

<https://doi.org/10.7236/IIBC.2019.19.3.17>

IIBC 2019-3-3

공용 클라우드 기반 PC 실습실 서비스의 실시간 비용 예측 모델 연구

A Study on Realtime Cost Estimation Model of PC Laboratory Service based on Public Cloud

조경운*, 신용현**

Kyung-Woon Cho*, Yong-Hyeon Shin**

요 약 IaaS는 인프라 하드웨어를 소유하지 않고 필요에 따라 임대하는 새로운 방식의 컴퓨팅 서비스로서 비용측면에서 매우 효율적인 것으로 알려져 있다. 일반적으로 사용량에 따른 과금 방식을 채택하므로 인프라 비용에 민감한 서비스를 운영하기에 적합하다. 이러한 서비스를 운영하는 책임자는 예상되는 소요 비용을 조기에 파악하여 클라우드 활용 정책을 적시에 변경하기를 원할 것이다. 그러나 클라우드 서비스 제공자들은 침수시간 지연된 과금 정보를 제공하여 신속한 대처가 불가능하다. 본 논문에서는 가상 머신 인스턴스 수준에서 사용량을 모니터링하고 이에 기반하여 실시간 IaaS 비용 예측 모델을 제안한다. 이 모델 검증에 위하여 1학기 동안의 공용 클라우드 기반의 PC 실습실 서비스를 운용하였으며, 이를 통하여 부과된 실제 금액과의 차이는 평균 5.2% 이하임을 확인하였다.

Abstract IaaS is well known as a very cost effective computing service which enables required infrastructures to be rented on demand without ownership of real hardwares. It is very suitable for price sensitive services due to pay-per-use style. Operators of such services would want to adjust utilization policy quickly by estimating costs for cloud infrastructures as soon as possible. However, swift response is not possible due to that cloud service providers provide a dozen or so hours delayed billing information. Our work proposes a realtime IaaS cost estimation model based on usages monitored by virtual machine instance. We operate PC laboratory service on a public cloud during full semester to validate our suggested model. From that experiment, an averaged disparity between estimation and actual cost is less than 5.2%.

Key Words : Realtime Cost Estimation Model, IaaS Metering, Public Cloud, PC Laboratory Service

I. 서 론

IaaS는 인프라 하드웨어를 소유하지 않고 필요에 따라 임대하는 새로운 방식의 컴퓨팅 서비스로서 비용측면

에서 매우 효율적인 것으로 알려져 있다^[8]. 일반적으로 사용량에 따른 과금 방식을 채택하므로 인프라 비용에 민감한 서비스를 운영하기에 적합하다^[1]. 이러한 서비스를 운영하는 책임자는 예상되는 소요 비용을 조기에 파

*정회원, 이화여자대학교 임베디드소프트웨어연구센터

**정회원, 서울과학기술대학교 컴퓨터공학과(교신저자)

접수일자 2019년 4월 10일, 수정완료 2019년 5월 10일

게재확정일자 2019년 6월 7일

Received: 10 April, 2019 / Revised: 10 May, 2019 /

Accepted: 7 June, 2019

**Corresponding Author: yshin@seoultech.ac.kr

Dept. of Computer Science & Engineering, Seoul National University of Science & Technology, Korea

악하여 클라우드 활용 정책을 적시에 변경하기를 원할 것이다. 그러나 클라우드 서비스 제공자(CSP, Cloud Service Provider)들은 지연된 과금 정보를 제공하여 신속한 대처가 불가능하다^[2]. 이는 CSP 수준의 비용 산출은 대규모 배치 작업이기 때문에 적게는 수시간부터 많게는 일단위의 지연이 발생하기 때문이다. 만약 CSP에 의해 부과되는 클라우드 사용 비용을 실시간으로 파악할 수 있다면 관리 운영상의 선제적 조치가 가능할 것이다.

