Getting Started with the Network Monitoring and Alarm System Project

Hi! I (a previous intern of the IRIS-HEP fellows program 2024) created this manual to help you quickly immerse yourself in the ongoing project. The goal of the project is to enhance and optimize a network monitoring system by developing and refining alarms that notify users of potential problems in network performance. In this document, I've outlined key starting points, important tools, contacts, and potential challenges to help you get up to speed and continue where I left off.

Prepared by

YANA HOLOBORODKO

LinkedIn Yana Holoborodko

goloborodko.yana2016@gmail.com

Under the guidance of

PETYA VASILEVA (University of Michigan)

LinkedIn Petya Vasileva

petyav@umich.edu

+

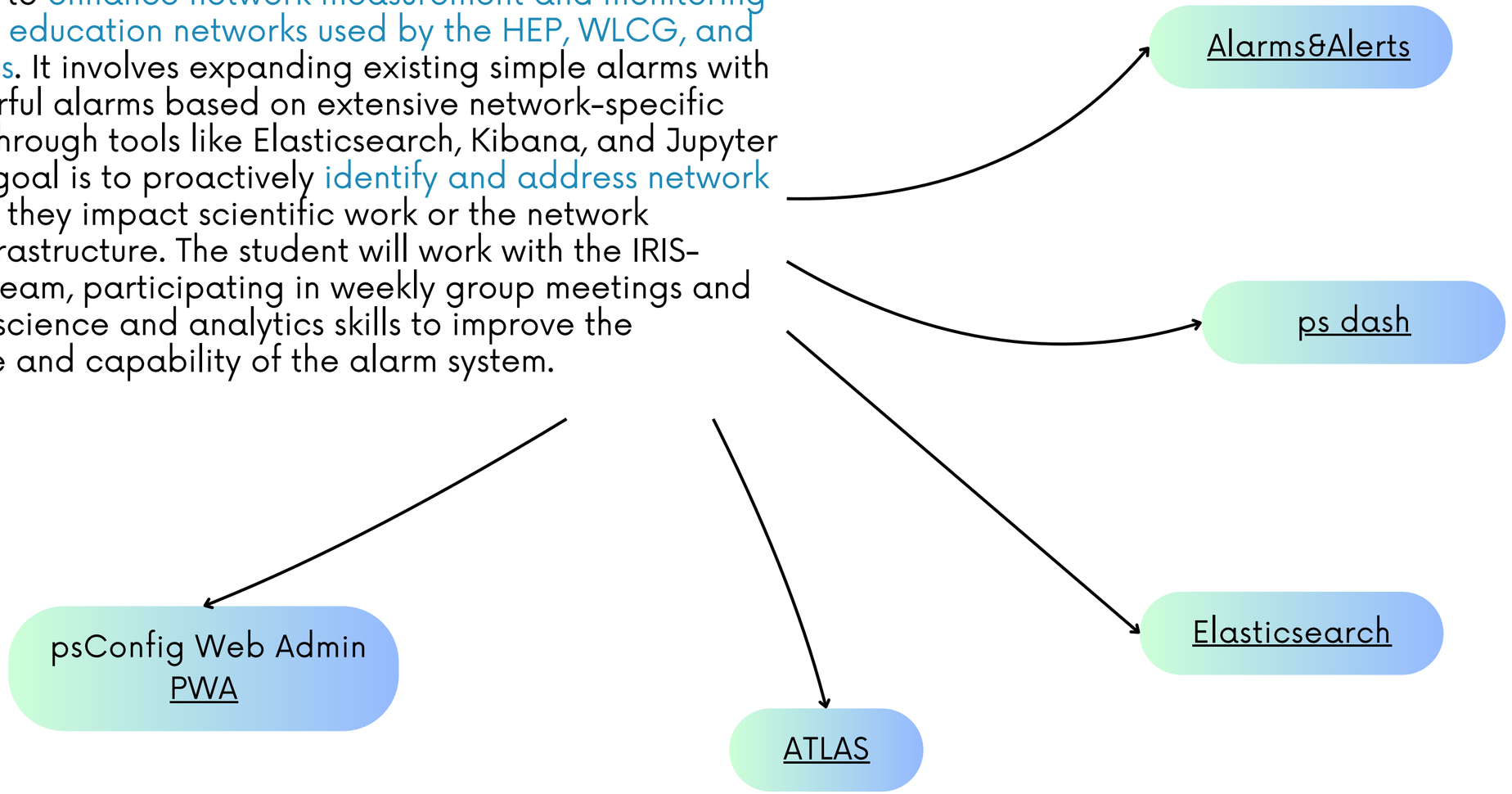
SHAWN MCKEE (University of Michigan)

