SNMP Agent HLD

Requirement

1.0	Register and handle requests for individual MIBs.	
2.0	Support scalar and tabular OIDs.	
3.0	Support SNMP floating types.	
4.0	Reconnect to snmpd in case connectivity loss.	
5.0	Store up-to-date values provided by collectd read plugins.	
6.0	Support all SNMP versions	
7.0	Support standard MIBs where possible	
8.0	Traps will not be supported. Get, Get_next and walk will be supported	

Overview

SNMP

Simple Network Management Protocol (SNMP) is an Internet-standard protocol for collecting and organizing information about managed devices on IP networks and for modifying that information to change device behaviour. Devices that typically support SNMP include routers, switches, servers, workstations, printers, modem racks and more.

Extending SNMP using AgentX

AgentX is a protocol defined in RFC 2741 allowing a master agent to be extended by independent sub-agents. The architecture was designed to be simple in respect of authorization, privacy and encoding. It offers full support for data retrieval (Get, GetNext and GetBulk), data modifications (Set) and notifications (Trap). AgentX transport mappings are specified for UNIX domain sockets and TCP. Every connection is spit up into several sessions, which in turn can convey several transactions.

Management Information Database (MIB)

Management Information Base (MIB) is a collection of Information for managing network element. The MIB is comprised of managed objects identified by the name Object Identifier (Object ID or OID). Each Identifier is unique and denotes specific characteristics of a managed device.

There are two types of Managed Object or Object ID:

- Scalar - managed object that always has a single instance (e. g. system description)
- Tabular - have multiple instances, such as the rows of a table (e. g. CPU with multiple cores each has individual frequency)

Design

SNMP Agent Plugin

The collectd SNMP Agent plugin registers with a SNMP Master Agent as sub agent using net-snmp library. The plugin manages MIBs provided in a configuration file where each OID is mapped to appropriate type of values that are collected by collectd read plugins. The plugin is also responsible for collecting, handling and registration of scalar and tabular values, providing data to master agent when GET/GETNEXT request is received.

Plugin configuration

The following configuration options should be supported by snmp_agent collectd plugin:

\<Table\> – Defines a table consisting of several Data blocks being its columns.

    IndexOID (optional) – OID that is handled by the plugin and is mapped to numerical index value that is generated by the plugin for each table record.

    SizeOID (optional) – OID that is handled by the plugin. Returned value is the number of records in the table.

    \<Data\> – defines table column while being inside Table block.

        \<IndexKey\> - Contains all data to build snmp table index.

            Source – This value indicates which field of corresponding collectd metric is taken as a key in SNMP table index.

            Regex (optional) – This option can be used to parse variables out of field given in Source.

            Group (optional) – Can be specified in case groups are used in Regex.

        \</IndexKey\>

        Plugin – Read plugin name whose collected data will be mapped to specified OIDs.

        Type – Collectd's type that is to be used for specified OID. Types are read from the collectd TypesDB.

        TypeInstance – Collectd's type-instance that is to be used for specified OID.

        OIDs – Configures the OIDs to be handled by plugin. Values for these OIDs are taken from collectd data type specified by **Plugin**, **Type**, **TypeInstance** fields of this block. Number of the OIDs configured should correspond to number of values in specified Type. For example two OIDs "IF-MIB::ifInOctets" "IF-MIB::ifOutOctets" can be mapped to "rx" and "tx" values of "if_octets" type.

        Scale (optional) – Values taken from collectd are multiplied by this value.

        Shift (optional) – Is added to values from collectd after they have been multiplied by Scale value.

    \</Data\>

    …

\</Table\>

…

\<Data\> – defines scalar field, must be put outside Table block

    Plugin – (see above)

    PluginInstance – Allowed only for scalars. It is used to map a field of a specific plugin instance.

    Type – (see above)

    TypeInstance – (see above)

    OIDs – (see above)

    Scale – (see above)

    Shift – (see above)