CSP에서 비용을 과금하는 방식은 다양한 측면의 자원들에 대한 사용량을 기반으로 한다^[3,4]. 과금의 대상이 되는 자원은 주로 CPU, 네트워크 그리고 스토리지로 크게 구분할 수 있다. CPU 과금은 프로세싱 능력에 따라 단위 시간 기준 요금에 상이하며, 사용 시간에 따라 비용이 비례하는 방식이다. 네트워크 비용은 전송량에 따라 과금이 결정되며, 스토리지는 시간당 공간 사용량과 초당 입출력 요청 회수(IOPS)를 조합한 방식을 주로 사용한다. 이처럼 각 자원의 사용량에 따른 비용의 합이 IaaS의 전체 사용 비용이 된다.

현재의 공용 클라우드 인프라 비용에 대한 산출 과정은 CSP에 의해 결정되며, 이 과정에서 사용자가 개입하지는 않는다. 그러나 가상 머신 인스턴스 수준에서 사용량을 실시간으로 모니터링할 수 있다면 이를 기반으로 IaaS 비용을 예측할 수 있다. 물론 인스턴스 내부에서 수집된 자원 사용량이 CSP에서 판단하는 자원 사용량과 큰 차이가 없을 만큼 정밀해야 한다. 본 논문은 공용 클라우드의 비용을 실시간으로 예측하기 위한 IaaS 비용 예측 모델을 제안한다. 이 모델 검증에 위하여 1학기 동안의 공용 클라우드 기반의 PC 실습실 서비스를 운용하였으며, 이를 통하여 부과된 실제 금액과의 차이는 평균 5.2% 이하임을 확인하였다.

II. 클라우드 PC 실습실 플랫폼

기존의 컴퓨터 실습실은 여러 강좌의 많은 사용자들이 공유하여 활용하며, 주로 Windows 기반의 PC 환경이다. 이 때문에 PC 자원의 관리 부담과 함께, 다양한 교과목과 사용자들의 운영체제나 S/W적인 요구사항을 충족할 수 없는 경우가 빈번하게 발생한다. 이에 대한 해결 방안으로 멀티부팅, 운영체제 스트리밍, PC 가상화 등 다양한 방법이 시도되었으나, 관리나 비용의 문제를 완전히 해결하지 못하였다. 다양한 강의 환경과 수강생 고유의 PC 환경은 템플릿화된 가상 환경을 지원하는 공용 클

라우드를 활용하여 해결가능하다.

CLABO(CLOUD LABORatory)^[5]는 공용 클라우드 기반의 PC 실습실 환경 솔루션으로서 수강생들은 자신만의 가상 머신을 할당받아 장소의 제약 없이 컴퓨팅 환경을 이용할 수 있다. 강사는 강좌에 최적화된 성능, 운영체제와 응용 소프트웨어 환경 등의 컴퓨팅 환경을 가상 머신 템플릿으로 구성할 수 있고, 모든 수강생들에게 일괄 배포할 수 있다. 원격 데스크탑이나 일반 터미널을 이용하여 가상 머신 환경에 접속할 수 있으며, 일정 시간 사용하지 않으면 자동으로 가상 머신이 중단되어 불필요한 비용 낭비를 방지할 수 있다. 수강생별 최대 이용 가능 시간을 설정하여 클라우드 사용 비용을 제어할 수 있으며, 본 논문에서 제안한 IaaS 비용 예측 모델에 따라 실시간으로 비용을 확인할 수 있다.

