

오픈스택 환경에서 효율적인 자원 관리를 위한 2 노드 기반

토폴로지 연구

강효성⁰ 홍충선
 경희대학교 컴퓨터공학과
 {kanghs, cshong}@khu.ac.kr

Topology Research Based on 2 Nodes for Effective Resource Management in Openstack Environment

HyoSung Kang⁰ ChooSeon Hong
 Department of Computer Engineering, Kyung Hee University

요약

오픈스택은 2010년 처음 개발된 이후 지금까지 많은 개인과 기업이 참여한 클라우드 구축 오픈소스 프로젝트이다. 이를 활용한 여러 클라우드 서비스가 개발되어 상용화되고 있지만 설치를 위한 하드웨어적인 요구사항이 많고 그 과정이 복잡하다는 단점이 존재한다. 이로 인해 시스템 에러 발생시 문제해결을 위한 시간적 소모가 크다는 문제점도 동반한다. 본 논문에서는 이런 문제점들을 개선하기 위해서 기존 토폴로지 구성을 좀 더 보완한 새로운 토폴로지 구성방법을 제안한다.

1. 서론

오픈스택(Openstack)은 2010년 Rackspace와 미국항공우주국(NASA)에 의해 개발된 개인 혹은 공용 클라우드 구축을 위한 오픈소스 커뮤니티이다[1]. 2012년 Foundation으로 독립한 후, Intel, IBM, Dell, Cisco, HP, NTT, SUSE, VMware 등 대형 기업들이 참여하였으며 현재 세계 100여국, 850여 조직에서 8000여 멤버가 오픈스택 프로젝트에 참여하고 있다[2]. 최근 2014년 4월, 9번째 릴리즈인 아이스하우스(Icehouse)가 대중에게 공개되, 오브젝트 스토리지 탐색기능과 클라우드의 리소스 현황을 한눈에 파악할 수 있는 셀리오미터(Ceilometer) 기능이 포함되어 사용자 인터페이스 기능이 대폭 개선된다[3].

오픈스택이 이처럼 주목 받는 이유는 역시 오픈소스라는데 있다. 많은 기업들이 여기에 참여하고 있지만 아마존, 애플과 같은 기업들의 기술 독점, 비공개로 인하여 클라우드 컴퓨팅 시장에서 이익을 내하고자 하는 다른 기업들에게는 대안이 필요했다. 오픈소스 프로젝트인 오픈스택이 등장하면서 기업들은 이를 차용해 자신들의 목표에 맞게 다양한 서비스 개발로 활용하게 된다.

이처럼 오픈스택은 이제 미래 클라우드 컴퓨팅 기술의 한 분야를 차지하고 있지만 이를 구성하기 위한 하드웨어 장비의 요구사항이 많고 토폴로지를 구성하기 위한 설치방법이 복잡하다는 단점이 존재한다.

본 논문에서는 오픈스택 공식 사이트에서 제안하는 표준 토폴로지 구성안을 소개하고 여기서 가지는 문제점과 이를 개선하기 위한 효과적인 토폴로지 방법을 제안하고자 한다.

2. 관련연구

2.1 3개 노드 기반의 표준 토폴로지

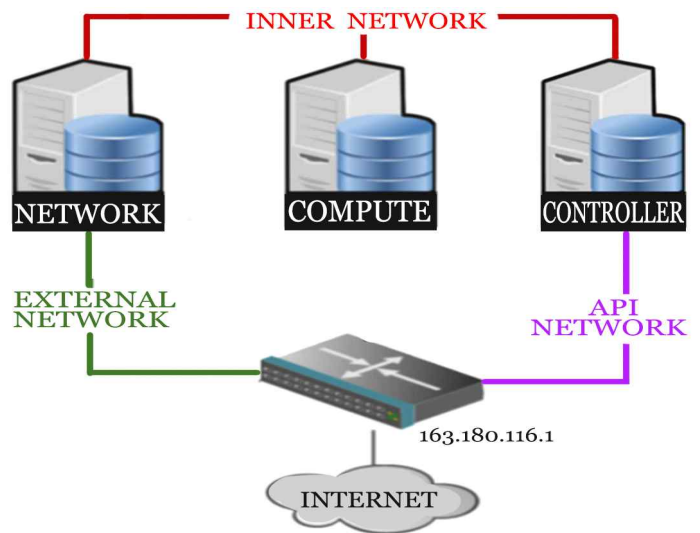


그림 1. 3개 노드로 구성된 오픈스택 표준 토폴로지

그림 1은 오픈스택 공식 가이드에서 제안하는 3개의 노드로 구성된 오픈스택 표준 토폴로지이다[4]. 네트워크, 컴퓨트, 컨트롤러 이렇게 3개의 노드가 각 노드별로 하나의 서비스를 제공하는 방법이다. 여기에 컴퓨트노드를

