

# 유닉스 시스템 기반의 가상화 기술 성능 비교 연구

\* 안홍준

\* 고려대학교 컴퓨터정보통신대학원 정보통신공학과  
e-mail : hong-joon.an@hp.com

## Benchmark of Virtualization Technologies based on UNIX system

\*Hong-Joon, An

\* Graduate School of Computer & Information Technology, Korea University

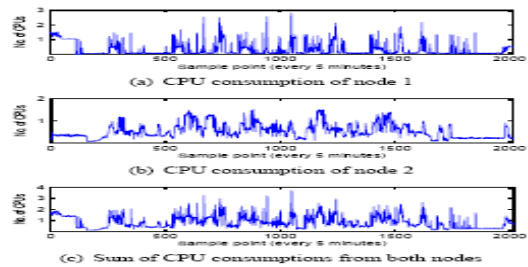
### 요 약

오늘날 산업계에서 시스템 통합은 IT 를 계획하는데 있어서 비용을 줄이고 효율성을 향상 시키기 위한 중요한 방법으로 부각되고 있다. 가상화의 장점은 단일서버에 다중의 운영체제를 통합할 수 있는 환경을 제공해 준다. 그러나 올바른 가상화 기술과 시스템 통합을 위한 가상화 구성방법 선택은 중요한 과제이다. 따라서 이 연구의 목적은 유닉스 시스템 가상화 기술을 설명하고 유닉스 시스템 기반 상에서 벤치마킹 테스트 도구를 활용하여 HP 가상화 기술인 nPars, vPars, VM의 성능을 비교테스트를 하고자 한다.

### 1. 서론

대다수의 데이터 센터에서는 시스템의 안정적 운영에 치우친 나머지 시스템 자원 활용에 대한 구체적 관리가 부재한 상태다. 또한 유휴 자원이 있어도 시스템간 동적 자원할당체계 및 기술의 부족으로 쉽게 필요한 곳에 사용되기는 어렵다. 이러한 문제로 인해 IT 시스템에 대한 과잉 투자 및 중복 투자가 심화되고 있으며 특히 고가의 서버 시스템에 대한 투자 대비 활용률이 30~40% 수준에 머물고 있는 상황이다. 반면에 데이터 센터 내의 다른 일부 시스템에 부하가 집중되고 있다. 또한 과거 10 여 년 동안 서버의 수는 150 배 가량 증가하였고 이와 관련된 비용 역시 폭증하였다[1]. 이로 인해 많은 기업과 기관들은 IT 투자가치를 극대화하고 자원에 대한 효율적 활용을 위해 여러 방안을 요구하게 되었으며 이를 충족하기 위하여 가상화 기술이 등장하게 되었다[2]. 서버 가상화는 CPU, 메모리, 입출력 장치 등의 다양한 서버 자원을 하나의 집합으로 묶어서 사용자의 필요에 따라 자유롭게 나누어 사용하는 개념으로 작은 여러 대의 서버들을 보다 쉽게 관리하고 재할당 하여 효율을 높이는 데 사용된다. 시스템 통합은 TCO(Total Cost Of Ownership)를 줄이고 ROI(Return On Investment)를 높이기 위해 데이터 센터에서 일반적으로 많이 사용하고 있다. 그림 1(a)와 그림 1(b)는 일주일 동안의 데이터 센터내의 두 서버의 시스템 CPU 사용량을 보여주

고 있다. 두 서버의 peaktime 시점이 틀리다. 만약 두 시스템이 하나로 통합되어 두 개의 가상서버로 사용된다면 두 개의 가상서버 모두 QOS(quality-of-service)를 충족시킨다[3].



<그림 1> 데이터 센터 서버 사용의 예

그러나 아무리 시스템 통합이 자원의 효율성과 성능을 크게 높이더라도 업무성격에 맞는 올바른 가상화 기술 선택과 구성은 중요한 이슈가 되었다[4]. 또한 서버 통합을 위해 가상화 솔루션을 구현하여 사용 중인 관리자의 고민은 가상화 시스템을 구현했을 때 각각의 워크로드가 가지는 성능 영향의 정도이다.

