Abstract

Real-world applications require the solution of large a sparse system of algebraic equations that arise from the discretization of partial differential equations with the help of supercomputers. Modern supercomputers are heterogeneous - each node composed of multi-core CPUs and many-core GPUs. Porting existing sequential applications specifically to the GPU architecture has lead to poor utilization of CPU computing power. In this respect, we develop hybrid parallel smoothers for the geometric multigrid method which is one of the most efficient solvers for a system of equations. We study the performance of multigrid method in terms of total execution time by employing different hybrid parallel approaches, viz. accelerating the smoothing operation using only GPU across all multigrid levels, alternately switching between GPU and CPU based on the multigrid level and our proposed novel approach of using combination of GPU and CPU across all multigrid levels. The performance of the hybrid parallel approaches, implemented using MPI, CUDA and OpenMP, is compared against the MPI only approach.

In the first part of the work, we have implemented the hybrid parallel approaches for the Jacobi and Gauss-Seidel smoothers and tested it to solve the system arising from the discretization of the Poisson equation. A coloring strategy is developed to color the degrees of freedom (DOFs) such that the independent set of DOFs are assigned the same color and are updated in parallel on GPU. We adopted two of the commonly used techniques, CSR Scalar and CSR Vector, that perform sparse matrix-vector multiplication on GPU, to implement the smoothing iterations. Further, the strong scaling behavior of the hybrid parallel smoothers is studied across different problem sizes, finite element types, standard and multilevel multigrid.

In the second part of the work, we have implemented the hybrid parallel approaches for Vanka-type smoother which are typically used to solve the saddle-point problem arising from the Navier-Stokes equation. We studied the time taken by the two operations viz. assembling and solving of the local systems in cell and nodal Vanka. We have accelerated the operation of assembling the local system using the hybrid parallel approaches. A similar coloring strategy is developed to assign the same color to independent cells or pressure DOFs. The task of

determining the neighbors of each cell or DOF is offloaded to GPU as it is an $\mathcal{O}(N^2)$ operation. The operation of solving the assembled local systems is parallelized using OpenMP on CPU. Similar to the first part, experiments are performed to study the strong scaling results across different problem sizes, number of OpenMP threads, standard and multilevel multigrid.

The experimental results for the different smoothers show that the scaling performance of the hybrid parallel approaches is bounded by the degree of achievable thread parallelism which in turn is dependent on the parallel workload per process and the algorithm itself. To improve the scaling behavior, we propose a combination approach that uses a workload heuristic to decide the best approach to be applied at each level of multigrid. The combination approach improves the scaling behavior in addition to resulting in a significant speedup under appropriate workload and number of MPI processes compared to the MPI only approach.