GENI Monitoring Meeting

Stéphane Blais and David P. Wiggins, GPO

GEC21, October 22, 2014

At the GPO Office, Hoosier Room, IMU Main Level, 4-5:30 PM

Agenda

- Discussion of existing monitoring data
  - Inter-aggregate links
  - Stats for interfacevlans
  - Alerting with messy data
- Possible new development
  - Database schema changes
  - JSON schema changes
  - Other new features

Discussion of existing monitoring data

- Completeness of the data
  - Need to be able to show the end-to-end path between two different racks
  - Attribute traffic back to users
- Representation of inter-aggregate links
- Removing stale or obsolete data from the database
- Alerting in the presence of absent, stale, or incorrect data
- Providing statistics for interfacevlans

Possible Database Schema Changes

Using the ops_interfacevlan table as a representative example:

Field	Change?
$schema	Remove. rest_call_handler should control this.
id	Remove? urn is good enough?
selfRef	Remove. rest_call_handler can fill this in.
urn	Leave as-is
ts	Leave as-is
tag	Leave as-is
interface_urn	Leave as-is
interface_href	Remove. rest_call_handler can fill this in.

These changes:

- Eliminate many data redundancies
- Simplify the task of understanding and populating the tables
- Apply to both object and relationship tables

Sample JSON response to data query


```
[
  {"tsdata":[
        {"ts":1412124480000000,"v":7917},
        {"ts":1412124600000000,"v":7026},
        {"ts":1412125320000000,"v":8731}],
    "eventType":"ops_monitoring:rx_bps”,
    "subject": “https://gmoc-db.grnoc.iu.edu/info/interface/rtr.wash.ion.internet2.edu/xe-0/2/2”
    "id":"rx_bps:rtr.wash.ion.internet2.edu:xe-0/2/2",
    "description":"bits per second received on this interface",
    "units":"float",
    "$schema": “http://www.gpolab.bbn.com/monitoring/schema/20140828/data#”
  },
  {“tsdata”:[….] }
]
```


For reference, this is what a typical response to a data query currently looks like. We will look at some possible changes to this format in the next slides.

JSON data response: top-level list

[does not match data schema (schema does not start with a list)


```
 {"tsdata":[
      {"ts":1412124480000000,"v":7917},
      {"ts":1412124600000000,"v":7026},
      {"ts":1412125320000000,"v":8731}],
  "eventType":"ops_monitoring:rx_bps",
  "subject": "https://gmoc-db.grnoc.iu.edu/info/interface/rtr.wash.ion.internet2.edu/xe-0/2/2"
  "id":"rx_bps:rtr.wash.ion.internet2.edu:xe-0/2/2",
  "description":"bits per second received on this interface",
  "units":"float",
  "$schema": "http://www.gpolab.bbn.com/monitoring/schema/20140828/data#"
 },
 {"tsdata":[….] }
]
```


Currently, the response is a list [ ] of JSON objects each of which conform to the data schema. The response as a whole does NOT pass JSON validation. Special-case validation code is needed for this one response. Ideally, the entire response would pass JSON validation without special treatment.

JSON data response: top-level list FIX


```
{
  "$schema": http://www.gpolab.bbn.com/monitoring/schema/20141205/data#
  “measurements”: [
    {
      "tsdata":[
        {"ts":1412124480000000,"v":7917},
        {"ts":1412124600000000,"v":7026},
        {"ts":1412125320000000,"v":8731}],
      "eventType":"ops_monitoring:rx_bps”,
      "subject": “https://gmoc-db.grnoc.iu.edu/info/interface/rtr.wash.ion.internet2.edu/xe-0/2/2”
      "id":"rx_bps:rtr.wash.ion.internet2.edu:xe-0/2/2",
      "description":"bits per second received on this interface",
      "units":"float”,
    },
    {“tsdata”:[….] }
  ]
}
```


- $schema only appears once in the entire response
- No impact on database table population
- Collectors must parse new format

JSON data response: timestamps


```
[
  {"tsdata":[
        {"ts":1412124480000000,"v":7917},
        {"ts":1412124600000000,"v":7026},
        {"ts":1412125320000000,"v":8731}],
    "eventType":"ops_monitoring:rx_bps”,
    "subject": “https://gmoc-db.grnoc.iu.edu/info/interface/rtr.wash.ion.internet2.edu/xe-0/2/2”
    "id":"rx_bps:rtr.wash.ion.internet2.edu:xe-0/2/2",
    "description":"bits per second received on this interface",
    "units":"float",
    "$schema": “http://www.gpolab.bbn.com/monitoring/schema/20140828/data#”
  },
  {“tsdata”:[….] }
]
```


- Timestamps could probably be milliseconds instead of microseconds
- Could reduce the response size significantly
- We would want to make this change across the entire schema for consistency
- Database populators would need to change to store milliseconds
- Collectors would have to deal with both time granularities for a while, facilitated by schema versioning

