

PCIA 클라우드 서비스 모델링 및 자원 구성에 따른 성능 영향도 분석

윤빈풍*, 곽종욱**

PCIA Cloud Service Modeling and Performance Analysis of Physical & Logical Resource Provisioning

Binfeng Yin*, Jong Wook Kwak**

요약

클라우드 컴퓨팅은 융통성 있는 연산능력과 대량의 데이터를 분석하는 효과적인 환경을 제공한다. 본 논문에서는 공용 클라우드를 위한 새로운 타입의 자원 배치 구조를 정의하고 이를 PCIA라고 명명한다. 아울러, PCIA 구조에서 서비스 모델에 대해 소개하며, 이를 위한 클라우드 명명 기법도 제시한다. 제안된 클라우드 시스템은 어플리케이션의 다양한 특징에 따라 사용자의 가상 자원 요구에 기반하는 적절한 물리적 자원을 선택하게 된다. 이를 통해 자원 제공의 변화에 따른 성능 변화의 연관성 및 클라우드 시스템의 효과적 구축을 위한 시스템 구성 요소의 영향도를 분석하며, 클라우드 서비스 제공자와 사용자 양쪽 모두에게 중요한 리소스 프로비저닝의 주요 기준을 제시한다.

▶ Keywords : 클라우드 시스템, 가상화, PCIA, 자원 제공, 성능 분석

Abstract

Cloud computing provides flexible and efficient mass data analysis platform. In this paper, we define a new resource provisioning architecture in the public cloud, named PCIA. In addition, we provide a service model of PCIA and its new naming scheme. Our model selects the proper number of physical or virtual resources based on the requirements of clients. By the analysis of performance variation in the PCIA, we evaluate the relationship between performance variation and resource provisioning, and we present key standards for cloud system constructions, which can be an important resource provisioning criteria for both cloud service providers and clients.

•제1저자 : 윤빈풍 •교신저자 : 곽종욱

•투고일 : 2014. 1. 9, 심사일 : 2014. 1. 28, 게재확정일 : 2014. 2. 3.

* 영남대학교 기계공학과(Dept. of Mechanical Engineering, Yeungnam University)

** 영남대학교 컴퓨터공학과(Dept. of Computer Engineering, Yeungnam University)

▶ Keywords : cloud system, virtualization, PCIA, resources provisioning, performance analysis

I. 서론

일반적으로 서버는 높은 성능의 환경에서 운영된다. 하지만 전통적인 데이터 센터에서의 평균 서버 활용률은 20% ~ 25% 내외로 매우 낮은 실정이다[1][2]. 다수의 가상 서버가 하나의 물리적 서버 하에서 실행 될 때, 다양한 응용 프로그램들은 각각의 가상 서버에서 수행 될 수 있고, 이 때 응용 프로그램에 따른 서버 자원의 효율적 분배는 시스템 전체적인 측면에서 중요한 성능 기준이 된다. 특히, 이는 시스템의 운영비용을 감소시키며, 아울러 저장 장치의 공간 사용을 줄이고 물리적 서버의 에너지 사용 또한 감소시킬 수 있다. 이러한 측면에서 효율적인 서버 가상화는 경제적이고 친환경적이라 할 수 있다[3].

전통적인 클라우드 시스템은 인프라스트럭처 레이어(IaaS), 플랫폼 레이어(PaaS), 어플리케이션 레이어(SaaS)의 3개 레이어 계층으로 분류할 수 있다. 그림 1은 클라우드 컴퓨팅 환경에서 각 계층 간의 연관성을 보여 주고 있다. 이러한 3개의 레이어는 일반적으로 퍼블릭, 프라이빗, 하이브리드 클라우드 시스템에서 각각 구현 된다[4][5]. 이 가운데 인프라스트럭처 레이어(IaaS)는 클라우드 사용자에게 필수적인 인프라스트럭처와 융통성 있는 컴퓨터 자원을 제공한다. 이를 통해 사용자는 하드웨어 장치를 구입하거나 보수할 필요 없이 IaaS 위에서 자신만의 시스템 환경을 직접 구축할 수 있게 된다.

IaaS는 클라우드 컴퓨팅의 전통적인 서비스 계층으로서

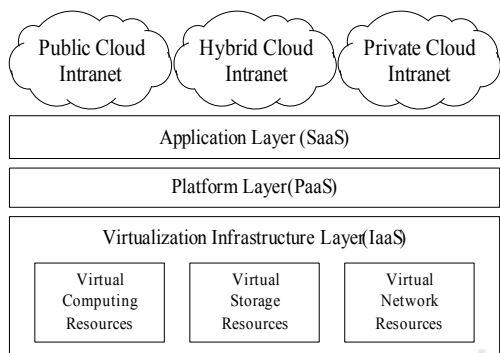


그림 1. 클라우드 컴퓨팅 모델의 계층적 구조
Fig. 1. Hierarchical Relationship of Cloud Computing

효율적이고 편리한 인프라스트럭처 플랫폼을 제공한다[6]. 하지만 문제는 무한한 인프라스트럭처 자원이 클라우드 환경에 있는 것처럼 보이지만 그렇지 않다는 데 있으며, 이를 극복하기 위해 부족한 자원의 사용률을 최대화 하는 것과 유한한 자원을 사용자에게 무한한 자원처럼 보이도록 하는 것이 중요하다. 자원의 요구는 유동적인 자원 프로비저닝(provisioning)을 통해 다양한 응용 프로그램의 조건에 맞는 자동화된 분배에 따라 적절한 물리적 자원으로 맵핑될 수 있다[7].

본 논문에서는 공용 클라우드(public cloud)에서 새로운 타입의 자원 프로비저닝 구조를 제안하고, 이를 PCIA(Public Cloud Infrastructure Architecture)라 명명한다. PCIA는 고정된 수의 가상머신 환경인 vCloud와 고정된 수의 물리적 호스트 환경인 pCloud로 구성된다. PCIA에서는, 물리적인 자원과 가상화된 자원의 분리된 관점으로부터, 가상머신이 어플리케이션의 다양한 요구와 특성 따라 능률적이고 융통성 있게 서로 다른 물리적 호스트에 분배될 수 있다. 이렇게 함으로써, 한정된 자원 상황 하에서 자원 활용 효율을 높이는 것이 가능하게 되고, 이를 통해 더 나은 성능의 클라우드 컴퓨팅 환경을 제공한다.

