 모델로서, 자원의 신뢰성이란 각 자원들이 주어진 환경에서 담당 기능을 원활 하게 수행할 수 있는 능력이다. 이를 측정하여 가장 신뢰성이 높은 자원으로 작업을 할당한다. 두 번째로, 실행시간과 대기시간에 따른 스케줄링 기법인 위한 반응시간 최적화된 자원 관리 스케줄링(Response time Optimization-based Resource Management Scheduling : RORMS)(Assuncao 외 2009)은 큐이론을 바탕으로, 각 자원이 할당되는 시점과 이전의 자원 처

리 시간을 비교하여 작업 처리시간이 가장 빠른 자원을 선택하는 기법으로, 짧은 대기시간과 활용 율을 높였다. 본 논문에서는 클라우드 환경의 프로비저닝 스케줄링을 위해서 퍼지로직 기반 스케줄링을 가상머신 프로비저닝에 도입하기 위해 새로운 방법과 메커니즘을 제시한다.

### 2.2 퍼지로직

퍼지로직은 데이터나 시스템의 복잡성을 설명하며, 모델의 복잡성을 단순화 하는 방법을 제공한다. 퍼지로직은 모호한 정도를 조절할 수 있는 집합으로서 온도, 키, 몸무게, 속도 등의 연속형 데이터의 경계가 불분명한 척도를 분류할 수 있는 기법이다. 2치 논리인 불 논리(Boolean logic)는 0, 1(완전한 거짓, 완전한 참)을 판단하는 것이다. 하지만 퍼지로직은 다치 논리(multi-value logic)이며, 소속도(degrees of membership)와 진리도(degrees of truth)를 다루기 때문에 0과 1사이의 연속된 논리값을 계산한다.

이와 같이 퍼지이론은 수학적으로 애매모호한 데이터 모델링 평가문제에 있어서 단순화하여 데이터 모델링을 더욱 쉽게 하며, 현재는 인공지능 신경망(ANN), 기계학습(Machine Learning)등과 같은 다양한 하이테크 기법에 활용되고 있으며 하이브리드식의 알고리즘에도 적용한다.

퍼지로직은 연속형 입력 자료를 퍼지화한 퍼지 함수(fuzzy function)와 이를 계산하기 위한 퍼지 룰(If-Then

based)을 결합하여 결과를 출력하며, 이 결과 값을 여러 역퍼지화(Defuzzification) 방법을 통해 실제 출력 값을 도출한다. 본 논문에서는 퍼지 규칙의 추론 방법으로 맘다니(Mamdani)의 Min-Max 합성법을 이용하고, 실제 실수 값을 출력을 위한 역퍼지화에는 무게중심법(COG, Center Of Gravity)(Lee, 1990)를 이용한다.

## 3. 퍼지로직 기반 프로비저닝 스케줄링

### 3.1 시스템 구성

프로비저닝 스케줄링을 위해서는 클라우드 서버의 환경을 구성하기 위해 필요 컴포넌트를 구성해야 한다. 본 논문에서도 클라우드 환경의 프로비저닝 스케줄링을 위해 퍼지 로직 기반의 자원평가를 이용한 가상머신 프로비저닝 스케줄링(FVPRE: Fuzzy logic driven Virtual machine Provisioning scheduling using Resource Evaluation)을 구성하기 위해서 Fig. 2와 같이 클라우드 서버 환경을 정의한다.

- Software as a Service

SaaS는 사용자에게 서비스를 직접 제공하는 모듈로서, 사용자가 클라우드 서비스 이용을 하기 위해 Portal을 이용하여 접속을 한다. 사용자가 서비스 요청을 하게 되면, Admission Service에서는 서비스 요청을 확인하며, 콘텐

