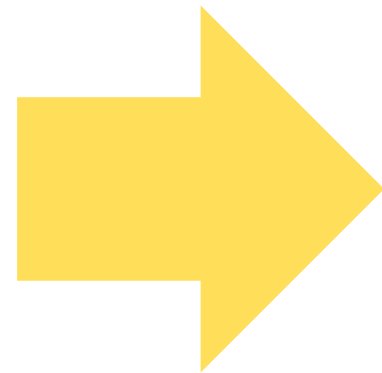


**Shamjith K V**  
Scientist-F

**Abhishek Patil**  
Project Associate

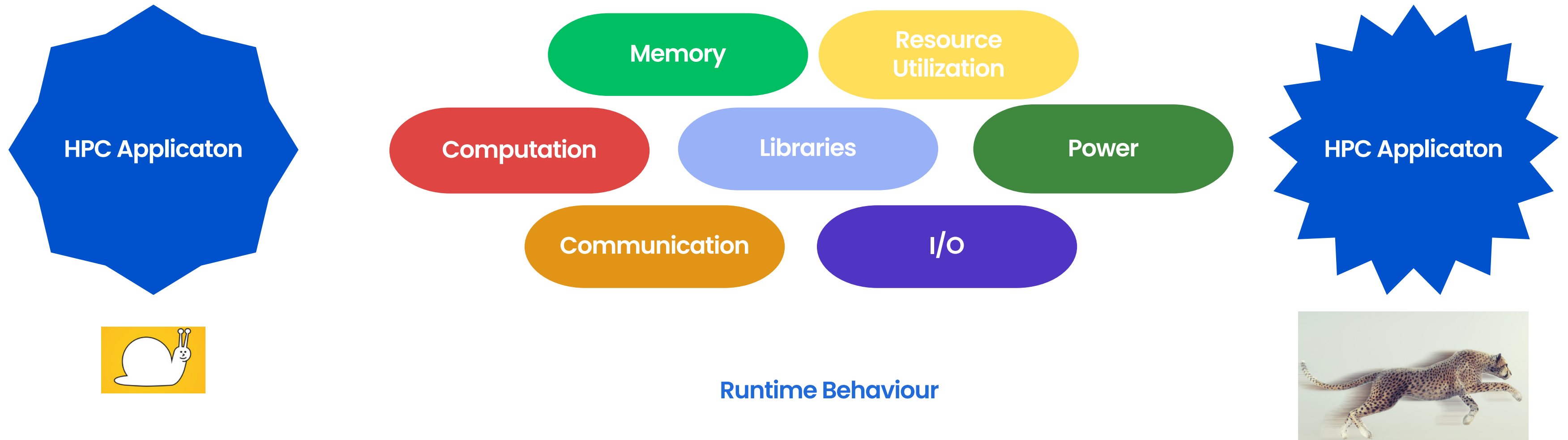


# HPC Applications



HPC Cluster

# HPC Applications – Runtime Behaviour



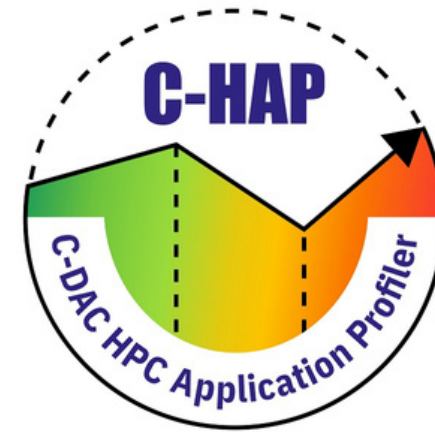


# HPC Application Profiling

HPC application profiling helps identify **performance bottlenecks**, **optimize code**, and **improve efficiency**. It provides insights into the application's **runtime behavior** and **resource utilization**, leading to better **scalability**, **faster execution**, and **reduced cost**.



# CDAC HPC Application Profiler (C-HAP)

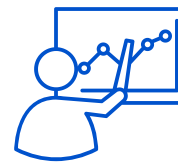


# Key Features



## Application Performance summary

Brings out the details of execution time , no. of processes, architecture, application characteristics



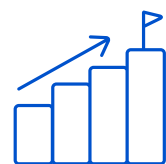
## Guided profiling

From the performance summary, user can get to the specific sections of bottlenecks with respect each processes and threads



## Hotspot identification

Sections of the code that consumes more time compared to other sections of code per each process



## Potential Performance suggestions

Provides verbose suggestions for improving the hotspot sections, like vectorizing certain sections, Combining the arithmetic operations, Info on Slow Data structures.



## Multi-dimension analysis

Analysing the application with respect to execution, memory usage, communication, I/O and Power



## Minimal Overhead

The overhead added for carrying out profiling is very minimal.



