Feature Project Proposal——Akraino Profiling

Helloway He
(helloway.wewe@gmail.com)
04/03/2019
Background - Requirement for Unified Metrics Exposition Format

- Variety of edge computing devices – How to choose the most suitable device to run the edge computing apps/services
- Various monitoring and management systems, different exposed metrics data formats of backend – Hard to pick out the optimal one from the competing products for suppliers
  - Give priority to supporting official de facto standard
  - How to seamlessly connect all other monitoring systems to gather more metrics
  - How to reduce development complexity

These questions require us to
- Have the widely accepted standard that can expose the metric data collected by the backend in a unified format
- Integrate and support the common data format to enable seamless multi-system docking - More comprehensive metrics data, better scheduling
Background – OpenMetrics Metrics Exposition Format Standard

- CNCF sandbox project launched by the Prometheus community, Prometheus is the second project graduated from CNCF after Kubernetes
- Main contributors
  - Google, Prometheus, InfluxData, SolarWinds, Open Census, Uber, Data Dog, etc.
- Aims to create a standard specifically for exposing metric data
- Uses the Prometheus exposition format as the starting point for its standard, evolve Prometheus’ metrics format and semantics into a recognized de facto standard specification
- Data model supports multi-dimensional definition
  - Metrics format - Time series data consist of the name of the metric and a series of labels (key-value pairs)
    - For example: `http_requests_total{method="POST", code="200"}`.
  - Metrics type
    - **Counter** a cumulative metric that only can increase, things like the number of requests served
    - **Gauge** a metric that can go up and down, things like temperature or current memory usage.
    - **Histogram** usually things like request durations or response sizes
    - **Summary** like Histogram, calculate quantiles

OpenMetrics Metrics Data Format
### Feature Project Proposal - Akraino Profiling

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Companies Participating / Committers</th>
<th>Requested Release / Timeline</th>
<th>Informational</th>
</tr>
</thead>
</table>
| Akraino Profiling  | Akraino Profiling to provide the exposed metrics data that are compliant with the OpenMetrics standard to unify the end-to-end metric format for edge computing scenarios.  
1. Pre-analysis    | Analyze which metrics data required to be monitored, then aggregate them into the requirement document; (Preliminarily plan to focus on IOT scenario).                                                                | Huawei ARM Intel                    | R2                                                                            | Impacted Blueprint Family – Applies to all BP Families and Blueprints  
|                    | 2. Exporter -- Collect metrics data from the monitored object and expose them to the monitoring systems in OpenMetrics standard format. (Compliant with the OpenMetrics)                                   |                                       |                              | See next slide for additional details |
|                    | Analyze which exporters need to be add/redevelop/updated according to the metrics requirement document, then develop them                                                                                     |                                       |                              |                                                                                |
|                    | 3. Adapter (optional, pluggable and dynamically loaded component) -- Convert OpenMetrics metrics data exposed by exporters to their own proprietary format that can be directly processed by those monitoring systems. (Compatible with OpenMetrics) |                                       |                              |                                                                                |
|                    | Analyze which BP monitoring systems need to develop the adapter:  
• If the BP monitoring system can't directly process the metric data which follow the OpenMetrics format standard, it needs to develop the adapter to do metric format transformation to dock with it.  
• Otherwise, there is no need to develop the adapter for the monitoring system of the blueprint that can directly process the OpenMetrics metric data. |                                       |                              |                                                                                |
|                    | 4. Integration                                                                                                                                     |                                       |                              |                                                                                |
|                    | To develop scripts based on ansible/helm to integrate Profiling into each BP, i.e., deploy the exporters and converters that BP needs, and then validate in the Akraino environment.                                |                                       |                              |                                                                                |
|                    | 5. Future plans to extend to more Akraino Blueprint Families/BPs.                                                                                   |                                       |                              |                                                                                |
Akraino Profiling Framework

- Add/redevelop/update exporters for them to collect metrics data from the monitored objects and expose in OpenMetrics format.
- **Adapter** is the optional component
  1. For those monitoring systems that are not designed for processing the OpenMetrics metric data -- Develop the corresponding adapter for them to transform OpenMetrics metric data format exposed by exporters to their own proprietary format that can be directly processed by themselves.
  2. For those monitoring systems that are designed for processing the OpenMetrics metric data -- No need to develop adapter for them as they can directly process the OpenMetrics data from exporters.

Akraino Profiling consists of:
- Requirement Doc – Required;
- Exporter – Required;
- Adapter – Optional;
- Integration – Required;

Monitored Objects (App, Container, VM, Hardware, …)

OpenMetrics metrics data

BP Monitoring System Adapter

ELIOT: Edge Lightweight and IoT Blueprint - Prometheus
Integrated Edge Cloud (IEC) Blueprint Family - Not sure
Network Cloud Blueprint - Prometheus
Time-Critical Edge Compute Blueprint - Not sure

Monitored Objects (App, Container, VM, Hardware, …)
Akraino Profiling Example 1 - ELIOT

- **Pre-analysis**
  Monitoring Metrics Data: eliot_node1_sound, eliot_node1_temperature,… The requirements document is roughly as follows.

- **Exporter**
  Add a new node exporter/update an existing node exporter to collect and expose metrics data which are compliant with OpenMetrics standard;

- **Adapter (Eliot Not required)**
  Eliot uses Prometheus as the monitoring system that can directly process the OpenMetrics metric data collected;

- **Integration**
  Develop scripts based on ansible/helm to integrate Profiling with Eliot.

---

**Monitoring Metric Data**

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Metric Type</th>
<th>Metric Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eliot_node1_sound</td>
<td>gauge</td>
<td>Sound (noise) level</td>
</tr>
<tr>
<td>eliot_node1_temperature</td>
<td>gauge</td>
<td>Temperature in C and F</td>
</tr>
</tbody>
</table>

---

**ELIOT Manager**

- Node/Resource
- Network controller
- KubeEdge
- Services
- Monitoring

**ELIOT Node**

- On-board Services
- Light Container Runtime
- Light Kubelet/KubeEdge
- Monitoring
- Profiling
- Lightweight OS/Hypervisor
- X86/ARM32/ARM64

---

**Monitoring (developer/user tools)**

Using prometheus and cadvisor to monitoring resources

**Profiling**

Add/update exporters to collect and expose the metric data to Prometheus server on the Eliot Manager in the OpenMetrics format.
Take Ganglia (on the cloud) and gmond (Ganglia collector daemon at the edge) as an example:

- **Pre-analysis**
  - Determined the monitoring metrics;
- **Exporter**
  - Redeveloped `gmond_exporter` to serve OpenMetrics metrics for ganglia on the Cloud;
- **Adapter (Ganglia required)**
  - Developed `ganglia_adapter` for ganglia to convert the OpenMetrics metric to ganglia metrics as it can't directly process the OpenMetrics metric collected by `gmond_exporter`;
- **Integration**
  - Developed scripts based on ansible/helm to integrate Profiling which includes cloud part and edge part.