Analysis of SDN Controller Cluster in Large-scale Production Networks
draft-zcao-sdnrg-controller-00

Abstract

Software Defined Networking necessitates an efficient, scalable, secure Controller Cluster. This document analyzes the problems of implementing and deploying a successful controller system and figures out the requirements therein. This document also proposes a set of benchmarks for controller performance evaluation.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 05, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. Introduction

Software Defined Networking is now a term that different players may express different opinions. Yet despite all that, the separate of data and control via the use of controller and generic forwarding devices is the key component of this architecture. In this architecture, as depicted in Figure 1, the controller system exposes APIs to the upper layer SDN applications, while communicating with the data plane devices using protocols like Openflow and OVSDB. The data plane devices in this architecture include both physical devices and virtual devices that connect virtual machines.

```
++----++ ++----++ ++----++
|    |   |    |   |
++----++ ++----++ ++----++
             |
+---------------------+  Controller System
             |
++----++ ++----++ ++----++
|    |   |    |   |
++----++ ++----++ ++----++
```

Figure 1: SDN General Architecture
2. Terminology

Several Terms are defined in this document.

SDN Controller Cluster (SCC): an integrated system that manages a large scale production network and handles control plane communications with a big number of data plane devices. SCC normally consists of a cluster of controllers which exposes outside as a logically integrated entity.

Controller Instance (CoIN): an instance of a controller, multiple CoINs consists the SCC.

South-bound Interface: the interface that implements a certain type of protocols for communication with the data plane devices.

North-bound interface: the interface exposed to the upper layer applications, normally APIs to abstract the functions supported by the SCC.

West-East Interface: the interface between different CoINs. The West-East Interface is considered as internal, without need of exposing to the outside.

3. SDN Controller Cluster Analysis

This section reviews and analyzes the SDN Controller Cluster from two aspects including protocols, functionalities.

3.1. Protocol Aspect

We will take Openflow as an example of the south-bound interface in the following description if no additional notes apply.

The OpenFlow protocol supports three message types, controller-to-switch, asynchronous, and symmetric.

Controller-to-switch messages are initiated by the controller and used to directly manage or inspect the state of the switch.

Asynchronous messages are initiated by the switch and used to update the controller of network events and changes to the switch state.

Symmetric messages are initiated by either the switch or the controller and sent without solicitation.
Controller-to-switch messages are generally trigger by the upper layer SDN applications to manage and acquire the state of the switch. A scalable SDN Controller Cluster MUST support a reasonable number of connections from SDN applications. And if necessary, the SCC SHOULD check the conflicts therein.

SCC is the responder of synchronous messages. Considering the number of data plane devices in the large-scale production networks, the number of connections the SCC could support is a big impactor to the real life performance. And when the switch does not find a match in its local flow table for a packet, the packet is forwarded to the SCC as an attachment, which will add to more processing pressure.

Symmetric messages include HELLO, ECHO and Experimental extensions. These messages do not incur much state keeping and handling logics, so do not put much pressure on the SCC.

In addition to the above analysis, the south bound interface normally requires a pre-established secure connections (e.g., Openflow uses SSL), each of which consumes computing resources on the SCC.

3.2. Functionality Aspect

From the functionalities aspects, the SCC SHOULD at least supports the following items.

1. North-bound API. The abstraction of the services that the SCC can provide to upper layer applications.

2. Route Computing. The essential function that the SCC provides to compute routes so that the results can be written to the data plane devices via the south-bound interface.

3. Consistent Data Store. The required data storage to assist SCC functioning correctly. Consistency is a key requirement so that multiple requesters get the consistent results possibly processed by different controller instances (CoIN). Should expose necessary APIs for queries and retrievals.


5. West-East Communication. The internal communication between different CoINs. Distributed system techniques may apply here.

6. Structured Logging, naming logging services. Logging data SHOULD be stored distributedly.
7. Security module, to avoid malicious flow requests to flood the controller.

8. South-bound Adaption. Adaption layer to support multiple south-bound protocols such as Openflow, ForCES, OVSDB, etc.

From the above analysis, the SCC functional image can be depicted as below.

```
+------------------------------------------------------------------+
|                          North-bound API                          |
+------------------------------------------------------------------+

//                                \
+----------------------------------+
| Route | Data | West| ======+ |West-  | Data | Rout |
|Compute| Store| east| +      + |east   | Store| Comp |
+-------+   +-----+   +----+ +      + +-------+   +-----+   +----+ +

+-------+   +----------+     +      + +-------+   +----------+     +
| Topo  |___|Structured|     +      + | Topo  |___|Structured|     +
|Manager|   | Logging   |     +      + |Manager|   | Logging   |     +
+-------+   +----------+     +      + +-------+   +----------+     +
+       +       +       +       +       +       +       +
| Security | +       +       | Security | +
+       +       +       +       +       +       +       +
+----------------------------------+
| South-bound Adaption             |
|                                  |
+----------------------------------+
```

Figure 2: SDN Controller Cluster

4. SDN Controller Requirements

Based on the analysis in previous sections, we figure out the requirements for SDN controller cluster as below.

- Availability. Service availability.
- Scalability. Scalable to the network scale.
- Reliability. Reliable to work load and application requests.
- Consistency. Consistent behavior.
5. Benchmarks and Open Questions

This section lists the open questions for future studies of SDN controller in large scale production networks.

We try to list the benchmarks used for a controller system.

1. Availability benchmarks.
   a. flow requests per second
   b. API calls per second
   c. topology discovery delay, could be depicted using second per port or second per thousand flows.
   d. number of concurrent connections between the controller and devices.

2. Scalability benchmarks, this particularly investigates the relationship between two or more metrics
   a. API call latency vs. number of API calls per second
   b. topology discovery delay vs. network scale (flow size or device number, or concurrent connections )
   c. flow request delay vs. network scale
   d. route computing delay vs. network scale

3. Reliability benchmarks.
   a. Link and switch failures
   b. COINs failures.

   a. probability of inconsistent query results
   b. latency of updating the whole COINs DHT
6. Related Work

Many studies are around the problem of a scalable and reliable controller platform. NOX [NOX] is a simple openflow controller and it is single threaded. Materoo [Maestro] achieves a simple programming model for control function development by having a single-threaded event-loop and it was demonstrated that Materoo can achieve linear scalability. Onix [ONIX] is a platform on top of which a network control plane can be implemented as a distributed system. Onix adopts DHT (Distributed Hast Table) techniques for its extensibility. The ‘Controller Placement Problem’ was also proposed for large production networks [Placement].

7. IANA Considerations

This document does not have any IANA requests.

8. Security Considerations

The SDN security requirements are specified in [I-D.hartman-sdnsec-requirements], and the security analysis of Openflow is presented in [I-D.mrw-sdnsec-openflow-analysis].

9. References

9.1. Normative References


9.2. Informative References

[I-D.hartman-sdnsec-requirements]

[I-D.mrw-sdnsec-openflow-analysis]

[I-D.sin-sdnrg-sdn-approach]


Authors’ Addresses

Zhen Cao  
China Mobile  
Xuanwumenxi Ave. No.32  
Beijing, Xicheng District  100053  
China  
Email: zehn.cao@gmail.com, caozhen@chinamobile.com

Chen Li  
China Mobile  
Xuanwumenxi Ave. No.32  
Beijing, Xicheng District  100053  
China  
Email: lichenyj@chinamobile.com