## Integration with LRMs

The software is well integrated with SLURM resource manager so that it can seamlessly submit jobs on to the HPC cluster for performance analysis



## Web Interface

The profiling job submission, and analysis are provided through a web interface to avoid the trouble in using commandline interfaces

# Job Submission for Profiling

HPC PROFILER

Dashboard

Job Submission

Application Profile Info

Executable

Source

Inputs

Job Type

No. of Procs

Job to Profile

Job Name

Executable

Source

Input Parameters

Type of Job

No. of Processes

Advanced Options

Submit

Reset

Profiling Status

Last 10 Jobs

Show entries

Search:

Job Id	Job Name	Exec Status	Profile Status
19829	kmeansDM_senthilsir	COMPLETED	<a href="#">View Profile Data</a>
19828	kmeans8_28_09	COMPLETED	<a href="#">View Profile Data</a>
19827	kmeans16_28_09	CANCELLED by 1033	Job is Cancelled
19826	kdemo_28_09	COMPLETED	<a href="#">View Profile Data</a>
19820	kmeans_testing_source_file	COMPLETED	<a href="#">View Profile Data</a>
19819	kmeansRErun	COMPLETED	<a href="#">View Profile Data</a>
19818	kmeansDM3_xhost	COMPLETED	<a href="#">View Profile Data</a>
19810	kmeansDM2	COMPLETED	<a href="#">View Profile Data</a>
19809	KmeansDM1	COMPLETED	<a href="#">View Profile Data</a>
19807	kmeans_AVX512_26-09	COMPLETED	<a href="#">View Profile Data</a>

Showing 1 to 10 of 10 entries

Previous1Next

# Application Summary

CDAC HPC PROFILER

Dashboard

Job Submission

Application Profile Info

Profile Level

Application Summary

Node Name

Nodes

Process Id

Process Id

Application "Demo\_IIT\_CDAC\_Meet" Summary

Experiment Name	batch_test
Application	/home/neerajs/jobInfo/kmeans_-g
Timestamp	2022-12-09 15:57:43
Experiment Type	MPI
Machine	ssl-cn02
Architecture	x86_64
Micro Architecture	SKYLAKE
Model Name	Intel(R) Xeon(R) Gold 6126 CPU @ 2.60GHz
Cache Size	19712 KB
Number of Cores	12
OS Version	Linux 3.10.0-862.el7.x86_64 #1 SMP Fri Apr 20 16:44:24 UTC 2018
Total no. of Process	8

Potential Speedup

Application "Demo\_IIT\_CDAC\_Meet" Potential Speed Up

Potential Speedup If Fully Vectorised	4.82
Potential Speedup If Only FP Arithmetic	2.69

Configuration Summary

run_command	10000 100 100
profile_start	{ unit = s ; value = 0 ; }
mpi_command	srunk --mpi=pmi2 --job-name=Demo_IIT_CDAC_Meet --ntasks= --ntasks-per-node=
omp_num_threads	1

Application "Demo\_IIT\_CDAC\_Meet" Execution Summary

Total Time (s)	143.75
Time in Analyzed Loops (%)	90.6
Time in Analyzed Innermost Loops (%)	90.6
Compilation Option Used	kmeans_-g: Intel 18.0.3.222 - l/opt/ohpc/pub/intel_2018/compilers_and_libraries_2018.3.222/linux/mpi/intel64/include - std=c99 -g -o ./g/kmeans_-g - L/opt/ohpc/pub/intel_2018/compilers_and_libraries_2018.3.222/linux/mpi/intel64/lib/debug -L/opt/ohpc/pub/intel_2018/compilers_and_libraries_2018.3.222/linux/mpi/intel64/lib -Xlink enable-new-dtags -Xlinker -rpath -Xlinker

Application "Demo\_IIT\_CDAC\_Meet" Characterization

Application is bound to User Code

Computation time	91.1%	Time spent in running application code,High values are usually good This is high, Check the CPU performance section, TMAM and vectorization for more advise
MPI time	7.12%	Time spent in MPI library call code,High values are usually bad This is very low, This code may benefit from a higher processor count
OMP time	0	Time spent in OMP region,High values are usually bad This is negligibile, focus on improving other section first
IO time	0.23%	Time spent in Filesystem IO,High values are usually bad This is negligibile, focus on improving other section first

Application and Architecture

Application Characterization



# Process Level Profiling Information

CDAC  
HPC PROFILER

Process level information

Job Submission

> Application Profile Info

Profile Level

Process Level

Node Name

node\_ssl-cn01

Process Id

214856

Application "pmi2\_16p\_testing" Program Sections

Show 10 entries

Search:

Function Name	Module	Source Info	Coverage (%)	Time w.r.t Walltime (s)	Time Min (s) (TID)
__GI__printf_fp_l	libc.so.6	NA	11.05	0.11	0.11 (214856)
__mpn_mul_1	libc.so.6	NA	1.32	0.01	0.01 (214856)
__parse_one_specmb	libc.so.6	NA	1.21	0.01	0.01 (214856)
_IO_default_xsputn	libc.so.6	NA	1.24	0.01	0.01 (214856)
_IO_file_xsputn	libc.so.6	NA	2.00	0.02	0.02 (214856)
_IO_vfprintf	libc.so.6	NA	6.71	0.07	0.07 (214856)
buffered_vfprintf	libc.so.6	NA	2.57	0.03	0.03 (214856)
hack_digit.13661	libc.so.6	NA	3.02	0.03	0.03 (214856)
main	kmeans_-g-O3	kmeans.c:8-206 stdlib.h:280-280	59.68	0.61	0.61 (214856)

