Simulation and Scientific Computing (SIWIR-1)
Assignment-1

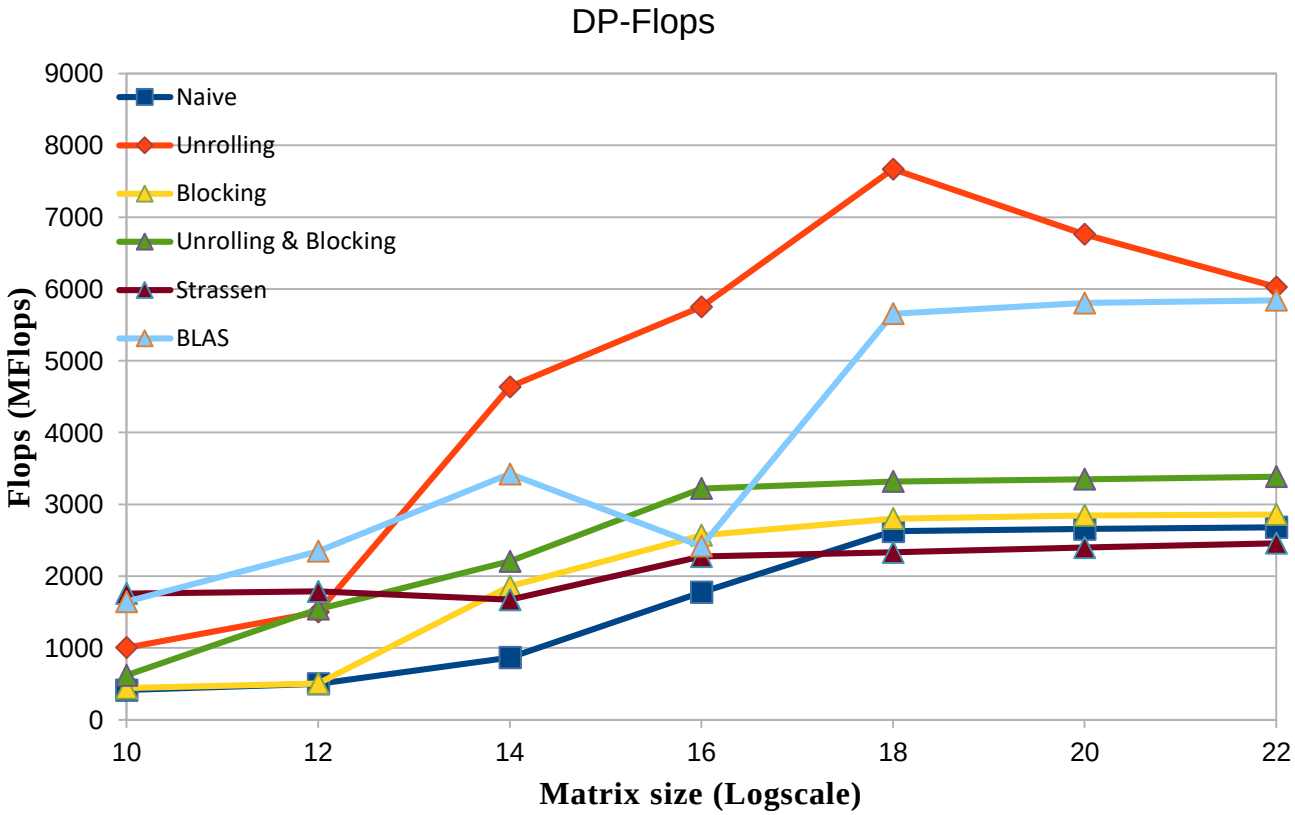
Matrix – Matrix Multiplication Report

- **Team Member**: Gholap,Vinayak (22065422)
  Patel, Mayank (22080480)

- **Description**:
  This document contains performance graphs for various optimization for matrix-matrix multiplication along with the CBLAS implementation. We have plotted graphs by interchanging inner loop for following case.

  1. Naive multiplication without any optimization.
  2. Implementation with Unrolling.
  3. Implementation with Blocking.
  4. Implementation with Unrolling and Blocking.
  5. Strassen implementation.
  6. ATLAS implementation of CBLAS function dgemm.