LinkedIn Shawn McKee

smckee@umich.edu

Enabling Advanced Network and Infrastructure Alarms

The project aims to enhance network measurement and monitoring for research and education networks used by the HEP, WLCG, and OSG communities. It involves expanding existing simple alarms with new, more powerful alarms based on extensive network-specific data collected through tools like Elasticsearch, Kibana, and Jupyter Notebooks. The goal is to proactively identify and address network problems before they impact scientific work or the network measurement infrastructure. The student will work with the IRIS-HEP/OSG-LHC team, participating in weekly group meetings and using their data science and analytics skills to improve the diagnostic range and capability of the alarm system.



SOME OF EXISTING ALARMS

Bad one-way delay

measurements is generated if a node reports time greater than 100ms

Large clock correction

alarm calculates clock corrections for all nodes that appear as both source and destination

Complete packet loss

alarm is created when a link drops all packets

Firewall issue

is an alarm generated when node is involved in links that lost 100% of its packets for all tests in a given period or when the number of links (having lost all packets) is more than 10

High packet loss

problem alerts for packet loss above 2%.

Alarms

- Analytics
  - Frontier
    - Failed queries
    - Too many threads
    - Bad SQL queries
  - WFMS
    - indexing
  - Elasticsearch
    - status
- Networking
  - Perfsonar
    - bad owd measurements
    - large clock correction
    - firewall issue
    - complete packet loss
    - unresolvable host
  - Infrastructure
    - indexing
  - Sites
    - destination cannot be reached from multiple
    - destination cannot be reached from any
    - bandwidth increased from/to multiple sites
    - bandwidth decreased from/to multiple sites
    - high packet loss
    - source cannot reach any
    - bandwidth increased
    - bandwidth decreased
  - RENs
    - path changed
- Virtual Placement
  - XCache
    - dead server
    - large number of connections
    - external test
- SLATE
  - Squid
    - failovers
    - server down
- WFMS
  - User
    - Too much walltime consumed

Current Subscriptions

Category	Subcategory	Event	Tags
Networking	Perfsonar	bad owd measurements	*
Networking	Perfsonar	large clock correction	*
Networking	Perfsonar	firewall issue	*
Networking	Perfsonar	complete packet loss	*
Networking	Perfsonar	unresolvable host	*
Networking	Infrastructure	indexing	*
Networking	Sites	destination cannot be reached from multiple	*
Networking	Sites	destination cannot be reached from any	*
Networking	Sites	bandwidth increased from/to multiple sites	*
Networking	Sites	bandwidth decreased from/to multiple sites	*
Networking	Sites	high packet loss	*
Networking	Sites	source cannot reach any	*
Networking	Sites	bandwidth increased	*
Networking	Sites	bandwidth decreased	*
Networking	RENs	path changed	*

Showing 1 to 15 of 15 entries

Update Subscription

WHERE TO START?

1. Understand the goal the team is working on

To cut a long story short, the team is working on creating a notification system that will (indeed, it already does this) inform users of the network about existing problems in its operation. There can be many of them: lost packets when sending packets from host to host, severe reductions in host bandwidth, or, for example, too much load on the system by one user trying to send or receive too much data, etc. The team tries to use analytics to identify possible problems and develop an alarm for them, which will regularly check for them on the network and notify subscribed users to make them aware of the problem or even suggest ways to fix the system.

Alarms&Alerts

Alarms&Alerts

Here you can

- create an account, login and try to subscribe to some alarms to see how it works (it sends the message about found problem to your email if the alarm identifies it)
- read documentation and understand the process of deployment the new alarm
- find this json file with list of all currently working alarms and their description

2. Understand the data you will work with

The data used to find certain anomalies is synthetic data obtained by running tests and storing their results in a database.
Perfsonar (performance Service-Oriented Network monitoring ARchitecture) - a network measurement toolkit. Perfsonar in this project runs network performance tests by regularly sending data between hosts (network nodes) to measure key metrics like latency, throughput, and packet loss.

perfSONAR

These tests are automated and scheduled to run across multiple hosts, with the results stored in a central database for analysis. The toolkit uses a mesh configuration to define which hosts to test, allowing for continuous monitoring of network paths and detection of performance anomalies between specific network nodes.

perfSONAR
pSConfig Web Admin

Which tests does Perfsonar run?

