

분산 컨트롤러 클러스터 환경에서의 ODL 클러스터 고가용성 성능평가

정우석, 구영훈, 심규석, 김명섭
고려대학교

{hary5832, gyh0808, kusuk007, tmskim}@korea.ac.kr

Performance Evaluation of ODL High Availability on the Distributed Controller Cluster Environment

Woo-Suk Jung, Young-Hoon Goo, Kyu-Seok Shim, Myung-Sup Kim
Korea Univ.

요약

최근에 소프트웨어 정의 네트워킹(SDN)에 대한 관심이 높아지고 다양한 오픈소스 컨트롤러들이 제시되고 있다. 컨트롤러의 중요한 성능 지표 중 하나는 클러스터를 구성하는 컨트롤러들 중 한대의 컨트롤러에서 장애가 발생할 경우, 다른 컨트롤러가 그 업무를 대신 수행할 수 있는 고가용성(High Availability)이다. 본 논문에서는 최근 가장 많은 조명을 받고 있는 오픈소스 컨트롤러인 OpenDaylight (ODL) 컨트롤러가 다양한 장애상황에서 장애를 극복하는지를 확인하고, 장애 극복까지 소요된 시간을 측정하여 클러스터 고가용성 성능을 테스트하였다. 실험을 통해 ODL 컨트롤러는 3-Node 클러스터 환경에서 하나의 컨트롤러가 다운되는 모든 장애 상황에 대하여 복구가 이루어 지는 높은 고가용성을 보장하며, Shard의 수가 증가하고 Leader가 분산됨에 따라 서비스 지연시간이 늘어남을 알 수 있었다.

I. 서론

현존하는 프로토콜 중심의 네트워킹이 가지는 근본적인 한계를 인식하고 이를 소프트웨어 정의 네트워킹(Software Defined Networking: SDN)이라는 새로운 흐름으로 해소하자는 노력이 지난 몇 년 동안에 매우 중요한 연구 추세이다. [1,2]

이에 따라 최근 소프트웨어 정의 네트워킹(SDN)이 많은 사람들의 관심을 끌고 있으며, 다양한 데이터 센터들이 자신들의 네트워크에 SDN을 사용하기 시작하고 있다. 현재 SDN의 구현을 위해 가장 많이 사용되는 프로토콜은 오픈플로우로써 이는 SDN에 해당되는 통신 프로토콜이며 컨트롤러가 data-plane을 처리하고 플로우를 스위치 및 라우터에게 전송하는 control-plane 역할을 맡는다. [3]

최근 많은 조명을 받고 있는 OpenDaylight(ODL)는 중앙 집중적인 네트워크 제어 및 관리 서비스를 제공하는 프레임워크를 구현하기 위해 리눅스 재단에서 후원하는 SDN 컨트롤러 개발 프로젝트이다. 현재 ODL 프로젝트에는 Cisco, Juniper, Ericsson, VMware와 같은 글로벌 IT 업체들이 개발에 참여하고 있으며, 릴리즈 된 최신 버전은 Lithium 버전이다. OpenFlow만을 지원하는 기존의 SDN 컨트롤러들과 달리, ODL의 경우 BGP, SNMP, LSIP, NETCONF 등의 다양한 SBI(South-Bound Interface)를 지원한다는 장점이 있다.

이러한 SDN 환경하에서 클러스터로 묶인 컨트롤러들 중 한대의 컨트롤러에서 장애가 발생할 경우, 다른 컨트롤러가 즉시 그 업무를 대신 수행하여 시스템 장애를 복구 하는 고가용성은 매우 중요한 성능 지표 중 하나이다.

본 논문에서는 분산 컨트롤러 클러스터 환경에서의 ODL 클러스터 고가용성 성능평가를 위하여 다양한 장애상황을 설정하고, 이를 ODL 컨트롤러가 극복하는지 여부와 각 장애 상황 별 장애 극복 시간을 측정하였다.

2 장에서는 실험 환경 및 ODL 컨트롤러의 구조 및 동작원리에 대하여 설명하고, 3 장에서는 실험 방법과 결과 그리고 4 장에서는 결론을 설명한다.

