Spike Detection in Alert Correlation:

A dive into Outliers and simple Math

Nishant Singh

Senior SRE

SRECon 21

Agenda

1 Background: Quick Introduction of Linkedin Stack

2 Alert Correlation & Problem Statement

3 Anomaly Detection & Modified Z-Score

4 Challenges & Summary

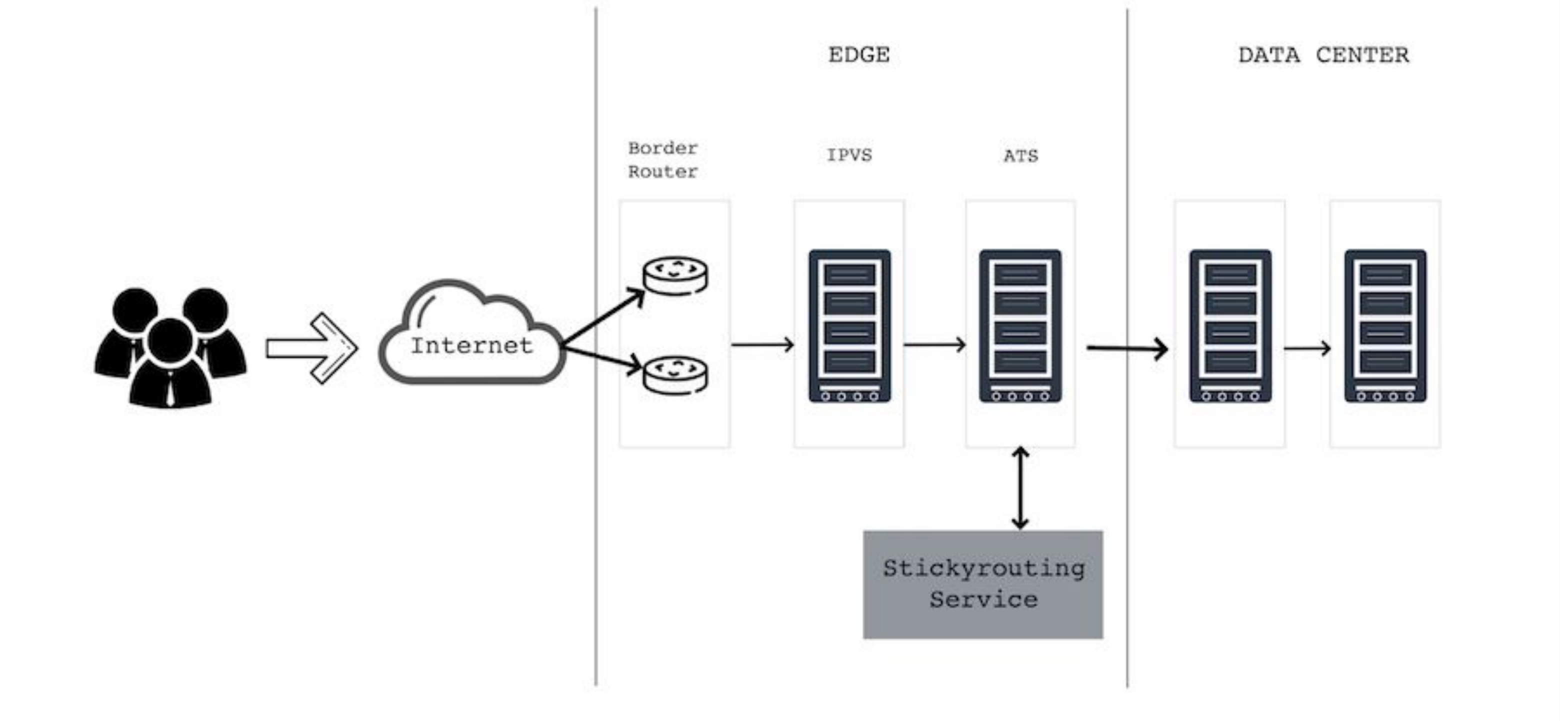
$whoami

- Senior Site Reliability Engineer @ Linkedin
- Production-SRE Team
  - Reduce MTTD & MTTR
  - Disaster Recovery
- Worked on:
  - Cloud - AWS, Azure
  - Micro-services
  - Traffic Engineering
  - Databases

