# OpenDaylight Performance Stress Test Report

## v1.0: Lithium vs Helium

## Intracom Telecom SDN/NFV Lab

www.intracom-telecom.com | intracom-telecom-sdn.github.com | sdn-nfv@intracom-telecom.com

# OpenDaylight Performance Stress Tests Report v1.0: Lithium vs Helium

**Executive Summary**

In this report we investigate several performance aspects of the OpenDaylight Lithium RC0 controller and compare them against the Helium SR3 release. The investigation targets on stability and scalability tests. Stability tests explore how controller throughput behaves in a large time window with a fixed topology connected to it, the goal of which, is to detect performance fluctuations over time. Scalability tests measure controller performance as the switch topology scales, giving hint on the controller's upper bound.

## Contents



## 1. Introduction

In this report we investigate several performance aspects of the OpenDaylight controller and compare them between the Helium and Lithium releases. The investigation targets several objectives such as controller throughput, switch scalability and stability (sustained throughput).

For our evaluation we have used NSTAT [1], an open source environment written in Python for easily writing SDN controller stress tests and executing them in a fully automated and end–to–end manner. The key components in NSTAT tests are the SDN controller and the SouthBound OpenFlow traffic generator. NSTAT is responsible for automating and sequencing every step of the testing procedure, including controller and generator lifecycle management and orchestration, test dimensioning (i.e. iterating over a large space of experimental parameters), online performance and statistics monitoring, and finally reporting. NSTAT does exhaustive stress testing, which means it repeats the test over a multi-dimensional experimental space within the same run. In this way the user is able to execute multiple test cases within a single run.

In all tests we have used MT-Cbench [2] as the SouthBound traffic generator. MT-Cbench is a direct extension of the C-bench emulator [3] which uses threading to generate OpenFlow traf-fic from multiple streams in parallel. The motivation behind this extension was to be able to boot–up and operate network topologies with OpenDaylight much larger than those being able with the original Cbench. We note here that, as with the original Cbench, the switches emulated with MT–Cbench implement only a subset of the OpenFlow protocol, and therefore the results presented here are expected to vary in real conditions. In future releases of NSTAT we plan to provide improvements in our generator towards providing better OpenFlow protocol support.

## 2. Results summary

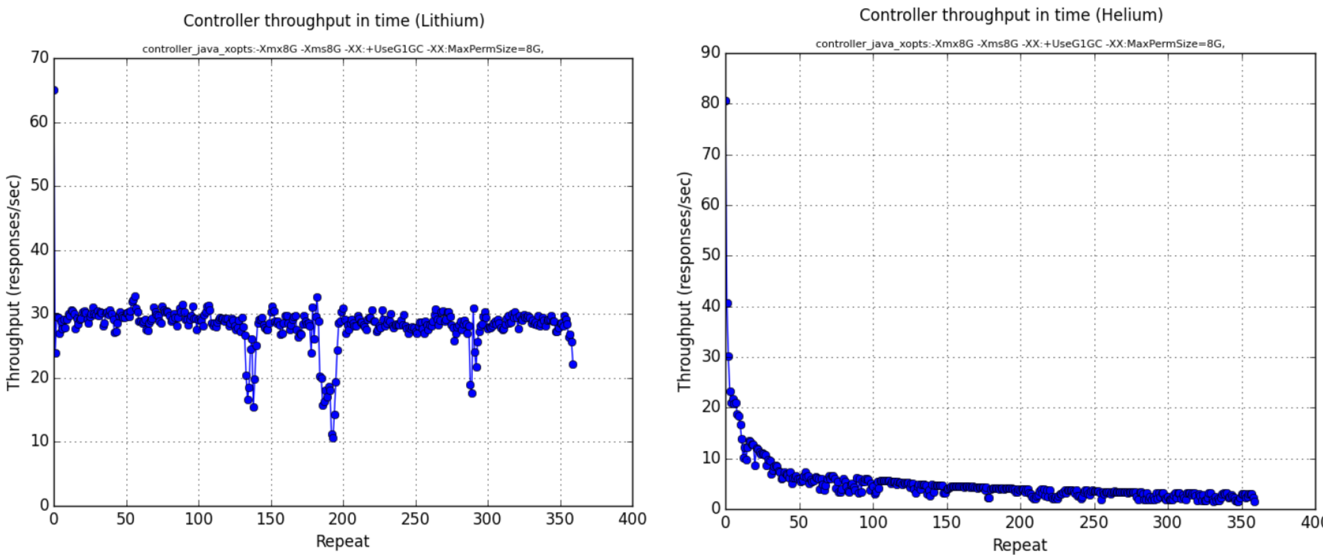
Our key findings in stress tests where the controller DataStore is involved witness notable improvements in the Lithium DataStore, which translate into better throughput in scalability tests as well as better sustained throughput in stability tests. Specifically, in a test running for an hour with aggressive SouthBound traffic that is specifically designed to stress the DataStore, Lithium exhibits steady behavior unlike Helium that shows a downward slope all the way down to zero. In scalability tests that involve the DataStore we observe a significant improvement from 1 to $5k$ switches reaching up to 8x. In the same cases involving just the OpenFlow plugin, the improvement is smaller yet evident (by almost $1.5x$ in the best case).