본 연구는 미래창조과학부 및 한국정보화진흥원의 미래네트워크연구 시험망 구축운영 사업의 일환으로 수행하였음. [2014-Z349, 개방형 모바일 네트워크 플랫폼 국제확산 연구 및 실증시험] *Dr. CS Hong is the corresponding author

추가하면서 클라우드 자원의 추가 확보가 가능하여 멀티노드 방식이라고도 한다. 네트워크노드는 오픈스택 인스턴스의 네트워크서비스를 제공하는 노드이다. 자원 공유를 위해 노드 간 연결된 내부망과 각 인스턴스를 인터넷에 연결해 주는 외부망 구성을 위해 2개의 NIC가 필요하다. 컨트롤러노드 또한 내부망과 웹 UI를 통해 사용자에게 클라우드 제어환경을 제공해주는 외부망 연결을 위해 2개의 NIC가 필요하다. 컴퓨터노드는 오픈스택에서 생성한 다수의 가상 인스턴스를 동시에 실행하기 위한 논리적 서비스만을 제공하기 때문에 노드 간 내부망 연결을 위한 NIC 1개가 필요하다.

노드명	컨트롤러	네트워크	컴퓨터
프로세스	1	1	1
메모리	2 GB	512 MB	2 GB
스토리지	5 GB	5 GB	10 GB
HDD	1	1	2
NIC	2	2	1

표 1. 표준 토폴로지 구축을 위한 노드별 하드웨어 정보

이 외에도 표준 토폴로지 구축을 위해서는 각 노드별로 서로 다른 요구사항이 존재하는데 표 1은 각 노드별 최소 하드웨어 사양을 나타낸다[4].

표 1처럼 하드웨어의 사양이 다른 만큼 오픈스택 공식 가이드에서 제공하는 각 노드별 설치방법도 서로 다르다[4]. 이런 차이는 오픈스택 설치과정에서도 어려움을 주지만 설치 도중 에러 발생시 문제해결을 위한 시간적 낭비도 함께 가지게 된다.

3. 제안사항

3.1 2노드 기반의 확장형 토폴로지

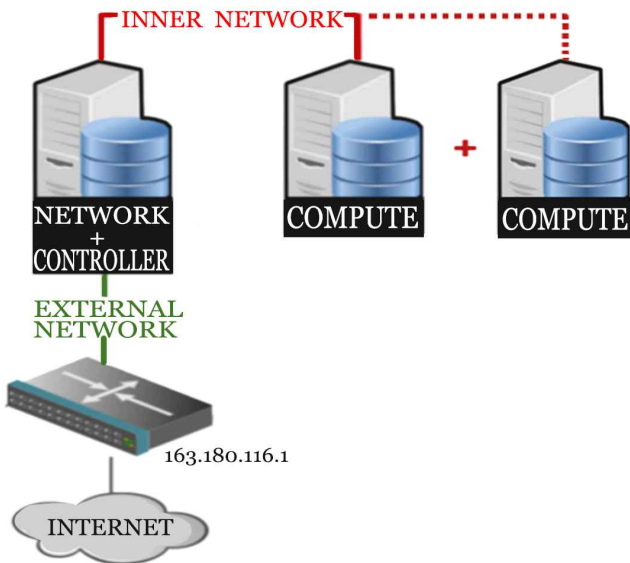


그림 2. 2개 노드 기반의 확장형 토폴로지

그림 2는 본 논문에서 제안하는 2개 노드 기반의 확장형 오픈스택 토폴로지이다. 컨트롤러와 네트워크를 하나의 노드로 구성하고 노드가 감소한 만큼 다수의 컴퓨터노드를 추가 확장함으로써 자원 관리를 보완한 형태의 토폴로지이다.

