MACHINE LEARNED MACHINES: REINFORCEMENT LEARNING EXPLORATION FOR ARCHITECTURE CO-OPTIMIZATION

RAHUL JAIN

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI
OCTOBER 2017

MACHINE LEARNED MACHINES: REINFORCEMENT LEARNING EXPLORATION FOR ARCHITECTURE CO-OPTIMIZATION

by

RAHUL JAIN

Department of Computer Science and Engineering

Submitted

in fulfillment of the requirements of the degree of Doctor of Philosophy

to the

Indian Institute of Technology Delhi

October 2017

THESIS CERTIFICATE

This is to certify that the thesis entitled **Machine Learned Machines: Reinforcement Learning Exploration for Architecture Co-optimization**, submitted by **Rahul Jain**, to the Indian Institute of Technology Delhi, for the award of the degree of **Doctor of Philosophy**, is a bona fide record of the research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof Preeti Ranjan Panda
Phd Supervisor
Professor
Department of Computer Science and Engineering
Indian Institute of Technology Delhi
New Delhi, 110010

Place: New Delhi
Date:

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Prof. Preeti Ranjan Panda for his patience, motivation, research expertise, and support. He always gave the freedom to pursue varied ideas and helped me enhance my research skills. His guidance helped me improve my research methodology and writing skills.

I am immensely grateful to Sreenivas Subramoney for being my industry mentor from Intel as part of the Prime Minister's Doctoral Fellowship. I thank him for taking out the time from his busy schedule for our discussion meetings and provide valuable industry perspective to our research. I would also like to acknowledge CII and SERB for the partial financial support and management of the PM Fellowship.

Besides my mentors, I would like to thank other faculty members, Prof Anshul Kumar, Prof M. Balakrishnan, Prof. Kolin Paul, and Prof Sarangi of the CSE Department who provided with valuable feedback during the various interactions and presentations in IIT-Delhi. My sincere thanks to Prof Mausam and Ankit Anand for providing useful feedback and suggestions on the machine learning models used in this research. I am thankful to Prof Sarangi and the Baadal Team for providing high-end computational infrastructure for the experiments.

I would also like to thank my lab members Rajshekar and Prathmesh for their valuable feedback on my research writings. I was very fortunate to have got unstinting support from other lab members, Sandeep, Gayathri, Neetu, Lokesh, Sakshi and Hadi, and heartily thank them for it.

Rahul Jain

ABSTRACT

KEYWORDS: Chip Multiprocessors (CMP), Energy Efficient Computation, Dynamic Voltage and Frequency Scaling (DVFS), Dynamic Cache Partitioning, Dynamic Cache Co-partitioning, Multiple Resource Management, Co-optimization, Reinforcement Learning

Modern multi-core systems provide huge computational capabilities which can be used to run multiple processes concurrently. The integration of multiple cores on a single chip has resulted in very large performance gap between the processor and the off-chip memory. This has resulted in multiple levels of on-chip caches which are often shared among multiple applications executing simultaneously on various cores. The increasing core count results in larger shared resources like the last level cache, interconnect network, memory controllers, etc. To achieve the best possible performance within limited power budget the various system resources need to be allocated effectively. Any mismatch between runtime resource requirement and allocation results in a sub-optimal performance.

Different optimization techniques exist for addressing the problem of mismatch between the dynamic requirement and runtime allocation of the system resources. Choosing between multiple optimizations at runtime is complex due to the non-additive effects, making the scenario suitable for the application of machine learning techniques. We have proposed and evaluated different reinforcement learning (RL) based techniques targeting multiple resource allocation at runtime. We study two different co-optimization problems and show that the co-optimization techniques result in better system metrics than any of the techniques applied individually. We study the application of reinforcement learning (RL) techniques to the various architecture optimization problems. The RL technique is capable of learning the utility function of the various resource allocations at runtime by interacting with the architecture. We have explored multiple agent RL models based on independent, cooperative and coordinated learners.

