Issue 64

GAṆANAM (गणनम्)

MARCH 2026

HIGH PERFORMANCE COMPUTING NEWSLETTER
INSTITUTE FOR PLASMA RESEARCH, GANDHINAGAR

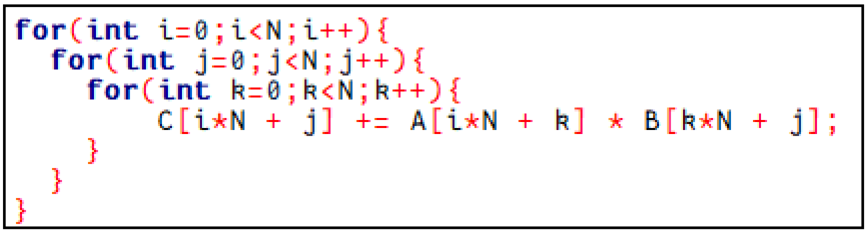
Accelerating Matrix Multiplication with OpenBLAS: A Case Study

Matrix multiplication is one of the most fundamental operations in scientific computing, machine learning, and numerical simulations. However, implementing matrix multiplication using straightforward nested loops leads to inefficient performance when working with large datasets. In High Performance Computing, optimizing such core operations is essential for improving program efficiency. One effective approach is to use optimized mathematical library such as OpenBLAS.

OpenBLAS is an open-source implementation of the BLAS (Basic Linear Algebra Subprograms) standard developed to deliver high performance. It provides optimized routines for common linear algebra operations, including vector and matrix computations. By leveraging CPU-specific optimizations, multi-threading, and efficient memory usage, OpenBLAS can significantly outperform standard loop implementations for matric multiplication.

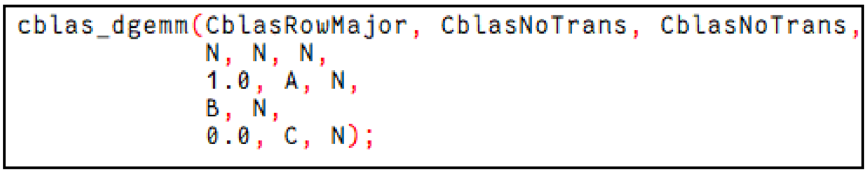
In this case study, matrix multiplication performance is improved by integrating the C BLAS interface provided by OpenBLAS. Instead of computing each element using nested loops, the optimized function cblas_dgemm is used to perform matrix multiplication. This function is designed to efficiently handle large matrices and take advantage of hardware-level optimizations. To use OpenBLAS, a program simply includes the cblas.h header file and links against the OpenBLAS library during compilation. Once integrated, the program can call optimized BLAS routines directly, reducing both code complexity and execution time.

A) Nested Loop Matrix Multiplication



```
for(int i=0;i<N;i++){
  for(int j=0;j<N;j++){
    for(int k=0;k<N;k++){
        C[i*N + j] += A[i*N + k] * B[k*N + j];
    }
  }
}
```

B) Optimised Matrix Multiplication using OpenBLAS



```
cblas_dgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans,
            N, N, N,
            1.0, A, N,
            B, N,
            0.0, C, N);
```

The performance results are illustrated in the comparison graph shown in Figure 1. between the nested-loop implementation and the optimized implementation using OpenBLAS. Since raw experimental measurements often contain fluctuations due to system noise, CPU scheduling, and cache effects, a smoothing technique using a rolling average was applied to the collected timing data. This smoothing process removes sharp spikes and irregular variations while preserving the overall performance trend, making the comparison clearer and easier to interpret. As expected, the execution time of the nested loop implementation increases rapidly with matrix size due to the cubic computational complexity $O(N^3)$ of matrix multiplication.

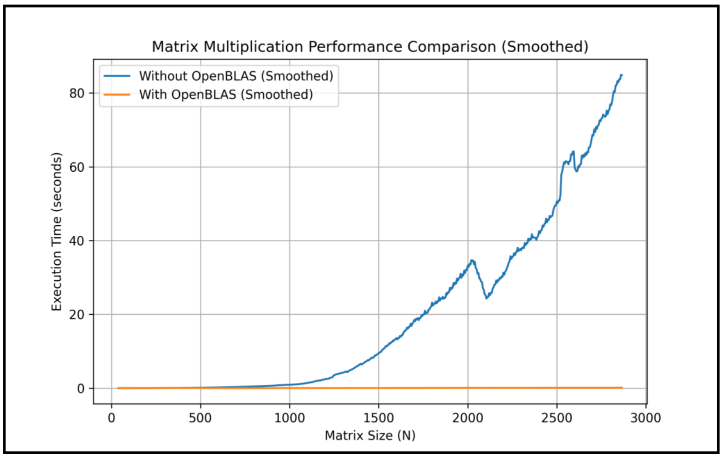


FIGURE 1: Time vs Matrix Size (N) Comparison graph between nested loop implementation and OpenBLAS Implementation

In contrast, the implementation using OpenBLAS remains significantly faster across all matrix sizes. The optimized cblas_dgemm routine utilizes hardware-level optimizations such as cache blocking, SIMD vectorization, and multi-threading. As the matrix size increases, the performance gap becomes increasingly evident, demonstrating a substantial speedup compared to the standard nested-loop approach. These results highlight the effectiveness of optimized numerical libraries in improving computational efficiency in high-performance computing applications.

