SONiC
Robustness Monitor (RM)

Team: INV
Ashwini Udupa
Wilson Lo

創新 品質 虛心 力行
Innovation Quality Open Mind Execution
Why?

Tackle issues like SONiC system run-out-of-resource crash. e.g.: SAI_STATUS_TABLE_FULL, Orchagent(OA) crash

```
SONiC-Inventec-d7054 ERR syncd: _brcm_sai_l3_route_config:1133 L3 route add failed with error Table full (0xffffffff).

SONiC-Inventec-d7054 ERR syncd: :- processEvent: failed to execute api: create, key: SAI_OBJECT_TYPE_ROUTE_ENTRY: {"dest":"200.0.0.0/24","switch_id":"oid:0x210000000000","vr":"oid:0x3000000003a"}, status: SAI_STATUS_TABLE_FULL

SONiC-Inventec-d7054 NOTICE syncd: :- exit_and_notify: sending switch shutdown request notification to OA

SONiC-Inventec-d7054 NOTICE orchagent: :- handle_switch_shutdown_request: switch shutdown request
SONiC-Inventec-d7054 INFO supervisord: orchagent terminate called after throwing an instance of 'std::invalid_argument'

SONiC-Inventec-d7054 INFO swss.sh[7111]: <skip> INFO exited: orchagent (terminated by SIGABRT (core dumped); not expected)
```
What’s RM?

1. **Functionalities**
   a. Refresh & cache all the latest Critical Resource Monitoring (CRM) and config value ranges.
   b. Before writing to APP_DB/CONFIG_DB (or other DB requiring validation),
      - Check config value range & block if needed.
   c. Available to monitor other system values.

2. **Advantage**
   Save to write first, then delete value in DB due to setting failure.
<snip>
diff --git a/orchagent/routeorch.cpp b/orchagent/routeorch.cpp
index 18e8d61..1442d49 100644
--- a/orchagent/routeorch.cpp
+++ b/orchagent/routeorch.cpp
@@ -791,6 +791,21 @@
 bool RouteOrch::addRoute(IpPrefix ipPrefix, IpAddresses nextHops)
 */
     if (it_route == m_syncdRoutes.end())
+    if (route_entry.destination.addr_family == SAI_IP_ADDR_FAMILY_IPV4)
+    {
+        if (!gCrmOrch->isRunOutOfCrmRes(CrmResourceType::CRM_IPV4_ROUTE))
+        {
+            SWSS_LOG_ERROR("Resource ipv4_route table full. Failed to create route %s with next hop(s) %s", 
+                              ipPrefix.to_string().c_str(), nextHops.to_string().c_str());
+            /* Clean up the newly created next hop group entry */
+            if (nextHops.getSize() > 1)
+            {
+                removeNextHopGroup(nextHops);
+            }
+            return false;
+        }
    }
    route_attr.id = SAI_ROUTE_ENTRY_ATTR_NEXT_HOP_ID;
    route_attr.value.oid = next_hop_id;
<snip>
Full Code change

---
orchagent/crmorch.cpp | 51 +++++++++++++++++++++++++++++++++++++++++++++++++
orchagent/crmorch.h  | 1 +
orchagent/routeorch.cpp | 15 +++++++++++++++++++
3 files changed, 67 insertions(+)

diff --git a/orchagent/crmorch.cpp b/orchagent/crmorch.cpp
index ab6c0b7..a55919b 100644
--- a/orchagent/crmorch.cpp
+++ b/orchagent/crmorch.cpp
@@ -279,6 +279,57 @@ void CrmOrch::handleSetCommand(const string& key, const vector<FieldValueTuple>&
 }

+bool CrmOrch::isRunOutOfCrmRes(CrmResourceType resource)
+{
+    SWSS_LOG_ENTER();
+    sai_attribute_t attr;
+    uint32_t reservedCounter = 0;
+    switch (resource)
+    {
+        case CrmResourceType::CRM_IPV4_ROUTE:
+            attr.id = SAI_SWITCH_ATTR_AVAILABLE_IPV4_ROUTE_ENTRY;
+            reservedCounter = 10;
+            break;
+        default:
+            SWSS_LOG_ERROR("Failed to get CRM resource type %u. Unknown type.\n", resource);
+            return true;
+    }
+
+    sai_status_t status = sai_switch_api->get_switch_attribute(gSwitchId, 1, &attr);
+    if (status != SAI_STATUS_SUCCESS)
We are Ready!