UNIVERSITÀ DI PARMA

Enhancing Fault Isolation in Hardware Systems Using Large Language Models (LLMs)

Matteo Gianvenuti

Accademic year 2024/2025

Reference

MBDA

- Internship thesis program
- MBDA is a European multinational corporation, specialized in defense

Thesis' objectives

- Explore the use of LLMs in hardware diagnostic to analyse diagnostics test results, identify any possible anomalies and try to find their possible causes
- In an offline working environment characterized by challenging limited resources without the support of cloud infrastructures or high-end GPU, like the resources available on computer laptop

Reference

Operative steps

Step One – Evaluating LLM Understanding of the System Schema

- Evaluate the SOTA in hardware diagnostics
- Assess whether the LLM can comprehend the hardware system structure and interconnections
- Determine whether a tabular, Json or descriptive representation is more effective to prepare the model
- Test whether the model can infer dependencies between components and recognize interconnection patterns
- Validate whether the model accurately understands the system and its relationships before optimizing its diagnostic capabilities

Reference

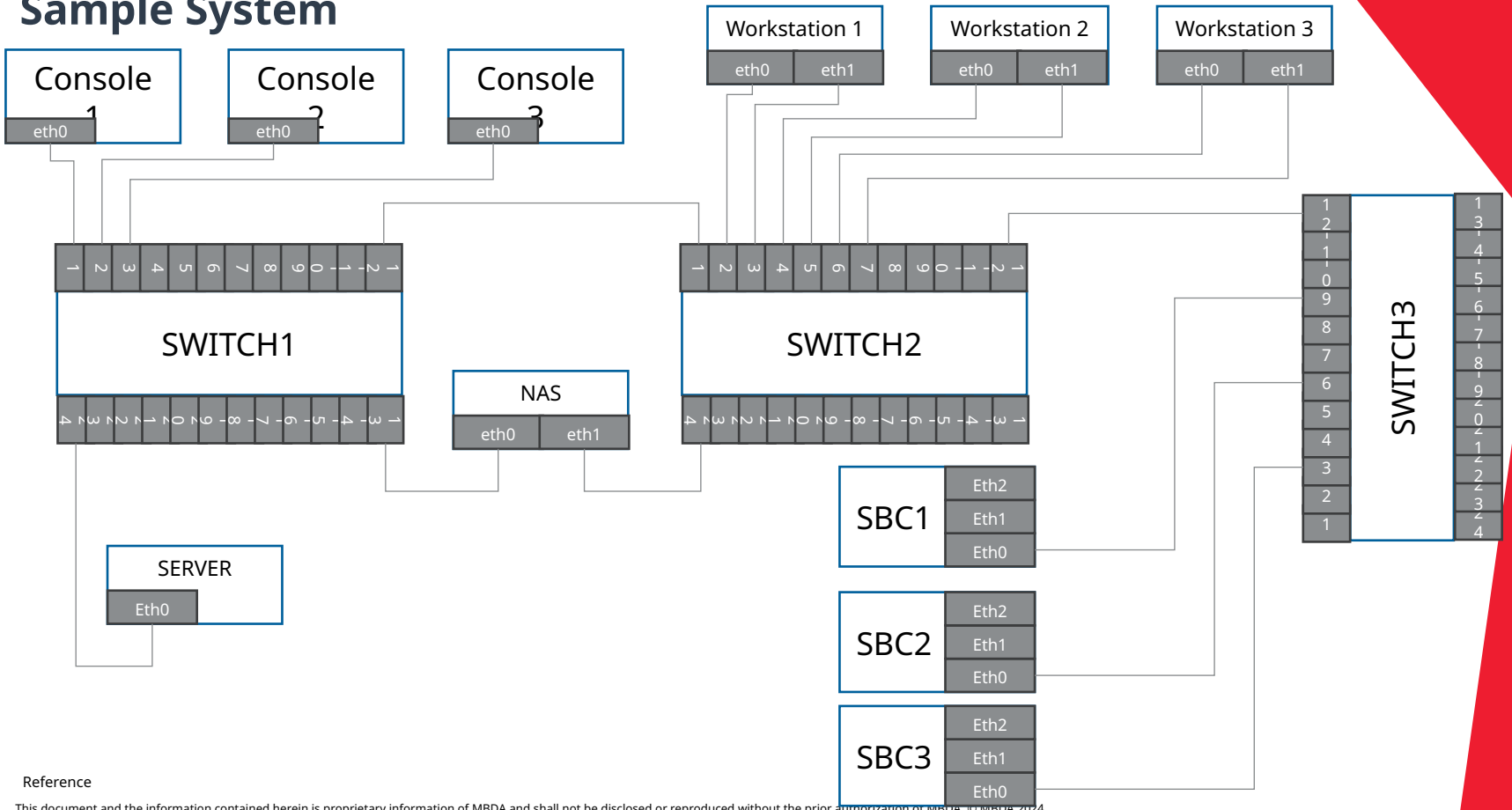
Operative steps

Step Two – Evaluating the LLM ability of analyse logs and telemetry data generated by the diagnostic software

- Use a simulator to generate structured data tailored to system diagnostics. Design realistic fault scenarios to feed into the LLM
- Use the LLM to identify anomalies and their possible causes

Reference

Sample System



Reference

Step One – Preliminary Activities

State of the Art & Literature Review

- Analysis of academic publications, technical articles, and white papers on:
  - Hardware diagnostics
  - LLMs applied to unstructured data analysis
  - LLMs for fault detection
  - Anomaly detection and fault isolation techniques
- Fault isolation, identification of best practices and current limitations in existing solutions

Prompt Engineering

- In-Context Learning to teach the system architecture and how to reply

Technological Evaluation