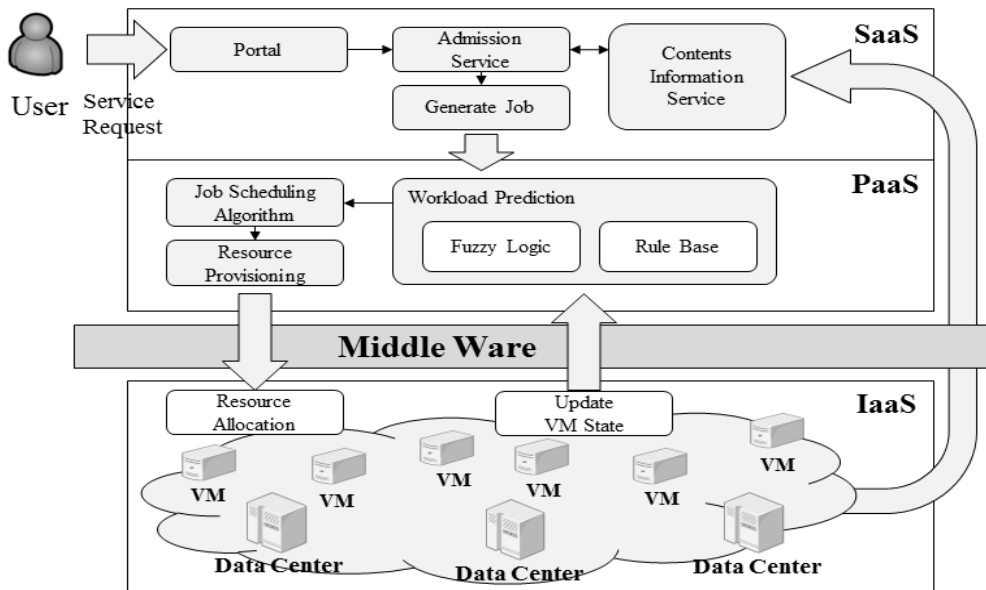


Fig. 2. Cloud Server Architecture

츠 서비스 제공을 하기 위해 Job Generate에서 작업을 생성한다. 생성된 작업은 PaaS를 통해 전달이 되며, 사용자는 Contents Information Service를 통해 클라우드 서비스를 제공받게 된다.

• Platform as a Service

PaaS는 작업 요청에 대해서 자원들의 상태를 고려하여 부하분산을 줄이고, 최적의 QoS를 제공하기 위한 모듈이다. Workload Prediction은 가상머신의 상태에 따라서 앞으로의 부하 분산을 예측하기 위한 모듈로서, 퍼지 로직과 룰베이스를 통해 현재의 가상머신에 대한 자원평가를 하게 된다. 자원평가를 완료하면 Job Scheduling Algorithm 모듈에서는 평가 정보를 기반 하여 새로운 작업에 대한 요청 시 어느 가상머신을 활용할지 계산한다. 이후, Resource Provisioning을 통해 최종적으로 어느 가상머신에 할당할지 결정한다.

• Infrastructure as a Service

클라우드 서비스 제공을 위해 작업을 처리하는 부분으로서, 플랫폼과 인프라를 연결하는 미들웨어와 가상머신과 데이터 센터로 구성되어 있다. PaaS로부터 가상머신 할당 정보를 받으면, Resource Allocation에서는 해당 가상머신에 작업을 할당한다. 가상머신에서는 작업을 처리하며, 현재의 가상머신 상태를 PaaS에 전송하기 위해 Update VM Status를 통하여 가상머신의 정보를 전송하게 된다. 또한, 작업이 완료되면 SaaS로 전송하게 되어 사용자에게 서비스를 제공한다.

3.2 작업처리 구조

클라우드 서비스의 프로비저닝을 위해서는 작업 처리 구조에 대해 정의를 해야 하며, 본 논문에서는 큐잉모델 기반의 작업처리 구조에 기반된 IaaS 환경에서 스케줄링 기법을 제안한다. 큐잉모델기반 작업처리 구조는 Fig. 3과 같다.

큐잉모델 기반 작업처리 구조는 가상 머신 안에 작업을 처리하는 Physical Computing이 있으며, 이를 처리하기 전에 작업들은 큐에서 대기하며 큐는 선입 선 처리 방식인 FCFS 이다. Application Provisioning은 작업 스케줄링에 따르는 가상머신을 정하게 되며, 작업처리시간, 복수개의 가상머신들의 상태에 따라 다르기 때문에 m/m/∞ 이다. 또한 작업이 끝난 이후에는 Update VM Status로부터 현재의 가상머신의 상태를 Workload Prediction으로 전송한다.

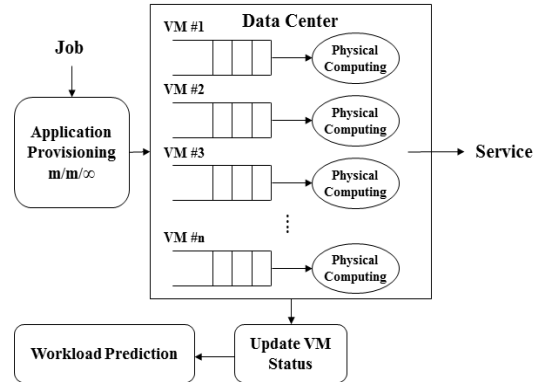


Fig. 3. Queueing-Model based Job Processing Structure

Table 1. Virtual machine Statics Parameter

Property Name	Description	Value Type	Unit
VM ID	가상머신 식별자	Integer	1 ~ i
CPU Utilization	가상머신 CPU 활용률	Float	%
RAM Utilization	가상머신 주 기억 장치 활용률	Float	%
Net	네트워크의 지연시간	Float	Mbps
Processing Time	가상머신의 작업처리 속도	Integer	
Queue Size	가상머신의 작업 저장 갯수	Integer	Job
VM State	가상머신의 상태	Float	
VM Available	가상머신의 가용능력	Float	

3.3 프로비저닝 스케줄링 알고리즘

클라우드 시스템의 구성과 작업처리 구조에 대해 정의하였으며, 다음으로 퍼지 로직 기반의 자원평가를 이용한 프로비저닝 스케줄링에 대해 설명한다. 가상머신의 상태에 따라 자원평가와 작업 스케줄링을 구성하기 위해 Table 1과같이 가상머신의 정적속성을 정의한다.

