

가상화 시스템의 성능 평가 방법

장 지 응[†] · 한 세 영^{††} · 김 진 석^{†††} · 박 성 응^{††††}

요 약

가상화 시스템은 시스템 활용률을 높이는 반면, 추가된 가상화 계층으로 인해 응용 프로그램의 성능 저하를 가져오므로, 본 논문에서는 두 가지 측면을 모두 고려하여 비 가상화 시스템과 성능을 비교할 수 있도록 하는 가상화 시스템의 성능평가 방법을 제안하였다. 가상화 시스템과 비 가상화 시스템을 비교할 수 있는 성능 비교 지표(metric)로 통합작업처리량 대비 시스템 낭비율을 정의하고, 가상화 시스템과 비 가상화 시스템에서 각각 시스템 낭비율과 통합작업처리량을 구할 수 있는 방법을 제시하였다. 또한 제안한 성능 평가 방법을 이용하여 다양한 응용 프로그램들을 운영하는 가상화 시스템과 비 가상화 시스템의 성능을 비교함으로써 본 연구에서 제시하는 가상화 시스템의 성능평가 방법이 적합함을 보이고, 더불어 응용 프로그램의 서로 다른 특성이 가상화 시스템에 미치는 영향을 분석하였다.

키워드 : 가상화, 성능 평가 방법

Performance Evaluation Methodology in Virtual Environments

Jiyong Jang[†] · Saeyoung Han^{††} · Jinseok Kim^{†††} · Sungyong Park^{††††}

ABSTRACT

Consolidating servers into a virtualized system increases entire system utilization, while suffers from performance degradation due to the additional virtualization layer. In this paper, we proposed a performance evaluation methodology for comparing virtualized systems with native non-virtualized systems. We defined a system waste rate per consolidated throughput as a metric, and described the method for calculating system waste rate and consolidated throughput for both of virtualized systems and non-virtualized systems. Using the proposing performance evaluation methodology, we established testbeds, evaluated their performance, and compared the metrics of both systems. As a result of the evaluation, we could show the appropriateness of our methodology and analyze the effect of the application characteristics.

Keyword : Virtualization, Performance Evaluation

1. 서 론

최근 가상화 기술을 이용한 컴퓨팅 시스템의 보안, 성능 최적화, 시스템 관리, 시스템 자원 사용의 효율성 증대 등의 측면이 사용자들의 다시 주목을 받고 있다 [1]. 또한, 컴퓨팅 자원의 성능은 기하급수적으로 증가하는데 비해, 그 한계에 훨씬 못 미치는 자원 활용이 이루어지고 있으므로, 가상화 기술을 통해 그 자원 활용을 높이는 여러 가지 응용이 선보여지고 있다. 특히 CPU 활용률이 낮은 다수의 시스템을 하나의 시스템으로 통합하여 단일 시스템의 CPU 활용률을 높이는 방안은 가상화 기술이 보여주는 중요한 장점이자 특징

중의 하나라고 할 수 있다. 따라서 가상화 기술이 지닌 가장 큰 장점중의 하나인 자원 활용의 유연성과 효율성 증가라는 측면을 부각하여, 가상화 시스템의 성능을 평가하는 많은 시도들이 있었다[2, 3, 4, 5, 12, 13, 14]. 그러나 이러한 시도들은 특정 자원들의 효율성 또는 성능을 높이는 면만 부각하였을 뿐 가상화 기술의 적용으로 인한 오버헤드의 발생이나 전체 성능의 저하 등은 반영하지 못하였다.

CPU 활용률만을 보고 가상화 시스템이 비가상화 시스템에 비해 성능이 증가했다고 할 수 없으며, 또한 반대로 처리 속도나 처리량(throughput) 만을 보고 가상화 시스템의 성능이 통합하기 전의 비가상화 시스템보다 단순히 나빠졌다고 할 수도 없다. 이렇게 가상화 시스템과 비가상화 시스템의 성능 비교가 모호한 이유는, 가상화 시스템의 성능 평가 방법이 아직 규격화 되어있지 않기 때문이다. 즉, 시스템의 성능 평가를 위한 워크로드가 단일 시스템을 가정으로 한 것이 아닌 다수의 운영체제 및 어플리케이션을 기반으로

† 정 회 원: 삼성전자 정보통신연구소 연구원
†† 정 회 원: 서강대학교 컴퓨터학과 박사과정
††† 정 회 원: (주)NHN 플랫폼서비스개발팀 개발자
†††† 정 회 원: 서강대학교 컴퓨터학과 부교수
논문접수: 2007년 11월 30일
수 정 일: 2008년 3월 7일
심사완료: 2008년 5월 6일

한 것이어야 하는데, 기존에 존재하는 벤치마킹 툴은 단일 하드웨어와 단일 운영체제를 가정하고 워크로드를 생성하고 시스템의 성능 평가를 수행하기 때문에 가상화 시스템의 성능 평가의 도구로서 적합하지 않다.

본 논문에서는 가상화 시스템의 성능을 기존의 비 가상화 시스템과 비교할 수 있는 방법적인 연구를 수행하고, 실험을 통하여 그 적합성을 보이고자 한다. 이를 위해서 우선 가상 머신간의 통신이나 네트워크의 구성, 그리고 가상 머신 위에서 가동되고 있는 응용 프로그램의 종류에 따라 크게 네 가지 가상화 시스템을 구분하고, 그 중 가장 많이 채택되고 있는 중소기업 모델과 호스팅 모델을 성능 시험의 대상으로 설정하여 그 구성에 알맞게 실험 환경을 만들었다. 또한 앞서 제시한 가상화 시스템에서 야기되는 성능 평가의 모호성을 제거하기 위하여, 시스템 자원의 활용도와 벤치마크 어플리케이션을 사용하여 얻은 작업처리량(Throughput)을 이용하여 통합작업처리량 대비 CPU 낭비율(W_c^l)이라는 새로운 성능 지표를 정의하였다. 통합작업처리량 대비 CPU 낭비율은 가상화 시스템이 비가상화 시스템에 비해 지니고 있는 장점 중의 하나인 시스템 자원의 활용률을 얼마나 높이는 가를 평가하는 동시에, 비가상화 시스템 대비 작업처리량(Throughput)을 비교함으로써 가상화 시스템이 지니고 있는 전체적인 성능을 나타내는 지표로 이용하였다.

가상화 시스템의 성능을 평가하는 실험환경을 조성하기 위하여 본 논문에서는 의사 가상화(Para-Virtualization) 기술로 구현한 응용 프로그램인 Xen을 기반으로 한 가상화 시스템을 구성하였다. 동일한 성능을 가진 가상화 시스템과 비가상화된 시스템을 구성하고 앞서 정의한 통합작업처리량 대비 CPU 낭비율을 이용하여 비가상화 시스템 대비 가상화 시스템의 성능을 실시간으로 모니터링하고 이를 분석하였다. 본 논문에서 별도로 성능 평가를 수행하지는 않았으나, VMWare 등과 같이 Binary translation을 기반으로 하는 가상화 시스템의 경우도, 같은 구성과 방법으로 그 성능을 비가상화 시스템 대비 가상화 시스템의 성능을 비교하고 평가할 수 있을 것으로 기대한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련 있는 기존의 연구 사례를 살펴보고, 3장에서는 현재 가상화 시스템을 네 가지 형태로 분류하여 정의하였다. 4장에서는 성능 평가의 기준으로 사용되는 통합작업처리량 대비 CPU 낭비율을 정의하고 이를 적용하는 방법에 대해 설명하였다. 5장에서는 3장과 4장에서 정의한 가상화 모델과 성능 평가 방법에 근거하여 실험 환경을 구성하고 이를 평가하여 가상화 시스템의 벤치마킹 도구로서의 적합성을 검증하였다. 마지막으로 6장에서는 본 논문의 결론과 향후 과제에 대해 논의하였다.

2. 관련 연구

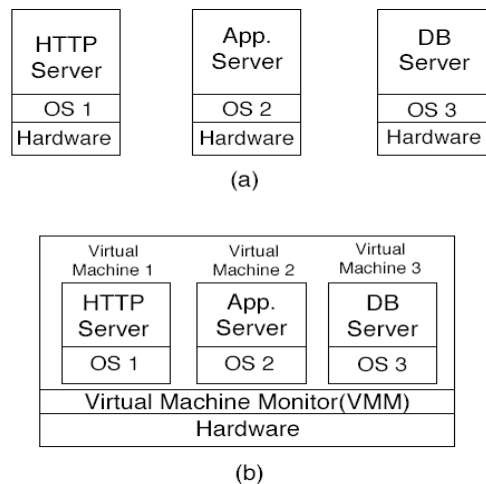
가상화 시스템과 비가상화 시스템의 성능을 비교하기 위

한 연구로, 가상화 시스템을 이론적으로 모델링 하고 가상화를 위한 오버헤드를 수학적으로 분석한 기존연구가 있다 [6]. 가상화 시스템의 오버헤드를 정의하고, 큐잉 네트워크의 수학적 모델링을 기반으로 가상화 시스템의 성능을 수치화 하여, 간단한 서버 통합 예제를 통해 가상화 시스템의 성능 모델링을 어떻게 적용하는지를 보여주고 있다.

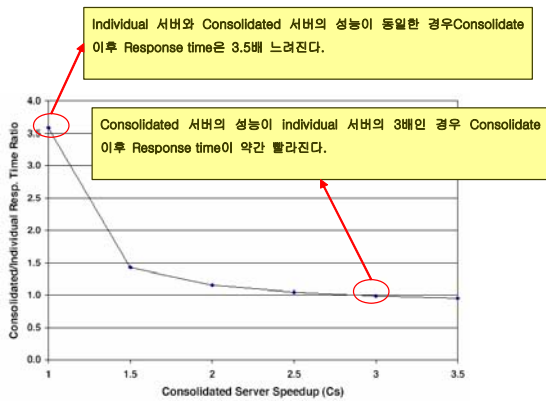
큐잉 네트워크 모델로 가상화 시스템을 모델링 하고, 입력 파라미터로서 워크로드 부하와 서비스 요청 값을 입력하여 그 결과로 응답시간이 계산할 수 있는데, 이를 시스템의 성능으로 정의한다. 한편, 가상화를 위한 추가 계층에 기인한 오버헤드를 성능 저하의 요인인 슬로우 다운(slowdown)으로 정의하고, 슬로우 다운 요소를 반영한 가상화 시스템의 서비스 요청 량을 큐잉모델을 이용하여 최종적으로 계산한다.