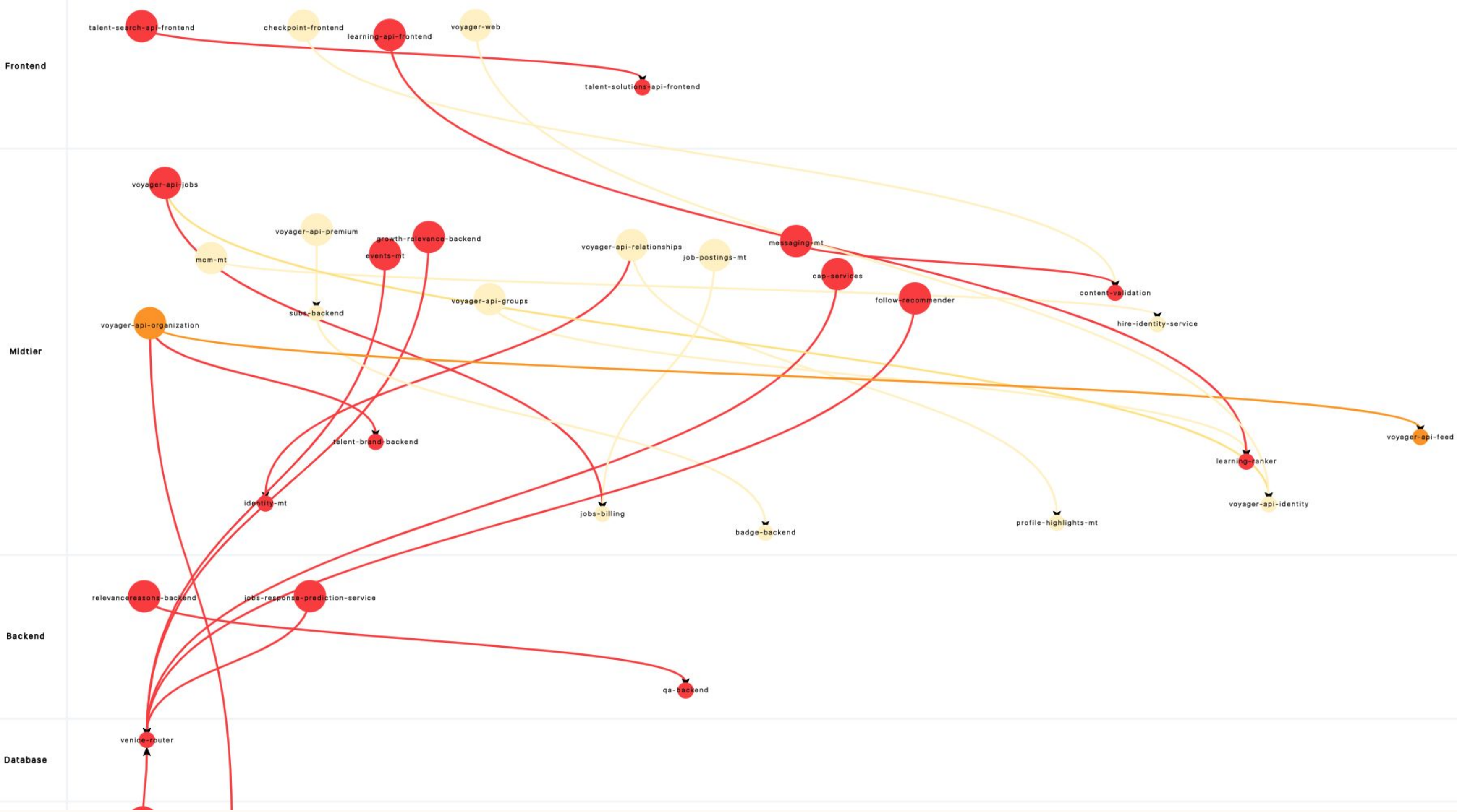
Background

LinkedIn Stack

Under the hood



An Instance of LinkedIn Services



What happens when a Production outage happens

Finding Needle in a haystack

Image source: https://comic.browserling.com/extra/22

False Pager Escalations

In the middle of night

- Paged due to your service being unhealthy due to a dependency ?
- Woken up because someone thinks that your service might be responsible ?
- Spending hours trying to figure why your service is broken?

So we needed a correlation system!

Alert Correlation

Need

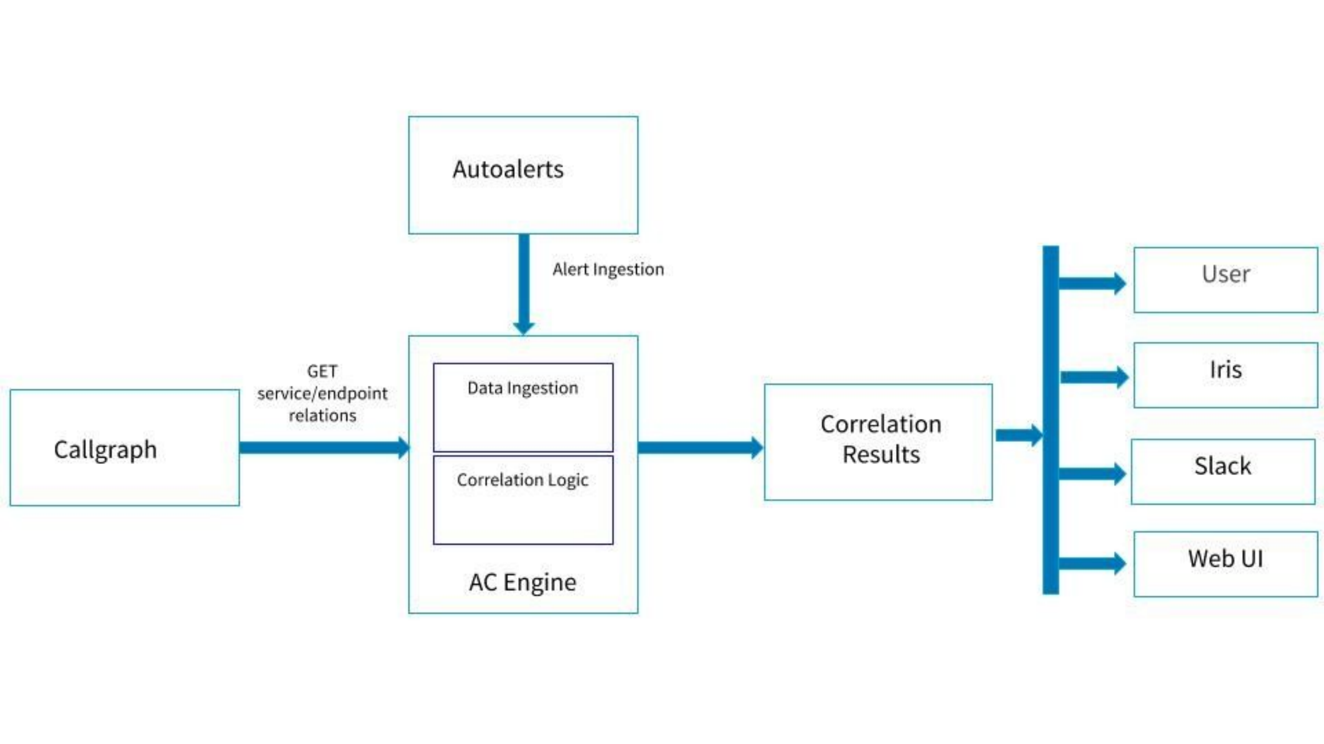
- Find a problem with a service between a given time .
- Reduce MTTR on incidents
- Reduce False escalations