- Comparison of existing LLMs:
  - Llama3.2-3B-Instuct
  - Phi-4-mini-reasoning
  - Mistral-7B-Instruct-V0.3

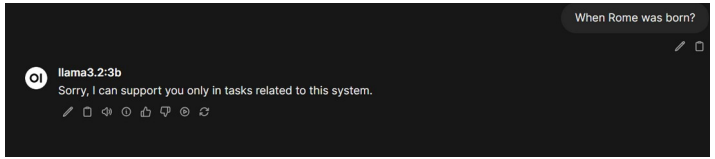
Reference

Step One – Prompt Engineering and ICL

```
Your task:
You will be given a network topology. Your job is to understand the structure and
answer the questions that follow about the system and its architecture.
Use the format shown in the examples to give your answers, do not add additional,
inconsistent or unrelated information.
```

```
Naming rules: All components with numbers in their names (e.g., Switch-1, Console-3,
SBC-2) are distinct devices. The number is critical for identification.
- Multi-device components: Use '{device}-{device number}' (e.g., Switch-1,
Console-3).
- Single device components: Use the base name (e.g., Server, NAS).
- Network interfaces: Append '-{interface}' (e.g., Workstation-1-eth0, NAS-eth1).
- Switch ports: Use '{Switch}-{switch number}-port-{port number}' (e.g., Switch-1-
port-14, Switch-3-port-7).

Network components:
Switch-1, Switch-2 and Switch-3 with 24 ports each.
Console-1, Console-2 and Console-3 each with its own network interface eth0.
Workstation-1, Workstation-2 and Workstation-3 each with its own two network
interfaces eth0 and eth1.
Server with its own network interface eth0.
NAS with its two network interfaces eth0 and eth1.
SBC-1, SBC-2 and SBC-3 each with its own network interfaces eth0, eth1 and eth2.
```

```
Q&A examples:
What is connected to Switch-2-port-5?
The Workstation-2-eth1.

What is connected to Console-3-eth0?
The Switch-1-port-3.

What is connected to NAS-eth1?
The Switch-2-port-24.

What is connected to Switch-3-port-6?
The SBC-2-eth0.

What is connected to Workstation-3-eth2?
The eth2 network interface does not exist for Workstation-3.

What is connected to SBC-2-eth1?
There is nothing connected to SBC-2-eth1.

What is connected to Switch-2-port-12?
The Switch-3-port-12.

What is connected to Switch-1-port-14?
There is nothing connected to Switch-1-port-14.

<Any other sentence/phrase/question not related to this system>
Sorry, I can support you only in tasks related to this system.
```



