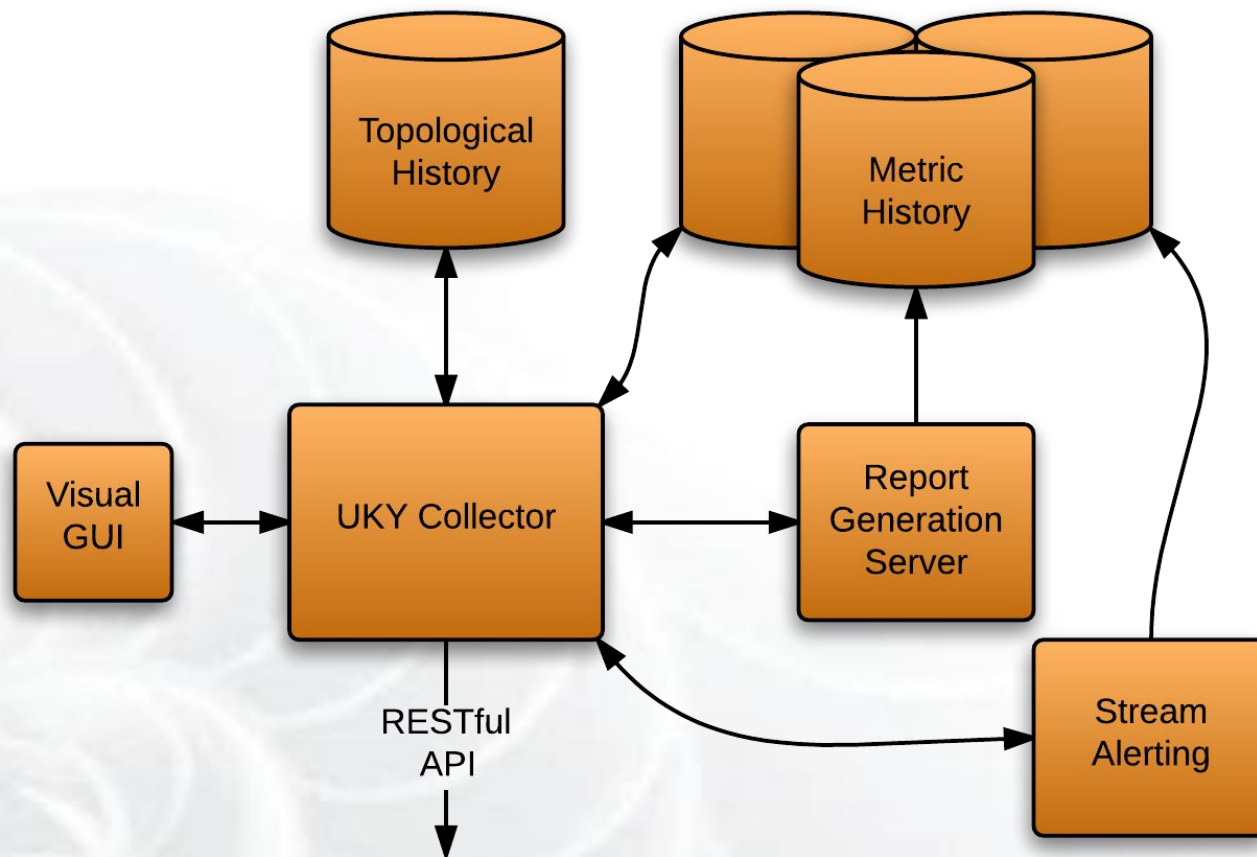


GEC24 Monitoring Support for Experimenters and Developers

Caylin Hickey
Cody Bumgardner

- What's new:
 - Reporting
 - Aggregate Status Page
 - API
- Future Options
- Q&A
- Walkthrough



Reports provide access to aggregated historical metric information as well as a way to generate a breakdown of recent aggregate outages over a given time period.

Generate New Report

Report Type

Report Target

Report Range

Aggregate Outages [By Aggregate]

Compute Resources Provisioned [By Aggregate]

Interface Network Throughput [By Aggregate]

Network Throughput [By Slice]

Network Throughput [By User]

Slice Count [By User]

Slices Provisioned [By User]

Sliver Count [By Aggregate]

Slivers Provisioned [By Aggregate]

Slivers with Compute Resources [By Aggregate]

Users with Slivers [By Aggregate]

VLAN Network Throughput [By Aggregate]

Once a report has been generated, the averages or totals (depending on the report type) are broken down and displayed based on subgroups based on the report type. The averages / totals for the report are also listed to help give a better idea of the overall usage / events represented in the report.

