

# A Development of Adaptive VM Migration Techniques in Cloud Computing

HwaMin Lee<sup>†</sup>

## ABSTRACT

In cloud computing, server virtualization supports one or more virtual machines loaded on multiple operating systems on a single physical host server. Migration of a VM is moving the VM running on a source host to another physical machine called target host. A VM live migration is essential to support task performance optimization, energy efficiency and energy saving, fault tolerance and load balancing. In this paper, we propose open source based adaptive VM live migration technique. For this, we design VM monitoring module to decide VM live migration and open source based full-virtualization hypervisor.

**Keywords :** Cloud Computing, Hypervisor, Migration, Virtual Machine, Virtualization

## 클라우드 컴퓨팅에서 적응적 VM 마이그레이션 기법 개발

이 화 민<sup>†</sup>

## 요 약

클라우드 컴퓨팅에서 서버 가상화는 한 대의 물리적인 서버를 다수의 가상머신으로 분할하여 다양한 운영체제 및 애플리케이션을 구동하는 기술이다. 가상머신의 마이그레이션은 현재 실행 중인 가상머신을 소스 호스트에서 다른 물리적인 장치인 타겟 호스트로 이동하는 것이다. 가상머신의 라이브 마이그레이션은 작업 수행 성능의 최적화와 저전력 지원 및 에너지 절감, 결함포용, 노드들 간의 부하 균형을 제공하기 위한 필수적인 요소이다. 본 논문에서는 오픈소스 기반의 적응적 VM 라이브 마이그레이션 기법을 제안한다. 이를 위해 적응적 VM 마이그레이션 시점을 결정하는 VM 모니터링 모듈을 제안하고 오픈소스 기반 전가상화를 지원하는 하이퍼바이저를 설계하였다.

**키워드 :** 클라우드 컴퓨팅, 하이퍼바이저, 마이그레이션, 가상머신, 가상화

### 1. 서 론

클라우드 컴퓨팅은 다양한 컴퓨팅 서비스를 사용하는 장소와 디바이스에 상관없이 사용자에게 제공하는 것을 말한다. 클라우드 컴퓨팅은 가상화(virtualization)를 기본으로 하고 있으며, 특히 하이퍼바이저(hypervisor)라는 서버 가상화 기술이 많이 사용되고 있다[1]. 가상화는 물리적인 한 개의 자원을 논리적으로 분할하여 효율적으로 나눠 사용하거나, 물리적으로 다른 수 개의 자원을 논리적으로 통합하여 사용하는 기술로서 실제 사용자들이 물리적인 자원들을 논리적으로 다양하게 제어할 수 있게 해주는 기술이다[2]. 가상화

는 컴퓨팅 자원 통합을 통해 자원 활용률을 높이고, 동적인 자원 할당과 운영을 통해 시스템의 유연성을 높이는 반면 관리 비용을 낮추며, 다양한 운영체제의 활용을 가능하게 하여 가용성과 확장성을 높인다. 현재 대표적으로 사용되는 하이퍼바이저 가상화 솔루션에는 Xen[3], VMware vSphere[4], Hyper-V[5], VirtualBox[6] 등이 있다. 하이퍼바이저 서버 가상화 기술에서는 스케줄링, 라이브 마이그레이션(live migration), 결함포용(fault tolerance) 등 다양한 기법을 제공하고 있다. 특히, 라이브 마이그레이션을 이용하면 가상머신(virtual machine)을 효율적이며 유연성 있게 배치 및 운영할 수 있으며, 특히 서비스 제공 중인 애플리케이션의 작업 중단 없이 서버의 물리적인 보수와 가상머신의 이동이 가능하다. 클라우드 컴퓨팅 시스템에서 정전 등 전력 소비 문제가 발생하는 경우 또는 결함포용 이슈가 발생하는 등의 경우에서 지속적인 클라우드 컴퓨팅 서비스를 제공하기 위해 한 호스트의 VM(Virtual Machine)을 다른 호스트의 VM으로 이주시키는 VM 마이그레이션이 필요하다[7-8]. 또한

※ 이 논문은 2014학년도 순천향대학교 교수 연구년제와 2014년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2014R1A1A2057878)에 의하여 연구되었음.