Scope

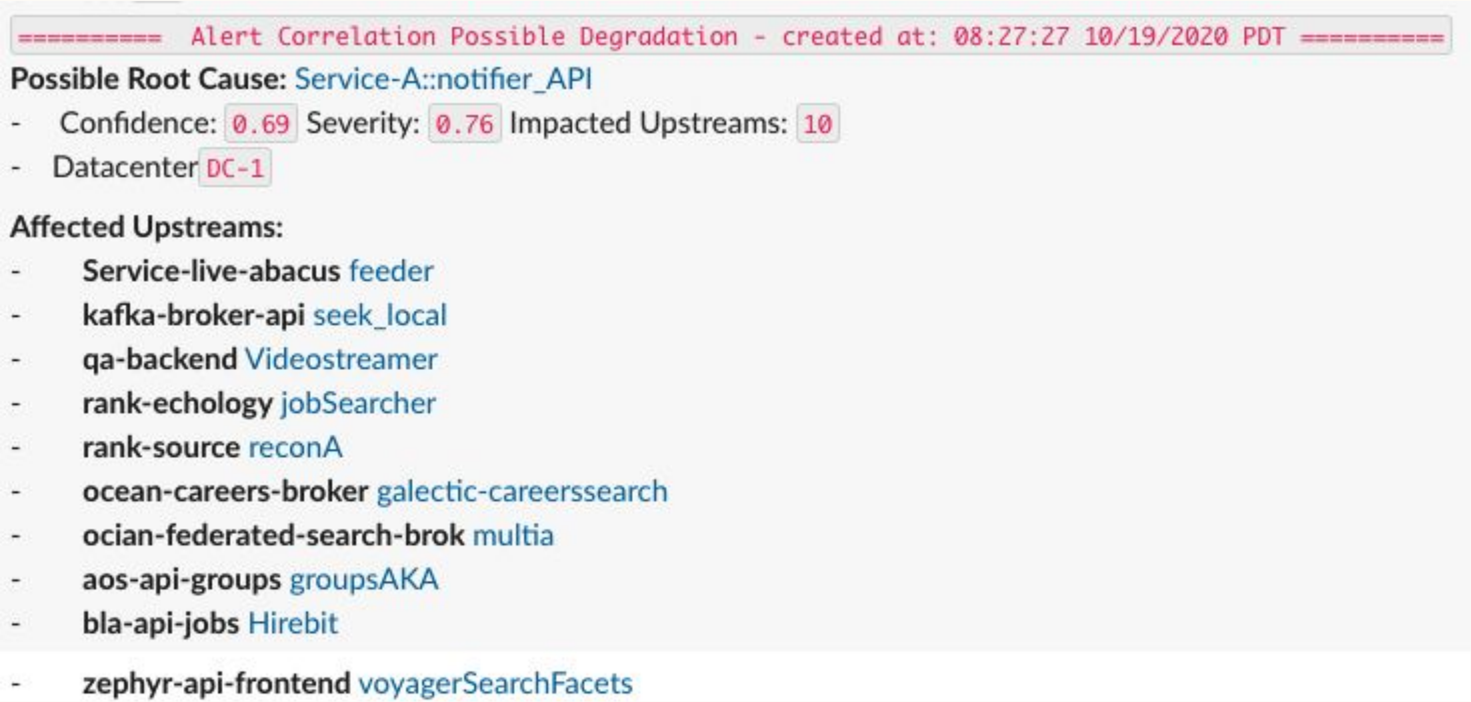
- A service has high latency or high error rates

Alert Correlation

A framework that automates the alert correlation process to identify unhealthy microservice(s).



Alert Correlation Slack Recommendations



```
==========  Alert Correlation Possible Degradation - created at: 08:27:27 10/19/2020 PDT ==========
```

