Scout: Improving the diagnosis process through domain-customized incident routing

Jiaqi Gao, Nofel Yaseen, Robert MacDavid, Felipe Vieira Frujeri, Vincent Liu, Ricardo Bianchini, Ramaswamy Aditya, Xiaohang Wang, Henry Lee, David Maltz, Minlan Yu, Behnaz Arzani
Availability and maintaining service level objectives (SLOs) are the biggest challenges facing cloud operators today.
Incidents can and do happen

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of public incidents</td>
<td>69</td>
<td>21</td>
</tr>
<tr>
<td>between February to July 2020</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum resolution time</td>
<td>14 h 12 m</td>
<td>19 h 49 m</td>
</tr>
<tr>
<td>Average resolution time</td>
<td>4 h 40 m</td>
<td>5 h 28 min</td>
</tr>
</tbody>
</table>
Life cycle of an incident

REPORT INCIDENT
- Monitoring system
- Customer reports

FIND THE FAILING COMPONENT
- Find right team
- Check monitoring systems
- Can we fix it?

DIAGNOSE AND FIX PROBLEM
- Find problematic device
- Understand the cause
- Fix
Finding the right team is time consuming

CDF

Time (normalized)

Multiple teams investigate

Single team investigates

FIND THE FAILING COMPONENT

Find right team
Check monitoring systems
Can we fix it?
Example incident: storage problem

1. Can’t write to storage!
2. Must be storage issue
3. Storage is good, network must be slow
4. No congested links
5. Need more information from customer
6. Connection fail to init, SLB failing
7. SLB is good, network must be dropping
8. Packet is reaching to SLB
9. Customer opens too many connections and exhaust SNAT pool, behavior is expected
Why multiple teams get involved?

Studied 200 misrouted incidents in Azure
1. Lack of domain knowledge

- Storage team doesn’t know network is functioning or not
2. No cloud teams are responsible, more misrouting

- ISP or customer outside the cloud is experiencing issues
3. Concurrent incidents

- One failure causes multiple incidents in multiple teams
How to reduce misrouting?
Existing solutions

**Application specific diagnosis system**

- NetPoirot [SIGCOMM-16]
- DeepView [NSDI-18]
- Sherlock [SIGCOMM-07]

**Natural language processing**

- NetSieve [NSDI-13]

Too many applications in the data center

Ignores essential domain knowledge
Incident routing problem revisit

- Incident
- Domain knowledge
- Monitor data
- Machine learning?
- SLB
- Network
- Storage
Solve the whole problem at once?

▷ Hard to build a single, monolithic incident routing system

- **Curse of dimensionality**
  Huge feature vector with no enough training examples

- **Uneven instrumentation**
  A subset of teams will always have gaps in monitoring

- **Constantly changing**
  Stale components and monitors

- **Limited visibility**
  Hard to understand appropriate feature set for each team
Scout: team-specialized ML-assisted gatekeeper

▷ “Is my team responsible for the incident?”

One team, one scout
Leverage domain knowledge
Evolve independently
Scout design

Incident → Domain Knowledge → Monitor data → Computation engines → Classification result
Physical networking team

**Scope**
Every switch & router in DC

**Monitors**
PingMesh, Everflow, NetBouncer, etc.

**Statistics**
58% incidents investigated by PhyNet went through multiple teams
97.6 hours per day wasted on unnecessary investigations
CHALLENGE 1

How to process huge amount of monitoring data?

Millions of devices in the Cloud
Incident guided investigation

Incident Description

“Server X.c10.dc3 is experiencing problem connecting to storage cluster c4.dc1”

Regular Expression

Server: X.c10.dc3
Cluster: c4.dc1

Devices

Monitor data

CPU Usage
Link loss rate
Ping latency
CHALLENGE 2

How to create a feature vector out of the monitoring data?

Variable number of devices  Mixed types of monitoring data
How to build a fixed width feature vector?

▷ Per-component feature
  ○ Event: count number of events
  ○ Time-series: normalize and calculate statistics (percentiles, average, etc.)

▷ Multiple components
  ○ Compute statistics across multiple components (percentiles, average, etc.)
CHALLENGE 3

▸ Which computation engine?
Supervised learning: random forest

- Learns based on history incidents, high accuracy
- Low accuracy on new incidents
- Interpretable, able to provide more insights
Change point detection for new incidents
Change point detection for new incidents

- Easy to compute
- Higher accuracy on new incidents
- Low accuracy on old incidents
Model selector

- Incident itself tells whether it is new or not
- Use meta-learning to identify new incidents
The anatomy of a Scout

Incident -> Configuration file -> Model Selector

Model Selector: Incident -> Monitor data

Monitor data: Random Forest -> Change point detection

Random Forest, Change point detection -> Computation engines

Computation engines -> Classification result
Evaluation
**Evaluation setup**

<table>
<thead>
<tr>
<th><strong>DATASET</strong></th>
<th><strong>LABEL</strong></th>
<th><strong>BASELINE</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>9 months of incidents in Azure</td>
<td>Whether incident is resolved by PhyNet</td>
<td>Current incident routing system without Scout</td>
</tr>
<tr>
<td>Randomly split into training and testing set</td>
<td></td>
<td>Runbooks, past-experience, NLP based routing system</td>
</tr>
</tbody>
</table>
## Overall performance

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline</strong></td>
<td>87.2%</td>
<td>91.9%</td>
<td>0.89</td>
</tr>
<tr>
<td><strong>PhyNet Scout</strong></td>
<td>97.5%</td>
<td>97.7%</td>
<td>0.98</td>
</tr>
<tr>
<td><strong>Delta</strong></td>
<td>10.3%</td>
<td>5.8%</td>
<td>0.09</td>
</tr>
</tbody>
</table>

10% improvement means?
Gain in of the PhyNet Scout

Gain in
Send incident to PhyNet directly

Save more than 20% of the total investigation time in 40% of incidents
Gain out of the PhyNet Scout

Gain out
Reject incident to PhyNet

Close to the best possible gain
Things we did not talk about

▷ The design and evaluation of Scout Master

▷ Extended evaluation
  ○ System performance over time
  ○ Sensitivity analysis
  ○ Other supervised learning algorithms

▷ Lessons we learnt from deployment

Please check our paper for more detail
Conclusion

- Incident misrouting is the main challenge for maintaining service level objectives in the cloud.
- Scout: a distributed team-specialized gate-keeper can reduce investigation time.
Thanks!