† 종신회원 : 순천향대학교 컴퓨터소프트웨어공학과 부교수  
Manuscript Received : July 28, 2015  
First Revision : August 31, 2015  
Accepted : August 31, 2015

\* Corresponding Author : HwaMin Lee(leehm@sch.sc.kr)

VM이 가동(active) 중인 동안 작업량 재분배, 하드웨어 업그레이드, 소프트웨어 유지보수, 자원 관리, 가용성 향상 등의 이유로 해당 VM의 라이브 마이그레이션이 요구될 수 있다. VM이 가동 중인 경우에 사용자는 VM에서 이용하던 서비스를 끊김 없이 계속 수행될 수 있어야 하며, 이를 VM 라이브 마이그레이션이라고 한다[9-10].

클라우드 시장의 초점이 가상화 엔진인 하이퍼바이저에서 클라우드 플랫폼으로 옮겨가고 있다. 비공개 소스를 사용할 경우, 한 회사에서 모든 개발 및 오류를 수정해야 하지만, 오픈소스 플랫폼을 이용할 경우, 여러 회사 및 대학에서 소스를 수정하고 업데이트 된 소스를 공유할 수 있다. 리서치 기관인 가트너는 오픈소스 소프트웨어가 2016년에는 90%까지 확대돼 필수적인 소프트웨어 포트폴리오로 포함될 것[11]이며, 2012년에 오픈소스 클라우드를 이용한 응용 개발 마켓의 시장 규모가 90억 달러(\$)를 넘었다고 발표하였다[12]. 이에 본 논문에서는 오픈소스 기반의 적응적 라이브 VM 마이그레이션 기법을 제안한다. 이를 위해 각 오픈소스 클라우드 환경에서 사용하는 VM 가상화 및 클라우드 환경에서 지원하는 모듈을 분석하고, 분석을 통하여 통합된 다중 오픈소스 클라우드 환경에 필요한 모듈 및 기법을 연구하였

다. 그리고 적응적 VM 마이그레이션 시점을 결정하는 모니터링 모듈을 제안하고 오픈소스 기반 전가상화를 지원하는 하이퍼바이저를 설계하였다.

본 논문의 2절에서는 주요 하이퍼바이저 솔루션들에서 제공하는 라이브 마이그레이션 기법들을 소개하고, 3절과 4절에서는 본 논문에서 제안하는 VM 마이그레이션을 위한 하이퍼바이저의 구조와 VM 라이브 마이그레이션 알고리즘에 대해 기술한다. 마지막으로 4절과 5절에서는 제안 기법을 적용한 시뮬레이션 결과에 대해 분석하고 결론을 맺는다.

## 2. 관련 연구

### 2.1 하이퍼바이저

클라우드 컴퓨팅의 중요 기술 중 하나인 가상화는 성능과 기능 사이에서 최적의 균형을 제공하기 위한 방법이다. 가상화는 전가상화(full virtualization), 하드웨어 지원 가상화(hardware-assisted virtualization), 반가상화(paravirtualization)의 3가지 유형으로 나눌 수 있으며, 성능, 호환성, 기능 면에서 서로 장단점을 가지고 있다[2]. 서버 가상화 기술은 하이

