THEORY AND APPLICATIONS OF NON-BLOCKING SLOT SCHEDULERS

POOJA AGGARWAL

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI
JUNE 2016

THEORY AND APPLICATIONS OF NON-BLOCKING SLOT SCHEDULERS

by

POOJA AGGARWAL

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

JUNE 2016

Certificate

This is to certify that the thesis titled **THEORY AND APPLICATIONS OF NON-BLOCKING SLOT SCHEDULERS** being submitted by **Ms. Pooja Aggarwal** for the award of **Doctor of Philosophy** in **Computer Science and Engineering** is a record of bona fide work carried out by her under my guidance and supervision at the **Department of Computer Science and Engineering, Indian Institute of Technology Delhi**. The work presented in this thesis has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma.

Smruti Ranjan Sarangi
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology Delhi
New Delhi- 110016

Acknowledgements

I owe my gratitude to all those people who have made this dissertation possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost, I would like to thank my advisor Prof. Smruti Ranjan Sarangi for giving me an opportunity to work with him and for guiding me through out my research work as well as courses. The blessings, help and guidance given by him shall carry me a long way in the journey of life, on which I am about to embark. He has supported, inspired, and encouraged me in many different ways through different stages. I feel so fortunate to have been one of his students.

My sincere thanks to Prof. Kolin Paul, Prof. Sanjeeva Parsad and Prof. S. Rao for serving on my Student Research Committee and for the suggestions which incented me to widen my research from various perspectives. I would like to thank NetApp India for their support and collaboration.

I also take this opportunity to thank all my fellow mates of Srishti group, for their effort to make the department such an excellent environment to work. I would like to specially thank Seep, Abhishek, Prathmesh, Sandeep, Rajshekhar and Kapil for all the help, discussions and great company.

I would like to thank the staff of the Department of Computer Science for their administrative and technical help. I would thank Vandana Ahluwalia for her timely assistance in lab related issues.

Deepak Ravi, thanks for all the brainstorming sessions and constructive criticisms at different stages of my research which were thought-provoking and they helped me focus my ideas. My roommate, Anuradha Sharma, thank you very much for making the atmosphere of our room as friendly as possible and for motivating me through out my duration at IIT Delhi.

Most importantly, my thanks go to my family. I wish to thank my parents and my brother and

sister for their endless and unconditional love, for their sustained support and encouragement. I am very grateful to Prateek, my husband, for his love, support and care. I would for certain not be here without your support.

Pooja Aggarwal

Abstract

In this thesis, we consider the design and applications of parallel non-blocking slot scheduling algorithms. Slot schedulers divide time into discrete quanta called *slots*, and schedule resources at the granularity of slots. They are typically used in high throughput I/O systems, data centers, video servers, and network drivers. In this thesis, we first present an efficient implementation of a concurrent non-blocking slot scheduler, and then we apply our slot scheduler to solve scheduling and resource allocation problems. Typically, a resource in a concurrent system may be accessed by only one thread at a time. To share a single resource, threads co-ordinate using mutual exclusion algorithms. When threads require simultaneous access to multiple resources, the task of coordinating them properly is the resource allocation problem. However, these mutual exclusion algorithms (using locks) suffer from inherent problems such as deadlocks, starvation, priority inversion, which makes the lock-based implementation untenable.

We propose a family of parallel lock-free and wait-free linearizable algorithms to map the resource demands to a set of time slots. This mapping of resources to currently available time slots introduces many problems. In specific, we look at problems for reserving, as well as freeing a set of contiguous slots.

We show that in a system with 64 threads, it is possible to get speedups of 10X by using lock-free algorithms as compared to a baseline implementation that uses locks. We additionally propose wait-free algorithms, whose mean performance is roughly the same as the version with locks. However, they suffer from significantly lower jitter and ensure a high degree of fairness among threads. We next propose a class of algorithms using software transactional memory (STM) that are in the middle of the spectrum. They equitably balance fairness and throughput, and are much simpler to design and verify.

Subsequently, we propose applications of our slot scheduler. First, we applied our slot scheduler in developing a parallel architectural simulator to schedule concurrent accesses of shared memory structures. Prior approaches for this problem either used locks, or used methods that introduced a lot of simulator error. Our method provides impressive speedups and minimizes

simulation error.

With the advent of solid state drives (SSDs), storage vendors are increasingly preferring them because their I/O throughput can scale up to a million IOPS (I/O operations per second). These disruptive changes in the world of storage systems and software has made the scheduler a bottleneck. Supporting these devices is well beyond the means of conventional scheduling techniques, and thus a more scalable solution is required. To genuinely satisfy our requirements of having a high throughput scheduler, we extend our non-blocking slot scheduling algorithms for I/O intensive workloads and make them aware of dependencies, deadlines, request lengths, priorities, and evaluate their feasibility for SSDs with different RAID configurations.

We next propose a data structure inspired by our techniques to help us implement lock-free and wait-free algorithms. To implement non-blocking operations in a linearizable and non-blocking manner, requires us to associate additional information with memory words in a data structure. Current approaches either use some bits in a memory word to pack additional information (*packing*), or use the bits to store a pointer to an object that contains additional information (*redirection*), and the original data item. The former approach restricts the number of bits for each additional field and this reduces the range of the field, and the latter approach is wasteful in terms of space. We propose a novel method called a memory word expander. It caches information for a set of memory locations that need to be augmented with additional information. It supports traditional get, set, and CAS operations, and tries to maintain state for a minimum number of entries.

Contents

List of Figures

List of Tables