Hybrid MPI-OpenMP Paradigm on SMP clusters: MPEG-2 Encoder and n-body Simulation

Truong Vinh Truong Duy, Katsuhiro Yamazaki, Kosai Ikegami, and Shigeru Oyanagi
Graduate School of Science and Engineering, Ritsumeikan University
duy@hpc.cs.ritsumei.ac.jp, yamazaki@se.ritsumei.ac.jp, ikegami@hpc.cs.ritsumei.ac.jp, oyanagi@cs.ritsumei.ac.jp

***Abstract*—Combining both shared memory and message passing parallelization within the same application, the hybrid MPI-OpenMP paradigm is an emerging trend for parallel programming to fully exploit distributed shared-memory architecture. In this paper, we improve the performance of MPEG-2 encoder and n-body simulation by employing the hybrid MPI-OpenMP programming paradigm on SMP clusters. The hierarchical image data structure of MPEG bit-stream is eminently suitable for the hybrid model to achieve multiple levels of parallelism: MPI for parallelism at the group of pictures level across SMP nodes and OpenMP for parallelism within pictures at the slice level within each SMP node. Similarly, the work load of the force calculation which accounts for upwards of 90% of the cycles in typical computations in n-body simulation is shared among OpenMP threads after ORB domain decomposition among MPI processes. With n-body simulation, experimental results demonstrate that the hybrid MPI-OpenMP program outperforms the corresponding pure MPI program by average factors of 1.52 on 4-way cluster and 1.21 on 2-way cluster. Likewise, the hybrid model offers a significant performance improvement of 18% compared to MPI model for the MPEG-2 encoder.**

Keywords-Hybrid Parallel Programming; MPI; MPEG-2; n-body; OpenMP

I. Introduction

Large scale highly parallel systems based on cluster of SMP architecture are today's dominant computing platforms which support a wide diversity of parallel programming paradigms. A combination of shared memory and message passing paradigms within the same application, hybrid programming, is expected to provide a more efficient parallelization strategy for clusters of SMP nodes. The hybrid paradigm outweighs other paradigms in terms of parallelism and communication costs. This paper describes the performance improvement of MPEG-2 encoder and n-body simulation by employing the hybrid MPI-OpenMP programming paradigm.

II. MPEG-2 Encoding Application

A. MPEG Overview

MPEG is an encoding and compression system for digital multimedia content. The hierarchy of layers in an MPEG bit-stream is arranged in the following order: Sequence, Group of Pictures (GOP), Picture, Slice, Macro-block, and Block (Figure 1), which is suitable for applying the hybrid paradigm.

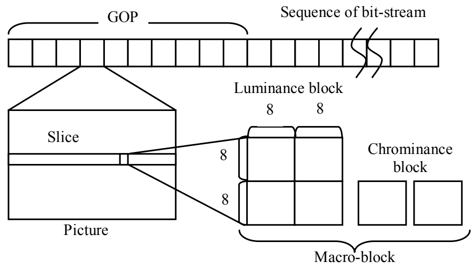

Fig. 1. The hierarchy of layers in an MPEG bit-stream.

B. Exploiting Two Levels of Parallelism

1) Parallelism at GOP level: In the block partitioning, GOPs are divided into as many relatively equal blocks as MPI processes, and each carries out one block. In the cyclic partitioning, GOPs are distributed to MPI processes in a fashion similar to round-robin: the first process gets the first GOP, the second process gets the second GOP, and so on, until no more GOPs remain.

2) Parallelism at slice level: Parallelism at slice level is achieved by using multiple OpenMP threads running inside MPI to process slices within a picture. The coder forms a 'prediction error' picture that represents the difference between the predicted macro-block and the macro-block being encoded. The prediction error is transformed with the DCT (Discrete Cosine Transform), quantized to reduce the number of bits and formed into a coded bit-stream out. Then it is inversely quantized and IDCT transformed for predicting subsequent pictures. DCT, Quantization, Inverse Quantization and IDCT are executed in parallel using multiple threads in Open MP (Figure 2).

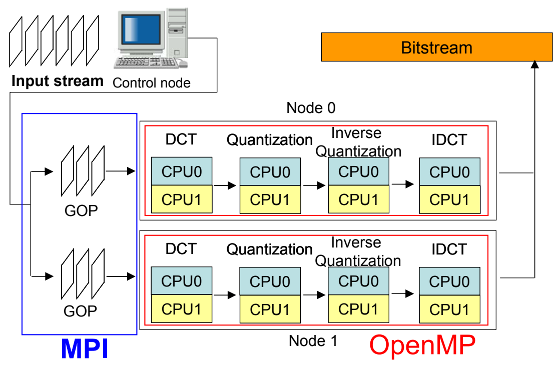

Fig. 2. Applying two levels of parallelism to the input bit-stream.

III. THE N-BODY SIMULATION

A. Tree Code Algorithm

The n-body problem involves advancing the trajectories of n bodies according to their time evolving mutual gravitation field. The essence of the tree code is the recognition that a distant group of bodies can be well-approximated by a single body, located at the center of mass with a mass equal to total mass of the group. It represents the distribution of the bodies in quad-tree for 2D space or oct-tree for 3D space. After the tree construction phase, the force on a body in the system can be evaluated by traversing down the tree from root.