The first co-optimization problem is targeted towards optimization of the system energy-delay-product (EDP). We present a novel method, Machine Learned Machines (MLM), by using Online Reinforcement Learning (RL) to perform dynamic partitioning of the last level cache (LLC), along with dynamic voltage and frequency scaling (DVFS) of the cores and uncore (interconnection network and LLC). The DVFS+DCP co-optimization problem is modeled using independent, cooperative and coordinated multi-agent learners and compared on different system metrics.

The second co-optimization problem is targeted towards optimization of the system throughput (STP). We propose a novel problem of dynamic cache co-partitioning (DCCP) of the multiple cache levels and present a coordinated multi-agent RL technique, Machine Learned Caches (MLC). MLC results in better system metrics compared to the independent application of the dynamic cache partitioning (DCP) techniques at multiple cache levels.

The explored machine learning techniques are found to be effective in handling the different co-optimization problems resulting in better system metrics compared to state-of-the-art research work. We study the various co-optimization techniques targeting different system metrics and study their effect on system EDP, system throughput (STP), and Fairness. The proposed MLM co-optimization techniques are effective in improving the system EDP by up to 21.7% with less than 5% STP degradation on a 4-core system. The proposed MLC co-partitioning exhibited 9.35% STP improvement with 13.5% system EDP improvement on an 8-SMT core (4 physical core) system.

सार

मुख्य शब्द: चिप मल्टिप्रोसेसर, ऊर्जा कुशल गणना, गतिशील वोल्ट और आवृत्ति स्केलिंग (डीवीएफएस), गतिशील कैश विभाजन, गतिशील कैश सह-विभाजन, एकाधिक संसाधन प्रबंधन, कोपाटीकरण, सुदृढीकरण सीखना

आधुनिक बहु-कोर सिस्टम भारी कम्प्यूटेशनल क्षमताएं प्रदान करते हैं, जो कई प्रोसेस को एक साथ चलाने के लिए इस्तेमाल किया जा सकता है। एकल चिप पर एकाधिक कोर के एकीकरण के परिणामस्वरूप प्रोसेसर और ऑफ-चिप मेमोरी के बीच बहुत बड़ी प्रदर्शन अंतर है। इसके परिणामस्वरूप चिप कैश के कई स्तर होते हैं, जो अक्सर विभिन्न कोर पर एक साथ क्रियान्वित करने वाले कई अनुप्रयोगों में साझा होते हैं। बढ़ते कोर गिनती बड़े साझा संसाधनों जैसे अंतिम स्तर कैश, इंटरकनेक्ट नेटवर्क, मेमोरी नियंत्रक इत्यादि में होती है। सीमित बिजली बजट के भीतर सर्वोत्तम संभव प्रदर्शन प्राप्त करने के लिए विभिन्न सिस्टम संसाधनों को प्रभावी ढंग से आवंटित करने की आवश्यकता है। रनटाइम संसाधन आवश्यकता और आवंटन के परिणामों के बीच किसी भी बेमेल को उप-इष्टतम प्रदर्शन में नहीं मिलता है।

गतिशील आवश्यकता और सिस्टम संसाधनों के क्रम समय आवंटन के बीच बेमेल की समस्या को संबोधित करने के लिए अलग अनुकूलन तकनीक मौजूद हैं। रन-टाइम पर एकाधिक ऑप्टिमाइजेशन के बीच चुनना जटिल कारण है, क्योंकि नॉन-एडिटिव इफेक्ट्स, मशीन सीखने की तकनीक के आवेदन के लिए परिदृश्य उपयुक्त है। हमने कई सुदृढीकरण सीखने (आरएल) आधारित तकनीकों का मूल्यांकन किया है जो रनटाइम पर कई संसाधन आवंटन को लक्षित करते हैं। हम दो अलग-अलग सह-अनुकूलन समस्याओं का अध्ययन करते हैं और दिखाते हैं कि सह-अनुकूलन तकनीक व्यक्तिगत रूप से लागू की जाने वाली किसी भी तकनीक की तुलना में बेहतर प्रणाली मीट्रिक का परिणाम देती हैं। हम विभिन्न वास्तुकला अनुकूलन समस्याओं के लिए सुदृढीकरण सीखने (आरएल) तकनीकों के अनुप्रयोग का अध्ययन करते हैं। आरएल तकनीक वास्तुकला के साथ बातचीत करके रनटाइम पर विभिन्न संसाधन आवंटन के उपयोगिता कार्य को सीखने में सक्षम है। हमने स्वतंत्र, सहकारी और समन्वित शिक्षार्थियों पर आधारित एकाधिक एजेंट आरएल मॉडल का पता लगाया है।