Loop ID	Module	Source Info	Function Name	Level	Coverage %
31	kmeans_-g-O3	kmeans.c:19-21	main	Innermost	26.48
29	kmeans_-g-O3	kmeans.c:18-150	main	InBetween	16.07
30	kmeans_-g-O3	kmeans.c:19-21	main	Innermost	13.16

Suggestions

FMA

Presence of both ADD/SUB and MUL operations.  
Workaround(s):  
- Pass to your compiler a micro-architecture specialization option:  
  \* use axHost or xHost  
- Try to change order in which elements are evaluated (using parentheses) in arithmetic expres  
For instance a + b\*c is a valid FMA (MUL then ADD).  
However (a+b)\* c cannot be translated into an FMA (ADD then MUL).

Source Code

```
6 // Creates an array of random floats. Each number has a value from 0 - 1
7 float* create_rand_nums(const int num_elements) {
8     float *rand_nums = (float *)malloc(sizeof(float) * num_elements);
9     assert(rand_nums != NULL);
10    for (int i = 0; i < num_elements; i++) {
11        rand_nums[i] = (rand() / (float)RAND_MAX);
12    }
13    return rand_nums;
14 }
15
16 // Distance**2 between d-vectors pointed to by v1, v2.
17 float distance2(const float *v1, const float *v2, const int d) {
18     float dist = 0.0;
19    for (int i=0; i<d; i++) {
20        float diff = v1[i] - v2[i];
21        dist += diff * diff;
22    }
23    return dist;
24 }
25
26 // Assign a site to the correct cluster by computing its distances to
27 // each cluster centroid.
28 int assign_site(const float* site, float* centroids,
29                const int k, const int d) {
30     int best_cluster = 0;
31     float best_dist = distance2(site, centroids, d);
32     float* centroid = centroids + d;
33    for (int c = 1; c < k; c++, centroid += d) {
```

Time coverage  
(loops and functions)

# Hotspot Identification

CDAC  
HPC PROFILER

Dashboard

Job Submission

Application Profile Info

Profile Level

Process Level

Node Name

node\_ssl-cn01

Process Id

214856

Application "pmi2\_16p\_testing" Program Sections

Show 10 entries

Search:

Function Name	Module	Source Info	Coverage (%)	Walltime (s)	(s) (TID)
__GI__printf_fp_l	libc.so.6	NA	11.05	0.11	0.11 (214856)
__mpn_mul_1	libc.so.6	NA	1.32	0.01	0.01 (214856)
__parse_one_specmb	libc.so.6	NA	1.21	0.01	0.01 (214856)
_IO_default_xsputn	libc.so.6	NA	1.24	0.01	0.01 (214856)
_IO_file_xsputn	libc.so.6	NA	2.00	0.02	0.02 (214856)
_IO_vfprintf	libc.so.6	NA	6.71	0.07	0.07 (214856)
buffered_vfprintf	libc.so.6	NA	2.57	0.03	0.03 (214856)
hack_digit.13661	libc.so.6	NA	3.02	0.03	0.03 (214856)
main	kmeans_-g-O3	kmeans.c:8-206 stdlib.h:280-280	59.68	0.61	0.61 (214856)

Loop ID	Module	Source Info	Function Name	Level	Coverage %
31	kmeans_-g-O3	kmeans.c:19-21	main	Innermost	26.48
29	kmeans_-g-O3	kmeans.c:18-150	main	InBetween	16.07
30	kmeans_-g-O3	kmeans.c:19-21	main	Innermost	13.16

Suggestions

FMA

Presence of both ADD/SUB and MUL operations.  
Workaround(s):  
- Pass to your compiler a micro-architecture specialization option:  
\* use axHost or xHost  
- Try to change order in which elements are evaluated (using parentheses) in arithmetic expres  
For instance a + b\*c is a valid FMA (MUL then ADD).  
However (a+b)\* c cannot be translated into an FMA (ADD then MUL).

Source Code

```
6 // Creates an array of random floats. Each number has a value from 0 - 1
7 float* create_rand_nums(const int num_elements) {
8     float *rand_nums = (float *)malloc(sizeof(float) * num_elements);
9     assert(rand_nums != NULL);
10    for (int i = 0; i < num_elements; i++) {
11        rand_nums[i] = (rand() / (float)RAND_MAX);
12    }
13    return rand_nums;
14 }
15
16 // Distance**2 between d-vectors pointed to by v1, v2.
17 float distance2(const float *v1, const float *v2, const int d) {
18     float dist = 0.0;
19    for (int i=0; i<d; i++) {
20        float diff = v1[i] - v2[i];
21        dist += diff * diff;
22    }
23    return dist;
24 }
25
26 // Assign a site to the correct cluster by computing its distances to
27 // each cluster centroid.
28 int assign_site(const float* site, float* centroids,
29                const int k, const int d) {
30     int best_cluster = 0;
31     float best_dist = distance2(site, centroids, d);
32     float* centroid = centroids + d;
33     for (int c = 1; c < k; c++, centroid += d) {
```

