NetRadar: Monitoring the datacenter network

Yun Chen
About me

• Working in Baidu
• Focus on AIOps
  • Datacenter network monitoring
  • Time-series anomaly detection
  • Service diagnosis
Outline

• Motivation: Datacenter network monitoring is important
• Method: The design of NetRadar
• Summary
The internal traffic is massive

Traffic distribution in Baidu

Traffic distribution in Facebook
Baidu network architecture is complex

- More than 30 datacenters
- Tens of thousands of network devices
- Hundreds of thousands of servers
Network failure is common

- Device failure happens most of time
  - Device's availability is 99.99% (52 min a year)
  - Probability of failure-free: $0.9999^{10,000} \approx 0.37$
- Redundancy can't avoid network failure entirely
  - Route convergence takes time
  - Congestion due to oversubscription
Outline

• Motivation: Datacenter network monitoring is important
• Method: The design of NetRadar
  • Measurement
  • Data analysis
• Summary
End-to-end monitoring is significant to SRE

• SRE cares about network issues that applications perceived
• Device monitoring helps little to SRE
  • Device availability != application availability
    • Device fails but applications work well: redundancy
    • Application perceives network issue but devices are healthy: congestion
      • 36% issues are caused by congestion

• End-to-end probing meets SRE's need
  • Perceives the same network issues with applications
Sampling is necessary in the Baidu scenario

- Simple way costs too much
  - Method: probe between every possible pair of servers
  - Network cost: 533Mbps bandwidth for every server
    - Hundreds of thousands of servers
    - Probing every 30 seconds
    - Probing message size: 10KB
    - Cost: \(100,000 \times 2 \times 10KB \div 30s = 533Mbps\)

- Requirement of sampling
  - Sufficient samples for every network path
Hierarchical and evenly distributed sampling to ensure sufficiency

- Hierarchical probing
  - Consider network architecture

- Samples evenly distributed across the paths
  - Assign similar amount samples for each path
Data analysis is needed to close the gap between observations and SRE's concern

• End-to-end probing results
  • Reachability between server pairs

<table>
<thead>
<tr>
<th>Level</th>
<th>SrcUnit</th>
<th>DstUnit</th>
<th>SrcIp</th>
<th>DstIp</th>
<th>Reachable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToR</td>
<td>ToR 1</td>
<td>ToR 2</td>
<td>SIP₁</td>
<td>DIP₁</td>
<td>Success</td>
</tr>
<tr>
<td>Cluster</td>
<td>Cluster 1</td>
<td>Cluster 2</td>
<td>SIP₂</td>
<td>DIP₂</td>
<td>Fail</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Datacenter</td>
<td>Datacenter 1</td>
<td>Datacenter 2</td>
<td>SIPₙ</td>
<td>DIPₙ</td>
<td>Success</td>
</tr>
</tbody>
</table>

• SRE's concern
  • ToR or Cluster or Datacenter failure?
  • Failure between Clusters or Datacenters?
Simple idea: detection for every possible failure SRE concerned

- Enumerate all failures for given architecture
- Detect for every possible failure separately
  - An example: detection for Cluster 1
    - Gathering all probing result
      - SrcUnit = Cluster 1 or DstUnit = Cluster 1
    - Count probing times(n) and drop times(k)
    - Is k too large for given n?
      - Errors detection
      - Alert condition: \( P(X \geq k) \) is very small

\[
P(X = k) = \binom{n}{k} (1 - p_0)^{n-k} p_0^k
\]

\[
P(X \geq k) = \sum_{i=k}^{n} P(X = i)
\]
Detection separately triggers false positive

- Issue introduction
  - Cluster 1 failure with 25% drop rate
- Detect Cluster 1 failure successfully
  - Drop rate of Cluster 1 increased sharply, Alert

- False positive: Datacenter 1 failure
  - Half of samples in Datacenter 1 are from Cluster 1
  - Drop rate of Datacenter 1 increased because of Cluster 1 failure
Multi-dimensional data analysis helps to eliminate false positives (1)

- Label probing result with tags
  - Tags name: SrcIp, SrcToR, SrcCluster, SrcDatacenter, DstIp, DstToR, DstCluster, DstDatacenter
  - Tags value: depend on SrcIp, DstIp

- Multi-dimension data
  - Dimension: tag name
  - Dimension value: tag value

<table>
<thead>
<tr>
<th>Dimension combination</th>
<th>Reachable</th>
</tr>
</thead>
<tbody>
<tr>
<td>SrcIp=SIP₁ &amp; SrcToR=ToR₁ &amp; SrcCluster=Cluster₁ &amp; SrcDatacenter=Datacenter₁ &amp; DstIp=DIP₁ &amp; DstToR=ToR₃ &amp; DstCluster=Cluster₂ &amp; DstDatacenter=Datacenter₁</td>
<td>Success</td>
</tr>
</tbody>
</table>
Multi-dimensional data analysis helps to eliminate false positives (2)

- Failure can be represented by dimension combination
  - \( \text{SrcCluster} = \text{Cluster}_1 \& \text{DstCluster} = \text{Cluster}_2 \)
- Feature of failure dimension combinations
  - High contribution
  - High uniformity

Cluster 1 failure with 25% drop rate
An efficient multi-dimensional data analysis algorithm based on decision tree

- Select the most distinguishing dimension
  - Hellinger distance

- Stop condition for splitting
  - Uniformly distributed
  - Little drops
Summary

• Measurement
  • End-to-end probing is used in consideration of SRE's need
  • Sampling is necessary in the Baidu scenario
    • All possible paths are evenly distributed

• Data analysis
  • Detection for every failure separately may trigger false positive
  • Multi-dimensional data analysis can eliminate false positive
    • Basic idea of multi-dimensional data analysis
    • An efficient multi-dimensional data analysis algorithm
Thanks