## 3. Experimental setup

The setup used in all experiments presented in this report is summarized in the Table 1.

**Table 1:** Stress tests experimental setup.

| | |
|---|---|
| Operating system | Centos 7, kernel 3.10.0 |
| Server platform | Dell R720 |
| Processor model | Intel Xeon CPU E5–2640 v2 @ 2.00GHz |
| Total system CPUs | 32 |
| CPUs configuration | 2 sockets × 8 cores/socket × 2 HW-threads/core @ 2.00GHz |
| Main memory | 256 GB, 1.6 GHz RDIMM |
| Controllers under test 1 | OpenDaylight Lithium RC0 |
| Controllers under test 2 | OpenDaylight Helium SR3 |




(a) (b)

**Fig. 1:** Throughput value.

## 4. Part A – Stability Stress Tests

Stability tests explore how controller throughput behaves in a large time window with a fixed topology connected to it. The controller accepts an aggressive rate of incoming OpenFlow packets from the switches (Packet–In's) and its response throughput is sampled periodically. We evaluate two different controller configurations: in the first one, the controller is configured to directly reply to the switches with a predefined message at the OpenFlow plugin level ("RPC mode"). In the second one, the controller additionally performs updates in its DataStore ("DataStore mode"). In all cases, the MT–Cbench generator is configured to operate at "Latency mode", which means that each switch sends a Packet–In message only when it has received a reply for the previous one.

### 4.1 Test A1 500 switches, Datastore mode, 1 hour running

NSTAT configuration:

- Controller: Datastore mode
- MT–Cbench generator: Latency mode, 50 switches per thread, 10 threads, 8 seconds between thread creation
- 360 internal generator repeats, 10 seconds each (1h total running mode)

### 4.2 Test A2 500 switches, RPC mode, 1 hour running

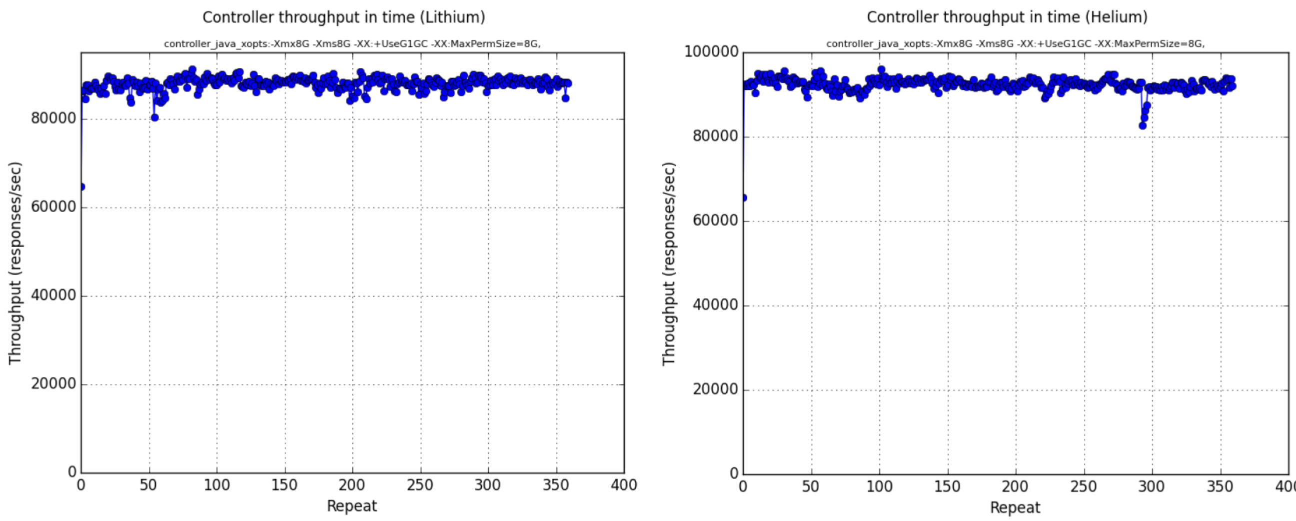
NSTAT configuration:

- Controller: RPC mode
- MT–Cbench generator: Latency mode, 50 switches per thread, 10 threads, 8 seconds between thread creation
- 360 internal generator repeats, 10 seconds each (1h total running mode)

### 4.3 Test A3 500 switches, Datastore mode, 12h running

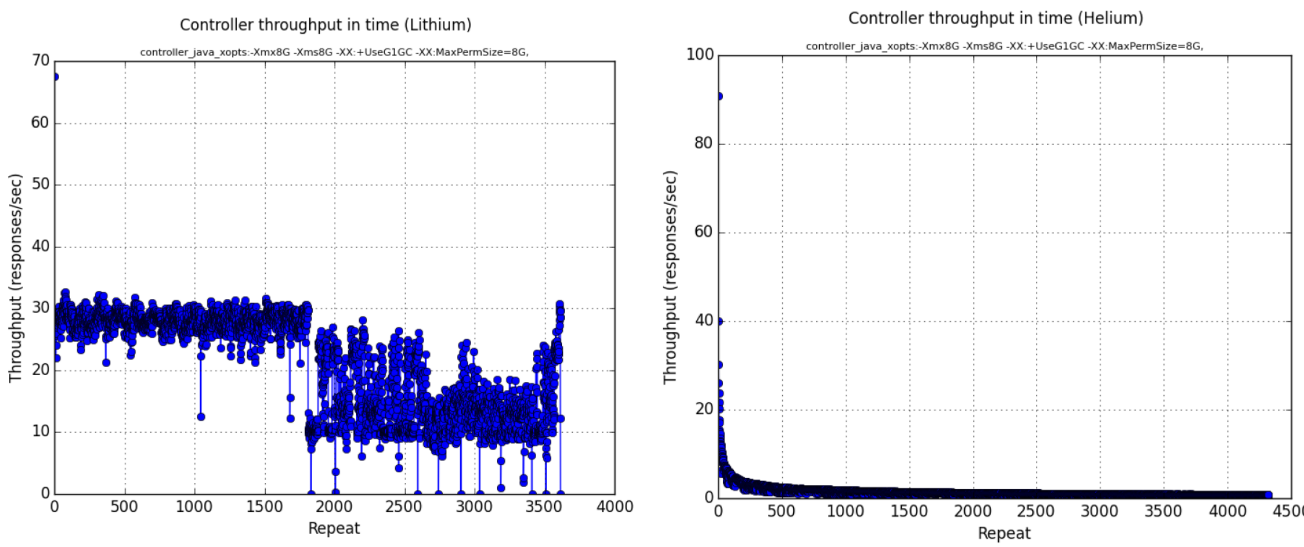
NSTAT configuration:

- Controller: Datastore mode
- MT–Cbench generator: Latency mode, 50 switches per thread, 10 threads, 8 seconds between thread creation
- 4320 internal generator repeats, 10 seconds each ($12h$ total running mode)




(a) (b)

**Fig. 2:** Throughput value.




(a) (b)

