Parallel Computing 52 (2016) 1–21

Contents lists available at ScienceDirect

Parallel Computing

journal homepage: www.elsevier.com/locate/parco

Balancing conflicting requirements for grid and particle decomposition in continuum-Lagrangian solvers

Hariswaran Sitaraman*, Ray Grout

Computational Sciences Center, National Renewable Energy Laboratory, 15013 Denver West Pkwy, Golden, CO 80401, United States

ARTICLE INFO

Article history:
Received 12 September 2014
Revised 19 June 2015
Accepted 23 October 2015
Available online 30 October 2015

Keywords:
Load balancing
Lagrangian particle tracking
Particle in cell
Exascale simulations

ABSTRACT

Load balancing strategies for hybrid solvers that involve grid based partial differential equation solution coupled with particle tracking are presented in this paper. A typical Message Passing Interface (MPI) based parallelization of grid based solves are done using a spatial domain decomposition while particle tracking is primarily done using either of the two techniques. One of the techniques is to distribute the particles to MPI ranks to whose grid they belong to while the other is to share the particles equally among all ranks, irrespective of their spatial location. The former technique provides spatial locality for field interpolation but cannot assure load balance in terms of number of particles, which is achieved by the latter. The two techniques are compared for a case of particle tracking in a homogeneous isotropic turbulence box as well as a turbulent jet case. A strong scaling study is performed to more than 32,000 cores, which results in particle densities representative of anticipated exascale machines. The use of alternative implementations of MPI collectives and efficient load equalization strategies are studied to reduce data communication overheads.

1. Introduction

Hybrid continuum-Lagrangian solvers are used in turbulent reacting and multiphase flow simulations [1–4], particle in cell (PIC) methods in plasma simulations [5–8] and in astrophysical simulations [9,10]. These involve calculation of grid based field variables through partial differential equation (PDE) solves and the movement of a large number of particles that represent droplets, charged particles or atoms through the domain using interpolated values of the velocity fields from the underlying computational grid. The update of seed particles using an interpolated vector field (typically velocity) defined on a grid is also used in the computation of stream lines as a post processing step in visualization [11].

The parallel decomposition of the grid based calculations is often the central concern and done primarily via spatial decomposition. This then constrains the particle decomposition to be a compromise, where the common approaches are limits of the locality/load balance trade off. One of the approaches for work sharing of particle calculations is by the spatial partitioning used for the PDE solve. MPI ranks own each of the grid partitions as well as the particles that are present within. Particles are transferred to neighboring ranks when they switch grids across partition boundaries. In many cases, the particle counts are similar to the number of grid points [12,13]. Hence, this method is well suited for the case of uniformly distributed particles in space where the number of particles per grid partition is more or less equal. This situation is rarely seen in particle simulations, but does arise in gyrokinetic simulations of fusion plasmas [14] and early on in some cosmological calculations before particles start clustering

* Corresponding author. fax.: +1 3032753171.
E-mail address: Hariswaran.Sitaraman@nrel.gov, hariswaran@gmail.com (H. Sitaraman).

http://dx.doi.org/10.1016/j.parco.2015.10.010
0167-8191/© 2015 Elsevier B.V. All rights reserved.

in localized regions [15]. The co-locality of field data and particle positions for a given rank makes the process of exchanging data between the grid and particles efficient, because the data need not be transferred between ranks. For the case when the particles are clustered in regions of space, which occurs frequently in plasma PIC, turbulent reacting flow and astrophysical simulations, the spatial particle partitioning technique can be highly load imbalanced. An alternative technique is to partition all of the particles equally among all MPI ranks irrespective of their spatial locations. This achieves load balance in terms of particle movement calculations and field variable interpolation but spatial locality with respect to the grid is lost. Additional communication is required for obtaining grid based field variables from other ranks to perform interpolation.

The contributions presented in this paper are as follows. Firstly, the computational imbalance due to particle update arising from an uneven distribution of particles in a turbulent flow is quantified. Secondly, the cost of the data transfer necessary to restore load balance by distributing the particles independently of the grid is studied. This is done in the context of realistic particle diffusion rates in turbulent flows and a strong scaling study is performed to a maximum of 32,000 compute cores. Strong scaling to large number of compute cores on a state of the art petascale machine, gives rise to particle densities per rank that is representative of current production particle simulations on emerging multi-petaflop machines and anticipated exascale architectures. Thirdly, the overhead from the global communication necessary for bookkeeping and the effectiveness of alternate implementations to reduce this overhead and restore performance is quantified. Finally, a hierarchical load balancing algorithm to preserve grid data locality by generating a balanced particle distribution is outlined. This algorithm also includes flexibility to address many-core nodes by allowing a hierarchical distribution that can preserve locality at several levels.

