NetAssistant: Dialogue Based Network Diagnosis in Data Center Networks

Haopei Wang, Anubhav nidhi Abhashkumar, Changyu Lin, Tianrong Zhang, Xiaoming Gu, Ning Ma, Chang Wu, Songlin Liu, Wei Zhou, Yongbin Dong, Weirong Jiang, Yi Wang

Network Engineering Team
Motivation
NetAssistant Design
Deployment & Gain
Lessons & Future Work
Motivation

High visibility is important in datacenter networks

Especially for network users

Network oncalls handler requests/tickets

Time consuming and labor intensive
Monitoring data, incidents records

There is a gap!
NetAssistant Design

Our Idea: Leverage a task-oriented dialogue system

Automatically answers diagnosis questions

Three layers of abstraction

1. Chat service
   - Dialogue Engine
2. Diagnosis workflows
   - Workflow Engine
3. Retrieval of monitoring data
   - Data Engine
NetAssistant Design

Workflow Engine:

- Diagnosis and troubleshooting experience
- Query monitoring data, Detect anomaly, Make decision, Perform operations
- Knowledge Experience
- Atomic Functions
- Troubleshooting Guide
- Workflow Converter
- Executable Workflows
NetAssistant Design

Dialogue Engine:

A typical task-oriented dialogue system

What kind of question is this?

Server network health check?
Switch config check or status?
Any incident related to a computing application?
NetAssistant Design

Data Engine:

Performance bottleneck
- Query monitoring data
- E.g., sFlow for 100 links
- E.g., syslog for 1000 switches

Our Idea:
Only anomalies/jitters are important
We combine:
- Reactive/on-demand querying
- Proactive alerting
NetAssistant Design

Sample dialogues between NetAssistant and the users:

Network User → NetAssistant:
- How was the network last night at 9pm?
- Please provide a name of the network:
  - Santa Clara data center
- Healthy. Connectivity: v4, v6, ISP, ...
- Traffic: total, QoS, ... Routing: ...

Network User ← NetAssistant:
Deployment & Gain

First version was launched in April 2020

- Starting with only 2 workflows, now 100+
- Iterate on technology and functionality every few weeks
- Now ~200 uses per day

What is the gain of this project?

- Save human labor time
- Directly intercept oncalls
- Reduce oncall duration time
Deployment & Gain

Intercepted a considerable proportion of oncalls in 2023

Reduce oncall duration time
Lessons & Future Work

Lessons:
• Earn user trust
• FN is more harmful than FP
• Empowering our users

Future work:
• Explore the potential of LLM in AIOps
• Challenges:
  Understanding diagnosis logic
  Processing real time monitoring data
THANKS
### Table 3: Monitoring primitives and data volume

<table>
<thead>
<tr>
<th>Monitoring Primitive Category</th>
<th>Data Volume per Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connectivity (e.g., PingMesh [18], EverFlow [27], etc.)</td>
<td>65GB</td>
</tr>
<tr>
<td>Traffic (sFlow, SNMP, etc.)</td>
<td>12TB</td>
</tr>
<tr>
<td>Switch Syslog</td>
<td>35GB</td>
</tr>
<tr>
<td>Host Monitoring</td>
<td>4.3GB</td>
</tr>
<tr>
<td>Routing Configuration</td>
<td>425G</td>
</tr>
<tr>
<td>Optical Module (DDM or DOM)</td>
<td>5.5GB</td>
</tr>
<tr>
<td>Other Monitoring Primitives</td>
<td>27GB</td>
</tr>
</tbody>
</table>
### Table 4: Commonly used workflows

<table>
<thead>
<tr>
<th>Workflows</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>check_pod_network</td>
<td>Data center level network status workflows, including connectivity (internal, external, overlay, underlay, v4, v6, subnets and etc.), bandwidth &amp; utilization (different types of links, different granularity), switches and existing network incidents &amp; changes.</td>
</tr>
<tr>
<td>check_az_network</td>
<td>IP level network status workflows, including software stack check, hardware status check, network environment (nearby switches) check.</td>
</tr>
<tr>
<td>check_idc_network</td>
<td>Switch health status check, including metrics from switch OS (syslog), protocol (e.g., SNMP, BMP), hardware (e.g., linecard, OTN) and external monitors.</td>
</tr>
<tr>
<td>check_region_network</td>
<td>Physical link level status workflows, mainly used by network team, including physical metrics, traffic and protocol status checking.</td>
</tr>
<tr>
<td>check_storage_service</td>
<td>Network service level workflow, checking network status of involved servers, upstream and downstream network traffic, QoS management and etc.</td>
</tr>
<tr>
<td>check_computing_service</td>
<td></td>
</tr>
<tr>
<td>check_p4_network</td>
<td></td>
</tr>
</tbody>
</table>
## Appendix

<table>
<thead>
<tr>
<th></th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FP</strong></td>
<td>9.48%</td>
<td>12.33%</td>
<td>11.6%</td>
<td>10.63%</td>
<td>9.62%</td>
<td>8.45%</td>
</tr>
<tr>
<td><strong>FN</strong></td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0.43%</td>
<td>1.25%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 5: Accuracy Evaluation Results of **NetAssistant**
(a) Daily Oncall Usage for July and August in 2023 (CDF)  
(b) The 50th Percentile Usage by Day of the Week

Figure 8: Daily Usage Results of NETASSISTANT