**Fig. 3:** Throughput value.

### 4.4 Test A4 16 switches, RPC mode, 1 hour running time, external generator repeats

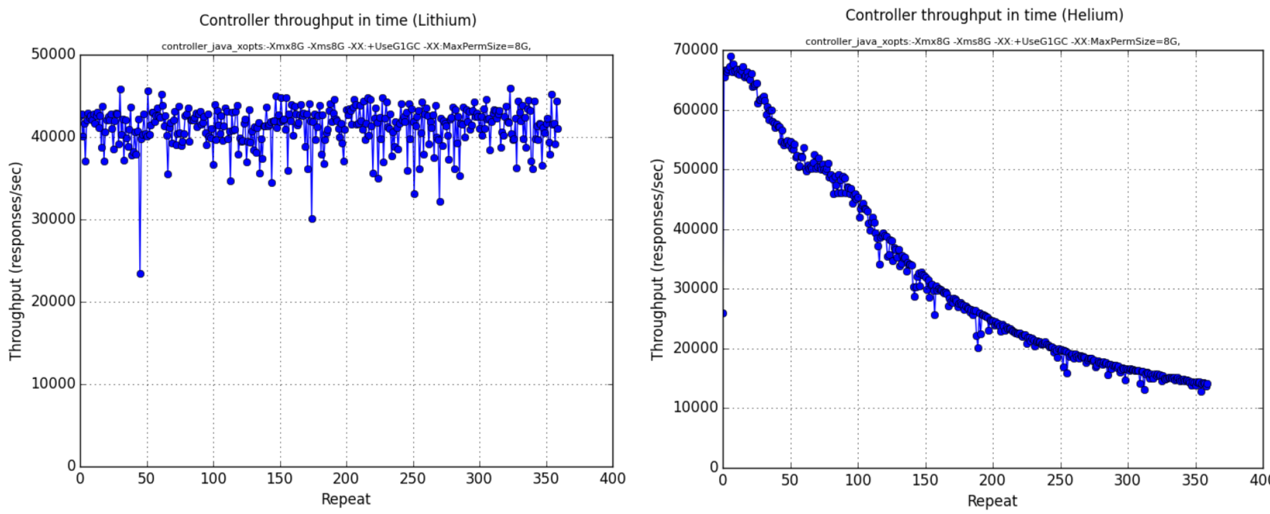
NSTAT configuration:

- Controller: RPC mode
- MT–Cbench generator: Latency mode, 16 switches per thread, 1 thread
- 360 external generator repeats, 10 seconds each ($1h$ total running mode)

Helium degradation can be attributed to the fact that this test restarting topology which seems to trigger the DataStore "pure RPC" case (Test 4.2).