Details -

Name	VLAN Network Throughput for Aggregate [al2s.internet2.edu]
Type	vlan-throughput-aggregate
Start	Feb 24 2016 15:45:56
End	Mar 02 2016 15:45:56
Created On	Mar 02 2016 15:46:08
Completed On	Mar 02 2016 15:47:09
Averages	
Rxbps	1.3 Mbps
Txbps	1.6 Mbps

sdn-sw.alba.net.internet2.edu:xe-5/3/2.0 -

Averages

rxbps 12 bps

Show entries Search:

ID	↑↓	rxbps	↑↓	rxpps	↓↑	txpps	↑↓	avgnoc	↑↓	txdps	↑↓	avgfqy	↑↓
+ sdn-sw.alba.net.internet2.edu:xe-5/3/2.0:1085		12		0		0		0		0		0	

Showing 1 to 1 of 1 entries Previous **1** Next

The Aggregate Status Page displays comprehensive historical information for aggregates known to the monitoring system. The motivation behind this page is to give experimenters a way, through sorting and filtering, to make decisions for which aggregate to use for their experiments based on the information collected.

Currently, the page supports sorting on:

- Cumulative (Advertised) VM Count
- Aggregate management type
- Aggregate Name

and filtering based on

- Aggregate management type
- Stitching availability

Aggregate Status

Aggregate List [used VMs]

Sorted By:

Aggregate by [VM Count]

uwashington-ig [1 VM]
umich-ig [2 VMs]
rutgers-ig [15 VMs]
nps-ig [16 VMs]
umkc-ig [17 VMs]
nysernet-ig [25 VMs]
utc-ig [40 VMs]
wisconsin-ig [52 VMs]
uky-ig [60 VMs]
max-ig [65 VMs]

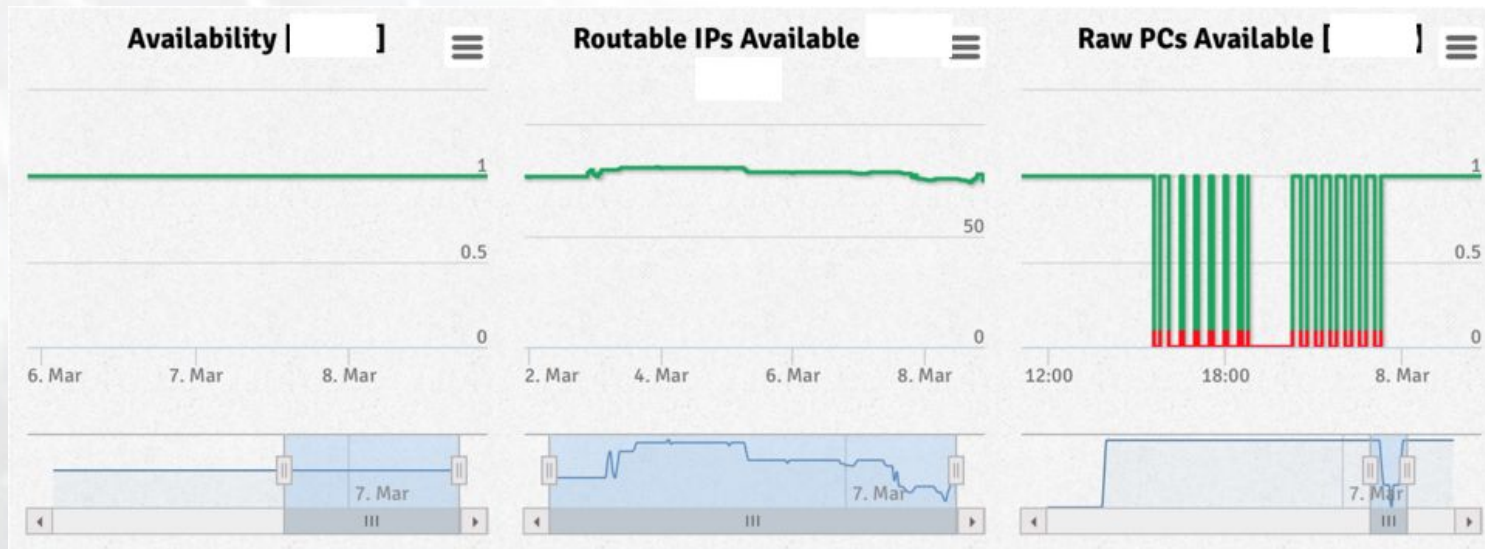
List Options:

- Display InstaGENI sites
- Display ExoGENI sites
- Display OpenFlow sites
- Display only Stitching sites

Show Status of All Listed Aggregates

Aggregate Status Page

The page initially loads sorted by VM count and displays the first five entries in the list to the user. This provides a small sample of likely candidates to choose for launching an experiment. Each aggregate window is supplemented by information about the aggregates recent history, as well as node metrics associated with said aggregate, in order to further assist in making a decision.



Exposure of GENI topological information and historical metric data accessible through a RESTful API architecture

- Uniform request schema submitted as a JSON object
- Two targets for requests
 - /API/info
 - /API/data
- Current initial draft of documentation is posted to:
<http://groups.geni.net/geni/wiki/GENIMonitoring/API>

Requests are sent in one of two ways:

- [GET] /API/<type>?q={request}
- [POST] /API/<type>

The same request schema can be used for both types of responses with only minor changes in included parameters.

- Schema defined at <https://genimon.uky.edu/schema/request>
- General parameters include:
 - **obj** : JSON array describing requested objects
 - **id** : List of id's (either ID or URN) of objects to return
 - **type** : Type of object being requested (eg: aggregate, node)
 - **output** : (*xml* | *json*) style of output (currently JSON and xml supported)

Returns information from the topological history captured by the Collector through the process of mapping via the JSON

- /API/info request parameters:
 - **infoType** : (*simple* | *detailed*) detailed displays information about child objects (optional, defaults to detailed for small sized requests by ID or simple for all other requests)
 - **ts** : JSON object detailing timestamp(s) for request
 - **ts** : Timestamp for historical /API/info requests (optional: if omitted, defaults to current information)

Example:

[https://genimon.uky.edu/API/info/?q={"obj":{"id":\["pks2.sdn.uky.edu_node_pc1"\],"type":"node"},"output":"xml"}](https://genimon.uky.edu/API/info/?q={)

```
<response>
  <request>
    <url>
      genimon.uky.edu/API/info/?q={%22obj%22:{%22id%22:[%22pks2.sdn.uky.edu_node_pc1%22],%22type%22:%22node%22},%22output%22:%22xml%22}
    </url>
  </request>
  <objects>
    <object>
      <collector_specific_info>
        <internal_id>4cbd4cac-0ace-412b-988d-d31fefe6b7b0</internal_id>
        <last_info_update>1457476512000</last_info_update>
        <last_checked_for_metrics>1457480793000</last_checked_for_metrics>
      </collector_specific_info>
      <urn>urn:publicid:IDN+pks2.sdn.uky.edu+node+pc1</urn>
      <id>pks2.sdn.uky.edu_node_pc1</id>
      <aggregate>
        <collector_internal_id>00407cac-6144-46d0-aa8d-ecf347e300be</collector_internal_id>
        <id>ukyyps2-ig</id>
        <urn>urn:publicid:IDN+pks2.sdn.uky.edu+authority+cm</urn>
        <selfRef>
          https://pks2.sdn.uky.edu:5001/info/aggregate/ukyyps2-ig
        </selfRef>
      </aggregate>
      <selfRef>
        https://www.pks2.sdn.uky.edu:5001/info/node/pks2.sdn.uky.edu_node_pc1
      </selfRef>
      <nodeType>Server</nodeType>
      <memTotalKB>49310720</memTotalKB>
      <virtualizationType>Xen</virtualizationType>
      <schemaVersion>
        http://www.gpolab.bbn.com/monitoring/schema/20140828/node#
      </schemaVersion>
      <ts>1457476507000</ts>
      <interfaces>
        ...
      </interfaces>
      <slivers/>
    </object>
  </objects>
</response>
```

