Cisco ACI API

APIC CPU 사용률 어디까지 알아봤니?

2023. 7

Cisco ACI API

Approach for APIC CPU Util

2023. 7

YAME! ACI API series

100) Introduction – API inspector
110) Introduction – ACItoolkit, cobra
120) APIC CPU Utilization
130) Endpoint Tracker
140) APIC Port Binding
150) APIC vPC Binding
160) MSO Port Binding
170) Log

APIC CPU utilization

- From the configuration guide,

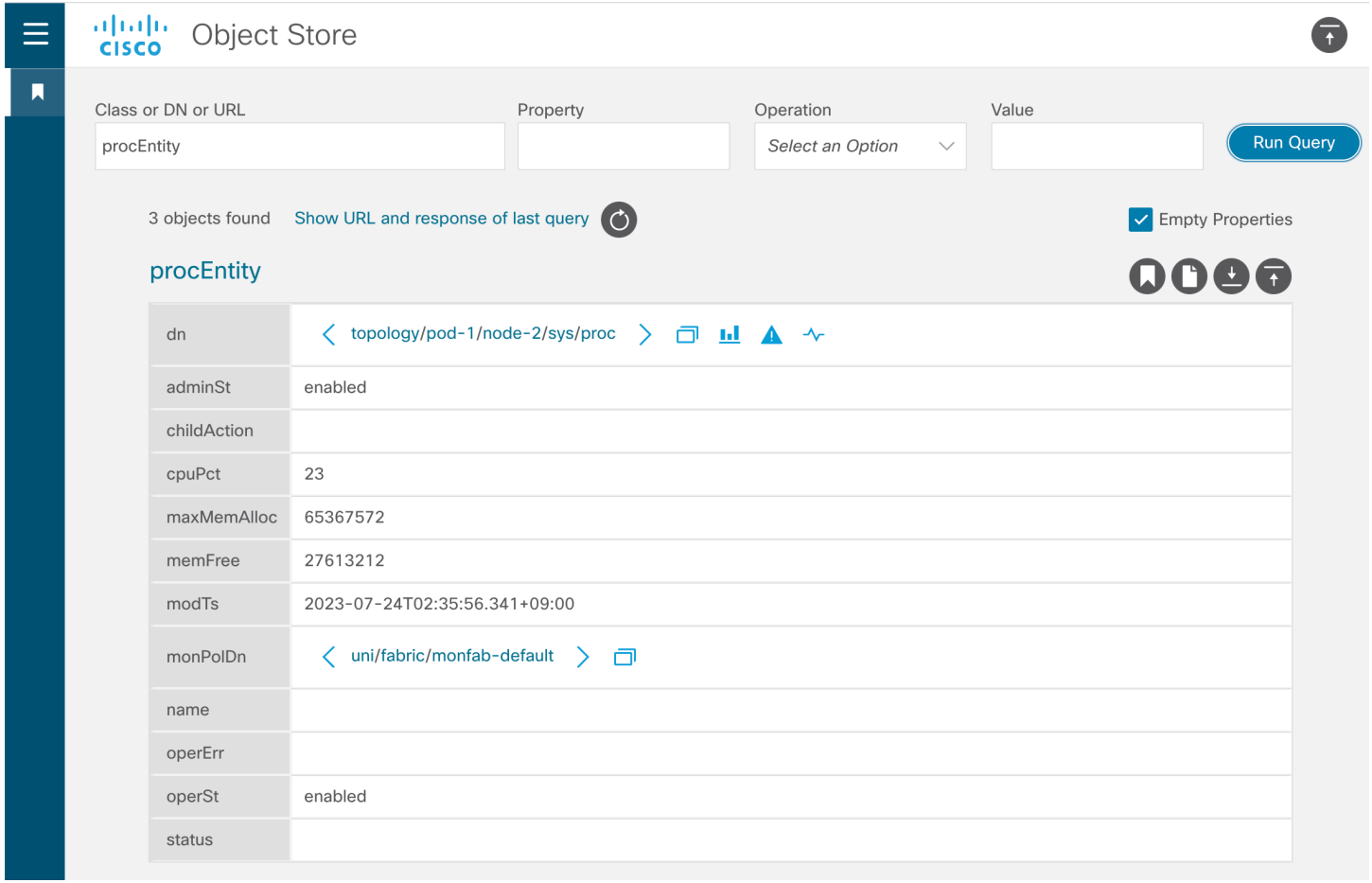
  `https://apic-ip-address/api/node/class/procEntity.xml?`

