

## 멀티코어 기반 어플리케이션 운용을 위한 데스크탑 가상화 구성 및 성능 분석

박준용\*

### VDI deployment and performance analysis for multi-core-based applications

Junyong Park\*

\*Engineer, Infra SW Team, Hanwha Systems, Seongnam, 13524 Korea

#### 요약

최근 VDI(Virtual Desktop Infrastructure)가 오피스 업무 환경뿐만 아니라 고사양의 멀티코어 기반 어플리케이션을 사용하는 워크로드에도 많이 사용되면서 VDI에 대한 실시간성과 안정성에 대한 요구 사항이 증대되고 있다. 그에 따라 VDI에서 원격접속에 사용되는 디스플레이 프로토콜과 가상머신의 성능 최적화 또한 중요성이 높아졌다. 본 논문에서는 멀티코어 기반 어플리케이션 운용을 위한 데스크탑 가상화 구성을 위해 두 가지를 제안한다. 첫 번째는 멀티 프로세싱으로 인한 고부하 상황에서 최적의 성능을 가진 디스플레이 프로토콜의 코덱 구성을 제안한다. 두 번째는 가상머신 간 CPU 경합 시 스케줄링 지연을 줄이기 위한 가상 CPU 스케줄링 최적화 방안을 제안한다. 시험 결과 Blast Extreme의 H.264 코덱이 가장 좋고 안정적인 프레임이 보여졌으며 스케줄링 최적화를 통해 가상 CPU의 스케줄링 성능이 개선됨을 확인하였다.

#### ABSTRACT

Recently, as Virtual Desktop Infrastructure(VDI) is widely used not only in office work environments but also in workloads that use high-spec multi-core-based applications, the requirements for real-time and stability of VDI are increasing. Accordingly, the display protocol used for remote access in VDI and performance optimization of virtual machines have also become more important. In this paper, we propose two ways to configure desktop virtualization for multi-core-based application operation. First, we propose a codec configuration of a display protocol with optimal performance in a high load situation due to multi-processing. Second, we propose a virtual CPU scheduling optimization method to reduce scheduling delay in case of CPU contention between virtual machines. As a result of the test, it was confirmed that the H.264 codec of Blast Extreme showed the best and stable frame, and the scheduling performance of the virtual CPU was improved through scheduling optimization.

**키워드**: 데스크탑 가상화, 불균일 기억 장치 접근, 가상 CPU, 가상머신, 원격접속 프로토콜

**Keywords**: desktop virtualization, non uniform memory access, VCPU, virtual machine, display protocol

Received 31 August 2022, Revised 8 September 2022, Accepted 13 September 2022

\* Corresponding Author Jun-Yong Park(E-mail: jy104.park@hanwha.com, Tel:+82-31-8091-3777)  
Engineer, Infra SW Team, Hanwha Systems, Seongnam, 13524 Korea

Open Access <http://doi.org/10.6109/jkiice.2022.26.10.1432>

print ISSN: 2234-4772 online ISSN: 2288-4165

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서 론

가상화 기술이 보편화되면서 실시간성과 안정성이 요구되는 워크로드 또한 데스크탑 가상화 기술이 도입되었다. 오피스 업무 용도의 가상화 환경에서는 수용될 만한 성능 저하도 고성능을 요구하는 어플리케이션 환경에서는 수용 불가능한 문제가 될 가능성이 있다. 가상화 기술은 광범위한 워크로드를 지원하기 위해 CPU 주기를 줄이고 전력을 절약하여 높은 I/O 처리를 효율적으로 구현하도록 발전해왔다. 그러나 많은 응용 프로그램에서는 CPU 사용률이 높아지고 전력 소비가 많더라도 I/O 지연 시간을 최소화하고 성능을 보장할 것을 요구한다. 이에 따라 사용하는 워크로드에 최적화된 가상화 환경을 구성하는 것 또한 중요해졌다[1, 2].

데스크탑 가상화에서 사용자에게 쾌적한 환경을 제공하기 위해선 가상머신의 성능을 보장하도록 구성하고 워크로드에 적합한 디스플레이 프로토콜을 선정하는 것이 중요하다. 특히 멀티코어 기반 어플리케이션이 구동되는 환경에선 가상머신의 가상 CPU(VCPU) 구성이 중요한 고려 요소다. CPU는 서버가 가지고 있는 유한한 자원이고 각 가상머신의 게스트 OS가 가장 동적으로 요구하는 자원이기 때문이다. 서버 관리자는 VCPU를 과도하게 할당하는 것을 피해야 한다. 과도한 VCPU 할당은 서버 리소스가 낭비되고 가상머신 간에 CPU 경합을 심화시켜 성능을 보장할 수 없기 때문이다.

고사양을 요구하는 워크로드에 데스크탑 가상화를 적용하기 위한 연구는 꾸준히 진행되어 왔다. [3]에서는 가상화 솔루션인 VMware, Xen, Hyper-V, KVM에서 가상머신의 벤치마크 성능을 비교 및 분석하였다. 분석 결과 VMware 하이퍼바이저의 가상머신이 타 가상화 솔루션에 비해 우수한 성능을 보였다. [4]에서는 기존 연구 결과를 바탕으로 데스크탑 가상화의 성능을 분석하기 위해 대표적인 멀티코어 기반 시스템인 함정전투체계를 운용하는 환경에서 디스플레이 프로토콜별 프레임 비교 시험을 진행하였고 VMware 데스크탑 가상화 솔루션인 Horizon View 환경에서 Blast Extreme 프로토콜이 가장 높은 성능을 보여주었다. 기존 연구들은 디스플레이 프로토콜의 코덱별 성능 차이와 실 운용 시 리소스 공유에 의해 발생 가능한 가상머신의 CPU 경합을 고려하지 않았다.