Returns information from the metric measurement storage collected by polling the /data endpoints in the GENI monitoring framework

- /API/data request parameters:
 - **eventType** : JSON array of metric measurement types requested from objects (A list will be appended to the API documentation page shortly)
 - **ts** : JSON object detailing timestamp(s) for request
 - **gte** : Start of range for metric data (milliseconds)
 - **lte** : End of range for metric data (milliseconds)

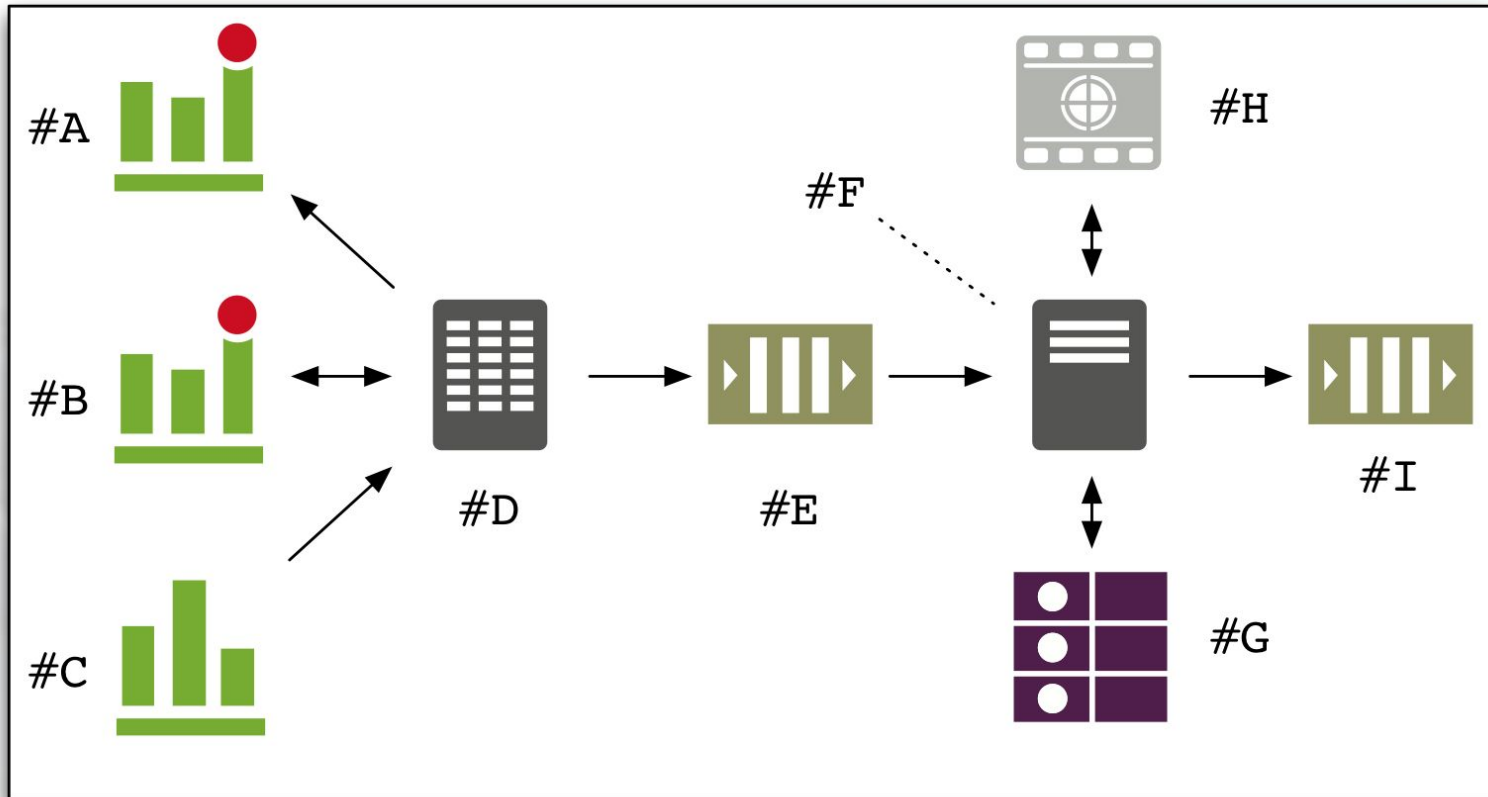
Example:

[https://genimon.uky.edu/API/data/?q={"obj":{"id":\["pks2.sdn.uky.edu_node_pc1"\],"type":"node"},"output":"json","ts":{"gte":1456260153962,"lt":1456263753962}}](https://genimon.uky.edu/API/data/?q={)

```
[
  [
    {
      "$schema": "http://www.gpolab.bbn.com/monitoring/schema/20151105/data#",
      "description": "ops_monitoring:swap_free for pks2.sdn.uky.edu_node_pc1 of type node",
      "eventType": "ops_monitoring:swap_free",
      "id": "swap_free:pks2.sdn.uky.edu_node_pc1",
      "subject": {
        "href": "http://www.gpolab.bbn.com/monitoring/schema/20140828/node#",
        "id": "pks2.sdn.uky.edu_node_pc1",
        "urn": "urn:publicid:IDN+pks2.sdn.uky.edu+node+pc1"
      },
      "tsdata": [ {
        "ts": 1456263285852,
        "v": 78.2
      }, {
        "ts": 1456263013211,
        "v": 78.2
      },
      ...
    ]
  ]
]
```

- Currently the API is open like the Aggregate Status page with user information omitted from the results. This is due to an upcoming overhaul of the GENI Monitoring GUI authentication system to come inline with other federated tools to allow easier access for users.
- Allow options for further depth of returned info from info requests (ie: children of children of children, parent of parent)
- Define custom request types

- Expose published datastream via API.
- Expand alerts system to include user-defined event types.
- Move from central polling to distributed polling.
- Monitoring agents on hosts
 - System-level monitoring
 - Network-level testing/verification
 - GENI software monitoring.



#A: Pull Data
 #B: Push Data
 #C: Metering Data
 #D: Collection aggregator
 #E: SpecX data queue

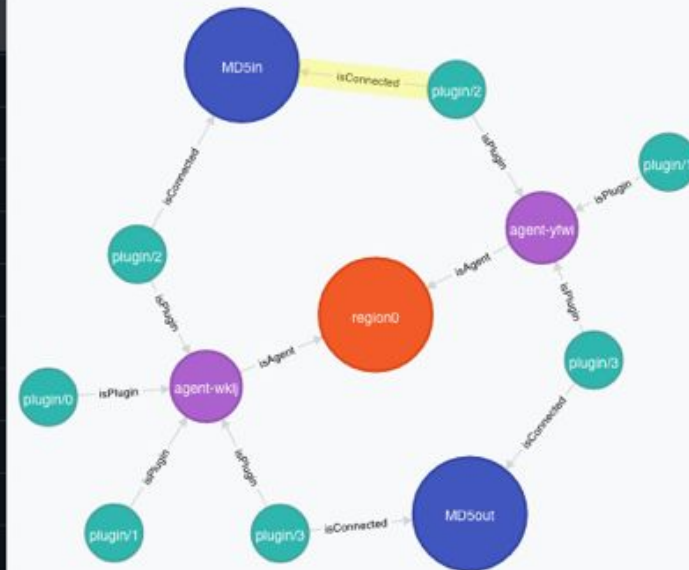
#F: State Controller
 #G: Status Key-Value store
 #H: State API Interface
 #I: State/Utilization queue


```

"region": "region-myvey",
"agent": "agent-hemt",
"cpu-summary": "Intel(R) Xeon(R) CPU X5650 @ 2.67GHz",
"cpu-ident": "Intel64 Family 6 Model 44 Stepping 2",
"cpu-core-count": "12",
"cpu-idle-load": "99.83",
"memory-available": "45007228928",
"nic-0-ip": "10.163.200.30",
"fs-0-available": "191488241664",
"sys-uptime": "149 days, 00:12:58",
"sys-os": "GNU/Linux Ubuntu 14.04,
...