Table 1. A Characteristics of Major Hypervisors

종류	특징
Xen [3]	<ul style="list-style-type: none"> <li>- 동일 컴퓨터 하드웨어에서 동시에 여러 OS를 제공하는 하이퍼바이저 시스템</li> <li>- 하드웨어 가상화를 위한 오픈소스 표준임</li> <li>- Hardware-assisted &amp; Paravirtualization</li> <li>- Full Virtualization</li> <li>- BareMetal Hypervisor</li> </ul>
VMware vSphere [4]	<ul style="list-style-type: none"> <li>- VMware ESX 서버를 이용하여 설치되는 가상화 솔루션</li> <li>- 가상 디스크의 마이그레이션을 지원하는 라이브 스토리지 마이그레이션(Storage vMotion) 기능을 제공함</li> <li>- Virtual Machine Disk Format(VMDK)이라는 가상 디스크 이미지 파일을 이용함</li> <li>- Full Virtualization</li> <li>- BareMetal Hypervisor</li> </ul>
KVM [13]	<ul style="list-style-type: none"> <li>- 가상 머신 관리자가 라이브 마이그레이션 및 KVM 기반 VM을 생성, 수정, 시작, 중지시킴</li> <li>- AMD 호스트와 Intel 호스트 간 마이그레이션이 가능함</li> <li>- Full Virtualization</li> <li>- Hosted Hypervisor</li> </ul>
Oracle VM VirtualBox [6]	<ul style="list-style-type: none"> <li>- x86 가상화 소프트웨어 패키지로 오픈소스임</li> <li>- 하이퍼바이저와 dom0(관리 도메인)로 구성</li> <li>- Hardware-assisted &amp; Paravirtualization</li> <li>- Full Virtualization</li> <li>- Hosted Hypervisor</li> </ul>
IBM Power VM Enterprise [14]	<ul style="list-style-type: none"> <li>- IBM의 POWER 프로세서에 특화된 POWER 하이퍼바이저가 서버의 하드웨어와 호스트 OS 사이에 위치하여 물리적 하드웨어 자원들과 논리적 VM들 간에 추상화 계층을 제공함</li> <li>- Full Virtualization</li> <li>- Hosted Hypervisor</li> </ul>
Microsoft Hyper-V Server 2012 [5]	<ul style="list-style-type: none"> <li>- Stand-alone hypervisor 기반 가상화 솔루션</li> <li>- 여러 VM들을 동시에 마이그레이션 할 수 있는 Simultaneous Live Migration을 지원함</li> <li>- VM에 연결된 virtual hard disk를 마이그레이션 하는 Live Storage Migration을 지원함</li> <li>- 네트워크 서브넷이 다른 호스트나 클러스터 간 마이그레이션이 가능함</li> <li>- Full Virtualization</li> <li>- BareMetal Hypervisor</li> </ul>
OpenVZ (Open Virtuozzo) [15]	<ul style="list-style-type: none"> <li>- Single Kernel이므로 모든 게스트들이 호스트가 사용하는 것과 같은 커널의 버전을 사용해야 함</li> <li>- OS-assisted Virtualization</li> <li>- Hosted Hypervisor</li> </ul>

퍼바이저 가상화, I/O 가상화, 물리적 파티셔닝(partitioning), 논리적 파티셔닝, 프로세스 파티셔닝 등으로 구성된다. 하이퍼바이저는 가상화된 자원을 생성할 수 있으며, 하나의 물리적인 서버에 서로 다른 운영체제를 가진 시스템들이 서로 아무런 영향을 끼치지 않고 다양한 애플리케이션을 수행할 수 있게 지원한다. 하이퍼바이저 가상화 기술은 베어메탈(bare-metal) 하이퍼바이저와 호스트 기반(hosted) 하이퍼바이저로 구분된다. Table 1은 현재 대표적으로 사용되는 하이퍼바이저 솔루션들의 특징을 정리한 것이다.

### 2.2 VM 마이그레이션

초기 클라우드 컴퓨팅 시스템에서는 결합포용 및 부하 균형을 위해 비실시간(non-live) VM 마이그레이션 기법을 사용하였다. 하지만 클라우드 컴퓨팅에서 운영 중인 애플리케이션이나 데이터베이스 등 서비스의 중단 없이 마이그레이션을 지원하기 위해서는 새로운 기법이 필요하게 되었고 이에 따라 라이브 마이그레이션 기술이 등장하게 되었다. VM 라이브 마이그레이션은 Fig. 1과 같이 여섯 단계로 구성된다[1].