또한 본 논문에서는 클라우드 시스템의 효과적인 구축을 위하여, 클라우드 시스템의 논리적 및 물리적 구성 요소별 영향도의 차이를 분석하며, 사용자가 요청하는 다양한 응용 프로그램의 특성에 따른 클라우드 시스템의 성능 변화를 관찰한다. 이를 통해 클라우드 서비스 제공자나 서비스 사용자 모두에게 중요하게 취급될 수 있는 자원 제공 및 요청의 기준을 제시하며, 이를 바탕으로 물리적 자원의 제한 하에서 최적의 성능을 도출하기 위한 적절한 수의 가상 자원을 선택하는 것과 사용자의 가상머신 요청에 따른 물리적 자원의 적절한 할당에 있어서, 클라우드 서비스 제공자와 사용자 모두에게 클라우드 시스템 구축의 중요한 기준으로 활용될 수 있는 시스템 가상화의 주요 선택 기준을 제시한다.

II. 관련 연구

물리적 호스트와 가상머신은 동적으로 선택되어 활용될 수 있다. 그리고 높은 성능의 프로세서와 메모리 그리고 저장장치와 같은 시스템 구성 요소들은 응용 프로그램의 요구에 따

라 효율적이고 융통성 있게 몇 개의 서로 다른 물리적 호스트에 분배될 수 있다. 이와 같은 클라우드 서비스는 잘 정의된 자원 프로비저닝 환경에서 제공될 수 있다[8][9]. 이러한 자원 프로비저닝 기술에 관한 관련 연구로, Zhang 등은 자원 소비와 자원 배치 두 가지 측면에서의 조정 접근법을 제안하였다[10]. 이들은 별도의 미들웨어 함수를 구축하여, 이를 클라우드 자원 활용에 사용하였다. Aoun 등은 클라우드 시스템에서 제공될 수 있는 몇 가지 형태의 기본 서비스에 대해 고려하였으며, 그들이 어떻게 서비스의 프로비저닝을 이끌어 낼 수 있는지 보여 주었다[11]. Henzinger 등은 FlexPRICE라 불리는 시스템 환경을 제안하였다[12]. 주어진 환경에서 클라우드 서비스 사용자들은 자원 활용의 효율성을 최적화 하는지에 따라 그들이 요구하는 속도와 가격의 가상머신을 구매할 수 있게 하였다. Rehman 등은 프로비저닝의 변화 접근법에 따른 영향력을 연구하였고, 같은 가상 자원이 다른 물리적 계층의 클라우드 시스템에 맵핑되었을 때, 응용 프로그램의 성능에 어떻게 변화를 가지는가에 대한 결과를 보였다[7].

이상의 기법들은 자원 분배와 관리 관점에서 과도한 자원의 소비를 유도하거나 가상자원과 물리자원의 비균등적 맵핑을 유발할 수 있었다. 본 논문에서 제안된 새로운 클라우드 시스템 정의 기법은 서로 다른 가상 호스트의 수와 물리적 호스트의 구조 하에서 클라우드 사용자 혹은 서비스 제공자가 물리적 및 논리적 호스트의 세밀한 선택을 가능하게 한다. 또한 본 논문에서는 PCIA 모델링 환경 하에서 물리적 논리적 클라우드 시스템의 구성 변화에 따른 응용 프로그램의 성능 변화 추이를 세밀하게 분석하여 서비스 제공자와 사용자 모두에게 유용한 서비스 모델의 선택에 도움을 주고자 하였다. 이는 특히 물리적 호스트 수 혹은 가상 머신의 수에 제약이 있는 클라우드 선택 환경에서 중요한 의사 결정 기준으로 활용될 수 있다.

III. Public Cloud Infrastructure Architecture (PCIA)

물리적 컴퓨팅 자원의 추상화나 동적인 자원 배치의 기본적인 원칙은 인프라스트럭처 레이어의 기능에 근거하여 하드웨어 내부의 자동화 관리를 통하여 효율적으로 적용될 수 있다. 일반적으로 클라우드 프로비저닝은 사용자의 관점과 다양한 자원 활용 상태에 따라 가상 자원을 서로 다른 물리적 자원에 맵핑함으로써 구현된다.

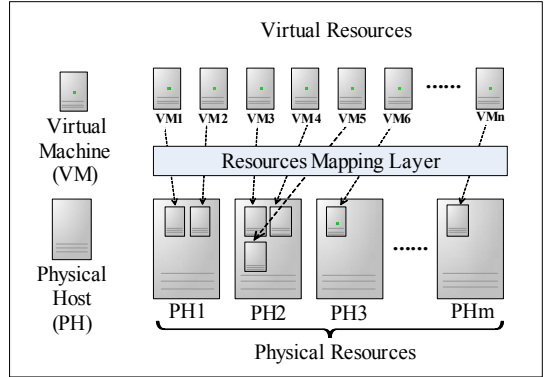


그림 2. 퍼블릭 클라우드 인프라스트럭처 아키텍처
Fig. 2. Public Cloud Infrastructure Architecture (PCIA)

본 논문에서는 그림 2에서 보여지는 것과 같은 퍼블릭 클라우드 인프라스트럭처 아키텍처 (PCIA) 모델을 소개한다. 다음은 이 모델에 대한 구조 형식적 정의이다. 우선, 주어진 모델은 고정된 수의 가상 머신 환경과 고정된 수의 물리 호스트 환경으로 구성된다.

