Project Report

On

Profiling Tool Development

Submitted
In partial fulfilment for the
award of the degree of

PG-Diploma in High Performance of Computing in Application Programming

(C-DAC, ACTS (Pune))

Guided By:
Mr. Om Jadhav

Submitted By:
Sudhanshu Kamble (230340141028)
Poonam Waghmare (230340141016)
Kothimbire Dipali (230340141012)
Abhijeet Shinde (230340141001)
Aditya Bhalavi(230340141002)

Centre for Development of Advanced Computing

(C-DAC), ACTS (Pune- 411008)

Acknowledgement

This is to acknowledge our indebtedness to our Project Guide **Mr. Om Jadhav ,** C-DAC ACTS, Pune for her constant guidance and helpful suggestion for preparing this project **Profiling Tool Development.** We express our deep gratitude towards her for inspiration, personal involvement, constructive criticism that she provided us along with technical guidance during the course of this project.
We take this opportunity to thank Head of the department **Mr. Gaur Sundar** for providing us such a great infrastructure and environment for our overall development.

We express sincere thanks to **Mrs. Namrata Ailawar**, Process Owner, for their kind cooperation and extendible support towards the completion of our project.

It is our great pleasure in expressing sincere and deep gratitude towards **Mr.Himashu Sharma (Program Head)** and **Miss. Gayatri Pandit** (Course Coordinator, PG-DHPCAP) for their valuable guidance and constant support throughout this work and help to pursue additional studies.

Also, our warm thanks to **C-DAC ACTS Pune**, which provided us this opportunity to carry out, this prestigious Project and enhance our learning in various technical fields.

Poonam Waghmare (230340141016)

Kothimbire Dipali (230340141012)

Abhijeet Shinde (230340141001)

Aditya Bhalavi (230340141002)

Sudhanshu Kamble (230340141028)

ABSTRACT

High-Performance Computing (HPC) has become instrumental in solving complex problems across various domains, such as scientific research, engineering simulations, and data analytics.. This project aims to explore the capabilities of five prominent profiling and optimization toolkits: HPCToolkit, Intel Advisor, Intel VTune, TAU, and LIKWID, in the context of enhancing the performance of HPC applications.

HPCToolkit is a comprehensive performance analysis suite that provides insights into program execution behavior, bottlenecks, and parallelism. Intel Advisor offers guidance for vectorization, threading, and memory utilization, helping developers tune their code for optimal performance. Intel VTune specializes in deep profiling, allowing developers to identify performance hotspots and memory usage patterns. TAU provides profiling and tracing capabilities across various programming models and platforms, while LIKWID focuses on low-level hardware performance analysis. This project involves the creation of a script that integrates the capabilities of these toolkits into a unified workflow. The script streamlines the process of profiling and optimizing HPC applications, providing developers with a comprehensive toolkit to enhance application performance. By using this script, developers can effectively evaluate the performance characteristics of their HPC applications, identify bottlenecks, and implement targeted optimizations. In conclusion, this project demonstrates the power of combining various profiling and optimization toolkits to create a unified script for enhancing the performance of HPC applications.

Table of Contents

S. No	Title	Page No.
	Front Page Acknowledgement Abstract Table of Contents	1 2 3 4
1	Introduction	5
2	Objective	6
3	Literature Review	7
4	Methodology and Techniques	8-11
5	Implementation	12-16
6	Results	17-20
7	Conclusion	21
8	Future Scope	22
9	References	22

Introduction:

In the realm of High-Performance Computing (HPC), the quest for optimal application performance is paramount. HPC applications tackle complex challenges spanning scientific research, engineering simulations, and data analysis, necessitating tools that delve into performance nuances. This project delves into five potent profiling and optimization toolkits – HPCToolkit, Intel Advisor, Intel VTune, TAU, and LIKWID – to empower developers in enhancing HPC application performance.

Each toolkit brings a unique perspective to the optimization landscape. HPCToolkit dissects program execution behavior, Intel Advisor guides vectorization and threading, Intel VTune explores deep performance profiling, TAU offers cross-platform profiling, and LIKWID delves into hardware performance metrics. The project crafts a script that merges these toolkits, simplifying profiling and optimization workflows.

The script automates instrumentation and profiling, integrates vectorization insights, unveils intricate performance metrics, and consolidates outputs. By amalgamating these analyses, developers gain a comprehensive view of application performance bottlenecks and avenues for optimization. This integrated approach offers a holistic solution for enhancing the efficiency of HPC applications, ultimately contributing to the advancement of high-performance computing capabilities.

Objective :

To Develop an advanced High-Performance Computing (HPC) profiling tool that enables comprehensive analysis and optimization of parallel and distributed applications, facilitating improved program performance and resource utilization.
The tool goes beyond basic profiling capabilities and offers comprehensive analysis features for developers in identifying performance bottlenecks, optimizing parallel execution, and making efficient use of computing resources.
The ultimate aim is to enhance the overall performance of HPC applications, leading to faster computations and better resource allocation.

Literature Review:

- **Intel Advisor**

The tool is designed to help programmers identify performance bottlenecks and provide insights into how to improve the efficiency of their applications.

- **HPCToolkit**

HPC Toolkit is a tool for measurement and analysis of program performance on computers ranging from multicore desktop systems to the largest GPU-accelerated supercomputers.

- **Intel VTune**

It empowers developers to enhance parallelism, and improve vectorization, resulting in faster and more efficient code execution.

