

## 애플리케이션 특성을 반영한 서버 가상화 기술 적용방안

김현정<sup>o</sup>, 이상길\*, 이철훈\*

<sup>o</sup>국가정보자원관리원,

\*충남대학교 컴퓨터공학과

e-mail: acceler@korea.kr<sup>o</sup>, {sk0137, cleec}@cnu.ac.kr\*

## A Study for Applying for the Server Virtualization Technology based on Application Characteristics

Hyeon-Jeong Kim<sup>o</sup>, Sang-Gil Lee\*, Cheol-Hoon Lee\*

<sup>o</sup>National Information Resources Service,

\*Dept. of Computer Engineering, Chungnam National University

### ● 요약 ●

서버 가상화 기술은 초기 하이퍼바이저 방식에서 비즈니스 민첩성을 높일 수 있는 컨테이너 기술로 진화하고 있다. 하지만, 컨테이너 기술은 운영체제를 공유하고 잦은 빌드와 배포로 보안과 안정성에 대한 문제가 제기되고 있다. 이에 따라 본 논문에서는 서버 가상화 기술인 하이퍼바이저와 컨테이너 기술을 비교분석하고 애플리케이션 특성을 분석한다. 하이퍼바이저 기술은 하드웨어 가상화를 통해 안정성이 높은 반면 복잡하고 무거우며 속도가 느린 단점이 있다. 컨테이너 기술은 하이퍼바이저에 비해 가볍고 성능이 향상되는 반면 보안 및 안정성에 문제가 발생할 수 있다는 단점이 있다. 이를 통해 미션 크리티컬 워크로드를 가진 애플리케이션은 안정성이 우수한 하이퍼바이저 기술이 적합하고, 자원 사용이 가변적인 애플리케이션은 서버 확장이 유연하고 성능이 우수한 컨테이너 기술이 적합하다고 제안한다.

**키워드:** 서버 가상화(server virtualization), 하이퍼바이저(hypervisor), 컨테이너(container)

## I. Introduction

현재 4차 산업혁명으로 전세계는 그간 경험하지 못한 변화를 겪고 있다. 인공지능이 업무를 처리하고, 자율주행자가 운전자의 조작없이 스스로 목적지까지 찾아간다. 이와 같은 변화는 인공지능, 사물인터넷(IoT), 빅데이터 등 지능정보기술이 기존 산업과 융합되어 우리 사회를 변화시키고 있는 것이다. 이 변화의 바탕에는 새로운 서비스와 융복합 서비스를 가능하게 하는 핵심인프라로서 클라우드 컴퓨팅이 있다.

클라우드 컴퓨팅은 네트워크를 통해 IT 서비스를 제공하는 컴퓨팅 환경을 말한다. 이것은 2000년대 중반부터 발전하기 시작한 가상화 기술에 기반하고 있다. 가상화는 한 개의 자원을 여러개 논리적 자원으로 분할하여 사용할 수 있도록 하는 개념이다. 이러한 개념을 서버에 적용하여 하나의 물리적 서버에 여러 개의 가상머신을 올려 사용할 수 있도록 한 것이 서버 가상화이다. 초기의 가상화는 물리적 하드웨어 자원을 가상화한 하이퍼바이저 방식을 사용하였다. 하지만 가상머신마다 별도의 게스트 운영체제를 설치하여 복잡하고 무겁다는 지적이 있었다. 이를 개선하고자 운영체제를 공유할 수 있는 컨테이너 기술이 각광을 받게 되었다. 하지만 컨테이너 기술은 운영체제를 공유하고 잦은 빌드와 배포로 인해 보안과 안정성에 대한 문제가 제기되었다.

따라서, 본 논문에서는 가상화 기술의 특징과 애플리케이션 특성을 분석하여 애플리케이션에 적합한 가상화 기술을 제안하고자 한다.

## II. Preliminaries

### 1. 하이퍼바이저 기술 개요

하이퍼바이저 기술은 CPU, 메모리 등 물리적인 하드웨어 자원을 가상화하고 논리적으로 분할하여 가상머신을 제공하는 기술이다. 하이퍼바이저 가상화의 아키텍처는 Fig. 1과 같다. 호스트 운영체제와 하이퍼바이저를 설치하고 가상머신을 올린 후 가상머신마다 게스트 운영체제와 응용프로그램을 설치하는 구조이다.



Fig. 1. 하이퍼바이저 가상화 아키텍처

이와 같은 하이퍼바이저에는 어떤 방식으로 설치하느냐에 따라 Native와 Hosted 방식으로 나뉜다. Native(베어메탈) 방식은 호스트 운영체제 없이 하이퍼바이저를 하드웨어에 직접 설치하는 방식이고 Hosted 방식은 하이퍼바이저를 호스트 운영체제 위에 설치하는 방식이다[1][3]. 그리고, 가상화를 어디까지 지원하느냐에 따라 전가상화와 반가상화로 나뉜다. 전가상화 방식은 전체 하드웨어를 가상화하는 방법으로 하이퍼바이저를 통해서 물리자원 접근이 가능한 방식이며, 전가상화의 물리자원 요청과정에서 성능을 높인 방식이 반가상화 방식이다[2][3]. 이와 같은 하이퍼바이저의 대표적인 예는 KVM, Xen, Hyper-V, ESXi 등이 있다.