$$PCIA = \langle vCloud, pCloud \rangle \tag{3.1}$$

PCIA는 식 3.1과 같이 정의되며, 이때 vCloud는 고정된 수의 가상머신으로 구성된 클라우드 플랫폼이며, pCloud는 고정된 수의 물리머신으로 구성된 클라우드 플랫폼이다. 주어진 PCIA는 각각의 vCloud와 pCloud에 대해

$$\begin{aligned} &(V \subset vCloud) \wedge (V \subset pCloud) \\ &(P \subset vCloud) \wedge (P \subset pCloud) \end{aligned} \tag{3.2}$$

이며, 이때 V와 P는 식 3.3과 같이, 각각 v로 이루어진 가상 머신과 p로 이루어진 물리적 호스트를 의미한다.

$$\begin{aligned} V &= \{v | v \text{ is a virtual machine} \} \\ P &= \{p | p \text{ is a physical host} \} \end{aligned} \tag{3.3}$$

이와 같은 PCIA 구조 하에서, 다음과 같은 서비스 모델 (service model)을 소개한다. 서비스 모델과 이에 대한 프로비저닝 관점으로부터의 PCIA는 새로운 서비스모델 S로 정의된다. S는 $S = \langle \alpha, \beta, i, j \rangle$ 의 4개 요소로 구성되며, 식 3.4와 같이 표기 된다.

$$S = {}_{\alpha}S_{\beta}(i, j) \quad (3.4)$$

이때, α 는 식 3.3에서 정의된 v 혹은 p 가운데 하나로 할당되며, β 는 서비스의 시나리오 번호이다. i는 물리적 호스트의 개수를 의미하며, j는 가상 머신의 개수이다. 이상의 식 3.2에서 식 3.4를 통해 다음과 같은 Pi와 Vj를 새롭게 정의한다.

$$P_i = \{p_i | p_i \text{ is physical host } i\}$$

$$V_j = \{v_j | v_j \text{ is virtual machine } j\} \quad (3.5)$$

PCIA 서비스 모델에서 다양한 수의 물리 호스트는 다양한 수의 가상 머신에 할당될 수 있다. 또는 그 반대로 다양한 수의 가상 머신이 다양한 수를 가진 물리 호스트로 분배될 수 있다. 이러한 기능을 제공함으로써 다음의 식 3.6의 표현이 물리 호스트와 가상 머신의 어떠한 조합에도 적합한 일반적인 서비스모델을 나타낼 수 있게 된다.

$${}_{\alpha}S_{\beta}(i, j) = \{vS_{\beta}(i, j), pS_{\beta}(i, j)\} \quad (3.6)$$

$(\alpha = v \text{ or } p, 1 \leq \beta \leq k, 1 \leq i \leq l, i \leq j \leq i \times m)$

이 때, k는 최대 시나리오 번호이며, l은 최대 물리 호스트 번호이다. 그리고 m은 물리 호스트가 실행할 수 있는 최대 가상 머신의 번호이다. 또한 물리 호스트와 가상 머신 구성 요소들의 표기법 관점에서 식 3.6은 식 3.5를 활용하여 식 3.7과 같이 표기 될 수 있다. 식 3.7의 표현 모델을 이용하여 각 물리 호스트에 존재하는 가상 머신의 구성을 효과적으로 파악할 수 있게 한다.

$${}_{\alpha}S_{\beta}(i, j) = \{P_1(V_1 \dots V_{\alpha}), P_2(V_{\alpha+1} \dots V_b), \dots, P_i(V_{e+1} \dots V_j)\} \quad (3.7)$$

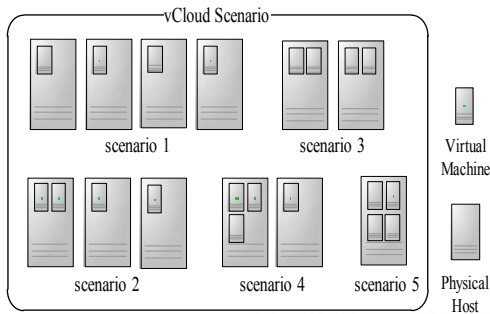


그림 3. vCloud 서비스 모델
Fig. 3. vCloud Service Model in PCIA

이상의 정의를 바탕으로 PCIA 모델을 서비스 모델 관점에서 설명하면 다음과 같다. 그림 3은 PCIA의 vCloud 서비스 모델을 나타낸다. 주어진 그림 3은 식 3.8과 같이 나타낼 수 있다. 그리고 그림 4에서 보여지는 pCloud 서비스 모델은 식 3.10과 같이 표현된다. 다음은 이와 관련한 세부사항이다.

$${}_{\alpha}S_{\beta}(i, j) = \{vS_{\beta}(i, j)\} \quad (3.8)$$

$(\alpha = v, 1 \leq \beta \leq 5, 1 \leq i \leq 4, j = 4)$

PCIA의 서비스 모델 vCloud를 소개한 그림 3에서 물리 호스트의 개수는 유동적이다. 하지만 가상 머신의 수는 4로 고정되어 있다. 따라서 식 3.8에서 보여지는 것과 같이 최종적으로 5가지의 시나리오가 존재한다. 식 3.8을 이용하여 vCloud 서비스 모델은 식 3.9와 같이 정의 될 수 있다.

$$vCloud = \{vS_1(4,4), vS_2(3,4), vS_3(2,4), vS_4(2,4), vS_5(1,4)\}$$

$$vS_1(4,4) = \{P_1(V_1), P_2(V_2), P_3(V_3), P_4(V_4)\}$$

$$vS_2(3,4) = \{P_1(V_1), P_2(V_2), P_3(V_3, V_4)\}$$

$$vS_3(2,4) = \{P_1(V_1, V_2), P_2(V_3, V_4)\}$$

$$vS_4(2,4) = \{P_1(V_1, V_2, V_3), P_2(V_4)\}$$

$$vS_5(1,4) = \{P_1(V_1, V_2, V_3, V_4)\} \quad (3.9)$$

한편, 그림 4는 PCIA에서의 pCloud 서비스 모델의 예를 보여 준다. 물리 호스트의 수는 i로 고정되어 있고 동일하지 않은 j개의 가상머신 분산에 따른 k개의 시나리오가 존재할 때, pCloud는 다음 식 3.10과 같이 정의된다.

