

이종 클라우드 플랫폼 연동을 위한 클라우드 브로커 구조

강동기, 이용규, 김대순, 윤찬현
한국과학기술원 전기및전자공학과
e-mail:dkkang@kaist.ac.kr, lyonggyu@kaist.ac.kr, sundae21@kaist.ac.kr,
chyoung@kaist.ac.kr

An Architecture of Cloud Broker for Linkage of Heterogeneous Cloud Platform

Dong-Ki Kang, Yong-Gyu Lee, Dae-Soon Kim, Chan-Hyun Yoon
Dept of Electrical Engineering, KAIST

요 약

단일 클라우드 환경에서 발생하는 비효율적인 자원 사용 및 서비스 신뢰성 문제를 극복하기 위하여 다중 클라우드 서비스 제공자가 존재하는 멀티 클라우드 환경을 고려할 수 있다. 멀티 클라우드 환경에서 각 클라우드 서비스 제공자는 이종의 클라우드 플랫폼을 구축하므로 이를 일괄적으로 접근, 제어 및 관리하기 위한 통합 모듈이 요구된다. 본 논문에서는 현재 오픈소스로 공개되어 있는 OpenStack, Eucalyptus 및 CloudStack과 같은 이종의 클라우드 플랫폼을 연동 및 접근할 수 있는 클라우드 브로커 구조를 제안한다.

1. 서론

클라우드 컴퓨팅 기술이란, 하드웨어 및 소프트웨어 자원을 사용하는 범용 사용자가 자원을 직접 구축하거나 운용할 필요 없이 자원 인프라를 이미 가지고 있는 클라우드 서비스 제공자에게 일정한 자원 사용 요금을 지불하고 서비스를 제공받는 온 디맨드(On-Demand) 형식의 새로운 컴퓨팅 서비스 패러다임을 의미한다[1].

클라우드 서비스를 제공하기 위한 클라우드 환경은 크게 단일 클라우드 서비스 환경과 멀티 클라우드 서비스 환경으로 나눌 수 있다. 단일 클라우드 서비스 환경은 사용자에게 서비스를 제공하는 주체가 유일하며, 제공자가 가지고 있는 데이터센터내의 자원만을 이용한다. 이러한 환경에서는 서비스를 요청하는 사용자의 수가 급속히 증가하거나, 응용이 요구하는 하드웨어 수준이 높아질 경우, 요구사항을 제대로 수용할 수 없고 서비스 수준(QoS : Quality of Service)을 만족할 수 없게 되므로, 신뢰성을 하락시킬 수 있다. 뿐만 아니라, 사용자가 요구하는 다양한 종류의 서비스에 맞는 맞춤형 자원을 제공하기 어렵다.

멀티 클라우드 환경에서는 단일 클라우드 서비스 환경과 달리 클라우드 서비스를 제공하는 주체가 다수 존재하며, 클라우드 서비스 사용자는 자신이 요청한 서비스 내용에 따라 적절한 클라우드 서비스 제공자를 직접 선택할 수 있다. 뿐만 아니라 특정한 클라우드 서비스 제공자의 자원이 부족하거나 혹은 남을 경우 다른 클라우드 서비스 제공자로부터 자원을 빌려오거나 혹은 빌려줌으로써 클라우드 자원 사용의 효율성을 높이고 서비스 사용자에게는

높은 신뢰성을 제공할 수 있다.

본 논문에서는 멀티 클라우드 환경에서 각 클라우드 서비스 제공자가 구축한 이종의 클라우드 플랫폼을 연동하여 사용자에게 단일화 된 클라우드 서비스를 제공할 수 있는 이종 클라우드 연동 브로커를 제안한다. 본 브로커는 RESTful 웹 서비스를 기반으로 이종의 클라우드 플랫폼과 클라우드 접속 어댑터 모듈간의 통신을 수행한다. 또한 본 브로커에서는 접속하는 사용자의 요구사항 패턴 정보를 기록하여 이를 기반으로 효율적인 가상 자원 프로비저닝을 달성할 수 있도록 한다.

2. 이종 클라우드 플랫폼 연동 브로커

그림 1에서는 이종의 클라우드 플랫폼을 연동하는 클라우드 브로커의 구조를 보인다. OpenStack[2], Eucalyptus[3] 및 CloudStack[4]이 클라우드 브로커에 연동되는 이종 클라우드 플랫폼으로 정의되었다. 먼저 User Layer에서는 다수의 클라우드 서비스 사용자(Client)들이 처리하고자 하는 Request를 요청하면, 클라우드 브로커의 Request Reader가 입력된 Request를 수신하고 이를 파싱하여 처리할 수 있는 오브젝트의 형태로 변환한다. Service Scheduler는 Multi Priority Queue를 가지고 있는데, 입력된 여러 Request의 우선순위를 고려하여 적절한 Queue에 저장시키고 이를 스케줄링 정책을 기반으로 처리하는 역할을 수행한다. Resource Manager는 연동된 클라우드 서비스 제공자의 자원을 관리한다.