- Run Code 121-APIC_CPU.py
  - `2023-07-22 15:29:06.128500 ['24', '21', '16'] ['node-1', 'node-2', 'node-3'] 3`

- This is only way to get the CPU utilization.
  - No SNMP OID support at CIMC and APIC

APIC object store

https://APIC_IP_address/visore.html



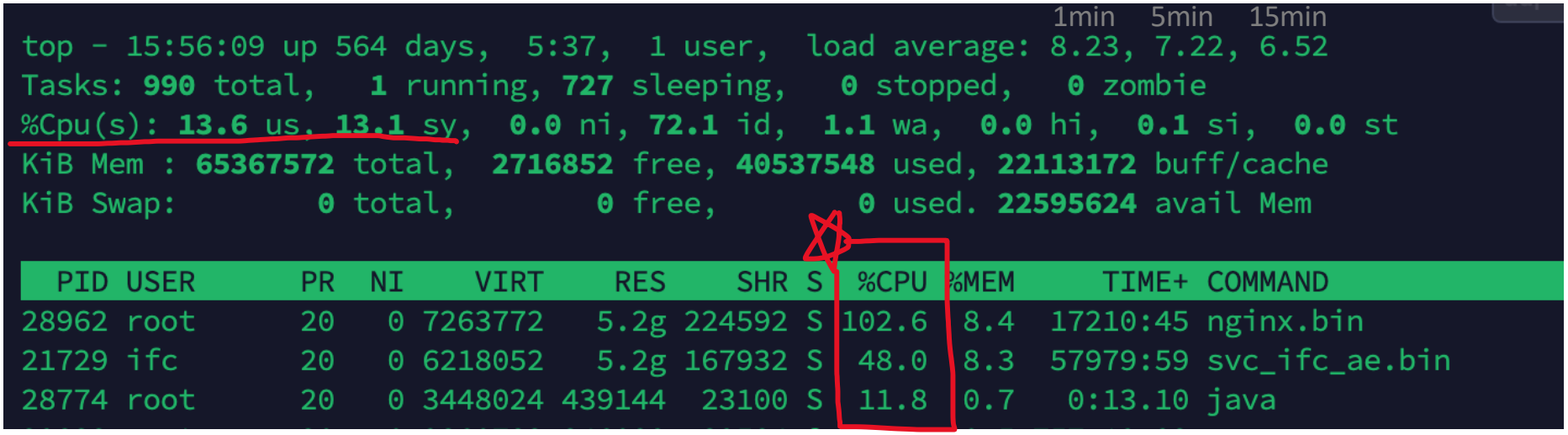
procEntity.json result Example

Looks like <per APIC>. One APIC has two CPU sockets and one CPU has 6 cores ➔ eqptCPU.json


```
{
    "procEntity": {
        "attributes": {
            "adminSt": "enabled",
            "childAction": "",
            "cpuPct": "32",
            "dn": "topology/pod-1/node-1/sys/proc",
            "maxMemAlloc": "65367572",
            "memFree": "23826972",
            "modTs": "2022-12-08T07:52:13.263+09:00",
            "monPolDn": "uni/fabric/monfab-default",
            "name": "",
            "operErr": "",
            "operSt": "enabled",
            "status": ""
        }
    }
},
```

