Universiteit
Leiden
The Netherlands

Structured parallel programming for Monte Carlo Tree Search

Mirsoleimani, S.A.

Citation

Mirsoleimani, S. A. (2020, June 17). *Structured parallel programming for Monte Carlo Tree Search*. *SIKS Dissertation Series*. Retrieved from https://hdl.handle.net/1887/119358

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/119358

Note: To cite this publication please use the final published version (if applicable).

Cover Page

Universiteit Leiden

The handle http://hdl.handle.net/1887/119358 holds various files of this Leiden University dissertation.

Author: Mirsoleimani, S.A.
Title: Structured parallel programming for Monte Carlo tree search
Issue Date: 2020-06-17

Structured Parallel Programming

FOR

Monte Carlo Tree Search

S. Ali Mirsoleimani

Structured Parallel Programming

FOR

Monte Carlo Tree Search

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof. mr. C.J.J.M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op woensdag 17 juni 2020
klokke 11.15 uur

door

Sayyed Ali Mirsoleimani
geboren te Abadeh, Iran
in 1986

Promotoren:
Prof. dr. H. J. van den Herik
Prof. dr. A. Plaat
Copromotor:
Dr. J. A. M. Vermaseren Nikhef

Promotiecommissie:
Prof. dr. P. J. G. Mulders Vrije Universiteit Amsterdam
Prof. dr. F. J. Verbeek
Prof. dr. H. A. G. Wijshoff
Dr. F. Khunjush Shiraz University, Shiraz, Iran
Dr. W. A. Kosters
Dr. ir. A. L. Varbanescu Universiteit van Amsterdam

HEPGAME, ERC Advanced Grant No. 320651
The research reported in this thesis has been additionally funded by Nikhef, the Nationaal instituut voor subatomaire fysica.

In the first year, the research reported in this thesis has been performed at Tilburg center for Cognition and Communication (TiCC) at Tilburg University, the Netherlands.

The research reported in this thesis has been completed at Leiden Centre of Data Science (LCDS) hosted by Leiden Institute of Advanced Computer Science (LIACS) at the Faculty of Science, Universiteit Leiden, the Netherlands.

SIKS Dissertation Series No. 2020-08
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Copyright © 2020 by S.A. Mirsoleimani

An electronic version of this dissertation is available at
`http://openaccess.leidenuniv.nl/`.

I would like to dedicate this thesis to my wife Elahe and to my parents,
for all of their love and support.

In loving memory of my grandfathers,
Bahram and Abolghasem

Preface

The thesis is part of a bigger project, the HEPGAME (High Energy Physics Game). The project started in 2011 when Jos Vermaseren developed the first ideas on improving FORM at Nikhef, Amsterdam. In 2012 he submitted an ERC advanced research grant together with Tilburg University. It was accepted on 12/12/2012. Half a year later in July 2013, the program started. The main objective for HEPGAME was the utilization of AI solutions, particularly by using MCTS for simplification of HEP calculations. One of the issues is solving mathematical expressions of interest with millions of terms. Up to 2011, these calculations were executed with the FORM program, which is software for symbolic manipulation. These calculations are computationally intensive and take a large amount of time. Hence, the FORM program was parallelized to solve large equations in a reasonable amount of time. Therefore, any new algorithm, for instance, the ones based on MCTS, should also be parallelized. Here our research comes in. It is dedicated to parallelization of MCTS on multi-core and many-core processors. The research was ambitious and challenging. Therefore, we divided the research area into three main parts: (1) the evaluation of current methods for parallelization of MCTS, (2) addressing the shortcomings in these methods, and (3) providing new ways of parallelization for MCTS. In the first part, we investigated the current methods and evaluated them in terms of performance and scalability on both multi-core and manycore processors. In the second part, we examined how we can solve the actual shortcomings in the existing parallelization methods for MCTS. The third part was dedicated to finding new ideas, methods, and ways beyond the existing ones to parallelize MCTS.

Sayyed Ali Mirsoleimani, Leiden, July 2019

Contents

List of Definitions

List of Figures

List of Tables

List of Listings

List of Abbreviations

3PMCTS	Pipeline Pattern for Parallel MCTS.
FIFO	First In, First Out.
FMA	Fused Multiply Add.
GFLOPS	Giga Floating Point Operations per Second.
GIPS	Giga Integers per Second.
GSCPM	Grain Size Controlled Parallel MCTS.
HEP	High Energy Physics.
HEPGAME	High Energy Physics Game.
ILD	Iteration-Level Dependency.
ILP	Iteration-Level Parallelism.
ILT	Iteration-Level Task.
ISA	Instruction Set Architecture.
MC	Memory Controller.
MCTS	Monte Carlo Tree Search.
MIC	Many Integrated Core.
NUMA	Non Uniform Memory Access.

OLD Operation-Level Dependency.
OLP Operation-Level Parallelism.
OLT Operation-Level Task.

PPS Playouts per Second.
PS Problem Statement.
PW Percentage of Wins.

RNG Random Number Generation.
RQ Research Question.

SMT Simultaneous Multithreading.

TBB Threading Building Blocks.
TD Tag Directories.
TPFIFO Thread Pool with FIFO scheduling.

UCB Upper Confidence Bound.
UCT Upper Confidence Bounds for Trees.
UMA Uniform Memory Access.

VPUs Vector Processing Units.