Check psConfig Web Admin to see all the configurations (ask to create an acount for you first if you need to explore them). Each configuration is a json file where hosts to test, its schedule and type are included.

psConfig Web Admin
PWA

Hint: in case you need to extract the information from configurations or just need the list of all hosts in interest you can run 'pip install psconfig-client' in your development environment and use API.

https://github.com/marian-babik/psconfig-client/blob/main/psconfig/api.py

ATLAS EXPERIMENT | Distributed Computing Analytics

ATLAS

Elasticsearch

elastic

Which data does it produce and where to access it?

Perfsonar collects different data depending on the tests and puts it into Elasticsearch. You can see a visualisation of this database on Elasticsearch. To get access to extracting and processing this data, you need to ask your mentors to create an account here, or you can request an account on the ATLAS platform (confirmation may take up to 10 days), where you can create a Jupiter Notebook for code development.

jupyter

Hint: having got your account in ATLAS you can start by running this Jupyter Notebook and trying to do visualization to understand the data

Just one more thing about the data

Generally, 3 types of network tests are taken into account and they are run on ~314 hosts:

Trace Tests (ps_trace)

Trace tests are used to map the path between two hosts, identifying each network hop along the way. They help in pinpointing specific segments in the network that may cause delays or failures. By analyzing the route taken by data, trace tests provide insight into routing issues.

Frequency: usually run every 5/10/15 minutes -> produces lots of data. It's optimal to query for 2-hour time interval

Throughput Tests (ps_throughput)

Throughput tests measure the amount of data that can be successfully transferred between two hosts over a network in a specific amount of time. These tests are crucial for assessing network capacity and identifying congestion points or bandwidth bottlenecks.

Frequency: usually run every 2h. It was optimal for me to query for 6-hour time intervals, if needed it will not take a long time to query for all 24hour or even several days.

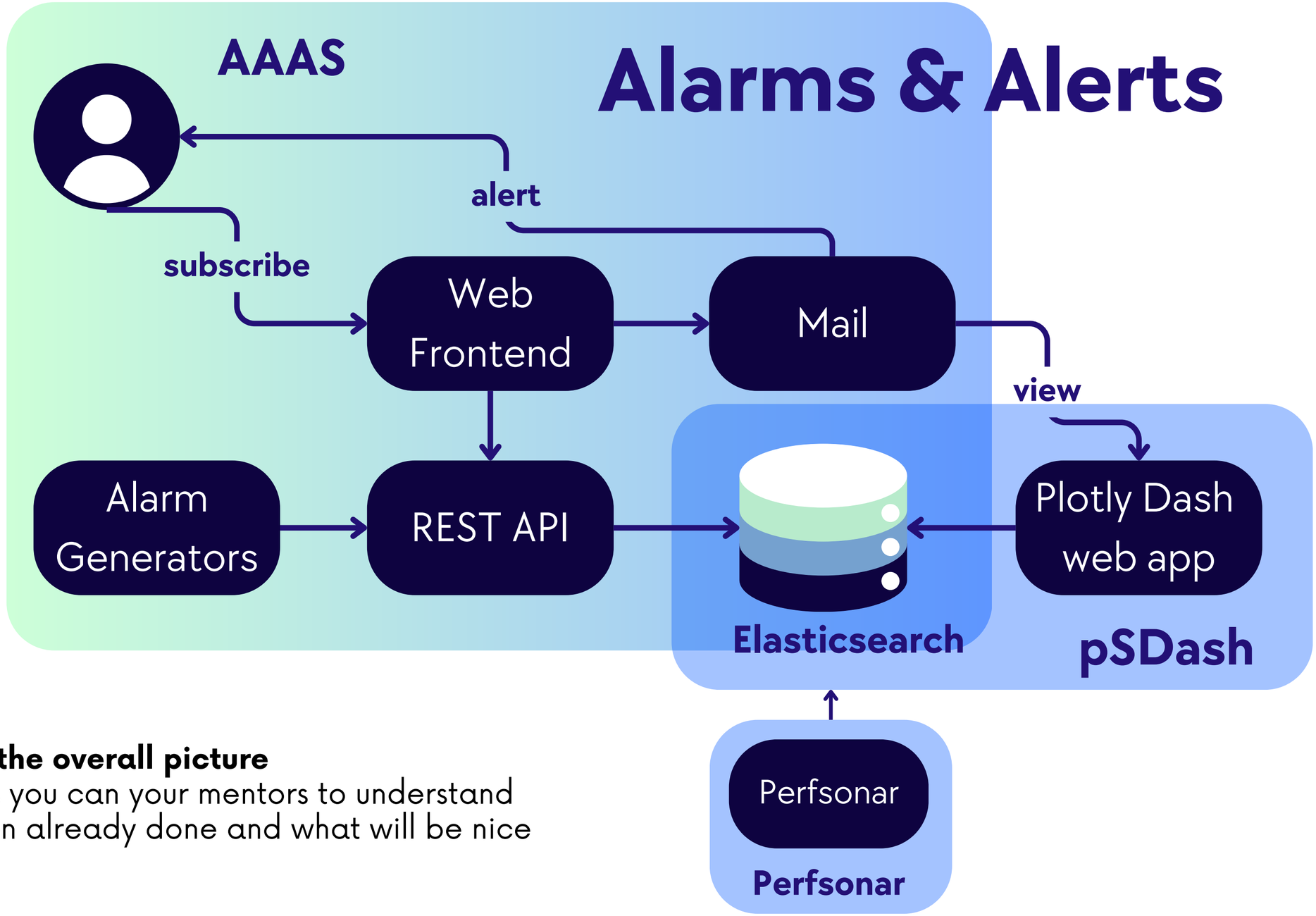
Latency Tests (ps_owd - One-Way Delay)

