

Intel Unnati Industrial **Training Program**

(PROJECT REPORT)

Submitted by

TEAM : DYNAMIC

Team Members:

1. AP ANUPAMA

PROBLEM STATEMENT

The Problem statement is Power Management Telemetry. It deals with finding the solution on how power consumption can be managed efficiently. This Problem statement is very relevant while considering the current world with the various technological developments that makes the lives of people more comfortable.

In the modern era as the technology advances the power consumption also rises. This leads to many issues in power management as even imagining a single day without electricity makes the lives of people seem impossible.

Power Management Telemetry deals with the system's power supply and consumption.

In this Project the following things are analyzed:

- Get the telemetry data of CPU, NIC, TDP and memory of the system.
- To run a traffic to consume 100% utilization of the system using containers.

https://drive.google.com/drive/folders/1R_YZcqHE2mzAGCWI5Cd_ITOQAQOdpnoQc?usp=sharing

SOLUTION

Telemetry data collection has been done in the Linux Operating System. For this an open source tool named PowerAPI has been installed. The complete telemetry data collection has been done using Docker.

PowerAPI is designed to analyze, measure and optimize the energy usage by hardware and software in a system in an efficient manner. After the installation of Docker, MongoDB has also been installed. The containers that were installed include MongoDB, Grafana, InfluxDB.

Programming language Python has been used. Using Docker and containers the task of collecting the Telemetry data has been done. The telemetry data of TDP, NIC, CPU and memory is collected.

Also to run a traffic to consume 100% utilization of the system using containers is done using stress utility. For this, stress utility has been installed in linux.

<https://github.com/anupama178/intel-project.git>

Features Offered:

Researched and identified the open source tools for Power Management.

Eg: PowerAPI

By obtaining this Telemetry data of CPU,TDP,NIC and Memory of the system we can analyze :

1. how Power can be managed efficiently in a system.
2. Thus analyzing how electricity bill can be reduced.

- Docker containers have been very useful in maintaining optimal and efficient linux environment.
- Stress utility has been used for running traffic for consuming 100% utilization of the system using the container MongoDB.
- Containers like MongoDB,InfluxDB and Grafana are used to store the telemetry data.
- The telemetry data of system working under Stress has also been obtained.
- How efficiently the system manages the Power is analyzed.

Process Flow:

1. Docker Installation:

Docker has been installed in linux for obtaining the the telemetry data.

2. PowerAPI has been cloned using the GitHub repository.
3. docker PowerAPI image has been built and run is done.
4. Containers like MongoDB, InfluxDB and Grafana has been installed.
5. Container names and IDs were then verified.
6. Python 3 Version has been installed as the programming language used here is Python.
7. Using command nano telemetry_script.py the python program for getting telemetry data has been verified.
8. Then the telemetry data of CPU usage percent,Memory use percent,network traffic bytes per second has been obtained.
9. Then pip install psutil was done to get telemetry data of the TDP and NIC.
10. Then the stress-ng was installed to obtained to run the traffic to consume 100% utilization.

```
memory_usage_percent:: command not found
network_traffic_bytes_per_sec:: command not found
ledc@H10:~/Downloads/quick-start-powerAPI/powerapi_custom$ nano telemetry_script.py
ledc@H10:~/Downloads/quick-start-powerAPI/powerapi_custom$ python3 telemetry_script.py
{'timestamp': datetime.datetime(2024, 7, 12, 5, 42, 40, 160510), 'cpu_usage_percent': 30.5, 'memory_usage_percent': 50.2, 'network_traffic_bytes_per_sec': 1024.5}
ledc@H10:~/Downloads/quick-start-powerAPI/powerapi_custom$ w
```

