Towards Smart and Intelligent SDN Controller

- Through the Generic, Extensible, and Elastic Time Series Data Repository (TSDR)

YuLing Chen, Dell Inc.
Rajesh Narayanan, Dell Inc.
Sharon Aicler, Cisco Systems Inc.
TSDR Team, ODL Lithium
July 20, 2015
Agenda

Time Series Data Analysis Introduction
TSDR Objectives in ODL
TSDR in Lithium
Roadmap and Future Directions
TSDR beyond ODL
Demo
The Power of Time Series Data
What is Time Series Data

• Time Series Data is a sequence of data points with time stamps.
  – Measurements
  – Log files
  – Events generated from machines or software

• Huge amount of time series data being generated every day.
  – Cloud Infrastructures
  – Software applications
  – Network equipment
  – Security appliances
  – IoT devices
Why we need time series data analysis

• The power of time series data analysis is...
  – leveraging what happened in the past (historical view)
  – together with what is happening now (real-time view)
  – to predict what’s going to happen next (predictive data analysis)
  – and take proactive actions (prescriptive data analysis with automation)

• Time Series Data Analysis has been successful in many areas including...
  – Financial Market
  – Weather forecasting
  – Economics
  – Health care
  – Insurance

• The Goal of TSDR in ODL is to apply time series data analysis in SDN.
  – Big data technologies make the time series data analysis possible on high velocity of data
Example Use Case – Traffic congestion prediction with automated control

SDN controlled network

OpenDaylight + TSDR

1. Collect stats from the network and store into TSDR
2. Data analysis through data analytics engines integration
3. Traffic flow redirection from A->F to A->B->F and A->D->E->F

- Red: Predicted congestion path in the next 24 hours
- Green: Healthy path in the next 24 hours
Other example data driven applications

- Traffic classification
- Congestion control
- Traffic pattern prediction
- Traffic redirection with route analysis
- Network issue events prediction
- Security and Auditing analysis
- Troubleshooting network problems
- Resource optimization
- Network Performance Analysis
TSDR Objectives in ODL
TSDR goals in ODL

• To help with the scalability and performance of ODL controller
  – In Helium, the time series data, such as OpenFlow stats, were only available from the InMemory data store.
  – In Helium, the OpenFlow stats data started to drop from InMemory data store after three seconds in large deployment scenarios.
  – Leveraging Stats Plane concept to separate time series data processing from the control plane and data plane.

• To enable and encourage data driven applications built from ODL controller
  – For example, a traffic pattern prediction with reconfiguration app could be built on top of ODL controller and TSDR.

• Help to create an intelligent and ‘smart’ controller
  – With various data driven applications leveraging data from TSDR and feeding the analytics result back to the SDN controller for dynamic flow configuration.
To realize SDN Stats Plane using TSDR

- Separates statistics collection and storage from control plane.
- Generic, extensible, and elastic architecture framework supporting various types of time series data.
- Creates new data-driven application platform for SDN.
To provide a generic platform for time series data

✓ A Data Collection Framework
  o To incorporate a broad range of data collectors for different types of time series data.
  o To facilitate open integration with the specification of polling, pushing, and notification interfaces for time series data collection.

✓ A Common Data Model
  o to transform different types of time series data into a common data representation format.

✓ A Scalable and pluggable Data Repository
  o To store large amount of time series data.
  o To allow plugin of different types of data stores.

✓ A generic open integration API
  o For integration with third party analytics engines.

✓ An optimized time series data maintenance solution
  o Periodic Data Aggregation and Purging solutions optimized for time series data

#ODSummit
To enable advanced analytics for business optimization
- with third party analytics engine integration

✓ Descriptive time based data analytics on different data sources
  o Leveraging the common time series data model.
  o Leveraging time stamps that are common in the data model.
  o Leveraging integration with third party data analytics engine or visualization tools.

✓ Predictive and Prescriptive data analytics
  o Automated pattern discovery.
  o Event prediction based on time series data analytics
  o Automated correlation among multiple data sources
  o Prescriptive actions based on the advanced analytics results.
  o By integration with advanced data analytics engines.