Reference

Step One – System Representation

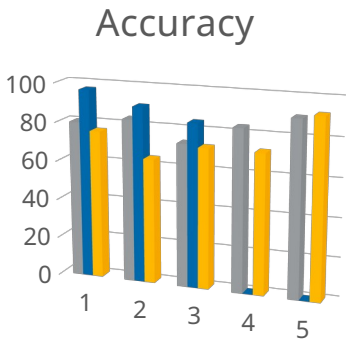
```
Network topology:
The Console-1-eth0 is connected to Switch-1-port-1.
The Console-2-eth0 is connected to Switch-1-port-2.
The Console-3-eth0 is connected to Switch-1-port-3.
The Server-eth0 is connected to Switch-1-port-24.
The NAS-eth0 is connected to Switch-1-port-13.
The NAS-eth1 is connected to Switch-2-port-24.
The Switch-1-port-12 is connected to Switch-2-port-1.
The Switch-3-port-12 is connected to Switch-2-port-12.
The Workstation-1-eth0 is connected to Switch-2-port-2.
The Workstation-1-eth1 is connected to Switch-2-port-3.
The Workstation-2-eth0 is connected to Switch-2-port-4.
The Workstation-2-eth1 is connected to Switch-2-port-5.
The Workstation-3-eth0 is connected to Switch-2-port-6.
The Workstation-3-eth1 is connected to Switch-2-port-7.
The SBC-1-eth0 is connected to Switch-3-port-9.
The SBC-2-eth0 is connected to Switch-3-port-6.
The SBC-3-eth0 is connected to Switch-3-port-3.
All unspecified device, ports and interfaces are not connected to any device.
```

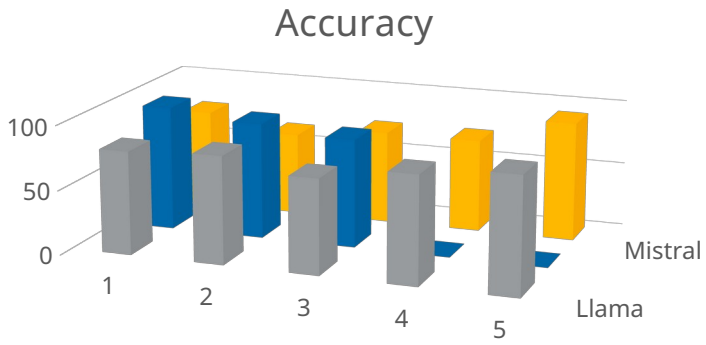

```
{
  "devices": [
    {
      "name": "Switch-1",
      "type": "Switch",
      "interfaces": [
        { "name": "port-1", "connected_to": { "device": "Console-1", "interface": "eth0" } },
        { "name": "port-2", "connected_to": { "device": "Console-2", "interface": "eth0" } },
        { "name": "port-3", "connected_to": { "device": "Console-3", "interface": "eth0" } },
        { "name": "port-4", "connected_to": null },
        { "name": "port-5", "connected_to": null },
        { "name": "port-6", "connected_to": null },
        { "name": "port-7", "connected_to": null },
        { "name": "port-8", "connected_to": null },
        { "name": "port-9", "connected_to": null },
        { "name": "port-10", "connected_to": null },
        { "name": "port-11", "connected_to": null },
        { "name": "port-12", "connected_to": { "device": "Switch-2", "interface": "port-1" } },
        { "name": "port-13", "connected_to": { "device": "NAS", "interface": "eth0" } },
        { "name": "port-14", "connected_to": null },
        { "name": "port-15", "connected_to": null },
        { "name": "port-16", "connected_to": null },
        { "name": "port-17", "connected_to": null },
        { "name": "port-18", "connected_to": null },
        { "name": "port-19", "connected_to": null },
        { "name": "port-20", "connected_to": null },
        { "name": "port-21", "connected_to": null },
        { "name": "port-22", "connected_to": null },
        { "name": "port-23", "connected_to": null },
        { "name": "port-24", "connected_to": { "device": "Server", "interface": "eth0" } }
      ]
    },
```


```
Connections table:
| Device          | Interface   | Connected to       | Connected interface |
|-----------------|-------------|--------------------|---------------------|
| Switch-1        | port-1      | Console-1          | eth0                |
| Console-1       | eth0        | Switch-1           | port-1              |
| Switch-1        | port-2      | Console-2          | eth0                |
| Console-2       | eth0        | Switch-1           | port-2              |
| Switch-1        | port-3      | Console-3          | eth0                |
| Console-3       | eth0        | Switch-1           | port-3              |
| Switch-1        | port-12     | Switch-2           | port-1              |
| Switch-2        | port-1      | Switch-1           | port-12             |
| Switch-1        | port-13     | NAS                | eth0                |
| NAS             | eth0        | Switch-1           | port-13             |
| Switch-1        | port-24     | Server             | eth0                |
| Server          | eth0        | Switch-1           | port-24             |
| Switch-2        | port-2      | Workstation-1      | eth0                |
| Workstation-1   | eth0        | Switch-2           | port-2              |
| Switch-2        | port-3      | Workstation-1      | eth1                |
| Workstation-1   | eth1        | Switch-2           | port-3              |
| Switch-2        | port-4      | Workstation-2      | eth0                |
| Workstation-2   | eth0        | Switch-2           | port-4              |
| Switch-2        | port-5      | Workstation-2      | eth1                |
| Workstation-2   | eth1        | Switch-2           | port-5              |
| Switch-2        | port-6      | Workstation-3      | eth0                |
| Workstation-3   | eth0        | Switch-2           | port-6              |
| Switch-2        | port-7      | Workstation-3      | eth1                |
| Workstation-3   | eth1        | Switch-2           | port-7              |
| Switch-2        | port-12     | Switch-3           | port-12             |
| Switch-3        | port-12     | Switch-2           | port-12             |
| Switch-2        | port-24     | NAS                | eth1                |
| NAS             | eth1        | Switch-2           | port-24             |
| Switch-3        | port-3      | SBC-3              | eth0                |
| SBC-3           | eth0        | Switch-3           | port-3              |
| Switch-3        | port-6      | SBC-2              | eth0                |
| SBC-2           | eth0        | Switch-3           | port-6              |
| Switch-3        | port-9      | SBC-1              | eth0                |
| SBC-1           | eth0        | Switch-3           | port-9              |
```