Hotspot Detection

# Application Profiling – without optimization

CDAC  
HPC PROFILER

Dashboard

Job Submission

Application Profile Info

Profile Level

Application Summary

Node Name

Nodes

Process Id

Process Id

Application "mpiExp8" Summary

Experiment Name	batch_test
Application	/home/neerajs/jobInfo/kmeans_mpiicc_O3
Timestamp	2022-09-05 14:16:50
Experiment Type	MPI
Machine	ssl-cn01
Architecture	x86_64
Micro Architecture	SKYLAKE
Model Name	Intel(R) Xeon(R) Gold 6126 CPU @ 2.60GHz
Cache Size	19712 KB
Number of Cores	12
OS Version	Linux 3.10.0-862.el7.x86_64 #1 SMP Fri Apr 20 16:44:24 UTC 2018
Number of processes observed	0
Number of threads observed	0

Application "mpiExp8" Potential Speed Up

Potential Speedup If Fully Vectorised	1.43
Potential Speedup If Only FP Arithmetic	1.10

Configuration Summary

run_command	10000 100 100
profile_start	{ unit = s ; value = 0 ; }
mpi_command	srun --mpi=pmi2 --job-name=mpiExp8 --ntasks= --ntasks-per-node=
omp_num_threads	1

Application "mpiExp8" Execution Summary

Total Time (s)	25.93
Time in Analyzed Loops (%)	44.3
Time in Analyzed Innermost Loops (%)	32.2
Compilation Option Used	
Suggested Compilation Options	Not Available

Application "mpiExp8" Characterization

Application is bound to User Code

Computation

Time spent in running application code,High values are

Characteristics

Execution Time



# Application Profiling – with optimization

CDAC  
HPC PROFILER

Dashboard

Job Submission

Application Profile Info

Application "retestMPI2" Summary

Experiment Name	batch_test
Application	/home/neerajs/jobInfo/kmeans_-g-O3-xHost
Timestamp	2022-09-05 14:44:06
Experiment Type	MPI
Machine	ssl-cn01
Architecture	x86_64
Micro Architecture	SKYLAKE
Model Name	Intel(R) Xeon(R) Gold 6126 CPU @ 2.60GHz
Cache Size	19712 KB
Number of Cores	12
OS Version	Linux 3.10.0-862.el7.x86_64 #1 SMP Fri Apr 20 16:44:24 UTC 2018
Number of processes observed	0
Number of threads observed	0

Execution Time

Application "retestMPI2" Characterization

Application is bound to User Code

Computation time	26.8%	Time spent in running application code,High values are usually good This is very low, focus on improving other section first
MPI time	59.7%	Time spent in MPI library call code,High values are usually bad This is high, Check the MPI breakdown for improvements
OMP time		Time spent in OMP region.High values are usually bad

Characteristics

Application "retestMPI2" Potential Speed up

Potential Speedup If Fully Vectorised	1.03
Potential Speedup If Only FP Arithmetic	1.13

Configuration Summary

run_command	10000 100 100
profile_start	{ unit = s ; value = 0 ; }
mpi_command	srunk --mpi=pmi2 --job-name=retestMPI2 --ntasks= --ntasks-per-node=
omp_num_threads	1