Telemetry data of CPU,Memory and Network Traffic

```
ledc@W10: ~/Downloads/quick-start-powerAPI/powerapi_custom
NIC Telemetry: {'network_stats': {'incoming_bytes': 254965790, 'outgoing_bytes': 16052602}}
TDP Telemetry: {'cpu_percent': 1.3}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254965790, 'outgoing_bytes': 16052602}}
TDP Telemetry: {'cpu_percent': 1.0}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254965856, 'outgoing_bytes': 16052737}}
TDP Telemetry: {'cpu_percent': 0.3}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254966174, 'outgoing_bytes': 16052953}}
TDP Telemetry: {'cpu_percent': 1.5}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254975645, 'outgoing_bytes': 16060835}}
TDP Telemetry: {'cpu_percent': 1.0}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254975816, 'outgoing_bytes': 16061006}}
TDP Telemetry: {'cpu_percent': 1.5}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254976182, 'outgoing_bytes': 16061141}}
TDP Telemetry: {'cpu_percent': 1.0}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254976704, 'outgoing_bytes': 16061370}}
TDP Telemetry: {'cpu_percent': 1.3}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254976824, 'outgoing_bytes': 16061370}}
TDP Telemetry: {'cpu_percent': 1.0}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254977490, 'outgoing_bytes': 16063891}}
TDP Telemetry: {'cpu_percent': 1.0}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254977655, 'outgoing_bytes': 16064062}}
TDP Telemetry: {'cpu_percent': 1.0}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254982534, 'outgoing_bytes': 16073008}}
wTDP Telemetry: {'cpu_percent': 1.3}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254982714, 'outgoing_bytes': 16073008}}
wTDP Telemetry: {'cpu_percent': 2.0}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254982780, 'outgoing_bytes': 16073143}}
w
TDP Telemetry: {'cpu_percent': 1.0}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254983068, 'outgoing_bytes': 16073564}}
TDP Telemetry: {'cpu_percent': 1.0}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254983351, 'outgoing_bytes': 16074040}}
TDP Telemetry: {'cpu_percent': 2.0}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254984063, 'outgoing_bytes': 16074490}}
TDP Telemetry: {'cpu_percent': 0.0}
NIC Telemetry: {'network_stats': {'incoming_bytes': 254984129, 'outgoing_bytes': 16074625}}
TDP Telemetry: {'cpu_percent': 1.3}
```