## 2. 컨테이너 기술 개요

컨테이너 기술은 운영체제의 커널이 프로세스를 격리하여 가상 자원인 컨테이너를 제공하는 기술이다. 컨테이너 기술의 아키텍처는 Fig. 2와 같다. 먼저, 호스트 운영체제와 컨테이너 엔진을 설치하고 그 위에 애플리케이션이 포함된 컨테이너 이미지를 배포하여 실행하는 구조이다.



Fig. 2. 컨테이너 가상화 아키텍처

이와 같은 프로세스 격리를 제공하기 위해 리눅스의 namespace와 cgroup을 이용한다. cgroup을 이용해 CPU, 메모리 등의 자원을 할당하고, 프로세스, 파일시스템 등의 환경을 namespace로 격리하여 컨테이너를 제공한다[2][3]. 이와 같은 컨테이너 기술의 대표적인 예는 리눅스 컨테이너(LXC), 도커(Docker), 클라우드 파운드리(Cloud Foundry), cri-o 등이 있다.

## 3. 애플리케이션 특성

애플리케이션마다 자원을 요구하는 사항이 다양하게 존재한다. 안정적인 서비스가 중요한 경우가 있고 안정적인 서비스보다는 신속한

서비스가 중요한 경우도 있다. 이와 같은 애플리케이션의 다양한 요구사항을 만족시키기 위해서는 애플리케이션 마다의 다른 요구사항을 파악하고 이에 맞는 자원의 적절한 배분이 이루어져야 한다.

가장 먼저 고려해야 할 애플리케이션으로 미션 크리티컬한 워크로드를 가진 애플리케이션이 있다. 이 애플리케이션은 장애 발생 시 파급효과가 크기 때문에 안정적인 서비스 운영이 전제되어야 한다. 또한 보안사고 발생 시 막대한 피해가 예상되므로 보안이슈가 더 중요하게 다루어져야 한다. 온라인 banking, 전사적자원관리(ERP), 고객관계관리(CRM), 그룹웨어 등 기관의 핵심업무가 대표적이다.

다음으로 고려가 필요한 애플리케이션으로 자원 사용이 가변적 애플리케이션이 있다. 이와 같은 애플리케이션은 접속자 폭주가 언제 발생할지 몰라 자원 사용 예측이 불가능한 서비스로 서버 확장 등과 같은 자원 사용의 유연성이 제공되어야 한다. 업무 집중기간이 있는 서비스나 대민서비스 같은 일시적인 사용량의 변동 가능성이 높은 서비스들이 대표적이다.

Table 1. 애플리케이션(AP)별 특징 및 사례

구분	미션 크리티컬 AP	가변적 자원사용 AP
특징	·사업이나 조직의 생존에 필수적인 시스템 ·보안과 안정성 중요	·시장과 환경변화에 따라 자원사용이 유동적인 시스템 ·비즈니스 민첩성 중요
사례	·온라인 banking, 항공기 운영시스템, 전사적 자원관리(ERP), 그룹웨어 등	·기관대표홈페이지, 인구총조사시스템, 지원금 신청서비스, 재난안전서비스 등

## III. The Proposed Scheme

서버 가상화 기술인 하이퍼바이저 기술과 컨테이너 기술을 비교분석함으로써 애플리케이션 특성에 적합한 기술을 제안하고자 한다.

하이퍼바이저 기술은 호스트 운영체제, 하이퍼바이저를 설치하고 그 위에 게스트 운영체제를 설치하여 가상머신을 제공한다. 이로 인해 가상머신이 다른 가상머신과 완전히 격리되어 게스트 운영체제의 선택이 가능해지고 보안성이 높아진다. 또한 장애가 발생해도 전체 시스템에 영향을 주지 않도록 가상머신 내에서 차단되어 안정성이 우수하다. 가상화 환경이 구현됨으로써 자동화 및 중앙화가 되어 운영관리가 용이하다. 그러나, 가상머신마다 게스트 운영체제가 설치되어 복잡하고 시스템이 무거워지는 단점이 있다.[2] 애플리케이션을 레거시에서 하이퍼바이저 기반 가상화로 전환시에는 컨테이너 방식보다 용이하게 전환할 수 있다.