2. Related work

The spatial domain decomposition based algorithm has been extensively used in production particle-in-cell (PIC) hybrid and molecular dynamics solvers [16–19]. The general concurrent particle in cell (GCPIC) algorithm [5] has been used for particle in cell plasma simulations [20] where the grid partitions are adjusted in such a way that they contain equal number of particles. This algorithm does not ensure equality of grid points for the PDE solve and is used when particle computations are more expensive than grid calculations. The approach where particles are equally shared among all compute cores is seldom used due to the overhead associated with restricted data locality and increased data transfer for cases where particles are scattered throughout the computational domain. The question of whether restricted data locality will be offset with better load balance of particles among all MPI ranks remains unanswered and an efficient paradigm to balance both data locality and particle load balance is studied here. The spatial as well as particle sharing algorithm has been compared by Qiang et al. [21] in the case of particle beam PIC simulations. The particle sharing algorithm exhibited better strong scaling compared to the spatial algorithm for a maximum of 32 MPI ranks. A more extensive comparison study is performed here using 1000s of compute cores, which is relevant in emerging petascale and exascale architectures. In both the techniques discussed above, ranks tend to have only one sided information. For instance in the spatial partitioning method, every rank has the information about the data to be sent regarding the number of particles that are leaving its domain to the receiving MPI ranks. But the receiver is uncertain about the amount of data coming and which rank it is from. In the particle sharing algorithm, every MPI rank has the information about the grid partition its particles belong to and the amount of data it needs to receive from the grid partition. But the rank owning the grid partition is unaware of the grid data it needs to send and to which rank. Therefore, a bookkeeping step is required prior to actual data transfer in both the techniques which is typically achieved through collective communications such MPI Allgather or Alltoall. These collective communication methods return arrays whose length is proportional to the total number of ranks and hence tend to be costly when used with large number of MPI ranks [22,23]. The use of collectives in this scenario also results in the bookkeeping of null information regarding no exchange between two MPI ranks. The use of one sided remote memory access (RMA) communication paradigm in the MPI-2 standard can circumvent these problems and its performance pertaining to particle tracking is studied here. The use of one sided communication has been previously studied for Monte Carlo particle codes [24] and is seen to scale well on multiple compute cores. The use of RMA and Partitioned Global Address Space (PGAS) languages on hardware with support such as the “Gemini” interconnect have been previously studied to reduce communication costs for large number of small messages [25]. A factor of 5–10 times increase in effective data transfer rate for data sizes on the order of 8 B to 2kB has been reported using one sided communication as opposed to two sided message passing.

Dynamic load balancing of PIC simulations using a taskfarm approach have been studied by Othmer et al. [26] where a master MPI rank distributes tasks to other ranks as soon as they are finished with their work. The slave ranks have to wait in a loop to receive tasks which includes transfer of both particle and grid data.

Load balancing strategies for N-body simulations such as in astrophysics have been implemented through Orthogonal Recursive Bisection (ORB) techniques [27,28] which partition the domain recursively along each direction leading to a tree data structure. ORB have been successfully used along with particle algorithms such as fast multipole methods (FMM) and Barnes Hut (BH) algorithm for effective load balancing. N-body solvers involve particle to particle interactions and do not involve particle mesh interactions which is the focus of the current study.

Octrees and space filling curves [29,30] have been used in Adaptive Mesh Refinement (AMR) PDE solvers for load balancing the number of grid points per MPI rank. An octree data structure is created for each level of refinement and a space filling curve is used to traverse the leaves of the forest and partitioned appropriately to balance load. These are single objective load balancing methods which do not involve both particles and particle mesh interactions. They greatly differ from combined particle-grid systems because the two are mutually orthogonal entities and load balancing has to happen over both particle and grid domains with minimal data movement. It might be possible to adapt these approaches for a hybrid particle mesh problem, but this has