Double Precision Flops:

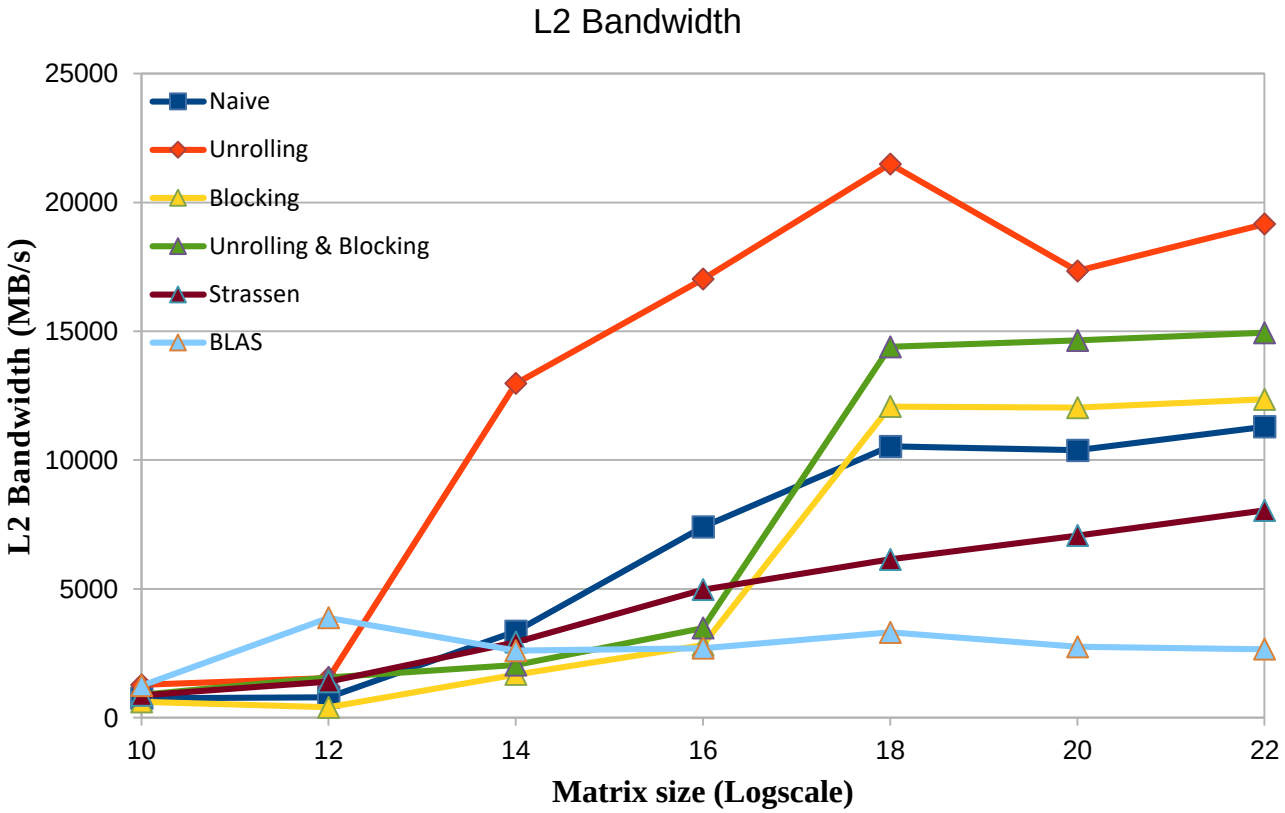


Graph plot of DP FLOPS for all our optimizations and BLAS

Key Observation:

- Only unrolling with loop interchange displays highest flops in large matrices as loop interchange increases register reuse.
- For size of matrix above 64×64, our code becomes memory bound. We try to reduce pressure on bus or latency for transferring data from memory to cache by using various optimization. Therefore, we can see increase in Flops in each of the optimization.
- By combining Unrolling and Blocking, we get higher Flops than each optimization for matrix size above 128×128.

