

A PARALLEL APPROACH OF INTEREST MANAGEMENT IN EXASCALE SIMULATION SYSTEMS

Elvis S. Liu

University College Dublin and IBM Research, Ireland

elvisliu@ie.ibm.com

ABSTRACT

Interest management in parallel/distributed simulation is a filtering technique which is designed to reduce bandwidth consumption of data communication and therefore enhances the scalability of the system. This technique usually involves a process called “interest matching”, which determines what data should be sent to the participants as well as what data should be filtered. However, existing interest matching algorithms are mainly designed for serial processing which is supposed to be run on a single processor. As the problem size grows, these algorithms may not be scalable since the single processor may eventually become a bottleneck. In this research, a parallel approach of interest matching is developed, which is suitable to apply on exascale parallel/distributed simulation systems.

Keywords: Simulations, Gaming, Systems

1. INTRODUCTION

In the coming years we expect to reach a computational power equivalent to a thousandfold that of the current most powerful supercomputer (exascale systems). Computational advances have opened the way for a growing number of computer simulation applications across many fields. As the scale grows, providing scalable data distribution through interest management becomes one of the major design requirements of large-scale simulation systems. The basic idea of interest management is simple: all participants should only receive data that are of interest to them. This data filtering process, however, may introduce considerable computational overhead. If the cost of interest management is too high, it would degrade the overall performance of the simulation. Over the years, numerous interest management schemes have been proposed which sought to reduce the computational overhead and, at the same time, to maintain the high precision of data filtering. These schemes, however, are designed

for serial processing which is supposed to be run on a single processor. As the problem size grows, these algorithms may not be scalable since the single processor may eventually become a bottleneck. Furthermore, large-scale simulations are executed on parallel/distributed high-performance computing systems, deploying the existing schemes on these systems would be unsuitable, and the performance cannot be guaranteed.

In this research, a parallel approach of interest matching algorithm is being developed, which is suitable for deploying on exascale parallel/distributed simulation systems. The new algorithm enables the multiple processors to work simultaneously and thus enhances the overall runtime efficiency of the matching process.

2. BACKGROUND AND RELATED WORK

Aura-based interest management (e.g., (Greenhalgh and Benford 1995)) use auras to represent the interests of each participant. When auras overlap, a connection between the owners of the auras is established and messages are exchanged through the connection. This approach provides a much more precise message filtering mechanism than the zone-based approaches (e.g., (Macedonia, Zyda, Pratt, Brutzman, and Barham 1995)); however, more computational effort is required for testing the overlap status for the auras. DIVE (Carlsson and Hagsand 1993) and MASSIVE (Greenhalgh and Benford 1995) are the early DVE systems that adopt this type of schemes. In the High-Level Architecture (HLA) (DMSO 1998), the Data Distribution Management (DDM) services allow the participants to specify “update regions” and “subscription regions” to represent their interests. These regions are similar to the auras except they must be rectangular and axis-aligned when using in a two- or three-dimensional space.

The interest matching algorithms are designed to solve the “trade-off” between runtime efficiency and

filtering precision for aura-based interest management. They have usually been applied on high precision filtering schemes, such as HLA DDM, which ensures the participants receive the minimal set of data that are of interest to them. In addition, they provide a way to efficiently reduce the computational overhead of the matching process.

In an early paper (Van Hook, Rak, and Calvin 1994), Van Hook et al. pointed out that the matching process of the aura-based approach (referred to as “object-based approach” in the paper) could be computationally intensive. To solve this problem, Van Hook et al. proposed a crude grid-based filtering approach to cull out many irrelevant entities before a more compute-intensive procedure is carried out for finer discrimination. (Morgan, Storey, and Lu 2004) proposed a collision detection algorithm for aura-based interest matching. The algorithm uses aura overlap for determining spatial subdivision. The authors argued that it more accurately reflects the groupings of entities that may be interacting than existing collision detection algorithms and provided performance figures to demonstrate its scalability. Recently, more robust matching algorithms (Raczy, Tan, and Yu 2005; Liu, Yip, and Yu 2005; Pan, Turner, Cai, and Li 2007) based on dimension reduction were proposed. These algorithms are designed specifically for HLA-compliant systems, and thus adopt the use of rectangular auras (i.e., regions of the HLA). The basic idea of dimension reduction is to reduce the multidimensional overlap test to a one-dimensional problem, which is more computationally efficient than the original problem.

3. PARALLEL INTEREST MATCHING

This section describes a parallel interest matching algorithm which facilitates parallelism by distributing the workload of the matching process across shared-memory multiprocessors. The algorithm divides the matching process into two phases. In the first phase it employs a spatial data structure called uniform subdivision to efficiently decompose the virtual space into a number of subdivisions. We define as work unit (WU) the interest matching process within a space subdivision. In the second phase, WUs are distributed across different processors and can be processed concurrently.