성능 평가의 예제를 나타내기 위해서 (그림 1)과 같이 비 가상화 시스템을 하나의 가상화 시스템에 통합시키는 환경을 구축하고, 서비스 요청 량에 따른 응답시간을 계산하여, 실험 결과를 (그림 2)의 도표로 나타내고 있다. (그림 2)에서 그래프의 x-축은 통합할 서버의 성능이 통합하기 전 개별 서버의 성능에 비해 몇 배가 좋은지를 나타내고, y-축은 개별서버 환경에서의 응답시간 대비 통합서버환경에서의 응답시간 비율을 나타낸다. (그림 2)에 따르면, 3개의 개별 서버들이 같은 성능을 가진다고 하면, 같은 성능을 가진 서버에 가상화 시스템을 구축하고 3개의 개별 서버들을 통합했을 경우, 응답시간이 약 3.5배 느려지고, 개별 서버의 세 배의 성능을 가진 서버로 통합하는 경우에는 전체적인 응답시간이 약간 빨라질 것으로 예측하고 있다.

위의 연구 [6]의 경우, 수학적인 시스템 모델링을 기반으로 하여 가상화 시스템과 비 가상화 시스템을 비교하므로, 가상화 시스템과 비가상화 시스템의 성능을 비교하기는 하지만, 그 성능을 측정할 수 있는 방법을 제시하고 있지는 않다. 따라서 구체적으로 가상화 시스템 특성을 고려하여 그 성능을 평가할 수 있는 방법이 필요하다.



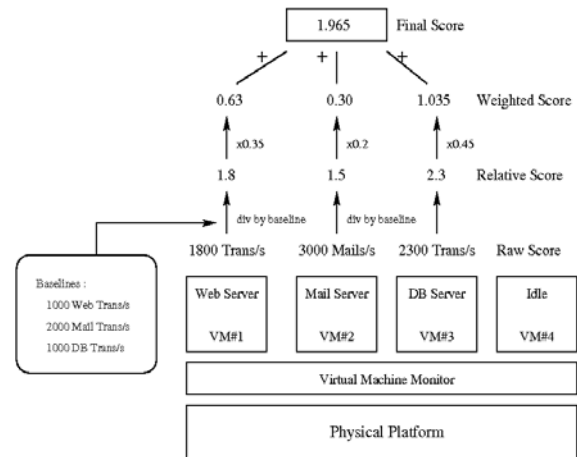
(그림 1) 가상화 전의 시스템 환경과 가상화 후의 시스템 환경



(그림 2) 가상화 시스템의 성능 비교 결과

규격화된 가상화 시스템의 성능 평가에 대한 사용자들의 요구들을 수용하고, 가상화 시스템의 성능을 측정하는 구체적인 방법으로 vConsolidate이 제안되었다 [7]. 단순한 통합을 목적으로 하는 대부분의 가상화 시스템은 하나의 하드웨어에서 구동하고 있는 각 가상 머신이 서로 다른 응용프로그램을 운영하고 있는 경우가 많은데, 이때 이들의 성능 측정값을 어떻게 하나로 통합할 것인지에 대한 문제를 해결하기 위한 방안을 제시하고 있다. 즉, vConsolidate에서는 가상화 하기 전 동일한 시스템 사양을 가지고 기준이 되는 성능 값을 측정하고, 가상화 한 이후 각 가상머신 상에서 같은 성능 값을 측정 후, 구한 성능 데이터를 이 기준 측정값으로 나누어 각 응용 프로그램에서의 성능 측정값을 표준화 (Normalization)하여 비교성능점수를 구한다. 마지막으로 가상화 시스템의 성능을 나타내는 최종적인 지표를 구하는 방법으로 각각의 응용 프로그램에서 얻어진 비교성능점수들을 미리 정의한 가상머신별 가중치를 곱하여 하나의 값으로 환산한다.

(그림 3)에서 이 예제를 보여주고 있다. 즉, 한 데이터 센터에서 웹 서버, 이메일 서버, 데이터베이스 서버를 하나의 가상화 된 시스템으로 통합하려고 하고, 이 데이터 센터에서 운영하는 서버의 이용률과 중요도에 따라 미리 각각의 서버에 0.35, 0.2, 0.45의 가중치(Weight Factor)를 부여했다고 가정하자. 먼저, 각 서버의 성능 기준 측정값을 구하기 위해 가상화 기술이 적용되지 않은 비가상화 시스템에서 각 서버에 적합한 벤치마크 테스트를 수행하고, 그 얻어진 결과 값이 1000 Web trans/sec, 2000 Mail trans/s, 1000 DB trans/sec 라고 하자. 이 때 동일한 벤치마크 테스트를 가상화 기술로 통합된 서버에서 각 가상 머신별로 수행했을 때 얻어진 결과가 1800Web trans/sec, 3000 Mail trans/sec, 2300 DB trans/sec 이라면, 비가상화 시스템에서 얻어진 성능 기준 측정값으로 표준화 시킨 비교 값은 1.8, 1.5, 2.3이 된다. 이제 이 값들에 미리 정의된 가중치를 곱하여 하나의 값으로 환산하면 1.965라는 결과를 얻게 되고, 이를 가상화 시스템의 성능을 나타내는 지표로서 사용할 수 있다. 같은 방법을 다른 가상화 시스템에 동일하게 적용하여 가상화 시스템 간의 성능 분석을 할 수 있다.



(그림 3) vConsolidate 성능 시험 결과 계산 방법의 예제

vConsolidate에서는 여러 개의 가상머신의 성능을 의미를 가진 가중치 값을 통해 하나의 성능 지표로 나타내는 방법을 제안하면서 가상 머신의 성능을 표준화하기 위한 기준으로 비 가상화 시스템의 성능을 사용하므로, 동일한 하드웨어와 응용프로그램을 가진 서로 다른 가상화 기술들의 성능을 비교하는 방안으로 적합하지만, 가상화 시스템과 비가상화 시스템의 성능을 비교하기 위한 방법으로는 부적합하다.

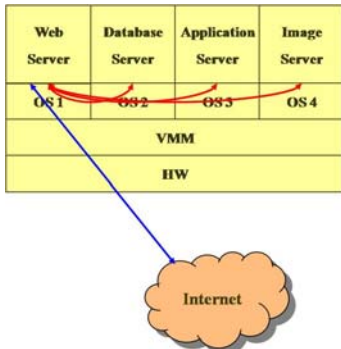
따라서 본 논문에서는 시스템의 가상화에 기인한 시스템의 활용도 증가와 응용 프로그램의 성능인 작업처리량의 저하, 두 가지 측면을 모두 고려하여, 가상화 시스템과 비 가상화 시스템의 성능을 비교할 수 있는 구체적인 성능 평가 방법을 제시하고자 한다.

3. 가상화 시스템의 분류와 정의

본 장에서는 현재 데이터 센터를 중심으로 가장 많이 응용되고 있는 가상화 시스템의 구성과 적용 사례를 기반으로 네 종류의 가상화 시스템 타입, 즉 엔터프라이즈(Enterprise) 타입, 중소기업 타입, 호스팅 타입, 클러스터 타입으로 분류하고 정의한다.

3.1 엔터프라이즈(Enterprise) 가상화 모델

엔터프라이즈 타입의 가상화 시스템은 서로 데이터 교환을 위한 통신이 필요한 서로 다른 응용 프로그램들을 각 가상 머신에 구축하여 통합한 시스템이다 (그림 4). 주로 전자상거래(e-Commerce) 서버들이나 대기업의 데이터베이스 서버와 그 데이터베이스 서버의 데이터를 이용하는 응용 프로그램 서버들을 하나의 시스템으로 통합한 경우가 여기에 속한다. 이 타입의 가상화 시스템의 경우는 가상 머신 사이에 통신이 많고 서로의 성능에 영향을 미치므로 성능 평가 수행 시 오버헤드와 시스템 지연 시간으로 인한 장애가 발생할 수 있다.

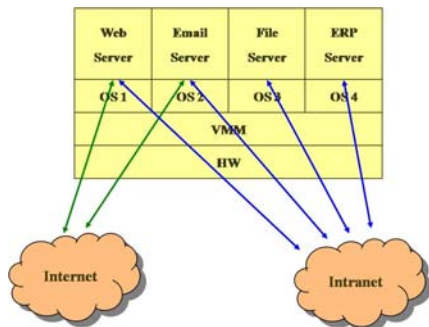


(그림 4) 엔터프라이즈 타입 가상화 시스템

3.2 중소기업형(SMB-Small and Business) 가상화 모델

두 번째 가상화 시스템의 유형은 중소기업 타입의 가상화 시스템이다. 각 가상 머신이 서로 다른 응용 프로그램을 운영하지만, 그들 사이에 데이터 교환을 위한 통신이 거의 없는 시스템이다 (그림 5). 즉, 엔터프라이즈 타입과 마찬가지로 각 가상 머신이 서로 다른 응용 프로그램을 운영하지만 각 응용 프로그램들 간의 연관성이 적어서 데이터 교환 등의 내부 통신이 외부와의 통신에 비해 상대적으로 적은 시스템이다.

규모가 크지 않은 기업 현장에서 웹 서버, 메일 서버, 파일 서버, 회계 관리 나 판매 재고 관리 등을 위한 다양한 서버 군을 필요로 할 때, 이를 위하여 각각 별도의 서버를 구축하는 것은 공간적인 면에서 그리고 재정적인 면에서 부담이 될 수 있다. 이런 경우 필요한 서버들을 하나의 서버로 통합하여 가상화 시스템을 구축하면 기업의 재정적인 부담을 줄임과 동시에 기업에서 필요로 하는 시스템을 구축할 수 있게 된다. 비록 이러한 구성의 가상화 시스템이 중소기업에서 상당 부분 응용되고 있는 것은 사실이지만, 현재 구축 사례로 나와 있는 대부분의 가상화 시스템이 이 구성을 채택하고 있다.