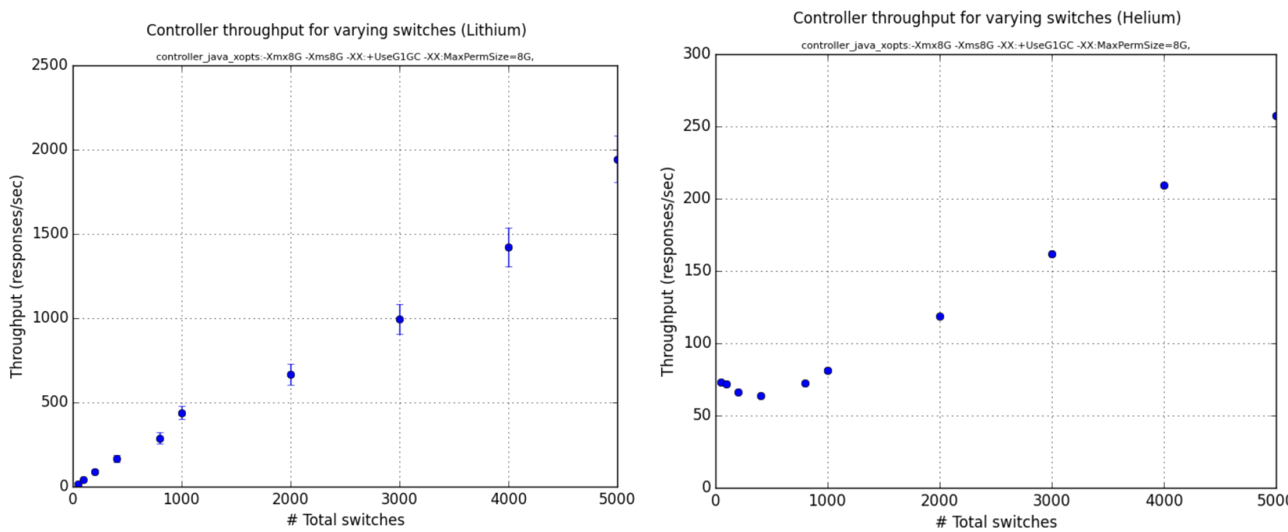
## 5. Part B – Scalability Stress Tests

Scalability tests measure controller performance as the switch topology scales, giving hint on the controller's upper bound. We perform two kinds of stress tests *active* and *idle scalability* tests. With the active tests we try to explore how controller throughput scales as the topology scales. In this case we send OpenFlow traffic from booted topologies with varying number of switches and measure controller throughput. With the idle tests we explore how to optimally boot–up topologies of varying size. In this case we gradually boot up switches in idle mode and count installed switches and installation times. Switches are booted in batches and with a fixed delay between




(a) (b)

**Fig. 4:** Throughput value.




(a) Lithium throughput (b) Helium throughput

**Fig. 5:** Throughput value.

each batch. As in the stability tests, in these tests as well we explore both operation modes of OpenDaylight (RPC and DataStore), and we also use the MT–Cbench generator in Latency mode.

## 5.1 Test B1: Active scalability test, Datastore mode

NSTAT configuration:

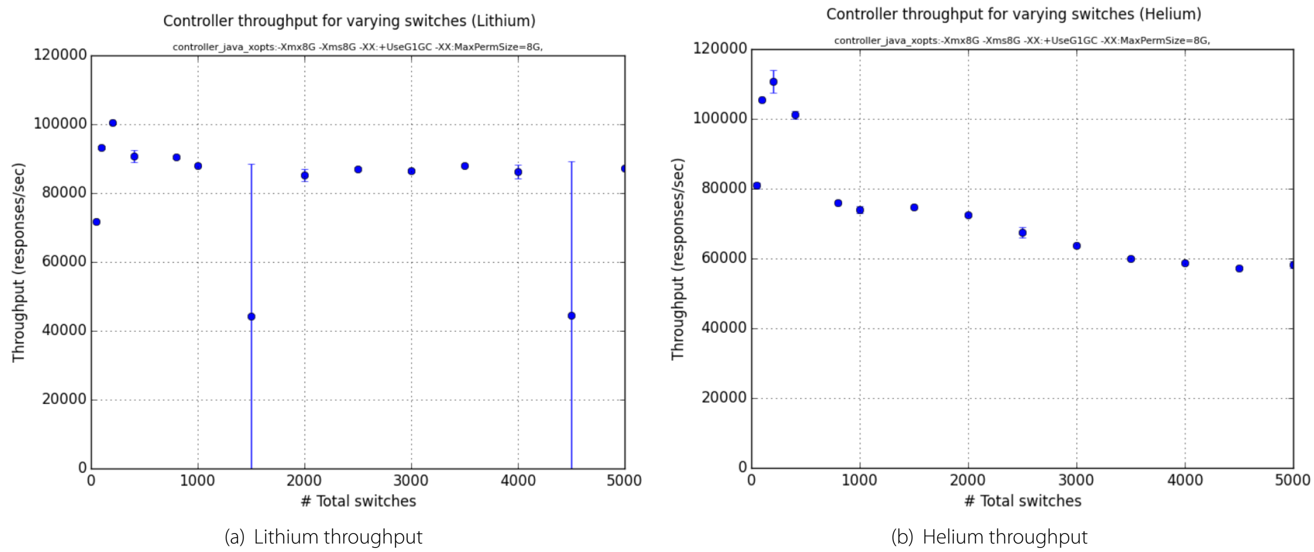
- Controller: Datastore mode
- MT–Cbench generator: Latency mode, 50 switches per thread, [1,2,4,8,16,20,40,60,80,100] threads, 15 seconds inter–thread creation delay ($1h$ total running mode)

Lithium starts outperforming Helium for switch numbers larger than 400. After that, it consistently exhibits better relative throughput by an ever increasing factor that reaches almost $8x$ for 5000 switches.

