AONOS Redis database schema design

zhengweitang.zwt@alibaba-inc.com
leixin.lei@alibaba-inc.com
guixiaodong.gxd@alibaba-inc.com

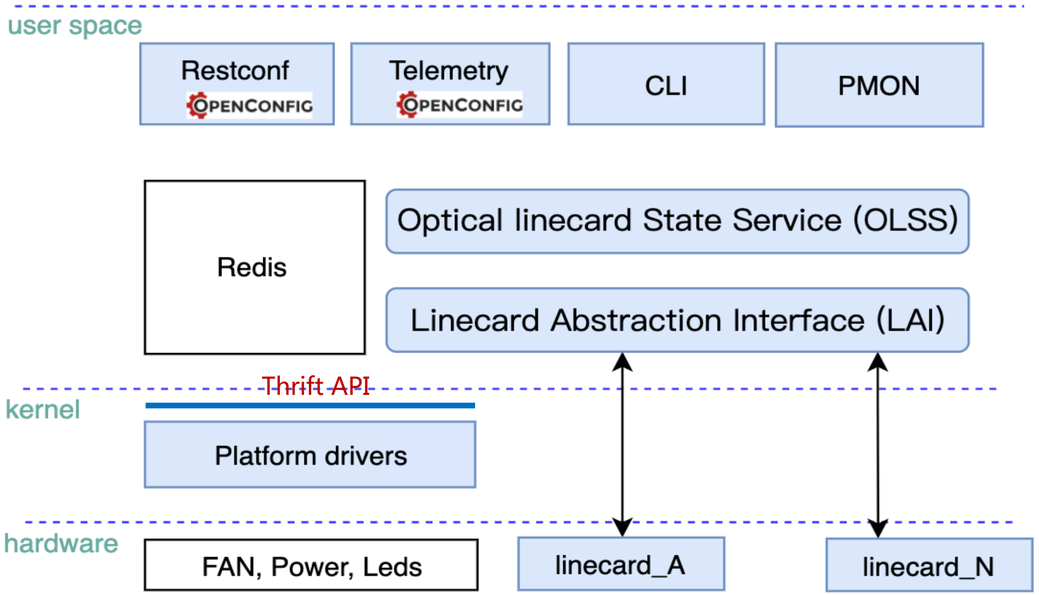
Alibaba Cloud
2023-4-1

AONOS software architecture and Redis

- LAI provides standard APIs to manage OTN linecards
- Thrift API provides standard APIs to manage peripheral devices

- PMON container manages the peripheral devices
- OLSS and Syncd contaners manage OTN linecards

- Redis container functionalities
- Save all data in different Redis containers
- Message bus for different user space applications
- Disaggregate software modules into containers