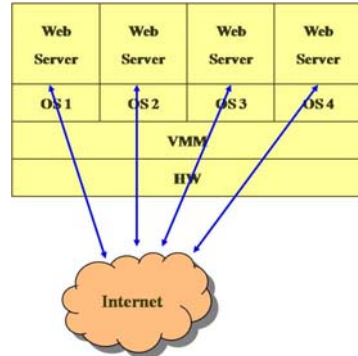


(그림 5) 중소기업 타입 가상화 시스템

3.3 호스팅(Hosting) 가상화 모델

세 번째로 호스팅 타입의 가상화 시스템은 웹 호스팅 서버나 데이터베이스 호스팅 서버를 가상화 기술을 활용하여 구축하는 것으로, 같은 종류의 응용 프로그램들을 하나의 가상화 시스템에서 운영하는 것이다 (그림 6). 이 타입에서도 가상 머신들 사이의 통신은 거의 없고 가상 머신에서 운

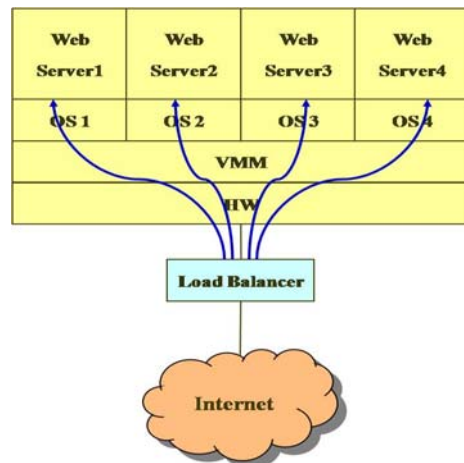
영되는 서버와 외부 시스템 사이의 통신이 주를 이룬다. 비록 서버 구성의 차이로 호스팅 타입의 가상화 시스템을 따로 구분을 했으나, 서버간의 통신이 적다는 점과 서버 어플리케이션의 종류가 상이함을 제외하고는 넓은 의미에서 중소기업 타입의 가상화 시스템에 포함시킬 수 있다.



(그림 6) 호스팅 타입 가상화 시스템

3.4 클러스터(Cluster) 가상화 모델

용량이 큰 워크로드를 같은 작업을 처리할 수 있는 동일한 컴퓨팅 시스템에 분산시킴으로써 작업의 효율성과 처리 시간 단축을 위한 목적으로 사용되는 시스템 구성이 클러스터링인데, 가상화 시스템 에서도 이러한 목적으로 구현되는 형태를 고려할 수 있다. 클러스터 타입의 가상화 시스템은 같은 종류의 응용을 통합한다는 점에서 호스팅 타입의 가상화 시스템과 비슷하게 보일 수 있지만, 클러스터 타입의 가상화 시스템에서 각 가상 머신들은 같은 종류의 응용프로그램을 운영할 뿐만 아니라 동일한 서비스를 제공하기 때문에 다른 작업을 개별적으로 수행하는 호스팅 타입의 가상화 모델과는 차이가 있다. (그림 7)에서와 같이 웹 서비스를 예로 들면, 하나의 웹 서비스를 제공하는 여러 대의 웹 서버들을 가상화 시스템으로 통합하고, 중간에 부하분산기(Load Balancer)를 위치하여 클라이언트로부터 웹 서비스에 대한



(그림 7) 클러스터 타입 가상화 모델

요청이 있을 때, 하나의 서버로 운영하기 어려운 부하를 분산 시키는 구성을 적용해 볼 수 있다.

그러나 일반적인 클러스터 서버는 서버 한대의 용량(Capacity)을 초과하는 대 용량의 워크로드를 여러 대의 서버에 분산하여 처리하기 위한 목적으로 구성하므로 여러 서버에 작업을 분산시키기 위하여 구성된 시스템을 다시 하나의 서버로 통합하는 것은 의미 없는 구성으로 남을 수도 있다.

현재 가상화 시스템의 주 용도는 기존 응용프로그램에 대한 통합에서 중간 계층(Mid-tier) 응용프로그램의 통합으로 넘어가는 과정에 있다. 이는 앞서 정의한 가상화 시스템의 유형으로 보면 중소기업 타입과 호스팅 타입이 섞여 있는 것이다. 앞서 언급한 바와 같이 호스팅 타입은 중소기업 타입의 일부로 볼 수 있으므로, 논문에서는 중소기업 타입의 가상화 시스템을 기반으로 성능 평가를 정의하고 실험하였다.

4. 성능 시험 방법 정의

본 장에서는 가상화 시스템의 성능 평가에 필요한 제반 요소인 가상화 시스템의 성능 평가 지표(Metric)를 먼저 정의하고, 이 성능 지표를 적용할 수 있는 가상화 시스템의 실험 환경을 3장에서 정의한 가상화 시스템의 구성 중 중소기업 타입에 따라 구현하였다. 또한, 실제 가상화 시스템에 적용할 수 있는 워크로드를 디자인하고 이 워크로드를 가상화 시스템에 적용했을 때 얻어지는 결과를 수치화하는 과정을 나타내었다.

4.1 성능 지표(Metric) 정의

4.1.1 시스템 자원 활용률(System Resource Utilization)

본 논문에서는 시스템 활용률을 나타내기 위한 시스템의 자원으로서의 가장 대표성을 가진 CPU를 평가의 대상으로 삼았다. 이는 실제 가상화 시스템의 적용 사례에 따른 가상화 기술을 적용했을 경우 시스템의 성능 및 작업 처리 속도를 가장 많이 좌우하는 시스템 자원은 CPU로 알려져 있기 때문이다 [11]. 비가상화 시스템에서 가장 많이 낭비되어지는 시스템 자원 역시 CPU이고, 시스템의 자원 중 가장 수치적인 정보를 가장 쉽게 알아내고 이용할 수 있으므로, 본 논문에서 새로운 성능 지표를 정의함에 있어서 CPU 활용률을 시스템 활용률을 대체하는 용어로 삼았다.

따라서 비 가상화 시스템의 시스템 활용률은 각 시스템의 CPU 활용률의 평균이고, 이들 시스템을 하나의 시스템으로 통합하였을 경우, 그 가상화된 시스템의 시스템 활용률은 각 가상머신의 CPU 활용률의 합이 된다.

4.1.2 단위 시간당 작업 처리량(Throughput)

시스템의 성능을 평가하는 대부분의 성능 평가 도구(Benchmarks)는 그 성능 평가 도구가 생성하는 워크로드에

대하여 얼마나 빠른 속도로 또 얼마나 많은 작업량을 수행할 수 있는가로 그 성능 평가의 기준으로 삼고 있다. 다수의 운영체제 및 응용프로그램이 단일의 하드웨어에서 구동되는 가상화 시스템의 경우 어떠한 워크로드를 적용하여 성능을 평가할 것인가에 대한 문제가 발생할 수 있다. 이러한 문제 제기는 자연스럽게 가상화 시스템에 적합한 워크로드의 정의라는 주제로 전환될 수 있는데, 본 논문에서는 가상화 시스템에서 운영되는 각각의 가상 머신의 특징에 따라 해당 가상머신을 평가하는 벤치마크를 선정하여 각각의 성능 평가 결과를 통합함으로써 가상화 시스템의 성능을 측정하는 또 하나의 지표로 삼기로 한다.

앞서 가상화 시스템의 성능 지표의 하나로써 시스템 활용률 또는 CPU 활용률을 사용하겠다고 밝혔다. 시스템의 활용률과 더불어서 그 시스템이 나타내는 작업 처리의 성능(Throughput)을 함께 제시해야만 객관적인 차원의 성능 비교가 일어날 수 있는데, 이것을 뒷받침 하는 예로 다음을 제시할 수 있다. 5%의 CPU 활용률을 보이는 NFS 서버와 10%의 CPU 활용률을 보이는 웹 서버가 있다고 하자. 두 대의 서버는 비가상화 서버이며 각각 1.6 records/sec, 1000 replies/sec의 처리량을 보이고 있다. 이 두 서버를 통합하여 가상화 시스템을 구축하면 CPU 활용률은 30%로 증가하지만 각 서버의 처리량은 가상화 오버헤드로 인하여 0.5 records/sec, 900 replies/sec로 감소할 수 있다. 이 경우에 비가상화 시스템에 비해 가상화된 시스템은 CPU 자원의 높은 활용률을 가지게 되었으나, 반대로 시스템에 생기는 부하로 인하여 단위 시간당 작업처리량은 저하된 것을 확인할 수 있다. 이처럼 자원의 활용률 내지 단위 시간당 작업 처리량만으로 시스템의 성능을 비교할 경우 수치적인 우열 관계를 단순히 비교할 수 있으나, 시스템의 통합과 가상화에 따르는 장점과 단점을 포함하는 비교는 할 수 없음을 주시해야 한다.

따라서 본 논문에서는 CPU의 활용률과 더불어 벤치마크 어플리케이션을 이용한 시스템의 작업 처리 성능을 동시에 계산하고 두 변수를 통합함으로써 가상화 시스템의 성능을 평가하는 지표로 정의한다.