위와 같이 가상머신의 정적 속성을 구성하였으며, i는 가상머신의 개수를 의미하며, Processing Time과 Queue Size는 가상머신의 성능을 의미한다. 그리고 VM Available는 자원평가를 위해 쓰이는 속성이다.

가상머신의 정적 속성에 따라 자원을 평가하여 작업 스케줄링을 하게 되는 프로비저닝 스케줄링 알고리즘의 구조는 Fig. 4와 같다.

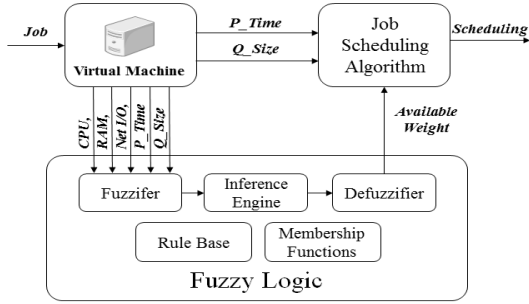


Fig. 4. Fuzzy based Algorithm structure

가상머신은 작업을 받으면 작업을 수행하며, 현재의 가상머신 상태인 CPU, RAM, Net I/O, Processing Time, Queue Size를 퍼지 로직으로 전송한다. 퍼지 로직은 자원을 평가하는 모듈이며, 룰 베이스와 멤버십 함수로 구성되어 있으며, 퍼지 엔진을 통해 추론된 가상머신 상태에 대한 가상머신의 가중치 정보를 작업 스케줄링으로 전송하게 된다. 이후, 작업 스케줄링 알고리즘에서는 계산을 통해 다음 작업처리가 가능한 가상머신을 선택하기 위해 계산한다. 클라우드 서버에서는 사용자에게 작업을 받았을 경우, 가용성이 가장 높은 가상머신이 작업을 할당받게 된다. 작업 스케줄링 알고리즘을 통해 할당하기 위한 식은 식 (1)과 같다.

$$\begin{aligned}
 Ptime_i &= Ptime_i / \text{Max} \left( \sum_{j=0}^i Ptime_j \right) \\
 Qsize_i &= |1 - (Qsize_i / \text{Max} \left( \sum_{j=0}^i Qsize_j \right))| \\
 Available_i &= Ptime_i + Qsize_i + VMstate_i \\
 Select\ VM &= \text{Max}(Available_i)
 \end{aligned} \quad (1)$$

각각의 가상머신의 처리 시간과 큐 사이즈와 가상머신의 가중치를 통해 가상머신 가용능력을 산출하며, 각 가상머신 중 가장 가용능력이 높은 가상머신을 작업할당한다. 여기서 가상머신의 가중치는 퍼지 로직을 통해 자원 평가로 인해 이루어진다.

### 3.4 퍼지 로직

퍼지 로직을 통해 자원 평가를 위한 FVPRE의 퍼지 모델은 Fig. 5와 같으며, 각 매개변수는 Table 2와 같다. 각 매개변수는 퍼지의 입력과 출력을 의미한다.

퍼지 로직기반의 자원 평가를 위해서 5가지의 퍼지 입

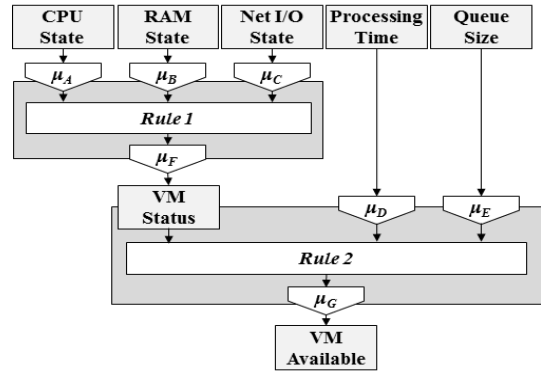


Fig. 5. Fuzzy Model

Table 2. Fuzzy Parameter Variable

Parameter	State variable	Linguistic variable
$\mu_A$ (입력)	CPU Utilization	Very Low(VL), Low(L), Middle(M), High(H), Very High(VH)
$\mu_B$ (입력)	RAM Utilization	Very Low(VL), Low(L), Middle(M), High(H), Very High(VH)
$\mu_C$ (입력)	Net	Very Low(VL), Low(L), Middle(M), High(H), Very High(VH)
$\mu_D$ (입력)	Processing Time	Very Low(VL), Low(L), Middle(M), High(H), Very High(VH)
$\mu_E$ (입력)	Queue Size	Very Low(VL), Low(L), Middle(M), High(H), Very High(VH)
$\mu_F$ (입력, 출력)	VM Status	Very Low(VL), Low(L), Middle(M), High(H), Very High(VH)
$\mu_G$ (출력)	VM Available	Very Low(VL), Low(L), Small Low(SL), Middle(M), Small High(SM), High(H), Very High(VH)

력변수를 사용했으며, 본 논문에서는 2단계에 걸쳐 추론을 위해 1가지의 입력 및 출력 변수, 1가지의 출력변수를 이용하며, 2단계에 걸쳐서 퍼지 추론을 하게 된다.