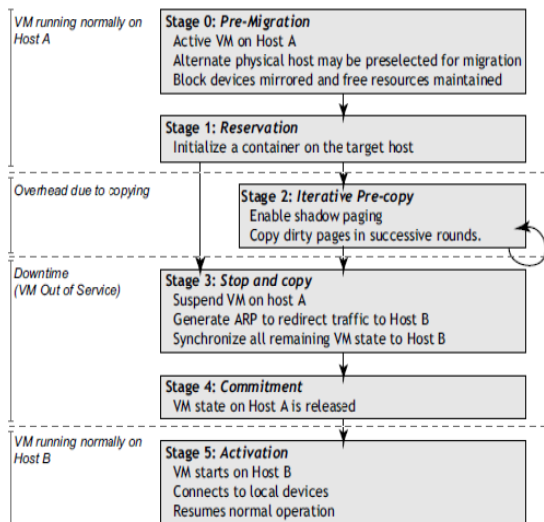


Fig. 1. Process Stages of VM Live Migration [1]

VM 라이브 마이그레이션을 위해 대부분의 솔루션들은 원래 호스트에서 수행되던 VM의 메모리 내용을 목표 호스트의 VM 메모리로 복사하는 Memory Copy 방식을 이용한다[1, 9]. 이것은 메모리의 내용을 목표 호스트의 VM으로 우선 복사하고, 아주 짧은 시간 동안 원래의 VM이 일시 정지한 후, 메모리의 변화된 부분(dirty page)만 재복사한 후, 목표 호스트에서 VM을 재시작하는 방식이다. 그러나 이것은 소프트웨어 에뮬레이션 방식으로 동작하기 때문에 속도가 느리다는 단점이 있다. 또 일부 솔루션들의 제한된 환경에서만 전가상화(Full Virtualization)가 지원되고 있다.

### 3. VM 라이브 마이그레이션을 위한 하이퍼바이저 설계

3절에서는 VM 라이브 마이그레이션 제공을 위한 오픈소스 기반 전가상화를 지원하는 하이퍼바이저를 설계한다. 이를 통해 소프트웨어와 하드웨어가 완벽히 분리되고 다른 애플리케이션들과 독립성이 보장되는 안전(secure)한 호스트를 구축할 수 있다. 전가상화 기법을 사용하여 사용자는 VM에서 수행되는 게스트 OS나 애플리케이션들이 실제 물리적 시스템에서 수행되는 것처럼 느끼게 되고, VM 간 독립성 및 보안성이 제공되며 VM 마이그레이션을 단순화시킬 수 있다. Fig. 2는 본 논문에서 제안하는 전가상화 지원 하이퍼바이저의 구조를 보여준다.

본 논문에서 제안하는 하이퍼바이저의 특징은 다음과 같다.

- 사용자 VM의 수정 없이 전가상화를 지원함
- VM의 OS 명령어 수행 시 하이퍼바이저가 오픈소스의 커널에 맞게 즉시 이진 변환함
- 한 번 변환한 OS 명령어는 다음 명령 수행 시 시간 절약을 위해 캐시에 저장해둠
- 사용자 레벨 명령어는 변환 없이 빠르게 수행함

