SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Low Latency Scheduling on Many-Core CPUs

Ismail Safa Toy

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Low Latency Scheduling on Many-Core CPUs

Niedriglatenz-Scheduling auf Mehrkern-CPUs

Author:	Ismail Safa Toy
Supervisor:	Prof. Viktor Leis
Advisor:	Marcus Müller
Submission Date:	Submission date

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Munich, Submission date Ismail Safa Toy

Acknowledgments

Abstract

This thesis improves parallel query execution by enhancing a scheduler based on the morsel-driven framework to effectively utilize the capabilities of modern many-core processors, with a specific focus on NUMA (Non-Uniform Memory Access) challenges. The growing decentralization of memory controllers in many-core systems complicates efficient data access and task scheduling, necessitating new approaches to maintain high performance and scalability. The enhanced scheduler dynamically assigns small data fragments, or morsels, to worker threads, adjusting in real-time to changes in data access speeds and task demands. This approach directly addresses NUMA-related issues by optimizing data locality and minimizing unnecessary data movement across different memory nodes, thus achieving elasticity. Extensive benchmarks, including comparisons with other multithreading schedulers like oneAPI and OpenMP, demonstrate the scheduler's effectiveness, especially in small and big tasks.

Contents

1 Introduction

1.1 Section

TODO: Explain the need, why it emerged etc.

2 Literature Review

2.1 Section

Todo: Compare common scheduling tactics. Give an overview and show the uniqueness of the morsels. Todo: Read more into NVidia Bend Todo: Read more into Volcano and time share based scheduling

3 Theoretical Part

3.1 Section

Todo: explain Numa, morsels and elasticity. Todo: explain solutions by graphical and pseudocode Todo: expand with OneApi etc.

4 Design of the scheduler

4.1 Section

Todo: explain the design choices Todo: Graphics Todo: code maybe

5 Implementation

Todo: code maybe Todo: explain the hardships faced

6 Optimizations

6.1 Section

Todo: explain the impact of NUMA, morsels and possibly small task optimizations
Todo: create lots of data

7 Performance Testing

7.1 Section

Todo: compare it with the others and unoptimized scenario Todo: Use TPC-H and other means to standartize the testing.

8 Discussion

8.1 Section

Todo: Detailed discussion on scheduler performance and NUMA handling. Assessment of morsel efficiency for small-scale tasks.

9 Conclusion & Future work

9.1 Section

Conclusions on the effectiveness of the NUMA-aware scheduler. Directions for future improvements and research.

Abbreviations

List of Figures

List of Tables