입력변수의 CPU, RAM, Net I/O는 컴퓨터 자원의 성능을 파악하는 필수적인 요소이다. Processing Time과 Queue Size는 자원이 실제 처리하는 평균속도와 여러 작업이 요청되었을 경우 FIFO 방식으로 처리하는 Queue에 대한 변수이다.

VM State와 VM Available은 퍼지를 이용한 출력 변수로서, 일반적인 자원 성능 평가 방법과 처리속도, 처리해야 하는 작업 개수를 종합하여 평가를 한다. 즉, 1차 추론은 VM State의 평가치이며 일반적인 자원 성능에 대한 평가이다. 2차 추론은 VM Available을 평가하며, 일

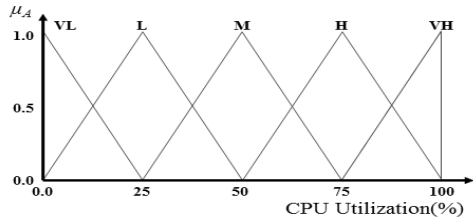
반적인 평가방법(VM State)과 처리속도, Queue 크기를 계산한다. 각 단계에 대한 내용은 다음과 같다.

첫 번째 단계는, CPU, RAM, Net I/O의 성능을 이용하여 가상머신의 상태(VM State)를 추론한다. 두 번째로, 측정된 가상머신의 상태와 가상머신의 프로세싱 성능과 큐에 있는 작업의 개수를 추론하여 최종적으로 가상머신의 가용도(VM Available)를 추론한다. 가상머신의 가용도를 이용하여 자원을 평가한다.

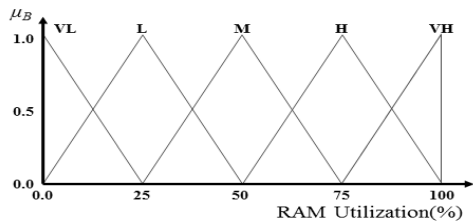
$\mu_A$ ,  $\mu_B$ ,  $\mu_C$  는 현재의 상태에 대해 퍼지를 구성하게 되며 멤버십 함수는 Fig. 6(a),(b),(c)와 같다. 또한  $\mu_C$ 는 식 2와 같은 과정을 통해 네트워크의 지연시간을 산출하여 멤버십 함수에 적용이 된다.

$$\mu_{C_i} = \text{Min}(\sum_{j=0}^i \text{Net}_j) / \text{Net}_i \leq 1 \quad (2)$$

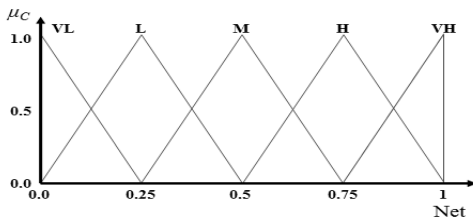
이를 통해  $\mu_F$ 를 추론하게 되며 멤버십 함수는 Fig. 7과 같다. 또한, 추론을 위한 룰 베이스 1은 Table 3과 같다.



(a)  $\mu_A$  of CPU Utilization Membership Function



(b)  $\mu_B$  of RAM Utilization Membership Function



(c)  $\mu_C$  of Net Membership Function

Fig. 6.  $\mu_A$ ,  $\mu_B$ ,  $\mu_C$  Membership Function

Table 3과 같이  $\mu_F(\text{VM State})$ 의 추론을 위해 125개의 룰과 각 멤버십 함수를 표현했으며, 각각의 퍼지 입력들은 AND 연산관계를 가지고 있다. 또한 역퍼지화를 위해서 무게중심법(COG)을 이용한다. 두 번째 추론 단계인  $\mu_G$ 를 추론하기 위해서  $\mu_D$ ,  $\mu_E$ 와 첫 번째 단계에서 추론된