Possible Root Cause: Service-A::notifier_API

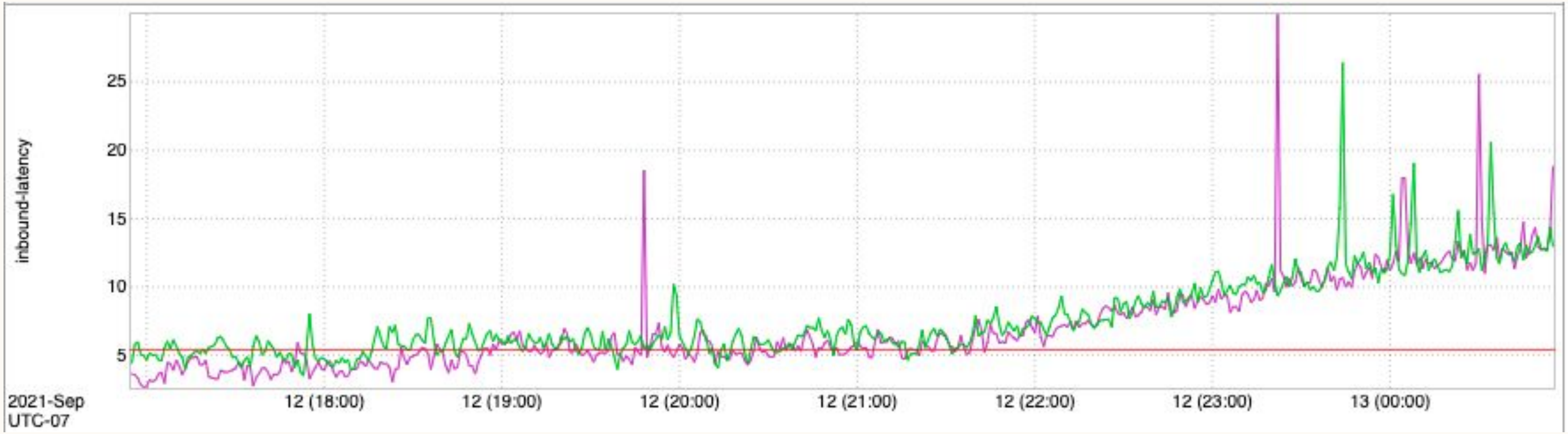
- Confidence: `0.69` Severity: `0.76` Impacted Upstreams: `10`
- Datacenter `DC-1`

Affected Upstreams:

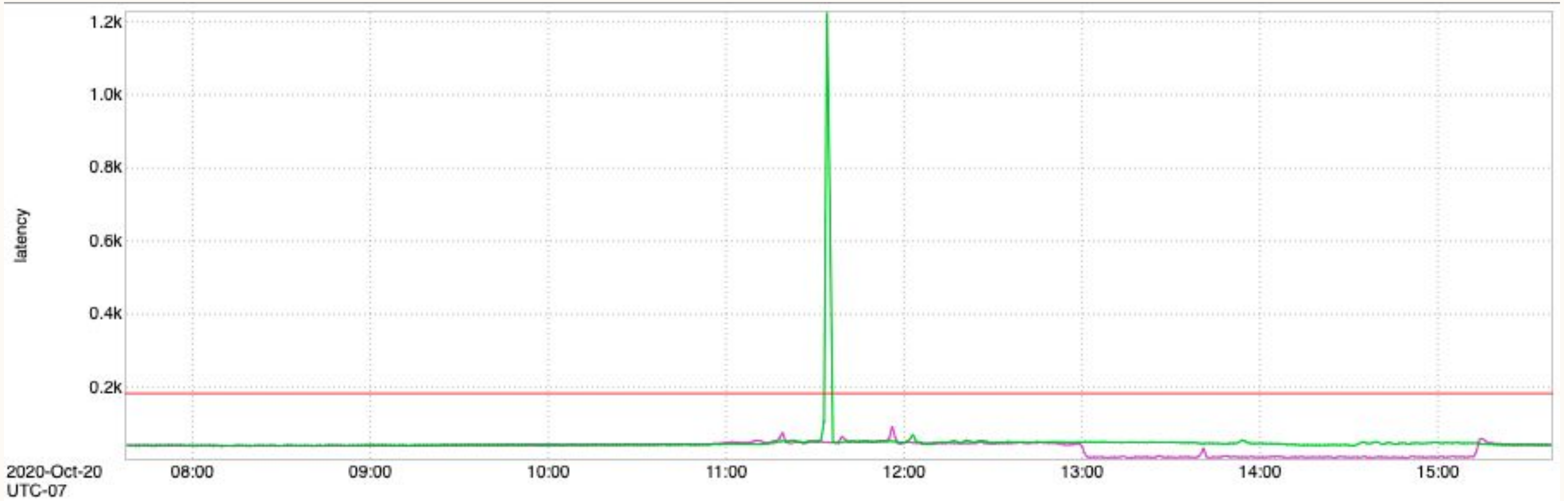
- **Service-live-abacus** feeder
- **kafka-broker-api** seek_local
- **qa-backend** Videostreamer
- **rank-echology** jobSearcher
- **rank-source** reconA
- **ocean-careers-broker** galectic-careerssearch
- **ocian-federated-search-brok** multia
- **aos-api-groups** groupsAKA
- **bla-api-jobs** Hirebit
- **zephyr-api-frontend** voyagerSearchFacets

Problem

A Real Issue


inbound-latency
25
20
15
10
5
2021-Sep
UTC-07
12 (18:00)
12 (19:00)
12 (20:00)
12 (21:00)
12 (22:00)
12 (23:00)
13 (00:00)

A Spike



latency
1.2k
1.0k
0.8k
0.6k
0.4k
0.2k
2020-Oct-20
UTC-07
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00