\</Data\>

Here is an example of the plugin configuration section of collectd.conf file:

```
<Plugin snmp_agent>
   <Data "memAvailReal">
      Plugin "memory"
      Type "memory"
      TypeInstance "free"
      OIDs "1.3.6.1.4.1.2021.4.6.0"
   </Data>
   <Table "ifTable">
      IndexOID "IF-MIB::ifIndex"
      SizeOID "IF-MIB::ifNumber"
      <Data "ifDescr">
         <IndexKey>
            Source "PluginInstance"
         </IndexKey>
         Plugin "interface"
         OIDs "IF-MIB::ifDescr"
      </Data>
      <Data "ifOctets">
         Plugin "interface"
         Type "if_octets"
         TypeInstance ""
         OIDs "IF-MIB::ifInOctets" "IF-MIB::ifOutOctets"
      </Data>
   </Table>
</Plugin>
```
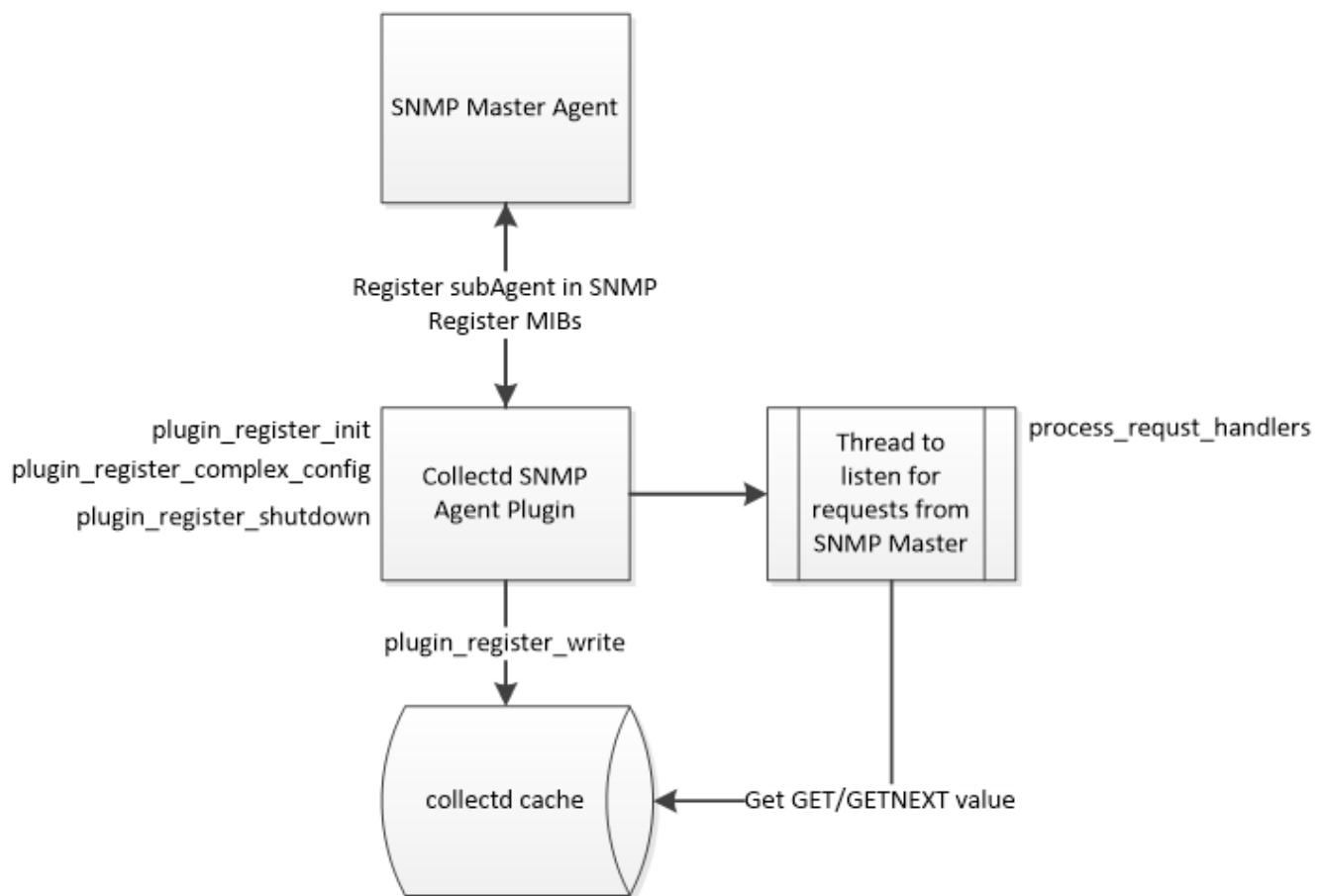
Implementation details

SNMP Master Agent

Register subAgent in SNMP
Register MIBs

plugin_register_init
plugin_register_complex_config
plugin_register_shutdown

Collectd SNMP Agent Plugin

Thread to listen for requests from SNMP Master

process_requst_handlers

plugin_register_write

collectd cache

Get GET/GETNEXT value

Figure 8. SNMP Agent plugin design.

Design description:

- SNMP Plugin registers itself on startup in SNMP Master Agent and registers scalar OIDs from the MIBs it owns: it's necessary to inform SNMP Master Agent that data requests for specified OIDs should be redirected to the SNMP Agent plugin for processing. Table OIDs will be registered at run time after new entry will be dispatched to collectd cache. In turn, plugin instance value will be the key for the table.
- SNMP plugin creates a separate thread to listen on socket for events/requests from SNMP Master Agent: checks for packets arriving on the SNMP port and processes them if some are found. Processing the GET/GETNEXT requests what means that OID or list of OID in data structure will be generated and forwarded to SNMP Agent for processing.
- Collectd cache contains metrics dispatched by the collectd read plugins. When GET/GETNEXT request is received, plugin gets data from collectd and translates requested values from collectd's internal format to SNMP format.

SNMP Master Agent handles the following:

- Listens on a socket for requests from SNMP Manager.
- Parses/forms/transmits SNMP v1, v2c, v3 messages.
- Responsible for authorization, encryption/decryption.
- Stores OIDs for all subagents running on the system to communicate with SNMP manager (*note*: in case OID is not registered, SNMP manager request will be discarded by AgentX and SNMP Agent plugin won't be notified)
- Uses Net-SNMP to communicate with SNMP sub-agents.
- Can be also represented as collector.

Considerations

Configuration Considerations

Deployment Considerations

e.g. MIB name and location.

API/GUI/CLI Considerations

Equivalence Considerations

Security Considerations

Alarms, events, statistics considerations

Redundancy Considerations

Performance Considerations

May have small impact on performance when huge amount of requests are received.

Testing Consideration

Other Considerations

Impact

The following table outlines possible impact(s) the deployment of this deliverable may have on the system.

Ref	System Impact Description	Recommendation / Comments
1		

Key Assumptions

The following assumptions apply to the scope specified in this document.

Ref	Assumption	Status
1		

Key Exclusions

The following exclusions apply to the scope discussed in this document.

Ref	Exclusion	Status
1		

Key Dependencies

The following table outlines the key dependencies associated with this deliverable.

Ref	Dependency	Status
1	Net-SNMP	
2		
3		
4		

Issues List

Ref	Issue	Status
1		