$${}_{\alpha}S_{\beta}(i, j) = \{pS_{\beta}(i, j)\} \quad (3.10)$$

$(\alpha = p, 1 \leq \beta \leq 3, i = 4, j = (4, 8, 16))$

PCIA 서비스 모델 pCloud의 그림 4에서 물리 호스트의 수는 4로 고정되어 있고 가상 머신의 수는 유동적이다. 이러한 방식으로 가상 머신을 각각의 물리호스트에 1, 2 그리고 4개씩 배정한다. 따라서 3개의 시나리오가 존재한다. 이를 이용하여 pCloud 서비스 모델은 식 3.11과 같이 정의 될 수 있다.

$$pCloud = \{pS_1(4,4), pS_2(4,8), pS_3(4,16)\} \quad (3.11)$$

$$pS_1(4,4) = \{P_1(V_1), P_2(V_2), P_3(V_3), P_4(V_4)\}$$

$$pS_2(4,8) = \{P_1(V_1, V_2), P_2(V_3, V_4), P_3(V_5, V_6), P_4(V_7, V_8)\}$$

$$pS_3(4,16) = \left\{ \begin{array}{l} P_1(V_1, V_2, V_3, V_4), P_2(V_5, V_6, V_7, V_8) \\ P_3(V_9, V_{10}, V_{11}, V_{12}), P_4(V_{13}, V_{14}, V_{15}, V_{16}) \end{array} \right\}$$

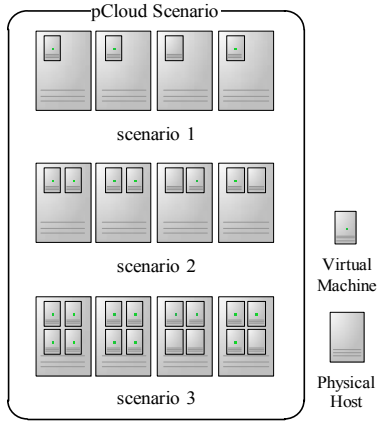


그림 4. pCloud 서비스 모델
Fig. 4. pCloud Service Model in PCIA

이상을 바탕으로 PCIA의 vCloud와 pCloud에 의해 제공되는 서비스 모델의 성능 변화는 다음 장에서 논의한다. PCIA 모델의 가상 자원과 물리 자원의 프로비저닝 변화에 따른 성능 변화는 클라우드 서비스 제공자와 클라우드 사용자에게, 시스템 관리 비용을 감소시키고 클라우드 구매 비용을 줄이기 위해 효율적으로 활용될 수 있다. 이러한 PCIA 모델은 물리적 자원의 제한 상황에서 사용자의 요구에 따른 적절한 수의 가상 자원을 선택하는데 도움을 주고 사용자의 가상머신 요구에 따른 적절한 물리 호스트의 선택에도 도움을 준다.

IV. 성능 평가

이 장에서는 PCIA 모델의 성능을 평가하고 이에 대한 분석결과를 소개한다. 또한 성능 수준에 영향을 주는 다양한 어플리케이션 특성과 시스템 구성을 고려하여 이에 대한 분석결과도 함께 제시한다.

표 1. 시스템 환경
Table 1. System Environment

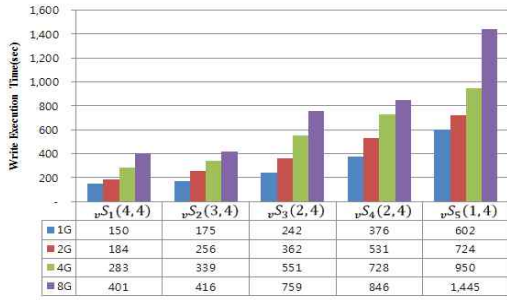
laas	Physical Host	VM
Processor	2.9GHz, Dual Core	1 vCPU
Main Memory	8GB, 1.333MHz DDR3	512M RAM
Storage Space	500GB, 7200 RPM	40GB Disk
NIC	100Mbps Ethernet NIC	
OS	Red Hat Enterprise Linux 5.6	
Xen VM	Xen 3.1.2	
JVM	Java 1.6.0_24	
Hadoop Frame	Hadoop 0.20.2	

실험을 위한 클라우드 시스템의 물리적 호스트 사양 및 실험 환경은 표 1과 같다. PCIA 환경에서의 Hadoop 분산 프레임 워크의 성능 측정을 위해서 다음과 같은 Hadoop 응용 프로그램을 사용한다. 응용의 특징 차이에 근거하여 다양한 성능 변화를 파악할 수 있도록, I/O 순차 쓰기와 읽기 [7], 랜덤 쓰기 [13], 정렬 [14] 그리고 워드 카운트 [15] 벤치마크를 포함시켰다. 또한 본 실험에서는 클라우드 시스템의 실험에 대한 정확도를 높이기 위해서 모든 실험을 10회씩 수행하였으며, 반복 실험 중 최고치와 최저치를 제외하고, 나머지 8회의 결과 값에 대한 평균값을 제시하였다.

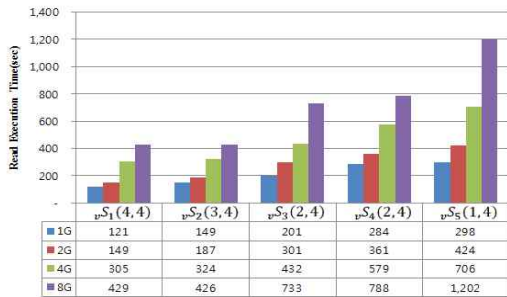
첫번째 실험으로 TestDFSIO를 활용하여 vCloud 환경에서 성능 평가를 수행 하였다. 그림 5에서 보여지듯이, $vS_1(4,4)$ 에서 $vS_5(1,4)$ 로 변화함에 있어, 각각의 VM이 독자적인 물리적 호스트에 할당 될수록, $vS_1(4,4)$ 가 최소의 실행 시간과 최대의 처리량을 나타냄을 알 수 있다.