Correlation does not mean Causation

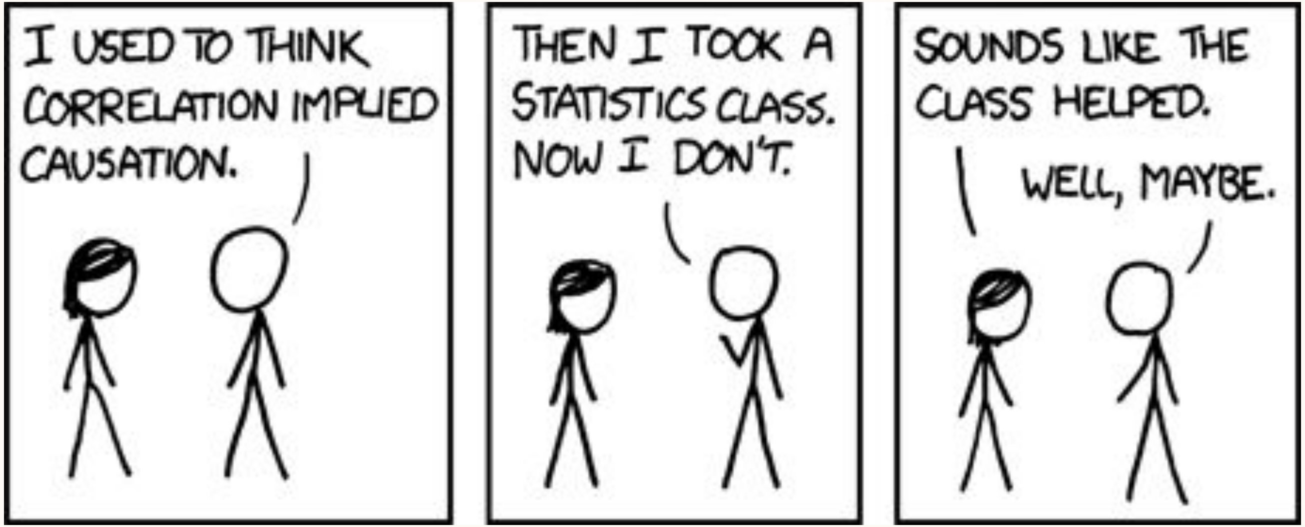

I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

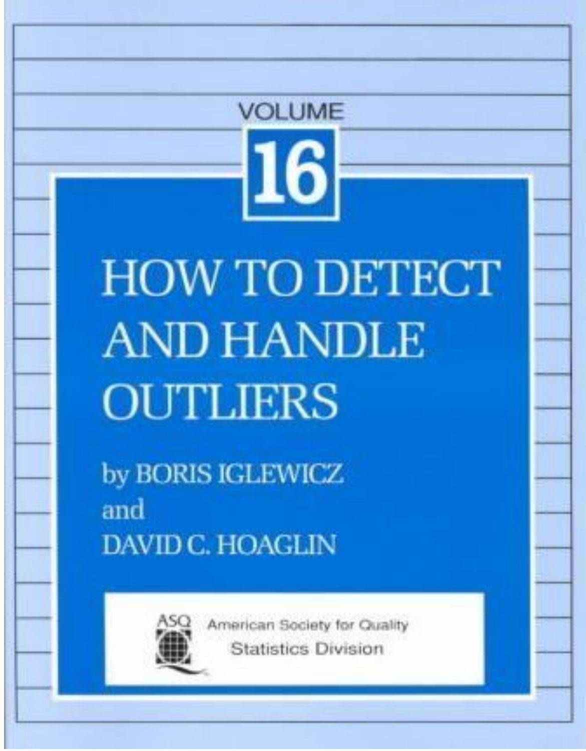
Image source - https://xkcd.com/

Problem Statement:
Finding the “right” needle in a needlestack

https://comic.browserling.com/87

Inspiration : Anomaly Detection

Dr. Boris Iglewicz, a renowned researcher and tenured faculty member within Temple University’s Fox School of Business, died Aug. 25. He was 75.



VOLUME

16

HOW TO DETECT AND HANDLE OUTLIERS

by BORIS IGLEWICZ
and
DAVID C. HOAGLIN

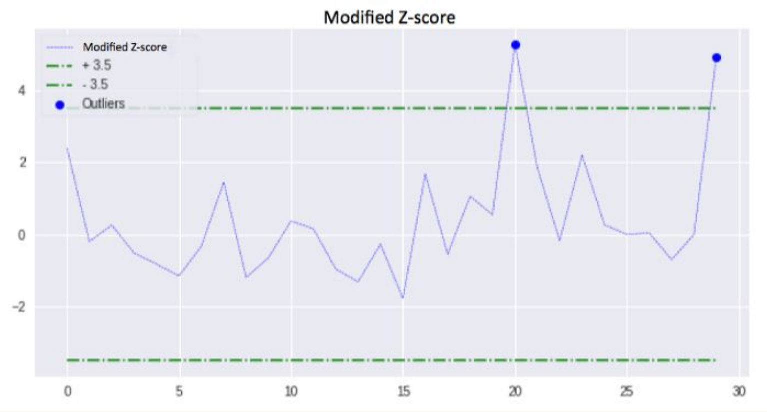
ASQ American Society for Quality
Statistics Division

Dr David Hoaglin, Currently teaches at University of Massachusetts Medical School

Modified Z-Score For Outlier Detection

$$M_i = \frac{0.6745(x_i - \tilde{x})}{MAD}$$

Iglewicz and Hoaglin recommend that modified Z-scores with an absolute value of greater than 3.5 be labeled as potential outliers.



MAD (Median Absolute Deviation)

The median absolute deviation(MAD) is a robust measure of how spread out a set of data is.

$$MAD = \text{median}\{|x_i - \tilde{x}|\}$$

MAD is a **robust statistic**, being more resilient to outliers in a data set than the standard deviation.