Latency tests measure the time it takes for data to travel from one host to another. This provides an indication of network responsiveness, highlighting delays that may affect the network performance.

Frequency: latency tests are often run more frequently than throughput and trace tests to ensure detection of network delays. Most of them run continiously.

ps dash

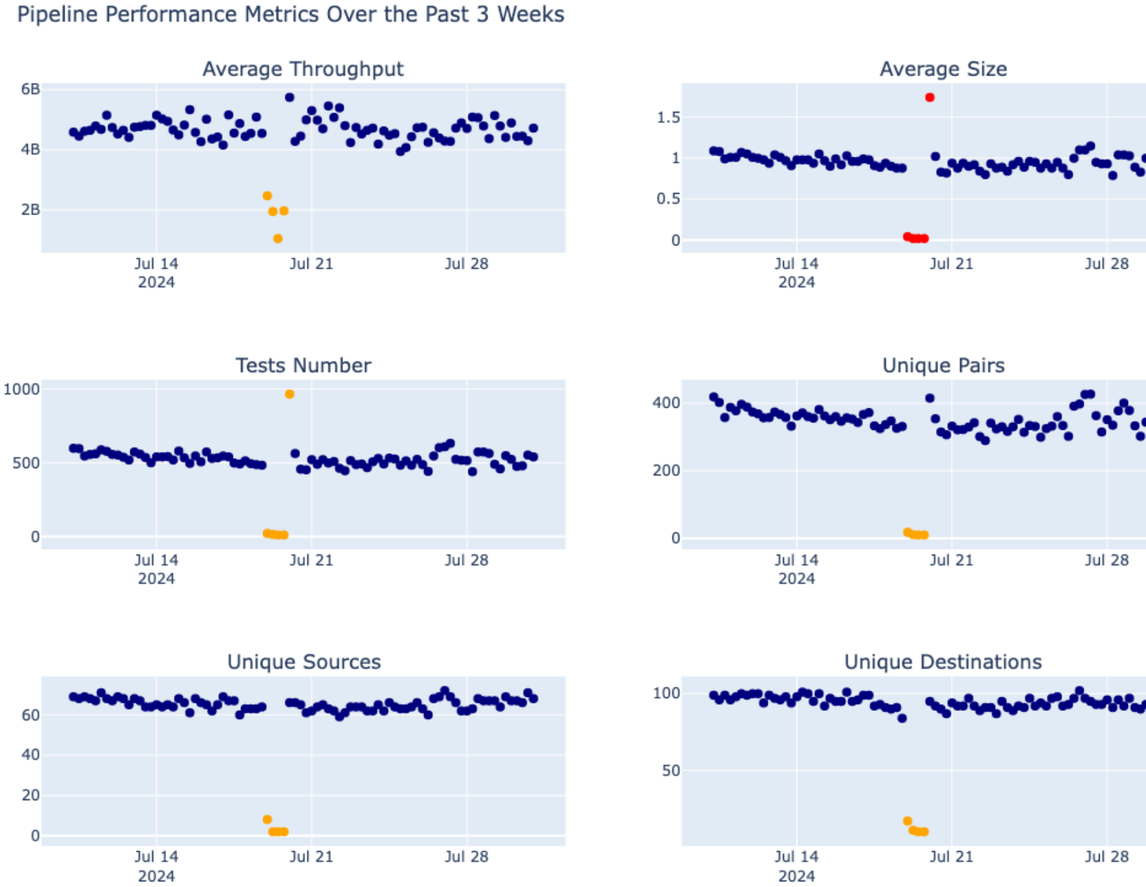
Hint: you can also see the visualization for alarms and different statistics on ps dash