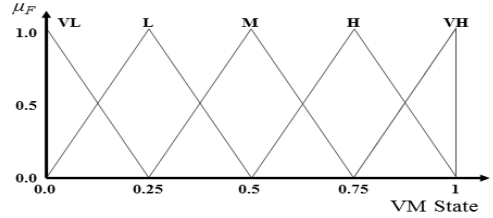


Fig. 7.  $\mu_F$  of VM State Membership Function

Table 3. Rule Base 1

Parameter	$\mu_C$										
	$\mu_A$	$\mu_B$	VL	L	M	H	VH				
VL	VL	R1	VL	R6	VL	R11	VL	R16	VL	R21	VL
	L	R2	VL	R7	VL	R12	VL	R17	VL	R22	VL
	M	R3	VL	R8	VL	R13	VL	R18	VL	R23	VL
	H	R4	VL	R9	VL	R14	VL	R19	VL	R24	L
	VH	R5	VL	R10	VL	R15	VL	R20	L	R25	L
L	VL	R26	VL	R31	VL	R36	L	R41	L	R46	L
	L	R27	VL	R32	L	R37	L	R42	L	R47	L
	M	R28	L	R33	L	R38	L	R43	L	R48	L
	H	R29	L	R34	L	R39	L	R44	L	R49	M
	VH	R30	L	R35	L	R40	L	R45	M	R50	M
M	VL	R51	L	R56	L	R61	M	R66	M	R71	M
	L	R52	L	R57	M	R62	M	R67	M	R72	M
	M	R53	M	R58	M	R63	M	R68	M	R73	M
	H	R54	M	R59	M	R64	M	R69	M	R74	H
	VH	R55	M	R60	M	R65	M	R70	H	R75	H
H	VL	R76	M	R81	M	R86	H	R91	H	R96	H
	L	R77	M	R82	H	R87	H	R92	H	R97	H
	M	R78	H	R83	H	R88	H	R93	H	R98	H
	H	R79	H	R84	H	R89	H	R94	H	R99	VH
	VH	R80	H	R85	H	R90	H	R95	VH	R100	VH
VH	VL	R101	H	R106	H	R111	VH	R116	VH	R121	VH
	L	R102	H	R107	VH	R112	VH	R117	VH	R122	VH
	M	R103	VH	R108	VH	R113	VH	R118	VH	R123	VH
	H	R104	VH	R109	VH	R114	VH	R119	VH	R124	VH
	VH	R105	VH	R110	VH	R115	VH	R120	VH	R125	VH

$\mu_F$  이용하여 최종적으로 가상머신을 평가하는 가중치를 산출한다. 또한,  $\mu_D$ (Processing Time),  $\mu_E$ (Queue Size)는 식 (3)과 같은 과정을 거친 후 퍼지화를 한다.

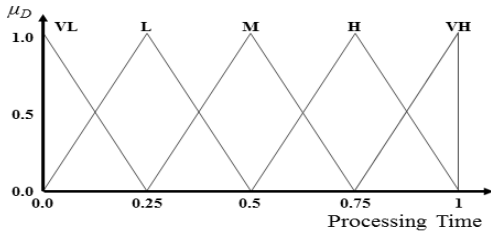
$$\mu D_i = (1 - (Ptime_i / Max(\sum_{j=0}^i Ptime_j))) \leq 1$$

$$\mu E_i = (1 - (Qsize_i / Max(\sum_{j=0}^i Qsize_j))) \leq 1 \tag{3}$$

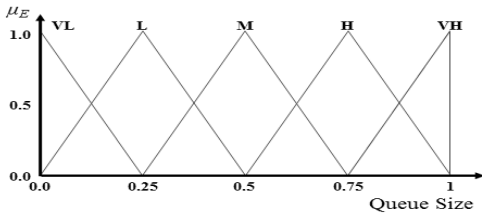
$\mu_D$ ,  $\mu_E$ 의 멤버십 함수는 Fig. 8(a),(b)와 같으며,  $\mu_G$ 의 멤버십 함수는 Fig. 9와 같다. 또한 추론을 위한 룰 베이스 2는 Table 4와 같다.

위와 같은 과정을 통해 125개의 룰과 멤버십 함수를 이용하여 가상머신을 평가를 완료한 후,  $\mu_G$ (VM Available)를 결정하게 된다.  $\mu_G$ 의 역퍼지화는 무게중심법(COG)를 이용한다.