- In the **standard deviation**, the distances from the mean are squared, so large deviations are weighted more heavily, and thus outliers can heavily influence it.
- In the **MAD**, the deviations of a small number of outliers are irrelevant.

A Simple Example

Assume you have the following set of number 4,14,14,14,14,14,15,15,15,15,15,15

Step 1 : The median (mid value) for the above number is 14.5

Step 2: Subtract the median from each value using $|x - \tilde{x}|$:

$| 4-14.5 | = 10.5$

$|14-14.5| = 0.5$

$|14-14.5| = 0.5$

$|14-14.5| = 0.5$

$|14-14.5| = 0.5$

$|14-14.5| = 0.5$

$|15-14.5| = 0.5$

$|15-14.5| = 0.5$

$|15-14.5| = 0.5$

$|15-14.5| = 0.5$

$|15-14.5| = 0.5$

$|15-14.5| = 0.5$

Step3: Calculate Median Absolute deviation by sorting the above result and finding median:

(10.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5) = 0.5

Step4: Calculate the **Modified Z Score** $|(0.6745(x – \tilde{x}) \ / \ MAD)|$ for all the original numbers.

4 → |(0.6745(4 - 14.5)/0.5)| = 14.1645

14 → |(0.6745(14 - 14.5)/0.5)| = 0.6745

14 → |(0.6745(14 - 14.5)/0.5)| = 0.6745

14 → |(0.6745(14 - 14.5)/0.5)| = 0.6745

14 → |(0.6745(14 - 14.5)/0.5)| = 0.6745

14 → |(0.6745(14 - 14.5)/0.5)| = 0.6745

15 → |(0.6745(15 - 14.5)/0.5)| = 0.6745

15 → |(0.6745(15 - 14.5)/0.5)| = 0.6745

15 → |(0.6745(15 - 14.5)/0.5)| = 0.6745

15 → |(0.6745(15 - 14.5)/0.5)| = 0.6745

15 → |(0.6745(15 - 14.5)/0.5)| = 0.6745

15 → |(0.6745(15 - 14.5)/0.5)| = 0.6745

Step5: Anything **greater than 3.5** is a outlier

Spike Detection Challenges

1. We needed correct data points to mark outliers in 30 mins window.

2. More than one metric to work with..

3. Find outliers in near real time as recommendations are generated.

4. We wanted ~0 False Negatives

Our Approach

1. Get all the service graphs being affected due to a service-endpoint.

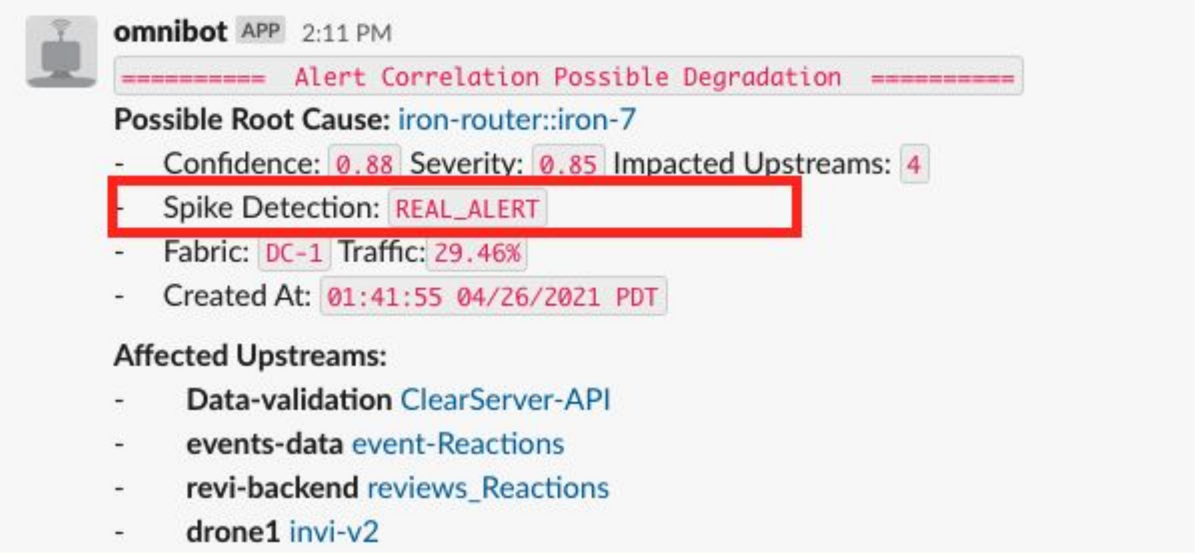
2. For each service graphs get data from Autometrics to fetch correct data points.

3. For each of the graphs you now find outliers by passing it to modified z-score algorithm.

4. Clean , Combine the data from each of graph for final decision making

5. Once you have outlier data we need to take decision as follows:
    a. If you find any graph with no spikes classify it to be a REAL ALERT
    b. In case we find 5 spike data points to be consecutive and around 70 % of all the graph are having same trends, we will call it a REAL ALERT
    c. Anything Less than 70% is a SPIKE