```

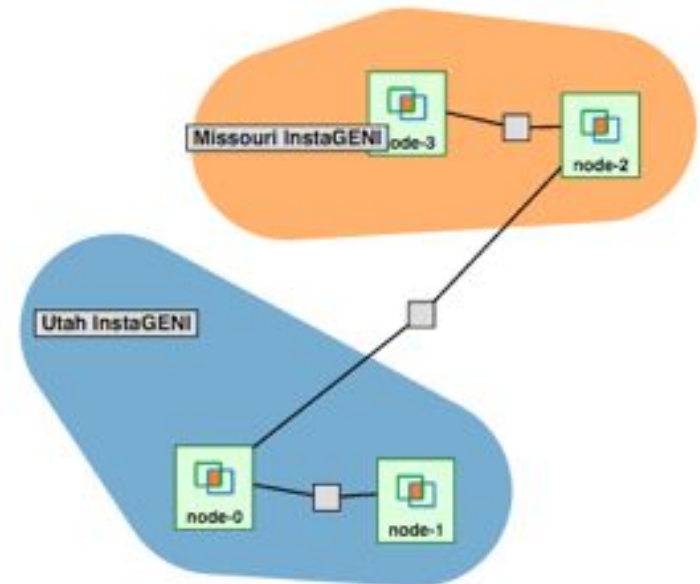
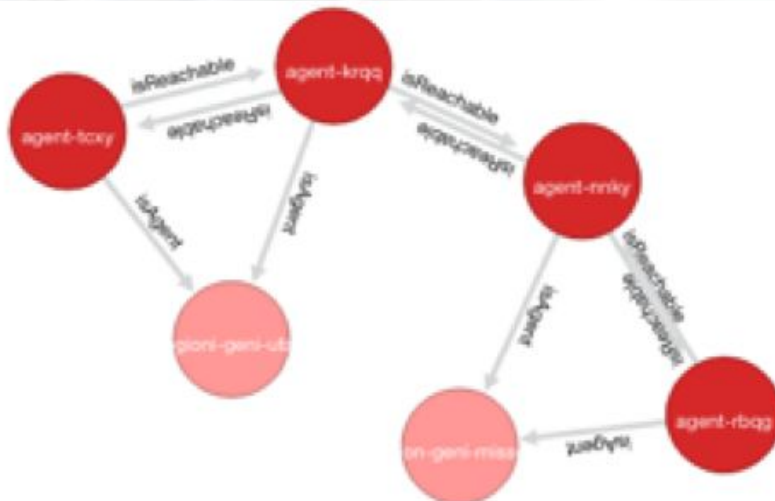
isConnected [32]	
Properties	
incomingCount	26
outgoingCount	0
runtime	26002
perfout	0.0
src_region	region0
src_plugin	plugin/2
perfin	1.0
application	MD5in
loop	1
isGlobal	true
src_agent	agent-yfwi
timestamp	1424618518703



```

"region": "regioni-geni-utah",
"agent": "agent-wkeg",
"plugin": "plugin/3",
"discoveredagents": "region-geni-missouri_agent-zlmm,regioni-geni-utah_agent-dbof",
"regioni-geni-utah_agent-dbof": "10.10.1.2_10.10.1.1",
"region-geni-missouri_agent-zlmm": "10.10.2.2_10.10.2.1",
"isinternet": "true",
"runtime": "868865003",
"timestamp": "1448305379109",

```



[region2]
[agent0]
platform = GENI
environment = Virtual
location = utah-ig
resources = cores:8,ram:16,...
[plugin/0]
pluginname = benchmark
b-cpu = 10
b-net_region0_agent0 = 50
b-net_region1_agent0 = 10
...

[region3]
[agent0]
platform = GENI
environment = Virtual
location = uk-ig
resources = cores:1,ram:1,...
[plugin/0]
pluginname = benchmark
b-cpu = 2
b-net_region0_agent0 = 10
b-net_region1_agent0 = 1
...

Offer:
[region2][agent0]
b-cpu = 10
b-net_region0_agent0 = 50
b-net_region1_agent0 = 10

Workload Evaluations:
inode_id = 1001

Optimal Match:
b-cpu-eval = 2
b-net_region0_agent0 = 10
b-net_region1_agent0 = 1

Accept: [region3][agent0]

Offer:
[region3][agent0]
b-cpu = 2
b-net_region0_agent0 = 10
b-net_region1_agent0 = 1

Questions?

-Raise hand if you want an experimenter account and don't have one.

URL: <https://genimon.uky.edu>