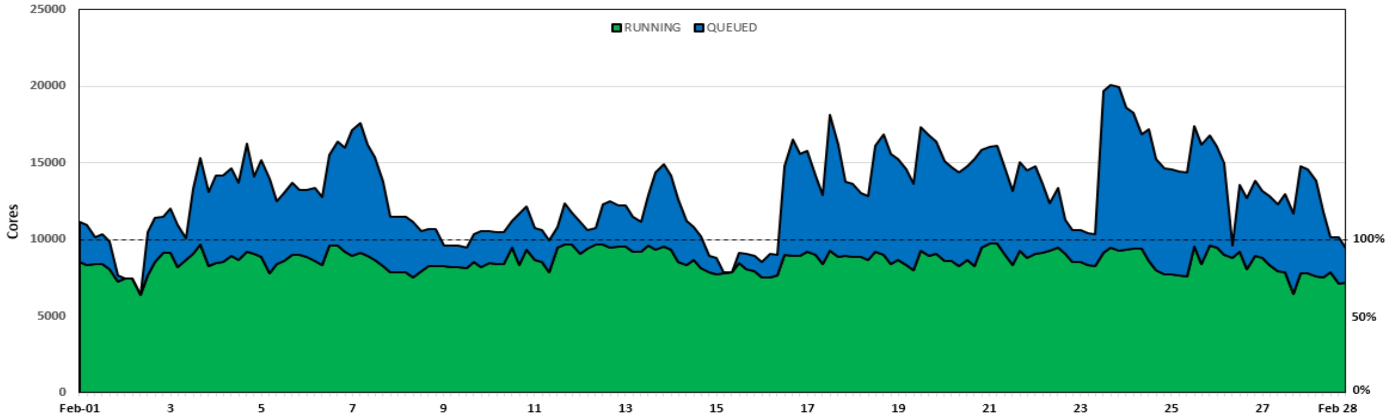
Refereces:

1)OpenBLAS. *OpenBLAS: An Optimized BLAS Library*. Available at: https://github.com/OpenMathLib/OpenBLAS

2)Jack Dongarra, Jeremy Du Croz, Sven Hammarling, and Richard Hanson. *An Extended Set of Fortran Basic Linear Algebra Subprograms.* ACM Transactions on Mathematical Software, 14(1), 1–17, 1988

ANTYA Utilization: FEBRUARY 2026

ANTYA DAILY OBSERVED WORKLOAD



ANTYA HPC Users' Statistics

FEBRUARY 2026

Total Successful Jobs~ 826

> Top Users (Cumulative Resources)

● **CPU Cores**	**Dr. R Ganesh**
● **Walltime**	**Someswar Dutta**
● **Jobs**	**Souvik Mondal**

ANTYA Usage, Updates and News

- **Scheduled Downtime:** There was no downtime of ANTYA for FEBRUARY 2026.
- **Job Submissions:** The highest job loads were observed in the *serialq*, *regularq*, *mediumq*, *longq and ansysq* queues, reflecting sustained user activity across multiple workloads in various queues.
- **Cluster Utilization:** The system maintained an average utilization of ~86% and peak utilisation of ~97%.

Packages/Applications Installed: No new modules have been installed this month. To view list of available modules.

> module avail

Other Recent Work on HPC

DeepCXR: A Hybrid Edge–Cloud AI–IoT Framework for Chest X-Ray Screening in Field Deployments	**Agraj Abhishek**
Effect of pulsed neutral gas injection on heat flux scrape-off layer (SOL) width in ADITYA-U tokamak	**Sk Injamul Hoque**
Prototype implementation of an on-premise Retrieval-Augmented Generation (RAG) based conversational AI system for administrative assistance	**Vijaykumar Keshavlal Patel**
Active control of tokamak plasma turbulence through transient fueling	**Sarveshwar Sharma**
AI-Accelerated Forward Grad–Shafranov Equilibrium Prediction for Tokamak Simulators	**Agraj Abhishek**
Overview of vacuum pumping related technology development and capacity building for Indian Fusion Program	**Samiran Mukherjee**
Collisionless scattering of fast runaway-like electron beams in warm magnetized plasmas	**Devendra Sharma**
High Heat Flux Performance of Curved Tungsten Monoblock with and without Twisted Tape Cooling Inserts	**Rajamannar Swamy Kidambi**
Recent progress on atomic and molecular spectroscopy in Indian fusion research	**UTSAV**

Acknowledgement

The HPC Team, Computer Division IPR, would like to thank all Contributors for the current issue of *GAṆANAM*.

Join the HPC Users Community

hpcusers@ipr.res.in

If you wish to contribute an

GAṆANAM (गणनम्)

Contact us

HPC Team (***hpcteam@ipr.res.in***)
Computer Division, IPR

Disclaimer: "*GAṆANAM*" is IPR's informal HPC Newsletter to disseminate technical HPC related work performed at IPR from time to time. Responsibility for the correctness of the Scientific Contents including the statements and cited resources lies solely with the Contributors.