Panoptes: A Network Telemetry Ecosystem - Part Deux

Varun Varma, Sr. Principal Engineer
vvarun@oath.com
Panoptes is:

- Greenfield Python based network telemetry platform that provides real time telemetry and analytics @ Yahoo
- Implements discovery, polling, distribution bus and numerous consumers
Architecture

- Device Specific Plugins (SNMP, API)
  - Discovery Plugins
  - Polling Plugins
  - Enrichment Plugins
- Plugin Framework
- Celery
- Redis
- Zookeeper
- Kafka
- Time Series DB
- CMDB
- Chef
Core Concepts

- Plugins
- Resources
- Metrics
- Enrichments
- Data encoding and distribution
Plugins:

- Are Python classes conforming to a well defined API
- Can collect/process and transform data from any source
  - SNMP
  - API
  - CLI
  - *
- May be of three types:
  - Discovery
  - Enrichment
  - Metrics
Resources:

- Are abstract representations of what should be monitored
  - In the context of network telemetry, these would be usually be the network devices to monitor
- Are ‘discovered’ using discovery plugins
  - Which, usually, would would talk to a Configuration Management Database - but could also be from topology walks
- Have and id, endpoint and various metadata
  - For example, the vendor name or operating system version of a device would be it’s metadata
- Are specified within Panoptes with a DSL
Metrics:

- Are numbers that can be measured and plotted
  - Example is the bytes in/bytes out counter of an interface
- Are generally fast changing
  - Or have the potential to do so
- Can be collected through various means:
  - SNMP
  - API
  - CLI
  - Streaming
Enrichments:

- **Are metadata in addition to metrics**
  - For interfaces, we collect metrics like bytes in and bytes out and enrichments like interface name and description

- **Can be any data type**
  - Unlike metrics which can only be numeric

- **Can come from sources other than the device being monitored**
  - The geo location of the device or the ASN number to name mapping
Enrichments - pt. 2:

- Usually are more expensive to process than metrics
  - Might need complex transformations
  - And therefore:
    - Are collected at a rate less than those for metrics
      - We collect interface metrics every 60 seconds, but enrichments every 30 minutes
    - Are cached

- Overall, let us scale more by being efficient about data collection
Data Encoding and Distribution

- Panoptes is a distributed system
  - Discovery, enrichment and polling are all decoupled from each other
- Kafka and/or Redis are used to pass data between all subsystems
  - This makes it so that you can extend or introspect any subsystem
- JSON is used to encode all data within Panoptes
  - It’s non-performant but developer/operator friendly
Workflow

Collect Data → Post Process → Message Bus

TSDB
- Graphing
- Alerting
- Grid

MySQL

API
- UI
- CLI

Analytics/Reporting
Scaling and Operations
Scale: Orders of Magnitude

100K
Servers

10K
Network Devices

100
Network Sites

1M
Time Series

60
Seconds

10
Systems Replaced
Scaling Issues

- Panoptes was built to be horizontally scalable and free of single points of failure from day one
  - Performance or high-availability are not easy to bolt on afterwards
- We chose Python to be developer friendly - it was every bit as slow as we thought
  - High throughput actions are delegated to C extension modules
  - Ditto for JSON serialization for all data
- We broke everything - Redis, Zookeeper, Kafka
  - Redis allows ‘only’ 10,000 clients to be connected by default :)
Divide and Conquer: Federated API

- Due to availability concerns, each site has its own MySQL cluster
  - Telemetry data must be available during a network partition
  - Centralized telemetry store might not be reachable in all cases
- Each API endpoint acts as a tribe node
- If a tribe node doesn’t have the requested data, it returns a pointer to the node that does through a find API
Covered Systems

- Interface metrics for Arista, Cisco, Juniper
- System metrics for A10 (AX, TH), Arista EOS, Brocade TrafficWorks, Cisco IOS-XE, Juniper (MX, SRX)
- Functional metrics for VIPs (A10 AX, TH, Brocade), A10 LSN, Juniper SRX
- Ethernet state and topology for Cisco IOS, NX-OS
Operational Experiences

- Metrics across different platforms or versions of even the same OS from vendors aren’t consistent
  - Normalizing these metrics was our single biggest time drain
- SNMP has it’s faults, but is still ubiquitous
  - Specially so in a multi-vendor, multi-platform and multi-generational networks such as ours
- Performance of APIs was *much* better than SNMP
- We knew that we didn’t know how we would use the data
  - Using Kafka proved to be the right choice - we already have 3 separate consumers
Operational Experiences - Pt. 2

- We don’t expose ‘raw’ data to external systems
  - It’s tempting to give access to external teams via Kafka, but that would lead to friction if we want to change our internals
  - Instead, we expose APIs which abstract away all our internals

- Custom UIs are useful
  - And enabled by APIs
API Examples

Realtime – Purpose
Specific

Bulk/Historical - Generic

```json

```

```json

```
Centralized Telemetry

- We push metrics to our in-house time series database and alerting service (centralized telemetry)
- Custom dashboard service our user base is familiar with
- Economies of scale – no need to provision new hardware or software

Here we see control and data plane CPU statistics for a load balancer in one of our West Coast data centers.
Custom UI Examples
Future: Streaming Telemetry
Proposed Architecture

Device 1 → Resource Cache

Device 2 → Streaming Telemetry Collector

Device 3 → Enrichment Cache

Device n

Panoptes Framework

Celery
Redis
Zookeeper
Kafka
A recap: we were here at NANOG 70...
Since then:

- Added an enrichment subsystem
- Built many plugins to poll system and functional metrics
- Worked out scaling issues
- And most importantly...
I’m here to announce open sourcing of Panoptes
MVP

- Core Platform including discovery, enrichment and polling
- Interface metrics and enrichment plugins
  - Also heartbeat plugins
- Integration with InfluxDB
An example dashboard
Get it at: https://github.com/yahoo/panoptes
Questions?