- **TAU**

TAU(Tuning and Analysis Utilities) toolkit is a comprehensive profiling and tracing toolkit.

- **LIKWID**

LIKWID provides a set of tools and libraries which optimizes the performance of application.

Methodology and Techniques:

- **Problem identification:**

The HPC profiling toolkit aims to address the challenge of effectively optimizing High-Performance Computing (HPC) applications. Profiling plays a pivotal role in identifying performance bottlenecks, resource utilization issues, and areas for improvement within these complex applications

- **Tool Selection :**

The selection and integration of profiling tools such as hpctoolkit, Intel VTune, Intel Advisor, TAU, and LIKWID form the core of our approach. Each tool brings unique capabilities to the table, collectively offering a well-rounded analysis of HPC application performance. This integration enables users to gain insights into various aspects of their applications, including code execution, memory usage, parallelism, and hardware-level metrics. By combining these tools, we provide a cohesive solution that empowers users to comprehensively diagnose performance issues and optimize their HPC applications effectively.

- **Script Design and Automation :**

Our toolkit employs an automated script design to simplify the profiling process. We've developed individual scripts for each profiling tool and consolidated them into a single analysis script. This approach streamlines the user experience, automates the execution of multiple tools, and ensures consistent profiling across different aspects of HPC applications. With our automated script, users can initiate a comprehensive analysis effortlessly, saving time and effort while maintaining the accuracy of results.

- **Data Collection and Analysis Workflow:**

Our toolkit follows a seamless data collection and analysis workflow. Users execute the integrated analysis script, which orchestrates the execution of the individual profiling tools. The toolkit gathers performance data, including execution traces, hardware metrics, and memory usage statistics. Once the data is collected, it is processed and analyzed, generating insightful reports and visualizations. This workflow simplifies the complex process of profiling HPC applications, enabling users to swiftly identify performance bottlenecks and make informed optimization decisions.

- **Profiling Scenarios and Use Cases:**

Our toolkit caters to diverse profiling scenarios and use cases in the realm of HPC applications. Whether it's optimizing CPU-bound tasks, improving memory-intensive processes, or enhancing parallel code performance, our integrated solution offers tailored strategies for each scenario. By leveraging the capabilities of hpctoolkit, Intel VTune, Intel Advisor, TAU, and LIKWID, users can effectively address a wide range of performance challenges, making our toolkit a versatile asset for optimizing various aspects of HPC applications.

- **Interpreting Profiling Results:**

Our toolkit employs an automated script design to simplify the profiling process. We've developed individual scripts for each profiling tool and consolidated them into a single analysis script. This approach streamlines the user experience, automates the execution of multiple tools, and ensures consistent profiling across different aspects of HPC applications. With our automated script, users can initiate a comprehensive analysis effortlessly, saving time and effort while maintaining the accuracy of results.

- **Validation and Testing**

Our toolkit's functionality is rigorously validated through comprehensive testing. We assess its performance across a spectrum of HPC applications, comparing the results against those obtained from standalone usage of profiling tools. By verifying the consistency and accuracy of our toolkit's output, we ensure that users can confidently rely on the insights it provides. This validation process guarantees that our integrated solution effectively assists users in pinpointing performance bottlenecks and optimizing their applications with a high level of accuracy and reliability.

- **Performance Overhead Analysis**

We meticulously evaluate the performance overhead introduced by our toolkit during profiling. By conducting thorough benchmarking and measuring the impact on application execution, we ensure that the added overhead remains minimal. Our goal is to provide accurate insights while keeping interference with application behavior as low as possible. This approach guarantees that the profiling process does not compromise the accuracy of results and maintains the integrity of the application's performance characteristics.

- **Future Development and Enhancement**

Looking ahead, our toolkit's future development focuses on expanding its capabilities and usability. We plan to incorporate support for additional profiling tools, enhancing its compatibility with emerging HPC architectures. Additionally, we aim to refine the automation process, streamline user interactions, and provide more advanced optimization recommendations. By continuously improving and adapting to evolving HPC landscapes, our toolkit remains a valuable resource for developers seeking to maximize the performance of their applications.

Implementation:

- ## Intel Advisor Installation Script

```
^V#!/bin/bash

ADVISOR_PACKAGE="Intel offload advisor"

if command -v $ADVISOR_PACKAGE &>/dev/null; then
    echo "Advisor is already installed."

else

          cd /home/mobaxterm/proj16/Installation/

    DOWNLOAD_COMMAND="wget https://registrationcenter-download.intel.com/akdlm/IRC_NAS/992857b9-624c-45de-9701-f6445d845359/l_BaseKit_p_2
023.2.0.49397.sh"
     $ DOWNLOAD_COMMAND

    chmod +x  l_BaseKit_p_2023.2.0.49397.sh
    PATH="/home/mobaxterm/proj16/Installation/l_BaseKit_p_2023.2.0.49397.sh"
      echo "Installing advisor..."

     sh ./PATH


  if [ $? -eq 0 ]; then
        echo "Advisor installation completed successfully."
    else
        echo "Error: Advisor installation failed. Please check your package manager and try again."
        exit 1
    fi

fi
~
```

- ## Intel Advisor Shell Script

```
#!/bin/bash


module load advisor


echo "Enter configuration:"
read config

echo "Enter accuracy:low,high,medium:"
read accuracy

echo "Enter the project directory:"
read project_dir


echo "Enter the path to the binary file:"
read binary_path

echo "Enter name for gerated report:"
read vector_report

if [ ! -f "$binary_path" ]; then
    echo "Binary path not found: $binary_path"
    exit 1
fi

advisor --collect=offload --accuracy="$accuracy" --config="$config" --project-dir="$project_dir" -- "$binary_path"

echo "Opening Intel advisor GUI...."

advixe-gui "$project_dir"

echo "profiling done!........."

~
```