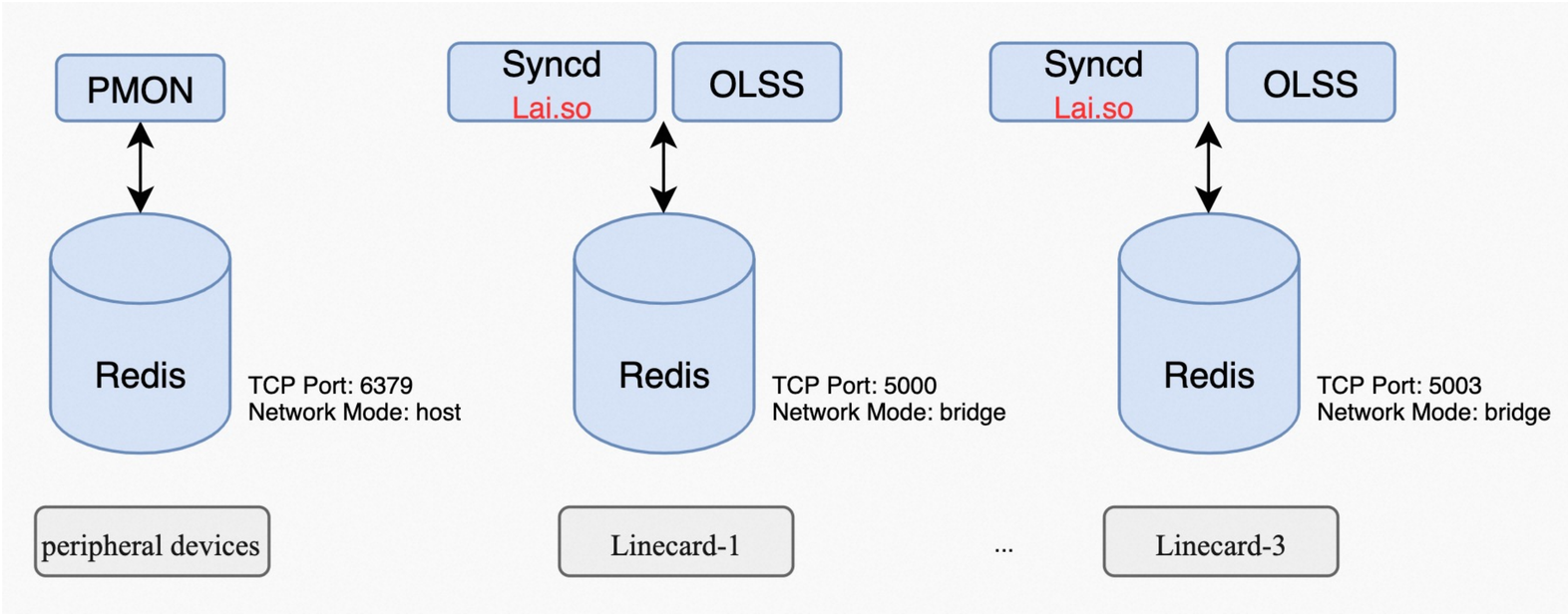
AONOS Architecture

Redis database Containers

- Enhance SONiC Multi-ASIC architecture for OTN linecards management
- a group of Redis+Syncd+OLSS containers manage one linecard
- The CU(control unit) and peripheral devices share one Redis container
- 4-linecards system includes 5(4+1) Redis containers

- These Redis database container expose different ports with different network mode.
- Each linecard’s data are isolated and independent, ensure data safety

for configuration, data backup and restore



Logical databases in a Redis container

- A Redis instance supports multiple logical databases, numbered from 0 to 15

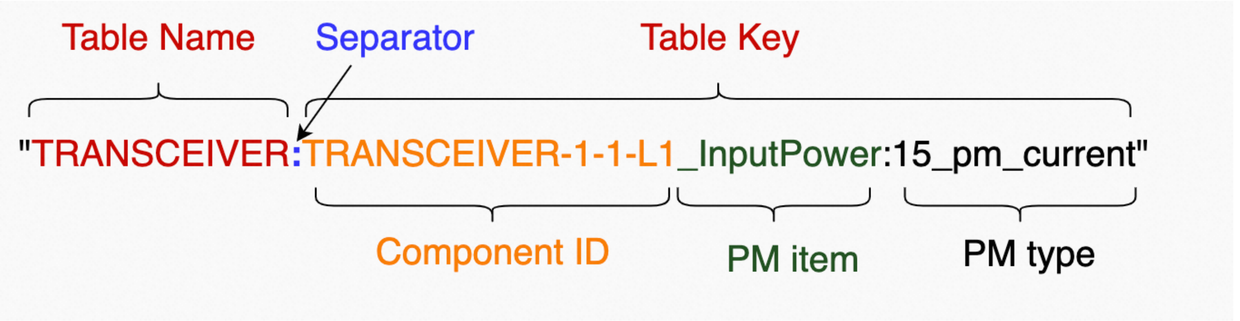
- Config database, index 4,
  save running configurations
- State database, index 6,
  save actual linecard and peripheral devices status
- Counters database, index 2,
  save current 15min, 24-hours PM data
- History database, index 10, GB_FLEX_COUNTER_DB in SONiC
  save history 15min, 24-hours PM data, history alarm and event

- Application database, index 0,
  save application data
- ASIC database (linecard database), index 1,
  save linecard LAI object state
- Flex_counter database, index 5
  save flex_counter configurations

Redis database schema design principle

- Table and key naming rule is fixed,
- field and value match OpenConfig definition,