II. ODL 클러스터 고가용성 성능평가

분산 컨트롤러 클러스터 환경에서의 ODL 클러스터 고가용성 성능평가를 위해 여러 가지 클러스터 장애 상황을 설정하고 North Bound에서 CRUD(Create, Read, Update, Delete)명령을 주기적인 요청을 수행하면서, 각 상황 별 장애 극복 여부 및 서비스 지연시간을 측정하였다.

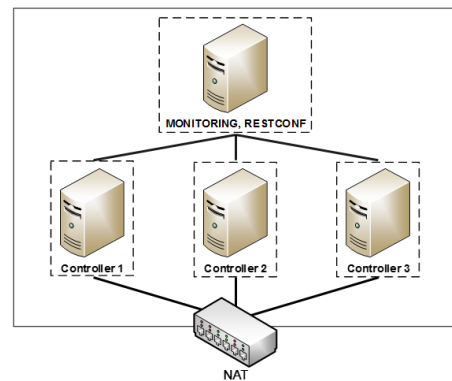


Figure 1. Experiment environment

Figure 1은 실험 환경을 도식화 한 것으로, 하나의 컴퓨터에 VM을 통하여 3-Node 클러스터 환경을 구현하였다.

성능평가 대상인 ODL은 계층적인 구조를 가진다. 가장 상위 계층인 Network Applications Orchestrations & Services 계층은 상위 레벨의 네트워크 애플리케이션으로 구성되어 ODL이 제공하는

본 논문은 BK21 플러스 사업(No. T1300573) 및 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원(B0101-15-233, 스마트 네트워킹핵심기술개발)을 받아 수행된 연구임

NBI(North-Bound Interface)인 REST API 를 통해 ODL 과 통신하게 된다. 두 번째 계층인 ODL 컨트롤러 플랫폼 계층은 모듈화와 확장성을 위해 OSGI(Open Service Gateway Interface) 프레임워크를 기반으로 개발되었다. 또한 ODL 컨트롤러 플랫폼 계층 내에 SAL(Service Abstraction Layer) 계층을 두어 컨트롤러가 제공하는 기능들에 대한 추상화 기능을 제공한다.

ODL 클러스터에서 컨트롤러는 Shard 라는 단위의 데이터 스토어에 데이터를 저장한다. 또한 고가용성을 보장하기 위해 각 컨트롤러의 Shard 데이터를 Raft algorithm 과 Gossip protocol 을 통해 동기화한다. Raft algorithm 은 Shard log 복제와 Shard Leader 선출에 사용되며, Gossip protocol 은 ODL 에서 Remote RPC 의 Route registry 복제를 위해 사용된다. 3-Node 환경에서 하나의 Leader 와 2 개의 Follower 가 존재하며, Leader 는 Shard 복제를 주도한다.

실험을 위하여 총 5 가지의 장애 상황을 설정 하였다. 장애상황은 Table 1 과 같이 하나의 Shard 로 이루어진 Single Shard 환경과 여러 개의 Shard 로 이루어진 Multi Shard 환경인 경우로 나누고, Multi Shard 인 경우에는 하나의 컨트롤러가 모든 Shard 의 Leader 인 Single Leader 환경과 Shard 의 Leader 가 각 컨트롤러들에 분산된 Distributed Leader 인 경우로 구분하여 각 상황에 대하여 Leader, Follower 또는 하나의 컨트롤러가 Down 된 경우로 설정하였다.

| | Shard Condition | Controller Condition |
|-------------|----------------------------------|----------------------|
| Test Case 1 | Single Shard | Leader Down |
| Test Case 2 | | Follower Down |
| Test Case 3 | Multi Shard / Single Leader | Leader Down |
| Test Case 4 | | Follower Down |
| Test Case 5 | Multi Shard / Distributed Leader | One Controller Down |

Table 1. Fault condition table

III. 실험 결과

ODL 컨트롤러에서는 Default Shard 설정은 default, topology, inventory, toaster, car 그리고 people 의 6 개 Shard 로 구성된 Multi Shard 환경이다. Single Shard 환경을 구성하기 위해 module-shard.conf 파일과 modules.conf 파일을 수정하여 실험하였다.

