Open Network Operating System

SDN Control Plane Performance

An ONOS™ PROJECT WHITE PAPER

JULY 2017

Open Networking Foundation

Table of Contents

List of Figures

1 Introduction

The Open Network Operating System (ONOS™) is an open source project hosted at ONF and is part of the Linux Foundation. ONOS is a Software Defined Networking (SDN) OS for service providers that has scalability, high availability, high performance and a rich set of network control and configuration abstractions, which make it easy to create apps and services. The platform is based on a robust distributed system architecture and has quickly matured to be both feature rich and production ready. The community has grown to include 50+ partners and collaborators who contribute to all aspects of the project including interesting use cases such as CORD™ (Central Office Re-architected as a Datacenter).

The performance, scalability and high availability offered by ONOS are features that are rapidly becoming important with the emergence of new classes of use cases and applications including 5G mobile, Internet of Things, Big Data analytics, mobile health etc., in addition to the release of SDN capable devices like the latest switch based on Barefoot Networks' Tofino silicon.

Past studies of SDN controllers that have included ONOS performance numbers have only provided a narrow view of the controller's performance and have omitted evaluation of the highly available controller cluster configuration. Furthermore, the methodologies used were subject to environmental bottlenecks and consequently failed to adequately evaluate the platform's true capabilities.

The objective of this white paper is to compare and contrast the performance and scalability of ONOS project's latest release (Kingfisher) with that of the previous release (Blackbird). Results prove that Kingfisher release has maintained or improved its perfomance and scalability over the Blackbird release while adding many new features and capabilities. We will also cover the methodologies to evaluate ONOS and explain why multiple test perspectives need to be considered.

Finally, we explain the network edge use case and its need for the level of performance, scalability and high availability offered by ONOS.

More information on the ONOS project and its community can be found on the ONOS wiki page.

2 Background

ONOS launched its first open source release, called Avocet, in December 2014 and the community has been issuing quarterly releases ever since. With its second release, called Blackbird, the project published a comprehensive white paper on SDN control plane performance, identifying the key performance metrics and a thorough performance evaluation of ONOS against these metrics [1][2]. This performance study demonstrated the performance, scalability and high-availability characteristics of the ONOS platform's key control features. For example, it showed that ONOS can execute several million flow operations per second, sub-10ms latency for control operations and its throughput performance scales with the number of ONOS instances while keeping the latency virtually constant. In many respects, this is the best one can expect from an SDN controller.

ONOS has had 9 releases since the Blackbird release in early 2015. Over this time, it has gained many new features and functions in its distributed core, northbound and southbound interfaces and has added support for numerous applications, southbound protocols and devices. At the same time, with each release, its performance and scalability have continued to get better. For example, the latest Kingfisher release shows that ONOS can react to a switch failure in less than 5ms compared to 14ms in the case of Blackbird release and the intent operations performance has increased from 150K/sec in Blackbird to 225K/sec in Kingfisher while maintaining 3 million flow operations per second.

3 ONOS Performance

3.1 Key Performance Metrics for a SDN Controller

In carrier-scale networks, the SDN control plane has to support high throughput to meet carrier requirements. Throughput of the SDN control plane can be evaluated by measuring its "throughput" under load at the northbound and southbound interfaces. Also, failures are a fact of life and the SDN control plane needs to be able to swiftly detect and react to these failures to ensure seamless operation. Fault recovery time in reaction to network element failure can be approximated by measuring the "latency" of the SDN control plane in detecting and/or reacting to such events. This is why we choose topology change latency, flow/intent operation throughput as key metrics for ONOS performance.

Multi-node controllers are also essential to enable scaling to accommodate network growth as well as to provide fault tolerance in the event of controller node failure. For these reasons, we expect operators to deploy the majority of controllers in multi-node configurations and hence it's not enough to test performance of a single-node controller configuration. A general theme of all our evaluations is to make measurements on ONOS as it scales from a single node to a cluster of three, five and seven nodes.

3.2 ONOS Performance and Scalability Evolution

The ONOS team published a comprehensive study of its controller platform with ONOS Blackbird release performance white paper two years ago, which establishes a number of important performance metrics, both in the terms of throughput and latency of operations and it does so while considering the critical dimensions of high-availability and scalability.

In every release, ONOS is tested for the metrics established in the Blackbird release white paper, and we continue to meet and exceed the Blackbird release results, which remain the best reported results in the industry.

Graphs below demonstrate few key performance numbers with both ONOS Blackbird release and Kingfisher release, including response time to switch/link failure and throughput of flow and intent operations. As shown, when compared with the Blackbird release, performance of the ONOS Kingfisher release improved in terms of reduced failure response time and increased flow/intent operations throughput.