그리고 본 논문에서는 VM을 운영하는 호스트 노드들의 모니터링 정보를 이용하여 적응적 VM 마이그레이션 시점

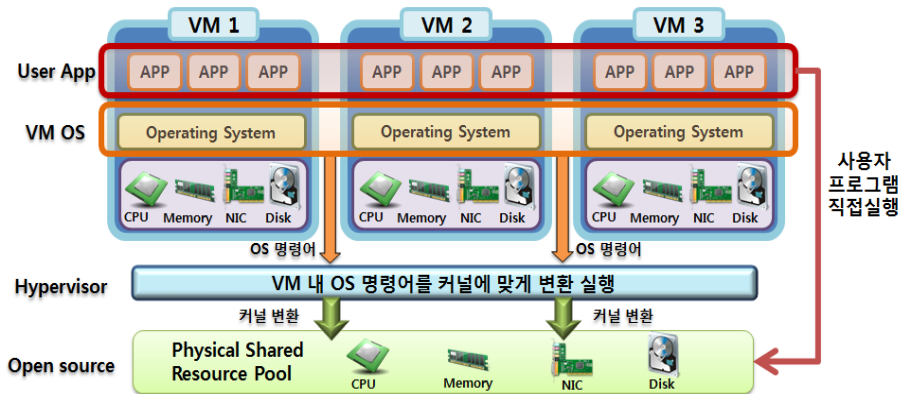


Fig. 2. An Architecture of Proposed Hypervisor Architecture

을 결정하는 모니터링 모듈을 제안한다. 클라우드 플랫폼에서 호스트의 결함, 성능, 부하 등을 해결하기 위해서는 상황에 따라 VM 마이그레이션을 선택적으로 결정하는 적응적 VM 마이그레이션이 요구된다. 적응적 VM 마이그레이션 시점을 발견하기 위해서는 각 VM 및 호스트의 상태의 모니터링이 선행되어야 한다. 이에 본 논문에서는 적응적 VM 마이그레이션을 위한 시점을 결정하기 위해 각 VM의 상태, 네트워크 상태, 프로세서 및 메모리 상태, 에너지 소비 정도 등의 정보를 수집하는 모니터링 모듈을 설계하였다. 또한 다중 클라우드 플랫폼에서 각 플랫폼 호스트의 성능에 따른 VM 마이그레이션 시점을 결정하기 위해 호스트 과부하, 성능 향상을 위한 로드 밸런싱에 관한 임계값을 결정하고 사용자 요구 및 SLA, 호스트 고장에 따른 호스트 이전에 관한 정책을 연구하였다. Fig. 3은 라이브 마이그레이션을 위해 VM들의 정보를 모니터링하고 모니터링 된 정보를 수집하여 마이그레이션 의사 결정을 수행하는 제어 흐름을 보여준다.



Fig. 3. Process of VM Information Monitoring for VM Live Migration Decision

### 4. VM 라이브 마이그레이션 알고리즘

본 논문에서 제안하는 VM 라이브 마이그레이션 알고리즘에서 사용하기 위해 정의한 자료구조는 Table 2와 같다.

Table 2. VM 라이브 마이그레이션 알고리즘의 자료구조

$V_{mem}$ (MB)	VM의 메모리 크기
$V_{mig}$ (MB)	라이브 마이그레이션 해야 하는 VM의 총 전송 작업량
$V_{th}$ (MB)	라이브 마이그레이션 수행을 결정하기 위한 임계치
$N$ (MB/s)	라이브 마이그레이션에 사용할 수 있는 네트워크 대역폭
$D$ (MB/s)	라이브 마이그레이션 동안 발생하는 dirty page 크기
$W$ (MB)	Writable Working Set 크기
$T_{mig}$ (s)	VM 라이브 마이그레이션에 소요되는 총 시간
$T_{down}$ (s)	라이브 마이그레이션으로 인한 작업 중지(downtime) 총 시간
$T_{resume}$ (s)	라이브 마이그레이션 후 작업 재시간(resume)에 소요되는 시간

알고리즘 1은 VM 라이브 마이그레이션 수행을 결정하기 위해  $V_{mig}(s)$ 와  $T_{mig}(s)$ 을 계산하는 알고리즘이다.

