Draining the flood
a combat against alert fatigue

Yu Chen
The Alert Flood in Baidu

- the amount of alerts is high
  - More than 100 alerts per person per day
    - Day time: ~75% in 17 hours
    - Night time: ~25% in 7 hours

- Highly Redundant
  - # effective alerts / # alert SMS < 0.15
## Observations & Solutions

<table>
<thead>
<tr>
<th>Observation</th>
<th>Reason</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicate ratio: 58%</td>
<td>• Persistent alerts</td>
<td>• Alert grouping</td>
</tr>
<tr>
<td></td>
<td>• Correlated alerts</td>
<td></td>
</tr>
<tr>
<td>Attention ratio: 25% (at night time)</td>
<td>• Over-aggressive alert importance</td>
<td>• Alert importance level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Delivery behavior</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Level calibration</td>
</tr>
<tr>
<td>Receivers per alert: 3</td>
<td>• In-effective oncall procedure</td>
<td>• Oncall schedule and escalation</td>
</tr>
<tr>
<td>Single instance alerts: 88%</td>
<td>• &gt; 40% only requires simple operations to recover</td>
<td>• Automatic self-healing</td>
</tr>
</tbody>
</table>
Alert Grouping

• Simple grouping
  – Remove simple duplicates

• Cross-module patterns
  – Reveal underlying issues

• Network connectivity detection
  – Suppress alert surge
Simple Grouping

- Grouping based on natural dimensions
  - Alert rule name
  - Deployment structure
    - Product, Module, Cluster, Instance
    - Datacenter, machine
Grouping Result

{group.ab-zxcvq.AB.all:instance:B_zxcvq_FATAL}{总体异常实例比例:1.36054%}{异常 (2):0.opr-zty5-zxcvq-000-cc.AB.bjdc,1.opr-zty5-zxcvq-000-cc.AB.bjdc}{05-02 16:49:36 - 16:54:09} {http://dwz.cn/... }

- **Rule name**
  - group.ab-zxcvq.AB.all:instance:B_zxcvq_FATAL
  - Instance level alert

- **Ratio**
  - 1.36054%

- **Instance list**
  - 0.opr-zty5-zxcvq-000-cc.AB.bjdc
  - 1.opr-zty5-zxcvq-000-cc.AB.bjdc

- **Time**
  - 05-02 16:49:36 - 16:54:09

- **Link to detail page**
  - http://dwz.cn/...
### Delivery with Grouping

#### Alert Source

<table>
<thead>
<tr>
<th>Alert info</th>
<th>Fire time</th>
<th>Linger time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: rule1</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>A: rule2</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>B: rule3</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>A: rule1</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>C: rule4</td>
<td>25</td>
<td>60</td>
</tr>
</tbody>
</table>

#### Linger Buffer

#### Delivered Alert

- A:rule1
- A:rule2
- A:rule1
Cross-Module Patterns

- **Caller / Callee**
  - Both alerts when callee is in trouble

- **Association rule mining**
  - Transaction window starting from every alert
Network Connectivity

- Network device failure can cause a lot of alerts
- Should trigger alerts for
  - Most rules
  - Most products
- Heuristic rule: 
  \[
  \text{score} = \sqrt{\frac{\text{# alerting rules}}{\text{total rules}}} \cdot \frac{\text{# alerting products}}{\text{total products}}
  \]
Linger Time

- Configurable
  - Different among alert rules
- Extra delay to receive alerts
  - Less punctual
- Need better ways to balance
Attention Ratio

- Check existence in interval $[t_{\text{alert}}, t_{\text{alert}} + t_{\Delta}]$
  - Access log of the monitoring system
    - View alert detail
    - View relevant curves
  - Login log of the production machine
- Exist: alert is attended
- Absent: alert is ignored
- Only applied to night time
Alert Calibration

- Importance levels
  - Critical: SMS + Phone to all receivers
  - Major: SMS + Escalation
  - Warning: SMS without Escalation
  - Notice: Mail

- Attention ratio should be compatible to levels
  - Push from managers
Alert Receivers

- Typical receivers of an alert
  - Primary oncall engineer
  - Secondary oncall engineer
  - Oncall engineer lead
  - Senior engineer
  - Manager
- Primary oncall engineer handles alerts usually
  - But alerts always sent to all
Oncall Escalation

- Alerting stages
  - One fixed stage
    - primary, secondary
  - Zero or more escalation stages

![Diagram showing the alerting stages and escalation stages with a and b minutes]
Oncall Escalation

Oncall schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Receiver</th>
<th>Unclaimed after(min)</th>
<th>Unsolved after(min)</th>
<th>Call phones</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-3</td>
<td>#z_ODA_lihui</td>
<td>5</td>
<td>10</td>
<td>yes</td>
</tr>
<tr>
<td>5-12</td>
<td>oda</td>
<td>10</td>
<td>20</td>
<td>yes</td>
</tr>
</tbody>
</table>

Fixed stage

Escalation stage
Automatic Self-healing

- Lazy log purge
  - Set an alert on disk free space
  - Delete some log when alert triggers
- Granularity
  - Instance level
    - "bin_control restart"
  - Module/Cluster level
    - "curl master.a.com"
- Alert
  - will not deliver
  - view alert log
Management Support

- Alert importance calibration
  - Lower importance level
- Oncall escalation
  - Include attention ratio into work evaluation
Decrease by 85%

Number of Alerts / Weekly

- Total
- DayTime
- Night
Remarks

- Reducing redundant alerts
  - Mining alert correlation for grouping
  - Estimate attention ratio for importance calibration
  - Receiver escalation mechanism
  - Alert self-healing mechanism
- Helpful on understanding root causes of issues
THANKS

chenyu07@baidu.com