예를 들면, 메모리 접근 지연시간은 로컬 접근, 외부 접근, 인터리브드 메모리 접근 등 접근 방식에 따라 매우 다양하고 성능에 많은 영향을 미친다[5].

소프트 파티션은 구현하는데 있어서 높은 유연성을 제공한다. 그러므로 자원의 지역성(locality)을 고려하지 않은 자원 할당은 워크로드 성능에 부정적인 영향

을 미친다. 따라서 이 연구의 목적은 유닉스 시스템의 가상화 기술을 설명하고 유닉스 시스템 기반상에서 벤치마킹 도구를 활용하여 HP 가상화 기술인 nPartitions[6], vPartition[7], virtual machine[8]의 성능 비교 테스트 하고자 한다. 다양한 실험을 통해 소프트웨어 파티션인 vPartitions 과 Virtual Machine 을 중점 비교한다. 두 가상화 방법 모두 HP 의 대표적인 가상화 방법이고 유닉스 운영체제 기반이며 데이터 센터에서 많이 사용하고 있다. 가상화 방법과 구성변경에 따른 성능 평가를 위하여 표준 벤치마크 도구인 SPECJBB2000[9]를 사용하여 서버의 성능을 평가하였다. 본 논문은 총 5 장으로 구성되어 있으며 먼저 제 2 장에서는 관련 연구로 가상화 기술인 하드웨어 파티션과 소프트웨어 파티션에 대해 설명하고, 제 3 장에서는 테스트 도구와 테스트 장비 사양에 대해 설명하고, 이 연구의 핵심인 제 4 장에서는 다양한 방법의 실험과 그 결과에 대해 자세히 기술한다. 마지막으로 5 장에서 결론과 향후 연구에 대해 기술한다.

2. 관련 연구

서버 가상화는 어플리케이션의 성능, 가용성을 최적화하고 TCO 을 감소시킨다. 이외에도 IT 관리를 더 쉽게 운영할 있게 하고 시스템 자원을 충분히 활용할 수 있게 한다. 서버 가상화 기술을 통한 이점을 살펴보면[4]

- 하나의 큰 시스템을 그 안에서 여러 시스템으로 나누어서 사용률을 극대화 하였음
- 파티션은 실시간으로 장애 복구가 가능한 서비스 인프라의 탄력성과 어떠한 상황변화 (시스템의 자원 증설/축소, 서비스의 추가/삭제 등)에도 빠르게 대처 가능한 유연성을 제공
- 사용자의 데이터 접근 및 서비스 중단 없이 시스템 운영 가능한 가용성 제공
- 시스템을 통합 시 물리적 공간을 줄이고 소비 전력 및 냉각(power and cooling) 비용이 줄며, 소수의 인원으로 시스템을 관리할 수 있으므로 운영비용 감소
- 새로운 어플리케이션이 개발되어 시스템에 적용할 때 기존 시스템 자원 확장성 제공

2.1 하드웨어 파티션

본 연구에서는 하드웨어 파티션인 HP nPartition[6]에 대해 설명한다. 셀(cell) 단위로 구성되며 모든 하드웨어 구성요소가 파티션 간에 분리 되어있다. 독립된 시스템 환경으로 사용하기 위해 시스템 하드웨어 자원을 나누어 모듈화 한다. 하나의 파티션은 하나 혹은 그 이상의 셀(CPU, 메모리, 입출력장치가 독립적으로 제공하는 모듈)을 가지고 있고 모든 입출력 장치는 셀들과 연결되어 있다. 파티션 안의 CPU, 메모리 그리고 입출력 장치는 독립적으로 어플리케이션을 수행한다. 각각 자기자신의 부트 인터페이스를 가지고 가동과 재가동을 독립적