퍼지 로직을 통해 가상머신의 가중치를 판단한 후, 작



(a)  $\mu_D$  of Processing Time Membership Function



(b)  $\mu_E$  of Queue Size Membership Function

Fig. 8.  $\mu_D$ ,  $\mu_E$  Membership Function

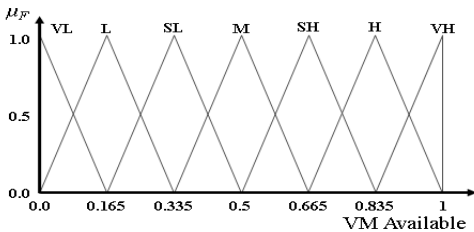


Fig. 9.  $\mu_G$  of VM Available Membership Function

Table 4. Rule Base 2

Parameter	$\mu_F$										
	$\mu_D$	$\mu_E$	VL	L	M	H	VH				
VL	VL	R1	VL	R6	VL	R11	VL	R16	VL	R21	VL
	L	R2	VL	R7	VL	R12	VL	R17	VL	R22	VL
	M	R3	VL	R8	VL	R13	VL	R18	VL	R23	L
	H	R4	VL	R9	VL	R14	VL	R19	L	R24	L
	VH	R5	VL	R10	L	R15	L	R20	L	R25	L
L	VL	R26	L	R31	L	R36	L	R41	L	R46	L
	L	R27	L	R32	L	R37	L	R42	SL	R47	SL
	M	R28	L	R33	L	R38	SL	R43	SL	R48	SL
	H	R29	L	R34	SL	R39	SL	R44	SL	R49	SL
	VH	R30	SL	R35	SL	R40	SL	R45	SL	R50	M
M	VL	R51	SL	R56	SL	R61	SL	R66	M	R71	M
	L	R52	SL	R57	SL	R62	M	R67	M	R72	M
	M	R53	M	R58	M	R63	M	R68	M	R73	M
	H	R54	M	R59	M	R64	M	R69	SH	R74	SH
	VH	R55	M	R60	M	R65	SH	R70	SH	R75	SH
H	VL	R76	M	R81	SH	R86	SH	R91	SH	R96	SH
	L	R77	SH	R82	SH	R87	SH	R92	SH	R97	H
	M	R78	SH	R83	SH	R88	SH	R93	H	R98	H
	H	R79	SH	R84	SH	R89	H	R94	H	R99	H
	VH	R80	H	R85	H	R90	H	R95	H	R100	H
VH	VL	R101	H	R106	H	R111	H	R116	H	R121	VH
	L	R102	H	R107	H	R112	VH	R117	VH	R122	VH
	M	R103	H	R108	VH	R113	VH	R118	VH	R123	VH
	H	R104	VH	R109	VH	R114	VH	R119	VH	R124	VH
	VH	R105	VH	R110	VH	R115	VH	R120	VH	R125	VH

업 스케줄링의 알고리즘에 적용이 되며, 작업을 할당받으면 가용성이 가장 높은 가상머신을 선택하게 된다.

## 4. 실험

본 논문에서 제안하는 FVPRE의 효과를 입증하기 위해서 DEVS 형식론(Zeigler 외, 2000)을 적용하여 클라우드 가상화 환경을 구성하고, 성능을 측정하기 위해 평균 작업 처리시간(Average of Turn Around Time), 작업 처리율(Throughput)을 측정한다.

### 4.1 테스트 환경 구성

FVPRE의 성능 측정을 위해서 DEVS 형식론 기반으로 Fig. 10과 같이 실험 테스트 환경을 모델링 한다.

실험 시나리오를 구성하기 위해서 첫 번째 아토믹 모델인 Job Generator는 작업을 생성하고 작업을 보내는 모듈로서, 사용자의 요청하는 작업을 모델링 한 것이다. 두 번째로, Provisioning은 작업 요청을 받으면, 가장 가용성이 높은 가상머신에게 작업을 할당하는 모듈이며, 작업을

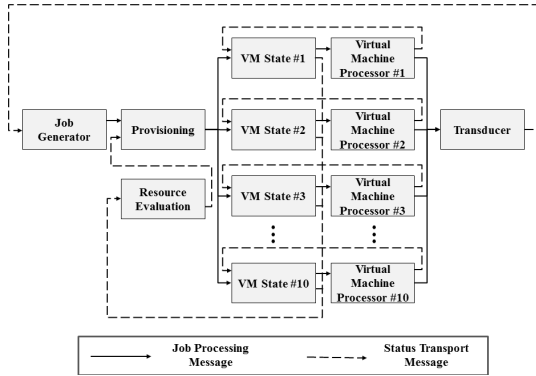


Fig. 10. Cloud Test Bed

할당하기 위해 Resource Evaluation의 정보를 이용한다. 세 번째로, VM Status는 작업을 할당받으면 처리하기 위해 Virtual Machine Processor로 작업을 보내기도 하며, 가상머신의 상태에 대해 전송하는 기능을 담당하고 있으며 가상머신의 모든 상태를 Resource Evaluation으로 전송한다. 마지막으로 Virtual Machine Processor에서 작업을 완료하게 되면 Transducer로 작업이 완료 되었다는 신호를 알리게 되며 활용율, 처리율, 평균 처리시간을 평가하게 된다. 커플링은 작업을 처리하는 순서를 의미하는 Job Processing Message와 계산이나 참조를 위해 상태를 전송하는 Status Transport Message로 구성된다. 총 10개의 가상머신을 구성하였으며, 사용자는 300개의 서비스를 요청한다.

위와 같이 시나리오와 테스트 환경을 구성하였으며, 다음으로 가상머신의 성능을 Table 5와 같이 구성하여 실험을 한다.

비교를 위한 실험 모델은 총 3가지이며, 첫 번째로는, 자원의 신뢰성을 측정하여 작업 할당을 하는 자원의 신뢰성 측정 알고리즘(Resource Reliability Measurement Algorithm : RRMA)(Park 외, 2007) 두 번째는, 실행 시간과 대기시간을 계산하여 반응 시간을 최적화시킨 자원 관리 스케줄링(Response time Optimization-based Resource Management Scheduling : RORMS) (Assouncao 외 2009) 마지막으로, 본 논문에서 제안하는 퍼지 로직 기반의 자원평가를 이용한 가상머신 프로비저닝 스케줄링(FVPRE: Fuzzy logic driven Virtual machine Provisioning scheduling using Resource Evaluation)을 비교분석 한다.