Application "retestMPI2" Execution Summary

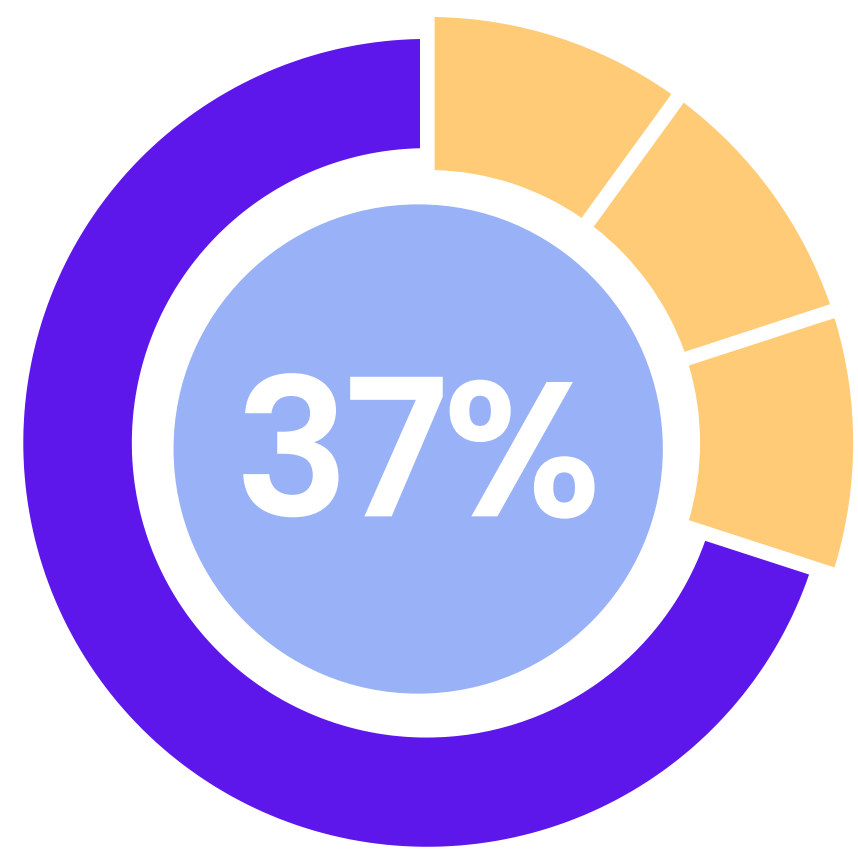
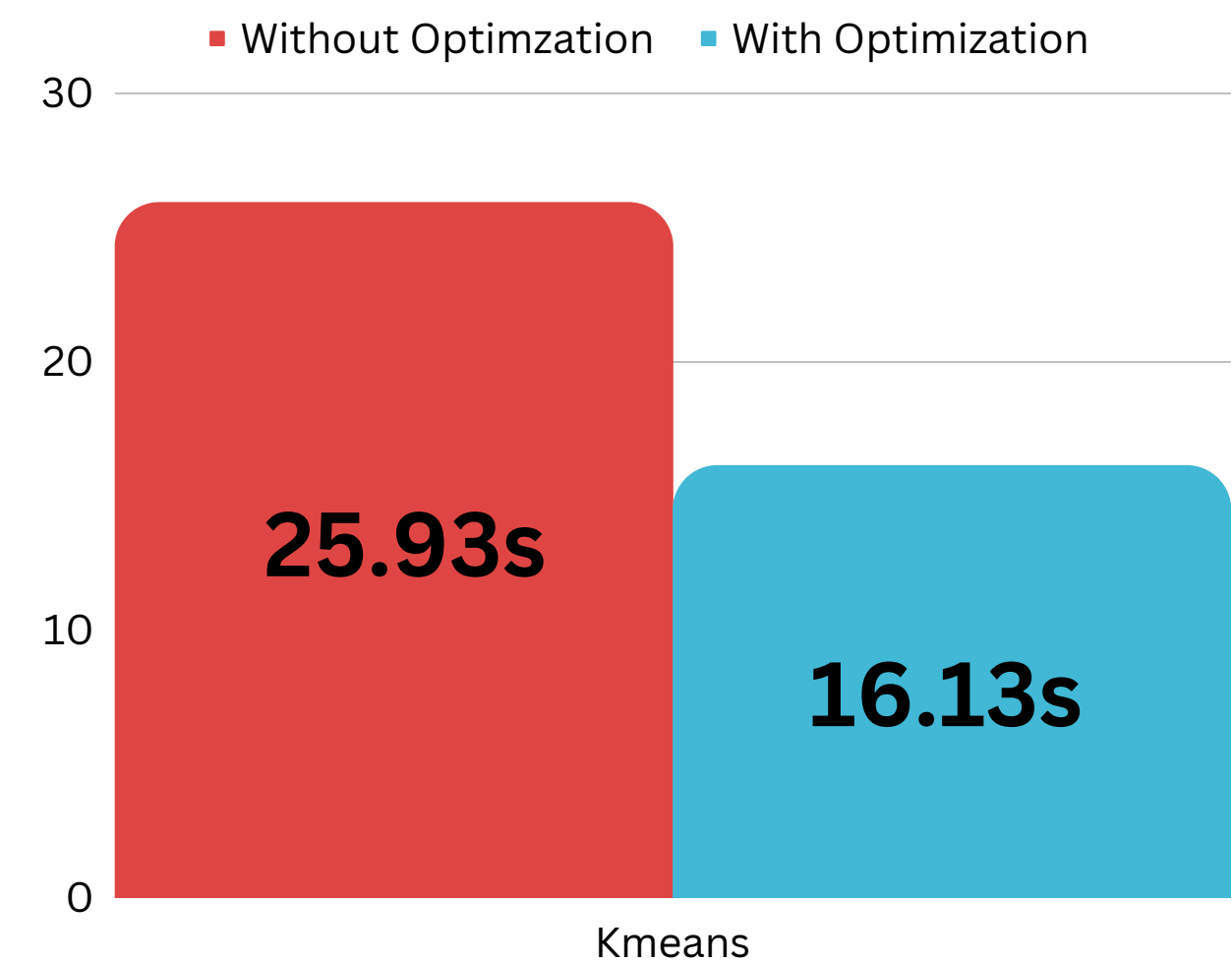
Total Time (s)	16.13
Time in Analyzed Loops (%)	26.9
Time in Analyzed Innermost Loops (%)	11.9
Compilation Option Used	kmeans_-g-O3-xHost: Intel 18.0.3.222 - l/opt/ohpc/pub/intel_2018/compilers_and_libraries_2018.3.222/linux std=c99 -g -O3 -xHost -o ./g/kmeans_-g-O3-xHost - L/opt/ohpc/pub/intel_2018/compilers_and_libraries_2018.3.222/linux -L/opt/ohpc/pub/intel_2018/compilers_and_libraries_2018.3.222/linux enable new dtags: Xlinker, rpath, Xlinker