본 논문에서는 가상머신의 CPU 경합을 줄이기 위한

가상 CPU 스케줄링 최적화 방안과 Blast Extreme 프로토콜 중에서도 코덱별 성능 비교 분석을 통해 멀티코어 기반 어플리케이션 운용에 적합한 데스크탑 가상화 구성을 선정하는 것을 목적으로 한다. 제안된 데스크탑 가상화 구성은 향후 성능에 민감한 워크로드의 데스크탑 가상화 도입 시 디스플레이 프로토콜 구성 및 성능 최적화에 참고 할 수 있다.

2장에서는 가상화 환경에서 추가적으로 고려해야 할 성능 메트릭을 배경 지식으로 다루며 3장에서는 VMware 솔루션의 데스크탑 가상화에 사용되는 Blast Extreme 프로토콜의 코덱 별 특성과 가상머신 스케줄링 최적화 방안에 관한 연구를 다룬다. 4장에서는 3장에서 연구한 내용을 바탕으로 Blast Extreme의 코덱별 성능 비교 시험을 진행하여 결과를 비교 분석하고 가상머신의 스케줄링 최적화 후 성능 메트릭 변화를 확인한다. 마지막 5장에서는 본 논문의 결론 및 향후 연구에 대해 다룬다.

## II. 배경 지식

### 2.1. CPU Ready

가상머신의 VCPU는 각 가상머신에 고유하게 할당되고 지속 점유하는 형태가 아닌 하이퍼바이저의 스케줄러에 의해 물리적 CPU를 타임 슬라이스 방식으로 할당되는 특징을 갖는다. 정지 상태의 VCPU는 실행 상태가 되기 전까지 어떠한 이벤트도 처리할 수 없다. 실행 상태의 VCPU에 전달된 이벤트는 지연 없이 바로 처리되지만 정지 상태의 VCPU에 전달된 이벤트는 VCPU가 실행 상태가 될 때까지 지연된다. 이러한 이벤트 지연 현상은 I/O 성능을 저하시키는 원인이 된다[5].

CPU에 의한 성능 저하가 의심되면 일반적으로 CPU 사용률을 참고한다. CPU 사용률이 높아서 성능이 저하된다면 유저는 서버 관리자에게 추가적인 VCPU 할당을 요구할 것이고, 서버 관리자는 서버의 가용한 리소스를 확인한 후 가능하다면 해당 가상머신에 VCPU를 추가적으로 할당할 것이다. 그러나 가상화 환경이라는 특수한 환경 때문에 CPU 사용률이 안정적이더라도 성능 저하가 발생할 수 있다. 가상머신은 자신이 이용하는 VCPU가 언제나 실행 상태에 있다고 간주하여 스케줄러에 의한 지연을 알아채지 못한다. 따라서 ESXi와 같

은 가상화 솔루션은 VCPU 사용률 이외에 CPU Ready 값을 모니터링 하도록 제공한다. CPU Ready란 가상머신이 실행할 준비가 되었지만 서버의 CPU 코어를 할당받기까지 대기하는 시간이다[6].

가상머신에 CPU를 할당하는데 지연 시간이 적을수록 운용자는 가상머신을 쾌적하게 이용할 수 있다. 만약 CPU 할당 지연 시간이 임계치 이상으로 올라갈 경우 게스트 OS의 CPU 사용률이 정상치더라도 스케줄링 지연에 의한 성능 저하가 발생할 수 있다.

### 2.2. Co-Stop

가상머신의 모든 VCPU는 동기식 진행이 필요하다. 하이퍼바이저는 가상머신에 할당된 VCPU의 동기성을 추적하기 위해 가상머신에 할당된 각 VCPU의 진행률 차이를 추적한다. 이 차이를 스큐라고 한다. 너무 높은 스큐는 가상머신이 모든 프로세서에 동기적으로 액세스할 수 없음을 나타낸다. 하이퍼바이저의 스케줄링 방식은 엄격한 공동 스케줄링(Strict co-scheduling)에서 완화된 공동 스케줄링(Relaxed co-scheduling) 방식으로 변화되어 왔다. 엄격한 공동 스케줄링은 가상머신의 모든 VCPU를 동시에 예약할 수 있는 충분한 물리 CPU(PCPU)가 있을 때까지 가상머신을 중지한다. 중지된 가상머신은 모든 가상 프로세서를 수용할 수 있는 충분한 물리적 프로세서를 기다려야 했기 때문에 스케줄링 지연이 발생할 수 있다[7].