III. 클라우드 인프라의 비용구성요소

클라우드 인프라 비용의 실시간 예측 모델을 설명하기 위해서 인스턴스 비용의 구성 요소를 분석해야 한다. 실행중인 인스턴스 비용($Cost_{inst}^{run}$)은 수식 (1)과 같이 CPU, 스토리지 및 네트워크의 각 자원별 요금의 합산으로 구성된다. CPU 비용은 인스턴스 사용시간(t)과 인스턴스 형식(T)에 의존적으로 결정되며, 스토리지 비용은 t , T 및 단위시간 스토리지 사용량(f_{sto})에 의해 산정된다. f_{sto} 는 특정 시각의 스토리지 사용량을 나타내는 것으로 이해할 수 있다. 네트워크 비용은 t 와 단위시간 네트워크 사용량(f_{net})에 의존적이다.

$$Cost_{inst}^{run}(t, T, f_{sto}, f_{net}) = Cost_{cpu}^{run}(t, T) + Cost_{sto}^{run}(t, T, f_{sto}) + Cost_{net}^{run}(t, f_{net}) \quad \dots (1)$$

각 자원별 비용들을 세분화하면, CPU 비용인 $Cost_{cpu}^{run}$ 는 인스턴스 형식에 의존적인 단위 시간 비용(P_{cpu}^T)을 기준으로 사용시간에 비례한다.

$$Cost_{cpu}^{run}(t, T) = P_{cpu}^T t \dots \dots \dots (2)$$

스토리지 비용 $Cost_{sto}^{run}$ 는 인스턴스의 스토리지 공간 사용에 따른 비용($Cost_{vol}^{run}$)과 스토리지 입출력 사용량에 따른 비용($Cost_{io}^{run}$)으로 세분할 수 있다. $Cost_{vol}^{run}$ 는 단위시간당 T 의 스토리지 공간 사용 비용(P_{vol}^T)과 t 의 곱으로 표현가능하다. $Cost_{io}^{run}$ 는 단위

스토리지 입출력 사용 비용(P_{io})과 t 시간동안의 f_{sto} 의 누적양의 곱으로 구성된다.

$$Cost_{sto}^{run}(t, T, f_{sto}) = Cost_{vol}^{run}(t, T) + Cost_{io}^{run}(t, f_{sto}) \\ = P_{vol}^T t + P_{io} \int f_{sto} \dots\dots (3)$$

네트워크 비용은 단위 데이터 전송량에 대한 비용(P_{net})을 기준으로 사용량에 비례하여 산출된다. P_{net} 은 보통 단위 입력 데이터 양에 대한 비용(P_{net}^{in})과 출력 데이터 양에 대한 비용(P_{net}^{out})으로 구성된다.

$$Cost_{net}^{run}(t, f_{net}) = P_{net} \int f_{net} = \\ P_{net}^{in} \int f_{net}^{in} + P_{net}^{out} \int f_{net}^{out} \dots\dots (4)$$

인스턴스의 비용은 수식(5)와 같이 정리할 수 있다. P_{cpu}^T 와 P_{vol}^T 는 인스턴스 유형에 따라 결정되는 고정값이며, P_{io} , P_{net} 역시 CSP에 의해 사전에 책정되는 고정값이다. 이러한 고정 계수를 제외한 나머지 항목들은 인스턴스 사용시간과 시간에 의존적인 스토리지 및 네트워크 사용량이다. 인스턴스 비용은 시간에 비례하여 비용이 증가하며, 실시간으로 인스턴스 비용을 정확히 산정하고자 한다면, 인스턴스 사용시간과 스토리지와 네트워크 사용량을 실시간으로 정확하게 측정할 수 있어야 한다.

$$Cost_{inst}^{run}(t, T, f_{sto}, f_{net}) = \\ P_{cpu}^T t + P_{vol}^T t + P_{io} \int f_{sto} + P_{net} \int f_{net} = \\ (P_{cpu}^T + P_{vol}^T)t + P_{io} \int f_{sto} + P_{net} \int f_{net} \quad (5)$$