컨테이너 기술은 리눅스 커널에서 제공하는 기능을 이용하여 프로세스를 격리한다. 이로 인해 필요한 소프트웨어만 배포하여 가볍고 성능이 향상되는 효과가 있다. 프로세스 형태로 실행되므로 실행환경에 종속 없이 빠르고 일관되게 배포가 가능하다. 또한, 자원 확장과 같은 요구사항에 신속하게 대응할 수 있는 신속성과 유연성을 가지고 있다. 쿠버네티스와 같은 오케스트레이션 시스템을 통해 관리와 운영이 용이하다[2]. 그러나 운영체제를 공유함으로써 격리 실패와 같은 보안 문제가 발생할 가능성이 있다. 또한 아직 확산단계에서 확인되지

않은 보안 위협들이 존재한다[4]. 애플리케이션을 기존 레거시에서 컨테이너 기반 가상화 환경으로 전환시에는 하이퍼바이저 방식에 비해 어렵다. 이것은 컨테이너를 쉽고 빠르게 배포하기 위해 애플리케이션을 마이크로서비스 아키텍처(MSA)로 전환해야 하기 때문이다. Table 2는 하이퍼바이저와 컨테이너 기술에 대한 비교를 보여준다.

Table 2. 하이퍼바이저와 컨테이너 비교

구분	하이퍼바이저	컨테이너
개념	·하드웨어를 가상화하여 가상머신 제공	·운영체제가 프로세스를 격리하여 컨테이너 제공
구성 방식	·호스트운영체제, 하이퍼바이저, 게스트운영체제로 구성	·호스트운영체제, 컨테이너 엔진, 컨테이너로 구성
장점	·게스트운영체제 선택 ·보안·안전성 우수 ·자동화 및 중앙화로 운영 관리 용이 ·컨테이너에 비해 애플리케이션 전환 용이	·소프트웨어만 배포하여 가볍고 성능 향상 ·신속성과 유연성 ·오케스트레이션시스템을 통해 운영 관리 용이
단점	·복잡하고 무거우며 속도가 느림	·보안, 안정성 문제 존재 ·애플리케이션 전환 어려움

하이퍼바이저와 컨테이너 기술을 비교한 결과, 각자 장단점이 있고 애플리케이션 요구사항에 맞는 가상화 기술이 필요하다는 점을 알 수 있다. 이를 토대로 애플리케이션에 적합한 가상화 기술을 제안하고자 한다.

먼저, 미션 크리티컬한 워크로드를 가진 애플리케이션은 사업 혹은 조직의 생존에 필수적인 시스템으로 장애와 보안사고에서 안전하다고 검증된 기술의 채택이 필요하다. 따라서, 보안과 안정성이 우수한 하이퍼바이저 기술이 적합하다.

둘째, 자원 사용이 가변적인 애플리케이션은 사용량에 따라 자원을 자유롭게 할당 회수가 필요하다. 따라서, 예측하지 못한 시스템 부하가 발생해도 신속하게 부하를 해소할 수 있도록 자원 확장이 가능한 컨테이너 기술이 적합하다.

#### IV. Conclusions

서버 가상화 기술을 적용하기 위해 하이퍼바이저, 컨테이너 기술을 비교분석하였다. 먼저, 하이퍼바이저 기술은 분리와 격리를 통해 보안 및 안전성 강화, 자동화 및 중앙화를 통한 운영관리가 용이하다는 특성이 있다. 컨테이너 기술은 프로세스를 격리하는 방식을 통해 가볍고 성능을 향상시키며 가변적인 자원사용에 신속하고 유연하게 대응할 수 있다.

이와 같은 가상화 기술의 특성을 고려하여 미션 크리티컬한 워크로드를 가진 애플리케이션은 보안과 안정성이 우수한 하이퍼바이저 기술이 적합하고 자원 사용이 가변적인 애플리케이션은 가볍고 성능이 우수한 컨테이너 기술이 적합하다.

향후에는 하이퍼바이저와 컨테이너 기술들이 현재의 장점을 살리면서 단점을 해소하는 방향으로 더욱 발전할 것이며, 컴퓨팅 자원들의

활용율을 극대화하고 성능도 훨씬 향상될 것이다. 이와 같은 클라우드 컴퓨팅 핵심 기술의 발전이 사물인터넷(IoT), 엣지컴퓨팅, 인공지능, 빅데이터 등의 새로운 서비스와 융복합 서비스를 견인할 것으로 기대된다.

#### REFERENCES

[1] Sungkyun Lee, "Cost reduction from virtual server integration," Department of information and telecommunications, Konkuk University, 2018.

[2] Taehyun Kim, Taeyoung Kim, youngeun Choi, mihee Choi, sungkun Jin, "Virtualization and Kubenetes", OSIA Standards & Technology Review, 33:2, 4-10, Nov. 2020.

[3] "The Evolution of Server Virtualization: VM and Container", Gabia Library, <https://library.gabia.com/contents/infrahosting/7426/>

[4] Daekyun Yoon, "Cloud Native Computing Diffusion and Security Issue", Cloudstore CEART Issue Report, Nov. 2019.