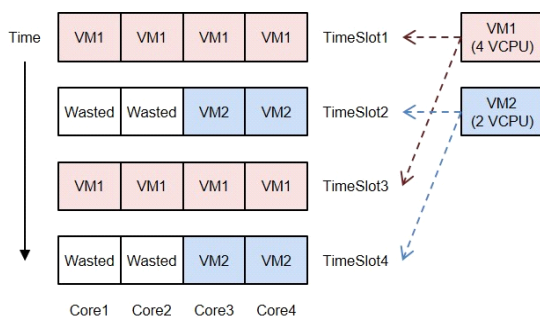


Fig. 1 Strict co-scheduling

그림 1은 엄격한 공동 스케줄링의 예시를 보여준다. 4개의 CPU 코어를 가진 서버에 각각 4개와 2개의 VCPU를 가지는 가상머신이 있다고 가정한다. 첫 번째 타임 슬롯 동안 스케줄러는 4개의 VCPU를 사용하는 VM1에 4코어를 모두 할당한다. 2개의 VCPU가 있는 VM2는 작

업을 예약 할 수 없기 때문에 대기 상태로 전환된다. 대기 상태 동안 가상머신은 실행할 준비가 되었지만 예약 가능한 CPU 코어가 없다. 이 때 CPU Ready가 발생하게 된다. 두 번째 타임 슬롯에서 VM1도 마찬가지로 대기하게 된다. 두 가상머신 모두 대기 상태가 발생하기 때문에 결과적으로 전체 성능이 저하된다.

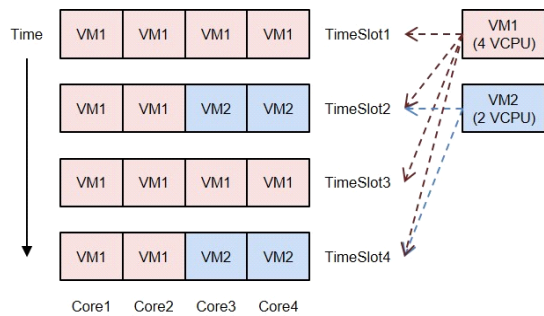


Fig. 2 Relaxed co-scheduling

그림 2의 완화된 공동 스케줄링 방식은 타임 슬롯 동안 가상머신의 모든 VCPU를 예약할 CPU 코어가 부족하더라도 계속해서 스케줄링한다. 이 때 가상머신의 일부 VCPU는 다른 VCPU보다 느리게 수행되기 때문에 스큐가 증가하게 되고 특정 제한을 초과하면 다른 VCPU가 따라잡을 수 있도록 VCPU의 공동 중지(Co-Stop)를 수행한다[6, 7].

어플리케이션은 모든 코어가 동일한 속도로 실행될 것으로 예상하므로 어떤 일이 발생하는지 인식하지 못한다. 공동 중지될 가능성을 줄이기 위해선 필요한 만큼의 VCPU를 사용하도록 가상머신의 크기를 조정하는 것이 중요하다.

## III. 관련 연구

### 3.1. Blast Extreme

VMware에서 개발한 디스플레이 프로토타입인 Blast Extreme은 NVIDIA와 협력을 통해 가상 GPU 기술을 이용하여 CPU 소모를 감소시키고 그래픽 처리 능력을 향상시키도록 발전해왔다. 또한 지연시간을 줄이고 대역폭을 줄이는 등 사용자 경험 향상을 위한 노력을 기울이고 있다. Blast Extreme은 구성에 따라 JPG/PNG 코덱, Blast 코덱, H.264 코덱, H.265 코덱을 사용한다. 코덱은

전송을 위해 화면 콘텐츠를 인코딩하거나 디코딩할 수 있는 프로그램을 말한다. 표 1에 Blast Extreme에서 지원하는 코덱의 종류별 특징을 나타내었다[8].

Table. 1 Blast Extreme protocol codecs.

Codec	Description
JPG/PNG Codec	<ul style="list-style-type: none"> <li>• supports a lossless format</li> <li>• high CPU and bandwidth utilization</li> </ul>
Blast Codec	<ul style="list-style-type: none"> <li>• supports applications that require the High-quality graphics support</li> <li>• low CPU and bandwidth utilization</li> </ul>
H.264 Codec	<ul style="list-style-type: none"> <li>• supports rapidly moving screen content</li> <li>• supports encoding offload to GPU</li> </ul>
H.265 Codec	<ul style="list-style-type: none"> <li>• provides up to 50% better compression with the same quality as H.264</li> <li>• low bandwidth / high CPU utilization</li> <li>• supports encoding offload to GPU</li> </ul>

JPG/PNG 코덱은 무손실 압축을 지원하며 고품질의 이미지와 텍스트 제공이 가능하지만 많은 CPU와 대역폭 사용이 필요하다. 정적 콘텐츠로 구성된 워드 프로세서 또는 스프레드시트와 같은 응용 프로그램을 사용하는 워크로드에 적합하다.

Blast 코덱은 JPG/PNG 코덱의 후속 모델로 고품질을 유지하면서 CPU 및 네트워크 대역폭을 더 적게 사용하도록 개선되었다. 오피스 업무 환경에 필요한 일반적인 Windows 어플리케이션뿐만 아니라 Photoshop 및 AutoCAD와 같은 그래픽 지원이 필요한 앱을 지원하도록 설계되었다.