4.1.3 성능 지표 정의

본 논문에서는 통합 작업 처리량 대비 CPU 낭비율(W_c^*)이라는 새로운 성능 지표 단위를 정의함으로써 가상화 시스템과 비가상화 시스템의 성능 비교를 수치적으로 환산하여 적용하도록 하였다. 여기서 CPU 낭비율은 시스템에서 사용되지 않은 CPU 자원을 뜻하고, 통합 작업 처리량은 각 서버에서 벤치마크 응용프로그램을 통해서 측정된 성능 결과값들을 하나로 통합한 값이다. W_c^* 의 값을 식으로 표현하면 다음과 같다.

$$W_c^* (\text{통합작업처리량 대비 CPU 낭비율}) = \frac{100 - \text{CPU Utilization}_{\text{통합}}}{\text{Throughput}_{\text{통합}}} \quad (1)$$

사용되지 않는 CPU자원의 양을 $100 - CPU\ Utilization_{\text{통합}}$ 을 C_w 로, 통합작업처리량($Throughput_{\text{통합}}$)을 T_c 로 각각 표기하였다. 이 때 $CPU\ Utilization_{\text{통합}}$ 은 앞에서 언급한 시스템 활용률을 뜻한다. 비가상화 시스템의 통합작업처리량은 가상화 시스템이 도달할 수 있는 성능의 최대치에 해당한다는 가정 하에 비가상화 시스템의 통합작업처리량 T_c^{native} 을 100으로 놓고, 가상화 시스템의 통합작업처리량 $T_c^{virtual}$ 은 이의 비교 수치로 환산하여 구하도록 하였다. 아래의 수식으로 가상화 시스템의 통합작업처리량이 환산되는 과정을 표현하였다.

$$T_c^{native} = 100 \tag{2}$$

$$Weight_k = \frac{100/n}{T_k^{native}} \tag{3}$$

$$T_c^{virtual} = \sum T_k^{virtual} \times Weight_k \tag{4}$$

(식 3)에서 k번째 서버 또는 k 번째 가상머신의 가중치 W_k 는 다른 성능 지표를 가진 개별 서버의 성능을 표준화(normalization)하여 하나의 값으로 통합하기 위해 구한 값이다. 본 논문에서는 각각의 서버가 가지는 중요도가 모두 같다는 가정 하에, n개의 서버가 있다면, 각 서버는 100 중 $100/n$ 의 통합처리량을 갖도록 한다. 일단 비가상화 시스템에서 성능 평가를 통해 (식 3)과 같이 가중치 W_k 가 구해지면, (식 4)에 이 값을 적용하여 가상화 시스템의 통합처리량 $T_c^{virtual}$ 를 구할 수 있다. 최종적으로 비가상화 시스템과 가상화 시스템의 통합작업처리량 대비 CPU 낭비율(W_c^*)을 구하는 방법은 (식 5)와 같다.

$$W_c^* = \frac{C_w}{T_c} \tag{5}$$

위 식에서 정의한 C_w 와 T_c 값을 구하기 위해 CPU 활용률을 통합하는 방법과 벤치마크 결과를 이용해 얻은 작업처리량(Throughput)을 통합하는 방법을 다음 절에서 제시하도록 하겠다.

4.2 성능 시험 방법 정의

4.2.1 워크로드 생성기 (Workload Generator)

위에서 언급한 각각의 가상머신에 대응하는 단위 시간당 작업 처리량(Throughput)을 구하기 위해서는 워크로드 생성기(Workload Generator)의 선정이 필요하다. 즉, 구성된 가상화 환경에서 게스트 운영체제 위에 각각의 서버를 구축한 뒤, 서버의 성능을 평가하는 벤치마크를 이용하여 단위 시간의 작업 처리량을 구하도록 한다. 본 실험에서는 웹서버, NFS 파일서버, 데이터베이스 서버를 실험적으로 운용할 것이며 각 서버의 성능 평가는 현재 가장 일반적으로 서버의 성능을 평가하는 도구로 이용되고 있는 *httpperf*, *IOzone*,

*MySql-bench*를 사용하였다. 이 벤치마크들의 특징은 통신 분산 시스템 환경(서버-클라이언트)에서 세션(Session)을 형성한 후 워크로드를 서버에 적용하여 서버의 작업 처리 능력을 평가하는 기본 틀을 가지고 있어, 각 서버의 작업처리량(throughput)을 제공한다.

각각의 워크로드 생성기는 서버의 특성에 따라 그 성능을 평가할 수 있도록 구체적인 연산 및 시스템 부하에 관한 정책을 가지고 있다. *httpperf* [8]의 경우 웹서버의 성능 평가를 위해 다양한 HTTP 워크로드를 생성하는데, 본 실험에서는 서버와 맺게 되는 세션의 총 수와 단위 시간(sec)당 연결 속도를 정해 줌으로 해당 서버의 처리 능력을 평가하도록 하였다. 그 성능의 결과는 단위 시간당 서버의 응답 속도(reply/sec)로 나타나게 된다. 예를 들어 서버와 600000(Connection)개의 통신 연결(Session)을 만들기를 원하고 단위 시간(sec)당 1000(rate)개의 연결을 맺도록 요청했을 경우, 만일 서버가 무리 없이 이 요청에 대해 서비스를 수행한다면 대략 10분간의 성능 평가가 이루어지게 될 것이다.

*IOzone*은 다양한 파일 시스템을 평가 할 수 있도록 고안된 벤치마크 어플리케이션이다 [9]. 본 실험에서는 분산 네트워크 환경에서 파일 시스템의 성능 평가를 시험해 보기 위해 NFS 서버를 구축하고 클라이언트 컴퓨터에서 서비스 요청을 할 수 있도록 하였다. 특히 Xen을 비롯한 다양한 가상화 시스템 지원 환경에서 디스크 입출력 병목현상으로 인한 성능 저하가 뚜렷하다는 점을 확인하기 위해 *IOzone* 벤치마크를 사용하기로 하였다. *IOzone*은 파일 시스템을 테스트 해볼 수 있도록 다양한 형태의 명령어를 제공하는데, 그 중에서 가장 부하가 많은 디스크 쓰기(Disk Write/Rewrite) 명령어를 통해서 그 성능을 살펴보도록 하였다. 특히 작업의 부하를 높이기 위해서 O_SYNC(Write Through) 옵션을 주고 테스트를 수행하였다. *IOzone*은 주어진 파일 사이즈를 레코드(Record)단위로 쪼개어서 읽고 쓰기를 반복하는 방식으로 작업을 처리한다. 본 실험에서는 다른 벤치마크 응용 프로그램의 평가 수행 시간에 맞추기 위해 파일의 크기와 레코드의 크기를 각각 9600MB과 16MB로 설정하였다. 수행 결과는 주어진 파일의 크기와 레코드의 크기에 따라 명령어가 종료되는 시간으로 나타나게 되는데, 파일 크기를 그 시간으로 나누어, Bytes/sec의 작업처리량(throughput)의 형태로 그 결과를 이용한다.

마지막으로 데이터베이스 서버의 성능을 평가하기 위해 사용한 벤치마크는 *MySql-bench*이다. *MySql-bench*역시 서버의 성능 평가를 위해 테이블 생성, 삭제, 레코드 입력, 삽입, 삭제 등과 같은 SQL 명령어를 이용한 다양한 모음으로 구성되어 있다 [10]. 본 논문에서는 여러 명령어 중에서 비교적 짧은 수행 시간을 통해서 결과를 나타내 줄 수 있는 *test-wisconsin*을 사용하였다. 두개의 데이터베이스 테이블을 생성한 후 31000개의 레코드를 삽입하고 다시 이를 삭제하면 모든 수행이 종료되는 프로세스를 지니고 있는데, 한번의 수행으로 한 번의 결과 출력이 이루어지는 특징으로 인하여 여러 번의 반복수행을 통해 그 성능을 측정하도록

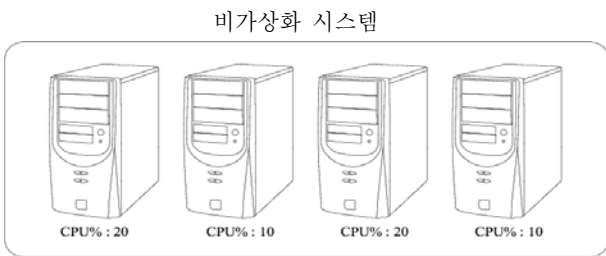
하였다. 이 벤치마킹 각 테스트 세트의 수행시간이 그 결과로 얻어지는데, 이를 시간으로 나누어 test/min 이라는 통합 처리량(throughput)의 형태로 변환하여 이용한다.

4.2.2 통합 CPU 활용률 생성 방법

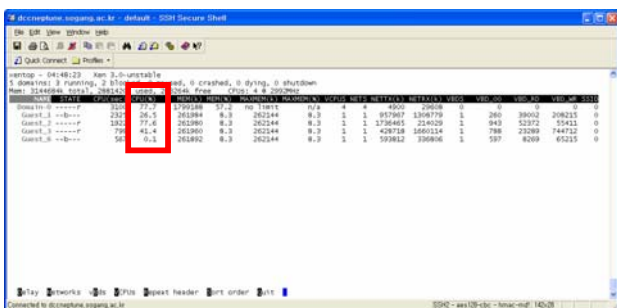
비가상화 시스템의 CPU 활용률의 통합은 운영체제에서 제공하는 시스템 정보를 이용하여 간단히 그 평균값을 취함으로써 값을 구하도록 하였다 (그림 8). 실험에 사용되는 모든 서버는 그 운영체제로 LINUX를 사용하고 있는데, 리눅스 파일시스템의 root/proc 디렉터리 밑에 stat 파일은 프로세서의 다양한 정보를 수치화하여 담고 있다. 이 파일의 정보를 사용하여 비가상화 시스템의 각각의 서버에 대한 CPU 활용률을 구하도록 한다.

가상화 시스템의 CPU 정보는 가상 머신 모니터 프로그램 Xen에서 제공하는 편리한 시스템 유틸리티인 Xentop을 이용하여 구하도록 하였다. Xentop은 가상 머신들의 시스템 정보와 각종 활동 상황에 대해서 모니터링(Monitoring)하는데, 각 가상 머신이 사용하는 CPU의 활용률 및 전체 시스템의 CPU 활용률에 대한 정보도 표시해준다 (그림 9). 본 논문에서는 Xentop의 소스를 수정하여 필요한 정보들을 본 실험을 위해 개발한 모니터링 프로그램에 직접 실시간으로 송신해 주도록 하였다.

이렇게 통합된 CPU 활용률을 통하여 <식 1>과 <식 2>에서 언급된 CPU 낭비율을 계산할 수 있다.



(그림 8) 비가상화 시스템의 시스템 활용률



(그림 9) Xentop 유틸리티 실행 화면의 예

4.2.3 통합작업처리량(T_c) 생성 방법

서로 다른 종류의 벤치마크 응용프로그램으로 성능을 측

정할 경우 성능을 나타내는 지표, 즉 작업처리량의 단위가 달라진다. 예를 들어 웹서버의 처리량의 단위가 단위 시간(sec)당 응답 수(reply/sec)인 반면 NFS 서버의 작업 처리량의 단위는 단위 시간(sec)당 바이트 수(bytes/sec) 내지는 단위 시간(sec)당 레코드 수(records/sec)등과 같을 수 있다. 통합작업처리량(T_c)을 구하기 위해서는 이렇듯 서로 다른 성능 지표를 가지고 있는 값들을 하나로 묶는 방법이 필요하게 된다.