### 알고리즘 1. $V_{mig}(s)$ 와 $T_{mig}(s)$ 계산 알고리즘

```

1: let  $V0 \leftarrow Vmem$ 
2: for  $i = 0$  to  $max$  do
3:    $Ti \leftarrow Vi / N$ 
4:    $\delta \leftarrow aTi + \beta D + \gamma$ 
5:    $Wi+1 \leftarrow \delta Ti D$ 
6:    $Vi+1 \leftarrow Ti D - Wi+1$ 
7:   if  $Vi+1 \leq Vthd$  or  $Vi+1 > Vi$  then
8:      $Vi+1 \leftarrow Ti D$ 
9:      $Ti+1 \leftarrow Vi+1 / N$ 
10:     $Tdown \leftarrow Ti+1 + Tresume$ 
11:    break
12:  end if
13: end for
14:  $Vmig \leftarrow \sum_{i=0}^{max} Vi$ 
15:  $Tmig \leftarrow \sum_{i=0}^{max} Ti$ 

```

### 알고리즘 2. VM 라이브 마이그레이션 알고리즘

```

1: if Event then
2:   if Event = hotspot then
3:     for all physical hosts(PHs) do
4:       Send information about resource usage and
         information about all VMs
5:     end for
6:     while bids < PHs - 1 and elapsedTime < maxTime
         do
7:       Receive bids
8:     end while
9:     for all bids do
10:      Calculate the utility for all VMCandidates and
        append to utilityArray
11:    end for
12:    for all utilities in utilityArray do
13:      Calculate migration cost
14:    end for
15:    Sort utilityArray based on utility
16:    Remove candidates that have a high cost
17:    Set selectedHost to the highest value in utilityArray
18:    for all physical machines do
19:      Send information about selectedHost
20:    end for
21:    Listen for acknowledge message from the selectedHost
22:    if no or negative response then
23:      Run redo migration algorithm
24:    else
25:      Run migration and exit
26:    end if
27:  end if
28: end if

```

알고리즘 2는 모니터링 모듈에서 VM들에 대한 모니터링 결과 특정 VM에서 작업량이나 에너지 소비량이 일정 수준 이상 증가하는 경우, VM 마이그레이션을 결정하고 수행하는 알고리즘이다.

