ERICSSON

Performance Management Northbound API

Cloud Execution Environment

INTERWORK DESCRIPTION

4/155 19-CRA 119 1860/5 Uen B

Contents

1 Introduction

This document describes the northbound interfaces of Cloud Execution Environment (CEE) used for Performance Management (PM). There is a tool integrated into CEE, that monitors the performance, as well as availability and alarm status of the host environment. This monitoring tool extends the telemetry functionality of OpenStack, that is used to track virtual resources for billing purposes.

The monitoring tool stores Key Performance Indicators (KPIs) of the host environment that can be used for real-time processing and display, as well as for offline analysis. These two main uses are supported by two northbound interface types: real-time using REST-based interfaces and the generated PM report files mainly used for offline post processing after download.

This document describes the structure and content of the PM report file in CEE and it lists the monitoring operations.

Note: This document does not describe the measurements. For detailed information about the various performance measurements refer to *Preconfigured Key Performance Indicators*.

2 PM Report File

The PM report file is an Extensible Markup Language (XML) file produced by an external script querying the monitoring system through its API. The produced report file conforms to measurement collection data file XML schema (`measCollec.xsd`), specified in `3GPP TS 32.435`. For further information, refer to the 3GPP TS 32.435 V7.2.0 (2006-06) specification, section 4.2.2, Reference [1].

The report file contains aggregated measurement data from the system. The external script collects data from the monitoring system during a 15 minute long granularity period, then averages the data and saves the results. The granularity period is always 15 minutes long and it is aligned to whole hours. The data collection is continuous, the report files are generated every hour of every day throughout the whole year.

The report files are saved on the local disk of the active virtual Cloud Infrastructure Controller (vCIC) under the following directory and file name:
`/var/cache/pmreports/AYYYYMMDD.hhmm+0000-MMDD.hhmm.xml`
For example:
`/var/cache/pmreports/A20141107.1700+0000-1107.1715.xml`

The report files can be collected from the vCICs by using the Secure File Transfer Protocol (SFTP). All the vCICs must be checked for report files since the active role can be moved from one vCIC to another, so the report files can be stored on different vCICs.

The report files are available at the vCICs for one month, if the storage capacity of the local disks allows that. The report files are deleted earlier to free up disk space if there is not enough space for the new report files.

Scope

This document provides a detailed description of all elements in the PM report file.

2.1 XML Structure Overview

This section describes the structure of the XML file.

The root class opener of the PM report file is the `measCollecFile` element. It contains the following elements:

- `fileHeader`
- `measData`

- `fileFooter`

Figure 1 and Figure 2 show the visualized structure of the PM report XML file. The attributes of each element are not shown in this picture.