이러한 성능 변화에 주된 요인은 네트워크와 관련된 오버헤드이다. $vS_1(4,4)$ 와 다른 시나리오 사이의 주된 차이점은, $vS_1(4,4)$ 에서는 네트워크 채널이나 대역폭에 대한 다른 가상머신과의 경쟁 없이, 각각의 가상머신이 자신만의 네트워크에 위치했다는 것이다. 하지만 $vS_3(2,4)$ 에서는 두 대의 가상머신이 하나의 네트워크 인터페이스를 공유한다. $vS_5(1,4)$ 에서는 네 대의 가상머신이 하나의 물리적 호스트에서 경쟁한다. 따라서 이로 인한 병목현상이 발생하며, 네트워크 오버헤드가 보장되지 않은 상황에서 $vS_1(4,4)$ 의 자원분배가 가장 큰 이점을 가진다고 추론할 수 있다.

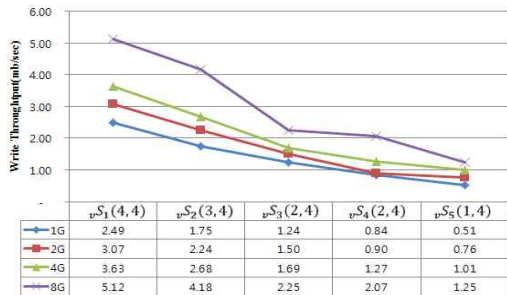
네트워크 채널, 즉 네트워크 대역폭과 함께 vCloud에 영향을 미치는 또 다른 요소는 물리 호스트의 저장장치를 공유하면서 생기는 I/O 채널 경쟁이다. $vS_1(4,4)$ 에서 각 가상머신은 각 물리 호스트에 독립적으로 위치했다. 이는 각 가상머신이 물리 호스트에서 별도의 I/O 프로세싱을 보장 받았다는 것을 의미한다. 이로 인해 실행 시간 감소라는 결과를 가져왔으며, 이 요소는 특히 I/O-bound 어플리케이션에서 중요하다.



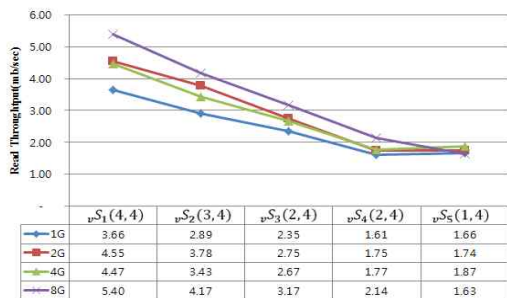
(a) write execution time



(b) read execution time



(c) write throughput



(d) read throughput

그림 5. vCloud에서 TestDFSIO 읽기 및 쓰기 연산
Fig. 5. Read and Write of TestDFSIO in vCloud

이러한 결과와 분석으로부터 다양한 클라우드 인프라스트럭처에서 충분한 레벨의 성능을 보장하기 위해서는, 가상 머신이 분리된 네트워크 채널과 독립적인 물리 I/O를 할당 받아야 한다는 것을 추론할 수 있다. 추가로 이것은 vCloud에서 적절한 시스템 선택의 중요성을 보여주고 있다. 그러므로 만약 사용자가 퍼블릭 클라우드 환경에서 충분한 물리 호스트를 활용할 수 있도록 허락된다면 각 가상머신을 배타적인 호스트에 분산시키는 것이 네트워크와 I/O 관점에서 가장 좋은 성능을 제공한다고 할 수 있다.

다음으로 표 2는 메모리의 크기가 클라우드 플랫폼에 미치는 영향력을 분석한 실험 결과이다. 표 2에서 가상머신의 메모리 용량을 256MB, 521MB 그리고 1024MB로 변화시켜 가며 TestDFSIO 벤치마크를 수행하였다. 특히, 그림 5에서 소개된 결과 가운데 최상의 시나리오 $vS_1(4,4)$ 에서 1~8GB의 데이터 사이즈 변화에 따라 메모리의 영향력을 확인해 보았다.

표 2에서 보여지듯이 실행 시간은 메모리 사이즈에 큰 영향을 받는다. 추가적으로 성능에 결정적인 영향을 미치는 주목할 만한 점은 비교적 작은 사이즈의 데이터로 테스트 했을 때에는 읽기 어플리케이션이 쓰기 어플리케이션보다 짧은 실행시간을 보여줬다는 것이다. 반대로 데이터의 사이즈가 8~16GB로 증가했을 때에는 그 결과는 반대였다. 즉, 쓰기 어플리케이션이 읽기 어플리케이션보다 짧은 실행시간을 보였다. 이러한 경향은 메모리와 데이터 사이즈가 커질수록 더 확대되어 나타났다. 표 2에서 *로 표시된 결과가 이러한 경향을 보여 준다.

이러한 분석으로부터 가상 머신에서 메모리 사이즈를 증가시키면 쓰기 어플리케이션은 읽기 어플리케이션보다 비교적 더 좋은 성능 향상을 제공한다고 말할 수 있다. 따라서 메모리

표 2. $vS_1(4,4)$ 의 읽기 및 쓰기 시간 변화 추이
Table 2. Read and Write Execution time in $vS_1(4,4)$

Data/RAM	WRITE(sec)			READ(sec)		
	256	512	1024	256	512	1024
1GB	150	51	35*	121*	49*	50
2GB	184	60*	37*	149*	109	111
4GB	283	93*	60*	255*	157	177
8GB	401*	154*	110*	429	277	310

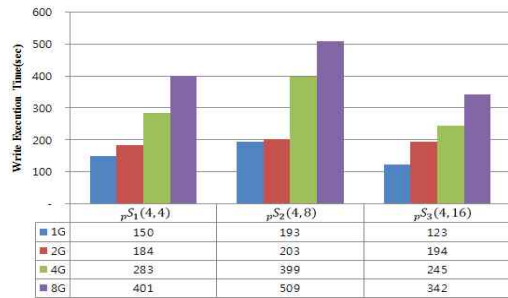
사이즈의 증가는 큰 데이터와 쓰기집중화(write-oriented)된 어플리케이션의 클라우드 환경에서 자원 프로비저닝에 중요한 후보가 될 수 있다고 결론 내릴 수 있다.