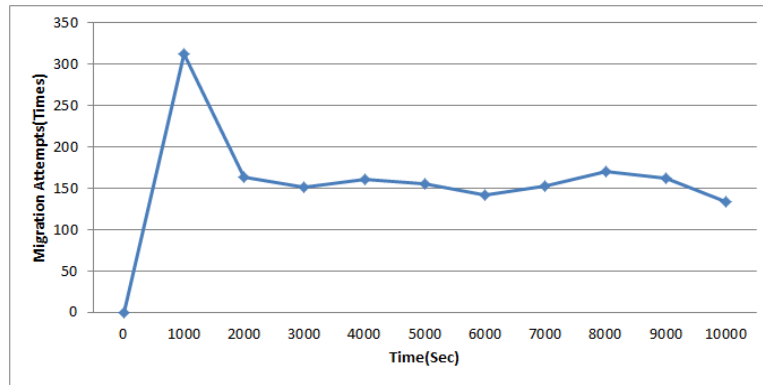


Fig. 4. VM Migration Attempts According to Process Time

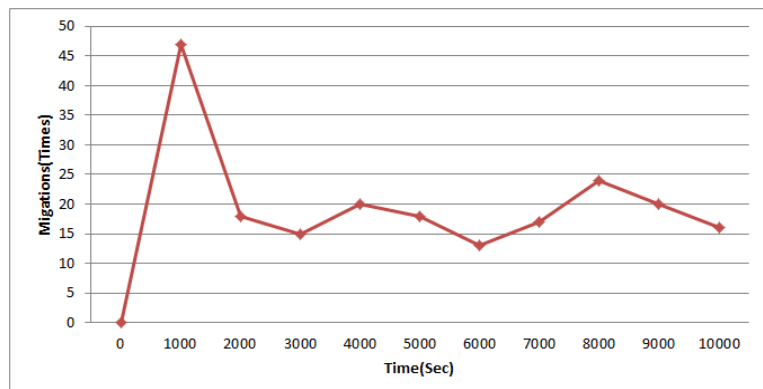


Fig. 5. The Number of VM Migration According to Process Time

## 5. 실험

본 논문에서 제안한 VM 라이브 마이그레이션 알고리즘의 성능을 검증하기 위해 CloudSim[16] 툴킷을 이용하여 시뮬레이션을 실시하였다. 실험을 위해 자바를 이용하여 시뮬레이터를 개발하였으며, 시뮬레이터는 인텔 i7-4790K 쿼드 코어(8 스레드), 8GB 메모리의 Ubuntu 14.04 환경에서 실행하였다. 시뮬레이션을 위해 본 논문에서는 하나의 클라우드 데이터센터 내에 Table 3과 같은 사양을 갖는 5개의 호스트 노드가 존재하고 10개~100개의 VM이 생성이 되어 작업을 수행한다고 가정하였다. 각각의 VM은 초기 작업량을 무작위로 할당하였고, VM의 메모리 크기는 128MB, 256MB, 512MB, 768MB, 1024MB 중 할당되도록 하였다. 또한 각 VM의 dirty page 비율은 초당 5000, 10000, 15000, 20000, ..., 50000 페이지를 가지도록 설정하였다.

Table 3. Node Specifications for Simulation

Host ID	Core	CPU	Memory	Network
1	4	3.6GHz	8GB	1Gbps
2	8	3GHz	16GB	1Gbps
3	4	3.5GHz	8GB	1Gbps
4	6	3.6GHz	8GB	1Gbps
5	8	2.4GHz	16GB	1Gbps

Fig. 4와 Fig. 5는 클라우드 컴퓨팅 시스템에서 실행 시간 흐름에 따른 VM 라이브 마이그레이션 시도 횟수와 본 논문에서 제안한 알고리즘을 통해 실제 VM 라이브 마이그레이션이 발생하는 횟수를 나타낸다. 실험 결과 실행 초기에 많은 VM 라이브 마이그레이션 시도가 발생하였고 실제 실행되는 횟수도 많았다. 그리고 VM 라이브 마이그레이션 시도가 많다고 해서 실제 마이그레이션의 실행으로 이어지지 않는다는 것과 실제 마이그레이션 실행 비율은 대략 10% 정도임을 확인할 수 있다.

Fig. 6은 실행 시간 흐름에 따른 클라우드 컴퓨팅 시스템의 성능을 보여준다. 실험 결과 VM 마이그레이션 수행을 통해 시스템 시작 초기를 제외하고 일정 수준의 시스템 성능이 유지됨을 보여준다. 본 논문에서 제안한 VM 라이브 마이그레이션을 이용하는 경우 평균 0.9984 정도의 시스템 성능을 나타냈다.

## 6. 결론

클라우드 컴퓨팅 시스템에서 최적의 작업 수행 성능 보장, 저전력 컴퓨팅 지원, 결합포용 서비스 제공 등을 위해 애플리케이션 실행의 중단 없이 한 호스트의 VM(Virtual Machine)을 다른 호스트의 VM으로 이주시키는 VM 라이브 마이그레이션이 필요하다. 이에 본 논문에서는 오픈소스 클