#### 4.2 실험

RRMA, RORMS, FVPRE의 성능을 측정하기 위해서

Table 5. Virtual Machine Performance

VM Number	Processing Performance	VM Number	Processing Performance
1	21	6	14
2	20	7	13
3	18	8	11
4	17	9	10
5	15	10	8

활용율, 평균 작업 처리시간, 작업 처리율을 측정한다.

첫 번째 실험으로, 작업 처리율(Throughput)은 가상머신의 작업 처리의 성능을 나타내는 지표로서, 가상머신에서 작업이 완료된 시점을 기준으로 평가한다. 작업 처리율은 식 (4)와 같으며 결과는 Fig. 11과 같다.

$$Throughput = \frac{Number\ Of\ Service\ Response(n)}{Finished\ Time} \quad (4)$$

시뮬레이션 시간이 600이 되었을 때, 각 대조군별 작업 처리율은 RRMA는 0.48, RORMA는 0.4883, FVPRE는 0.49로 나타났으며 FVPRE가 다른 대조군 보다 높은 성능을 보인다. 각 대조군별로 큰 차이는 없지만, FVPRE가 다른 모델보다 우세한 이유는, 다른 모델들은 예측을 기반으로 하기 때문에 그 예측의 정확도가 떨어질 수가 있기 때문이며 FVPRE는 현재의 상태에 따라 작업을 할당하기 때문이다. 또한 시뮬레이션으로는 약간의 차이를 보이지만, 실제 환경에서 작업의 요청 개수가 늘어날 경우 상대적으로 큰 차이를 보이게 된다. 따라서 FVPRE는 다른 대조군 모델 보다 작업의 처리율이 높다고 할 수 있으며, 작업 요청에 따라 원활한 서비스 제공이 가능하다.

두 번째 실험으로, 평균 작업 처리 시간(Average of Turn Around Time)은 가상머신별 요청 서비스에 대해 작업을 처리하는 평균 시간을 의미하며, 처리시간이 작을수록 빠른 서비스의 제공이 가능하다. 평균 작업 처리 시간의 공식은 식 (5)와 같으며, 결과는 Fig. 12와 같다.

$$Average\ of\ Turn\ Around\ Time = \frac{\sum_{i=1}^n Processing\ Time_i}{Number\ Of\ Service\ Response(n)} \quad (5)$$

시뮬레이션에서 시간이 600이 되었을 때, 평균 작업



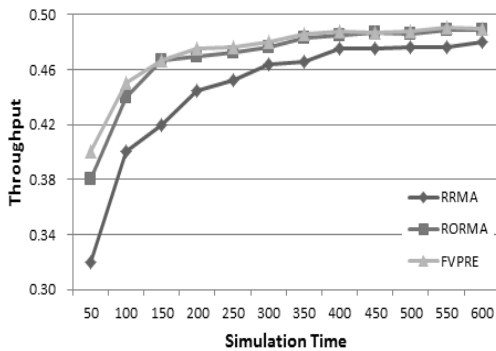


Fig. 11. Throughput Result

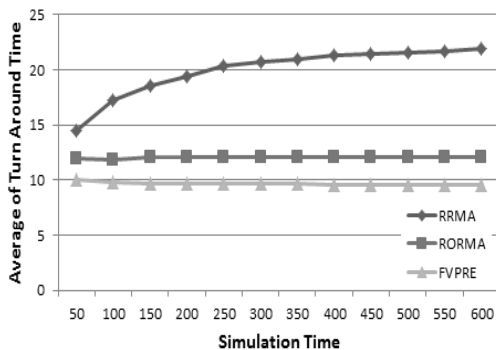


Fig. 12. Average of Turn Around Time Result

처리 시간은 RRMA는 21.879, RORMA는 12.092, FVPRE는 9.593로 나타났다. 따라서 FVPRE가 작업처리시간이 가장 짧은 것으로 나타났다. RRMA는 작업이 많아질수록 성능이 나쁜 이유는 자원의 신뢰성에만 의존하기 때문에 작업별 할당에 대해서는 고려하지 못하기 때문에 성능이 점차 감소된다. RORMA와 FVPRE는 평균작업 처리 시간이 일정한 이유는 가상머신 자원의 현재 상태를 고려하여 작업을 할당하기 때문에 일정한 간격을 유지한다. 실험 결과와 같이 FVPRE는 다른 대조군 보다 우세한 것으로 나타났기 때문에, 클라우드 환경에서의 작업 요청 시 가장 빠르게 작업을 처리해 주는 것으로 해석할 수 있다.