JSON data response: ops_monitoring prefix


```
[
  {"tsdata":[
        {"ts":1412124480000000,"v":7917},
        {"ts":1412124600000000,"v":7026},
        {"ts":1412125320000000,"v":8731}],
    "eventType":"ops_monitoring:rx_bps”,
    "subject": “https://gmoc-db.grnoc.iu.edu/info/interface/rtr.wash.ion.internet2.edu/xe-0/2/2”
    "id":"rx_bps:rtr.wash.ion.internet2.edu:xe-0/2/2",
    "description":"bits per second received on this interface",
    "units":"float",
    "$schema": “http://www.gpolab.bbn.com/monitoring/schema/20140828/data#”
  },
  {“tsdata”:[….] }
]
```


- ops_monitoring prefix seems unnecessary. Remove it.
- Also remove it in the REST call that queries for this data
- No impact on database population
- Collectors would see different eventType strings, but it shouldn’t matter?

JSON data response: subject field


```
[
  {"tsdata":[
        {"ts":1412124480000000,"v":7917},
        {"ts":1412124600000000,"v":7026},
        {"ts":1412125320000000,"v":8731}],
    "eventType":"ops_monitoring:rx_bps”,
    "subject": “https://gmoc-db.grnoc.iu.edu/info/interface/rtr.wash.ion.internet2.edu/xe-0/2/2”
    "id":"rx_bps:rtr.wash.ion.internet2.edu:xe-0/2/2",
    "description":"bits per second received on this interface",
    "units":"float",
    "$schema": “http://www.gpolab.bbn.com/monitoring/schema/20140828/data#”
  },
  {“tsdata”:[….] }
]
```


- Our schema convention is to use the _href suffix for fields that contain URLs, so add that here
- Diverges from UNIS schema, but we have been moving away from UNIS
- No impact on database population
- Collectors must parse new format

JSON data response: id field


```
[
  {"tsdata":[
        {"ts":1412124480000000,"v":7917},
        {"ts":1412124600000000,"v":7026},
        {"ts":1412125320000000,"v":8731}],
    "eventType":"ops_monitoring:rx_bps”,
    "subject": “https://gmoc-db.grnoc.iu.edu/info/interface/rtr.wash.ion.internet2.edu/xe-0/2/2”
    "id":"rx_bps:rtr.wash.ion.internet2.edu:xe-0/2/2",
    "description":"bits per second received on this interface",
    "units":"float",
    "$schema": “http://www.gpolab.bbn.com/monitoring/schema/20140828/data#”
  },
  {“tsdata”:[….] }
]
```


- The id field does not need to start with the event (rx_bps here)
  - The event is already available as eventType
  - id no longer directly references the relevant object; this is confusing
- No impact on database population
- Collectors must parse new format

JSON data response: $schema field


```
[
  {"tsdata":[
        {"ts":1412124480000000,"v":7917},
        {"ts":1412124600000000,"v":7026},
        {"ts":1412125320000000,"v":8731}],
    "eventType":"ops_monitoring:rx_bps”,
    "subject": “https://gmoc-db.grnoc.iu.edu/info/interface/rtr.wash.ion.internet2.edu/xe-0/2/2”
    "id":"rx_bps:rtr.wash.ion.internet2.edu:xe-0/2/2",
    "description":"bits per second received on this interface",
    "units":"float",
    "$schema": “http://www.gpolab.bbn.com/monitoring/schema/20140828/data#”
  },
  {“tsdata”:[….] }
]
```


- Don’t need to repeat $schema in every block of tsdata
- Already addressed in top-level list fix in a previous slide
- No impact on database population
- Collectors must parse new format

Revised JSON response to data query

Combining all of these changes, we get something like this:

```
{
  "$schema":http://www.gpolab.bbn.com/monitoring/schema/20140828/data#
  “measurements”: [
    {
        "tsdata":[
            {"ts":1412124480000,"v":7917},
            {"ts":1412124600000,"v":7026},
            {"ts":1412125320000,"v":8731}],
         "eventType":”rx_bps”,
         "subject_href":https://gmoc-db.grnoc.iu.edu/info/interface/…”
         "id":”rtr.wash.ion.internet2.edu:xe-0/2/2",
         "description":"bits per second received on this interface",
         "units":"float",
    },
    {“tsdata”:[….] }
  ]
}
```

Other Possible JSON Schema Enhancements

- Bidirectional navigation
  - Nodes list their interfaces, but interfaces don't specify their node. Fix all such instances of this.
  - Allows lightweight navigation of the monitoring data without needing to collect and store all of the data first
  - This is a sweeping change that dwarfs any of the other JSON schema changes that have been discussed
- Wildcard for event types in data queries
  - Objects can already be wildcarded, so this completes the feature
  - Gotchas with datastores supporting multiple aggregates
- Objects could specify what measurements they support
  - Would make querying data more straightforward
- Node schema
  - Associate VMs with the physical machines they're running on
  - Add a baremetal node type

Possible new features

- OpenFlow support
  - New flowspace resource type? What's in a flowspace resource?
  - Adding this is a certainty, not just a possibility
- Flexible install location
  - instead of requiring /usr/local/ops-monitoring
- Release packaging (RPM, .deb, BSD pkg)
- Documentation