Test Case 1 의 경우 Multi shard, Single leader 환경을 설정한 후 기존의 Leader 을 Down 시킨다. 이후 새로 선출된 Leader 에서 데이터를 변경하고, 데이터를 변경한 후 기존 Leader 를 다시 Up 시킨다. 새로운 Leader 에서 변경된 데이터가 기존 Leader 에 동기화 되는지 확인한 후, Down 상태에서 다시 Up 된 기존 Leader 에서 데이터를 변경한다. 변경된 데이터가 새로운 Leader 에 동기화 되는지 확인한다. 마지막으로 동기화 되기까지의 시간을 측정한다. 나머지 실험도 이와 같은 방식으로 실험한다.

Table 2 는 실험 결과를 정리한 것이다. Sync 는 Down 되었던 컨트롤러가 다시 Up 되고, 동기화 되기까지 측정된 시간, Veri 는 동기화가 이루어진 후

Verification 이 완료 되기까지 측정된 시간을 의미하고, Total 은 Sync 와 Veri 과정을 합한 시간을 의미한다. 실험 결과를 통해 ODL 컨트롤러는 3-Node 클러스터 환경에서 하나의 컨트롤러가 Down 되는 모든 장애 상황에 대하여 높은 고가용성을 보장함을 알 수 있었다. 또한 Single Shard 환경에서 Multi Shard 환경보다 짧은 서비스 지연시간은 보였으며, Single Leader 환경이 Distributed Leader 환경보다 짧은 지연시간을 보임을 알 수 있었다.

ODL 컨트롤러를 운용함에 있어 본 논문의 고가용성 성능평가 결과를 반영한다면, Shard 의 개수와 Leader 의 분산여부 결정을 통해 목적에 따른 최적의 ODL 클러스터 환경을 설정할 수 있다.

| | Single Shard | | Multi Shard / Single Leader | | Multi-Shard / Distributed Leader |
|------------|--------------|---------------|-----------------------------|---------------|----------------------------------|
| | Leader Down | Follower Down | Leader Down | Follower Down | One Controller Down |
| Sync (ms) | 198 | 256 | 226 | 232 | 262 |
| Veri (ms) | 537 | 516 | 559 | 550 | 553 |
| Total (ms) | 735 | 772 | 785 | 782 | 815 |

Table 2. Delay time table of each fault condition

IV. 결론

본 논문에서는 오픈소스 SDN 컨트롤러 중 하나인 ODL 컨트롤러의 클러스터 고가용성 성능평가를 위해 다양한 장애상황을 설정하고 이를 극복하는 시간을 측정하였다.

실험을 통하여 ODL 컨트롤러는 3-Node 분산 컨트롤러 클러스터 환경에서 하나의 컨트롤러가 장애를 발생 시키는 모든 상황에 대하여 서비스 복구가 이루어 지는 높은 고가용성을 보장함을 알 수 있었다. 또한, Single Shard 에서 Multi Shard 로 Shard 가 늘어나고 Single Leader 에서 Distributed Leader 로 Leader 가 분산됨에 따라 서비스 지연시간이 늘어남을 알 수 있었다.

참고 문헌

- [1] 김종원. "미래형 서비스 실증을 위한 오픈플로우 기반 SDN 시험환경 구축." 한국통신학회지 (정보와통신) 30.3 (2013): 43-50.
- [2] Mayoral, A., et al. "Experimental validation of automatic lightpath establishment integrating OpenDayLight SDN controller and Active Stateful PCE within the ADRENALINE testbed." Transparent Optical Networks (ICTON), 2014 16th International Conference on. IEEE, 2014.
- [3] McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J. OpenFlow: Enabling Innovation in Campus Networks. ACM SIGCOMM Comput. Commun. Rev. 2008, 38, 69- 74.