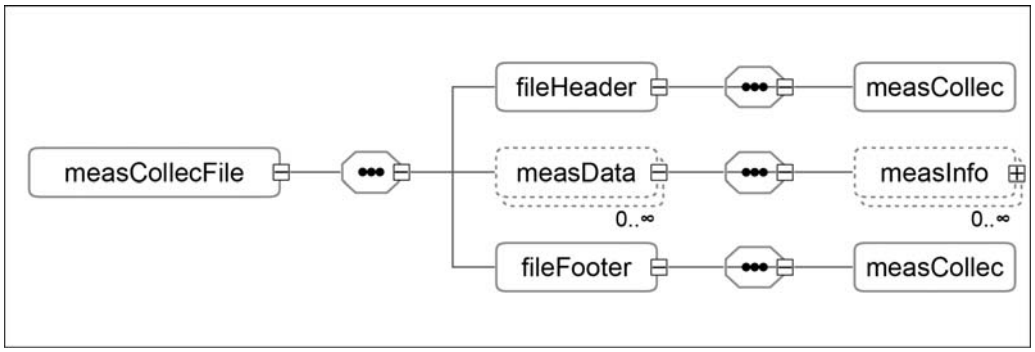

Figure 1 Element Overview

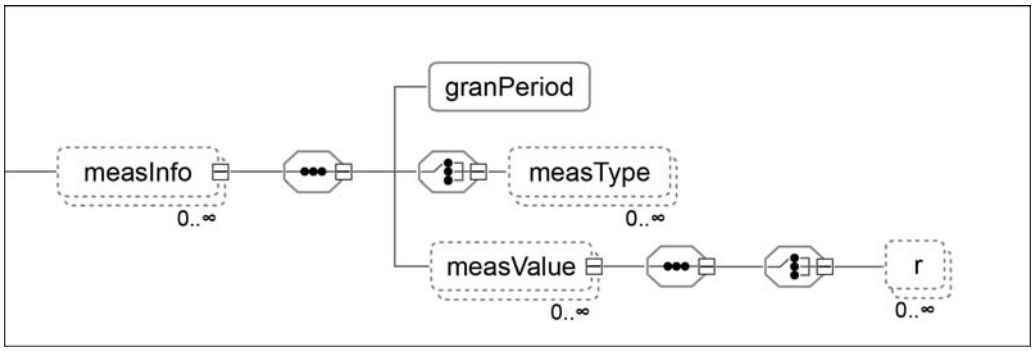

Figure 2 measInfo Element Overview

Each element and their attributes are explained in the following sections.

2.2 Element: fileHeader

The sub-elements of the `fileHeader` element contain information about the file format version number and the starting time of the measurement.

Table 1 shows a detailed description of the `fileFormatVersion` attribute of the `fileHeader` element:

Table 1 Attribute Description: fileFormatVersion

Attribute	`fileFormatVersion`
Description	The `fileFormatVersion` attribute identifies the 3GPP Technical Specification file format version and release number.
Type	`string`

Occurrence	Mandatory
Example	32.435 v7.0

Example 1 shows the `fileHeader` element and its attributes in the XML format:

```
<fileHeader fileFormatVersion="32.435 v7.0">
...
</fileHeader>
```

Example 1 fileHeader Element in XML Format

2.2.1 Element: measCollec

Table 2 shows a detailed description of the `beginTime` attribute of the `measCollec` element:

Table 2 Attribute Description: beginTime

Attribute	`beginTime`
Description	The `beginTime` attribute specifies the start date and time of the performance measurement.
Type	`dateTime`
Occurrence	Mandatory
Example	2015-01-12T08:12:10+00:00

Example 2 shows the `measCollec` element and its attributes in the XML format:

```
<fileHeader>
   ...
   <measCollec beginTime="2015-01-12T08:12:10+00:00"/>
   ...
</fileHeader>
```

Example 2 measCollec Element in XML Format

2.3 Element: measData

The `measData` element includes only one element: `measInfo`. The following sections describe the sub-elements of `measInfo` and their attributes.

2.3.1 Element: measInfo

The `measInfo` element includes the following sub-elements:

- `granPeriod`
- `measType`
- `measValue`

In the following sub-sections the elements are described with their attributes.

2.3.1.1 Element: granPeriod

This element contains information about the duration and the end time of the performance measurement.

Table 3 shows the detailed description of the `duration` attribute of the `granPeriod` element:

Table 3 Attribute Description: duration

Attribute	`duration`
Description	The `duration` attribute specifies the length of the granularity period in seconds. The granularity period is always 900 seconds (15 minutes).
Type	`duration`
Occurrence	Mandatory
Example	PT900S

Table 4 shows the detailed description of the `endTime` attribute of the `granPeriod` element:

Table 4 Attribute Description: endTime

Attribute	`endTime`
Description	The `endTime` attribute specifies the end time of the performance measurement.
Type	`dateTime`
Occurrence	Mandatory
Example	2015-01-12T08:27:10+00:00

Example 3 shows the `granPeriod` element in the XML format:

```
<measInfo>
   <granPeriod duration="PT900S" endTime=⇒
   "2015-01-12T08:27:10+00:00"/>
   ...
   ...
   ...
</measInfo>
```

Example 3 granPeriod Element in XML Format

2.3.1.2 Element: measType

Table 5 shows the detailed description of the `p` attribute of the `measType` element:

Table 5 Attribute Description: p

Attribute	`p`
Description	The `p` attribute identifies the type of the measurement. The value of this attribute matches the value of the `p` attribute of the `r` element, see Section 2.3.1.3.1 on page 7.
Type	`positiveInteger`
Occurrence	Mandatory
Example	`1`

```
<measInfo>
   ...
   <measType p="1">Processor load (15min average per core)⇒
   </measType>
   ...
   ...
   ...
</measInfo>
```

Example 4 measType Element in XML Format

2.3.1.3 Element: measValue

This element contains information of the measurement results for the resource being measured.