These numbers are based on an ONOS cluster of seven nodes. Our evaluations also show that as the number of cluster instances increases, the throughput numbers scale in a linear fashion while latency performance remains unchanged.

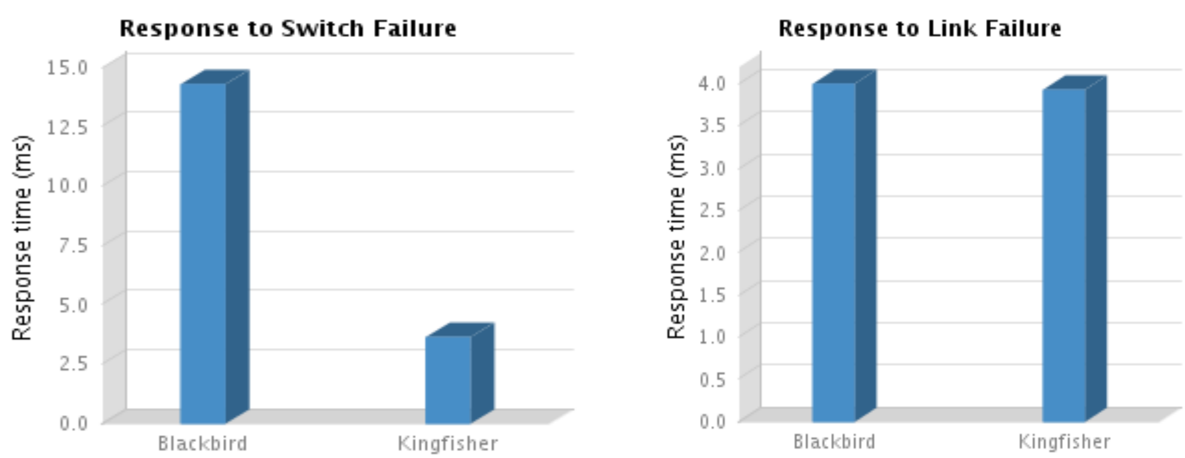

Figure 3.1: Comparison of Response Times to Switch & Link Failures

Note: Response time remains unchanged when ONOS cluster size grows and Kingfisher has improved/stable results compared to Blackbird

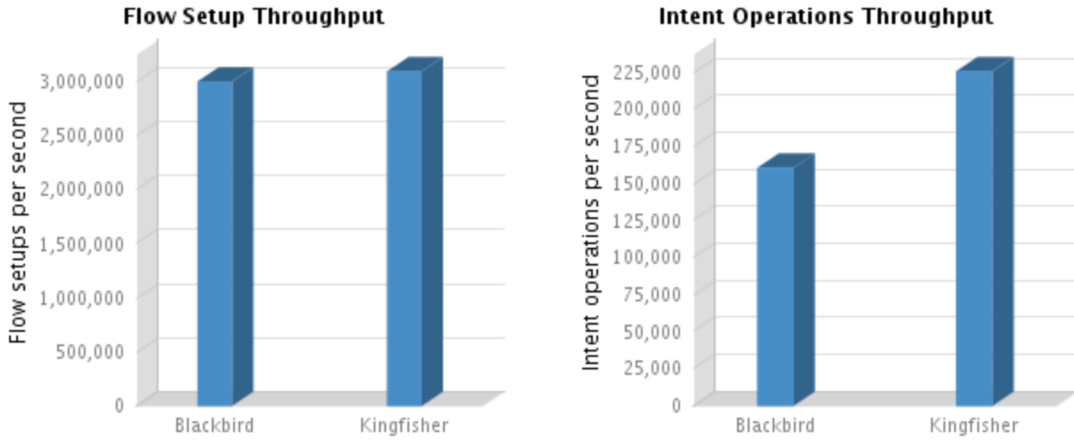

Figure 3.2: Comparison of Flow Setup and Intent Operations Throughputs

Note: Throughput scales in a linearly with ONOS cluster size and Kingfisher has equal or better performance on all metrics as compared to Blackbird.

ONOS has been continuously improving to support new features and applications to meet various requirements of the service providers.

- On the Northbound interface, ONOS continues to provide an excellent framework to enable DevOps and operations groups to specify what they need without worrying about how this will get instantiated on the underlying network. ONOS continues to evolve in terms of new features and functions, including applications and functions for L2 Forwarding, L3 Routing, Traffic Engineering, PCE, packet-optical, Bandwidth on Demand, Mobility, Enhanced GUI, and more.