L2 Bandwidth:

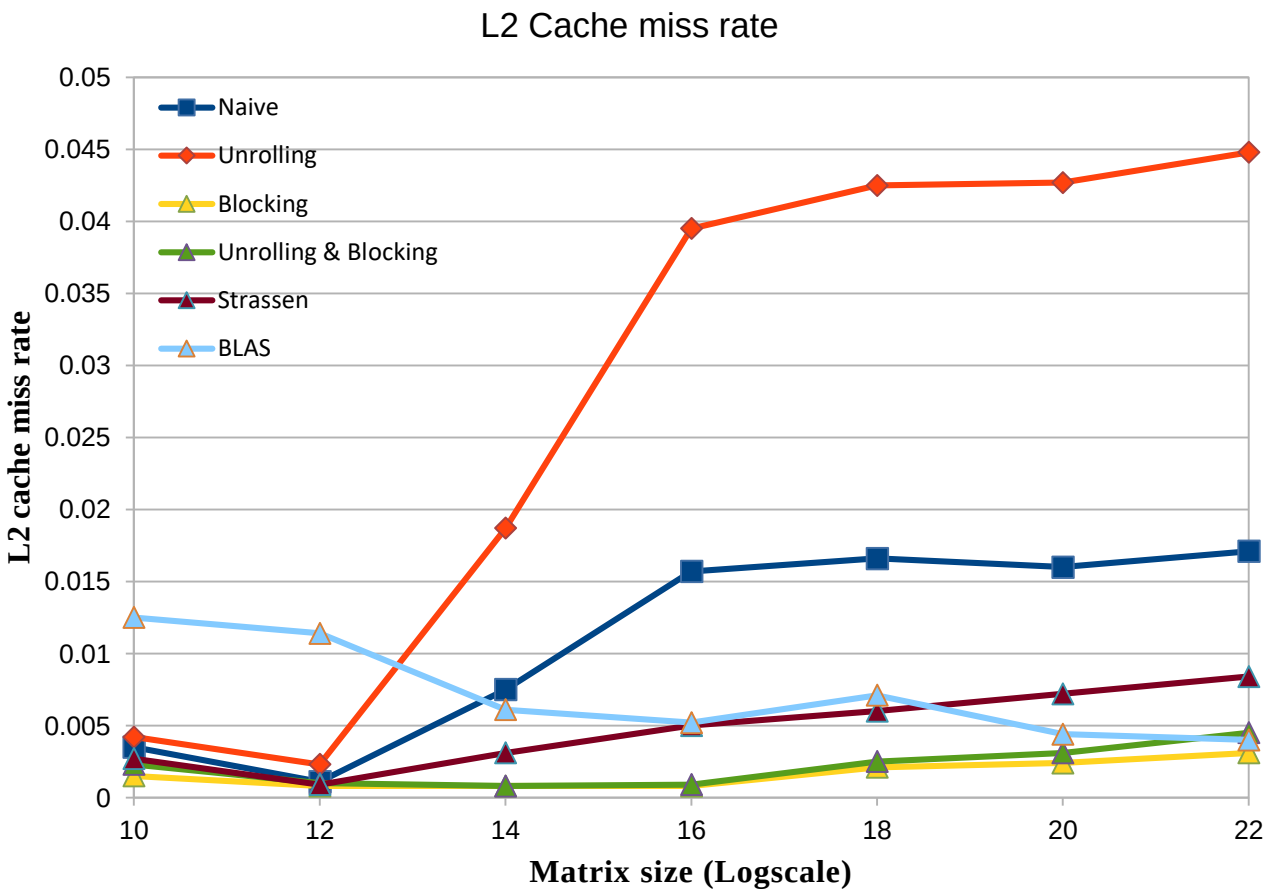


Graph plot of L2 Bandwidth for all our optimizations and BLAS

Key Observation:

- CBLAS has lowest L2 Bandwidth for matrix size above 512×512.
- L2 bandwidth increases linearly in Strassen implementation.
- A sharp increase in L2 bandwidth is observed for Unrolling case as we load whole Cache line and utilize all elements of Cache Line.
- Naive implementation without optimization shows lowest L2 Bandwidth compared to other optimization for matrix size greater than 512×512. Data is needed for computation but it is no longer available.