Results - Real Alerts



omnibot APP 2:11 PM

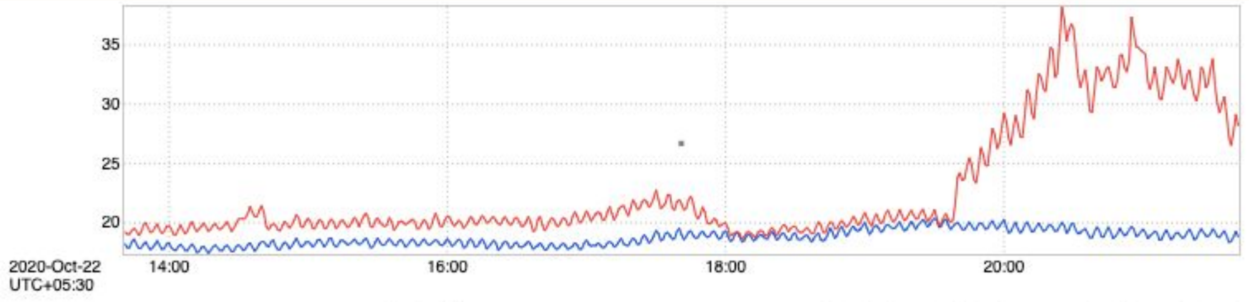
`==========  Alert Correlation Possible Degradation  ==========`

Possible Root Cause: iron-router::iron-7

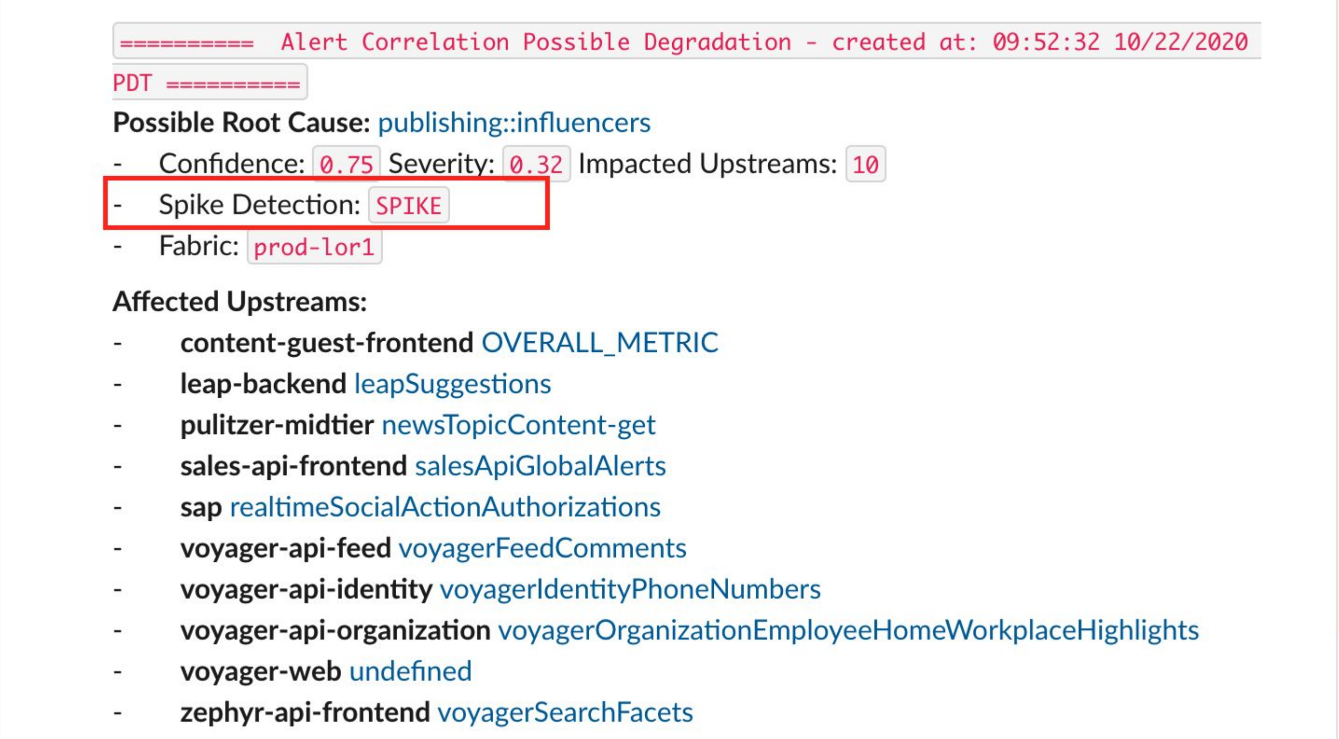
- Confidence: `0.88` Severity: `0.85` Impacted Upstreams: `4`
- Spike Detection: `REAL_ALERT`
- Fabric: `DC-1` Traffic: `29.46%`
- Created At: `01:41:55 04/26/2021 PDT`

Affected Upstreams:

- **Data-validation** ClearServer-API
- **events-data** event-Reactions
- **revi-backend** reviews_Reactions
- **drone1** invi-v2