✓ Automation based on analytics results
  o Automation actions triggered from analytics results for SDN controller optimization.
  o Integration with ODL Controller for re-configuration and redirection of the traffic flows.
To combine real-time and historical analytics

✓ Streaming data processing for real-time data analysis
  o Apply streaming data processing technologies for real-time data analysis.
  o Apply advanced data analytics on real-time streaming data.
  o Enable real-time automated actions for business optimization.

✓ Scalable data storage for historical view
  o Capture large amount of streaming data within limited time window.
  o Support active queries from the large time series data repository in reasonable response time.

✓ Feedback of historical data analytics result into real-time automation
  o Provide capability of feeding back the historical data analytics result into real-time automation engine.
TSDR Capabilities and Architecture Framework Roadmap

- Data Transportation Service
  - Data Transportation SPI
    - SNMP Alarms
    - Notification (Pub/Sub)
    - FTP CSV Files

- Northbound Open Integration API
  - Data Query Service
  - Data Aggregation Service
  - Data Purging Service

- Real-time processing (Data Cleansing, Filtering, and Pre-processing)
  - Data Transformation
    - Data Storage Service

- Prescriptive Analytics Integration
  - Automation Engine

- Data Collection Service
  - Data Collection SPI
    - SNMP Collector
    - Notification Collector
    - sFlow Collector
    - Syslog Collector

- TSDR Data Model
  - TSDR Persistence SPI
    - HBase Plugin
    - Cassandra Plugin
    - MySQL Plugin

#ODSUmmit
TSDR in Lithium
TSDR realizes SDN Stats Plane concept

- Separates statistics collection and storage from control plane.
- Generic, extensible, and elastic architecture framework supporting various types of time series data.
- Creates new data-driven application platform for SDN.
TSDR Data Services including Data Collection, Data Storage, Data Query, Data Purging, and Data Aggregation are MD-SAL services.

Data Collection service receives time series data published on MD-SAL from MD-SAL southbound plugins.

Data Collection service communicates with Data Storage service to store the data into TSDR.

TSDR data services access TSDR Data Stores such as HBase Data Store through generic TSDR Data Persistence Layer.
Functions and Capabilities delivered in Lithium

✓ Data Collection
  o A notification based data collector to collect OpenFlow Stats in the network

✓ Common Data Model
  o The first version of time series data model that incorporates measurements and log entries.

✓ Data Storage
  o TSDR persistence layer with SPI
  o Two TSDR data stores: HBase (NoSQL) and Apache H2(SQL)
  Note: HBase single node deployed on the same host as ODL controller is supported in Lithium.

✓ Query command
  o “tsdr:list” command to query the data from TSDR data stores.
    o tsdr:list {Category}[startTime][endTime]
    o Example: tsdr:list FlowStats ‘07/20/2015 08:00:00 AM’ ‘07/20/2015 08:15:00 AM’
      This command gives the latest 1000 records from TSDR datastore that matches the data category and time range.

#ODSumsit
TSDR Common Data Model in Lithium

- TSDR common data model in ODL
- Lithium captures two types of time series data:
  - Measurements
  - Log entries
- The common data model also supports two data granularities:
  - Fine-grained raw data
  - Aggregated roll up data
- The characteristics of the design:
  - Generic
  - Extensible
  - Scalable
  - Performance Optimized
- OpenFlow stats implementation delivered based on this data model:
  - Flow Stats
  - Interface Stats
  - Group Stats
  - Flow Table Stats
  - Queue Stats
  - Meter Stats

#ODS Summit
TSDR ODL Lithium Scalability and Performance Test Result

✓ With HBase single node on 6GB RAM, 4vCPU(4 core, 2.40 GHz) VM, we achieved:

- 255 OpenvSwitches
- Polling at 15 seconds interval from each switch
- In each poll,
  - #1273 flows, #1019 ports, and #64770 flow table entries (254 flow table per switch)
  - Each flow table entry generates 3 records
  - Each flow entry generates 2 records
  - Each port entry generates 12 records
- Data Collection turned on for five days
  - Observed steady data collection with reasonable CPU and Memory utilization rates
- With more than 500 million rows and 24 GB data in HBase tables
- Query results from “tsdr:list” command on Karaf console return within 2 ~ 3 seconds

Note: The benchmarking result could be higher if we leave the data collection for longer time.