H.264은 소프트웨어 인코딩 및 하드웨어 디코딩을 수행하며 GPU에 비디오 및 이미지 재생과 관련된 작업을 오프로드 한다. GPU로 관련 작업을 오프로드하기 때문에 CPU 소비가 줄어드는 장점이 있다. 서버의 물리적 GPU를 가상머신에 논리적으로 분배하는 VGPU 기능과 함께 사용할 경우 하드웨어 인코딩 또한 지원하여 까다로운 그래픽 워크로드에 사용하기 적합하다. 그림 3은 하드웨어 인코딩을 이용한 Blast Extreme의 동작을 나타낸다. 하드웨어 인코딩을 사용하면 렌더링뿐만 아니라 기존의 CPU가 수행하는 디스플레이 캡처 및 인코딩을 GPU로 오프로드하여 처리한다.

H.265 코덱은 H.264 코덱과 마찬가지로 하드웨어 인코딩 방식이 가능하며 동일한 품질로 최대 50% 향상된 압축을 지원한다. 이로 인해 필요한 네트워크 대역폭이 감소하지만 인코딩 및 디코딩에 H.264보다 CPU를 더

많이 사용하는 특징을 갖는다[8, 9].

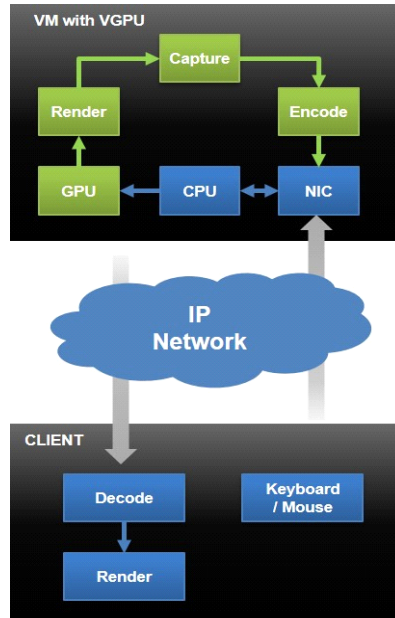


Fig. 3 Blast Extreme with H/W encoding

이처럼 Blast Extreme은 코덱의 동작에 따라 각각의 특성을 가지고 동작한다. 따라서 Blast Extreme 최적화를 위해 디스플레이 프로토콜의 코덱을 이해하고 워크로드에 적합한 방식을 적용하는 것이 중요하다.

### 3.2. 가상 CPU 스케줄링 최적화

대부분의 데이터 센터 및 클라우드 플랫폼은 NUMA (Non-Uniform Memory Access) 구조를 가진다. NUMA란 멀티프로세서 시스템에서 사용되고 있는 컴퓨터 메모리 설계 방법 중의 하나로, 메모리에 접근하는 시간이 메모리와 프로세서 간의 상대적인 위치에 따라 다른 시스템이다. 따라서 NUMA 기반 환경에서 성능 최적화를 위해선 상대적으로 가까운 로컬 메모리의 접근 빈도를 늘리는 것이 중요하다. ESXi 및 Xen과 같은 주요 가상화 솔루션도 NUMA 기반 시스템에서의 성능 최적화를 위해 가상 NUMA(VNUMA)와 같은 기능을 제공한다[10].

가상머신의 VNUMA 토폴로지는 물리적 NUMA 토폴로지와 최대한 유사하도록 구성된다. 따라서 CPU 패키지의 물리적 경계에 맞춰 소켓 당 가상머신 코어를 구성하는 것이 좋다. 가상머신의 소켓과 코어 구성을 최적

화하는 방안을 Powershell 기반 언어를 이용하여 아래 알고리즘으로 나타내었다.

**Algorithm VM NUMA Optimization**

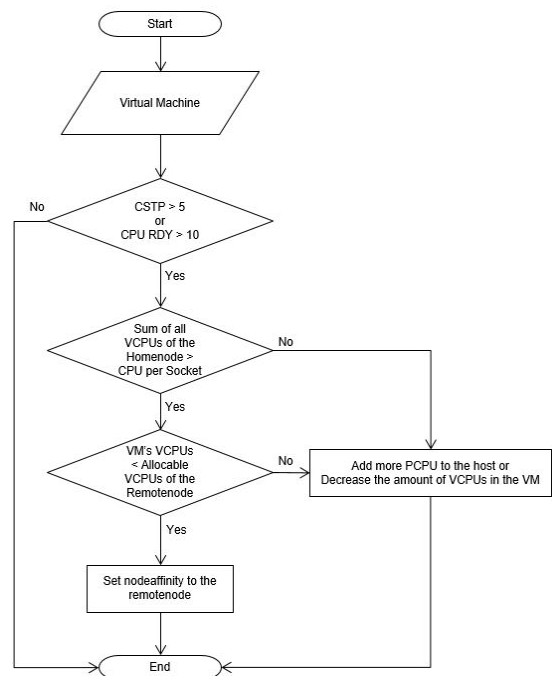
```
function OPTIMUM_CORES_PER_SOCKET($VM) {
    $NumaNodeMemory = $ParentHost.MemoryTotalGB /
        $ParentHost.NumCpuPkgs
    $HostCoresPerSocket = $ParentHost.NumCpuCores /
        $ParentHost.NumCpuPkgs
    if ( $VM.MemoryGB -le $NumaNodeMemory )
    {
        if ( $VM.NumCpu -le $HostCoresPerSocket )
        {
            $VMSpec = New-Object -Type
                VMware.Vim.VirtualMachineConfigSpec -Property
                @{"NumCoresPerSocket" = $VM.NumCpu}
            $VM.ReconfigVM_Task($VMSpec)
        }
        else
        {
            $VMSpec = New-Object -Type
                VMware.Vim.VirtualMachineConfigSpec -Property
                @{"NumCoresPerSocket" = $VM.NumCpu /
                    Ceiling($VM.NumCpu / $HostCoresPerSocket)}
            $VM.ReconfigVM_Task($VMSpec)
        }
    }
    else
    {
        $VMSpec = New-Object -Type
            VMware.Vim.VirtualMachineConfigSpec -Property
            @{"NumCoresPerSocket" = $VM.NumCpu /
                Ceiling($VM.MemoryGB / $NumaNodeMemory)}
        $VM.ReconfigVM_Task($VMSpec)
    }
    }
    return $VM.CoresPerSocket
}
```

