



# SIDDAGANGA INSTITUTE OF TECHNOLOGY

(An Autonomous Institute under Visvesaraya Technological University, Belgavi)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE (4th SEMISTER)



OPERATING  
SYSTEM

## VMWARE - PROJECT

### TOOL TO CAPTURE DYNAMIC MEMORY CONSUMPTION OF THE APPLICATION

**Presented by:**

*Akash L* – 1SI21AD007

*Suprith T S* – 1SI21AD050

*Atchutha Rao T A* – 1SI21AD064

# Table of CONTENTS

---

01

INTRODUCTION  
TO VM-WARE  
TOPIC

02

Real-time Data  
Update Module

03

Description and  
algorithm

04

Result  
Snippets

05

Conclusion

# 01

INTRODUCTION  
TO VM-WARE  
TOPIC

## TOOL TO CAPTURE DYNAMIC MEMORY CONSUMPTION:

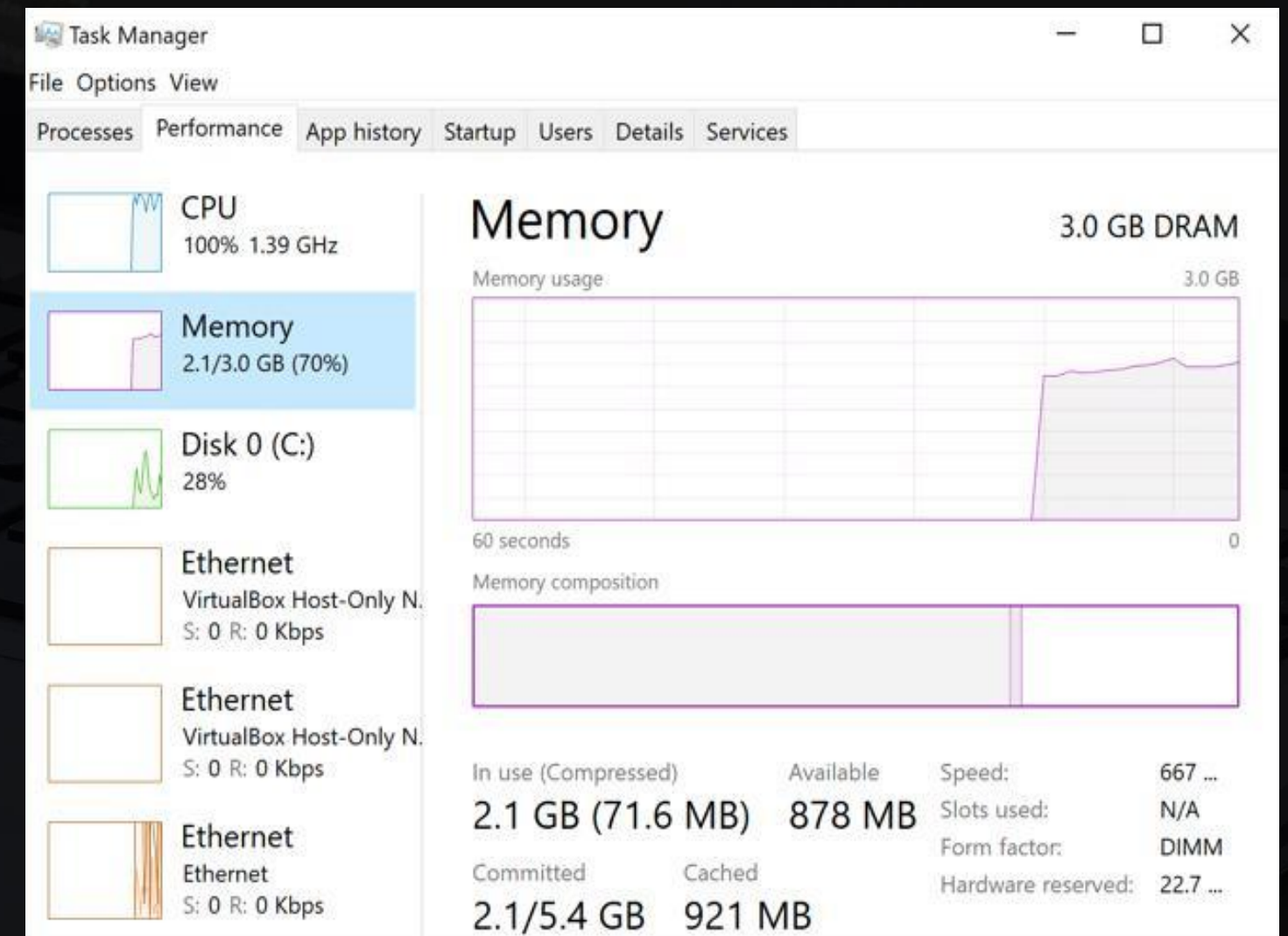
### TOOLS:

#### 1. Performance Monitoring Tools:

Task Manager (Windows): The built-in Task Manager in Windows provides real-time information about running processes, memory usage, and resource consumption.

#### 2. Real Time Monitoring Tool:

Some tools offer real-time monitoring, allowing developers to observe memory consumption as the application runs. This can be crucial for identifying sudden spikes in memory usage or gradual memory leaks.





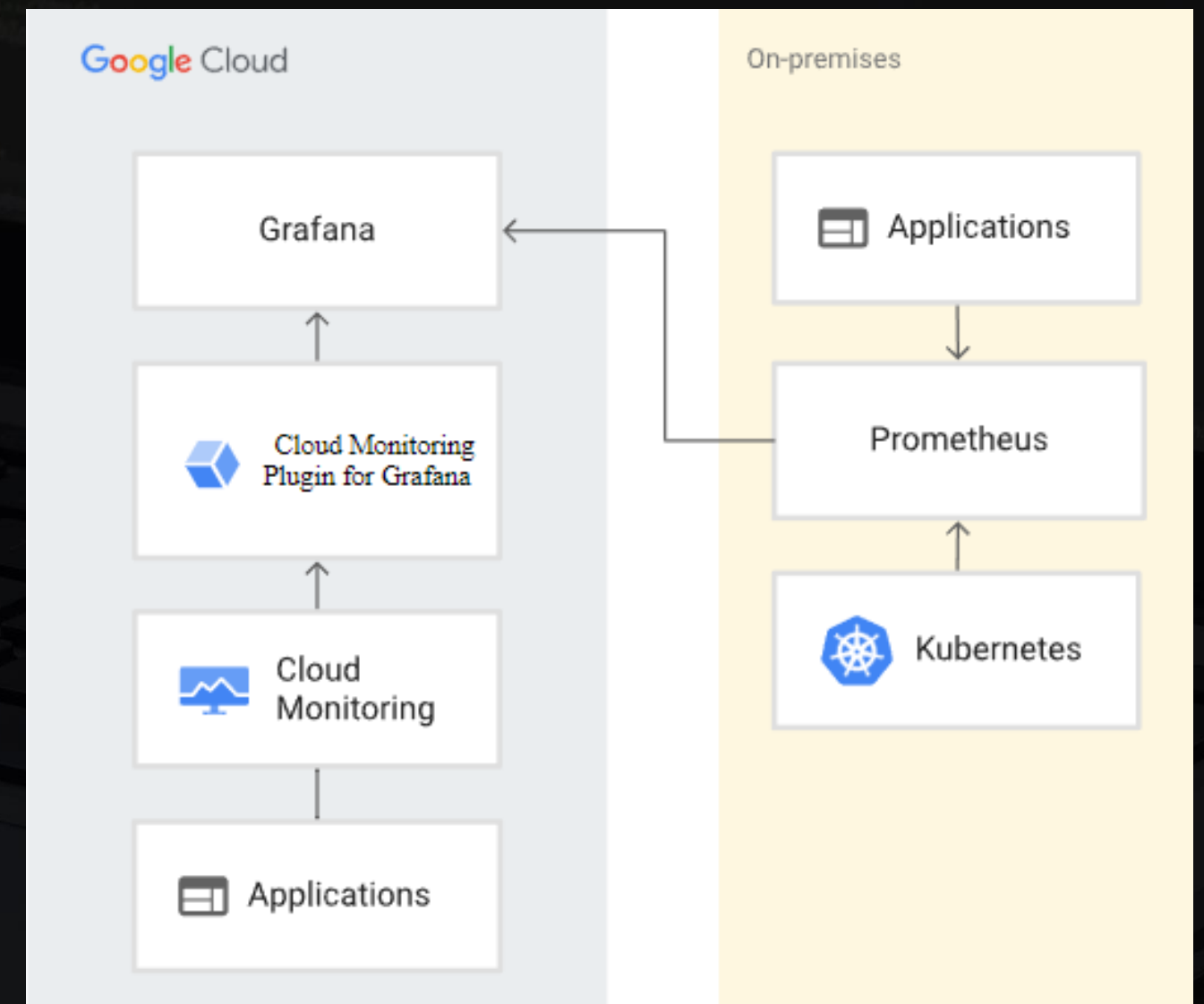
# TOOL TO CAPTURE DYNAMIC MEMORY CONSUMPTION:

## 3. Profiling and Monitoring Tools:

- **Valgrind:** A powerful instrumentation framework for debugging and profiling. Memcheck, a tool within Valgrind, can track memory leaks and invalid memory access.
- **GDB (GNU Debugger):** GDB can help debug memory-related issues using features like memory breakpoints and watchpoints.

## 4. Cloud Platform Monitoring:

- Cloud platforms like Amazon Web Services (AWS) and Microsoft Azure offer monitoring tools for applications hosted on their platforms, including memory usage metrics.



# 02

Real-time Data  
Update Module

## TOOL TO CAPTURE DYNAMIC MEMORY CONSUMPTION:

### REAL-TIME MONITORING:

- **Some tools offer real-time monitoring, allowing developers to observe memory consumption as the application runs. This can be crucial for identifying sudden spikes in memory usage or gradual memory leaks.**

### USES:

- **Performance Optimization:** Real-time monitoring helps administrators identify memory bottlenecks and resource contention. By tracking memory consumption, they can allocate resources more effectively, ensuring that VMs have sufficient memory to operate without causing performance degradation.
- **Predictive Analytics:** Advanced monitoring tools can analyze historical data and provide predictions about future memory consumption trends. This helps in optimizing resource provisioning and avoiding resource shortages.

# TOOL TO CAPTURE DYNAMIC MEMORY CONSUMPTION:

## Functionality:

Real-time data update module focused on monitoring memory usage of a specific application. This module continuously tracks and displays the memory consumption of the chosen application, updating the information at regular intervals.

## Tools Used:

1. psutil library: used to access memory-related information and monitor memory usage of specific applications.
2. time module: It allows you to work with time intervals, sleep for specific durations, and measure time intervals between events
3. os module: It allows you to execute system-level commands, retrieve environment variables, and perform various system-related tasks. In this code



03

Description and  
algorithm

## Description:

- The program provides a simple command-line dashboard to display memory information, monitor memory usage of specific applications, and stop applications based on user input. It uses the psutil library to gather process and memory information from the system.

## Algorithm:

- Display a dashboard with the following options:
  - Display memory information
  - Monitor memory usage of a specific application
  - Stop a specific application
  - Exit
- Get the user's choice from the dashboard.
- If the user chooses to display memory information, then do the following:
  - Get the total memory, available memory, used memory, and memory usage percentage from the operating system.
  - Display the memory information to the user.

4. If the user chooses to monitor memory usage of a specific application, then do the following:

- Get the name of the application from the user.
- Get the list of all running processes from the operating system.
- Find all processes that have the specified application name.
- Get the total memory usage of all the processes that have the specified application name.
- Display the memory usage of the application to the user.

5. If the user chooses to stop a specific application, then do the following:

- Get the name of the application from the user.
- Get the list of all running processes from the operating system.
- Find the process that has the specified application name.
- Stop the process.
- Display a message to the user that the application has been stopped.

6. If the user chooses to exit, then do the following:

- Exit the program.