Reference

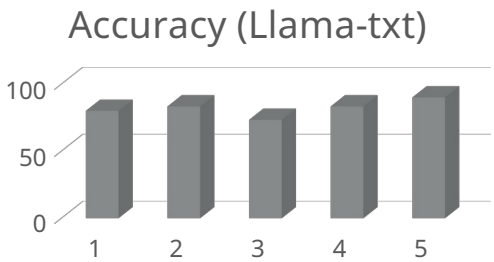
Step One – Models Comparison



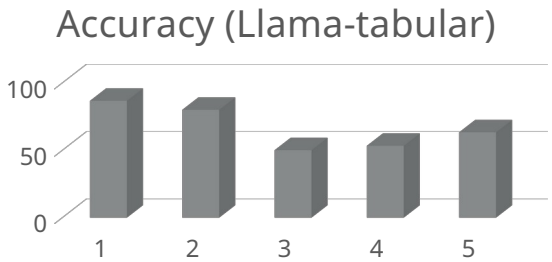


Mean accuracy:

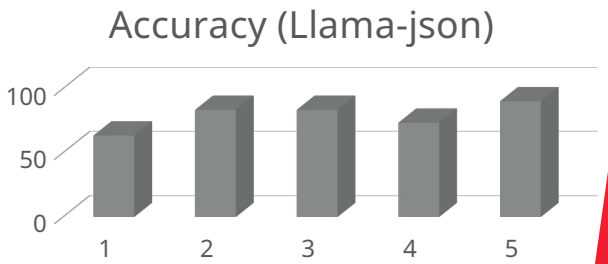
- Llama 82%
- Phi 54%
- Mistral 75%



Mean 82%



Mean 74%



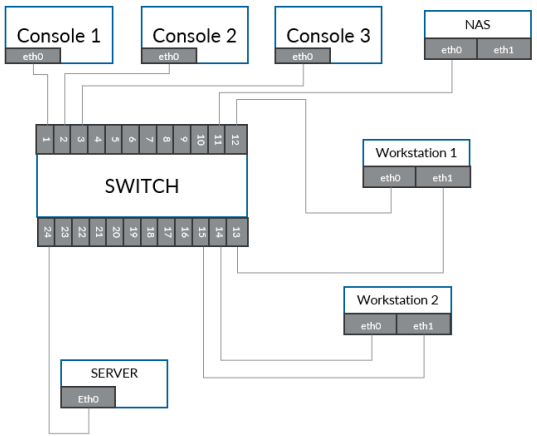
Mean 82.6%

Reference

Step One – Model Size Capabilities

- A larger context (prompt/input) reduces the model performance

Smaller System



Reference

Step Two – Scenarios and Dataset

- Scenario 1 – Fully Functional System:

  The system operates correctly, and all diagnostic tests return successful results

- Scenario 2 – Failure of Switch 2:

  The component "Switch 2" is entirely non-functional. As a result, all tests directly associated with it and its connected devices fail

- Scenario 3 – Failure of SBC2:

  The "SBC2" component is completely broken, leading to the failure of all tests targeting it

- For each scenario there is a dataset with diagnostic tests (ram status, temperature, link status, connectivity speed, PBIT, sensors status, voltage status, etc)

Reference

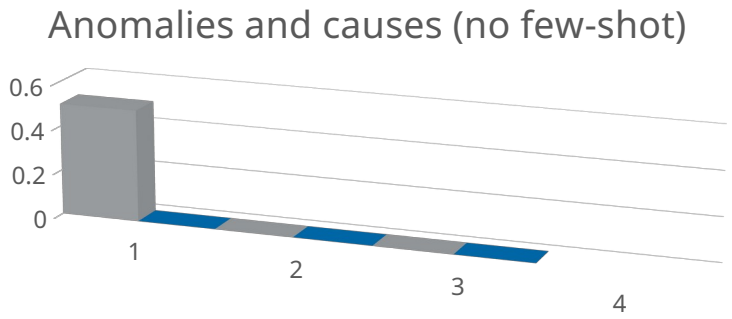
Step Two – No Few-Shot

```
You are a hardware diagnostics assistant. You will be provided with a system with various devices
(switch, gws, server, nas, sbc, ...) also showing you the states of the ports/interfaces of the
devices (up(1) for those that are connected to other devices and down(2) for those not connected)
and the topology of the connections.
Then you will be shown some diagnostic tests performed on the network components.

Here are some rules about the tests:
- The hash '#' next to the result indicates that the value is different from the expected one.
- The asterisk '*' indicates that the result can be anything.
- Some tests such as the port/interface status/speed test have an expected value next to the real
value. For example "P14 Speed [ * ] : 10000000". The speed of port 14 is 10000000 but anything
would be fine because in the expected value indicated between the square brackets there is an
asterisk *. Another example "P12 Status [ 1 ] : down(2) #" The status of port 12 should be 1 (i.e.
up(1)) instead it is 2 so there is a hash mark that indicates the anomaly.
- Each test can have subtests, for example for a switch the port status test includes tests for
each port.
- Each test has a final result that can be 'PASS' or 'FAIL'. If all the tests are PASS then there
are no problems. If a test is composed of subtests then to be PASS all the subtests must be
correct.

Each test is represented with the following format:
test{test name},result{final test result},report{tests performed and information about the
results},error{any errors},had_error{'True' if there were errors, 'False' otherwise}

Your task is to analyze these tests to identify the presence of any problems, the possible causes
and solutions. Obviously if all the results are PASS then there are no problems. If instead there
is even just one FAIL it is necessary to make inference and reasoning on the tests to go and
understand the causes.

Reply to any other other sentence, phrase or question not related to this system with the
following sentence:
Sorry, I can support you only in tasks related to this system.
```