IV. 비용 구성 요소의 사용량 검증

인스턴스 비용의 핵심적 결정 요소인 인스턴스 사용시간과 자원에 대한 사용량은 실시간으로 사용시간을 추적하고 자원을 모니터링함으로써 파악될 수 있다. 그러나 CSP들은 실시간 모니터링 기능을 제공하지 않으며, 클라우드 사용자가 직접 자원 사용량에 대한 실시간 모니터링 기능을 수행하여야 한다. CLABO 서비스는 인스턴스에 대한 다양한 모니터링 기능을 제공하고 있으며, AWS 상에서 제공되는 과금 기초 자료^[6]와 비교하여 실시간 사용량 모니터링의 적정성을 비교해 볼 것이다. 이는 역으로 AWS 모니터링 기능의 정확도를 상호 검증하는 역할

표 1. 사용량 검증 시나리오(19개)
 Table 1. Usage Verification Scenarios(19)

시나리오 코드	내용
Lixm	Linux 20분 기동후 단순 종료, t2.micro 사용
Lixs	위와 동일, 단 t2.small 인스턴스 사용
Wixm	Windows 20분 기동 후 단순 종료, t2.micro
Wixs	위와 동일, 단 t2.small 인스턴스 사용
Lins	일반 화면 크기로 VNC 연결 후 20분뒤 종료
Lifs	위와 동일, 단 전체 화면 크기로 VNC 연결
Wins	원격 데스크탑 연결 후 20분뒤 종료
Wifs	위와 동일, 단 전체 화면으로 연결
Ltxs	ssh 연결 후 20분동안 관리 작업 수행
Lgxs	ssh 연결 후 vi로 코드 작성 및 gcc로 빌드
Wvfs	Visual Studio를 통한 코드 작성 및 빌드
Lwfs	Firefox를 사용하여 웹탐색
Wwfs	Internet Explorer를 이용하여 웹탐색
Lefs	Eclipse를 이용하여 Java 프로그래밍
Wefs	위와 동일, 단 Windows 환경에서 수행
Lafs	Android Studio를 이용하여 앱개발, t2.small
LafM	위와 동일, 단 t2.medium 인스턴스 사용
Wafs	Android Studio를 이용하여 앱개발
WafM	위와 동일, 단 t2.medisum 인스턴스 사용

도 한다. 사용량 검증 시험은 표 1 과 같이 총 19가지의 다양한 인스턴스 활용 시나리오를 기준으로 인스턴스 사용시간, 스토리지 및 네트워크에 대한 사용량 비교를 진행하고자 한다.

시나리오 코드는 운영체제, 응용, 인스턴스 접속 방법, 인스턴스 유형별로 구분되는 4자리 영문자이다. 사용된 운영체제는 Linux(시나리오 코드 Looo 형식)와 Windows(Wooo 형식)이다. 2번째 영문자는 빈작업(i), 터미널작업(t), Linux개발도구(g), Windows개발도구(v), 웹도구(w), eclipse(e), Android Studio(a)로 대학교육과정에서 주로 사용되는 개발 도구 위주이다. 3번째 영문자는 VDI 연결여부에 대한 구분으로서, 연결없음(x), 창화면 VDI(n), 전체화면 VDI(f)로 구성된다. 마지막 영문자는 가상 머신의 인스턴스 종류로서, t2.micro(m), t2.small(s), t2.medium(M)이 사용되었다.

그림 1은 AWS 사용보고서상의 EC2 사용시간과 CLABO 모니터링상의 사용시간의 차이를 나타낸 것이다. 사용량 검증 시나리오별로 20분간 구동하였을 때의 사용시간으로서 CLABO 모니터링에 의한 사용시간이 평균 약 1.38분 정도 길다. EC2는 Windows 플랫폼의 경우는 1시간 단위로 시간이 산출되어 비교 대상에서는 제외하였다. 그림 2는 Lixm 시나리오의 사용시간을 20분

에서 70까지 변경하여 사용시간의 차이를 나타낸 것이다. 평균적으로는 CLABO 모니터링이 약 1.1분 가량 길다. CLABO 모니터링이 EC2 사용시간보다 전반적으로 긴 것은 AWS API를 통하여 인스턴스 종료 여부를 확인하는 과정에서 어느정도의 지연시간이 발생하기 때문이다. 그러나 전체적으로 인스턴스 사용 시간을 실시간으로 정확히 파악하는 것은 약간의 보정만으로 어렵지 않게 구현가능하다.

