An Evaluation of Partitioners for Parallel SAMR Applications*

Sumir Chandra and Manish Parashar

TASSL/ECE Department, Rutgers, The State University of New Jersey
94 Brett Road, Piscataway, NJ 08854 USA
{sumir, parashar}@caip.rutgers.edu

Abstract. This paper presents an experimental evaluation of a suite of dynamic partitioning/load-balancing techniques for adaptive grid hierarchies that underlie parallel structured adaptive mesh refinement applications. Partitioners evaluated include those included in popular software libraries such as GrACE, Vampire, and ParMetis. The overall goal of the research presented in the paper is an application-centric characterization of the partitioners as a function of the number of processors, problem size, and partitioning granularity.

1 Introduction

Dynamic adaptive mesh refinement (AMR) methods for the solution of partial differential equations, that employ locally optimal approximations, can yield highly advantageous ratios for cost/accuracy when compared to methods based upon static uniform approximations. These techniques seek to improve the accuracy of the solution by dynamically refining the computational grid in regions of high local solution error. Distributed implementations of these methods lead to interesting challenges in dynamic resource allocation, data-distribution and load balancing, communications and coordination, and resource management. Critical among these is the partitioning of the adaptive grid hierarchy to balance load, optimize communications and synchronizations, minimize data migration costs, and maximize grid quality (aspect ratio) and available parallelism.

This paper presents an experimental evaluation of a suite of dynamic domain-based partitioning and load-balancing techniques for distributed adaptive grid hierarchies that underlie parallel structured adaptive mesh refinement (SAMR) methods for the solution to partial differential equations. The partitioners evaluated include existing (ISP and G-MISP) as well as new (G-MISP+SP, pBD-ISP, and SP-ISP) techniques, and constitute a selection from popular software libraries, viz. GrACE [2], Vampire [3], and ParMetis [1]. A 3-D compressible turbulence application kernel solving the Richtmyer-Meshkov instability is used as the driving application in the evaluation. The primary motivation for the

* Research supported by NSF via grants WU-HT-99-19 P029786F (KDI) and ACI 9984357 (CAREERS) awarded to Manish Parashar. The authors thank Johan Steensland and Ravi Samtaney for providing access to Vampire and RM3D.

Table 1. Summary of evaluated partitioning techniques

Scheme	Suite	Description
SFC (ISP)	GrACE	Decomposition of the recursive linear representation of multi-dimensional hierarchy, generated using space-filling mappings
G-MISP	Vampire	Successive refinement of one-vertex graph (of workloads) by splitting vertices with total weight greater than a threshold
G-MISP+SP	Vampire	Similar to G-MISP scheme but uses sequence partitioning to assign internal blocks to processors
pBD-ISP	Vampire	Binary dissection of domain into p partitions with orientation of the cuts determined by the Hilbert space-filling curve
SP-ISP	Vampire	Dual-level parameterized binary dissection, with SFC-ordered one dimensional list partitioned using sequence partitioning
WD	ParMetis	Global workload algorithm for unstructured grids employing multi-level k-way graph partitioning

research presented in this paper is the observation that *even for a single application, the most suitable partitioning technique depends on input parameters and its run-time state* [4]. The goal of this work is an application-centric characterization of the partitioners as a function of the number of processors, problem size, and partitioning granularity. Such a characterization will enable the run-time selection of partitioners based on the input parameters and application state.

2 Experimental Evaluation of Partitioning Techniques

This paper evaluates a suite of six domain-based partitioning techniques [4], namely, space-filling curve based partitioning (SFC), geometric multi-level inverse space-filling curve partitioning (G-MISP), geometric multi-level inverse space-filling curve partitioning with sequence partitioning (G-MISP+SP), p-way binary dissection inverse space-filling curve partitioning (pBD-ISP), "pure" sequence partitioning with inverse space-filling curves (SP-ISP), and wavefront diffusion (WD) based on global work load. Table 1 summarizes these techniques.

The partitioners described above were evaluated using a 3-D "real-world" compressible turbulence application solving the Richtmyer-Meshkov (RM) instability. The RM3D application is part of the virtual test facility developed at the ASCI/ASAP center at the California Institute of Technology[1]. The Richtmyer-Meshkov instability is a fingering instability which occurs at a material interface accelerated by a shock wave. The experiments were performed on the NPACI IBM SP2, *Blue Horizon*, at the San Diego Supercomputing Center. Blue Horizon is a teraflop-scale Power3 based clustered SMP system from IBM with 1152 processors and 512 GB of main memory and AIX operating system.

The experiments used a base (coarse) grid of 128x32x32 with 3 levels of factor 2 space-time refinements with dynamic regridding and redistribution at regular intervals. The application ran for 150 coarse level time-steps in each case. The

[1] http://www.cacr.caltech.edu/ASAP.

experiments consisted of varying the partitioner used, the number of processors (16 - 128), and the partitioning granularity (2x2x2 - 8x8x8). The metrics used for evaluation include the total run-time, maximum load imbalance, and the corresponding AMR efficiency. AMR efficiency is the measure of effectiveness of AMR and affects partitioning and load balancing requirements. High AMR efficiency leads to finer granularity refinements. The experimental results for the three metrics are summarized in Fig. 1 and tabulated for 16 processors with granularity 2 in Table 2. Note that the absence of a bar for a partitioner in the graph indicates that the partitioner was not suitable for that combination.