3. Make sure you understand the overall picture
Try to ask as many questions as you can your mentors to understand what is going on, what has been already done and what will be nice to have.

4. What I've done and where I ended

Starting with the visualization, I noticed that some data might not be delivered to the dataset at all. For example, the graph shows the size of the data that was sent to Elasticsearch ps_throughput every 6 hours for three weeks. The nature of this 'anomaly' (in the form of a sharp decrease in measured values) is very difficult to determine, but it prompted my mentors and me to create an alarm that would test the condition of the internal pipeline. The idea of determining the condition of the data flows based on their size was not very valid, as different testing times and possible delays in their delivery could give false results.



Therefore, the alarm that I undertook to develop was designed as follows (described in general terms):

1. getting access to the configuration
2. reading all hosts and test types for them from the configurations, which should result in the corresponding records in Elasticsearch
3. extracting data for a certain period of time from Elasticsearch
4. checking if the records of expected hosts match the existing ones

Pitfall 1: the psConfig Web Admin can be accessed by lots of people and you can find there unexpected things like empty configuration, etc. Take it into account while working on your project

Pitfall 2: sometimes hosts that are not of this project's interest can happen to be in the Elasticsearch data. These are other users' configurations.

Pitfall 3: while implementing the alarm it was discovered that some of the hosts are unresolvable (they are either retired or currently offline)

*3rd pitfall became the reason for creating one more alarm

unresolvable host alarm

example of alarm's work

Problem

- PWA does not always represent up-to-date information about hosts. Therefore, some configurations include tasks which are not tested in real life as corresponding hosts are retired/offline. Consequently, the pipeline alarm searches for data that can not be found in the Elasticsearch and raises the false alarm.

Solution

Infrastructure alarm

- alarm that will discover which hosts are retired
- check if the host still resolves. Iterate through all hosts we are interested in and ping them. Then report to the user those that were unsuccessful.

```
{
    "category": "Networking",
    "subcategory": "Perfsonar",
    "event": "unresolvable host",
    "description": "This code checks whether all hosts in various configurations are.
resolvable via DNS. It is executed once a week and takes as input a mesh configuration that
contains multiple configurations and hosts. For each host, it performs a DNS resolution check.
If a host is not resolvable, it is flagged, and the corresponding configuration is noted for
necessary updates. This process is essential for keeping the network and results of other
alarms and alerts reliable and up-to-date. The code can be found here: [..].",
    "template": "The following host is not resolvable in the given configurations:\n%
{host}\nConfigurations affected:\n%{configurations}\nPlease review and update the DNS entries
or configurations as needed to resolve the issue."
 }
```

json file with alarm template

ATLAS Alarm & Alert System <aaas@analytics.mwt2.org>
кому мені

Dear YANA HOLOBORODKO,

Herewith a list of alarms you subscribed to. Preferences may be changed by goloborodko.yana2016@gmai

Mon, 30 Sep 2024 15:52:37 Networking/Perfsonar/unresolvable host unresolvable host
tags: iut2-net4.iu.edu, ps-Testbed Mesh Config
The following host is not resolvable in the given configurations:
iut2-net4.iu.edu
Configurations affected:
['ps-Testbed Mesh Config']
Please review and update the DNS entries or configurations as needed to resolve the issue.