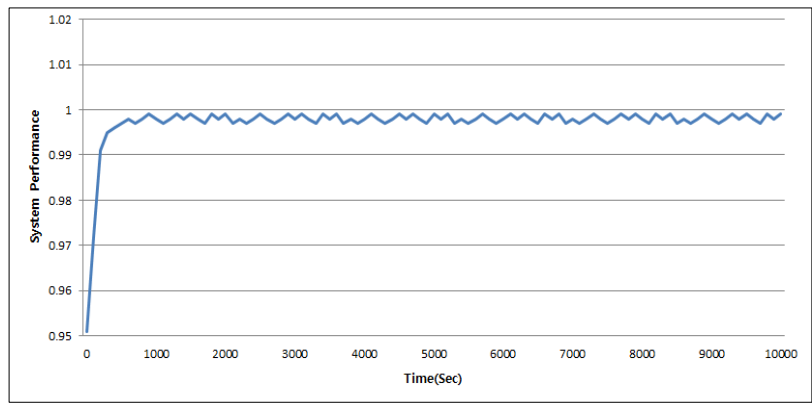


Fig. 6. Performance of Cloud Computing System According to Process Time

라우드 환경에서 사용하는 VM 가상화 및 클라우드 환경에서 지원하는 모듈을 분석하고, 다중 오픈소스 클라우드 환경에서 VM 라이브 마이그레이션을 지원하는 하이퍼바이저를 제안하였다. 이를 위해 적응적 VM 마이그레이션 시점을 결정하는 모니터링 모듈을 설계하고 오픈소스 기반의 VM 라이브 마이그레이션 알고리즘을 제안하였다. 향후 연구를 통해 OpenStack 환경에서 제안하는 하이퍼바이저와 VM 라이브 마이그레이션 알고리즘을 구현하고자 한다.

References

[1] Patel, Pradip D., et al, "Live Virtual Machine Migration Techniques in Cloud Computing: A Survey," *International Journal of Computer Applications*, Vol.86, No.16, 2014.

[2] Marshall, Understanding Full Virtualization, Paravirtualization, and Hardware Assist, VMware white paper, 2007.

[3] Abels, Dhawan, et al, An Overview of Xen Virtualization, Dell Power Solutions. pp.109-111, 2005.

[4] Guthrie, Forbes, Scott Lowe, and Kendrick Coleman, "VMware vSphere design," John Wiley & Sons, 2013.

[5] Velte, Anthony, et al, Microsoft virtualization with Hyper-V. McGraw-Hill, Inc., 2009.

[6] Romero and Alfonso, "VirtualBox 3.1: Beginner's Guide," Packt Publishing Ltd, 2010.

[7] Svård, Petter, et al, "Principles and performance characteristics of algorithms for live VM migration," *ACM SIGOPS Operating Systems Review*, Vol.49, No.1 pp.142-155, 2015.

[8] Akiyama, Soramichi, Takahiro Hirofuchi, and Shinichi Honiden, "Evaluating Impact of Live Migration on Data Center Energy Saving," *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on IEEE*, 2014.

[9] Xu, Fei, et al, "iaware: Making live migration of virtual machines interference-aware in the cloud," *IEEE Transactions on Computers*, Vol63, No.12, pp.3012-3025, 2014.

[10] Raju, R., et al, "Interpretation and evaluation of various hybrid energy aware technologies in cloud computing environment-A detailed survey," *Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on IEEE*, 2014.

[11] [Internet] <http://www.gartner.com/newsroom/id/1541414>.

[12] [Internet] <http://www.gartner.com/newsroom/id/2131115>.

[13] "RedHat KVM : Kernel-based Virtual Machine," Red Hat, 2009, [Internet] <http://www.redhat.com/f/pdf/rhev/DOC-KVM.pdf>.

[14] Hales, et al. "PowerVM virtualization on IBM system p: introduction and configuration," IBM Redbook, 2008.

[15] Kolyshkin, "Virtualization in linux. openvzintro. pdf. Documentation on OpenVZ," 2009.

[16] Calheiros, Rodrigo, et al., "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, Vol.41, No.1, pp.23-50, 2011.



이 화 민

e-mail : leehm@sch.ac.kr  
 2000년 고려대학교 컴퓨터교육과(학사)  
 2002년 고려대학교 컴퓨터교육학과(석사)  
 2006년 고려대학교 컴퓨터교육학과(박사)  
 2006년~2007년 특허청 전기전자심사본부  
 통신사무관  
 2007년~현 재 순천향대학교 컴퓨터소프트웨어공학과 부교수  
 관심분야: 클라우드컴퓨팅, 모바일컴퓨팅, 웰니스, IoT, 컴퓨터교육