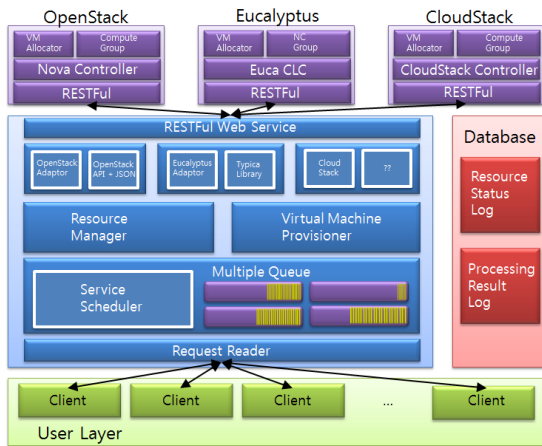


그림 1. 이중 클라우드 연동 브로커 구조

즉 연동된 클라우드 자원의 프로필 정보를 저장하고 있어서, 어떤 서비스 제공자가 사용자가 요청한 Request에 가장 알맞은 자원을 가지고 있는지 탐색하고 이를 매칭시키는 역할을 수행한다. Resource Manager에는 Resource Monitoring 모듈이 포함되는데, Resource Monitoring 모듈은 미리 정의된 일정한 주기를 기반으로 연동된 클라우드 서비스 제공자들의 자원 상태를 체크하고 이를 Database 모듈의 Resource Status Log에 저장하여 추후 클라우드 자원 선택 결정시 이를 반영한다.

Virtual Machine Provisioner는 사용자의 Request 요청 패턴을 미리 예측하고 예측된 정보를 바탕으로 클라우드 자원을 논리적인 Resource Pool에 미리 할당해놓는 기능을 제공한다. Request 요청 패턴 예측 정보는 과거에 사용자들이 클라우드 자원을 사용한 History 정보를 기반으로 도출될 수 있으며 다양한 기법을 사용할 수 있다.

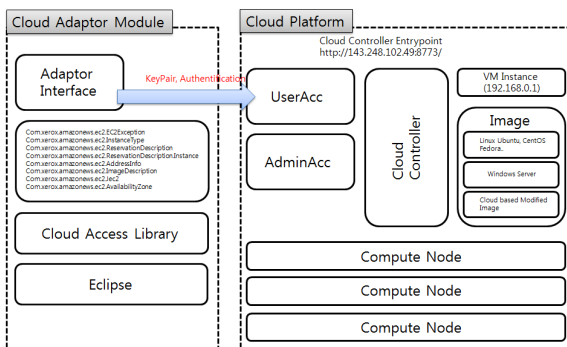


그림 2. 클라우드 어댑터 - 플랫폼 연동 구조

브로커에 연동된 이중의 클라우드 플랫폼에 접근 및 제어하기 위하여 각 플랫폼 어댑터 모듈이 포함된다. 어댑터 모듈에는 연동된 클라우드 플랫폼에 접속하고 자원을 요청하기 위한 라이브러리등이 포함된다. 클라우드 플랫폼과 어댑터는 기존의 SOAP 방식보다 더 간단하면서도 일관성을 제공하는 RESTful Web Service 기반으로 통신을 수행한다.

그림 2에서는 클라우드 브로커의 어댑터와 연동된 플랫폼과의 통신 구조를 보인다. 클라우드 어댑터는 자바 기반으로 Eclipse에 Cloud에 액세스할 수 있는 라이브러리를 포함하여 작성되며(예를 들어 Eucalyptus에는 Typica라는 라이브러리가 공개적으로 제공된다[5]) 이를 통해 개발된 여러 API를 통해 클라우드 플랫폼에 접근한다. 클라우드 플랫폼에는 다수의 계정이 존재하며, 클라우드 어댑터는 미리 가지고 있는 Key 정보를 가지고 인증을 받아서 접속한다. Cloud Controller는 접속된 계정이 요청하는 정보를 기반으로 VM instance를 제공하고 작업을 처리할 수 있도록 한다.

3. 결론

본 논문에서는 이중의 클라우드 플랫폼이 존재하는 환경에서 클라우드 서비스 사용자에게 가상적으로 단일화된 클라우드 자원을 제공하고 요구사항을 고려하여 최적의 자원을 선택할 수 있는 클라우드 연동 브로커 구조를 제안하였다. 제안된 클라우드 연동 브로커는 연결된 클라우드 플랫폼의 접근을 위한 클라우드 어댑터 모듈을 통해, 각 클라우드 서비스 제공자의 서비스 제공수준을 인지하여 사용자에게 적절한 자원을 제공한다. 또한 사용자가 요청하는 서비스 패턴 정보를 저장하여 이를 기반으로 효율적인 가상 자원 프로비저닝을 달성할 수 있도록 한다.

Acknowledgment

이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단-클라우드 Collaboration 기술 사업과 BK21 사업의 지원을 받아 수행된 연구임(No. 2012-0020522)

참고문헌

- [1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, M. Zaharia, Above the Clouds: A Berkeley View of Cloud Computing, Tech. Rep. UCB/EECS-2009-28, EECS Department, University of California, Berkeley (February 2009).
- [2] OpenStack Foundation, <http://www.openstack.org/>, 2012.
- [3] Eucalyptus Website, <http://www.eucalyptus.com/>, 2012
- [4] CloudStack Website, <http://www.cloudstack.org/>, 2012
- [5] A Java client library for Amazon Web Services, <http://code.google.com/p/typica/>, 2012
- [6] D. Kang, B. Kim, S. Kim and C. Youn, "Architecture of Multi Cloud Resource Broker in Cloud Federation", in Proceedings of KIPS YSEC 2012, April 2012.