본 논문에서는 성능 비교의 대상이 동일한 하드웨어 사양을 가지고 있는 비가상화 시스템 군(群)과 가상화 시스템이라는 점과 이 둘을 동일한 벤치마크 응용프로그램으로 성능 측정한다는 점을 착안하여 서로 다른 성능 지표(Metric)를 가진 값들을 표준화(Normalization)함으로써 통합된 하나의 값을 산출하도록 하였다. 통합작업처리량을 산출하는 과정에서 가중치를 통하여 표준화 작업이 이루어지게 되는데, 여기서 가중치를 결정하는 방법으로 비가상화 시스템의 통합작업처리량이 가상화 시스템이 다다를 수 있는 성능 결과 값을 초과할 수 없다는 점을 착안하여 값을 구하도록 한다.

<표 1>은 본 논문에서 가중치를 구하는 방법을 보여주는 하나의 예이다. 비가상화 시스템 환경에 4대의 서버가 설치되어 있다고 가정하고 가상화 시스템에서는 이 서버들을 4개의 가상머신을 구동하여 운영한다고 하자. 4대의 서버에서 산출되는 모든 성능 값의 비율을 100으로 놓고 각각의 서버들이 동등한 비중을 가지고 있다고 할 때 각각의 서버들이 가지게 되는 작업처리량의 최대치는 25가 될 것이다.

만약 실험군의 서버의 개수가 둘이라면 작업처리량의 최대치는 50, 여덟 대의 서버라면 12.5 등의 값을 가지게 된다. 여기서 작업처리량은 평균작업처리량(총 작업처리량을 실험 시간으로 나눈 값)과 가중치를 곱했을 때 구해지는 값으로 표기되었으므로 가중치는 작업처리량(위의 표에서는 25)을 평균작업처리량으로 나눈 값으로 결정된다. 일단 비가상화 시스템 환경에서 가중치를 구하게 되면 이 값을 이용하여 가상화 시스템 환경의 통합작업처리량의 비교치를 구할 수 있다.

<표 2>는 <표 1>에서 구한 가중치를 이용하여 각 가상 머신 상에서의 작업처리량과 이를 합하여 구한 통합작업처리량의 값을 구한 것이다. <표 2>에서 확인할 수 있는 것처럼 비가상화 시스템에서 구한 가중치를 가지고 통합작업처리량을 계산한 결과 65.42로 비가상화 시스템이 100인 것에 비해 작업처리량이 줄어들었음을 확인할 수 있다. 개별

<표 1> 비가상화 시스템의 서버별 성능 시험 결과의 예

	평균작업처리량	가중치	작업처리량 (= 평균작업처리량 * 가중치)
서버1 (웹 서버)	1000.10	0.0250	25
서버2 (웹 서버)	1000.10	0.0250	25
서버3 (NFS서버)	0.6031	41.45	25
서버4 (DB서버)	0.0697	358.58	25
통합작업처리량	-	-	100

<표 2> 가상화 시스템의 가상머신별 성능 시험 결과의 예

	평균작업 처리량	가중치	작업처리량 (= 평균작업처리량 * 가중치)
가상머신1 (웹 서버)	700.00	0.0250	17.50
가상머신2 (웹 서버)	680.00	0.0250	17.00
가상머신3 (NFS서버)	0.4000	41.45	16.58
가상머신4 (DB서버)	0.0400	358.58	14.34
통합작업처리량	-	-	65.42

적인 작업처리량을 비교해 보아도 비가상화 시스템의 각 서버들보다 못 미치는 성능을 나타냄을 확인할 수 있다. 이와 같이 본 논문에서는 가상화 시스템과 비가상화 시스템에 같은 가중치를 적용함을 통해서 서로 다른 단위와 그 값의 범위가 다른 여러 종류의 성능 시험 프로그램에서 얻은 결과를 통합하고 두 시스템간의 상대적인 성능을 도출하는 방법을 제안하였다.

<표 2>는 <표 1>에서 구한 가중치를 이용하여 각 가상머신 상에서의 작업처리량과 이를 합하여 구한 통합작업처리량의 값을 구한 것이다. <표 2>에서 확인할 수 있는 것처럼 비가상화 시스템에서 구한 가중치를 가지고 통합작업처리량을 계산한 결과 65.42로 비가상화 시스템이 100인 것에 비해 작업처리량이 줄어들었음을 확인할 수 있다. 개별적인 작업처리량을 비교해 보아도 비가상화 시스템의 각 서버들보다 못 미치는 성능을 나타냄을 확인할 수 있다. 이와 같이 본 논문에서는 가상화 시스템과 비가상화 시스템에 같은 가중치를 적용함을 통해서 서로 다른 단위와 그 값의 범위가 다른 여러 종류의 성능 시험 프로그램에서 얻은 결과를 통합하고 두 시스템간의 상대적인 성능을 도출하는 방법을 제안하였다.

4.3 성능 시험을 위한 시스템 구성

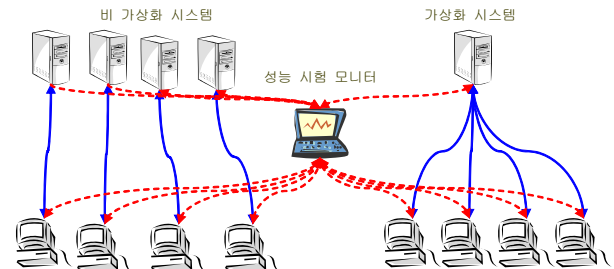
본 장에서는 성능 시험을 위한 실험 환경을 정의하도록 한다. 성능 평가를 위해서는 가상화 시스템과 비가상화 시스템의 실험 환경이 각각 구성되어야 한다. 먼저 본 논문에서는 중소기업 타입과 호스팅 타입을 기반으로 하여 가상화 시스템을 구성한다고 하였으므로 비가상화 시스템에는 서버의 역할을 할 다수의 시스템이 준비되어야 하며 또한 각각의 서버의 성능을 평가하는 클라이언트 머신이 준비되어야 한다. 비가상화 시스템에 대응하는 가상화 시스템은 한 대의 시스템 위에 여러 대의 가상 서버(Virtual Server)가 가동 될 것이므로 한 대의 시스템과 그 시스템에서 운영되는 가상 머신의 수만큼의 클라이언트가 필요하다.

(그림 10)에서 전체적인 실험 환경 구성의 예를 보여주고 있다. 가상화 시스템의 경우 중소기업 타입의 가상화 시스템으로 구성되었고 가상머신의 수는 4이다. 각각의 가상머신의 성능을 평가하는 4대의 클라이언트 머신이 구성되어 있다. 이 예제에서는 4대의 가상머신을 구성하고 두 대의 웹서버, 한 대의 데이터베이스 서버와 파일 서버로 실험 환경을 구성하였다. 각 가상머신에서 운영하고 있는 응용프로

그램 별로 그 응용프로그램에 서비스를 요청할 클라이언트를 구성하고, 클라이언트에서 서버로 서비스를 요청한 후 그 결과를 받을 수 있도록 성능 평가 프로그램들을 대응되는 각 응용프로그램 별로 선정해야 한다. 이 때 실제 가상화 시스템 한 대에서 4개의 서버를 운영하는 것이므로 네트워크 부분이 병목이 되어 성능이 저하된다면 시스템의 성능 평가에 방해가 되는 요인이므로 병목현상의 여부를 확인하고 혹 문제가 있을 경우 네트워크 카드(NIC)를 증설하거나 기가비트(Gigabit) 스위치나 네트워크 카드로 변경하는 등의 방법으로 네트워크가 병목현상을 일으키지 않도록 조치를 취해주어야 한다.

가상화 시스템에 대응하는 비가상화 시스템 환경을 구축할 때에도 가상화 시스템의 서버 구성과 동일하게 설치해 주도록 한다. 가상화 시스템의 실험 환경에서 4대의 가상머신을 구성했으므로 그와 비교 되는 비가상화 시스템도 4대의 서버로 구성되어야 한다. 각 서버에 가상화 시스템에서 설치했던 동일한 응용프로그램과 클라이언트 성능 평가 프로그램을 설치하도록 한다.

가상화 시스템의 경우 가상머신의 수 변경에 따른 확장성(Scalability)을 확인하기위해 그 수를 변경할 수 있다. 본 논문에서는 실험의 첫 단계에서 2대의 가상머신을 구성하여 서버의 종류 및 구성, 그리고 적용되는 워크로드를 변경해 가며 성능의 변화를 확인할 것이다. 다음으로 가상머신의 수를 4대로 구성하고 가상머신 수의 확장에 따른 성능의 변화를 확인하도록 한다.



(그림 10) 전체 실험 환경의 구성

5. 성능 평가

본 장에서는 앞서 정의한 가상머신의 모델(중소기업 타입)과 성능 평가 지표(통합작업처리량 대비 CPU낭비율)를 기반으로 실시한 실험의 성능 평가 결과를 나타내도록 한다. 실험이 진행되는 동안 가상화 시스템과 비가상화 시스템의 성능 결과를 실시간으로 바로 확인할 수 있다. 본 논문에서는 모든 성능 평가 항목에 있어서 5분간의 실험 결과를 도표와 그래프로 나타내도록 하였다.

성능 평가를 가능하는 요소로서 크게 두 가지 항목으로 나누어 살펴볼 수 있는데, 그 첫 번째는 가상화 시스템에 설치된 가상머신의 수의 확장에 따른 성능 변화이며, 두 번째는 실험 시스템의 서버의 구성과 워크로드의 조합에 따른

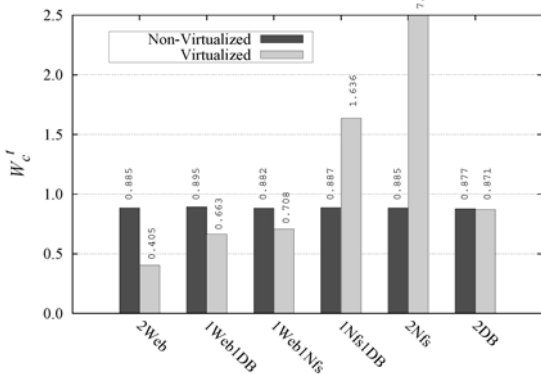
성능 변화이다. 기본적으로 가상머신의 수가 늘어날수록 시스템의 CPU 활용률은 높아지게 되므로 상대적으로 낭비되는 자원량은 줄어들 것이다. 그러나 가상머신의 수의 증가에 따른 가상머신 간 혹은 가상머신과 가상머신 모니터간의 내부 통신으로 인한 시스템 오버헤드가 발생하여 전체적인 작업처리량의 성능의 저하가 발생하며, 또한 복잡한 가상머신 수의 증가에 따른 복잡한 워크로드의 적용에 따른 시스템 부하는 성능을 저하시키는 요인이 된다.