- On the Southbound interface, ONOS continues to provide well-defined abstractions for discovery, configuration and programmability to support diversity of interfaces and devices. In addition to OpenFlow®, ONOS supports various protocols and devices. Southbound interfaces include OVSDB, NetConf, TL1, SNMP, PCEP, REST, BGP-LS, OSPF, and ISIS.

Even as ONOS has evolved and acquired significant new functionality, it has been able to deliver consistent performance and scalability from A through the K release. Because ONOS is built from the ground up as a scalable distributed system, performance has not been compromised with the addition of substantial functionality. To be specific, ONOS achieves this by taking into consideration the unique properties of each type of control plane state and maps it to a solution that provides the best semantics and performance for that state. This makes ONOS capable of seamlessly meeting demands for both high performance and high availability required for an SDN control plane even as new functionality and capability is added to the platform. More detailed information about

ONOS performance and scalability evaluations are available in ONOS Kingfisher white paper for technical audience [3].

4 Other SDN Controller Performance Studies

We are pleased to see other SDN controller performance reports publish their findings on the performance capabilities of ONOS [4]. The end-to-end test scenarios employed in these reports certainly offer an interesting insight into the overall performance of the system in the context of the existing network capabilities. However, just by themselves, such evaluation scenarios do not adequately highlight the performance capabilities, and more importantly, limitations of the controller platform.

Users need to know the capabilities and limitations of the controller and its behavior as a cluster grows in size. Thus, when measuring performance of SDN controllers, it is important that the controller be exposed to adequate levels of stress. Otherwise, its limitations may not be exposed until it encounters heavy loads in production.

For example, as demonstrated in the figure below, testing the controller performance through the REST API and against a small number of relatively slow switches does not sufficiently stress the controller. Testing against a single controller instance does not consider the effect that high-availability and scalability aspects have on performance. Consequently, the end-to-end tests, while informative, are by themselves insufficient to fully assess the controller platform.

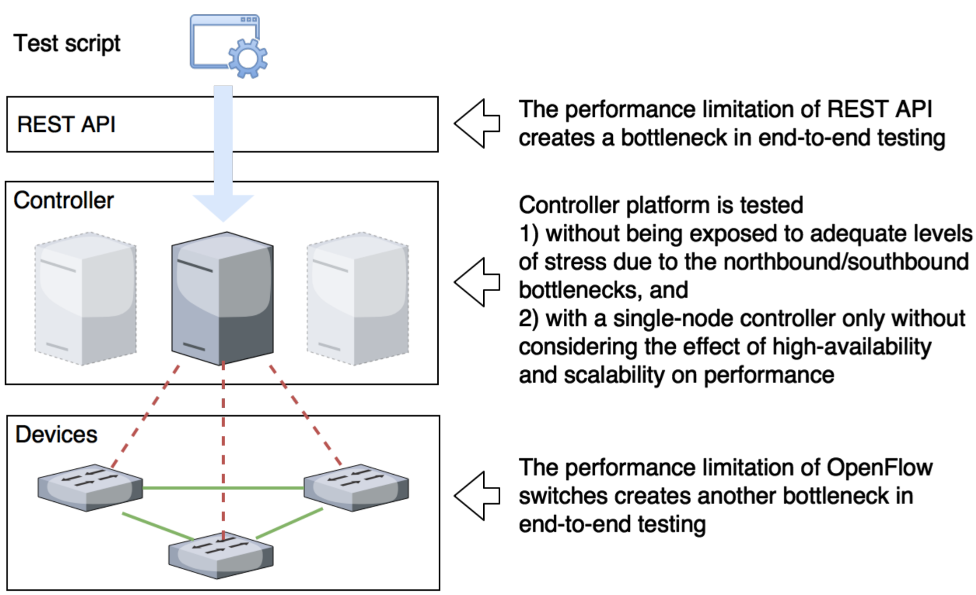

Figure 4.1: REST API Test Setup

Using the REST API for flow programming of several OpenFlow switches represents one aspect of usage. However, it should be noted that there are couple of issues in this approach that create performance bottlenecks outside the controller, thus potentially masking the actual internal performance capabilities of the controller:

- Use of REST APIs, which while certainly ubiquitous and very useful, is not especially efficient or fast when compared with others, e.g. gRPC.

- Limiting the experiments to a small number of switches have their own performance limit for accepting flow programming requests.

Therefore, the resulting numbers in an end-to-end approach speak more to the performance limitations of the REST API libraries and the switches than to the actual internal performance capabilities of the controller platform itself. The test driver applications used the ONOS core Java APIs, and for experiments that required OpenFlow interactions, a “null” southbound driver was implemented in ONOS to eliminate potential performance limits imposed by slower Openflow devices.