- **The shell script of hpctoolkit**

```bash
#!/bin/bash

set -e
source /home/apps/spack/share/spack/setup-env.sh

spack load hpctoolkit

mkdir -p /home/shavak/hpctoolkit/output

echo "Enter executable path"
read -p "executable_path: " executable_path

echo "Choose an event: REALTIME, CPUTIME, MEMLEAK"
read -p "event_name: " event_name

executable_name=$(basename "$executable_path")
output_dir="hpctoolkit-${executable_name}-measurement"

hpcrun -e "$event_name" -o "$output_dir" -t "$executable_path"

hpcstruct "$output_dir"

#database_dir="/home/shavak/hpctoolkit/output"

hpcprof -S "$output_dir" -o "$output_dir" "$output_dir"

hpcviewer "$output_dir"

echo "HPCToolkit analysis completed."
~
~
~
~
~
```

- **Vtune Installation Script**

```bash
user24@shavak:~/project/vtune_tool

#!/bin/bash

wget = "https://registrationcenter-download.intel.com/akdlm/IRC_NAS/dfae6f23-6c90-4b9f-80e2-fa2a5037fe36/l_oneapi_vtune_p_2023.2.0.49485.sh"

chmod +x l_oneapi_vtune_p_2023.2.0.49485.sh

./l_oneapi_vtune_p_2023.2.0.49485.sh

module load vtune

echo "VTUNE VERSION"

which vtune

echo "Intel VTune Profiler installation completed."
```

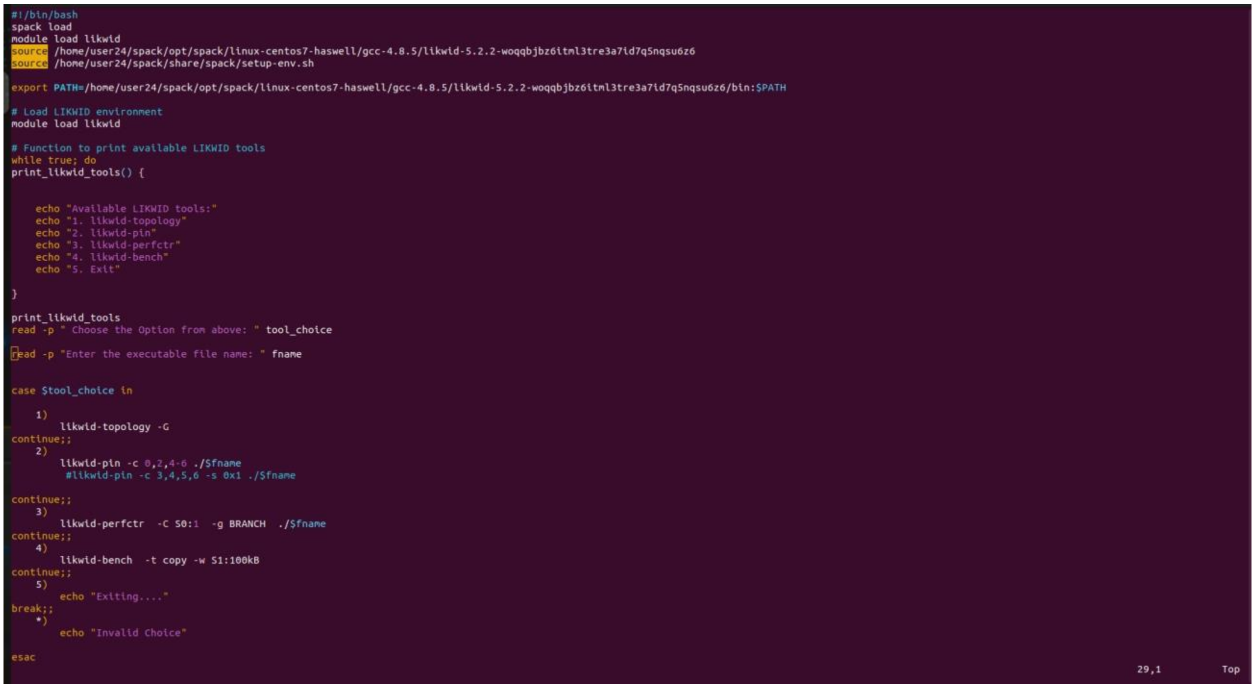
- ## The shell script of Intel VTune

```
user24@shavak:~/project/vtune_tool

#!/bin/bash

module load vtune/2023.0.0

echo "enter the events you want to profile
        (hotspots, memory-consumption, hpc-performance):"
read events

echo "enter the path to the application you want to profile:"
read app_path

echo "enter the path to the initial result directory:"
read result_dir

if [ ! -z "$(ls -A $result_dir)" ]; then

    echo "Initial result directory is not empty."

    count=1
    while true; do
        new_result_dir="${result_dir}_${count}"
        if [ ! -d "$new_result_dir" ]; then
            result_dir="$new_result_dir"
            mkdir -p "$result_dir"
            break
        fi
        ((count++))
    done
fi

vtune_cmd="vtune -collect $events -result-dir $result_dir -- $app_path"

echo "$vtune_cmd"

echo "profiling results are available in $result_dir"

res= $result_dir

vtune-gui $res
```