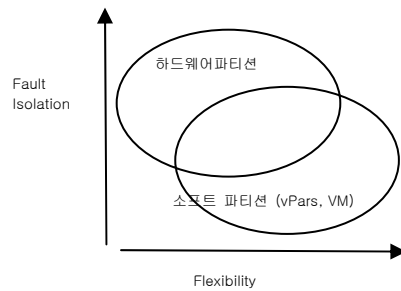
로 수행한다. 또한 하드웨어적으로 분리되어 있어서 하드웨어, 소프트웨어 장애는 하나의 물리적인 시스템 안에서 서로 영향을 미치지 않는다.

app	app	app	app
os	os	os	os
cell	cell	cell	cell
hardware			

<그림 2> Hardware Partition 소프트웨어 스택

2.2 소프트웨어 파티션

소프트 파티션은 하드웨어 파티션 보다 더 세부적인 기능을 제공한다. 본 연구에서는 소프트웨어 파티션인 HP 의 vPartitions[7]과 VM[8]에 대하여 설명한다. 소프트웨어 파티션의 주요 기능은 처리코어수준(vPars) 또는 시간 간격에 의해(VM)구현되고, 운영체제 분리만 제공하고 전기적 분리는 제공하지 않는다. 또한 소프트웨어 파티션을 사용하여 파티션을 더욱 세분화 할 수 있다. 그림 3 에서 보듯이 소프트웨어파티션은 더 많은 유연성을 가지는 반면 하드웨어 파티션은 완벽한 물리적 파티션이고 모든 하드웨어 장애에도 독립적이다.



<그림 3> 파티션 특징

2.2.1 vPartitions

단일 시스템을 하나의 대형 시스템 또는 여러 개의 소형 시스템으로 구성 할 수 있다. Vpar 를 통하여 각각의 운영체제에서 사용하는 어플리케이션이나 OS 를 독립적으로 운영할 수 있게 한다. 또한 시스템의 재가동 없이 CPU 자원을 vPartitions 간에 주고 받을 수 있다. vPartitions 를 구성하는 요소는

- vPartitions Monitor  
소프트웨어 파티션과 각 파티션의 커널에 하드웨어 리소스를 할당한다. 가상 파티션이 시작되면 monitor 는 가상파티션의 하드웨어 소유권을 전달하고 각각의 운영체제는 하드웨어 리소스의 소유권을 가진다.
- vPartitions database  
각 파티션에 할당된 하드웨어 자원에 대한 정보를 가지고 있다. monitor 가 기동될 때 파티션 데이터 베이스를 읽어 각 파티션 정보를 메모리로 로드 한다.
- Boot processor and Dynamic CPU  
Boot processor 는 파티션의 OS 커널이 기동될 때의 CPU 이다. Dynamic CPU 는 Boot processor 를 제

외한 나머지 CPU 를 의미한다. 각 파티션은 Boot processor 를 제외한 나머지 CPU 를 동적으로 재할당 할 수 있다.

app	app	app	app
OS	OS	OS	OS
Virtual partition monitor			
hardware			

<그림 4> vPartitions 소프트웨어 스택

### 2.2.2 Virtual Machine

CPU 할당을 세분화 하며 공유 입출력으로 운영체제 분리를 제공하는 소프트웨어 파티션 기술이다. VM 호스트와 게스트 시스템 두 가지 구성요소로 이루어진다.

- VM 호스트는 CPU, 메모리 및 입출력 장치를 가상화하여 게스트에 가상화를 제공하고 게스트가 필요로 하는 리소스를 할당할 수 있게 하며 리소스의 유연한 관리 및 물리적인 리소스의 효율적인 사용을 완벽하게 제어한다.
- 게스트시스템은 전적으로 실제 물리적 시스템에 대해서는 가상적인 개념이다. 게스트는 운영체제, 시스템 관리 유틸리티, 어플리케이션 및 네트워크 면에서 사용자가 설정한 가상 환경 내에서 운영되는 가상의 시스템이다.