Results - Spikes



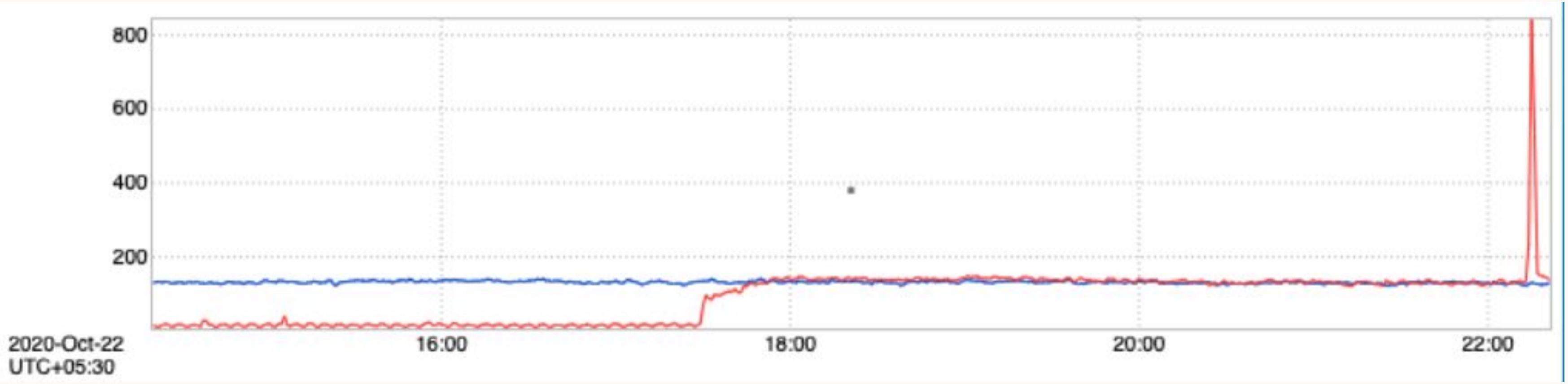
`==========  Alert Correlation Possible Degradation - created at: 09:52:32 10/22/2020 PDT ==========`

Possible Root Cause: publishing::influencers

- Confidence: `0.75` Severity: `0.32` Impacted Upstreams: `10`
- Spike Detection: `SPIKE`
- Fabric: `prod-lor1`

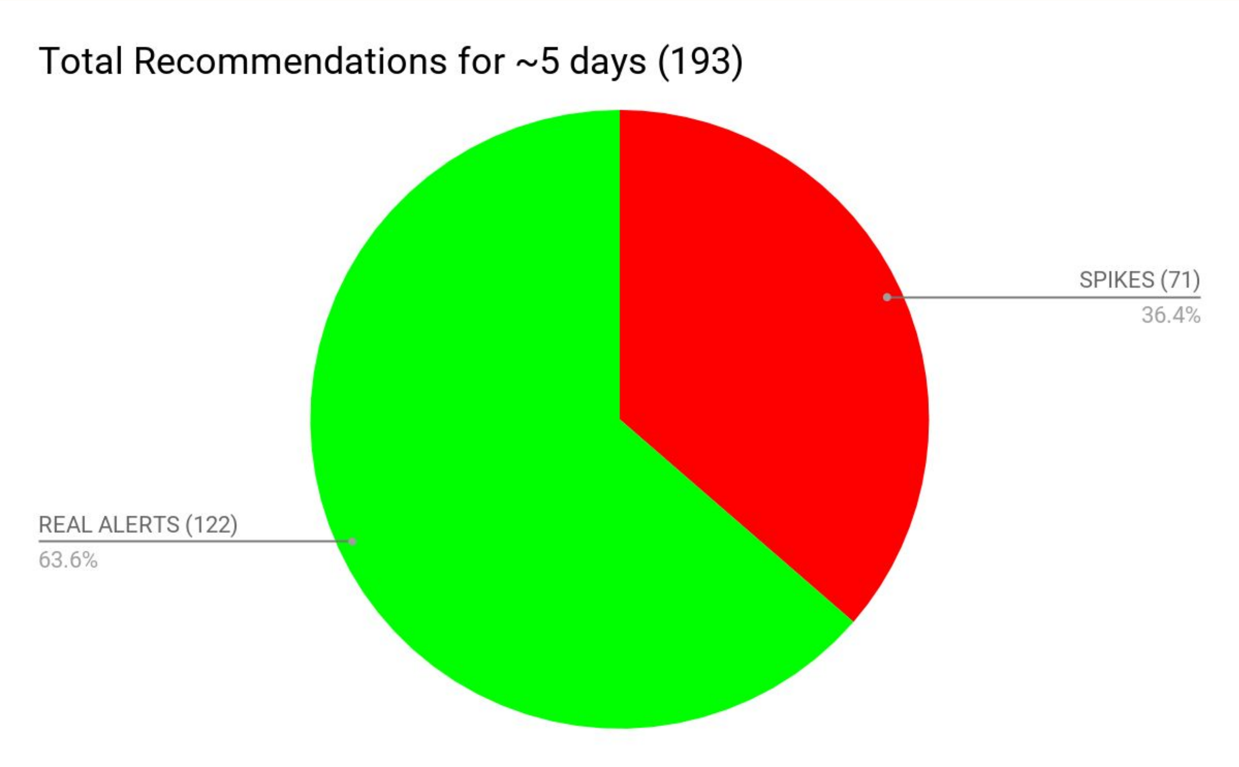
Affected Upstreams:

- **content-guest-frontend** OVERALL_METRIC
- **leap-backend** leapSuggestions
- **pulitzer-midtier** newsTopicContent-get
- **sales-api-frontend** salesApiGlobalAlerts
- **sap** realtimeSocialActionAuthorizations
- **voyager-api-feed** voyagerFeedComments
- **voyager-api-identity** voyagerIdentityPhoneNumbers
- **voyager-api-organization** voyagerOrganizationEmployeeHomeWorkplaceHighlights
- **voyager-web** undefined
- **zephyr-api-frontend** voyagerSearchFacets


800
600
400
200
2020-Oct-22
UTC+05:30
16:00
18:00
20:00
22:00

Results: Spike vs Real

Less than 1 % False Positives



Conclusion

- Simple statistics without any ML solved our problem.
- Follow Occam's razor for problem solving .
- Reduced toil by 30-40%.

Thank you