본 논문에서는 우선 2대의 가상머신을 구동하는 가상화 환경에서 여러 종류의 벤치마크 조합을 이용하여 성능 평가를 실시하며, 다음으로 이를 확장하여 4대의 가상머신이 구동하는 환경에서 성능 평가를 진행하였다. 실험 결과를 통해서 가상머신의 확장에 따른 가상화 시스템의 성능 변화뿐만 아니라 다양한 종류의 서버군에 따른 다양한 형태의 워크로드를 적용함으로써 현재 가상화 시스템에 알맞은 가상화 시스템의 서버 구성을 알아볼 수 있으며 또한 가상화 시스템의 전체적인 성능 저하를 가져오는 요인들에 대해서 분석할 수 있다.

본 논문의 실험 결과를 통해 얻으려는 궁극적인 목적은 가상화 시스템의 성능 평가 도구로서의 적합성과 신뢰성을 확인하는 것이다. 반복적인 실험과 다양한 워크로드의 적용을 통해서 가상화 시스템의 성능 변화의 추이를 확인하고 이를 비가상화 시스템의 성능과 비교 분석함으로써 가상화 시스템이 지닌 장점과 보안해야 할 점을 발견할 수 있는지를 확인하는 것이 주요한 점이라고 하겠다. 특히 서버의 종류와 그에 따른 워크로드의 조합에 따른 성능 변화가 벤치마크 모니터에 분명히 드러나는지 확인하는 것도 중요한 부분이다.

5.1 2대의 가상머신 성능 시험

첫 번째 실험에서 2개의 가상머신을 가상화 시스템에 활성화 시켜놓고 각각의 가상머신에 다양한 서버 조합을 구성하여 성능 평가를 실시하였다. (그림 11)에서 X-축은 <식 2>에서 정의한 통합작업량 대비 CPU 낭비율(W_c^t)이고 Y-축은 2개의 가상머신에 각각 설치된 서버의 종류를 의미한다. 첫 번째 막대그래프는 비가상화 시스템과 가상화 시스

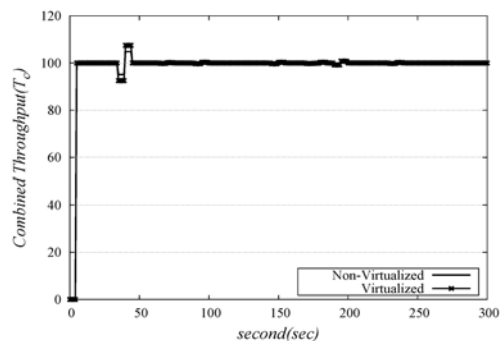
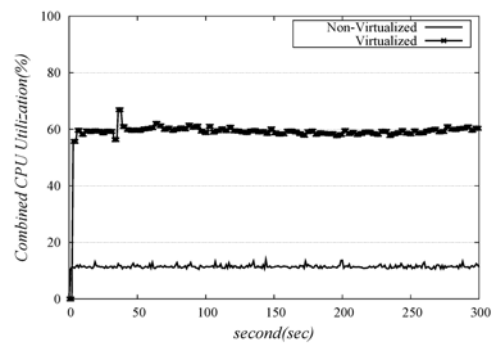


(그림 11) 가상화 시스템(2대의 가상머신 설치)과 비가상화 시스템의 성능 비교

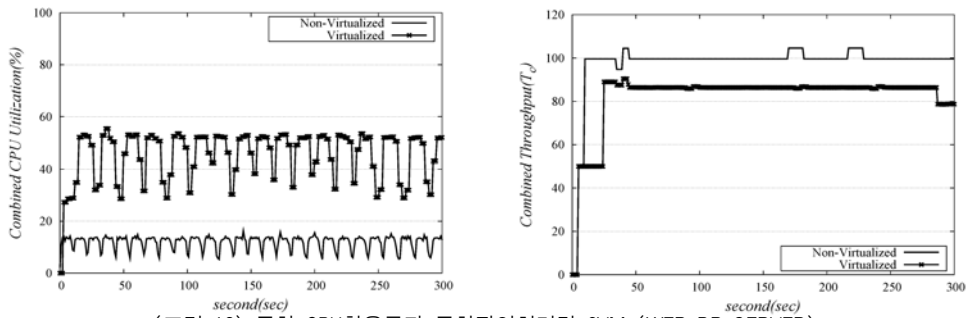
템에 각각 2대의 웹서버를 설치하고 클라이언트 벤치마크로 *httperf*를 사용했을 때 계산한 성능 비이다.

알장에서도 언급했듯이 각각의 워크로드 제너레이터는 대상 서버에 적합한 워크로드를 생성해 주는데, *httperf*의 경우 네트워크와 CPU에 부하가 많이 걸리는 워크로드를, *IOzone*은 네트워크와 디스크 입출력에 많은 부하가 걸리는 워크로드를, 그리고 *MySql-bench*의 경우 네트워크와 CPU, 디스크 입출력에 대한 워크로드가 적절히 섞여있는 워크로드를 생성한다.

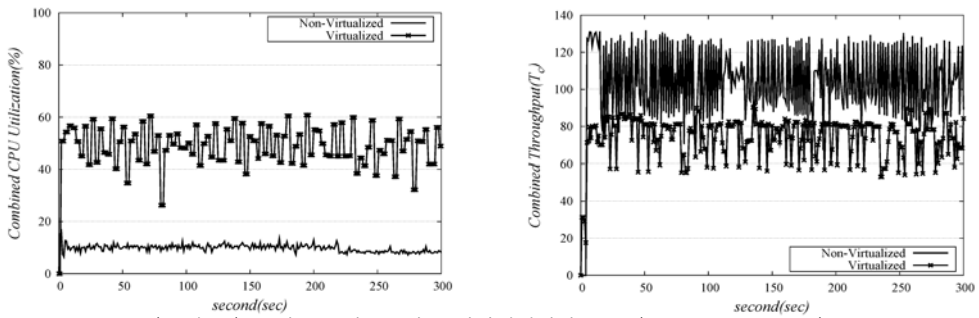
(그림 11)에서도 나타나듯이 비가상화 시스템의 경우의 통합작업량 대비 낭비율은 거의 모든 워크로드를 대상으로 고정되어 있는 것을 볼 수 있다. 이는 4장의 성능 평가 지표의 정의에서 비가상화 시스템의 통합작업처리량을 100으로 기준하여 결과 값을 환산했기 때문이며 본 실험 결과 모든 경우의 성능 평가에 있어서 CPU 활용률은 10%~15%로 큰 편차를 보이지 않았기 때문이다. 즉 비가상화 시스템의 성능 평가 결과는 가상화 시스템의 성능을 비교 평가하기 위한 잣대의 역할을 한다고 볼 수 있다. 반면에 가상화 시스템의 성능 결과는 다양한 분포를 보이고 있는데, 특히 CPU 부하가 많은 워크로드의 적용했을 때 상대적으로 적은 낭비율을 보였고, 디스크 입출력 부하가 많은 워크로드를 적용했을 때에는 CPU 활용률은 줄어들고 통합작업처리량은 떨어지는 결과를 보여주었다. 아래의 도표들은 (그림 11)의 각 성능 평가 결과에 대한 통합 CPU 활용률(C_w)과 통합작업처리량(T_c)의 결과 중 하나를 선정하여 도표화한 것이다. 모든 실험은 5분간 이루어졌으며 반복적인 실험을 통해서 결과의 신뢰성을 확인하였다.



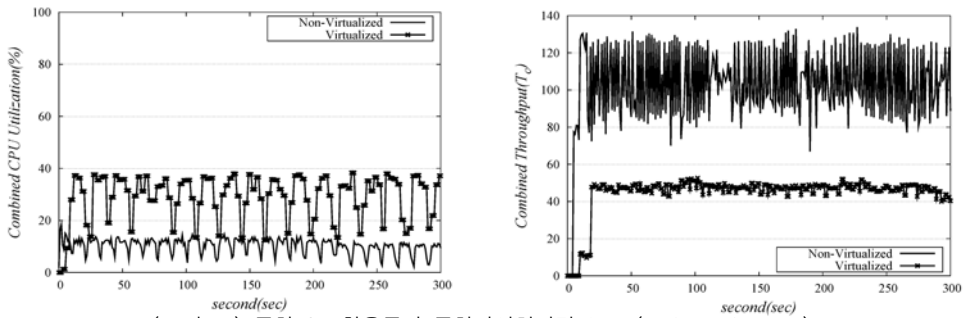
(그림 12) 통합 CPU활용률과 통합작업처리량 2VM (WEB-WEB SERVER)



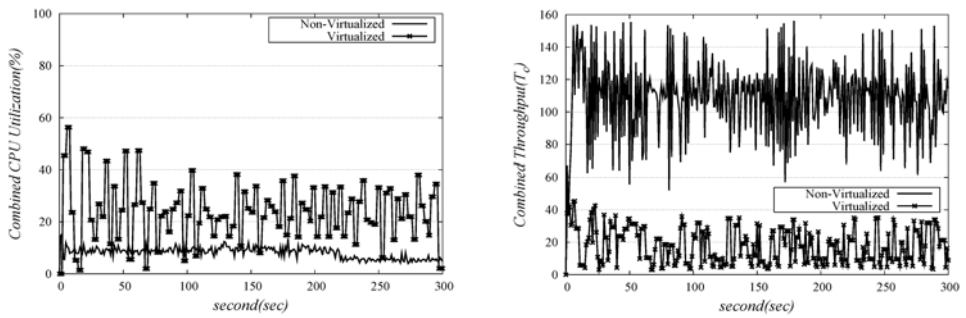
(그림 13) 통합 CPU활용률과 통합작업처리량 2VM (WEB-DB SERVER)