그림 6은 PCIA 모델의 pCloud 환경에서의 결과를 보여주고 있다. 전체적으로 그림 6에서 $pS_3(4,16)$ 이 $pS_2(4,8)$ 와 비교하여 읽기 연산과 쓰기 연산의 실행 시간 측면에서 결과 값에 큰 차이를 보인다. 다음은 이에 대한 분석이다.

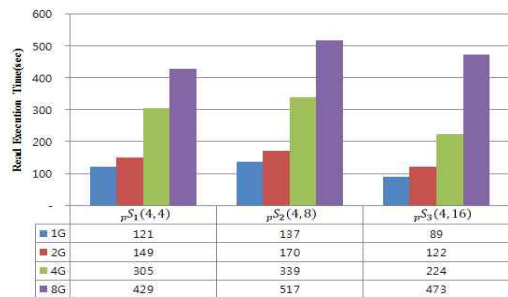
가상 머신의 개수 관점에서, $pS_1(4,4)$ 에서 $pS_2(4,8)$ 로 변화 시, VM의 수는 두 배가 되었다. Hadoop 컴퓨팅 시스템에서는 각각의 작업들에게 기본적으로 동일한 공간과 동일한 계산 능력이 할당된다. 따라서 데이터 처리 능력이 향상되리라 예상할 수 있지만, 주어진 실험 결과에서는 $pS_2(4,8)$ 이 $pS_1(4,4)$ 보다도 더 긴 실행 시간을 요구하였다.

이런 모순에 대한 주요 요인은 I/O 연산에 따른 병목현상 때문이다. 물리적 호스트에 맵핑되는 가상 머신의 수가 증가할 때 I/O 연산에 대한 오버헤드는 병목현상으로 나타난다. 어플리케이션의 실행 시간과 관련한 이러한 I/O 오버헤드는 결과적으로 전체 시스템의 성능 저하를 초래하게 된다.

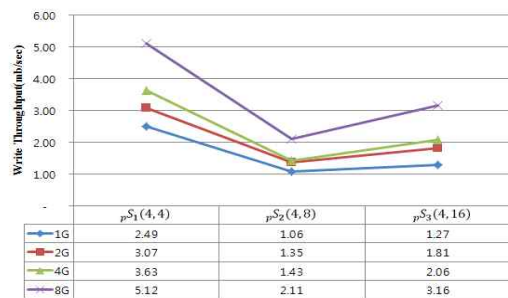
$pS_2(4,8)$ 과 $pS_3(4,16)$ 의 경우에 있어서의 성능을 비교해 보면, 두 가지 경우 모두 I/O 병목 현상을 가진다. 하지만, 두 배의 VM을 가지는 $pS_3(4,16)$ 의 경우를 살펴보면, $pS_2(4,8)$ 에 비해 최대 33% 더 뛰어난 성능을 보였다. 심지어 $pS_3(4,16)$ 의 성능이 $pS_1(4,4)$ 보다 최대 15% 실행 시간이 절약되었다. 성능 자체만을 놓고 보면 $pS_3(4,16)$ 이 최선의 선택일 수 있으나 VM의 할당 비용까지 고려한다면, $pS_1(4,4)$ 가 최선의 구성이라 할 수 있다. 왜냐하면 $pS_1(4,4)$ 의 경우보다 $pS_3(4,16)$ 은 4배 더 많은 VM을 요구하기 때문이다. 이러한 관계는 클라우드 사용자의 예산에 따라서 신중하게 고려되어야 한다.



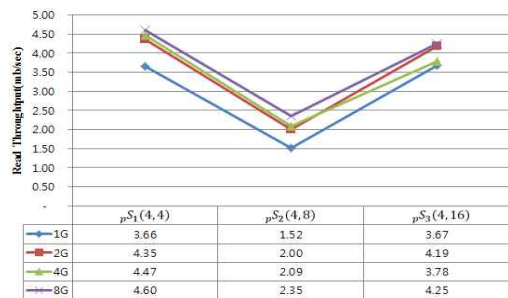
(a) write execution time



(b) read execution time



(c) write throughput



(d) read throughput

그림 6. pCloud에서 TestDFSIO 읽기 및 쓰기 연산
Fig. 6. Read and Write of TestDFSIO in pCloud

