A red and black rectangle

Description automatically generated

PyAssistant

Undergraduate Project 2024/2025

Student Name: Hassan Naveed Khan

Student ID: 2115119

Programme: BEng Software Engineering

Module and Title: SEC6201 Undergraduate Project

Marking Tutor:

# Declaration Page

# Abstract

# Table of Contents

# List of Figures

# List of Abbreviations

# Glossary

# Introduction

## Background

In 2025, process management has become very critical for the operating systems. No matter what operating system a machine is learning a process manager is always there which ensures that the applications and processes run smoothy. Process management can allow to track, monitor active processes. They are also used to optimize resource usage such as RAM usage, Storage distribution etc. One another important advantage of having process manager is that it help us troubleshoot performance issues. These are the main reasons process managers are very important to have.

As the technology keeps developing and advancing day by day, users and systems require more tools to manage their processes. With the new tools and processes coming lately some systems can become overloaded which can cause systems running very slow.

## Problem Statement

Most of the operating systems have their own process manager built in, such as Windows Task Manager for Windows and top for Linux. However, having even the most advanced process manager doesn’t mean that they are user-friendly. There is need of a platform that can provide functionalities to a specific user need.

## Objectives

Ok so let’s say a company have a customer service department where the team is working on non-stop problems for their customers, having no time to sort out problems for their computers. The process manager will allow us to have multiple users on different nodes such as “Customer Services” where the administrator will be able to manage their computers for the staff working in Customer Services. This is going to make sure that the administrator doesn’t have to go to the machines and interrupt the team working but instead he can just check it on his own computer.

The process manager will have different nodes, and, in each node, we will be able to see multiple hosts.

The admin will be able to :

**Memory Usage Monitoring and Management**

* This will display the real-time usage of RAM.
* This will display usage of ram taken by each process independently.
* Alert if the usage is reaching high.
* Visualisation of the RAM.

(The next step is to fill the below ones, its 25.01.25 04.11 am right now)

**CPU Usage Monitoring**

**Storage Usage Monitoring**

**Network Monitoring**

**Application Performance Monitoring**

**Process-Specific Monitoring**

# Literature Review

The growing complexity of modern computing systems and the technologies requires monitoring to ensure efficient performance, reliability, and security. System monitoring provides critical insights into resource utilization, logs, and network performance helping administrators to detect errors and to also prevent failures. This literature review examines existing research and technologies that are related to **system monitoring, alerting mechanisms** using python scripts. There will be also scripts that will be used to visualize the collected data and tools to visualize them in graphs and tables.

# System Monitoring Using Scripts

System monitoring is the process of continuously tracking the performance and health of various computing resources, such as CPU, memory, disk usage, and network activity. It is very essential in IT environments for tracking resources usage, application performance, network traffic and security related metrics. This literature focus on system reliability, optimization, and security and how these can be achieved by automating monitoring and alerting processes.

The role of python in modern system monitoring solutions is often highlighted, particularly in scripts that are used for custom monitoring in environments that do not require full-scale monitoring solutions or where specific, customizable data analysis is required.

# Python Libraries for System Monitoring

## psutil

It is a cross-platform library for retrieving information about system utilization, processes, and system resources such as CPU, memory, disks and network. It’s powerful tool that provides easy access to system-level information with just a few lines of python code. (Python, 2024a)

## Scapy

Scapy is a powerful interactive packet manipulation library written in python. Scapy is able to forge or decode packages of a wide number of protocols, send them on the wire, and capture them, match requests and replies, and much more (Scapy, 2024).

## Socket

Socket and socket API are used to send messages across a network. They provide a form of inter-process communication (IPC). The network can be a logical, local network to the computer, or one that’s physically connected to an external network, with its own connections to other networks. The obvious example is the internet, which you connect to via ISP(Nathan Jennings, 2024).

## **Subprocess**

The subprocess library allows you to run system commands and capture their output directly in python. The subprocess module allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes(Python, 2024b).  
  
Prometheus

Prometheus, a Cloud Native Computing Foundation Project, is a systems and service monitoring system. Prometheus is a powerful monitoring and alerting tool that stores data in time-series. API clients are used in in programming languages such as python, java etc to interact with Prometheus(pypi).

## Speedtest-cli

It is a command line interface for testing network interface with the help of the API Speedtest.net (pipit).

## PySMART

PySMART is a library for querying S.M.A.R.T data from hard drives, which allows us to monitor the health of the devices and the storage status of the devices.

## Boto3

The use of AWS SDK for python(boto3) is to create, configure and manage AWS Services such as Amazon Elastic Compute (Amazon EC2) and simple storage devices(Amazon S3). The SDK provides an object-oriented API as well as low-level access to AWS Services(Amazon, 2024).

## GPUtil

GPUtil is a python module for getting the GPU status from Nvidia GPU’s using nvidia-smi(Anaconda).