L2 Cache miss rate:

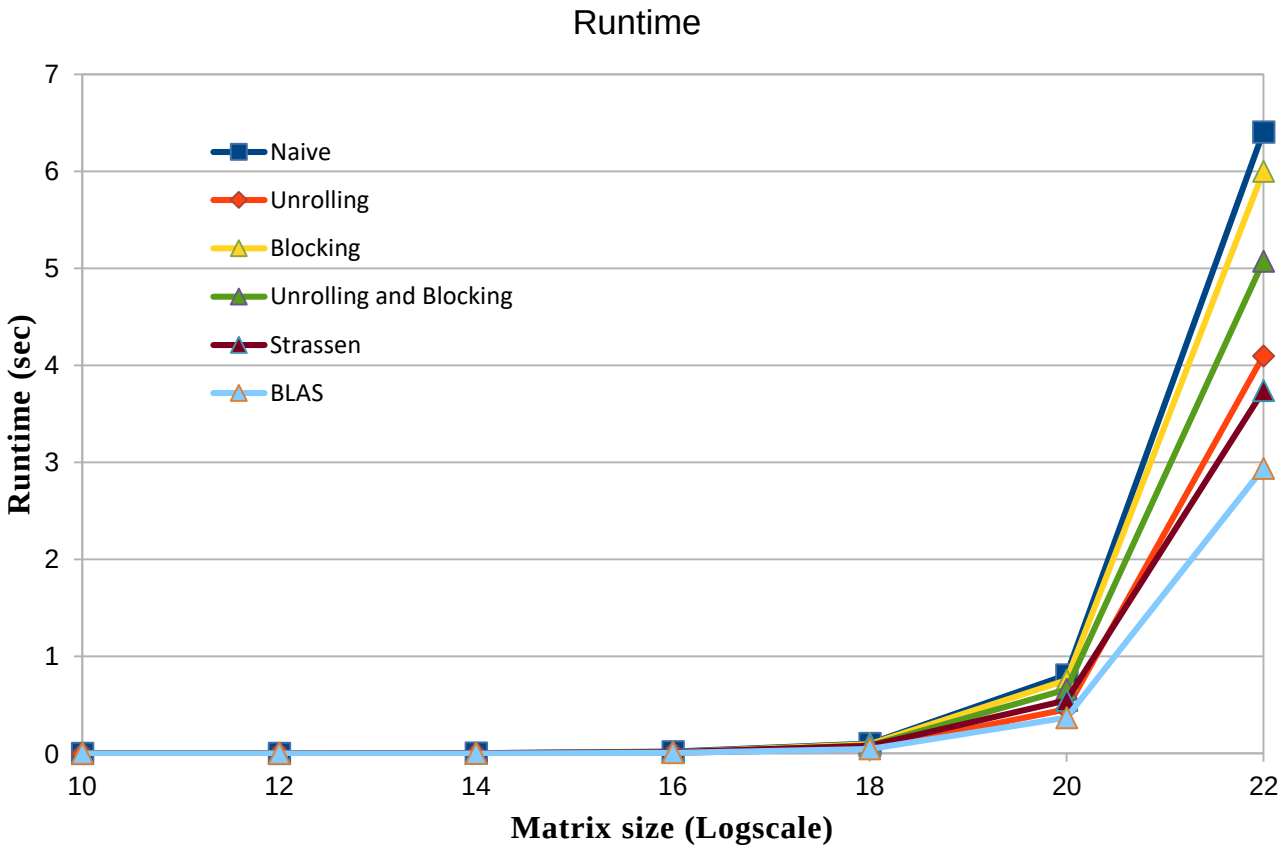


Graph showing the L2 Miss rates of all our optimizations and CBLAS

Key Observation:

- Lowest L2 miss rate is seen in Blocking and Unrolling for all matrix size as all data or cache lines are available in L2 cache.
- L2 miss rate decreases in BLAS for larger matrix size.
- We see that we have highest L2 miss rate for unrolling as Cache line is no longer available in L2 cache for computation. Same trend for Naive implementation.

Runtime:



Graph plotting the runtime of the various optimizations and CBLAS

Key Observation:

- Runtime Increases in all optimization for matrix size above 512×512.
- We can see for 2048×2048 matrix, runtime for BLAS is lowest.
- Strassen gives significant boost in performance for larger matrix.
- Unrolling helps utilize whole cache line. Therefore, its runtime is less than other optimization implementation.
- We can say that for smaller size of matrices, optimization techniques don't play a major role.