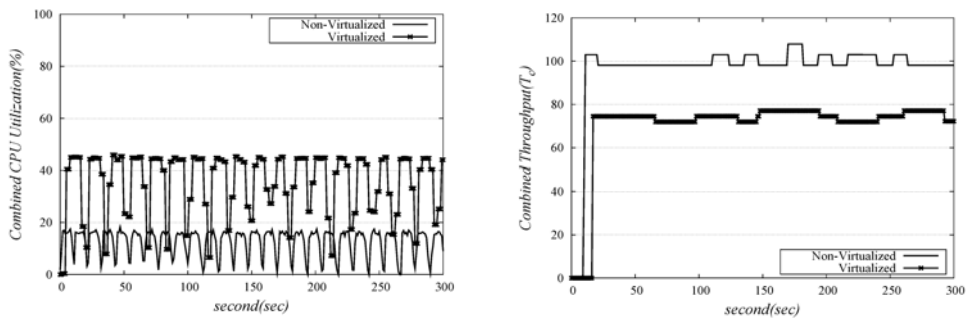
(그림 14) 통합 CPU활용률과 통합작업처리량 2VM (WEB-NFS SERVER)



(그림 15) 통합 CPU활용률과 통합작업처리량 2VM (NFS-DB SERVER)



(그림 16) 통합 CPU활용률과 통합작업처리량 2VM (NFS-NFS SERVER)



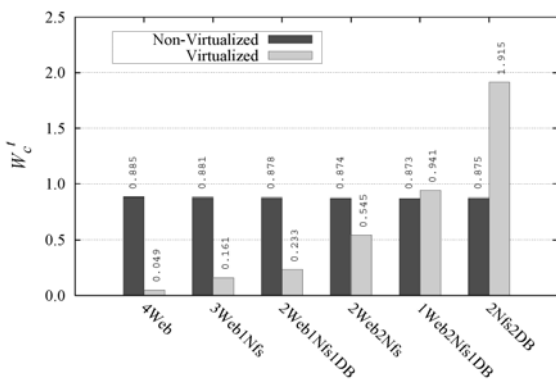
(그림 17) 통합 CPU활용률과 통합작업처리량 2VM (DB-DB SERVER)

모든 항목의 그림에서 나타나듯이 CPU 활용률의 평균은 비가상화 시스템에 비해서 높게 나오는 것을 확인할 수 있다. 반면 통합작업처리량 지수는 비가상화 시스템에 비해서 다소 떨어지는 양상을 보였다. 예를 들어 CPU의 부하가 많은 WEB-WEB (그림 12)의 경우나 WEB-DB (그림 13) 서버 구성 실험에서는 비가상화 시스템에 비해서도 거의 비슷하거나 다소 떨어지는 결과를 보여주었지만, 디스크 입출력 부하가 많은 NFS-DB (그림 15), NFS-NFS (그림 16) 서버 구성의 경우에는 그림에도 나타나듯이 현저한 성능 저하를 보여주었다. 뿐만 아니라 CPU 활용률 면에서도 다른 서버 구성들 보다 디스크 입출력의 워크로드가 적용되었을 때 큰 이득을 보이지 못하는 양상을 보였다. 이는 디스크 장치에 많은 작업량이 시스템 전체에 병목현상을 만들어 내는 결과로 분석할 수 있다. 기존의 가상화 기술 관련 연구에서도 알려져 있듯이 디스크 입출력 부하가 많은 작업에 관한 가상화 시스템의 성능 저하는 해결해야 하는 문제로 남아있다 [14].

5.2 4대의 가상머신 성능 시험

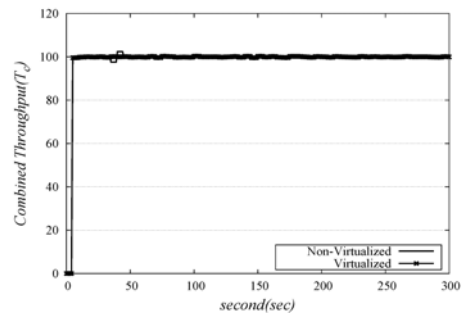
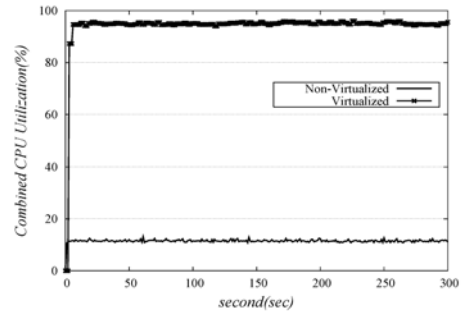
첫 번째 실험의 확장으로 4대의 가상머신을 가상화 시스템에 설치하고 운영했을 경우의 성능 평가를 실시하였다. 가상머신의 수가 늘어날수록 시스템 오버헤드는 증가하게 되므로 전체적인 CPU 활용률은 높아지는 양상을 보였다.

2대의 가상머신으로 성능 평가를 실시했을 때, 디스크 입출력에 대한 병목현상으로 통합작업처리량 수치에서 상대적으로 비가상화 시스템보다 낮은 처리율을 보였으므로 4대의 가상머신 환경에서는 NFS서버와 데이터베이스 서버의 양을 점차 늘여가면서 성능 변화의 추이를 살피도록 하였다. (그림 18)은 4대의 가상머신을 설치하고 이를 비가상화 시스템의 성능과 비교 평가한 결과이다. 2대의 가상머신을 설치하고 성능 평가를 실시했을 경우보다 통합작업처리량 대비 CPU낭비율 (W_c^A)의 수치가 훨씬 낮아졌음을 확인할 수 있다. 이것은 비단 가상머신의 증가에 따른 시스템 오버헤드의 차원뿐 아니라, 4대의 가상머신에서 워크로드를 동시에 처리하면서 유용하게 시스템 자원이 활용되었음을 보여주는 결과라고 할 수 있다. 특히 4대의 웹서버를 운영하고 이를 평가했을 때 통합작업처리량(T_c)의 수치가 비가상화 시스템

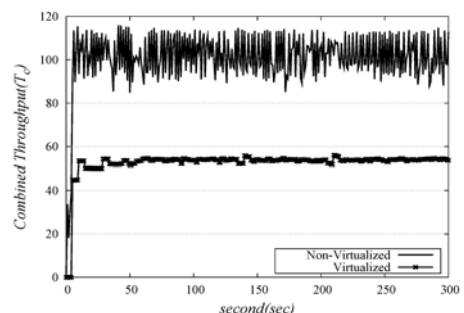
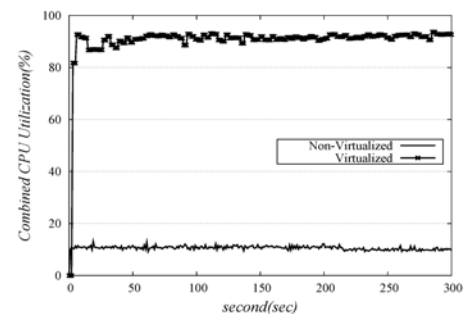


(그림 18) 가상화 시스템(4대의 가상머신 설치)과 비가상화 시스템의 성능 비교

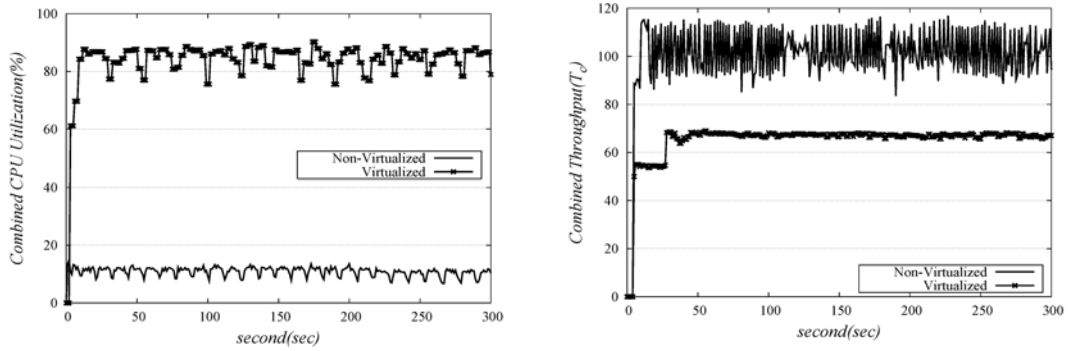
의 결과와 대등한 수준을 나타냄으로 이를 확인할 수 있다 (그림 19). (그림 18)의 수치상으로는 대부분의 결과에서 비가상화 시스템에 비해서 더 좋은 성능 평가 지수를 나타내고 있는데 이는 상대적으로 CPU 활용률의 증가의 폭이 컸기 때문이다. (그림 20)에서 부터 (그림 24)에 이르는 결과에서도 확인하듯이 NFS서버와 데이터베이스 서버의 수가 증가 할수록 통합 작업처리량 지수는 상대적으로 떨어지는 결과를 보였다.