B. Parallelization of Tree Code

Multiple levels of parallelism are achieved with the hybrid program implementing the parallel treecode (Figure 3). For the first level, the bodies are distributed in a balanced way among the MPI processes using Orthogonal Recursive Bisection (ORB) domain decomposition. Then each process constructs the tree and walks through it to build a list of interaction nodes for each body to calculate the force. For the second level, the force calculation which accounts for upwards of 90% of the cycles in typical computations is parallelized with OpenMP work-sharing threads running in each process.

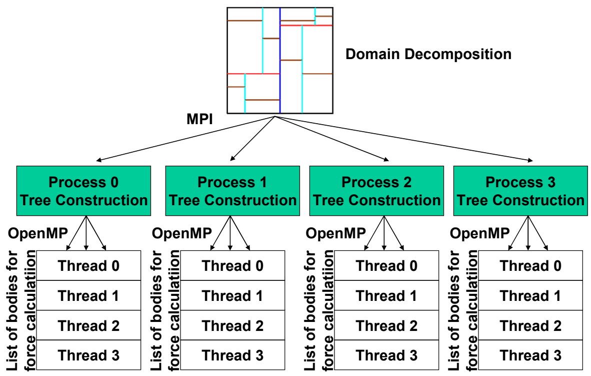

Fig. 3. Multiple levels of parallelism with the hybrid program.

IV. EXPERIMENTAL RESULTS

A. Compute Platforms

System specifications of two SMP clusters used for evaluating the parallel programs are detailed in Table I.

TABLE I
SYSTEM SPECIFICATIONS

Name	SMP Node	# of Nodes	# of CPUs	Network
Diplo	Quad Xeon 3GHz	4	16	Gigabit Ethernet
Atlantis	Dual Xeon 2.8GHz	16	32	Gigabit Ethernet

B. The MPEG-2 Encoder

The performance of MPEG-2 encoder for the sample stream consisting of 1920 frames with a length of 64 seconds in the Main Profile, High1440 Level on the 2-way Atlantis cluster is displayed in Figure 4. Clearly, the hybrid programs are better than the pure MPI programs whatever processors are used. The benefits of multiple levels of parallelism offer a significant performance improvement of 18% for hybrid programs. We also remark that the block partitioning outweighs cyclic partitioning with an average of 5% faster.

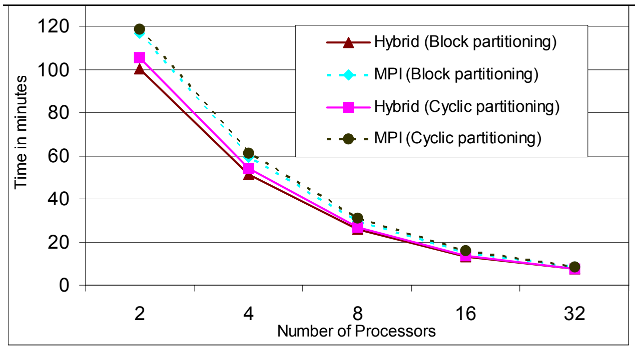

Fig. 4. Execution time of MPEG-2 encoder on 2-way Atlantis cluster.

C. The n-body Simulation

Figure 5 presents the execution time of the MPI and hybrid codes simulating the interactions among 10^5 bodies on 4-way Diplo and 2-way Atlantis. No matter how many processors are used, the hybrid implementation outperforms the pure MPI one by average factors of 1.52 on Diplo and 1.21 on Atlantis at all times. We observe that the factor on Diplo is higher than on Atlantis, resulting from a greater number of OpenMP threads. As a result, it is expected that this factor will be even higher in 8 or 16-way clusters.

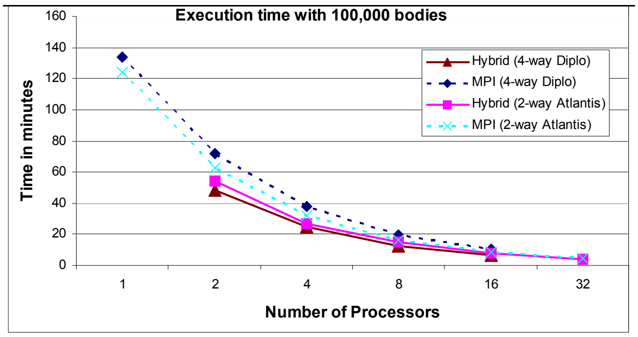

Fig. 5. Execution time of 10^5-body on 4-way Diplo and 2-way Atlantis.

V. CONCLUSION

In this paper, we studied the performance and the programming efforts for two different applications, MPEG-2 encoder and n-body simulation, under two parallel programming paradigms: pure MPI and hybrid MPI-OpenMP. Given the ability to achieve multiple levels of parallelism, the hybrid MPI-OpenMP programs outperform the corresponding pure MPI programs in terms of execution time for both applications whatever processors are used.

REFERENCES

[1] K. Yamazaki, K. Ikegami, and S. Oyanagi, “Speed Improvement of MPEG-2 Encoding using Hybrid Parallel Programming”, IPSJ and IEICE, FIT 2006, Information Technology Letters, LC-005, Vol.5, 2006.

[2] J. Dubinski, “A parallel tree code”, New Astronomy 1 (1996) 133-147.

[3] Treecode, treecode guide, http://ifa.hawaii.edu/~barnes/treecode/