Table 6 shows the detailed description of the `measObjLdn` attribute:

Table 6 Attribute Description: measObjLdn

Attribute	`measObjLdn`
Description	The `measObjLdn` attribute identifies the measured resources.

Type	string
Occurrence	Mandatory
Example	node-1.domain.tld

```
<measInfo>
   ...
   ...
   ...
   <measValue measObjLdn="node-1.domain.tld">
   ...
   </measValue>
   <measValue measObjLdn="node-2.domain.tld">
   ...
   </measValue>
   <measValue measObjLdn="node-3.domain.tld">
   ...
   </measValue>
   <measValue measObjLdn="node-4.domain.tld">
   ...
   </measValue>
   <measValue measObjLdn="node-5.domain.tld">
   ...
   </measValue>
</measInfo>
```

Example 5 measValue Element in XML Format

2.3.1.3.1 Element: r

This element contains the result of the measurement. The data collected during the 15 minute long granularity period is averaged.

Table 7 shows the detailed description of the p attribute of the r element:

Table 7 Attribute Description: p

Attribute	p
Description	The p attribute identifies the type of the measurement. The value of this attribute matches the value of the p attribute of the measType element, see Section 2.3.1.2 on page 6.
Type	positiveInteger
Occurrence	Mandatory
Example	<"1">

```
<measInfo>
   ...
   ...
   ...
   <measValue measObjLdn="node-1.domain.tld">
      <r p="1">0.5050</r>
   </measValue>
   <measValue measObjLdn="node-2.domain.tld">
      <r p="1">0.6350</r>
   </measValue>
   <measValue measObjLdn="node-3.domain.tld">
      <r p="1">0.0300</r>
   </measValue>
   <measValue measObjLdn="node-4.domain.tld">
      <r p="1">0.0300</r>
   </measValue>
   <measValue measObjLdn="node-5.domain.tld">
      <r p="1">0.4500</r>
   </measValue>
</measInfo>
```

Example 6 r Element in XML Format

2.4 Element: fileFooter

The `fileFooter` element contains information about the end time of the performance measurement. It has one sub-element: the `measCollec` element.

Table 8 shows a detailed description of the `endTime` attribute of the `measCollec` element:

Table 8 Attribute Description: endTime

Attribute	`endTime`
Description	The `endTime` attribute specifies the end date and time of the performance measurement.
Type	`dateTime`
Occurrence	Mandatory
Example	2015-01-12T08:27:10+00:00

```
<ManagedElementConfiguration>


   <fileFooter>
   <measCollec endTime="2015-01-12T08:27:10+00:00"/>
   </fileFooter>
</ManagedElementConfiguration>
```

Example 7 fileFooter Element in XML Format

2.5 Example

This section shows an example of the PM report XML file.

```
<?xml version="1.0" ?>
<measCollecFile xmlns="http://www.3gpp.org/ftp/specs/archive/32_series
/32.435#measCollec" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <fileHeader fileFormatVersion="32.435 V7.0">
      <fileSender/>
      <measCollec beginTime="2015-01-12T08:12:10+00:00"/>
   </fileHeader>
   <measData>
      <managedElement/>
      <measInfo>
         <granPeriod duration="PT900S" endTime="2015-01-12T08:27:10+00:00"/>
         <measType p="1">Processor load (15 min average per core)</measType>
         <measValue measObjLdn="node-2.domain.tld">
            <r p="1">0.5050</r>
         </measValue>
         <measValue measObjLdn="node-1.domain.tld">
            <r p="1">0.6300</r>
         </measValue>
         <measValue measObjLdn="node-4.domain.tld">
            <r p="1">0.0300</r>
         </measValue>
         <measValue measObjLdn="node-3.domain.tld">
            <r p="1">0.0300</r>
         </measValue>
         <measValue measObjLdn="node-5.domain.tld">
            <r p="1">0.4500</r>
         </measValue>
      </measInfo>
   </measData>
   <fileFooter>
      <measCollec endTime="2015-01-12T08:27:10+00:00"/>
   </fileFooter>
</measCollecFile>
```

Example 8 PM Report File

3 Monitoring API

This section describes the monitoring API.