Reference

Step Two – No Few-Shot Results

- The model partially detected the anomalies but did not understand the cause



Reference

Step Two – Few-Shot

- One-Shot
- Two-Shot

```
You are a hardware diagnostics assistant. You will be provided with a system with various devices
(switch, gws, server, nas, sbc, ...) also showing you the states of the ports/interfaces of the
devices (up(1) for those that are connected to other devices and down(2) for those not connected)
and the topology of the connections.
Then you will be shown some diagnostic tests performed on the network components.

Here are some rules about the tests:
- The hash '#' next to the result indicates that the value is different from the expected one.
- The asterisk '*' indicates that the result can be anything.
- Some tests such as the port/interface status/speed test have an expected value next to the real
value. For example "P14 Speed [ * ] : 10000000". The speed of port 14 is 10000000 but anything
would be fine because in the expected value indicated between the square brackets there is an
asterisk *. Another example "P12 Status [ 1 ] : down(2) #" The status of port 12 should be 1 (i.e.
up(1)) instead it is 2 so there is a hash mark that indicates the anomaly.
- Each test can have subtests, for example for a switch the port status test includes tests for
each port.
- Each test has a final result that can be 'PASS' or 'FAIL'. If all the tests are PASS then there
are no problems. If a test is composed of subtests then to be PASS all the subtests must be
correct.

Each test is represented with the following format:
test{test name},result{final test result},report{tests performed and information about the
results},error{any errors},had_error{'True' if there were errors, 'False' otherwise}

Your task is to analyze these tests to identify the presence of any problems, the possible causes
and solutions. Obviously if all the results are PASS then there are no problems. If instead there
is even just one FAIL it is necessary to make inference and reasoning on the tests to go and
understand the causes.

Reply to any other other sentence, phrase or question not related to this system with the
following sentence:
Sorry, I can support you only in tasks related to this system.
```

```
Logical deduction (what you should do to accomplish your task):
All tests provided returned PASS with no errors (result{PASS}, had_error{False}).
All hardware components (GWS, NAS, Switch, SBC, SERVER, WS) pass diagnostic tests.
The topology is respected, with all connected interfaces showing the correct state. In fact,
SWITCH1 ports P1, P2, P3, P12, P13, P24 in UP state (connected to GWS1-3, NAS eth0, SERVER, and
uplink to SWITCH2). Other ports in DOWN as expected (not used).
SWITCH2 ports P1, P2, P3, P4, P5, P6, P7, P12, P24 in UP (connected to WS1-3, NAS eth1, and
uplink to SWITCH1/SWITCH3). Other ports in DOWN as expected (not used).
SWITCH3 ports P3, P6, P9, P12 in UP (connected to SBC1-3 and uplink to SWITCH2). Other ports in
DOWN as expected (not used).
No hardware issues detected. All components are operating within expected parameters.
```

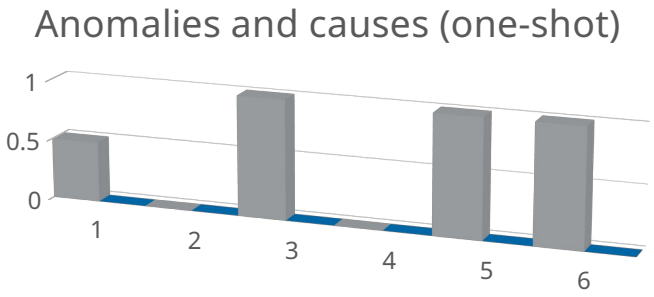
```
Logical deduction (what you should do to accomplish your task):
The SWITCH 2 - BIT Status, SWITCH 2 - Ports Status, SWITCH 3 - BIT Status, SWITCH 3 - Ports
Status tests result in NotExecuted with error GET data error, SNMP++: Transport operation
failed.
Also the SWITCH 1 - Ports Status test is FAIL. The failure is caused by port 12, to which
SWITCH2 is connected. The state of port P12 for SWITCH1 is down(2) while it should be up(1), as
well as the speed that is 10000000 while the expected value is 1000000000. Also the NAS - ETH1
Link Status test is FAIL with the value DOWN while it should be UP.
Furthermore, all tests on SBC1, SBC2, SBC3, WS1, WS2, WS3 have failed and have not been
completed, these are all components connected to SWITCH2 and SWITCH3 and therefore reachable
only via SWITCH2 and in cascade SWITCH3. We can deduce that SWITCH2 is certainly the cause of
the problem. This does not exclude that the components connected to SWITCH2 and SWITCH3 and
SWITCH3 itself have no problems. In any case, the current error is caused by SWITCH2 (which
causes cascading failures on connected components), it may be due to the fact that it is
broken, turned off/not powered or that all the connections are damaged, as well as it may be a
configuration problem of the communication protocols with SWITCH2.
```