Mon, 30 Sep 2024 15:52:37 Networking/Perfsonar/unresolvable host unresolvable host
tags: psonar-bwctl.brazos.tamu.edu, ps-Testbed Mesh Config
The following host is not resolvable in the given configurations:
psonar-bwctl.brazos.tamu.edu
Configurations affected:
['ps-Testbed Mesh Config']
Please review and update the DNS entries or configurations as needed to resolve the issue.

Mon, 30 Sep 2024 15:52:37 Networking/Perfsonar/unresolvable host unresolvable host
tags: iut2-net2.iu.edu, USATLAS Mesh Config, WLCG ATLAS Bandwidth Mesh Config
The following host is not resolvable in the given configurations:
iut2-net2.iu.edu
Configurations affected:
['USATLAS Mesh Config', 'WLCG ATLAS Bandwidth Mesh Config']
Please review and update the DNS entries or configurations as needed to resolve the issue.

Mon, 30 Sep 2024 15:52:37 Networking/Perfsonar/unresolvable host unresolvable host
tags: perfsonar-sandie-5.ultralight.org, USCMS Mesh Config
The following host is not resolvable in the given configurations:
perfsonar-sandie-5.ultralight.org
Configurations affected:
['USCMS Mesh Config']
Please review and update the DNS entries or configurations as needed to resolve the issue.

Mon, 30 Sep 2024 15:52:37 Networking/Perfsonar/unresolvable host unresolvable host
tags: perfsonar-sandie-7.ultralight.org, USCMS Mesh Config
The following host is not resolvable in the given configurations:
perfsonar-sandie-7.ultralight.org
Configurations affected:
['USCMS Mesh Config']
Please review and update the DNS entries or configurations as needed to resolve the issue.

github repository

*3rd pitfall became the reason for creating one more alarm

meta/throughput/latency/trace data compliance alarm

Problem

- We do not know if all test data is stored in Elasticsearch. Sometimes not all tests specified in the configurations can be found in the database afterwards. The other side of the problem is that not all hosts are listed in the configurations at all. Therefore, some hosts are simply omitted and are not tested for certain or even all groups of test types.

Solution

Infrastructure alarm

- an alarm that will check, according to the configurations, whether the expected host is found in the results of the tests that should have been performed
- the user will be notified of the list of hosts not found for each test group in ES, despite the fact that they were expected according to the configurations
- the user will receive a list of hosts that are not covered by the configurations for a particular test type

```
{
    "category": "Networking",
    "subcategory": "Perfsonar",
    "event": "tests' results not found in indices",
    "description": ""This code checks whether the throughput/trace/latency data for expected hosts are found in the Elasticsearch. It is executed daily and takes a list of expected hosts as input. The function queries Elasticsearch for throughput/trace/latency data and compares it with the list of expected hosts. It reports hosts that are mentioned in the configurations but can not be found in Elasticsearch. This process is crucial for ensuring that all the relevant information about hosts' testing is saved correctly in the database. The code can be found here: [..].",
    "template": ""Hosts expected but not found in the Elasticsearch ps-%{type} (%{percent}% (%{num_not_found}/%{num_expected}) out of included to configurations not found):\n%{hosts}"
}
```

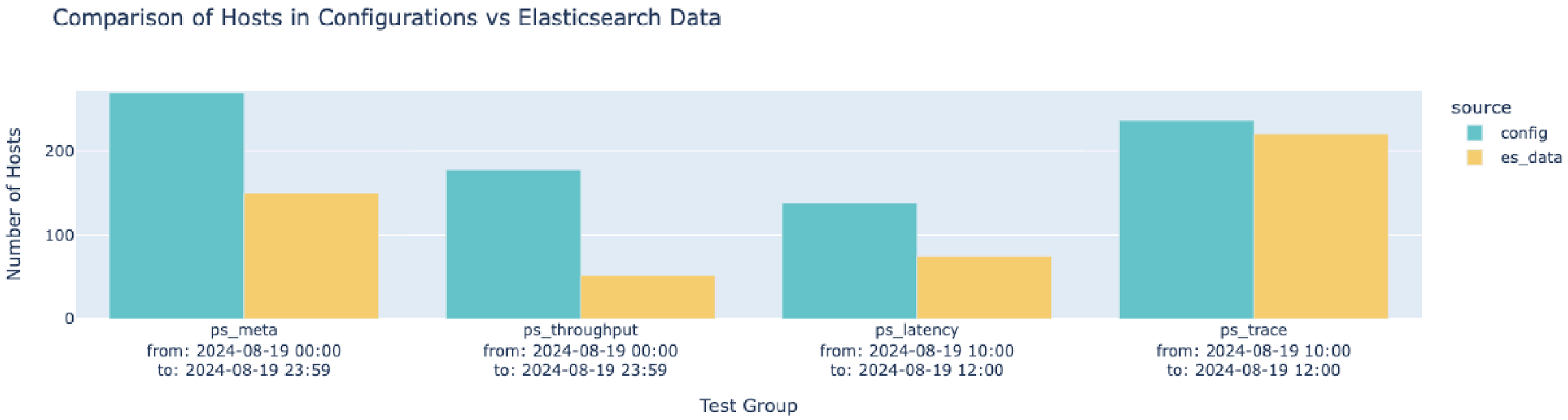
json file with alarm template example for throughput and Elasticsearch