✓ Would expect higher benchmarking data with Hadoop Cluster in the next release

#ODSummit
TSDR Links and references in ODL

- **TSDR Main page**
  
  [https://wiki.opendaylight.org/view/TSDR:Main](https://wiki.opendaylight.org/view/TSDR:Main)

- **TSDR Proposals in ODL Lithium**
  
  [https://wiki.opendaylight.org/view/Project_Proposals:Time_Series_Data_Repository](https://wiki.opendaylight.org/view/Project_Proposals:Time_Series_Data_Repository)

- **TSDR Design docs in ODL Lithium**
  

- **TSDR Installation Guide and User Guide**
  
  

- **TSDR Release Review and Release Notes**
  
  
TSDR Roadmap
In large data center deployment scenarios, TSDR Distributed Architecture provides better performance and scalability for both ODL controller and TSDR itself.

In distributed architecture, TSDR data services are deployed in a separate MD-SAL instance.

The data pushed onto MD-SAL messaging bus by ODL southbound plugin are propagated to the other MD-SAL instance for TSDR data services to process into TSDR data repository.
TSDR in Beryllium and beyond -- (1)

✓ Data Collection Framework with more data collectors support
  ▪ OVSDB collector to enable data applications in ODL OpenStack support

✓ Data Query Service
  ▪ With Open Integration APIs for third party data analytics engine integration

✓ Hadoop Cluster Support
  ▪ For performance and scalability benchmarking

✓ Data Visualization and analytics engine integration
  ▪ Grafana integration for visualization
  ▪ Direct visualization from ODL DLUX GUI
  ▪ Python wrapper around REST APIs for third party machine learning tools integration

✓ A compelling use case implementation using TSDR
  ▪ For example, with data collection on the traffic routes in SDN environment, visualize the traffic pattern over the time, predict the traffic pattern in the future, instruct SDN controller to reconfigure the network to optimize the traffic distribution in the network.

#ODSSummit
TSDR in Beryllium and beyond -- (1)

✓ Data Collection Framework with more data collectors support
  ▪ OVSDB collector to enable data applications in ODL OpenStack support

✓ Data Query Service
  ▪ With Open Integration APIs for third party data analytics engine integration

✓ Hadoop Cluster Support
  ▪ For performance and scalability benchmarking

✓ Data Visualization and analytics engine integration
  ▪ Grafana integration for visualization
  ▪ Direct visualization from ODL DLUX GUI
  ▪ Python wrapper around REST APIs for third party machine learning tools integration

✓ A compelling use case implementation using TSDR
  ▪ For example, with data collection on the traffic routes in SDN environment, visualize the traffic pattern over the time, predict the traffic pattern in the future, instruct SDN controller to reconfigure the network to optimize the traffic distribution in the network.

#ODSsummit
TSDR in Berrylium and beyond – (2)

- **Data Aggregation Service**
  - To aggregate raw data to roll-up data

- **Data Purging Service**
  - To purge the out-of-dated data periodically

- **ODL Clustering support**
  - Deployment of TSDR in ODL Clustering environment

- **TSDR Distributed Architecture in ODL**
  - Separating TSDR data services into a second MD-SAL instance

- **More TSDR Data Store plugins such as Cassandra plugin**
TSDR Beyond ODL
TSDR use cases beyond ODL

With ODL being adopted and used in various customer scenarios, we could see TSDR being used in the following possible use cases:

- IoT applications
- NFV use cases
- Software Defined Data Center analysis and automation
Other TSDR example use cases

✓ Using TSDR in IoT data analytics applications
  - With generic, open, elastic and scalable architecture, TSDR could easily fit into multi-level IoT data analytics architecture.

✓ Using TSDR for NFV data analysis and automation
  - With generic model based time series data platform, TSDR could be used for data analytics purposes in NFV architecture with optional SDN controller deployed together.

✓ Using TSDR in software defined data center analytics
  - Such as DDoS Attack detection and automated protection from such risks.
TSDR Demo Set up

Open vSwitches

switch1  switch2  switchN

host1  host2  hostM

Collects OpenFlow Stats

OpenDaylight SDNController

TSDR

HBase

#ODSummit
Visualization of data from TSDR data stores
Thank You #ODSummit