# 04

Result  
Snippets

## Results

Choosing option 1:

⊗ Memory Management Dashboard

- 1. Display Memory Information
- 2. Monitor Memory Usage of a Specific Application
- 3. Stop a Specific Application
- 4. Terminate Highest Memory Consuming Application
- 5. Exit

Enter your choice: 1

Memory Information:

Total Memory: 8320225280 bytes

Available Memory: 2232532992 bytes

Used Memory: 6087692288 bytes

Memory Usage Percentage: 73.20%

Press Enter to continue...|

# Choosing option :2

## Memory Management Dashboard

1. Display Memory Information
2. Monitor Memory Usage of a Specific Application
3. Stop a Specific Application
4. Terminate Highest Memory Consuming Application
5. Exit

Enter your choice: 2

All running process names: ['System Idle Process', 'System', '', 'Reg  
exe', 'svchost.exe', 'csrss.exe', 'wininit.exe', 'csrss.exe', 'servic  
t.exe', 'svchost.exe', 'WUDFHost.exe', 'svchost.exe', 'svchost.exe',  
exe', 'fontdrvhost.exe', 'dwm.exe', 'svchost.exe', 'svchost.exe', 'Su  
st.exe', 'svchost.exe', 'svchost.exe', 'svchost.exe', 'RuntimeBroker.  
svchost.exe', 'svchost.exe', 'Dell.TechHub.Instrumentation.SubAgent.e  
exe', 'svchost.exe', 'svchost.exe', 'svchost.exe', 'svchost.exe', 'M  
ost.exe', 'svchost.exe', 'svchost.exe', 'msedge.exe', 'DDVDDataCollect  
.exe', 'igfxCUIServiceN.exe', 'svchost.exe', 'svchost.exe', 'svchost.  
e', 'svchost.exe', 'WmiPrvSE.exe', 'ApplicationFrameHost.exe', 'Code.  
svchost.exe', 'svchost.exe', 'conhost.exe', 'wlanext.exe', 'spoolsv.e  
xe', 'armsvc.exe', 'svchost.exe', 'svchost.exe', 'svchost.exe', 'Nahi  
'svchost.exe', 'PEFService.exe', 'nvcontainer.exe', 'RtkAudUService6  
e', 'svchost.exe', 'esif\_uf.exe', 'MsMpEng.exe', 'jhi\_service.exe',  
e', 'msedgewebview2.exe', 'svchost.exe', 'WmiPrvSE.exe', 'WINWORD.EXE  
Host.exe', 'sihost.exe', 'mfefvtps.exe', 'SystemSettingsBroker.exe',  
rotectedModuleHost.exe', 'svchost.exe', 'SearchIndexer.exe', 'svchost  
, 'pwsh.exe', 'svchost.exe', 'igfxEMN.exe', 'svchost.exe', 'svchost.  
.exe', 'nvcontainer.exe', 'msedgewebview2.exe', 'DDVCollectorSvcApi.e  
'svchost.exe', 'conhost.exe', 'WidgetService.exe', 'NahimicSvc64.exe'  
'RtkAudUService64.exe', 'conhost.exe', 'DDVRulesProcessor.exe', 'mse  
'Widgets.exe', 'RuntimeBroker.exe', 'Dell.DCF.UA.Bradbury.API.SubAger  
cationWindows.exe', 'ShellExperienceHost.exe', 'svchost.exe', 'dllhos  
l.TechHub.exe', 'GameLibraryAppService.exe', 'msedgewebview2.exe', 'M  
untimeBroker.exe', 'ctfmon.exe', 'chrome.exe', 'Code.exe', 'msedge.ex  
, 'chrome.exe', 'NVIDIA Share.exe', 'svchost.exe', 'svchost.exe', 's  
are.exe', 'svchost.exe', 'servicehost.exe', 'mcapexe.exe', 'python.ex  
HealthSystray.exe', 'SecurityHealthService.exe', 'vgtray.exe', 'AWCC.  
'svchost.exe', 'svchost.exe', 'conhost.exe', 'svchost.exe', 'python.  
ce.exe', 'Dell.D3.WinSvc.exe', 'svchost.exe', 'RuntimeBroker.exe', 'F  
SubAgent.exe', 'conhost.exe', 'ServiceShell.exe', 'Dell.TechHub.DataM  
, 'Dell.TechHub.Analytics.SubAgent.exe', 'conhost.exe', 'POWERPNT.EXE  
webview2.exe', 'Dell.TechHub.Instrumentation.UserProcess.exe', 'svch  
svchost.exe', 'python.exe', 'MoNotificationUx.exe', 'svchost.exe', 'c

Enter the name of the application to monitor: chrome

Monitoring Memory Usage of 'chrome' (Press Ctrl+C to stop)...