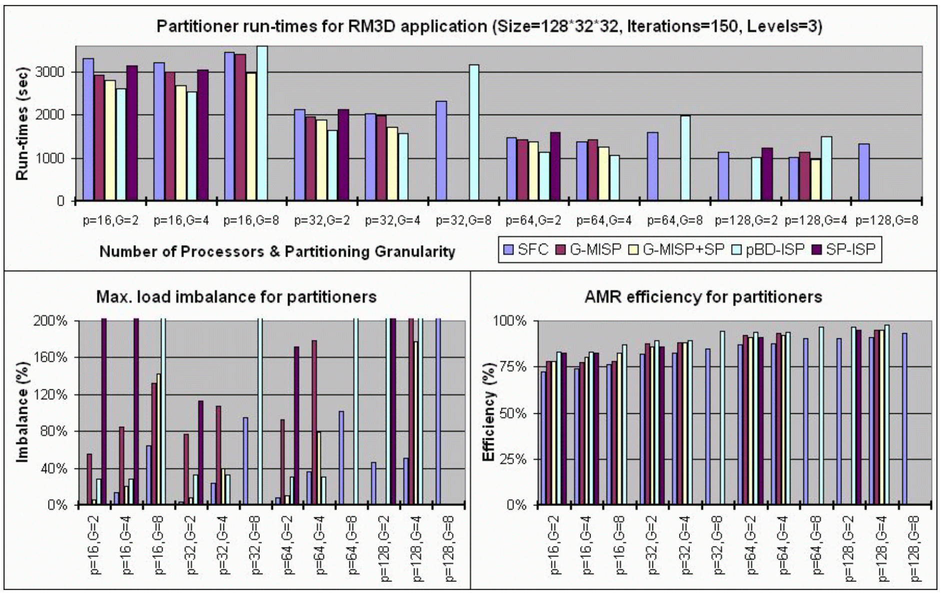

Fig. 1. Performance of partitioning techniques for RM3D application

The RM3D application required rapid refinement and efficient redistribution due to the shock wave introduced. The pBD-ISP, G-MISP+SP, and SFC partitioning schemes are best suited to the RM3D application as they are high-speed partitioners that attempt to distribute the workload as evenly as possible while maintaining good communication patterns. The pBD-ISP scheme is the fastest partitioner but generates average load balance which worsens with higher granularity. G-MISP+SP and SFC techniques yield excellent load balance but are relatively slower. The G-MISP scheme favors speed over load balance and has an average overall performance. The SP-ISP technique fares poorly due to partitioning overheads and high computational costs, resulting in higher partitioning time and poor load balance. All evaluated partitioning techniques scale reasonably well. The optimal partitioning granularity for an application may require a trade-off between the execution speed and the load imbalance. In the case of

RM3D application, a granularity of 4 gives the lowest execution time and yields acceptable load imbalance.

Table 2. Partitioner performance for RM3D on 16 processors with granularity 2

Partitioner	Run-time(s)	Max. Load Imbalance(%)	AMR Efficiency(%)
SFC	3315.22	1.629	72.388
G-MISP	2931.08	55.431	77.745
G-MISP+SP	2805.54	5.834	77.851
pBD-ISP	2601.05	28.498	83.169
SP-ISP	3136.32	204.548	82.207

SP-ISP, G-MISP and G-MISP+SP partitioning schemes fail for experiments on large number of processors using a higher granularity. SP-ISP scheme fails due to the large number of blocks created. G-MISP and its variant G-MISP+SP fail due to the effects of high granularity on the underlying partitioning mechanism. Finally, our ParMetis integration proved to be computationally expensive due to the additional effort required for adapting it to SAMR grid hierarchies. As a result, it could not compete with dedicated SAMR partitioners for the RM3D application and is not part of the results.

3 Conclusions

This paper presented an application-centric experimental evaluation of a suite of dynamic domain-based partitioning and load-balancing techniques for the SAMR RM3D compressible turbulence application. Each of the partitioners evaluated represented a unique compromise between partitioning speed and partitioning/load-balancing quality. The experiments focussed on the effect of the choice of partitioner and partitioning granularity on execution time. Metric included the overall run-time, load-balance achieved, and AMR efficiency. It was observed that partitioners with high partitioning speed and average-to-good load balance performed better for the RM3D application. The pBD-ISP and G-MISP+SP schemes generated acceptable load balance with high partitioning speed and high AMR efficiency.

References

1. Karypis, G., Schloegel, K., Kumar, V.: *ParMetis - Parallel Graph Partitioning and Sparse Matrix Ordering Library, ver. 2.0.* University of Minnesota, 1998.
2. Parashar, M., et al: *A Common Data Management Infrastructure for Adaptive Algorithms for PDE Solutions.* Technical Paper at Supercomputing, 1997.
3. Steensland, J.: *http://www.caip.rutgers.edu/˜johans/vampire.* Vampire, 2000.
4. Steensland, J., Chandra, S., Thuné, M., Parashar, M.: *Characterization of Domain-Based Partitioners for Parallel SAMR Applications.* IASTED International Conference on Parallel and Distributed Computing and Systems, 2000.