Monitoring API port: `7676`

For authentication, the PM API uses Keystone. A valid token has to be added to the request header (as `X-Auth-Token`).

The following operations are supported:

- Get node names
  See Section 3.1 on page 10.
- Get meter names
  See Section 3.2 on page 11.
- Get item value
  See Section 3.3 on page 11.

A Keystone token is required, for more information refer to the authentication section in the *OpenStack API Complete Reference*. The service URL can be checked in several ways, for example:

- Fields `internalURL` and `publicURL` in the output of `openstack catalog show pmapi` or in section `access/serviceCatalog/*/name=pmapi` of the Keystone token response
- Sections `bind` in `/etc/haproxy/conf.d/211-pmapi.cfg` on the vCICs

3.1 Get Node Names

Method: `GET`

URL: `/pm/nodes`

Description: Retrieve all node names

```
curl "http://192.168.2.21:7676/pm/nodes" -X GET -H ⇒
"X-Auth-Token: $token" | python -m json.tool

{"nodes": [
        {"node": "OpenStackCluster"},
        {"node": "compute-0-2.domain.tld"},
        {"node": "cic-0-1.domain.tld"}
]}
```

Example 9 Get node names

3.2 Get Meter Names

Method: `GET`

URL: `/pm/<node_name>/items`

Description: Retrieve all known meter names on host *<node_name>*

```
curl "http://192.168.2.21:7676/pm/compute-0-2.domain.tld/items" ⇒
-X GET -H "X-Auth-Token: $token" | python -m json.tool

{"meters": [
        {"name": "Processor load (5 min average per core)"},
        {"name": "CPU idle time"},
        {"name": "Free memory"},
        {"name": "Free disk space on /var/log"},
        {"name": "Percentage of space used on disk /var/log"},
        ...
]}
```

Example 10 Get meter names

3.3 Get Item Value

Method: `GET`

URL: `/pm/<node_name>/<item>`

Description: Retrieve the actual value of the `item` on host *<node_name>*

Optional parameters:

- `start`

  When the `start` parameter is used, all of the measurements are returned since `start_date`.

- `end`

When the `end` parameter is used, all of the measurements are returned from `start_date` until `end_date`, if `start_date` was defined.

The `start` and `end` parameters must be defined in the following format:

`YYYYmmddHHMM`

For example:

`201502161722`

```
encoded_item=$(python -c "import urllib; print urllib.quote('Percentage ⇒
of space used on disk /var/log','')")

curl "http://192.168.2.21:7676/pm/compute-0-2.domain.tld/${encoded_item}" ⇒
-X GET -H "X-Auth-Token: $token" | python -m json.tool

{"clock": "1462536645",
"name": "Percentage of space used on disk /var/log",
"value": "1.1164"}
```

Example 11 Get item value

```
encoded_item=$(python -c "import urllib; print urllib.quote('Percentage ⇒
of space used on disk /var/log','')")

curl "http://192.168.2.21:7676/pm/compute-0-2.domain.tld/${encoded_item}?⇒
start=201605040000&end=201606040000" -X GET -H "X-Auth-Token: $token" | ⇒
python -m json.tool

{"history": [
        {"clock": "1462320126",
        "name": "Percentage of space used on disk /var/log",
        "value": "0.9665"},
        {"clock": "1462320207",
        "name": "Percentage of space used on disk /var/log",
        "value": "0.9665"},
        {"clock": "1462320291",
        "name": "Percentage of space used on disk /var/log",
        "value": "0.9665"},
        ...
]}
```

Example 12 Get item history

Reference List

[1] *3GPP TS 32.435 V7.2.0 (2006-06): 3GPP TS; Telecommunication management; Performance measurement; eXtensible Markup Language (XML) file format definition (Release 7), section 4.2.2* , http://www.3gpp.org/DynaReport/32435.htm