- Make it easier to get/set data in CLI/RESTCONF/Telemetry
- Automatically map field to RESTCONF OpenConfig data structure



Config database schema

LAI Object name Component name

```
127.0.0.1:5000[4]> hgetall "TRANSCEIVER|TRANSCEIVER-1-1-L1"
 1) "index"
 2) "5"
 3) "logical-channel"
 4) "CH115"
 5) "och"
 6) "OCH-1-1-L1"
 7) "otn"
 8) "CH115"
 9) "parent"
10) "PORT-1-1-L1"
11) "port-id"
12) "1"
13) "port-type"
14) "LINE"
15) "enabled"
16) "true"
18) "vendor-expect"
29) "INNOLIGHT"
```

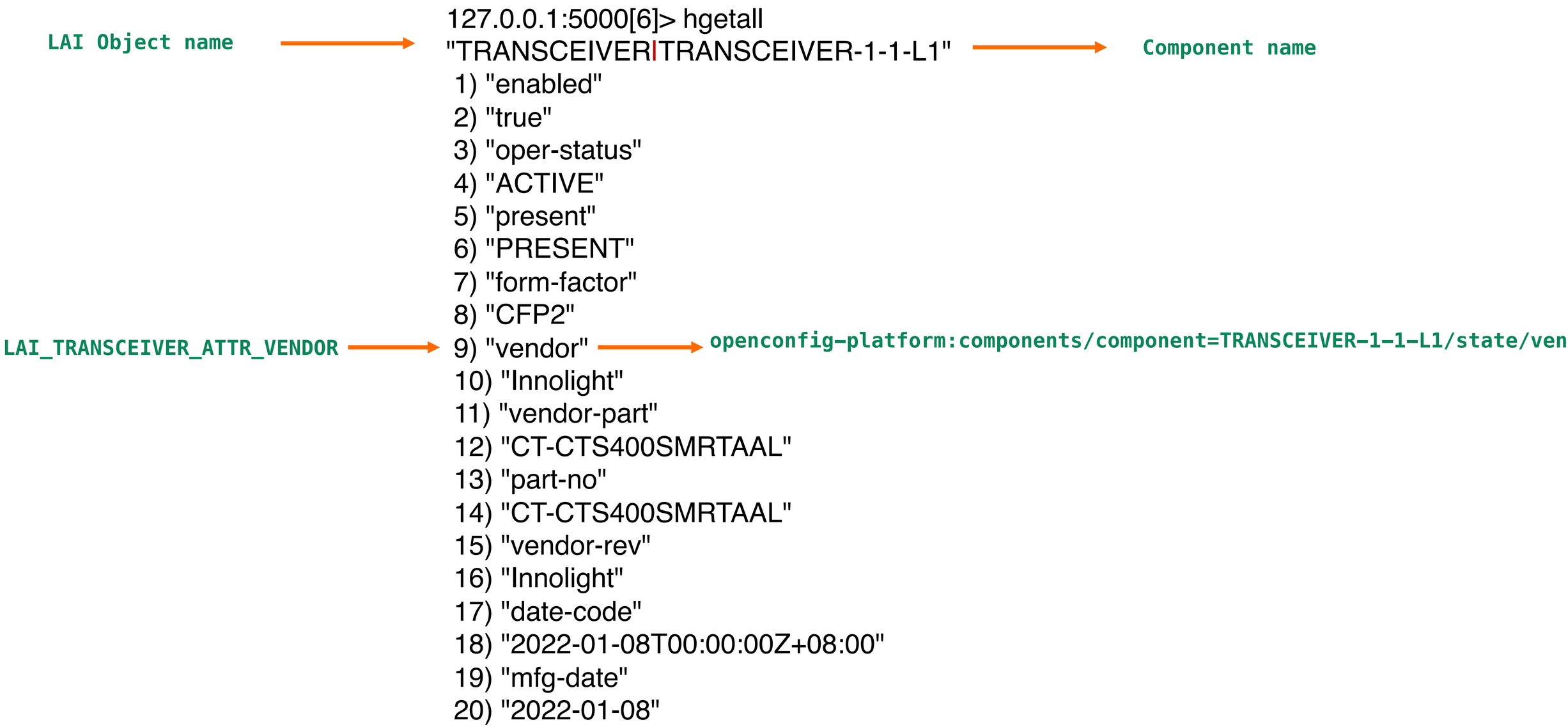
```
/**
* @brief Turns power on / off to the transceiver
*
* @type bool
* @flags CREATE_AND_SET
*/
LAI_TRANSCEIVER_ATTR_ENABLED,
```