These are some python libraries that I will be using, there will be more of them while I do my research.

With these libraries I will gather my data yes but how will I store my data ?

# **Store Collected Data**

Data storage in digital world has become a need today. Every system and application are incomplete. By storing data, we not only save it but also helps us to make decisions in future regarding predictions and system reliability.

## Ways of Storing Data:

Back in days the data used to be stored manually in paper records. This was very challenging, especially when it comes to restore the data, and it was time consuming. The risks of fire or theft could cause loss to the companies. After the introduction of digital storage, it began with the floppy disks aur tape drives which made it easy. But then later, when technology even advanced to a next level, the data was stored in servers that we call today database servers, which are more reliable and quicker.

## Decision Making for data storage:

Now days organisations and business depend on their data too much, even when it comes to decision making. When the company wants to store their data, they look at the data collection needs and the way their data needs to be stored. For example, when we talk about e-commerce website, we know that the website will use trend data in the market for their products, to sell them in numbers grow their business. We will find which methods we need according to our data storage. For now we will just gather suitable options.

## Ways to Store Data today

Now days there are many ways to store the data, many modern and efficient methods are available. These are some examples we can use to store the data.

### SQLite

SQLite is a C-language library that implements a small, fast, self-contained, high reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers(SQLite).

## Relational Databases (RDBMS)

Relational databases such as MySQL and PostreSQL, stores data in tables format. It is ideal for complex systems where multiple users must access the data at the same time. To organize the data SQL use queries.

## NoSQL Databases

NoSQL databases, like MongoDB is used to store unstructured data. These systems give flexibility. We also have the options to store and view data in JSON format while using NoSQL databases.

## Cloud Storage Systems:

Data can also be stored in Amazon S3(Cloud Object Storage) offering industry-leading scalability, data availability, security and performance. It is very cost-effective and reliable.

These are not the systems that I have decided to use yet, I still must decide which system I will pick to use depending on how it goes.

Once we have stored data safely, what if we want to view the data or compare them on our server? If this is an application for a customer or client, we need them to provide us a feedback, right? That means they need to be able to view the data and tell us if they are satisfied or if we need to improve or make any changes. To make it possible.

We can view the data gathered by our scripts which is saved in the databases, and to make it possible we need Web Server. For example we can work with flask framework which is built in python. Flask allows us to make APIs from which we can server the data. With the help of API, we can interact between the database server and web browser. The web browser will send a query for a request to the database and the database will send it back to the browser.

Once we received the data , it will be displayed. But to display the data we might need to gather data visualisation tools. Such as Plotly, Dash, Matplotlib or Seaborn. Let’s talk about them:

## Plotly

It is a popular library that allows us to make interactive and dynamic graphs(Plotly, b).

## Dash

It is an Plotly extension that works with Plotly which intergrates with flask and allows us to create an advanced dashboard(Plotly, a).

## Matplotlib

Matplotlib is a comprehensive library for creating a static, animated, and interactive visualizations in Python(Matploitlib).

Once all this is achieved, we need to make sure the data integrity, also to look in for things like Unauthorized access and data encryption. I have yet need to do further research in these areas as I only have tried python libraries that I mentioned above.

# Problem Statement

# Obejctives

# Research Questions

# Methodology

This chapter explains methods and approaches in easy language that work to **design, analyse** and **improve performance metrics** for a task manager based on Python language. This research is very detailed research which means practical solutions and qualitative ideas will also be used where needed.

## Justification

To analyse numbers and data quantitative method are being used. In this method measuring data and evaluating data is very important. Data such as **CPU Usage, memory consumption** and collecting **network usage** will be the main focus. There will be other aspects that will be also focused such as **types of security** which will be discussed later on. The practical solutions will be studied depending on the behaviour of the system.

## Data Sources and Collection

### Primary Data

The Primary Data will be collected through my experiments and testing. This means that the data will be collected by my python process manager. I will be collecting data metrics such as :

* **Performance Testing:** I will test the task manager in different situations so that I can understand the system efficiency and resources.
* **Real Time Monitoring:** Tools like ***psutil***will be used to collect the data in real time.
* **Custom Monitoring Tools:** I will be using libraries like Prometheus so that I will be able to test Advanced Monitoring and this way I will receive data that will be very useful to me.

### Secondary Data

The following is included in the Secondary Data:

* **Literature Review:** With my ***Literature Review ,Academic articles*** and ***the official Python Library Documentation*** will help me to get the insights for process management and then data visualisation.
* **Benchmark Data:** Whatever data that I have collected I will compare that with popular tools such as Prometheus and then I will analyse which one is better or suitable.
* **Case Studies:** I will test different Operating Systems and see how their task manager works, to get better ideas for a task manager. It will also help me to develop new ideas.
* **Library Documentation**: I will be using libraries such as psutil, matplotlib , pandas and others. Their official documentation will help me to understand the libraries even more so that I can integrate them in my task manager.