github repository

Overview of the problem meta/throughput/latency/trace data compliance alarms are reporting

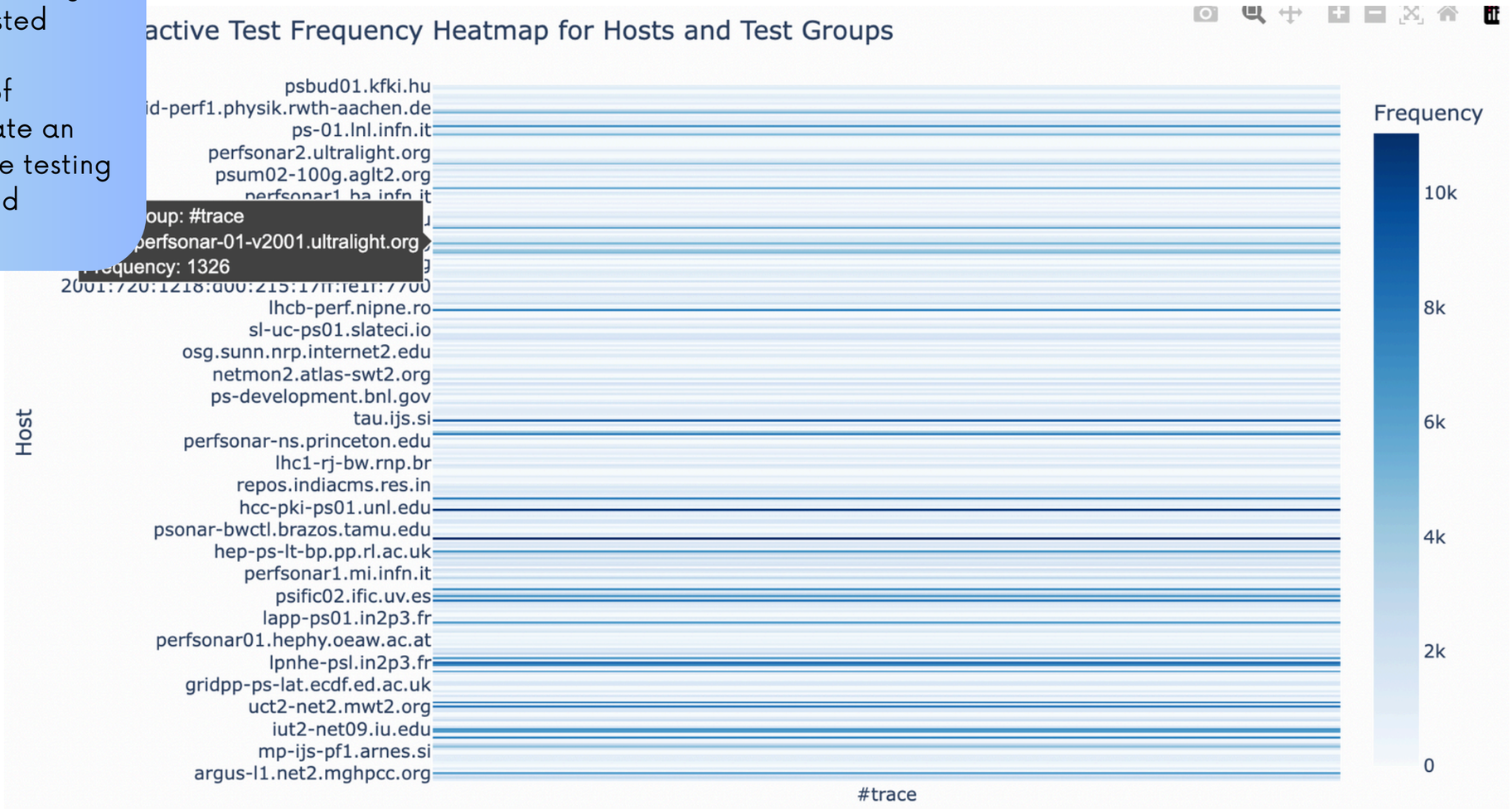
The number of hosts that were found in Elasticsearch is displayed in yellow, and the number of hosts that were covered by configurations and therefore expected to be found in Elasticsearch is displayed in turquoise. The user receives in the alert