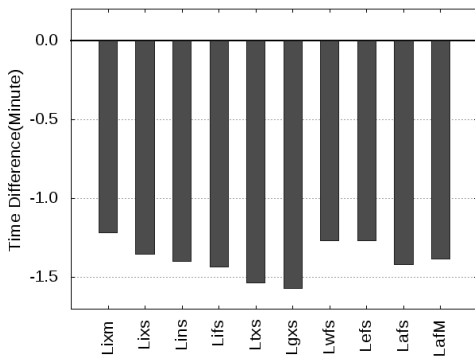


그림 1. 시나리오별 인스턴스 사용시간 차
Fig. 1. Instance usage time difference per scenario

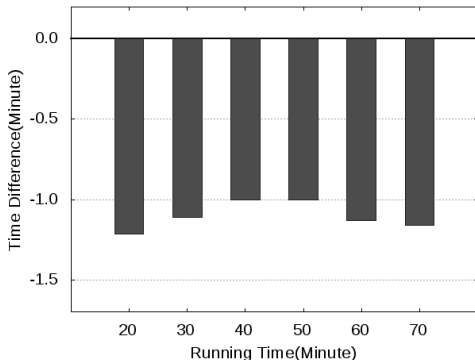


그림 2. 수행시간별 인스턴스 사용시간 차
Fig. 2. Instance usage time difference per running time

네트워크 입출력 사용량에 대한 차이는 그림 3, 그림 4에 표시하였다. 네트워크 입출력에 대해 전송량이 1MiB보다 큰 시나리오에 대해 오차 비율을 나타낸 것이다. CLABO 모니터링이 AWS 사용량 보고서보다 네트워크 입력 데이터를 많이 사용한 것으로 측정되었다. Lwfs와 Wwfs는 100MiB 수준 이상으로 가장 많이 사용하는 시나리오였으나 사용량의 오차는 1% 내외로 크지 않다.

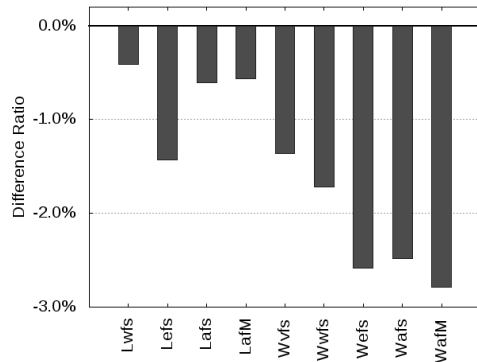


그림 3. 네트워크 입력량 차이율
Fig. 3. Inbound networking size difference ratio

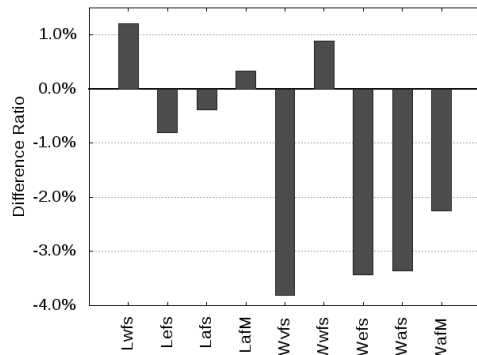


그림 4. 네트워크 출력량 차이율
Fig. 4. Outbound networking size difference ratio

네트워크 출력의 경우에도 네트워크 입력과 마찬가지로 전송량의 오차는 크지 않아, 실시간 네트워크 전송량을 모니터링하여 네트워크 출력에 따른 실시간 비용을 추산할 수 있다.