openconfig-platform:components/component=TRANSCEIVER-1-1-L1/
openconfig-platform-transceiver:transceiver/config/enabled

```
/**
* @brief Vendor expect
*
* @type char
* @flags CREATE_AND_SET
*/
LAI_TRANSCEIVER_ATTR_VENDOR_EXPECT,
```

New field which is not defined in OpenConfig.

state database schema



state database schema

Current Alarm is in state database, it contains

- ID
- resource
- text
- type-id

LAI standardize the OTN linecard alarms

Table Name Alarm ID

```
127.0.0.1:5000[6]> hgetall "CURALARM|PORT-1-1-C3#XCVR_MISSING"
 1) "id"
 2) "PORT-1-1-C3#XCVR_MISSING"
 3) "time-created"
 4) "16667790142984189 44"
 5) "resource"
 6) "PORT-1-1-C3"
 7) "text"
 8) "TRANSCEIVER MISSING"
 9) "type-id"
10) "XCVR_MISSING"
```

Component Name

src/sonic-lairedis/LAI/inc/laitypes.h

```
typedef enum _lai_alarm_type_t
{
…
LAI_ALARM_TYPE_XCVR_MISSING,
LAI_ALARM_TYPE_XCVR_UNSUPPORTED,
LAI_ALARM_TYPE_XCVR_MISMATCH,
…
}
```

counter database schema

PM Entity →

```
127.0.0.1:5000[2]> hgetall "TRANSCEIVER:TRANSCEIVER-1-1-L1
_InputPower:15_pm_current"
 1) "starttime"
 2) "16678755000000000000"
 3) "instant"
 4) "-8.33"
 5) "avg"
 6) "-8.34"
 7) "min"
 8) "-8.35"
 9) "max"
10) "-8.32"
11) "min-time"
12) "16678755775808428 48"
13) "max-time"
14) "16678756956446359 92"
15) "validity"
16) "incomplete"
17) "interval"
18) "900000000000"
```

→ PM type

Openconfig:oc-types:avg-min-max-instant-stats-precision2-dBm

PM module in Syncd → Calculate

```
/**
* @brief Input power
*
* @type lai_double_t
* @unit dBm
* @precision precision2
* @iscounter false
*/
LAI_TRANSCEIVER_STAT_INPUT_POWER,
```

history database schema

History PM is in history database, it contains

- start time
- instant/avg/min/max/min-time/max-time
- interval
- validity

```
127.0.0.1:5001[10]> hgetall "LINECARD:LINECARD-1-2_Temperature
:15_pm_history_16764327000000000000"
 1) "starttime"
 2) "16764327000000000000"
 3) "instant"
 4) "42.2"
 5) "avg"
 6) "42.2"
 7) "min"
 8) "42.2"
 9) "max"
10) "42.2"
11) "interval"
12) "900000000000"
13) "min-time"
14) "16764327008658664960"
15) "max-time"
16) "16764327008658664960"
17) "validity"
18) "complete"
```

history database schema

History Alarm is in history database, it contains

- alarm ID
- alarm created time
- alarm cleared time
- text
- type-id

```
127.0.0.1:5000[10]> hgetall "HISALARM:PORT-1-1-C1#RX_LOS#1680585836144396032"
 1) "id"
 2) "PORT-1-1-C1#RX_LOS"
 3) "time-created"
 4) "1680585566596684032"
 5) "resource"
 6) "PORT-1-1-C1"
 7) "text"
 8) "#CH(dBm):-40.0/-40.0/-40.0/-40.0#RX-LOS"
 9) "severity"
10) "NOT_ALARMED"
11) "type-id"
12) "RX_LOS"
13) "time-cleared"
14) "1680585836144396032"
```

Thanks !

Q&A