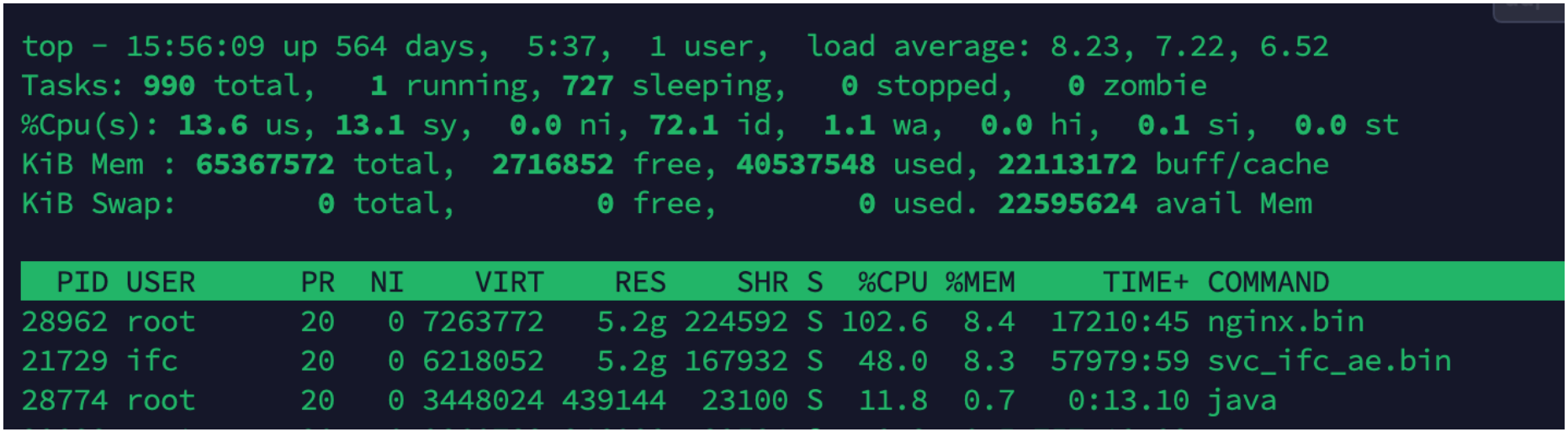
TOP

CPU = us (user level) + sy (system level)
id(le)



Limitation of <cpuPct of procEntity.json>

```
2023-07-22 15:54:51.147720 ['31', '29', '25'] ['node-1', 'node-2', 'node-3'] 3
```



Slow GUI

https://www.cisco.com/c/en/us/support/docs/cloud-systems-management/application-policy-infrastructure-controller-apic/217879-troubleshoot-a-slow-apic-gui.html

- If NGINX is congested, the API is congested.

procCPU5min.json or procApplicationCPU5min.json ???

- Not a solution.

<top> command at APIC ssh

```
 2023-07-24 02:12:20.080140
50.0 svc_ifc_ae+
44.4 svc_ifc_ae+
```

129_over_30_CPU.py

```
DC1-APIC1# top -b -n 1
top - 01:56:52 up 565 days, 15:38,  1 user,  load average: 7.64, 7.49, 6.81
Tasks: 994 total,   2 running, 727 sleeping,   0 stopped,   0 zombie
%Cpu(s):  4.3 us, 14.6 sy,  0.0 ni, 81.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 65367572 total,  3323980 free, 39919460 used, 22124132 buff/cache
KiB Swap:        0 total,        0 free,        0 used. 23242544 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
21729 ifc       20   0 6263372   5.2g 167932 S  68.4  8.4  58964:11 svc_ifc_ae.bin
 7736 admin     20   0  121328   9788   7592 R  21.1  0.0   0:00.10 top
 7738 root      20   0       4      4      0 R  21.1  0.0   0:00.04 svc_ifc_ae.bin
29822 root      20   0 9565792 351908  39612 S  10.5  0.5 977:25.98 sage
   47 root      20   0       0      0      0 S   5.3  0.0 173:09.77 ksoftirqd/6
 1340 root      20   0 1634732  17016      0 S   5.3  0.0 385:45.86 nomad
```