## 5.2 Test B2: Active scalability test, RPC mode

NSTAT configuration:

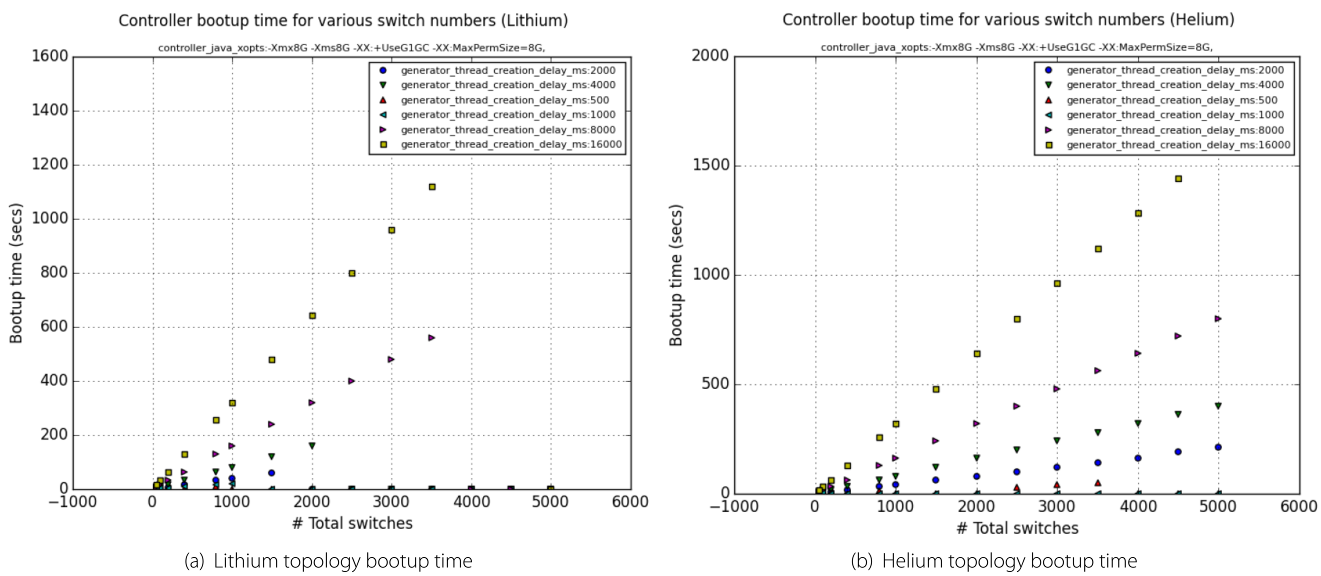
- Controller: RPC mode
- MT–Cbench generator: Latency mode, 50 switches per thread, [1,2,4,8,16,20,30,40,50,60,70,80,90,100] threads, 15 seconds inter–thread creation delay.




(a) Lithium throughput

(b) Helium throughput

**Fig. 6:** Throughput value.




(a) Lithium topology bootup time

(b) Helium topology bootup time

**Fig. 7:** Topology bootup time.

### 5.3 Test B3: Idle scalability test, RPC mode

NSTAT configuration:

- Controller: Datastore mode
- MT–Cbench generator: Latency mode, 50 switches per thread, [1, 2, 4, 8, 16, 20, 30, 40, 50, 60, 70, 80, 90, 100] threads, inter–thread creation delay [500, 1000, 2000, 4000, 8000, 16000] $s$

Helium achieves to successfully boot up large topologies with much less inter–thread creation delay than Lithium ($2s$) and hence in quite smaller total time. An inter–thread delay that always works for Lithium is that of $16s$

## 6. Future Releases

In future releases of this report we are going to additionally include:

- SouthBound stability and scalability tests with Mininet
- NorthBound stability and scalability tests
- system–level and controller performance statistics and profiling results.

This functionality is going to be part of future NSTAT releases.

## 7. Contributors

- Nikos Anastopoulos
- Vaios Bitos
- Panagiotis Georgiopoulos
- Apostolis Glenis
- Andreas Pantelopoulos
- Konstantinos Papadopoulos

## References

[1] "NSTAT: Network Stress Test Automation Toolkit." https://github.com/intracom-telecom-sdn/nstat.

[2] "MT–Cbench: A multithreaded version of the Cbench OpenFlow traffic generator." https://github.com/intracom-telecom-sdn/mtcbench.

[3] "Cbench: OpenFlow traffic generator." https://github.com/andi-bigswitch/oflops/tree/master/cbench.