노드명	컨트롤러+네트워크	컴퓨터
프로세스	1	1
메모리	2 GB	2 GB
스토리지	10 GB	10 GB
HDD	1	1
NIC	1	1

표 2. 제안된 토폴로지의 노드별 하드웨어 정보

표 2는 본 논문에서 제안한 토폴로지의 노드별 최소하드웨어 요구사항 정보이다. 표 1의 표준 토폴로지와 비교해서 제안된 토폴로지에서는 인터넷을 통한 외부망 연결만 필요하므로 최소 각 노드별 하나의 NIC와 HDD만으로도 충분히 구성이 가능하며, 하드웨어 요구사항이 감소한 만큼 설치방법도 간소화 할 수 있다. 또한 노드의 감소로 인해 생긴 리소스 제한이라는 문제점은 오히려 구성하려는 클라우드 서비스의 규모에 따라 컴퓨터노드를 추가 확장함으로써 능동적인 시스템 구성으로 가능하다는 강점으로 변모한다.

4. 성능분석

성능은 오픈스택 공식가이드에서 제안한 표준토폴로지 방식과 본 논문에서 제안한 2개 노드 기반 토폴로지 방식으로 각 인스턴스를 생성하고, 인스턴스가 외부망으로 접속하기 위한 게이트웨이로 Ping을 통해 100개의 패킷을 보낸 후 손실률과 걸린 시간을 측정하여 분석하였다.

이때 표준토폴로지의 생성된 인스턴스 IP는 163.180.116.101, 제안된 토폴로지의 생성된 인스턴스 IP는 163.180.116.118이며, 둘 다 163.180.116.1의 공통된 게이트웨이를 가진다.

```

--- 163.180.116.1 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 98997ms
rtt min/avg/max/mdev = 0.400/0.538/2.387/0.279 ms
openstack@openstack:~$
163.180.116.101:22에 연결됨.
    
```

그림 3. 표준 토폴로지의 패킷 손실률과 측정시간

그림 3은 표준 토폴로지의 패킷 손실률과 측정시간을 나타낸 것이다. 총 100개의 패킷을 보내고 손실없이 총 98997 ms 시간이 걸렸다.

```

--- 163.180.116.1 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 98996ms
rtt min/avg/max/mdev = 0.401/0.516/1.687/0.164 ms
openstack@openstack:~$
163.180.116.118:22에 연결됨.
    
```

그림 4. 제안된 토폴로지의 패킷 손실률과 측정시간

그림 4는 본 논문에서 제안한 토폴로지의 패킷손실률과 측정시간이다. 총 100개의 패킷을 보내고 손실없이 98996 ms 측정시간이 걸렸다.

두 토폴로지의 성능 분석결과, 모두 손실률이 없었으며 측정시간도 크게 차이가 나지 않는다. 제안된 토폴로지를 통해서도 충분히 오픈스택 공식 가이드에서 제안하고 있는 표준 토폴로지의 성능을 보여준다.

5. 결론

본 논문에서 제안된 2노드 기반의 확장형 오픈스택 토폴로지는 오픈스택 표준형 토폴로지가 가지는 하드웨어적인 설치공정의 어려움을 간소화 하면서도 오히려 기존 토폴로지의 기능을 충분히 보완하고 있다. 하지만 네트워크 노드의 역할을 축소함으로써 하나의 외부망을 통해 전체 생성한 가상장치의 네트워크를 보완하다보니 과부하로 인하여 가상장치의 속도 저하라는 단점을 안고 있다. 따라서 향후 이런 한계를 개선한다면 표준 토폴로지보다 더 효율적인 오픈스택 구성이 가능하다고 전망한다.

6. 참고문헌

[1] 황진경, 안재석, “오픈스택 클라우드 기술 동향”, OSIA Standards & Technology Review, 제26권, 제1호, pp.86-100, 2013. 3.

[2] Openstack[Online]. Available : <http://www.openstack.org>.

[3] 김병식, 이범철 “오픈스택을 이용한 클라우드 서비스 플랫폼 구축 및 활용”, 2014 동계 한국통신학회 종합 학술 발표회 논문집, pp.669-670, 2014. 1.

[4] Openstack Guide[Online]. Available : <http://docs.openstack.org>.