Optimized Multigrid solver

Team Name : “**ADG**”

Presented by,
Aravind Raman
Dhineshkumar Balasubramaniam
Ganesh Rajaram

Optimization strategies

Optimization techniques :

- Load / store in 1D vector
- Effective RBGS algorithm

```
    //red
    for (int i=1; i<N-1; i++){
        temp = 1+(i%2);
        for (j=temp; j<N-1; j+=2){
            if ((i*N+j) >= (N*N-1)/2 && (i*N+j) <= ((N*N-1)/2) + (N/2)){
                //leave points

            }else{
            u[i*N + j] = (func [i*N + j] +
                                    left_right * (u [i*N + j - 1] + u [i*N + j + 1]) +
                                          top_bottom * (u [(i - 1)*N + j] + u [(i + 1)*N + j])) / centre;
    }
        }
    }
#pragma omp barrier

#pragma omp for private(temp, j)
    //black
    for (int i = 1; i < N-1; i++) {
        temp = 2 - (i % 2);
            for (j = temp; j < N-1; j += 2) {
                if ((i*N+j) >= (N*N-1)/2 && (i*N+j) <= ((N*N-1)/2) + (N/2)){
                    //leave poits

                }else{
                u[i*N + j] = (func [i*N + j] +
                                    left_right * (u [i*N + j - 1] + u [i*N + j + 1]) +
                                          top_bottom * (u [(i - 1)*N + j] + u [(i + 1)*N + j])) / centre;


                }
            }
    }
```

Parallelization

- OpenMP

```
#pragma omp parallel
      {
#pragma omp for private(variable)
          }
#pragma omp barrier
```

Time to solution

For following case: (emmy cluster)

No. of Levels = **11**

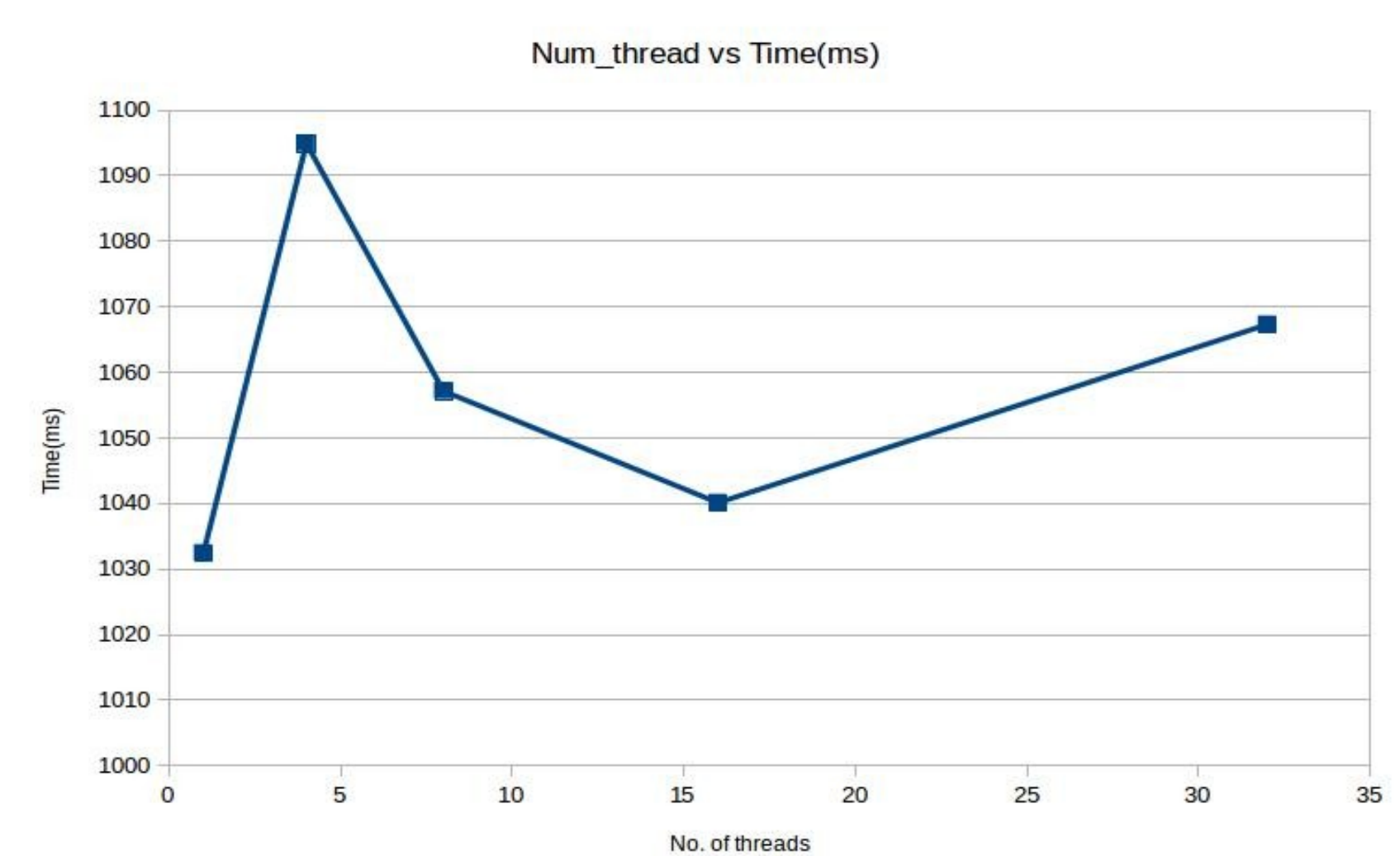
No. of V-cycles = **13**

export SET_NUM_THREADS = **16**

Time to solve the problem:

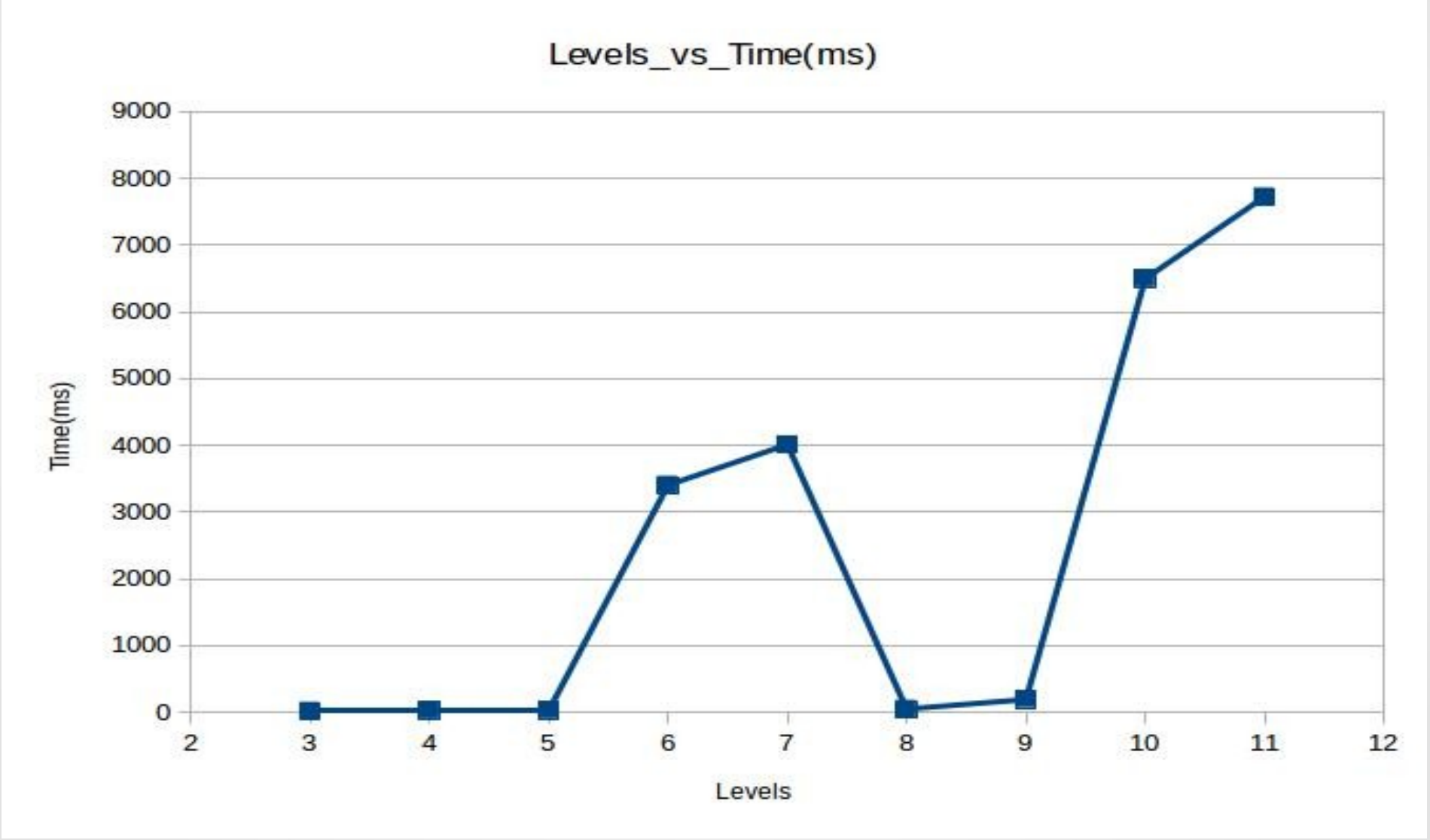
Without OpenMP	: **1.9 s**
With OpenMP	: **0.54 s**

Num thread vs Time (ms)



Levels vs Error norm

Levels vs Time (ms)



Conclusion

- We have successfully implemented the MG solver for the given PDE.
- We have found out the optimum performance of our solver with OpenMP.

Thanks for your attention.