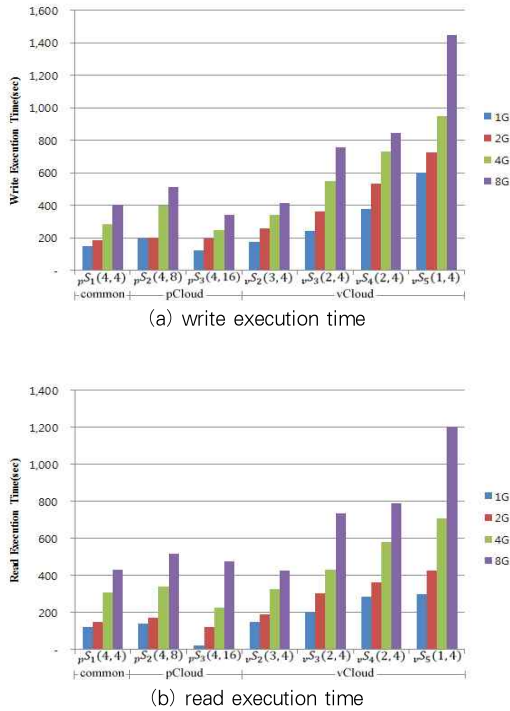


그림 7. vCloud와 pCloud의 실행 시간
Fig. 7. Execution Time between vCloud and pCloud

그림 7은 vCloud와 pCloud의 모든 시나리오에 대한 실행시간 측정 결과이다. 주어진 결과로부터 다음과 같은 사실을 확인할 수 있다. vCloud 환경과 비교하여 각 시나리오에 대한 pCloud의 성능 변화는 vCloud 보다 그 변화량이 적다. 이러한 결과는 물리적 호스트의 수가 고정된 제한적인 자원 상황 하에서, 조금이라도 더 나은 성능을 얻기 위해서는 pCloud의 프로비저닝을 보다 더 신중하게 수행할 필요가 있다는 것을 의미한다. 이와 비교하여 vCloud 환경에서는 자원 프로비저닝의 관점에서 볼 때 성능의 변화가 pCloud보다 크다. 따라서 pCloud에 비해서 vCloud의 자원 프로비저닝 전략이 성능 변화에 더 큰 영향을 준다고 할 수 있으며, 자원 프로비저닝의 판단은 용이할 수 있으나, 그에 따른 성능의 변화도 클 수 있다는 사실을 클라우드 서비스 제공자와 사용자 모두 인지해야 한다.

vCloud와 pCloud 환경에서 서로 다른 어플리케이션의 특징에 따른 성능 다양성을 분석하기 위해서 Sort와 WordCount 벤치마크를 활용하여 추가 실험을 수행하였다. 해당 실험 결과가 그림 8에 소개되어 있으며, 이는 vCloud

환경에서 실행시간의 변화를 나타내고 있다.

이 실험 환경에서 자원 프로비저닝의 변화에 따른 성능 변화를 분석해 보면 다음과 같다. 그림 8에서는 그림 5에서처럼 $vS_1(4,4)$ 가 짧은 실행시간을 보이고 $vS_5(1,4)$ 로 갈수록 실행시간이 증가하는 경향이 관찰되었다.

또한, 그림 8의 $vS_3(2,4)$ 와 $vS_4(2,4)$ 의 실험결과로부터 I/O에 대한 영향을 더 정확히 파악 할 수 있다. $vS_3(2,4)$ 와 $vS_4(2,4)$ 는 같은 수의 물리적 호스트와 가상머신을 가지고 있다. 이 환경에서 Sort 벤치마크는 거의 같은 실행 시간을 요구한다. 하지만, WordCount 벤치마크는 $vS_3(2,4)$ 에서 보다 나은 성능을 보여주었다. 그 이유는 I/O 오버헤드 관점에서 Sort가 WordCount 보다 덜 영향을 미치기 때문이다. 이러한 결과로부터 WordCount와 같이 I/O에 집중화된 응용에 있어서는, 자원 프로비저닝 전략에 있어서 이에 대한 고려를 우선해야 한다는 것을 의미한다.

그림 8에 소개된 실험 결과를 통해 추론할 수 있는 또 다른 사항은 실행 시간과 물리 호스트의 수에는 "Quasi-Constant 관계"가 있는 사실이다. 그림 8에서 실행 시간과 물리 호스트의 수의 곱으로 계산된 비용을 살펴보면, Sort에서는 1332와 1430사이를 보였고, WordCount에서는 2192에서 2811사이를 보였다. 즉, Sort에서의 비용값 분포가 WordCount 보다 더 균일하다. 비용값에서 이러한 균일한 분포는 실행된 응용이 계산 중심의 CPU-bound 프로그램이라는 점을 나타낸다. 이러한 결과로부터 클라우드 사용자는, 요구되는 실행 시간과 요구되는 연산력의 관점에서 클라우드 사용 비용에 따른 적절한 선택을 할 필요가 있다고 결론 내릴 수 있다.

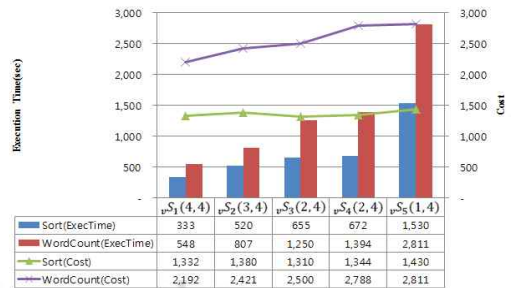


그림 8. vCloud에서 Sort와 WordCount 실험 결과
Fig. 8. Sort and WordCount in vCloud

그림 9는 pCloud 환경에서 Sort와 WordCount 벤치마크의 실행 결과를 보여준다. 이 실험에서 자원 프로비저닝 변화에 따른 성능 변화 분석은 다음과 같다. 그림 9에서는 가상 머신의 수가 증가함에 따라 Sort의 실행시간이 감소함을 보여주고 있다. 하지만 $pS_2(4,8)$ 에서의 WordCount 실행 시간은 그림 6의 pCloud TestDFSIO 벤치마크와 같은 형태의 실행 시간 변화를 보였다.

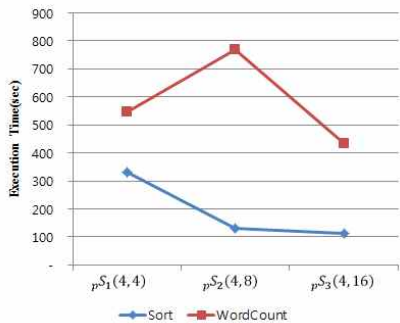


그림 9. pCloud에서 Sort와 WordCount 실행 결과
Fig. 9. Sort and WordCount in pCloud

이러한 형태의 실행 시간 패턴은 I/O-bound 응용의 특징이다. 그리고 WordCount의 실행 결과가 TestDFSIO와 비슷한 것은 WordCount도 I/O 성능에 큰 영향을 받는다는 것을 의미한다. 따라서 클라우드 서비스 제공자나 사용자가 I/O-bound 어플리케이션을 사용할 경우 I/O 요인을 첫번째 자원 프로비저닝 전략으로 고려해야 할 것이다.