- ## Tau Shell Script

```
#!/bin/bash

#source /home/user24/project/spack/share/spack/setup-env.sh

#source /path/to/your/spack/share/spack/setup-env.sh

#spack load tau
#export PATH=/home/apps/spack/opt/spack/linux-centos7-cascadelake/gcc-11.2.0/tau-2.30.2-n2jpt6v25fkzhf53u5gfs2oes2v52kay:$PATH

export PATH=/home/user24/project/spack/opt/spack/linux-centos7-haswell/gcc-4.8.5/tau-2.32-3glvqcdlatqmxlpoyey6zmtd7qkbszgb/bin:$PATH

export PATH=/home/user24/project/spack/opt/spack/linux-centos7-haswell/gcc-4.8.5/tau-2.32-3glvqcdlatqmxlpoyey6zmtd7qkbszgb/lib:$PATH

export PATH=/home/user24/project/spack/opt/spack/linux-centos7-haswell/gcc-4.8.5/tau-2.32-3glvqcdlatqmxlpoyey6zmtd7qkbszgb/include:$PATH

echo "Select an option:"
echo "1. Run pprof"
echo "2. Launch paraprof"
echo "3. Manage taudb"
echo "4. Launch taudb_gui"
read -p "Enter the number of your choice: " choice

case $choice in
    1)
        echo "Enter a path: "
        read user_input
        pprof "$user_input"
        ;;
    2)
        paraprof &
        ;;
    3)
        taudb_manage
        ;;
    4)
        taudb_gui &
        ;;
    *)
        echo "Invalid choice"
        exit 1
        ;;
esac
~
~
~
~
~
~
~
~
~
~
~
"Tau_script_1.sh" 43L, 1150C                                        27,9          All
```

- **LIKWID Shell Script**



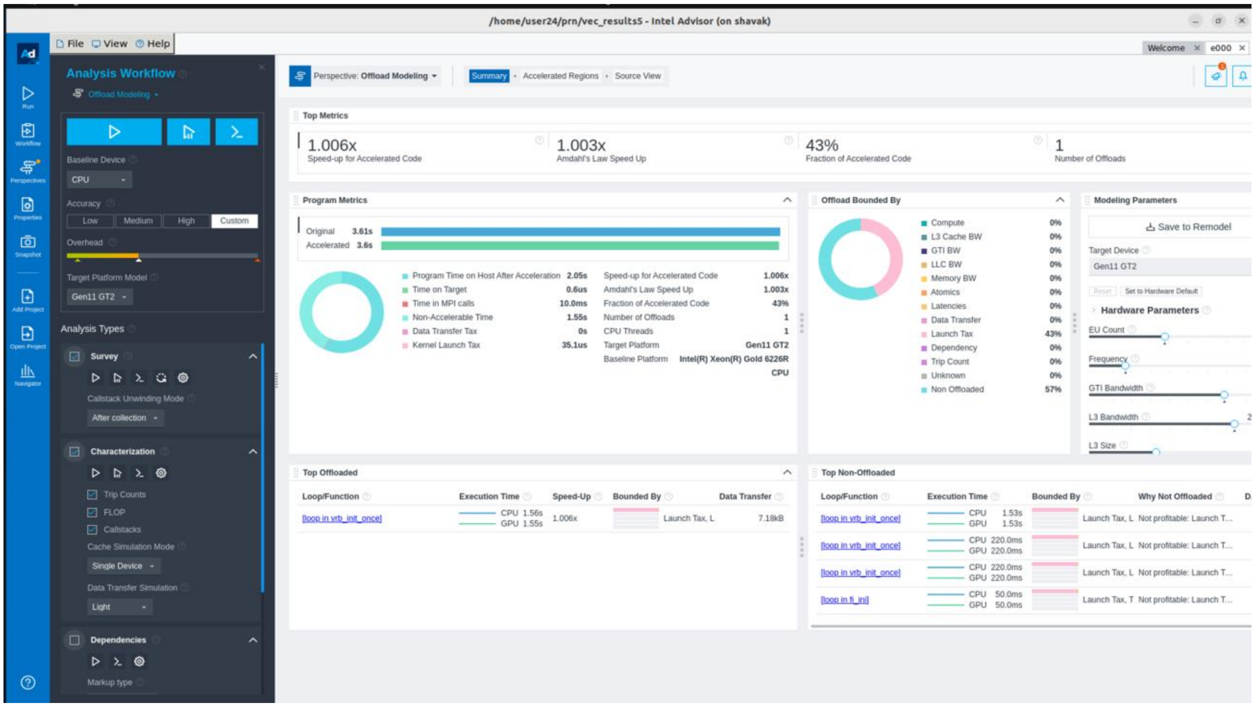
```bash
#!/bin/bash
spack load
module load likwid
source /home/user24/spack/opt/spack/linux-centos7-haswell/gcc-4.8.5/likwid-5.2.2-woqqbjbz6itml3tre3a7id7q5nqsu6z6
source /home/user24/spack/share/spack/setup-env.sh

export PATH=/home/user24/spack/opt/spack/linux-centos7-haswell/gcc-4.8.5/likwid-5.2.2-woqqbjbz6itml3tre3a7id7q5nqsu6z6/bin:$PATH

# Load LIKWID environment
module load likwid

# Function to print available LIKWID tools
while true; do
print_likwid_tools() {


    echo "Available LIKWID tools:"
    echo "1. likwid-topology"
    echo "2. likwid-pin"
    echo "3. likwid-perfctr"
    echo "4. likwid-bench"
    echo "5. Exit"

}

print_likwid_tools
read -p " Choose the Option from above: " tool_choice

read -p "Enter the executable file name: " fname


case $tool_choice in

    1)
        likwid-topology -G
continue;;
    2)
        likwid-pin -c 0,2,4-6 ./$fname
         #likwid-pin -c 3,4,5,6 -s 0x1 ./$fname

continue;;
    3)
        likwid-perfctr  -C S0:1  -g BRANCH  ./$fname
continue;;
    4)
        likwid-bench  -t copy -w S1:100kB
continue;;
    5)
        echo "Exiting...."
break;;
    *)
        echo "Invalid Choice"

esac
                                                                    29,1          Top
```