VM 을 이용을 이용하여 여러 가상 시스템을 동일한 물리적 시스템에서 실행한다. 게스트는 VM 호스트 관리자가 할당하는 모든 CPU 를 사용할 수 있으며, CPU 사용량은 VM 관리자가 허락하는 범위 내에서 사용이 가능하다. 이러한 방법을 통해서 VM 관리자는 가상 게스트 시스템들이 물리적 CPU 의 사용을 효과적으로 관리하고, 효율 및 성능을 향상시킬 수 있다. 여러 가상 시스템이 동일한 물리적 리소스를 공유하므로 입출력 장치 사용을 최대화하고 데이터 센터의 유지보수 비용을 줄일 수 있다. 하나의 게스트 시스템 내에서 어플리케이션 장애 및 시스템 이벤트가 다른 게스트 시스템에 영향을 주지 않는다

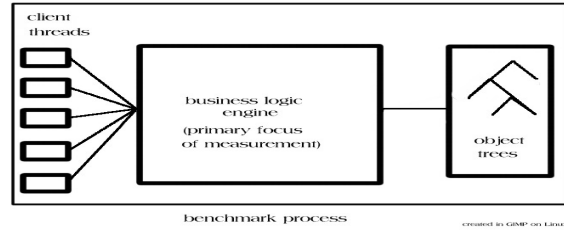
app	app	App	app
os	os	Os	os
Host OS			
Hardware			

<그림 5> Virtual Machine 소프트웨어 스택

## 3. 성능평가

### 3.1 성능 측정 방법

SPECJBB2000(JavaBusiness Benchmark)는 SPEC(standard Performance Evaluation Corporation)에서 개발한 소프트웨어 벤치마크 프로그램이다. SPECJBB2000 은 Java Server 어플리케이션을 수행할 수 있는 싱글 시스템의 성능을 측정하도록 디자인 되었다. 하나의 싱글 시스템에 3Tier 아키텍처를 구성하였으며, DBMS 를 메모리에 구성하여 모든 성능 수치의 결과는 Business Logic 를 처리하는 능력을 기준으로 수행되고 측정 단위는 BOPS (business operations per second)이다[9].



<그림 6> SPECJBB2000 프로세스

벤치마크는 16 warehouse 로 구성하고 Warehouse 의 증가에 따라 반복적으로 성능을 측정 하고 모든 warehouse 처리량의 평균값을 이 연구의 성능 측정값으로 하였다.

### 3.2 시스템 벤치마크 장비 사양

본 연구에서 다양한 방법의 가상화 구성을 위해 두 개의 셀을 가진 HP integrity rx8620 를 사용하였다. 16 개의 1.6GHz processor 와 64G 메모리로 구성되어 있다.

구분		BMT 서버
기종명		HP Integrity rx8620
CPU	processor 종류	Itanium2
	processor 수량	16
	core 수량	16
메모리	용량 (메모리칩 종류)	64GB
내장 디스크	1 개 사이즈 * 수량 (RPM)	73GB * 4 개 (15000)
네트워크연결	아댑터 수량 (대역폭), CAT6 케이블	Combo(2portFC+2port1000BT) 각 4 개 (1 Gbps)
운영체제	운영체제	HP-UX 11iV2
파티션	하드웨어파티션 2 파티션 이상 가능	하드웨어파티션 2

<표 1> 테스트 장비사양

## 4. 성능 평가 디자인

이 연구의 주요 목적은 아래 성능평가범위에 대한 다양한 구성의 하드웨어파티션과 소프트웨어파티션 에서 성능 영향을 비교 측정하는 것이다.