Telemetry data of TDP and NIC

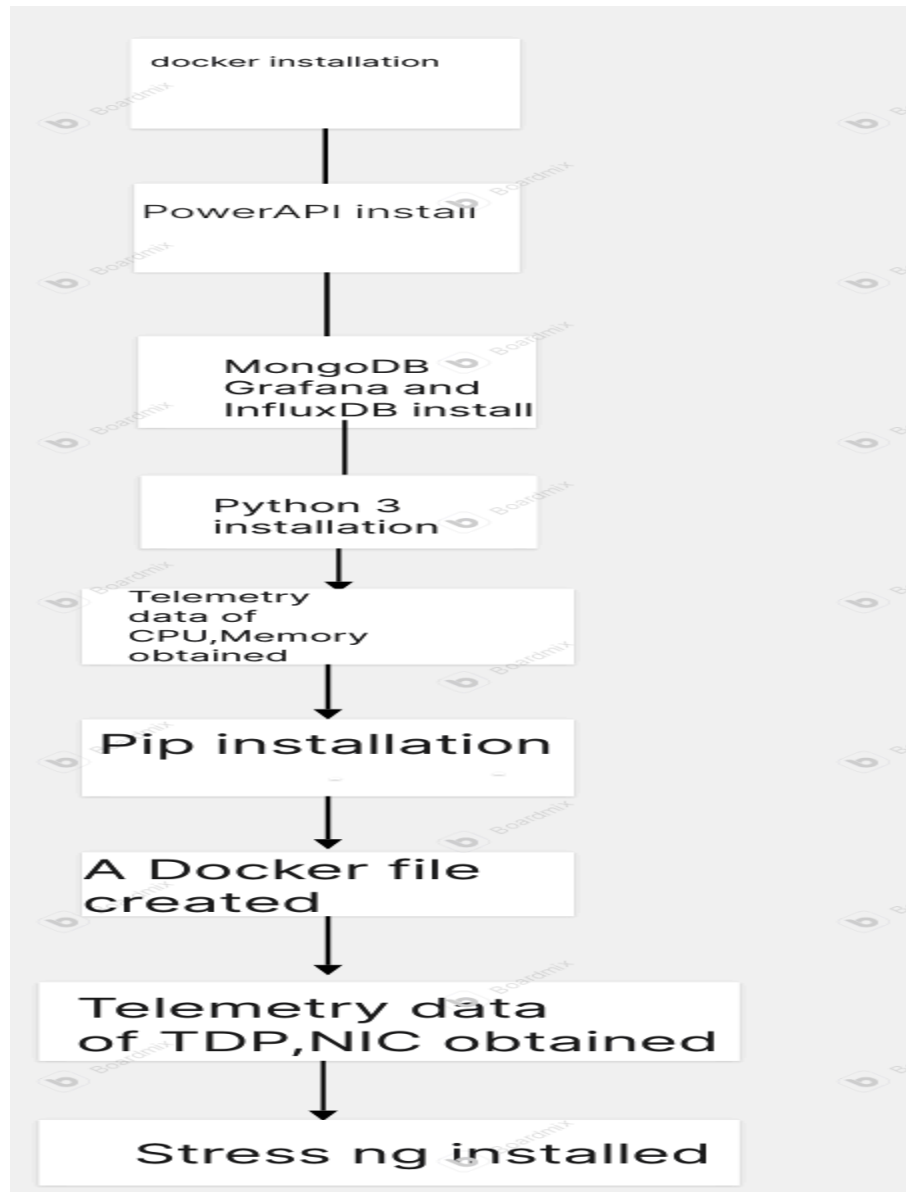
```

ledc@W10: ~/Downloads/quick-start-powerAPI/powerapi_custom
NIC Telenetry: {'network_stats': {'incoming_bytes': 254977655, 'outgoing_bytes': 16064062}}
TDP Telenetry: {'cpu_percent': 1.0}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254982534, 'outgoing_bytes': 16073008}}
wTDP Telenetry: {'cpu_percent': 1.3}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254982714, 'outgoing_bytes': 16073008}}

wTDP Telenetry: {'cpu_percent': 2.0}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254982780, 'outgoing_bytes': 16073143}}
w
TDP Telenetry: {'cpu_percent': 1.0}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254983068, 'outgoing_bytes': 16073564}}
TDP Telenetry: {'cpu_percent': 1.0}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254983351, 'outgoing_bytes': 16074048}}
TDP Telenetry: {'cpu_percent': 2.0}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254984063, 'outgoing_bytes': 16074498}}
TDP Telenetry: {'cpu_percent': 0.0}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254984129, 'outgoing_bytes': 16074625}}
TDP Telenetry: {'cpu_percent': 1.3}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254984831, 'outgoing_bytes': 16074854}}
TDP Telenetry: {'cpu_percent': 1.0}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254986279, 'outgoing_bytes': 16076138}}
TDP Telenetry: {'cpu_percent': 1.0}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254986345, 'outgoing_bytes': 16076265}}
TDP Telenetry: {'cpu_percent': 1.5}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254987641, 'outgoing_bytes': 16078109}}
TDP Telenetry: {'cpu_percent': 1.0}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254987863, 'outgoing_bytes': 16078338}}
TDP Telenetry: {'cpu_percent': 0.3}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254987863, 'outgoing_bytes': 16078338}}
TDP Telenetry: {'cpu_percent': 1.0}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254987929, 'outgoing_bytes': 16078473}}
TDP Telenetry: {'cpu_percent': 0.0}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254988084, 'outgoing_bytes': 16079092}}
TDP Telenetry: {'cpu_percent': 2.3}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254989462, 'outgoing_bytes': 16079554}}
TDP Telenetry: {'cpu_percent': 1.0}
NIC Telenetry: {'network_stats': {'incoming_bytes': 254990239, 'outgoing_bytes': 16079860}}
TDP Telenetry: {'cpu_percent': 1.2}

```


Architecture Diagram



Technologies Used

- Docker: Docker platform being designed to build, deploy, and manage applications using containers.
- Python: The Programming that was used to generate code to get Telemetry data.
- MongoDB: Used as a container to store the Telemetry data of CPU,NIC,TDP, and memory.
- PowerAPI : The Open Source used for getting the Power Telemetry Data.
- Stress: To create different types of loads, this was used so that CPU power management efficiency can be found easily.

Team Members and Contribution

1. AP ANUPAMA - Coding,Collecting the Telemetry data and Report Writing.

Conclusion:

Finally, in conclusion ,the Telemetry data of power consumption by CPU,TDP,NIC and memory has been obtained.Thus by telemetry data analysis solution for power management can be obtained.Open Sources like PowerAPI, Powertop have been analyzed for getting the Telemetry Data of different components of the system.This open sources helps to find out the power consumption.As in the modern world as the deployment of devices increases to make tasks more easy the more would be demand for energy,that is the power consumption rate increases day by day.So inorder to obtain less power consumption even while using many devices, analyzing of Power Telemetry data is very important which can reduce the current bill.So that can lead to the achievement of the goal of zero power consumption.