## 5. 결 론

클라우드 환경은 대용량의 데이터 처리나 서비스의 요청에 따라 대응할 수 있는 차세대 컴퓨팅 방법으로 주목을 받고 있다. 따라서 서비스 요청에 대해 작업을 원활히 처리할 수 있는 스케줄링 방안이 필요하다.

클라우드 컴퓨팅의 작업 처리를 위한 스케줄링을 위해 프로비저닝 기법이 필요하며, 이에 본 논문에서는 퍼지 로직 기반의 자원평가를 이용한 가상머신 프로비저닝 스케줄링(FVPRE: Fuzzy logic driven Virtual machine Provisioning scheduling using Resource Evaluation)을 제안했다. FVPRE는 각 자원들의 애매모호한 상태를 퍼지 로직을 이용하여 평가하였으며, 작업 요청 시 평가 결과에 따라 작업을 할당하게 된다.

클라우드 환경의 프로비저닝을 위해 FVPRE의 구조적인 설명과 자원관리를 위한 퍼지 로직기반의 자원 평가에 대해 설명하였다. 또한, FVPRE의 성능을 검증하기 위해 클라우드 환경을 모델링 및 시뮬레이션을 하였으며 실험 결과, FVPRE는 다른 대조군보다 높은 처리율과 낮은 작업 처리시간을 나타내어 우수한 성능을 확인하였다.

향후 연구로는 클라우드 서비스 제공에 대해서 사용자의 서비스 요청 내용이 변경될 경우, 변경된 내용에 따라 처리할 수 있는 스케줄링 기법에 대해 연구할 계획이다.

## 참고문헌

1. Assuncao, M.D. and Costanzo. (2009), "A.: Evaluating the Cost-Benefit of Using Cloud Computing to Extend the Capacity of Clusters", In: 18th ACM International Symposium on High Performance Distributed Computing, New York, pp. 141-150.
2. Bernard P. Zeigler, Herbert Praehofer, Tag Gon Kim (2000), "Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems", Academic Press, pp. 76-96.
3. Daniel Nurmi, Rich Wolski, Chris Grzegorzczak, Graziano Obertelli, Sunil Soman, Lamia Youseff and Dmitrii Zagorodnov (2009), "The Eucalyptus Open-source Cloud-computing System", Proceedings of 9th IEEE International Symposium on Cluster Computing and the Grid, pp. 124-131.
4. L. Cherkasova, R. Gardner. "Measuring CPU Overhead for I/O Processing in the Xen Virtual Machine Monitor", Proceedings of the USNIX Annual Technical Conference, April 2005
5. Lee, C.C. (1990), "Fuzzy Logic in Control Systems: Fuzzy Logic Controller", IEEE Trans. Systems, Man and Cybernetics, Vol. 20, pp. 404-435.
6. Ma, Y.B., Jang, S.H. and Lee, J.S. (2011), "Ontology-based Resource Management for Cloud Computing", The 3rd Asian Conference on Intelligent Information

- and Database Systems (ACIIDS) 2011, Daegu, Korea, pp. 343-352.
7. Park, D.H., Jang, S.H., Noh, C.H. and Lee, J.S. (2007), "Idle Resource Supplement Model and Validity Time Designation Model with Reliability Measurement in Grid Computing", Proceedings of Asia Simulation conference 2007, Seoul, South Korea, pp. 307-314.
  8. Mousumi Paul, Debabrata Samanta and Goutam Sanyal (2011), "Dynamic job Scheduling in Cloud Computing based on horizontal load", International Journal of Computer Technology and Applications, Vol. 2, Issue 5, pp. 1552-1556.
  9. Rajkumar Buyya, Rajiv Ranjan and Rodrigo N. Calheiros (2009), "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities", International Conference on High Performance Computing & Simulation(HPCS) 2009, Leipzig, Germany, pp. 1-11.
  10. Rodrigo Calheiros, Rajiv Ranjan and Rajkumar Buyya (2011), "Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments", International Conference On Parallel Processing(ICPP), Taipei, Taiwan, pp. 295-304.
  11. Stuart Russell and Peter Norvig (1995), Artificial Intelligence : A Modern Approach, PearsonEducation, pp.458-463.
  12. Shaout A. and McAuliffe P.(1998), "Job scheduling using fuzzy load balancing in distributed system", Electronics Letter, Vol.34, No.20, pp. 1983-1985.
  13. Ye Hu, Johnny Wong, Gabriel Iaszai and Marin Litoiu (2009), "Resource provisioning for cloud computing", CASCON '09 Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research, USA, New York, pp. 101-111.



**김 재 권** (jaekwonkorea@naver.com)

2011 가천의과학대학교 정보처리과 학사  
2011~현재 인하대학교 컴퓨터정보공학과 석사과정

관심분야 : 클라우드 컴퓨팅, 인공지능, 모델링&시뮬레이션



**이 종 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~현재 인하대학교 컴퓨터정보공학부 교수

관심분야 : 소프트웨어공학, 모델링&시뮬레이션