가상머신의 적절한 소켓 개수는 NUMA 노드 당 메모리 및 코어 수를 고려하여 결정하는 것을 권장한다. 가상머신의 메모리가 NUMA 노드의 로컬 메모리 크기보다 크다면 가상머신의 메모리 크기를 NUMA 노드의 로컬 메모리 크기로 나눈 값으로 소켓을 구성하여 VNUMA 토폴로지를 구성하도록 하는 것이 좋다. 가상머신의 메모리가 NUMA 노드의 로컬 메모리 크기보다 작더라도 가상머신의 VCPU의 개수를 NUMA 노드의 물리 코어 개수로 나눈 값으로 소켓을 생성하여 VNUMA 토폴로지 최적화하는 것을 권장한다.

주의할 점은 가상머신들의 VNUMA 토폴로지가 최적화되어 있더라도, 성능 저하가 발생할 수 있는 점이다. 스케줄러는 작업 선호도를 고려하기 때문에 가상머신 간 서로 통신하는 등 상호 작용이 빈번하다면 해당 가상머신들이 서로 다른 NUMA 노드에 있는지 검사한 후 같은 NUMA 노드에 위치하도록 마이그레이션 한다. 그러나 마이그레이션 동작은 CPU 부하 및 CPU 경합을 고려하지 않기 때문에, NUMA 노드간 부하 불균형이 발생하고 CPU 경합에 의해 Co-stop 혹은 CPU Ready와 같은 스케

줄링 지연이 발생할 수 있다. VMware는 이러한 스케줄러의 동작이 워크로드에 적합하지 않을 경우 가상머신의 노드 선호도 설정을 통하여 가상머신을 해당 NUMA 노드에서 동작하도록 하는 기능을 지원한다[11].

그림 4에 CPU 경합을 줄이기 위해 가상머신을 스케줄링할 NUMA 노드를 지정하는 방법을 나타내었다. VMware는 일반적으로 가상머신의 CSTP(Co-stop) 값은 5, RDY(CPU Ready) 값은 10을 넘지 않는 것을 권고한다[12].



**Fig. 4 VM NUMA Node Determination Flowchart**

CSTP 혹은 RDY 값이 임계치를 초과하면 가상머신의 로컬 NUMA 노드에서 스케줄링하고 있는 가상머신들의 전체 VCPU 수를 확인한다. 전체 VCPU 개수가 NUMA 노드에서 예약 가능한 CPU 코어 개수보다 크다면 특정 가상머신을 다른 NUMA 노드로 마이그레이션 해야 한다. 이 때 마이그레이션 대상 NUMA 노드는 해당 가상머신의 VCPU를 예약할 충분한 유휴 CPU가 존재해야 한다. 그렇지 않다면 스케줄링 조정을 통해 해결 가능한 문제가 아니라고 판단되어 가상화 서버를 운영하는 관리자 및 사용자는 전체 가상머신 VCPU 수를 줄일 것인지, 혹은 서버의 CPU 사양을 업그레이드 할 것

인지 의사결정이 필요하다.

#### IV. 성능 시험

본 장에서는 Blast Extreme 프로토콜의 코덱별 기본 벤치마크 성능 차이와 멀티 코어 기반 HCI 어플리케이션을 구동한 환경에서의 성능 차이를 분석한다. 또한 동일한 형상의 가상머신 여러 개에 CPU 경합을 발생시켜 스케줄링 최적화 적용에 따른 CPU Ready 및 Co-stop 값 변화를 관찰하였다. 해당 시험을 위한 하드웨어 시험환경은 표 2와 같다.

**Table. 2** H/W environment for test.

H/W	Spec
Server	CPU: Intel Xeon Gold 6140 2.3GHz (36 Physical Cores) GPU: NVIDIA A40
VM	VCPU: 12 VCPU VGPU: A40-4q
Client PC	CPU: Intel Core i7-10700 2.90GHz GPU: NVIDIA RTX A2000

데스크탑 가상화 구성에 사용되는 서버는 18 코어 CPU가 2개로 구성되어 있어 전체 36개의 물리적 코어를 갖는다. 해당 서버에는 12 VCPU를 가지는 가상머신을 생성하여 하드웨어 인코딩을 위한 VGPU를 할당하였다. 클라이언트 PC는 3화면 모니터를 통해 가상머신에 원격접속하며 하드웨어 디코딩을 위한 GPU가 구성되어있다.