많은 CSP에서 스토리지 입출력 사용량으로 활용하는 스토리지 입출력 요청수는 인스턴스 수준에서 정밀하게 파악하기는 용이하지 않다. 먼저 인스턴스 내부 모니터링 루틴이 활성화되기 전 발생한 입출력 활동들은 원천적으로 파악이 어렵다. 또한 스토리지 입출력량과 달리 입출력 요청들은 스토리지 계층의 정책에 따라 분리되거나 병합될 수 있기 때문이다.

그림 5는 CLABO에서 모니터링한 스토리지 입력 요청수(A)와 AWS 스토리지 사용량 보고서에서 확인된 입력 요청수(B)의 차이에 대한 비율((B-A)/B)이다. A 요청수는 인스턴스 IO 계층에서 모니터링된 개수로서 CLABO 모니터링 개시전 유발된 스토리지 요청수를 포함시키기 위해 스토리지 활동이 거의 없는 Lixm과 Wixm 시나리오를 기준으로 요청수를 가산하였다. 전반

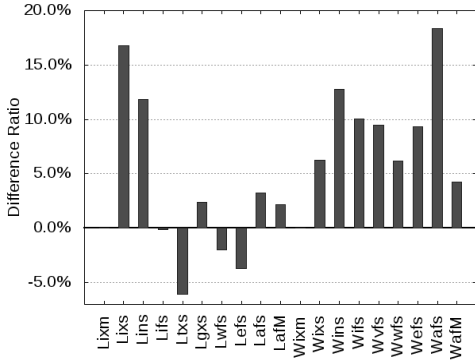


그림 5. 스토리지 읽기 횟수 차이율
 Fig. 5. Storage read count difference ratio

적으로 A, B간 스토리지 읽기 요청수의 차이가 많아 실시간으로 정확한 사용량을 추적할 수 없는 것으로 판단되지만, Lwfs 시나리오와 같이 읽기 요청수가 2,000을 넘는 Linux 운영체제의 경우에는 5% 미만의 차이를 보인다. Windows 운영체제 기반의 시나리오들은 모두 B 요청수가 월등히 많은 경향을 보인다. 그림 6은 스토리지 쓰기 요청수에 대한 비율을 나타낸 것이다. 스토리지 읽기의 경우와 마찬가지로 경향이 있으나 차이가 좀 더 심화되어 나타난다.

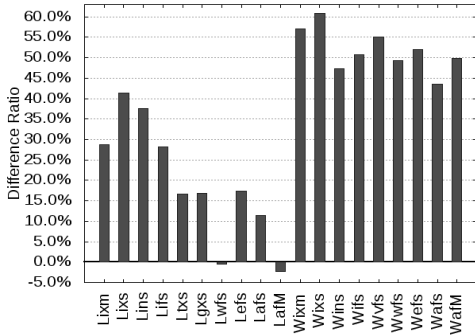


그림 6. 스토리지 쓰기 횟수 차이율
 Fig. 6. Storage write count difference ratio

V. 실시간 인스턴스 비용 예측 모델

인스턴스 비용 예측은 인스턴스 상태에 따라 구분해서 산정되어야 한다. 실행중인 인스턴스에 대한 실시간 비용 예측은 수식 (5)에 기반하여 측정 가능하지만, 스토리지 입력 요청수는 모니터링 개시전 초기 사용분을 보정하여 예측을 수행해야 한다. 중단된 인스턴스의 경우에는 해당 인스턴스의 스토리지 저장비용이 해당 기간동안의 과금되므로 중단된 인스턴스 비용은 수식 (6)과 같이 산정가