# Applications

# KMeans

## Without optimizations

Total Time (s)	25.93
Time in Analyzed Loops (%)	44.3
Time in Analyzed Innermost Loops (%)	32.2
Compilation Option Used	
Suggested Compilation Options	Not Available



## With optimizations

Total Time (s)	16.13
Time in Analyzed Loops (%)	26.9
Time in Analyzed Innermost Loops (%)	11.9
Compilation Option Used	kmeans_-g-O3-xHost: Intel 18.0.3.222 - l/opt/ohpc/pub/intel_2018/compilers_and_libraries_2018.3.222/linux std=c99 -g -O3 -xHost -o ./g/kmeans_-g-O3-xHost - L/opt/ohpc/pub/intel_2018/compilers_and_libraries_2018.3.222/linu -L/opt/ohpc/pub/intel_2018/compilers_and_libraries_2018.3.222/linu enable-new-dtags -Xlinker -rpath -Xlinker /opt/ohpc/pub/intel_2018/compilers_and_libraries_2018.3.222/linux -Xlinker -rpath -Xlinker



# Matrix Multiplication

## Without optimizations

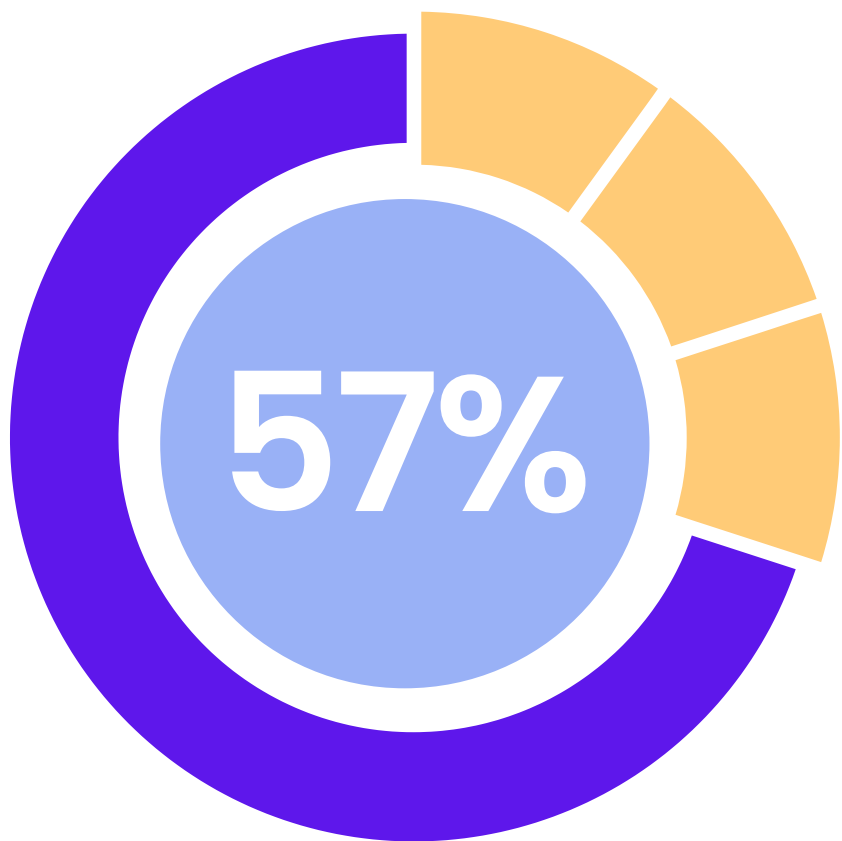
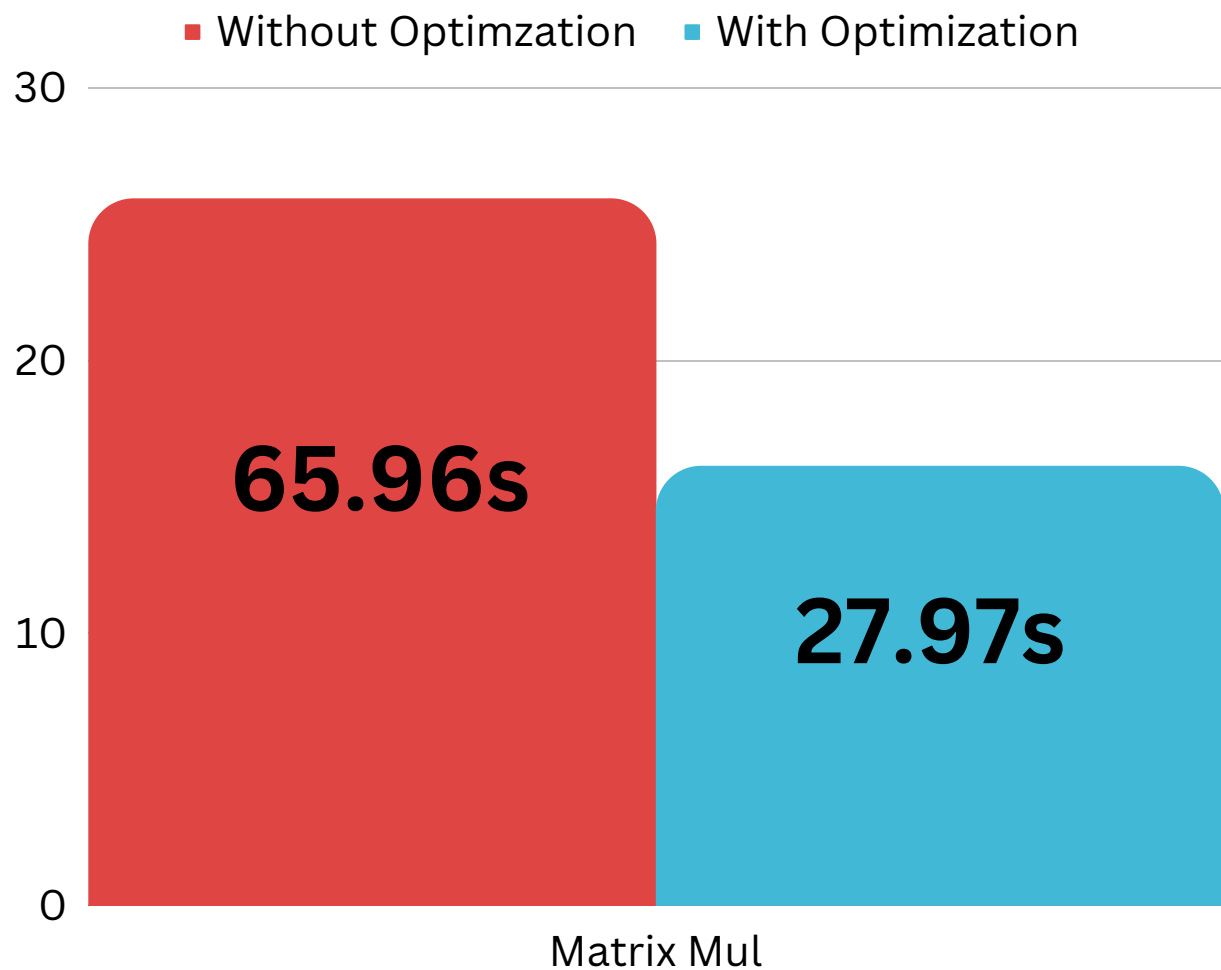
Application "mm\_mpi\_interchnage\_1" Execution Summary