पहली सह-अनुकूलन समस्या प्रणाली ऊर्जा-देरी-उत्पाद (ईडीपी) के अनुकूलन के लिए लक्षित है हम कोर और अनकोर के गतिशील वोल्टेज और आवृत्ति स्केलिंग (डीवीएफएस) के साथ अंतिम स्तर कैश (एलएलसी) के गतिशील विभाजन करने के लिए ऑनलाइन सुदृढीकरण सीखने (आरएल) का उपयोग करके, मशीन सीखा मशीन (एमएलएम) उपन्यास विधि पेश करते हैं (इंटरकनेक्शन नेटवर्क और एलएलसी)। डीवीएफएस + डीसीपी सह अनुकूलन समस्या को स्वतंत्र, सहकारी और समन्वित बहु-एजेंट शिक्षार्थियों का उपयोग करके मॉडलिंग किया जाता है और विभिन्न सिस्टम मैट्रिक्स की तुलना में किया जाता है।

दूसरी सह-अनुकूलन समस्या सिस्टम थ्रूपुट (एसटीपी) के अनुकूलन के लिए लक्षित है। हम कई कैश स्तरों के गतिशील कैश सह-विभाजन (डीसीसीपी) की एक उपन्यास समस्या का प्रस्ताव करते हैं और एक समन्वित बहु-एजेंट आरएल तकनीक, मशीन सीखित कैश (एमएलसी) पेश करते हैं। कई कैश स्तरों पर गतिशील कैश विभाजन (डीसीपी) तकनीकों के स्वतंत्र अनुप्रयोग की तुलना में बेहतर सिस्टम मीट्रिक में एमएलसी परिणाम होते हैं

अन्वेषित मशीन सीखने की तकनीकों को विभिन्न सह-अनुकूलन समस्याओं से निपटने में प्रभावी पाया जाता है जिसके परिणामस्वरूप अत्याधुनिक अनुसंधान कार्य के मुकाबले बेहतर सिस्टम मीट्रिक होते हैं। हम विभिन्न सह-ऑप्टिमाइज़ेशन तकनीकों का अध्ययन करते हैं जो अलग-अलग सिस्टम मैट्रिक्स को लक्षित करते हैं और सिस्टम ईडीपी, सिस्टम थ्रूपुट (एसटीपी), और निष्पक्षता पर उनके प्रभाव का अध्ययन करते हैं। प्रस्तावित एमएलएम सह-अनुकूलन तकनीक 4-कोर प्रणाली पर 5% से कम एसटीपी गिरावट के साथ 21.7% तक प्रणाली ईडीपी में सुधार के लिए प्रभावी हैं। प्रस्तावित एमएलसी सह-विभाजन 9.35% एसटीपी सुधार के साथ 13.5% प्रणाली ईडीपी सुधार 8-एसएमटी कोर (4 भौतिक कोर) प्रणाली पर प्रदर्शित किया गया।

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

RL	Reinforcement Learning
MDP	Markov Decision Process
ANN	Artificial Neural Network
MLM	Machine Learned Machines
MLC	Machine Learned Caches
LLC	Last Level Cache
NoC	Network on Chip
SMT	Simultaneous Multi-Threading
DVFS	Dynamic Voltage and Frequency Scaling
EDP	Energy Delay Product
DCP	Dynamic Cache Partitioning
DCCP	Dynamic Cache Co-partitioning
MLP	Memory-Level Parallelism
MCFQ	MLP and Cache-Friendliness aware Quasi-partitioning
UCP	Utility Based Cache Partitioning
XCH	XChange

NOTATION

cpi	cycles-per-instruction
ipc	instructions-per-cycle
tpi	time-per-instruction
tr	ratio of current and previous interval tpi values
α	Q-Update learning rate
γ	Q-Update reward discount factor