## Research Philosophy

A research philosophy is a belief about the way in which data about a phenomenon should be gathered, analysed and used (Galliers, 1991).

### Positivism

Positivism is a term used to describe an approach to the Study of the society that relies specifically on empirical evidence, such as controlled experiments and used statistics (Nickerson, 2024)

This project will focus on factual data and measurable data. I will use experimental and factual analysis to make improvements.

### Pragmatism

Pragmatism is an approach that suggests that there are in fact many different ways of interpreting the world and conducting research to investigate reality and that combination of different approaches may provide a boarder understanding of the phenomena being used (helm, 2024)

In this project practical and real-world solutions are also important. At some places or situations, I will have to use the qualitative analysis such as log analysis that comes under pragmatism approach.

## Tools and Technologies

To collect data and analyse it I will be using the following tools and libraries:

### Data Collection Tools

* **psutil:** To collect systems CPU, memory and disk usage.
* **Prometheus:** Is used for Advanced Monitoring and helps with comparison.

### Data Analysis and Visualisation Tools

* **pandas**: To organize data and manipulate the data.
* **matplotlib** and **seaborn**: To create graphs and charts.
* **plotly**: To create interactive dashboards and visualisations.
* **scarpy**: To analyse the network data.
* **R Programming:** To create visualisations using the libraries such as ggplot2.

## Data Analysis Approach

The data that will be collected will be cleaned, organized and be analysed using Python Libraries.

* **CPU** and **memory usage** will be displayed using graphs.
* To identify **issues and trends** the results will be compared.
* The **Interactive dashboards** will provide a better view of system performance.

## Challenges and Solutions

**Data Accuracy:** To collect accurate and correct data tools like psutil will be tested initially because some tools are limited for example psutil only works locally that means I will have to find alternative way if I am including cloud systems.

**Tool Integration:** Combining multiple tools can be very challenging, even though a task manager cannot be built without combining the data. This means data will be tested step by step, solved test by step and then integrate it step by step.

**Scalability**: As the task manager keeps developing by time, it will be handling more processes at a time, meaning more resources will be required, more data will be monitored and stored. This can be challenging as methods for multi-task need to be looked at.

**Error Handling:** While collecting Real-Time data we can get unexpected errors. Error-Handling mechanisms will need to be looked at so that the system or the task manager can run smoothly.

**Performance Comparison:** While comparing the performance of the task manager with tools like Prometheus some metrics such as CPU Usage, memory consumption and network traffic activity will be evaluated. This means there will be running similar tasks on both tools and to analyse the results so that strengths, weaknesses and areas of improvement can be identified for the task manager.

# System Design

# Implementation

# Results

# Analysis and Discussion

# Testing

# Challenges and Limitations

# Conclusion

# Recommendations

# References

Galliers, R. D. (1991). *cityu*. Retrieved December 08, 2024, from http://www.is.cityu.edu.hk/staff/isrobert/phd/ch3.pdf

helm. (2024). *What is Pragmatism?* Retrieved December 08, 2024, from https://www.nottingham.ac.uk/helmopen/rlos/research-evidence-based-practice/designing-research/types-of-study/understanding-pragmatic-research/section03.html

Nickerson, C. (2024). *simply psychology*. Retrieved December 08, 2024, from https://www.simplypsychology.org/positivism-in-sociology-definition-theory-examples.html

Scapy. (2024, 11 14). *Scapy*. Retrieved from Scapy: https://scapy.net/

Amazon (2024) *Boto3.* [Online] Available from: <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html.> [Accessed].

Anaconda ( ) *GPUtil.* [Online] Available from: <https://anaconda.org/conda-forge/gputil.> [Accessed].

Matploitlib ( ) Matploitlib. *,* [Online] ( ), pp. Available from: <https://matplotlib.org.> [Accessed].

Nathan Jennings (2024) *Socket Python Library.* [Online] Available from: <https://realpython.com/python-sockets/.> [Accessed].

pipit ( ) *Speedtest-cli.* [Online] Available from: <https://pypi.org/project/speedtest-cli/.> [Accessed].

Plotly (a) *Dash.* [Online] Available from: <https://dash.plotly.com/dash-core-components/graph.> [Accessed].

Plotly (b) *Plotly.* [Online] Available from: <https://plotly.com/python/.> [Accessed].

pypi ( ) *prometheus-api-client.* [Online] Available from: <https://pypi.org/project/prometheus-api-client/.> [Accessed].

Python (2024a) *Python pypi psutil.* [Online] Available from: <https://pypi.org/project/psutil/.> [Accessed].

Python (2024b) *Python Subprocess module.* [Online] Available from: <https://docs.python.org/3/library/subprocess.html#module-subprocess.> [Accessed].

SQLite ( ) *SQLite.* [Online] Available from: <https://www.sqlite.org.> [Accessed].

# Timeline/Gantt Chart