Total Time (s)	65.96
Time in Analyzed Loops (%)	29.3
Time in Analyzed Innermost Loops (%)	29.3
Compilation Option Used	mm_mpi_16384_interchnage: GNU 9.2.0 -mtune=generic -march=x86-64 -g
Suggested Compilation Options	mm_mpi_16384_interchnage: -O2

## With optimizations

Application "mpi\_16384\_21\_06" Execution Summary

Total Time (s)	27.97
Time in Analyzed Loops (%)	42.7
Time in Analyzed Innermost Loops (%)	42.7
Compilation Option Used	mm_mpi_16384: GNU 9.2.0 -mtune=generic -march=x86-64 -g -std=c99
Suggested Compilation Options	mm_mpi_16384: -O2

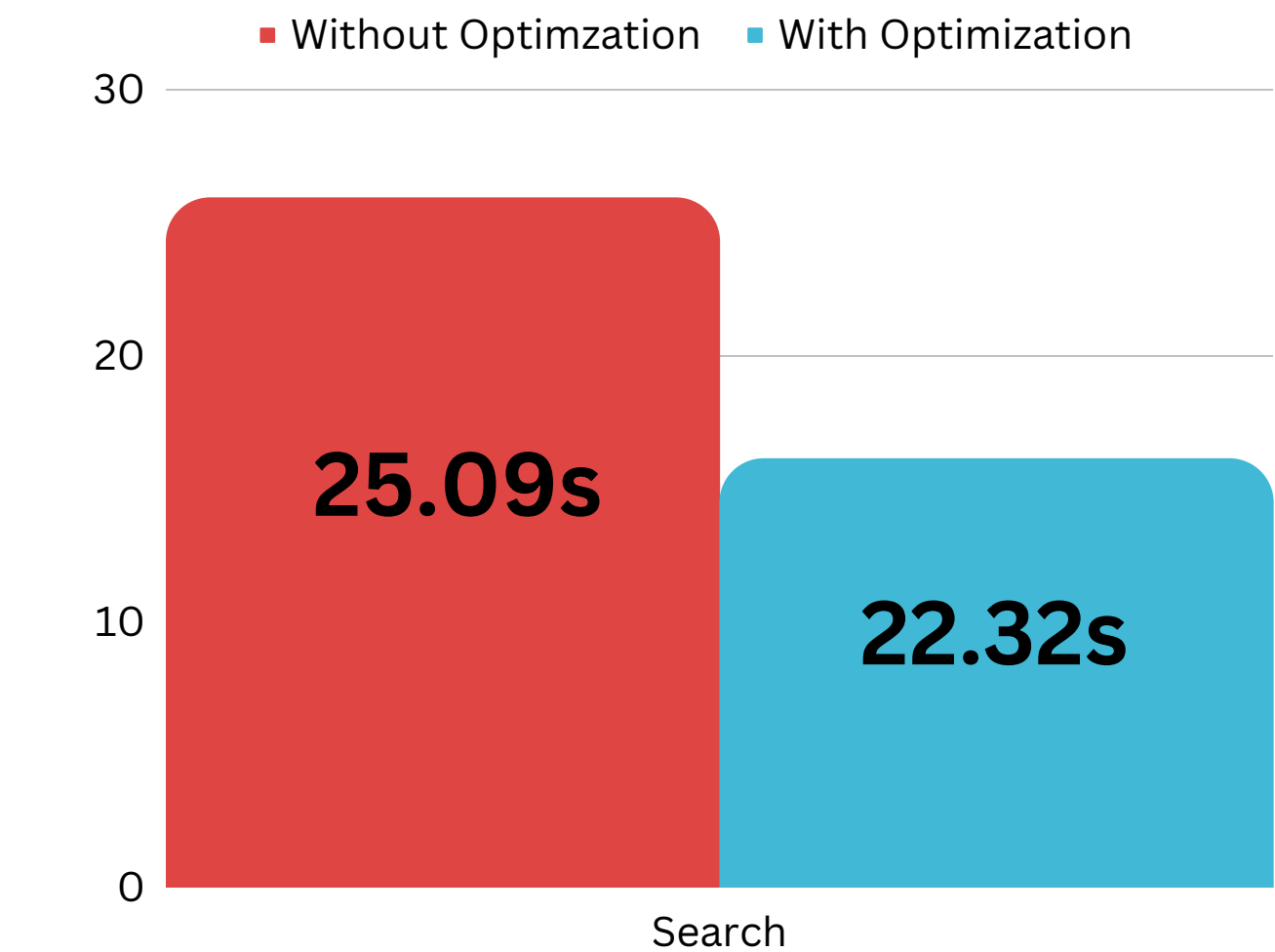


# Search App

## Without optimizations

Application "job\_search\_mpi\_21\_06" Execution Summary

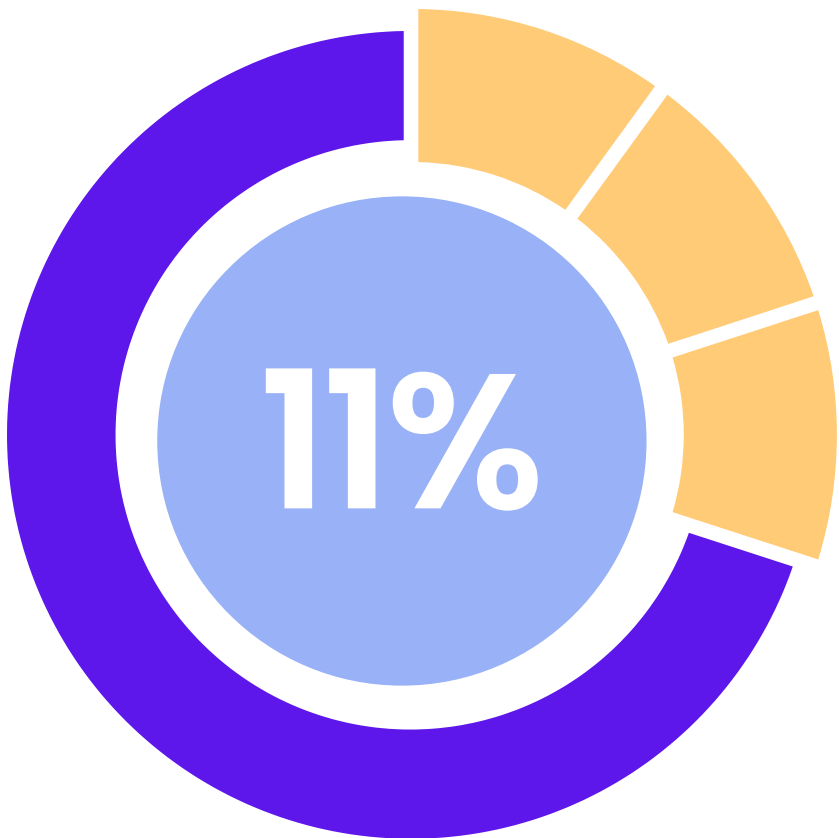
Total Time (s)	25.09
Time in Analyzed Loops (%)	95.3
Time in Analyzed Innermost Loops (%)	95.3
Compilation Option Used	search_mpi_-g: GNU 9.2.0 -mtune=generic -march=x86-64 -g -std=c99
Suggested Compilation Options	search_mpi_-g: -O2



## With optimizations

Application "job\_search\_mpi\_21\_06" Execution Summary

Total Time (s)	22.32
Time in Analyzed Loops (%)	99.8
Time in Analyzed Innermost Loops (%)	99.8
Compilation	search_mpi_optimized: Intel 18.0.3.222 -



# Demonstration



*Thank you*

**Contact Us**

**[nsmss@cdac.in](mailto:nsmss@cdac.in)**

**Feedback**

**[bit.ly/hpcprof](https://bit.ly/hpcprof)**

# Our Team



Senthil Kumar



Shamjith K V



Vineeth Simon



Abhishek Patil



Santhosh J



Arunachalam B