CS 530: High-Performance Computing
Benchmarking Matrix Multiplication: Sequential vs Multithreaded

Nathan Chapman

Department of Computer Science
Central Washington University

May 8, 2024

Contents

1 Introduction

- Many things can benefit from parallel execution
- Matrix multiplication is “embarissingly parallel”
- Makes it a good benchmark to compare sequential execution to parallel

2 Methods

- Julia
- Threadpools
- Multithreading
- Computational Complexity of sequential
- Computational Complexity of parallel

3 Results

- Timing plots for many long running experiments
- run timings multiple times

4 Discussion

- multithreading can be hard to implement
- sometimes it can be easier when the problem is embarissingly parallel
- new languages make parallelism trivial
- parallel implementation can save potentially massive amounts of time
- further parallelism and performance can be achieved with GPU programming like CUDA

5 Conclusion

- a lot of applications take a long time because a lot of things need to be done
- these applications can benefit from parallelism via multithreading
-