능하다. 이 상태의 인스턴스 비용은 단위시간당 T 의

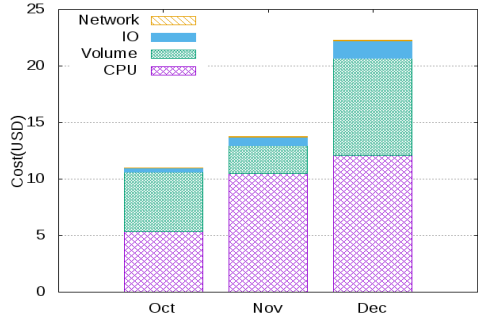


그림 7. 월별 CLABO 1개 강좌 IaaS 비용
 Fig. 7. Per-month IaaS cost for CLABO single class

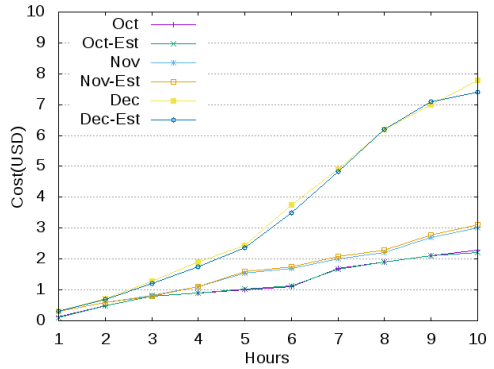


그림 8. 시간별 누적 IaaS 비용
 Fig. 8. Accumulated IaaS cost per hour

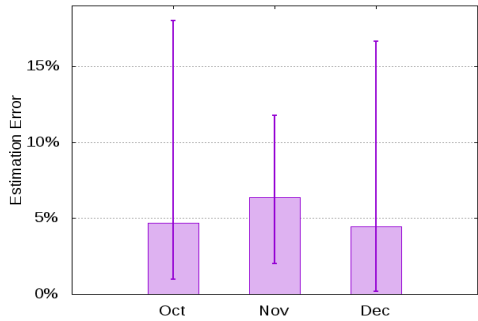


그림 9. IaaS 비용 예측 오차
 Fig. 9. IaaS cost estimation error

스토리지 공간 사용 비용(P_{vol}^T)과 t 의 곱으로 구성된다.

$$Cost_{inst}^{stop}(t, T) = Cost_{sto}^{stop}(t, T) = P_{vol}^T t \dots \dots \dots (6)$$

따라서, 실행중 상태와 중단 상태를 모두 합한 인스턴스의 비용($Cost_{inst}$)은 수식 (7)과 같이 정리된다.

$$Cost_{inst}(t_{run}, t_{stop}, T, f_{sto}, f_{net}) = Cost_{inst}^{run}(t_{run}, T, f_{sto}, f_{net}) + Cost_{inst}^{stop}(t_{stop}, T) \cdot (7)$$

VI. CLABO 서비스 비용 예측 실험

CLABO 서비스는 모든 생성되는 인스턴스에 대해서 실시간으로 비용을 추적하고, 사용자 및 강좌별로 비용 취합 기능을 제공한다. 실시간 인스턴스 비용 예측 모델의 정확도를 검증하기 위해 CLABO 서비스 기반으로 50여명의 수강생들을 대상으로 한 학기동안 진행된 강좌를 통해 예측 비용과 공용 클라우드 사업자에 의해 과금된 비용을 비교하고자 한다.

그림 7은 CSP에서 월별로 부과한 금액을 나타낸 것이다. 3개월에 대해서 인스턴스(CPU) 사용비용이 가장 큰 부분을 차지한다. 스토리지(VOLUME, IO)는 월별로 차이는 있지만, 15~50%까지 전체 비용을 차지한다. 중단된 인스턴스가 없다면 스토리지 불륨 사용에 따른 비용은 5% 미만일 것으로 예상된다. 그 외 네트워크나 스토리지 입출력에 의한 비용은 미미한 수준이다. t2.micro 기준의 Linux를 사용하여 전체 비용은 25USD 미만으로 매우 저렴한 편이다.