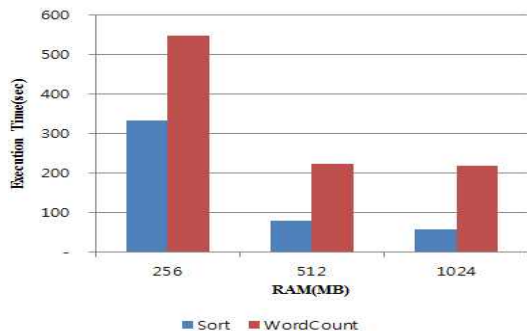


그림 10. $vS_1(4,4)$ 에서 Sort와 WordCount 실행 시간
Fig. 10. Execution Time of Sort and WordCount in $vS_1(4,4)$

마지막으로 메모리 용량에 따른 영향력을 분석해 보면 그림 10과 같다. 시나리오 $vS_1(4,4)$ 에서, 메모리의 용량을

256MB ~ 1024MB로 변화시키며 측정된 실행 시간을 나타낸다. 이 실험으로 인한 프로비저닝 변화에 따른 성능 변화는 다음과 같다.

메모리 사이즈가 256MB에서 512MB로 증가할 때 Sort 성능은 4배가량 증가하였다. 그리고 WordCount의 성능은 2.4배 증가하였다. 하지만 메모리 사이즈가 512MB에서 1024MB로 증가할 때에는 Sort와 WordCount 모두 큰 성능 향상을 보이지 못했다. 이러한 결과 역시 클라우드 서비스 제공자와 사용자 모두에게 성능과 가격의 손익관점 (trade-off)에서 적절한 메모리 사이즈를 결정하는 것이 중요하다라는 것을 보여준다.

V. 결론

IaaS는 기본적인 인프라스트럭처 서비스를 수행함으로써 신뢰할만하고 확장 가능한 클라우드 컴퓨팅 환경을 제공한다. 본 논문에서는 PCIA라고 명명된 새로운 타입의 공용 클라우드 구조를 제안하였다. 제안된 모델은 고정된 가상 머신을 가진 vCloud와 고정된 물리 호스트를 가진 pCloud로 정의된다. 아울러, PCIA 환경에서의 서비스 모델에 대한 개념을 소개하였다. PCIA 모델에서 가상 호스트는 다양한 물리적 구조에 따른 사용자의 요청에 따라서 혹은 다양한 어플리케이션의 특성에 따라서 서로 다른 물리적 호스트를 제공하는 것이 가능하다.

본 논문에서는 PCIA의 vCloud와 pCloud 각각의 서비스 모델에 대하여, 클라우드 시스템의 자원 프로비저닝과 성능 변화 사이의 관계에 대해서 분석하였다. 이를 통해 클라우드 환경에서 클라우드 서비스 제공자와 사용자 모두에게 중요한 기준이 될 수 있는 클라우드 서비스 선택의 기준을 제시하였다. 이러한 기준은 물리적 자원의 제한 하에서 최적의 성능을 도출하기 위한 적절한 수의 가상자원을 선택하는 것과 사용자의 가상머신 요청에 따른 물리적 자원의 적절한 할당에 있어서, 클라우드 서비스 제공자와 사용자 모두에게 클라우드 시스템 선택의 중요한 잣대로 활용될 수 있다.

참고문헌

- [1] K. Rangan, The Cloud Wars: \$100+ billion at stake. MerrillLynch, July 2008
- [2] L. Siegele, Let It Rise: A Special Report on

Corporate IT. TheEconomist, October 2008

[3] S. Sivathanu, L. Liu, Y.D. Mei, X. Pu, Storage Management in Virtualized Cloud Environment. 3rd International Conference on Cloud Computing, pages 204-211, July 2010

[4] V. Chang, G. Wills, D. De Roure, A Review of Cloud Business Models and Sustainability. International Conf. on Cloud Computing, pp. 43-50, July 2010

[5] W.T. Tsai, X. Sun, J. Balasooriya, Service-Oriented Cloud Computing Architecture. ITNG2010, pages 684-689, April 2010

[6] Iganacio M. Liorente, Cloud Computing for on-Demand Resource Provisioning. 7th NRENs and Grids Workshop, May 2009

[7] M.S. Rehman, M.F. Sakr, Initial Findings for Provisioning Variation in Cloud Computing. CloudCom2010, pages 473-479, December 2010

[8] P. Barham et al., Xen and the art of virtualization. SOSP'03, December 2003

[9] VMware, the powerful open source industry standard for virtualization, <http://www.vmware.com/>.

[10] Y. Zhang, G. Huang, X.Z. Liu, H. Mei, Integrating Resource Consumption and Allocation for Infrastructure Resources on-Demand. 3rd International Conference on Cloud Computing, pages 75-82, July 2010

[11] R. Aoun, E.A. Doumith, M. Gagnaire, Resource Provisioning for Enriched Services in Cloud Environment. CloudCom2010, pages 296-303, December 2010

[12] T.A. Henzinger, A.V. Singh, V. Singh, T. Wies, D. Zufferey, FlexPRICE: Flexible Provisioning of Resources in a Cloud Environment. CLOUD2010, pages 83-90, December 2010

[13] Hadoop RandomWriter, <http://wiki.apache.org/hadoop/RandomWriter/>.

[14] Hadoop Sort Benchmark, <http://wiki.apache.org/hadoop/Sort/>.

[15] Hadoop WordCount Benchmark, <http://wiki.apache.org/hadoop/WordCount/>

저 자 소 개



윤 빈 풍
 2012: 영남대학교
 컴퓨터공학과 공학사.
 현 재: 영남대학교
 기계공학과 석박사 연계 과정
 관심분야: 클라우드 시스템,
 운영체제, 고성능 컴퓨팅
 Email : love@ynu.ac.kr



곽 종 옥
 1998: 경북대학교
 컴퓨터공학과 공학사.
 2002: 서울대학교
 컴퓨터공학과 공학석사.
 2006: 서울대학교
 전기컴퓨터공학부 공학박사
 현 재: 영남대학교
 컴퓨터공학과 부교수
 관심분야: 컴퓨터 구조,
 고성능 컴퓨팅,
 임베디드 시스템
 Email : kwak@yu.ac.kr