- **MAIN.sh**

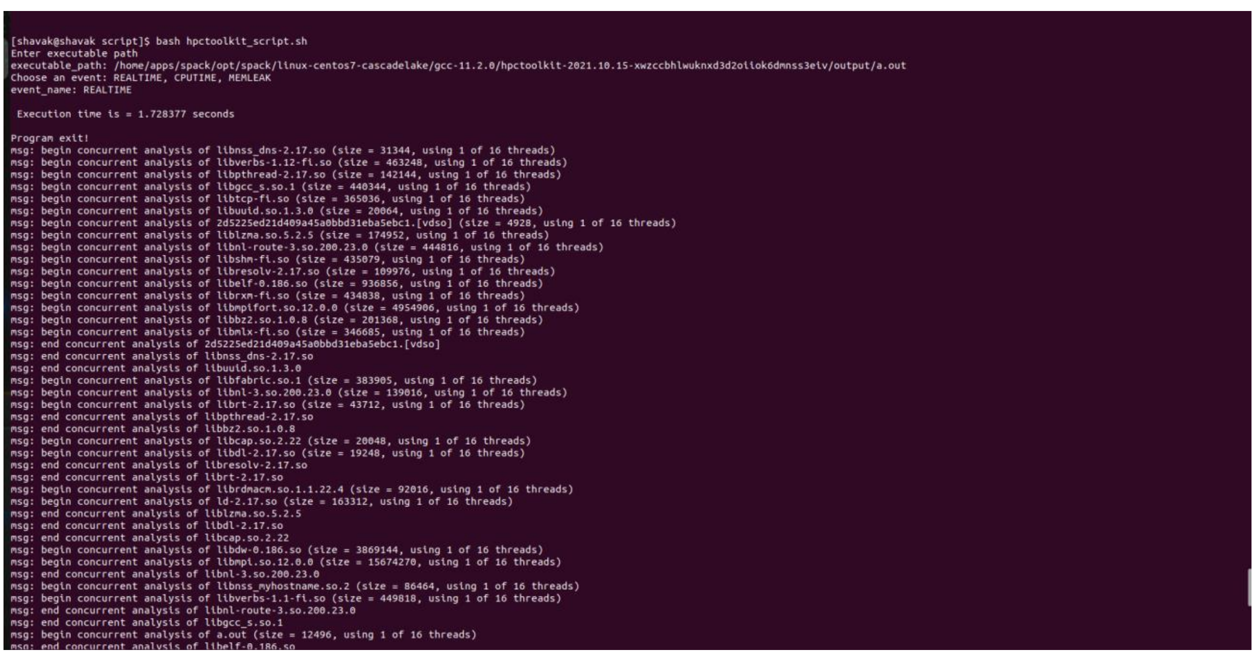
```
user24@shavak:~/prn
#!/bin/bash

set -e

select_profiling_Tool_Choice() {
    echo "Select a profiling Tool option:"

    echo "1. Intel Offload advisor"
    echo "2. Intel VTune"
    echo "3. HPCToolkit"
    echo "4. TAU"
    echo "5. Likwid"
    echo "6. Exit"
    read -p "Enter your choice: " choice
}


while true; do
    select_profiling_Tool_Choice

    case $choice in
        1)
            /home/user24/prn/IntelAdvisor_offload_script.sh

            ;;
        2)
            /home/user24/project/vtune_tool/main_vtune.sh

            ;;
        3)
            /home/apps/spack/opt/spack/linux-centos7-cascadelake/gcc-11.2.0/hpctoolkit-2021.10.15-xwzccbhlwuknxd3d2oiiok6dmnss3eiv/script/hpctoolkit_script.sh
            ;;
        4)
            /home/user24/project/tau/tau_main.sh

            ;;
        5)
            /home/user24/likwid_Scripts/likwid_script_v3.sh
            ;;
        6)
            echo "Exiting."
            exit
            ;;
        *)
            echo "Invalid choice. Please select a valid option."
            ;;
    esac
done
~
~
~
~
~
~
"all_tool.sh" 48L, 1095C
```

RESULTS:

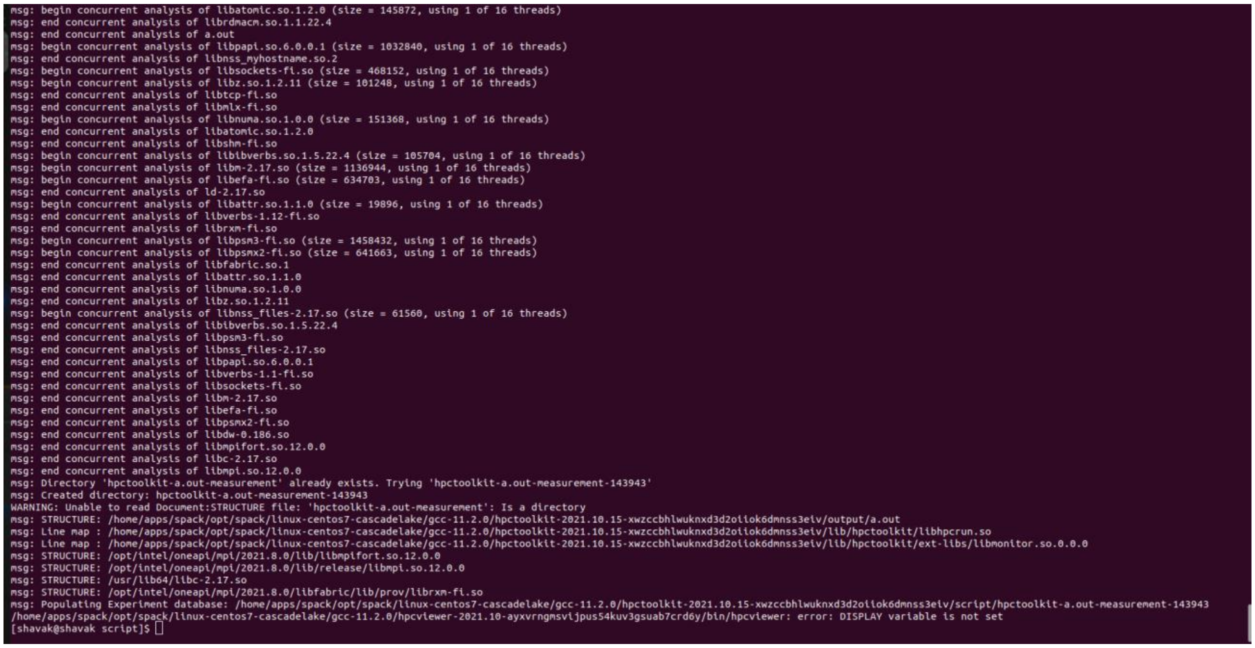
- ## IntelAdvisor



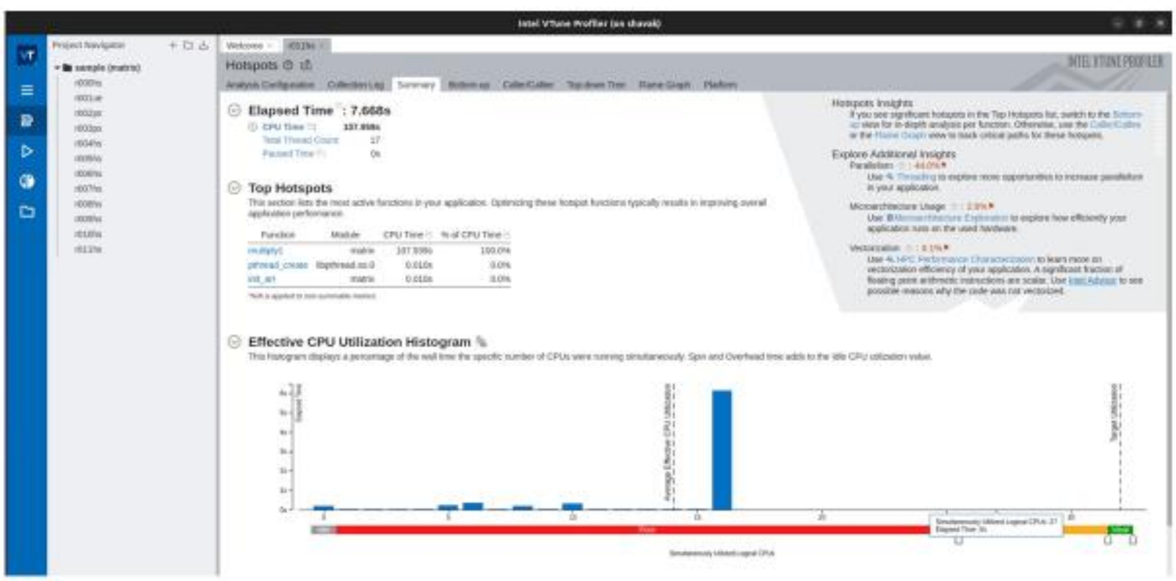
- ## HPCTOOLKIT



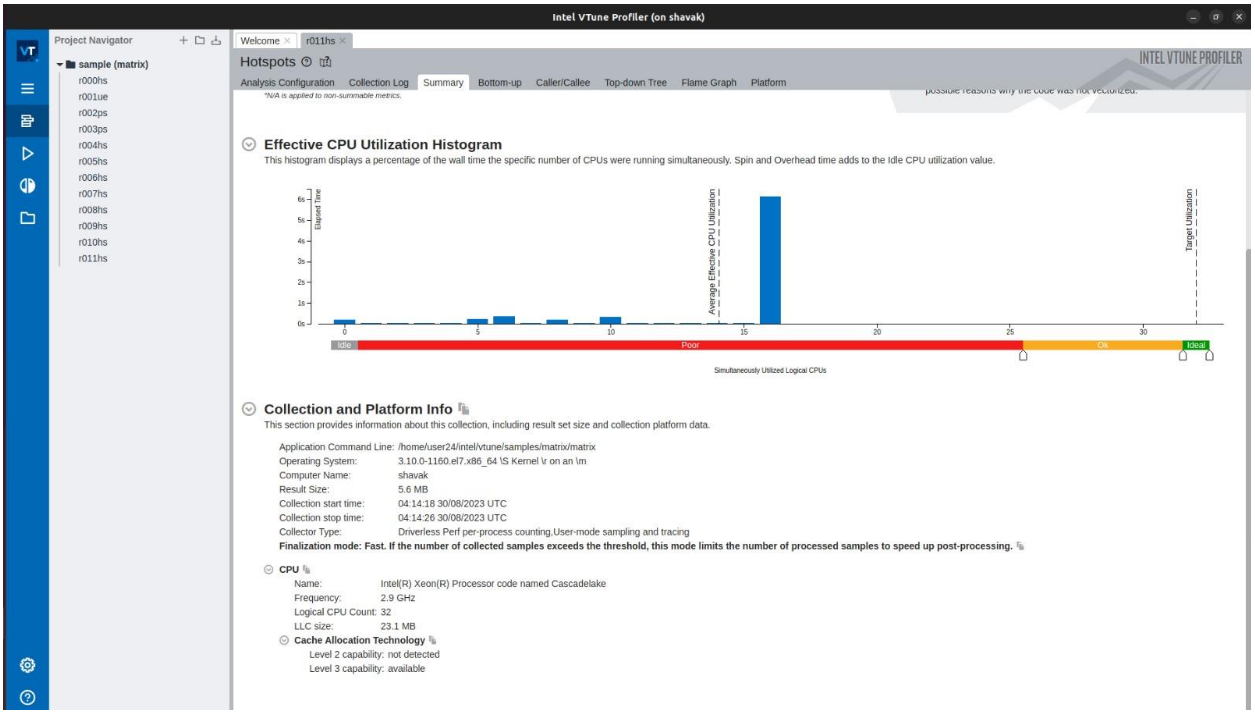
- **HPCTOOLKIT**



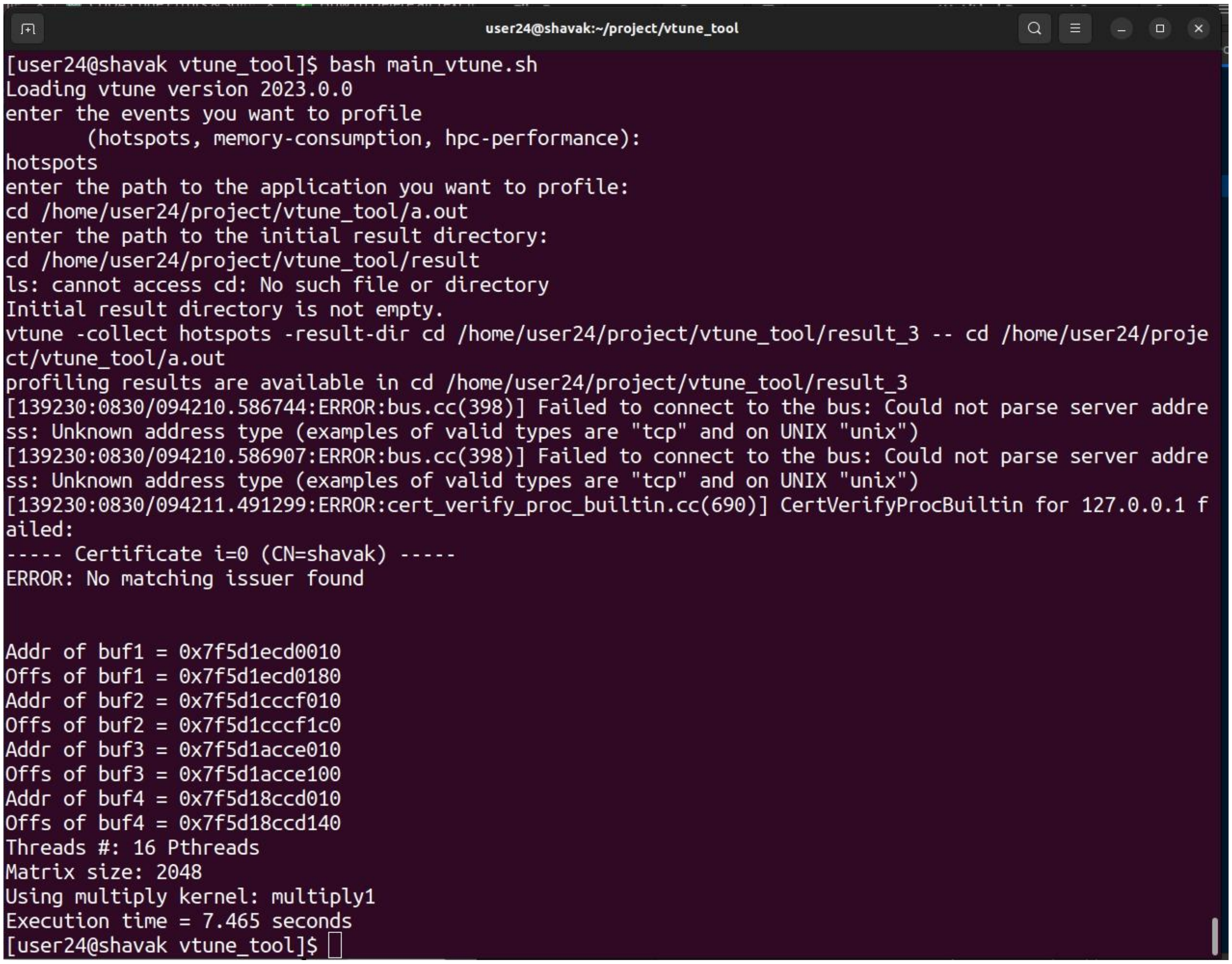
- **Intel Vtune**



- **Intel Vtune**


Intel VTune Profiler (on shavak)
Project Navigator
sample (matrix)
r000hs
r001ue
r002ps
r003ps
r004hs
r005hs
r006hs
r007hs
r008hs
r009hs
r010hs
r011hs
Welcome
r011hs
Hotspots
INTEL VTUNE PROFILER
Analysis Configuration
Collection Log
Summary
Bottom-up
Caller/Callee
Top-down Tree
Flame Graph
Platform
Effective CPU Utilization Histogram
This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU utilization value.
Elapsed Time
Average Effective CPU Utilization
Target Utilization
Idle
Poor
Ok
Ideal
Simultaneously Utilized Logical CPUs
Collection and Platform Info
This section provides information about this collection, including result set size and collection platform data.
Application Command Line: /home/user24/intel/vtune/samples/matrix/matrix
Operating System: 3.10.0-1160.el7.x86_64 \S Kernel \r on an \m
Computer Name: shavak
Result Size: 5.6 MB
Collection start time: 04:14:18 30/08/2023 UTC
Collection stop time: 04:14:26 30/08/2023 UTC
Collector Type: Driverless Perf per-process counting,User-mode sampling and tracing
Finalization mode: Fast. If the number of collected samples exceeds the threshold, this mode limits the number of processed samples to speed up post-processing.
CPU
Name: Intel(R) Xeon(R) Processor code named Cascadelake
Frequency: 2.9 GHz
Logical CPU Count: 32
LLC size: 23.1 MB
Cache Allocation Technology
Level 2 capability: not detected
Level 3 capability: available

- **Intel Vtune**



```
user24@shavak:~/project/vtune_tool