그림 8은 CLABO 실시간 비용 예측 기능을 통한 전체 인스턴스의 비용과 실제 CSP에 의해 부과된 비용을 1시간 단위로 표시한 것이다. AWS 사용보고서에서 제공되는 과금 정보가 1시간 단위의 정밀도를 가지고 있다. 그림 9는 월별 인스턴스 비용의 차이를 나타낸 것으로서 평균적으로 약 5.18%의 정확도를 가진다.

VII. 결론

클라우드는 총소유비용(TCO) 측면에서 매우 유리한 것으로 알려져 있지만, 유휴 인스턴스 사용이나 일시적인 사용량 폭증 등의 이유로 불필요한 비용이 발생할 가능성이 존재한다^[7,9]. 클라우드 서비스 이용자는 비용 모니터링을 면밀히하여 클라우드 운영 정책을 수행해야 하지만, CSP에서 제공하는 과금 정보는 십수시간 지연된 것으로서 실시간성을 보장하지 못한다. 본 논문에서 제안한 IaaS 비용 예측 모델은 실시간 비용 모니터링이 가능하며, 공용 클라우드 상의 PC 실습실 서비스를 통해 검증하여 5.2% 이하의 오차로 비용을 예측할 수 있다.

References

- [1] Bhardwaj, Sushil, Leena Jain, and Sandeep Jain. "Cloud computing: A study of infrastructure as a service (IAAS)." *International Journal of engineering and information Technology* 2.1 (2010): 60–63.
- [2] Chaudry, Rabia, et al. "Consumer Monitoring of Infrastructure Performance in a Public Cloud." *International Conference on Web Information Systems Engineering*. Springer, Cham, 2014.
- [3] Al-Roomi, May, et al. "Cloud computing pricing models: a survey." *International Journal of Grid and Distributed Computing* 6.5 (2013): 93–106. DOI: <https://doi.org/10.14257/ijgcd.2013.6.5.09>
- [4] Kim, Julie, and Hyokyung Bahn. "An Efficient Log Data Management Architecture for Big Data Processing in Cloud Computing Environments." *The Journal of The Institute of Internet, Broadcasting and Communication* 13.2 (2013): 1–7. DOI: <https://doi.org/10.7236/JIIBC.2013.13.2.1>
- [5] CLABO service, <http://sys-sw.clabo.codemayo.com>
- [6] AWS Cost and Usage Report, <https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/billing-reports-costusage.html>
- [7] Martens, Benedikt, Marc Walterbusch, and Frank Teuteberg. "Costing of cloud computing services: A total cost of ownership approach." *2012 45th Hawaii International Conference on System Sciences*. IEEE, 2012.
- [8] Kim, Bong-Hyun, and Sang-Young Oh. "A study on the SMART education system based on cloud and n-screen." *Journal of the Korea Academia-Industrial Cooperation Society* 15.1 (2014): 137–143. DOI: <https://doi.org/10.5762/KAIS.2014.15.1.137>
- [9] Ryu, G., and K. H. Yoo. "A DRM Based Content Protection Method in Content Oriented Educational Cloud System." *J. KIIT* 12 (2014): 137–147. DOI: <https://doi.org/10.14801/kiit.2014.12.12.137>

저 자 소 개

조 경 운(정회원)



- 1995년 2월 : 서울대학교 계산통계학과 학사
- 1997년 2월 : 서울대학교 전산과학과 석사
- 2012년 2월 : 서울대학교 컴퓨터공학부 박사
- 2000년 ~ 2016년 : ㈜클루닉스 연구

소장

- 2016년 4월 ~ : 이화여자대학교 임베디드소프트웨어연구센터 수석연구원/연구교수

신 용 현(정회원)



- 1988년 2월 : 서울대학교 계산통계학과 학사
- 1991년 2월 : 서울대학교 계산통계학과 석사
- 2004년 2월 : 서울대학교 전기컴퓨터공학부 박사
- 2005년 3월 ~ : 서울과학기술대학교

컴퓨터공학과 교수

※ This study was supported by the Research Program funded by the SeoulTech (Seoul National University of Science and Technology)