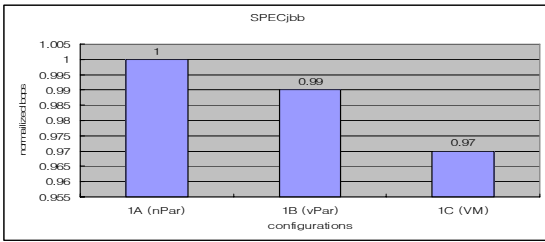
- 각 파티션 별 소프트웨어 스택
- 소프트웨어 파티션 안의 리소스 재배치
- 두 개의 운영체제 운영 시 워크로드 영향

하드웨어 파티션을 베이스 라인으로 설정 후 물리적으로 같은 양의 자원(CPU, 메모리, 입출력장치)으로 구성된 여러 소프트웨어파티션의 성능을 측정하였다. 성능 측정 I 에서는 하드웨어파티션과 소프트웨어파티션의 베이스라인 구성을 통해 파티션 별 성능평가를 하고, 성능 측정 II 에서는 하나의 소프트웨어파티션에 대하여 CPU 의 지역성(locality)변경 측정과 CLM 과 ILM 사용시의 성능 변화를 측정하고, 성능측정 III 에서는 두 개의 소프트웨어파티션 구성을 통한 실행 운영체제 증가에 따른 성능 변화를 측정하여 성능 측정 I II III 을 수치화 하였다.

### 4.1 성능 측정 I

최소 구성 테스트 (베이스라인 테스트)

- 1A : 하나의 nPar / 하나의 셀(8cores,16GB ILM[7,8])
- 1B : 하나의 vPar / 하나의 셀(8cores,16GB ILM)
- 1C : 하나의 VM / 하나의 셀(8cores, 16GB ILM)



<그림 7> 최소구성 테스트 측정결과  
각 파티션의 소프트웨어 스택의 오버헤드 측정결과  
vPars 는 1%, VM 은 3% 성능 저하를 확인하였다

4.2 성능 측정 II

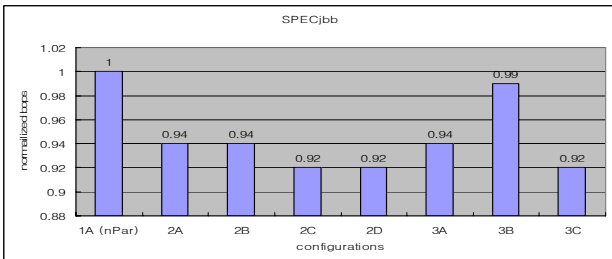
하나의 소프트 파티션

4.2.1 CPU 할당 위치 구성 변경 테스트

- 2A: 하나의 vPars / 두 개의 셀 nPars  
각각 셀 4cores, 16GB ILM
- 2B: 하나의 vPars / 두 개의 셀 nPars  
하나의 셀 8cores, 16GB ILM
- 2C: 하나의 VM / 두 개의 셀 nPars  
각각 셀 4cores, 16GB ILM
- 2D: 하나의 VM / 두 개의 셀 nPars  
하나의 셀 8cores, 16GB ILM

4.2.1 CLM 사용 성능 테스트

- 3A: 하나의 vPars / 두 개의 셀 nPars  
각각 셀 4cores, 12GB CLM[6,7], 4GB ILM
- 3B: 하나의 vPars / 두 개의 셀 nPars  
하나의 셀 8cores, 12GB CLM, 4GB ILM
- 3C: 하나의 vPars / 두 개의 셀 nPars  
8cores (입 출력 없는 셀), 12GB CLM, 4GB ILM

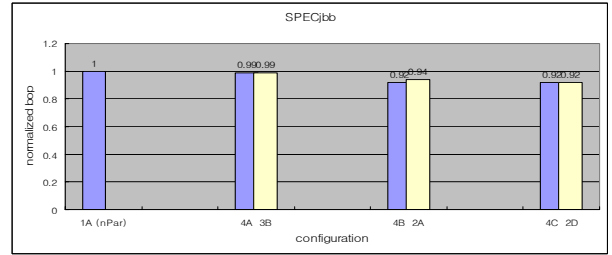


<그림 8> 싱글 소프트 파티션 테스트 측정결과  
여러 가지 CPU 의 지역성변경 측정 결과 베이스라인  
과 비교 시 최대 5% 성능 차이가 있고 입출력이 발생  
하는 셀에 CLM 을 사용하고 그 셀에서 CPU 를 모  
두 할당하여 사용할 경우 성능 차이 없음을 확인하였  
다.