[user24@shavak vtune_tool]$ bash main_vtune.sh
Loading vtune version 2023.0.0
enter the events you want to profile
       (hotspots, memory-consumption, hpc-performance):
hotspots
enter the path to the application you want to profile:
cd /home/user24/project/vtune_tool/a.out
enter the path to the initial result directory:
cd /home/user24/project/vtune_tool/result
ls: cannot access cd: No such file or directory
Initial result directory is not empty.
vtune -collect hotspots -result-dir cd /home/user24/project/vtune_tool/result_3 -- cd /home/user24/proje
ct/vtune_tool/a.out
profiling results are available in cd /home/user24/project/vtune_tool/result_3
[139230:0830/094210.586744:ERROR:bus.cc(398)] Failed to connect to the bus: Could not parse server addre
ss: Unknown address type (examples of valid types are "tcp" and on UNIX "unix")
[139230:0830/094210.586907:ERROR:bus.cc(398)] Failed to connect to the bus: Could not parse server addre
ss: Unknown address type (examples of valid types are "tcp" and on UNIX "unix")
[139230:0830/094211.491299:ERROR:cert_verify_proc_builtin.cc(690)] CertVerifyProcBuiltin for 127.0.0.1 f
ailed:
----- Certificate i=0 (CN=shavak) -----
ERROR: No matching issuer found


Addr of buf1 = 0x7f5d1ecd0010
Offs of buf1 = 0x7f5d1ecd0180
Addr of buf2 = 0x7f5d1cccf010
Offs of buf2 = 0x7f5d1cccf1c0
Addr of buf3 = 0x7f5d1acce010
Offs of buf3 = 0x7f5d1acce100
Addr of buf4 = 0x7f5d18ccd010
Offs of buf4 = 0x7f5d18ccd140
Threads #: 16 Pthreads
Matrix size: 2048
Using multiply kernel: multiply1
Execution time = 7.465 seconds
[user24@shavak vtune_tool]$
```

Conclusion

- In conclusion, the development of our profiling tool was an enlightening experience that allows us to develop one profiling toolkit that combines the features of five different toolkits we have used.