1. a list of all hosts that were not specified in the configurations
2. a list of all hosts that were in the configurations for this test type but did not appear in Elasticsearch



5. Where I ended and potential next steps

- This hitmap shows the frequency of testing each host over a two-hour period. As you can see, they are tested very unevenly. While one host is tested 1-2 times during this period of time, the other is tested several thousand times.
- To analyse the testing coverage of different hosts, you can try to create an alarm that will alert users when the testing intensity of one host is too high and critically low.



(testing period, break) in hours

]:

	schedule	type	hosts	meta_name	count_hosts
0	(0, 0)	latencybg	{psbud01.kfki.hu, btw-lat.t1.grid.kiae.ru, per...	{USCMS Latency, WLCG ALICE IPv4 Throughput, W...	155
1	(0, 0)	rtt	{psmsu01.aglt2.org, iut2-net09.iu.edu, lhcperf...	{WLCG CMS Latency, OPN Latency, USATLAS IPv4 T...	12
2	(0.016666666666666666, 0.0)	latency	{tech-ps.hep.technion.ac.il, psmsu01.aglt2.org...	{WLCG CMS Latency, LSST IPv4 Traceroute, LHCON...	17
3	(0.016666666666666666, 0.0)	throughput	{perfsonar-cache.osgdev.chtc.io, perfsonar-cac...	{osg-xcache-throughtput}	2
4	(0.08333333333333333, 0.03333333333333333)	trace	{osg.chic.nrp.internet2.edu, osg.newy32aoa.nrp...	{StashCache IPv4 Latency, USCMS IPv4 Traceroute...	14
5	(0.16666666666666666, 0.08333333333333333)	trace	{psbud01.kfki.hu, btw-lat.t1.grid.kiae.ru, ali...	{USCMS IPv4 Traceroute, USCMS Latency, WLCG AL...	262
6	(1.0, 0.5)	rtt	{t1-pfsn2.jinr-t1.ru, btw-lat.t1.grid.kiae.ru,...	{USCMS IPv4 Traceroute, USCMS Latency, LHCONE ...	43
7	(1.0, 0.5)	throughput	{netmon2.atlas-swt2.org, mwt2-ps04.campusclust...	{ps Bandwidth Testing}	9
8	(2.0, 1.0)	throughput	{perfsonar01.hep.wisc.edu, perfsonar-01-v2001....	{StashCache IPv4 Latency, USCMS IPv4 Traceroute...	8
9	(4.0, 2.0)	throughput	{lhcmon.bnl.gov, perfsonar.na.infn.it, tau.ijs...	{LHCONE IPv4 Bandwidth, OPN IPv4 Bandwidth, Be...	11
10	(6.0, 3.0)	throughput	{perfsonar2.recas.ba.infn.it, mwt2-ps02.campus...	{USCMS IPv4 Traceroute, USCMS Latency, WLCG AL...	124
11	(23.0, 11.5)	throughput	{mwt2-ps02.campuscluster.illinois.edu, psum02....	{WLCG CMS IPv4 Bandwidth, USCMS IPv4 Traceroute...	102

- Given that the configurations can be used to extract data on the frequency of test runs (schedule), type, and number of hosts in the group being monitored, I started working on trying to select trusted threshold to identify outliers. But I didn't have enough time to figure out how to select them in a meaningful way.
- Perhaps you have an idea of how to do it differently and better.