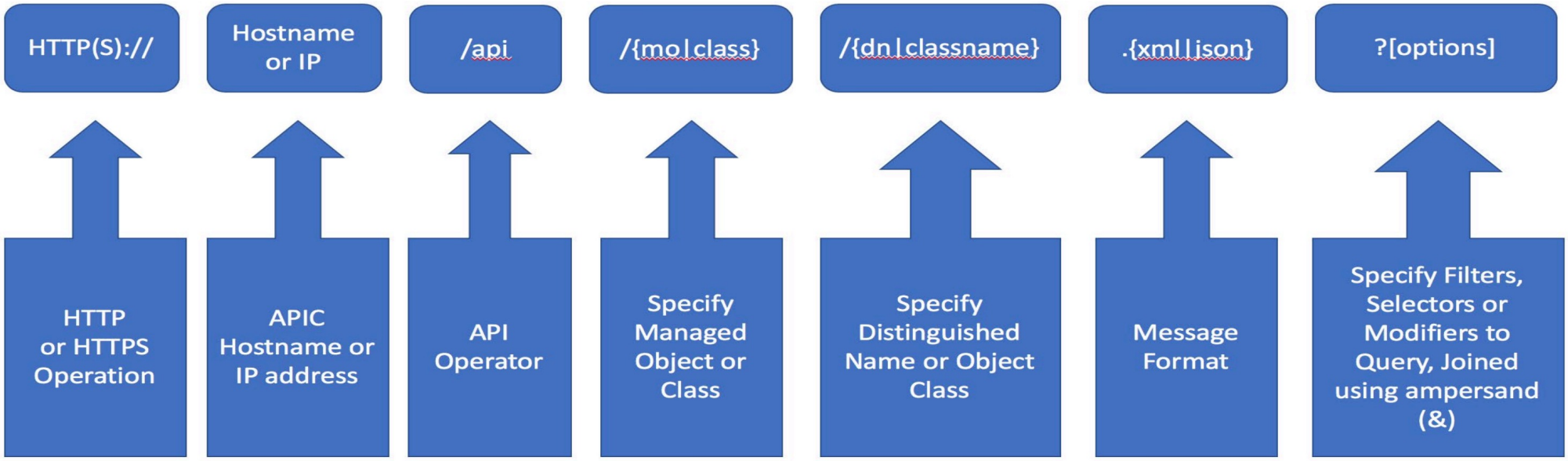
Monitoring ACI - Cisco Live

https://www.ciscolive.com/c/dam/r/ciscolive/emea/docs/2020/pdf/BRKACI-2271.pdf

Page. 89 REST API

Monitoring Using REST API

- Monitoring ACI using REST is a pull method where data is retrieved from APIC
- Requires read-only GET operation
- URL Format for a typical REST call



Monitoring Using REST API

- Resource Path changes with the type of MO or Object Class query
  - For Policy Based Managed Object MO such as Tenant, App Prof, EPG etc. query

    /api/mo/uni/tn-*tenant_name*/.....

  - For Node Based Managed Object MO such as Node Chassis, LC, FM, Interface etc. query

    /api/mo/topology/pod-*number*/node-*number*/sys/.....

  - Retrieving an Object Class information

    /api/class/....

Monitoring Using REST API

- REST offers wide range of filtering options to narrow down the query

Filter Type	Syntax	Cobra Query Property	Description
query-target	{self | children | subtree}	AbstractQuery.queryTarget	Define the scope of a query
target-subtree-class	*class name*	AbstractQuery.classFilter	Respond-only elements including the specified class
query-target-filter	*filter expressions*	AbstractQuery.propFilter	Respond-only elements matching conditions
rsp-subtree	{no | children | full}	AbstractQuery.subtree	Specifies child object level included in the response
rsp-subtree-class	*class name*	AbstractQuery.subtreeClassFilter	Respond only specified classes
rsp-subtree-filter	*filter expressions*	AbstractQuery.subtreePropFilter	Respond only classes matching conditions
rsp-subtree-include	{faults | health :stats : ...}	AbstractQuery.subtreeInclude	Request additional objects
order-by	*classname.property* | {asc | desc}	Not Implemented	Sort the response based on the property values

Monitoring Using REST API

- REST Filters can be applied in URI with a (?) symbol.
- Multiple Filters can be joined with (&) symbol

Example:

GET → https://{{apic-host-or-ip}}/api/mo/*topology/pod-1/node-1001/sys*.json?query-target=children&target-subtree-class=*eqptCh*

For more information :

https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/4-x/rest-api-config/Cisco-APIC-REST-API-Configuration-Guide-401/Cisco-APIC-REST-API-Configuration-Guide-401_chapter_0110.pdf

Important REST Queries (APIC)

- Monitoring CPU and Memory

  GET → https://{{apic-host-or-ip}}/api/class/procEntity.json

- Monitoring Disk Utilization

  GET → https://{{apic-host-or-ip}}/api/class/eqptStorage.json

- Monitoring Interfaces

  GET → https://{{apic-host-or-ip}}/api/mo/topology/pod-1/node-1/sys.json?query-target=subtree&target-subtree-class=l3EncRtdIf

- Monitoring APIC Cluster State

  GET → https://{{apic-host-or-ip}}/api/mo/topology/pod-1/node-1/sys.json