Chip Level Hybrid Parallelization for Multi-core Clusters

Tetsuya Nakao (0710052)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 5, 2009

Keywords: multi-core processor, cluster, hybrid parallelization, parallel programming model, MPI, OpenMP.

1 Introduction

In recent years, a multi-core processor has gained great popularity. It has multiple core and large L2 or L3 shared cache in the processor package. The feature of the multi-core processor is densely integrated architecture. In parallel computers, the multi-core processor has also become popular. According to TOP500 that ranks performance of world-wide parallel computers, 336 computer systems have quad-core processors and 153 computer systems have dual-core processors in its ranking lists of November, 2008. In JAIST, Altix4700 which has 98 dual-core processor is running as of January, 2009. Additionally, Cray XT5 which has 256 quad-core processor will be scheduled for introduction in March, 2009. It is certain that parallel computer which has multi-core processors will increase in the future.

A multi-core processor architectures are different from previous single-core processors. Therefore, new parallelization method is required to improve the performance of multi-core parallel computers. One of the typical parallelization method is PureMPI which communicates with each core by passing messages. However, this method can't use shared cache in the multi-core processors efficiently because of independent memory space by

process. And there is a possibility of great communication overhead because of passing messages among all cores. The other typical parallelization method is NodeHybrid which communicates with each node by passing messages and use shared memory parallelization within nodes. However, this method has possibility of shared memory parallelization overhead which use all cores within nodes. Therefore, a new method about the program parallelization is required in order to get great performance of multi-core parallel computers.

2 Chip level hybrid parallelization

We propose chip level hybrid parallelization that communicates with each node and multi-core processor chip by passing messages (MPI) and use shared memory parallelization (OpenMP) within chips, in multi-core cluster. Our method uses multi-core processor chip architectures (shared cacahe architectures) within nodes. To achieve our method, we must control some processes and threads within nodes and bind one process to one chip and threads to cores within the chip. In this research, We used Processor Affinity Mechanism that can assign processes and threads explicitly to cores.

Compared with typical PureMPI that communicates with all cores, our method can reduce communication overhead and use sahred cache efficiently. On the other hand, compared with typical NodeHybrid that use shared memory parallelization within nodes, our method has possibility of decrease shared memory parallelization overhead.

3 Evaluation

We performed some experiments with matrix product as numerical basic calculation and calculation of pseudo inverse matrix as actual application. The Calculation of pseudo inverse matrix is used in Fluorescence Molecular Tomographys which have been studied recently as bottleneck. The enhancement was compared against typical PureMPI and NodeHybrid as mentioned in Section 1. Quad-core cluster (Quad-core Opteron2.1GHz 2-sockets, GigabitEthernet, 4nodes) was used for the evaluation.

In the matrix product experiment, our method have got about the same performance compared with PureMPI and reduced processing time compared with NodeHybrid. This is due to the low communication overhead in PureMPI and the great shared memory parallelization overhead in NodeHybrid. In the calculation of pseudo inverse experiment, our method have reduced processing time by about 57% of typical PureMPI processing time and by about 51% of typical NodeHybrid processing time when using all nodes and matrix size 3840. This is due to the great communication overhead in PureMPI when using all nodes and the great shared memory parallelization overhead in NodeHybrid to use all cores within nodes for shared parallelization. Therefore, our method is a good balance between MPI communication overhead and OpenMP shared memory parallelization overhead.

4 Conclusion

Our work foucsed on using multi-core processor chip architecture (shared cache) within nodes and proposed chip level hybrid parallelization to reduce whole processing time. We performed experiments with matrix product and calculation of pseudo inverse matrix. In the matrix product experiment, our method got about the same performance compared with PureMPI and reduced processing time compared with NodeHybrid. In the calculation of pseudo inverse experiment, our method reduced processing time by about 57% of PureMPI processing time and by about 51% of NodeHybrid processing time when using all nodes and matrix size 3840. Therefore, our method will be one of the best parallelization model in multi-core cluster.