가상화 솔루션을 포함한 소프트웨어 환경은 표 3과 같다. 시험에 사용한 멀티 코어 기반 HCI로 전투체계 표적 생성 시뮬레이터를 사용하였다. 전투체계 표적 생성

**Table. 3** S/W environment for test.

Category	S/W Version
Hypervisor	VMware ESXi 7.0.2
VDI	VMware Horizon Client 8.4.0 VMware Horizon View Agent 8.4.0 VMware View Agent Direct-Connection 8.4.0
Display Protocol	Blast Extreme
VM OS	Windows Server 2019 standard
Application	Target Generator (up to 10,000 targets)

시뮬레이터는 동시에 다가오는 최대 수천 개의 위협에 대해 실시간으로 탐지하며 이에 대응하기 위해 화면에 전시한다. 따라서 멀티 코어를 사용하면서 VDI의 전시 성능까지 측정하기에 적합하다.

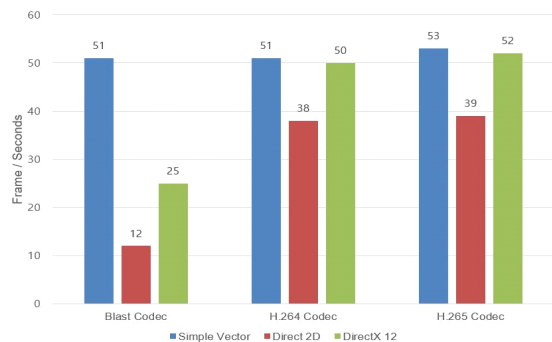
##### 4.1. 디스플레이 프로토콜 성능 시험

본 절에서는 VMware 기반 데스크탑 가상화 솔루션에서 Blast Extreme 프로토콜의 코덱별 성능을 분석한다. 분석을 위하여 두 가지 항목을 시험하였다.

첫째로, 벤치마크 측정 도구인 PassMark를 이용하여 벡터 도형을 화면에 도시하는 성능인 Simple Vector, 2D와 3D 그래픽 처리 성능인 Direct 2D와 DirectX 12를 측정하였다[13].

둘째로, 전투체계 표적 생성 시뮬레이터를 활용하여 표적을 최대 10,000개까지 단계적으로 생성하여 고부하의 상황에서 디스플레이 프로토콜의 전시 성능을 코덱별로 측정하였다.

그림 5에 Blast Extreme의 코덱 별 성능 측정 결과를 나타내었다. Simple Vector 항목에선 H.265 코덱이 가장 높은 결과 값이 나왔지만 모든 코덱 구성에서 거의 동일한 성능을 보여줬다. Direct 2D와 DirectX 12 항목에선 H.264 코덱과 H.265 코덱이 Blast 코덱에 비해서 뚜렷하게 높은 성능 결과를 보여주었다. H.265 방식이 H.264보다 근소하게 높았으나 유의미한 차이를 보이지 않았다. 벤치마크 결과를 종합하였을 때 정적인 화면 콘텐츠의 도시 성능은 모든 코덱이 비슷하나 빠르게 움직이는 화면 콘텐츠나 모션 그래픽과 같은 워크로드에선 H.264 혹은 H.265 코덱을 활용한 원격접속이 좋은 성능을 보여줬다.



**Fig. 5** The Result of Benchmark

표적 생성 시뮬레이터를 활용한 시험 결과를 그림 6에 나타냈다. 표적 생성은 단계적으로 1,000개, 3,000개, 5,000개, 10,000개 순으로 늘려가며 프레임을 측정하였다. Blast 코덱의 경우 3,000개 까진 40FPS 이상을 보였지만, 5,000개에서 38FPS, 10,000개에서 27FPS까지 프레임이 저하되었다. 반면, H.264와 H.265 코덱은 Blast 코덱보다 높은 성능을 보였다. 표적 3,000개의 부하 상황에서도 60FPS를 유지하였고, 5,000개 표적 상황에서 50FPS 이상, 10,000개 표적 상황에서 40FPS 이상을 유지하여 고부하인 상황에서도 최적의 성능을 보여주었다.

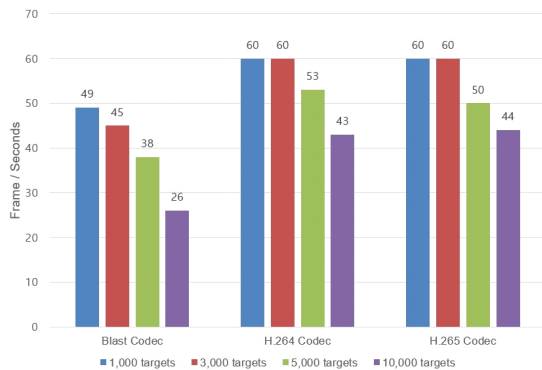


Fig. 6 The Result of Target display performance



Fig. 7 FPS at 5,000 targets with H.264



Fig. 8 FPS at 5,000 targets with H.265

정확한 차이 식별을 위해 그림 7과 그림 8에 표적 5,000개 상황에서 H.264와 H.265 코덱을 사용하였을 때의 프레임 변화를 각각 그래프로 나타내었다. H.264와

H.265 코덱이 평균적으로는 거의 동일한 FPS를 보이지만 H.264 방식이 H.265보다 프레임 등락이 작아 좀 더 안정적인 프레임을 보여준다. 결과를 종합하였을 때, Blast 코덱은 기존 JPG/PNG 코덱의 장점인 고품질을 유지하고 단점이었던 그래픽 지원이 필요한 워크로드에도 동작하도록 설계되었지만 고부하의 상황에 취약하고 다른 코덱에 비해 그래픽 처리 성능이 확연히 낮은 것으로 확인되었다. 그에 비해 H.264와 H.265는 고부하인 상황에서도 높은 프레임을 유지하였고 H.264 코덱이 H.265보다 고부하인 상황에서 좀 더 최적화된 성능을 보였다.