For the sake of consistency, aura is hereafter referred to as “regions” as per the terminology of HLA DDM.

3.1. First Phase: Hashing

Uniform subdivision is a common spatial data structure which has long been used as a mean of rapid retrieval of geometric information. Over the years, it has been studied extensively in many fields such as computer graphics and robotics. The idea of using hashing

for subdivision directory was first described in an early article written by Rabin (Rabin 1976) and was later discussed more generally in Bentley and Friedman’s survey (Bentley and Friedman 1979).

During the simulation, regions are hashed into the hash table. The algorithm uses the coordinate of a region’s vertex as a hash key. Given a key k , a hash value $H(k)$ is computed, where $H()$ is the hash function. The hash value is an n -dimensional index which can be matched with the index of a space subdivision, and therefore indicating that which subdivision the vertex lies in. Hence, the regions with hash key k are stored in slot $H(k)$. The hash function is given in

Definition 1.

Definition 1. Let $[SMIN_d, SMAX_d)$ be the boundary of a space in d dimension, for $d = 1, 2, \dots, n$. The boundary is uniformly divided into N_d sub-boundaries with unit length L_d . The hash function for transforming a key k_d into a hash value is defined as

$$H : \mathbb{R}^n \rightarrow \mathbb{Z}^n, H(k_d) = \lfloor \frac{k_d - SMIN_d}{L_d} \rfloor$$

There are two important properties of using a hash table for spatial decomposition. First, hash table collision means that regions in the same slot are potentially overlapped with each other; therefore, further investigation on their overlap status is required. This process will be left to the second phase of the algorithm. Second, if a region lies in multiple space subdivisions, it would be hashed into all of them. The algorithm assumes that the size of region is much smaller than a space subdivision. Therefore, a region would exist in at most four slots in the two-dimensional space (at most eight slots in the three-dimensional space). This assumption ensures that the computational complexity of the hashing process would be bounded by a constant.

Figure 1 illustrates the basic concept of the spatial hashing for two-dimensional space. In the figure, region A is hashed into slot (0,1); region B is hashed into slots (0,0), (0,1), (1,0) and (1,1); region C is hashed into slots (1,1) and (1,2); region D is hashed into (1,0), (1,1), (2,0) and (2,1). Note that if not all vertices of a region are hashed into the same slot, then the region exists in multiple subdivisions.

The hash table is constructed at the initialisation stage. During runtime, the position and size of regions may be frequently modified. Therefore, the algorithm needs to perform rehashing for the regions at every time-step. The complexity of this process is $O(n + m)$ where m is the number of subscription regions and n is the number of update regions.

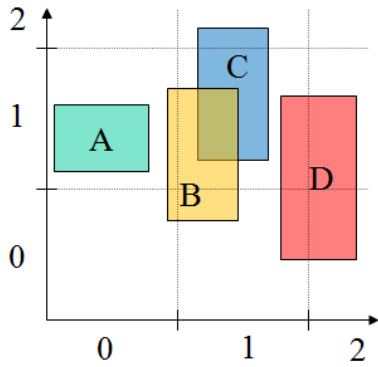


Figure 1: Hashing for Space Subdivisions

3.2. Second Phase: Sorting

After the hashing stage, each slot of the hash table represents a WU which will be distributed across different processors. The algorithm then places the WUs on a task queue. Each processor fetches WUs from the queue and performs interest matching for the corresponding space subdivisions. Since only one processor has the authority to manage each space subdivision, there will be no ambiguous matching result.

The spatial decomposition approach essentially transforms the large-scale interest matching process into several individual sub-problems. When a WU is being processed, each processor carries out a matching process only for the regions within the WU. Since using a brute-force approach to determine the overlapping status of the regions would be time consuming, a sorting algorithm based on dimension reduction would help to increase the computational efficiency. The preliminary design of dimension reduction is presented in (Liu, Yip, and Yu 2005). It reduces the multidimensional overlap test to a one-dimensional problem, which is defined as follows:

Two regions overlap in n -dimensional space if and only if their orthogonal projections¹ on the 1st, 2nd, ..., and n^{th} dimension overlap.

Figure 2 shows how the concept of dimension reduction works in two-dimensional space. In the figure, B-C overlap on x-axis; A-C, A-B, B-C, B-D, and C-D overlap on y-axis; hence, B-C overlap on two-dimensional space.

As discussed in (Dandamudi and Cheng 1995), the task queue approach is desired for task distribution and provides very good load sharing for shared-memory multiprocessor systems. When a processor

¹In the terminology of HLA, the orthogonal projection of a region is called “extent”.