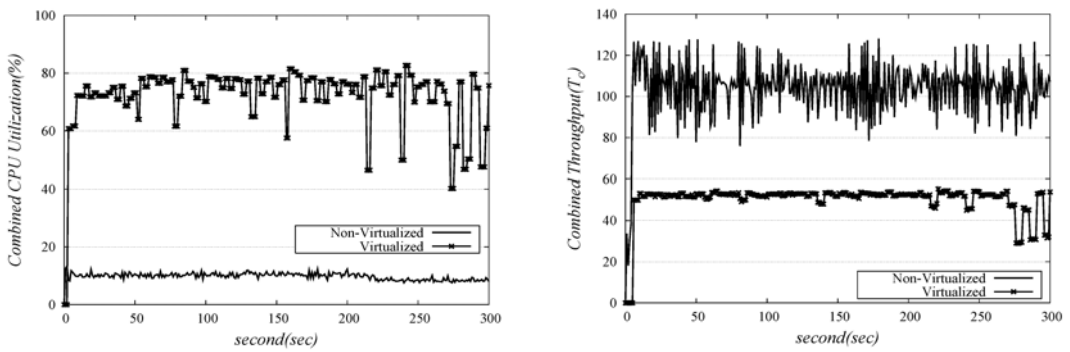
(그림 19) 통합 CPU활용률과 통합작업처리량 4VM (WEB-WEB-WEB-WEB SERVER)



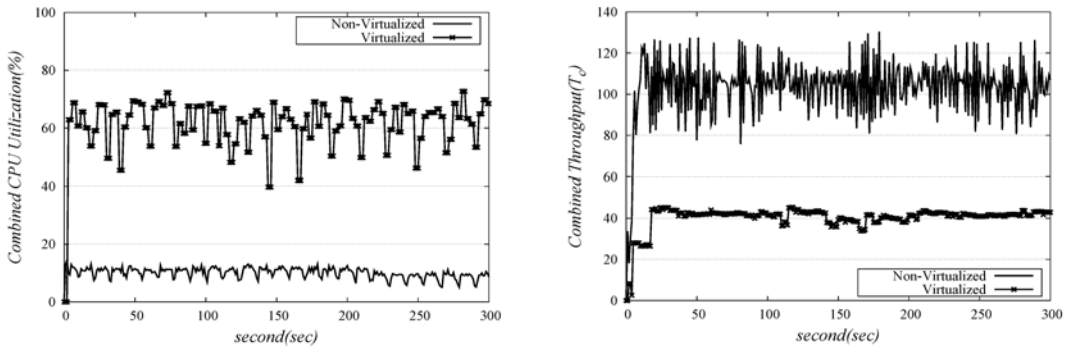
(그림 20) 통합 CPU활용률과 통합작업처리량 4VM (WEB-WEB-WEB-NFS SERVER)



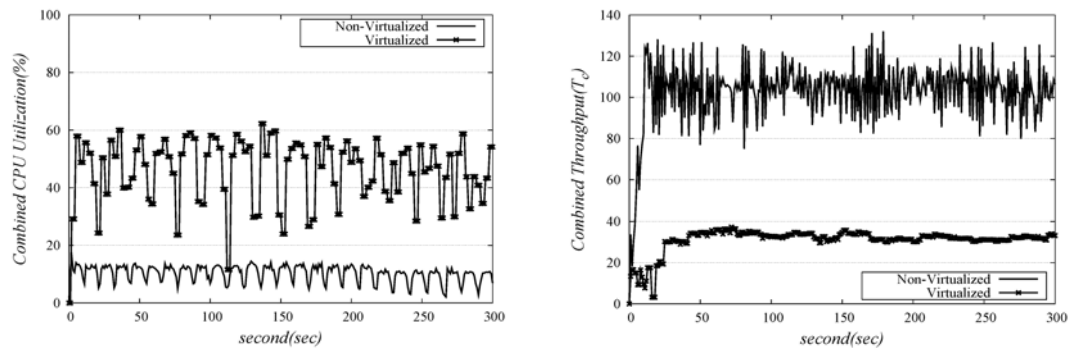
(그림 21) 통합 CPU활용률과 통합작업처리량 4VM (WEB-WEB-NFS-DB SERVER)



(그림 22) 통합 CPU활용률과 통합작업처리량 4VM (WEB-WEB-NFS-NFS SERVER)



(그림 23) 통합 CPU활용률과 통합작업처리량 4VM (WEB-NFS-NFS-DB SERVER)



(그림 24) 통합 CPU활용률과 통합작업처리량 4VM (NFS-NFS-DB-DB SERVER)

이 번 실험 결과에서 주목할 부분은 4대의 웹서버를 운영했을 때 (그림 19)와 3대의 웹서버 그리고 1대의 NFS서버를 운영했을 때 (그림 20)의 통합작업처리량의 감소폭이 매우 크다는 점이다. 4대의 웹서버를 운영했을 때의 통합작업처리량이 비가상화 시스템과 거의 대등한 반면, 1대의 NFS서버가 포함되었을 경우 전체적인 시스템의 작업처리율에 미치는 영향이 크을 알 수 있다.

6. 결론 및 향후 과제

가상화 기술이 다시 대두됨에 따라 가상화 기술의 효용성을 나타내기 위한 성능 평가 방법의 필요성이 대두되었고, 그에 대한 여러 가지 연구가 진행되어 왔다. 그러나 기존 연구들은 대부분 가상화에 의한 시스템의 활용률을 높이는 측면에서 가상화 시스템의 성능을 측정하거나, 가상화의 오버헤드를 고려한다 하더라도 서로 다른 가상화 기술이나 구성을 비교하기 위한 성능 측정에 대한 연구들이었다.

본 논문에서는 가상화 시스템이 시스템의 활용성을 증가시키는 측면 뿐 아니라 가상화에 의해 응용 프로그램의 성능이 저하되는 측면을 모두 고려하여, 비 가상화 시스템과 성능을 비교할 수 있는 성능 평가 방법을 제안하였다.

가상화 시스템의 성능 평가를 위하여 우선 가상화 시스템의 구성을 4가지 형태로 분류하여 정의하고, 이 중 일반적으로 사용되는 중소기업 타입의 가상화 시스템을 기반으로 성능 평가 방법을 제안하였다.

가상화 시스템이 기존의 비가상화 시스템과 비교해 지니고 있는 장점과 보완점이 드러나도록 시스템 자원의 활용률과 기존의 벤치마크 어플리케이션을 통한 작업처리량을 이용해 통합작업처리량 대비 CPU 낭비율(W_c^*)를 성능 비교 지표(metric)로 정의하였고, 다양한 워크로드의 시험과 가상머신 수의 변경에 따라 이 성능 비교 지표의 변화를 관찰하고, 분석하였다.

본 논문에서 제안한 가상화 시스템의 성능 실험 방법은, 시스템 활용률과 작업처리량 정보를 적절히 조합하여 비가상화 시스템과의 성능을 비교할 수 있는 틀을 마련하고, 기존의 가상화 시스템 성능 평가 방법이 지닌 한계를 보완하였다. 더불어 가상화 시스템의 성능 평가를 위해 새로운 구조의 벤치마크를 개발하기보다 기존에 시스템 평가 도구로 사용되었던 틀들을 이용하므로, 쉽게 다양한 가상화 시스템을 위한 성능평가 방법으로 적용될 수 있다.

본 논문에서는 가상화 시스템의 시스템 활용률을 측정하기 위하여 CPU 활용률을 사용하였다. 그러나 메모리, 네트워크 대역폭, 디스크 입출력 대역폭 등 다른 시스템 자원을 고려하여 시스템의 활용률을 정의하도록 확장하는 연구를 진행하고 있다.

참 고 문 헌

[1] M. Rosenblum, T. Garfinkel, "Virtual machine monitors:

- current technology and future trends," Computer(IEEE Computer Society), Vol 38, Issue 5, pp.39-47, May, 2005.
- [2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization," Proceedings of the nineteenth ACM symposium on Operating systems principles, Jun., 2003.
- [3] L. Cherkasova and R. Gardner, "Measuring CPU Overhead for I/O Processing in the Xen Virtual Machine Monitor," Technical Report HPL-2005-62, Internet Systems and Storage Laboratory, HP Laboratories Palo Alto, Mar., 2005.
- [4] B. Clark, T. Deshane, E. Dow, S. Evanchik, M. Finlayson, J. Herne, and J. N. Matthews, "Xen and the Art of Repeated Research," Proceedings of the FREENIX Track:2004 USENIX Annual Technical Conference, Jun., 2004
- [5] VMware, "VMware ESX Server 2 ESX Server Performance and Resource Management for CPU-Intensive Workloads," Technical Report ESX-ENG-Q405-133, VMware, Inc.
- [6] D. A. Menasce, "Virtualization: Concepts, Applications, and Performance Modeling," 31th International Computer Measurement Group Conference, Dec., 2005.
- [7] J. P. Casazza, M. Greenfield, and K. Shi, "Redefining Server Performance Characterization for Virtualization Benchmarking," Technical Report 1535-864X, Intel Corp.
- [8] httpperf. <http://www.hpl.hp.com/research/linux/httpperf/>
- [9] IOzone. <http://www.iozone.org/>
- [10] MySQL-Benchmark. <http://dev.mysql.com/doc/refman/5.0/en/mysql-benchmarks>.
- [11] J. E. Smith and R. Nair, "Virtual Machines: Versatile Platforms for Systems and Processes," Elsevier Inc., San Francisco, CA, 2005.
- [12] A. Menon, J. R. Santos, Y. Turner, G. Janakiraman, and W. Zwaenepoel, "Diagnosing Performance Overheads in the Xen Virtual Machine Environment," First ACM/USENIX Conference on Virtual Execution Environments (VEE'05), Jun., 2005.
- [13] A. Menon, A. L. Cox, and W. Zwaenepoel, "Optimizing Network Virtualization in Xen," 2006 USENIX Annual Technical Conference, Jun., 2006.
- [14] L. Cherkasova, and R. Gardner, "Measuring CPU Overhead for I/O Processing in the Xen Virtual Machine Monitor," Technical Report HPL-2005-62, HP Laboratories, Apr., 2005.



장 지 용

e-mail : jiyong97@sogang.ac.kr
2005년 서강대학교 컴퓨터공학과(학사)
2007년 서강대학교 동대학원
컴퓨터공학과(공학석사)
2007년~현 재 삼성전자 정보통신연구소
연구원

관심분야: 가상머신(Virtual Machine), 이동통신망, LTE,
망광리(OAM-Operation, Administration and
Management), SNMP



김 진 석

e-mail : sabin@sogang.ac.kr
2006년 서강대학교 컴퓨터공학과(학사)
2008년 서강대학교 대학원 컴퓨터공학과
(공학석사)
2008년~현 재 (주)NHN 플랫폼서비스
개발팀 개발자

관심분야: 프레임워크, 플랫폼서비스, 가상화, 성능평가 등



한 세 영

e-mail : syhan@sogang.ac.kr
1991년 포항공과대학교 수학과(이학사)
2003년 서강대학교 정보통신대학원
(공학석사)
2004년~현 재 서강대학교 컴퓨터학과
박사과정

1996년~2006년 (주)이엔지 기술본부
관심분야: 피어투피어 컴퓨팅, 분산처리 시스템



박 성 용

e-mail : parksy@sogang.ac.kr
1987년 서강대학교 컴퓨터학과(공학사)
1994년 미국 Syracuse University 대학원
(공학석사)
1998년 미국 Syracuse University(공학박사)
1998년~1999년 미국 Bell Communication
Research 연구원

1999년~현 재 서강대학교 컴퓨터학과 부교수
관심분야: Autonomic Computing, Peer to Peer Computing,
High Performance Cluster Computing and System