#### 4.2. 가상 CPU 스케줄링 시험

본 절에서는 가상 CPU 스케줄링 최적화 후 성능 변화를 비교한다. 성능 저하는 가상화 환경에 추가적인 가상머신을 구동할 때 발생 가능하다. 더 많은 가상머신이 구동되어 CPU 사이클 경쟁이 시작되면 기존 가상머신은 상대적으로 성능이 저하될 수 있다. 따라서 동일한 구성의 가상머신 4대에서 동일한 부하를 발생시켜 CPU 경합을 발생시켰다. 각 가상머신에서 표적 생성 시뮬레이터를 통해 10,000개의 표적을 생성하고 동일한 부하가 발생한 상황에서 VCPU 스케줄링의 성능 지표인 CPU Ready 값과 Co-Stop 값을 관찰하였다.

NAME	%IDL	%USED	%RUN	%RDY	%IDLE	%OVRLP	%CSTP
VM19	26	332.37	333.60	23.51	839.76	1.93	0.16
VM20	26	319.01	299.01	18.88	879.53	1.36	0.00
VM18	26	314.29	315.53	20.73	862.74	2.03	0.16
VM21	26	314.21	312.94	21.70	859.79	1.82	0.00

Fig. 9 RDY and CSTP values before optimization

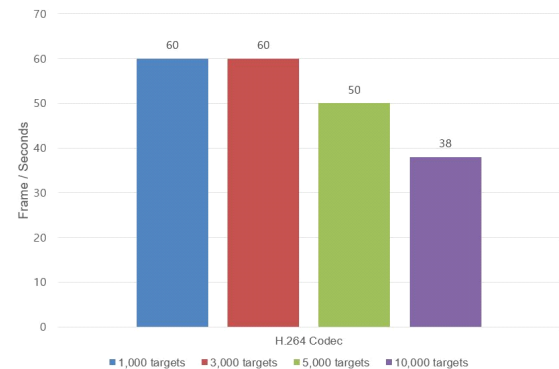


Fig. 10 Target display performance when 4VMs are competing

그림 9는 가상머신의 VCPU 스케줄링 최적화를 수행하지 않은 환경에서의 시험 결과를 보여준다. 성능 메트릭 중 USED는 가상머신의 VCPU 사용량을 나타낸다. 해당 값을 게스트 OS의 CPU 사용률로 환산하면 약 27%정도의 정상 수치임에도 RDY(CPU Ready) 값이 약 20까지 증가하여 스케줄링 지연에 따른 성능 저하가 발생하였다. CSTP(Co-Stop) 값의 경우에도 2개의 가상머신에서 0.16 정도의 값이 발생하였다. 해당 상황에서 앞선 시험에 가장 좋은 성능을 보였던 H.264 코덱의 원격 접속 프레임 결과를 그림 10에 나타냈다. 표적 3,000개까지는 스케줄링 지연이 발생하지 않았지만 5,000개 표적 이상의 부하 상황에서는 경쟁이 발생하여 CPU 경합이 없을 때 시험한 결과보다 프레임이 약 5% ~ 10% 낮게 측정되었다.

그림 11은 가상머신의 VCPU 스케줄링 최적화를 수행한 환경에서의 시험 결과를 보여준다. 최적화 이전과 동일한 워크로드를 수행하여 동일한 USED 값에서도 CPU Ready 값을 5이하로 유지하고 CSTP값이 발생하지 않아 스케줄링 지연이 발생하지 않았다. 표적 전시 성능 또한 CPU 경합이 없는 환경과 동일한 수준의 결과를 보였다.

NAME	NMLD	%USED	%RUN	%RDY	%IDLE	%OVRLP	%CSTP
VM21	26	346.41	288.59	1.55	902.38	1.50	0.00
VM20	26	339.93	290.50	2.50	899.76	1.01	0.00
VM18	26	339.10	289.97	2.16	902.29	0.92	0.00
VM19	26	305.99	255.11	1.35	933.73	1.64	0.00

Fig. 11 RDY and CSTP values after optimization

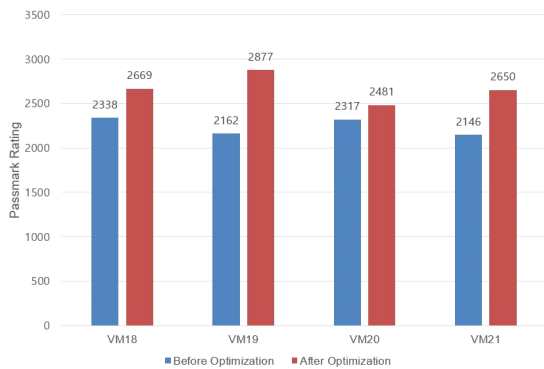


Fig. 12 VM performance after optimization

실제로 성능 개선이 이루어졌는지 판단하기 위해 벤치마크 측정 도구 PassMark의 종합 성능 지표인 PassMark Rating을 각 가상머신 별로 측정된 결과를 그

림 12에 나타냈다. 벤치마크 측정은 기존 시험과 동일하게 각 가상머신에 표적 10,000개의 부하를 발생시킨 상황에서 진행하였다. 벤치마킹 결과 각 가상머신에 평균 약 10% 이상의 성능 향상이 나타났다. 결과적으로 가상 CPU 스케줄링 최적화 적용을 통해 CPU 경합이 있는 상황에서 가상머신의 성능 개선이 가능함을 확인하였다.