- The choice of a profiling tool depends on the specific goals, application domain, and hardware platform. Developers seeking to optimize performance should consider the features of these tools and choose the one that aligns best with their requirements.
- The project is not only expanded our understanding of profiling techniques but also the importance of selecting the right tool for the task.
- The ongoing evolution of these tools ensures that software developers have robust toolkit to analyze, optimize, and enhance the performance of their applications in an increasingly demanding computing landscape.

Future Scope

- Performance Portability: As software spans various hardware architectures, from traditional CPUs to GPUs, FPGAs, and accelerators, profiling tools will need to offer performance analysis across heterogeneous platforms.
- Tools that can analyze code behavior and performance on different architectures will be essential for optimizing applications for various target systems.
- Server less and Function-as-a-Service: These tools will help developers optimize server less functions for better execution times and resource utilization.

References

- http://hpctoolkit.org/
- https://www.intel.com/content/www/us/en/developer/tools/oneapi/vtune-profiler.html#gs.4k990l
- https://www.intel.com/content/www/us/en/developer/tools/oneapi/advisor.html#gs.4k9a64
- https://hpc.llnl.gov/software/development-environment-software/tau-tuning-and-analysis-utilities
- https://www.cs.uoregon.edu/research/tau/home.php
- https://hpc-wiki.info/hpc/Likwid
- https://hpc.fau.de/research/tools/likwid/
- https://chat.openai.com/?model=text-davinci-002-render-sha