Parallelized Mini-Max Search and Alpha-Beta Pruning

Wei-Cheng Wu
Electrical Engineering
and
Computer Science
Honors Program,
NCTU
spencerwu85@gmail.com

Tsung-en Hsiao
Electrical Engineering
and
Computer Science
Honors Program,
NCTU
tsn@cs.nctu.edu.tw

Yi-Fan Wu
Department of
Computer Science,
NCTU
jameswu1212@gmail.com

Keywords

Parallel, Alpha-beta Search, Mini-max Search, AI, Game, Learning

1. INTRODUCTION

Mini-Max search with Alpha-Beta Pruning is a common method to implement game AI such as Chess, Go, Renju etc. We can utilize parallel programming techniques on those algorithms. We will implement several games with different parameters, and show how parallel programming speed up the calculation.

2. STATEMENT OF THE PROBLEM

We want to implement the Mini-Max Search and Mini-Max search with Alpha-beta Pruning in both parallel way and normal way. We can evaluate how parallelization affects game AI systems.

3. PROPOSED APPROCHES

Mini-Max search can be easily parallelized by spliting the tree from the root, make the children of the root be the root node of the threads created. We are going to utilize both CPU and GPU. We will compare all the results in a graph.

4. LANGUAGE SELECTION

In the CPU part, we want to use OpenMP. OpenMP is easier for us to convert our squential code into a parallel program. In the GPU part, we want to use OpenCL for our implementation since it is more portable and a open standard.

5. RELATED WORK

Several works have been done on this subject. Includes the utilization of GPU and CPU. Kamil Rocki et al of University of Tokyo implement mini-max tree searching on GPU. Brian Greskamp of UIUC shows how to parallizing chess program by PVSPlit algorithm.

6. EXPECTED RESULTS

We expect to spend less time through parrallized mini-max than alpha-beta pruning to run the same data. Also, we should have a better performance with more threads created and more CPU cores used in our model.

7. TIME TABLE

- November - Finish the sequencial program of our model.
- December - Conver the sdquencial code into parrallel program. will handle the GPU model. The openMP model will be done by and Wei-Cheng Wu.

8. REFERENCES

1. Parallel Minimax Tree Searching on GPU by Kamil Rocki and Reiji Suda, 2010.
2. Parallelizing a Simple Chess Program by Brian Greskamp, 2003.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235