Time Series Data Repository

Name

Time Series Data Repository

Repo Name

tsdr

Description

Time Series Data Repository (TSDR) project in OpenDaylight (ODL) contains a time series data repository and a set of time series data services as MD-SAL services modules that collects, stores, queries and maintains time series data in Open Daylight deployment environment. This includes statistics data as well as operational configuration data in the environment. For example, currently, OpenFlow plugin provides the capabilities for collecting OpenFlow statistics from OpenFlow enabled switches that connects to ODL Controller and stores the current value of the corresponding data in ODL operational data store. As one of TSDR data collection mechanisms, TSDR will be able to periodically poll the ODL operational data store to obtain the current value of the data and store them as time series data into TSDR. For another example, notification based data collection plugin is under development to collect various types of statistics data including OpenFlow statistics and made them available on MD-SAL messaging bus. As one of TSDR data collection mechanisms, TSDR will be able to subscribe to receive these data published on MD-SAL messaging bus and store them as time series data into TSDR. For the data captured in TSDR, northbound interfaces will be available in ODL northbound APIs for external parties to retrieve the time series data from the TSDR.

The goal of TSDR project is to create a scalable and extensible architectural framework for ODL to capture the time series data into a persistence data store. In this project, we will create a generic and extensible data model that is optimized for time series data collection, storage, query, and maintenance. We will also create an abstract and generic TSDR persistence layer that allows various third party products to plug in their own implementation of the data store to fit their needs. As deliverables for the first version of TSDR, we will provide the example TSDR plugin, which is an HBase TSDR plugin, to realize the functionality of time series data persistence, query, and maintenance including data purging and aggregation.

In the project, we will fully utilize MD-SAL's clustering capability to handle the performance and scalability in time series data processing. In large deployment scenarios, such as millions of flows being pushed to TSDR, one or multiple separate MD-SAL instances can be created to focus on time series data processing, which forms a distributed architecture. By doing so, we separate the time series data processing logic from the ODL controller so that the controller could focus on the core functionality of flow control.

With TSDR functionality being added to ODL, various applications could be built on top of ODL leveraging the data stored in TSDR over the time that captures the performance, health, and operational configuration in ODL environment. This includes applications for security risk detection, performance analysis, operational configuration optimization, traffic engineering, and network analytics with automated intelligence.

Component Diagrams

The following two figures show two deployment scenarios of the TSDR in ODL environment. In small to medium deployment scenarios, an integrated architecture deploys TS data services together with other MD-SAL services of ODL controller in one MD-SAL instance (or server); In large deployment scenarios, a distributed architecture deploys TS data services into one or multiple separate dedicated MD-SAL instances (or servers) so as to handle large amount of data that needs to be collected, stored, queried and maintained periodically.

blocked URL

blocked URL

Scope

Following are the components and services that we intend to create in TSDR project:

Data Collection Service

Data Collection Service collects time series data in ODL environment. In general, there are three ways for collecting data from ODL controlled network devices. One is for ODL to periodically poll/query the network devices for statistics and operational data, such as traditional SNMP data polling. Another approach is for the network devices actively pushing data to the controller, such as using NetFlow or sFlow protocols. The third one is leveraging the messaging bus for the network devices to publish the data to the messaging bus and the data collection service would subscribe and receive the corresponding data from the messaging bus.

Various southbound plugins will be created to handle different data being collected using different protocols and mechanisms. These plugins collect the data and push to ODL data store and/or the controller messaging bus. Data Collection service will periodically poll/query the ODL data store to obtain the data or subscribe to the data published by the southbound plugins onto the messaging bus.

After the data is collected, Data Collection service will inform Data Storage Service from MD-SAL to store the data into ODL Time Series Data Repository.

Data Storage Service

Data Storage Service receives the requests from Data Collection Service and constructs Time Series Data Model based on the request input data. Then it will call the generic TSDR Data Persistence Layer to persist the data into Time Series Data Store. Performance optimization needs to be considered in this service including batch processing and buffering.

Time Series Data Model

A generic Time Series Data Model will be created for storing, querying, aggregating, and purging of the time series data into persistence data store. This data model will be designed for the operations on the time series data without addressing the implementation details of the underlining data store.

HBase Time Series Data Repository Plugin

As an example plugin, we will provide a plugin using HBase on top of Hadoop clustering system to realize the TSDR functionality in ODL environment. The plugin implements the TSDR Persistence APIs to store, query, purge, and aggregate data in the time series data repository. A schema designed for optimizing the storage and query of the data will be provided in the implementation.

HBase Time Series Data Repository

As an example implementation of time series data repository, we will provide an example implementation using HBase on top of Hadoop clustering system to realize the TSDR functionality specified in this proposal. As a popular column-oriented NoSQL data base, HBase has been widely used in the related industry for various applications to handle the persistence of large amount of data.

Although we provide HBase as an example implementation, the architecture and framework this project provides will work with other data stores such as Cassandra, MongoDB, CouchDB, and among others. In the cases of using data stores other than HBase, corresponding plugins need to be created to plug into the generic Time Series Persistence Layer so as to implement the corresponding APIs for storing, querying, and maintaining the data in the time series data repository.

Data Query Service

Upon receiving the API calls from Northbound Interface, Data Query Service calls the TSDR Data Persistence API to query the Time Series data from TSDR. The API calls will be routed to southbound plugins such as HBase TSDR Plugin to process the request and return the values. The result will be converted and assembled by Data Query Service so as to conform to the Northbound Interface of the ODL controller. Performance optimization needs to be considered in this service including using various techniques provided from the data persistence layer.

Data Aggregation and Purging Service

Periodically there will be daemon processes at the background to provide Data Aggregation and Purging services. The raw historical data can be rolled-up to aggregated data and the original raw data could be purged from the DB tables to save the storage space as well as improve the data query speed with smaller data set. Data Aggregation and Purging Services make calls to TSDR Data Persistence API to aggregate and purge data from TSDR. The requests will be routed to southbound plugins such as HBase TSDR Plugin to complete the operations.

Time Series Northbound APIs

A set of interfaces will be added into existing ODL Northbound Interface for querying time series data from the TSDR. The API specification will be provided as project deliverables with URLs, request, and response details in json format.

Presentations

File:TSDR Proposal ODL.pptx

Resources Committed (developers committed to working)

- Yuling C (Dell) - yuling_c
- Sharon Aicler (Cisco)
- Basheeruddin Ahmed (Cisco)
- Mohnish Anumala (Dell) - manumala

Initial Committers

- Yuling C (Dell) - yuling_c

- Sharon Aicler (Cisco)
- Basheeruddin Ahmed (Cisco)
- Mohnish Anumala (Dell) - manumala

Vendor Neutral

Some of the base functionality described in this proposal is implemented in the Dell Active Fabric Manager. Some code may be leveraged for use in OpenDaylight. All contributed code will adhere to OpenDaylight's copyright and license policies.

Meets Board Policy (including IPR)