## V. 결론

본 논문에서는 멀티코어 기반 어플리케이션 운용에 적합한 데스크탑 가상화 구성을 선정하는 것을 목적으로 VMware 가상화 환경에서 코덱 별 디스플레이 프로토콜 성능을 분석하고 가상머신 간 CPU 경합을 줄이기 위한 가상 CPU 스케줄링 최적화 방안을 연구하였다.

VMware Blast Extreme 프로토콜의 H.264 코덱이 다른 코덱과 비교하여 부하가 있는 상황에서도 높은 성능과 안정성을 제공하는 것을 확인하였다. 표적 전시 시험의 경우 H.265 코덱은 H.264와 비슷한 FPS를 보였지만 프레임 편차가 커서 H.264 코덱이 상대적으로 균등하고 안정적인 프레임을 제공하였다. Blast 코덱의 경우 정적인 화면 콘텐츠를 전달하는 성능은 다른 코덱과 동등하였으나 고부하의 상황에서 빠르게 움직이는 화면 콘텐츠에는 취약한 것으로 확인되었다.

또한 가상머신의 소켓 및 코어 설정을 통해 VNUMA 구성을 최적화하고 가상머신을 스케줄링할 NUMA 노드를 알맞게 지정함으로써 CPU 경합이 있는 상황에서 가상 CPU의 스케줄링 성능이 향상됨을 확인하였다.

향후에는 데스크탑 가상화에서 비디오 및 오디오 스트리밍 서비스를 위한 I/O 지연 개선 방안에 대해 연구를 진행할 것이다.

## References

- [ 1 ] VMware, "Best practices for performance tuning of latency-sensitive workloads in vSphere VMs," VMware White Paper, Sep. 2013.
- [ 2 ] VMware, "Media and Entertainment Workloads on vSphere 6.7," VMware White Paper, Mar. 2019.
- [ 3 ] J. Y. Lim and D. S. Kim, "Performance Evaluation of Virtualization Solution for Next Generation Naval Combat



- Systems,” *Journal of The Institute of Electronics and Information Engineers*, vol. 56, no. 2, pp. 41-49, Feb. 2019.
- [ 4 ] K. S. Song, “A Study of Feasibility and Performance Analysis of VDI based on GPU Acceleration for Naval Combat System,” *Journal of The Institute of Electronics and Information Engineers*, vol. 58, no. 11, pp. 86-93, Nov. 2021.
- [ 5 ] J. S. Shin and H. Y. Kim, “Event Routing Scheme to Improve I/O Latency of SMP VM,” *Journal of Korean Institute of Information Scientists and Engineers*, vol. 42, no. 11, pp. 1322-1331, Nov. 2015.
- [ 6 ] VMware, “vSphere Monitoring and Performance,” VMware White Paper, Apr. 2020.
- [ 7 ] VMware, “The CPU Scheduler in VMware vSphere 5.1,” VMware White Paper, Jan. 2013.
- [ 8 ] VMware Digital Workspace Tech Zone. VMware Blast Extreme Optimization Guide [Internet]. Available: <https://techzone.vmware.com/resource/vmware-blast-extreme-optimization-guide>.
- [ 9 ] VMware End-User Computing Blog. Delivering a Transformational User Experience with NVIDIA GRID & VMware Horizon 7 [Internet]. Available: <https://blogs.vmware.com/euc/2016/05/transform-user-experience-nvidia-grid-vmware-horizon-7.html>.
- [10] S. Y. Kim, T. H. Lee, and Y. I. Eom, “Performance Analysis of NUMA-aware Locks in Virtualized Systems,” *Journal of KIISE Transactions on Computing Practices*, vol. 27, no. 2, pp. 107-109, Feb. 2021.
- [11] Nutanix portal. Intermittent CPU ready time due to NUMA action affinity on VMware ESXi [Internet]. Available: <https://portal.nutanix.com/page/documents/kbs/details?targetId=kA07V000000LXAwSAO>.
- [12] VMware Docs. esxtop data [Internet]. Available: <https://docs.vmware.com/en/VMware-Smart-Assurance/10.1.0/ip-manager-delopment-guide-101/GUID-8F62BA71-5C6B-41DE-9B9B-0BF24ADD743B.html>.
- [13] PassMark portal. PerformanceTest [Internet]. Available: <https://www.passmark.com/products/performancetest/index.php>.



**박준용(Junyong Park)**

2015년 충남대학교 정보통신공학과 학사 졸업.

2015년 ~ 현재 한화시스템 근무.

※관심분야 : 가상화, 클라우드, 함정전투체계, 미들웨어