Additionally, we believe it is not enough to cover performance using only a single-node controller. Given the vulnerability of a single-node setup, it is unlikely that an operator of a mission-critical network would use a single instance controller in a production setting. Furthermore, while performance of a single-node is a useful baseline, it does not paint a complete picture as it completely ignores crucial architecture, design, and implementation aspects of a fault-tolerant clustered controller.

It is impossible to extrapolate performance of a multi-node controller from a single-node configuration. Without appropriate design of distributed state management primitives and other coordination functions, one can easily get diminished performance or even weaker reliability with a multi-node controller. High-availability and ensuring correctness of operation in the face of failures certainly trumps performance and is therefore a critical dimension to be considered.

5 Service Provider Use Cases Demanding ONOS Performance

The network edge is where services are deployed to end users. Historically, these were purely connectivity services, but increasingly there is recognition that latency sensitive applications will require edge processing to meet performance requirements. Applications like augmented reality and coordination of autonomous vehicles cannot withstand the added time required to send messages back to centralized clouds for processing.

To meet these needs at the edge, service provider networks will need to support a wide variety of services in additon to a wide range of user and device characteristics. CORD (Central Office Re-architected as a Datacenter), perhaps the ‘killer’ use case leveraging ONOS, provides a compelling solution by re-architecting the central office at the network edge as a datacenter using a collection of commodity servers by integrating SDN, NFV, and cloud principles.

In this new network edge, service providers need the ability to manage multiple services and hide the complexity of orchestrating, provisioning, scaling and composing virtualized network functions. A service provider also wants to:

- Identify and provide differentiated QoS to subscribers and their individual applications and flows.
- Optimize spectrum usage on multiple RANs by managing flows.
- Enable enterprise customers to observe and manage users, their applications and flows and do control on a per flow basis.
- Create self driving networks or data driven networks that use analytics to better understand current state, predict future needs, and automatically make adjustments to ensure optimal network operation.

It is imperative that this network enables rapid deployment of new services and allows for elastic scalability based on demand. As a result, performance, scalability and high availability become very important at the network edge.

For example, the latest switch based on Barefoot Networks’ Tofino silicon can support very high throughput with programmable forwarding plane and very high flow mods per second. ONOS enables such switches to sustain high throughput control operations at the edge of the network, where service providers want to manage flows/applications on a per-subscriber basis and take advantage of the next generation Radio Access Networks requiring tight feedback control loops.

6 Conclusion

ONOS continues to be a lead SDN controller as measured by performance, scalability and resiliency. Performance has continually improved across releases even as ONOS has added support for multiple protocols and applications.

ONOS is designed to help service providers build carrier-grade software-defined networks architected for high availability, scalability and performance. It is important for the service providers to have a controller that can successfully manage the challenges of traffic and application explosion, thereby increasing the average revenue per user and lowering CapEx and OpEx. Therefore, it is important for ONOS to demonstrate its outstanding performance and scalability which is required in the key use cases of ONOS especially at the edge of the network where providers want to provide per-subscriber, per-flow services and also perform RAN optimizations.

The ONOS Controller's continued leadership in performance and scalability is a testament to its architecture and its community's commitment to service provider requirements.

7 References

[1] ONOS Blackbird white paper for general audience http://onosproject.org/wp-content/uploads/2014/11/Whitepaper-ONOSBlackbirdperformance-generalaudience-Apr-7.pdf

[2] ONOS Blackbird white paper for technical audience http://onosproject.org/wp-content/uploads/2014/11/PerformanceWhitepaperBlackbirdrelease-technical.pdf

[3] ONOS Kingfisher white paper for technical audience https://wiki.onosproject.org/download/attachments/13994369/Whitepaper-%20ONOS%20Kingfisher%20release%20performance.pdf?version=1&modificationDate=1501114080000&api=v2

[4] OpenDaylight & ONOS Performance white-paper http://onosproject.org/2016/05/20/opendaylight-onos-performance-white-paper/

About ONOS

ONOS is the open source SDN networking operating system for Service Provider networks architected for high performance, scale and availability. The ONOS ecosystem comprises the ONF, organizations that are funding and contributing to the ONOS initiative, and individual contributors. These organizations include AT&T, China Unicom, Comcast, Google, NTT Communications Corp., Deutsche Telekom, Verizon, Ciena Corporation, Cisco Systems, Inc., Ericsson, Fujitsu Ltd., Huawei Technologies Co. Ltd., Intel Corporation, NEC Corporation, Nokia, Radisys and Samsung. See the full list of members, including ONOS' collaborators, and learn how you can get involved with ONOS at onosproject.org.

ONOS is an independently funded software project supported by The Linux Foundation, the nonprofit advancing professional open source management for mass collaboration to fuel innovation across industries and ecosystems.