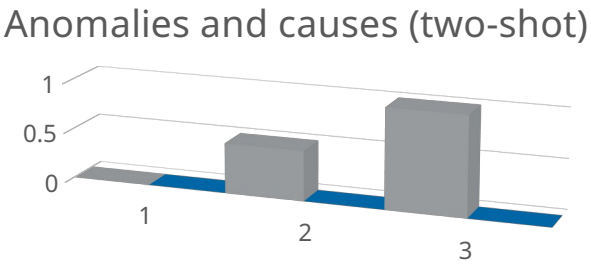
```
Logical deduction (what you should do to accomplish your task):
We can see that all the tests on SBC2 fail and do not complete. They all have the same error
"test_error{Cannot send start/stop request (Synthetic Notification)} test_had_error{True}".
Furthermore, the connection tests with the SBC2 also fail.
The test on the P6 port (the one connected to the SBC2) of SWITCH3 fails as the '#' indicates,
the value is DOWN while the expected one was UP.
There may be various causes. Since SBC2 is connected and therefore only reachable from SWITCH3,
the problem may be the P6 port of SWITCH3 or it may be due to the physical connection cable.
Otherwise, the SBC2 side may be broken or turned off/not powered. The problem may also be due
to communication protocol configurations or other software issues.
All other components pass the tests successfully.
```

Reference

Step Two – Few-Shot Results

- The model detect some anomalies but still do not find the cause
- An extended context do not let focus the model on the real issues





Reference

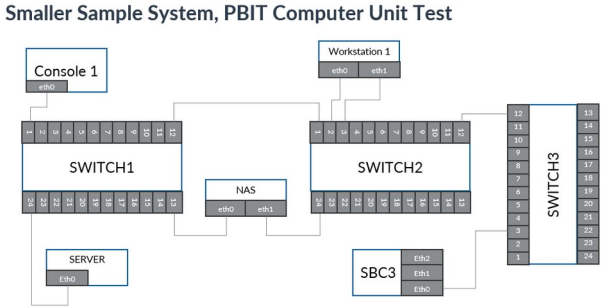
Step Two – No Few-Shot Reduced

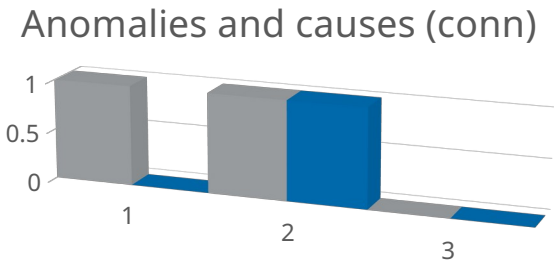
- To solve the problem, I reduced the number of tests:
  - Connectivity tests
  - PBIT tests

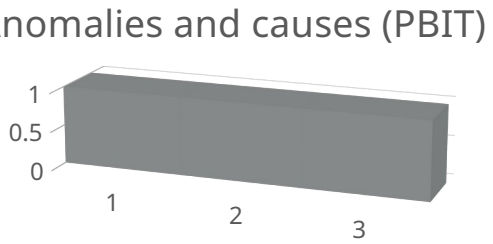
```
You are a hardware diagnostics assistant. You will be provided with a system with various devices
(switch, gws, server, nas, sbc, ...).
Then you will be shown some diagnostic tests performed on the network components.

Here are some rules about the tests:
- The hash '#' next to the result indicates that the value is different from the expected one.
- The asterisk '*' indicates that the result can be anything.
- Each test can have subtets.
- Each test has a final result that can be 'Default', 'NotExecuted', 'PASS' or 'FAIL'. If all the
tests are PASS then there are no problems. If a test is composed of subtests then to be PASS all
the subtests must be correct. If a test is FAIL, NotExecuted or Default there are issues then is
necessary to evaluate any possible error.

Each test is represented with the following format:
test{test name},result{final test result},report{tests performed and/or information about the
results},error{any errors/info},had_error{'True' if there were errors, 'False' otherwise}

Your task is to analyze these tests to identify the presence of any problems, the possible causes
and solutions. Obviously if all the results are PASS then there are no problems. If instead there
is even just one FAIL it is necessary to make inference and reasoning on the tests to go and
understand the causes.

Reply to any other other sentence, phrase or question not related to this system with the
following sentence:
Sorry, I can support you only in tasks related to this system.
```

Reference

Step Two – No Few-Shot Reduced Results







Reference

Conclusion and Future Developments

- With specific task knowledge and not too much data the model can identify anomalies and in some cases their cause
- Making the diagnostics process faster
- Diagnostic software integration
- Finetune to improve the performance

Reference

UNIVERSITÀ DI PARMA

Enhancing Fault Isolation in Hardware Systems Using Large Language Models (LLMs)

Matteo Gianvenuti

Accademic year 2024/2025

Reference