Total System Memory: 8320225280 bytes

'chrome' Memory Usage: 1090752512 bytes

Memory Usage Percentage by 'chrome' : 13.11%



Monitoring Memory Usage of 'Code' (Press Ctrl+C to stop)...

Total System Memory: 8320225280 bytes

'Code' Memory Usage: 1021665280 bytes

Memory Usage Percentage by 'Code' : 12.28%



## Choosing option :3

### Memory Management Dashboard

1. Display Memory Information
2. Monitor Memory Usage of a Specific Application
3. Stop a Specific Application
4. Terminate Highest Memory Consuming Application
5. Exit

Enter your choice: 3

Enter the name of the application to stop: chrome

### Memory Management Dashboard

1. Display Memory Information
2. Monitor Memory Usage of a Specific Application
3. Stop a Specific Application
4. Terminate Highest Memory Consuming Application
5. Exit

Enter your choice: 3

Enter the name of the application to stop: chrome

svmem(total=8320225280, available=1094938624, percent=86.8, used=7225286656, free=1094938624)

Process 'chrome' (PID: 3492) has been terminated.

Process 'chrome' (PID: 5836) has been terminated.

Process 'chrome' (PID: 6376) has been terminated.

Process 'chrome' (PID: 10364) has been terminated.

Process 'chrome' (PID: 11580) has been terminated.

Process 'chrome' (PID: 12084) has been terminated.

Process 'chrome' (PID: 19148) has been terminated.



## Choosing option :4

### 1. Highest memory consuming application :

```
Terminating Highest Memory Consuming User Application
```

```
Application Name: POWERPNT.EXE
```

```
Memory Usage: 230412288 bytes
```

```
Terminate this application? (y/n): y
```

```
Process 'POWERPNT.EXE' (PID: 21492) has been terminated.
```

```
█
```

### 2. Second highest memory consuming application :

```
Terminating Highest Memory Consuming User Application
```

```
Application Name: Code.exe
```

```
Memory Usage: 192028672 bytes
```

```
Terminate this application? (y/n): █
```

# 05

## Conclusion

### Conclusion:

This dashboard offers several functionalities related to memory management and monitoring of running applications. Let's conclude by summarizing the main features and potential improvements of the project:

#### Features:

1. Memory Information Display: The dashboard provides an option to display general memory information, including total memory, available memory, used memory, and memory usage percentage.
2. Monitor Memory Usage: Users can choose to monitor the memory usage of a specific application by entering its name. The dashboard continuously updates and displays the memory usage of the selected application in terms of memory used and percentage of total memory.
3. Stop Application: The dashboard allows users to stop a specific application by entering its name. The code tries to terminate the process associated with that application.
4. User-Friendly Interface: The dashboard utilizes a simple text-based interface with clear prompts, making it easy for users to navigate and interact with the different options.

## Potential Improvements:

1. Error Handling: The code could benefit from more robust error handling. For instance, when monitoring memory usage, it should handle cases where the specified application name doesn't match any running processes.
2. Enhanced User Experience: Implementing more user-friendly messages and prompts could improve the overall experience. Additionally, providing more detailed instructions or information about each option might help users understand their choices better.
3. Monitoring Multiple Applications: The current code focuses on monitoring a single application. Extending it to monitor multiple applications simultaneously could be a useful enhancement.
4. Graphical Visualization: Adding graphical representation of memory usage, such as plots or progress bars, could provide users with a quicker and more intuitive understanding of memory trends.

In conclusion, the provided memory management dashboard project offers a foundation for monitoring and managing memory usage of specific applications. It demonstrates how to use the `psutil` library to gather memory and process information, and it provides a starting point for further development and customization based on specific needs and user feedback.





**THANK YOU**