4.3 성능 측정 III

두 개의 소프트파티션

- 4A: 두 개의 vPars / 두 개의 셀 nPars  
각각의 셀 8cores, 16GB CLM(입출력 셀)
- 4B: 두 개의 vPars / 두 개의 셀 nPars  
각각 입출력을 가진 cell 에 12GB CLM, 다른  
셀에서 8core, 4GB ILM
- 4C: 두 개의 VM / 두 개의 셀 nPars  
각각 셀 8cores, 16GB ILM



<그림 9> 두 개의 소프트파티션 테스트 측정결과  
두 개의 가상 파티션을 수행 후 하나의 가상 파티  
션에서 수행한 결과 값과 비교 시 성능 차이가 없었  
고 CPU 의 구성변경 테스트 시 2%의 성능 저하가 생  
긴다.

5. 결론 및 향후 연구

오늘날 가상화 시스템은 산업계에서 많이 사용되고 있다. 그로 인하여 올바른 가상화 기술 선택 및 구성을 위한 가상화 환경 성능 비교 평가는 중요한 이슈가 되었다. 본 논문에서는 실험을 통하여 소프트 파티션구성이 하드웨어 파티션구성보다 효율성 및 비용 절감에 유리한 구성방법임을 확인하였다. 가상화 기술 구현 시 하드웨어파티션 대비 소프트웨어 파티션 성능이 저하 되는데 본 연구를 통하여 가상화 기술의 구성 방법에 따라 가변적인 성능을 확인하였고 시스템 성능을 수치화 하였다. 실험을 통해 소프트파티션은 소프트웨어 스택 상의 오버헤드가 매우 적음을 알 수 있었고 셀 기반의 시스템에서 CLM 과 자원의 지역성(locality of resources)이 많은 성능 영향을 미치고 있음을 확인하였다. 향후에 메모리 할당 단위 변경에 따른 테스트, 동일 가상화 환경에서 실행 운영체제 개수 증가에 따른 성능 테스트, 더 많은 메모리를 필요로 하는 실 환경 어플리케이션, 많은 데이터베이스를 사용하는 실 어플리케이션 환경에서 가상화 기술의 성능평가실험을 확장해 보고 싶다. 또한 시스템 운영자들이 서버 통합을 위한 최적의 가상화 기술을 적용하는데 본 논문의 실험결과가 유용하기를 바란다.

참고문헌

- [1] T. Bittman, "The Future of server virtualization" Garther research, JULY 17, 2003
- [2] Norman Jouppi, "Enterprise IT Trends and Implications for Architecture Research" Symposium on High Performance computer Architecture 2005
- [3] James Owens, "Quantifying the performance Isolation properties of Virtualization Systems" 2006 USENIX
- [4] Daniel A. Menasce, "Virtualization: concept, Application, and Performance modeling" proc. 31th computer measurement group Conf. 2005
- [5] K.G. Shin, "Performance Evaluation of Virtualization Technologies for server consolidation, April 11, 2007 HPL-2007-59
- [6] HP nPars <http://docs.hp.com/en/5992-3386/index.html>
- [7] HP vPars <http://docs.hp.com/en/T1335-90083/index.html>
- [8] HP VM <http://docs.hp.com/en/T2767-90105/index.htm>
- [9] SPEC <http://www.spec.org/jbb2005>