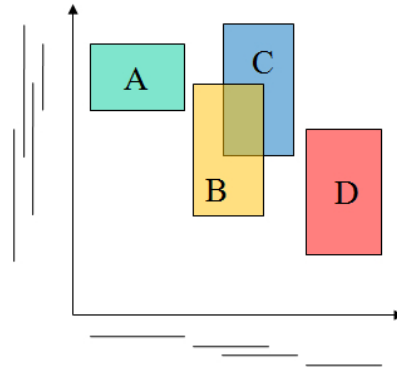


Figure 2: Dimension Reduction

finishes processing a WU, it would fetch another WU from the task queue immediately unless the queue is empty. Therefore, no processor would be idle until all WUs are fetched. The worst case happens only when all regions reside in a single space subdivision. In this situation, a single processor would be responsible for the matching of all of them.

4. SUMMARY AND FUTURE WORK

Over the last few years, several matching algorithms were proposed to speed up the interest matching process. However, these efforts have focused on sequential algorithms. As the problem size grows, using these algorithms does not satisfy the scalability requirement of DVE since the single processor may eventually become a bottleneck. An approach to alleviate this problem is to exploit the inherent parallelism of the matching process and the high availability of parallel computation infrastructures. In this research, a parallel matching algorithm for exascale simulation systems is developed. The proposed algorithm can be run on a cluster of computers that enables them to work simultaneously and thus enhances the overall runtime efficiency of the matching process.

The future work will concentrate on performing experimental comparisons of the runtime efficiency of the proposed algorithm and the parallel algorithm presented in (Liu and Theodoropoulos 2009). We will also test and compare the performance the proposed algorithm under different entity behaviors, number of nodes, and occupation density.

ACKNOWLEDGEMENT

This work was supported, in part, by Lero - the Irish Software Engineering Research Centre (www.lero.ie).

REFERENCES

- Bentley, J. L. and Friedman, J. H., 1979 . Data Structures for Range Searching. *ACM Computing Survey*, 11:397–409.
- Carlsson, C. and Hagsand, O. (1993). DIVE - A platform for multi-user virtual environments. *Computers & Graphics*, 17:663–669.
- Dandamudi, S. P. and Cheng, P. S. P. (1995). A hierarchical task queue organization for shared-memory multiprocessor systems. *IEEE Trans. Parallel Distrib. Syst.*, 6(1):1–16.
- DMSO (1998). High Level Architecture Interface Specification Version 1.3.
- Greenhalgh, C. and Benford, S. (1995). Massive: a collaborative virtual environment for teleconferencing. *ACM Transactions on Computer Human Interactions*, 2(3):239–261.
- Liu, E. S. and Theodoropoulos, G. K. (2009). An Approach for Parallel Interest Matching in Distributed Virtual Environments. In *Proceedings of the 13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2009)*, Singapore.
- Liu, E. S., Yip, M. K., and Yu, G. (2005). Scalable Interest Management for Multidimensional Routing Space. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST) 2005*, pages 82–85.
- Macedonia, M. R., Zyda, M. J., Pratt, D. R., Brutzman, D. P., and Barham, P. T. (1995). Exploiting Reality with Multicast Groups: A Network Architecture for Large-scale Virtual Environments. In *Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS) '95*, pages 2–10.
- Morgan, G., Storey, K., and Lu, F. (2004). Expanding Spheres: A Collision Detection Algorithm for Interest Management in Networked Games. In *Proceedings of the Entertainment Computing - ICEC 2004*.
- Pan, K., Turner, S. J., Cai, W., and Li, Z. (2007). An Efficient Sort-Based DDM Matching Algorithm for HLA Applications with a Large Spatial Environment. In *Proceedings of the 21st International Workshop on Principles of Advanced and Distributed Simulation (PADS 2007)*, pages 70–82, Washington, DC, USA.
- Rabin, M., 1976 . Probabilistic algorithms. In *Algorithms and Complexity: New Direction and Recent Results*, pages 21–39. Academic Press Inc, New York.
- Raczy, C., Tan, G., and Yu, J. (2005). A sort-based DDM matching algorithm for HLA. *ACM Transactions on Modeling and Computer Simulation*, 15(1):14–38.
- Van Hook, D. J., Rak, S. J., and Calvin, J. O. (1994). Approaches to relevance filtering. In *Proceedings of the Eleventh Workshop on Standards for the Interoperability of Distributed Simulations*, pages 26–30.

AUTHOR BIOGRAPHY

Dr. Elvis S. Liu is a Research Fellow of Irish Research Council and IBM Research, Ireland. His current research is focused on simulation technologies for Exascale high-performance computing systems. He received a B.Sc. from the University of Hong Kong, an M.Sc. from the University of Newcastle (UK), and a Ph.D. from the University of Birmingham (UK). His email address is <elvisliu@ie.ibm.com>.