Maseeh College of Engineering and Computer Science

PORTLAND STATE UNIVERSITY

ECE 688: WINTER 2024

ADVANCED COMPUTER ARCHITECTURE II

LINEAR EQUATIONS SOLVER

PROJECT PROPOSAL

MOHAMED GHONIM
PHANINDRA VEMIREDDY
AHLIAH NORDSTROM
ALEXANDER MASO

01/29/2024

Table of Contents

Project Proposal: Advanced Parallel Equation Solver for Large-Scale Systems

Mohamed Ghonim, Phanindra Vemireddy, Ahliah Nordstrom, Alexander Maso

Introduction and Context:

This project is proposed as a key practical assignment for the ECE588/688 (Advanced Computer Architecture II) class, in which we study multicore processors, multiple processor systems, and parallel programming. The objective of the project is to reinforce theoretical knowledge with a hands-on approach, exploring parallel computation in the context of solving large-scale systems of linear equations.

Project Overview:

The project involves developing a Parallel Programming Solver, which will implement both a serial and a parallel program. The task is to solve a system of linear equations represented by a 5000x5000 matrix using LU Decomposition. This matrix size is chosen to present a significant computational challenge, suitable for leveraging the capabilities of multithread or multiprocessor systems.

Rationale for Algorithm and Technology Choices:

- **LU Decomposition:** Given the focus on large-scale matrix operations and the course's emphasis on advanced computational techniques, LU Decomposition is an ideal choice. It is a well-established method for solving systems of linear equations, particularly effective for large matrices due to its decomposition approach, which breaks down a complex problem into simpler parts. It’s a systematic and efficient algorithm.

- **Pthreads for Parallelization:** The use of Pthreads (POSIX threads) aligns with the course’s exploration of multicore processor capabilities. Pthreads are a robust and versatile tool for implementing parallelism in shared-memory architectures, allowing us to exploit the full potential of multicore processing.

Project Structure:

1. **Serial Program Implementation:** We will first develop a serial version of the LU Decomposition algorithm. This implementation will set a performance baseline and provide a clear understanding of the algorithm’s complexity.

2. **Parallel Program with Pthreads:** Next, the algorithm will be parallelized using Pthreads. This phase will focus on distributing the computational workload

across multiple threads, optimizing memory usage, and minimizing synchronization overhead in a multicore environment.

3. **Performance Analysis:** Both versions will be tested on multicore and multiprocessor systems. Key metrics will include speedup, efficiency, and scalability. The primary performance indicator will be the comparison of execution times between the serial version on a single processor and the parallel version on 16 processors.

4. **Challenges and Solutions in Parallel Computing:** We anticipate encountering and addressing several challenges inherent in parallel computing, such as data race conditions, load balancing, and inter-thread communication, we will analyze any challenges encountered and will address them.

5. **Comprehensive Documentation:** Detailed documentation will be provided, including theoretical explanations, code implementation, and performance analysis, as well as future work and references used.

Future Exploration:

- **MPI Implementation:** If time permits, we plan to explore an MPI (Message Passing Interface) implementation, offering a comparison with Pthreads and insights into parallelization in distributed-memory systems.

- **Gaussian Elimination Algorithm:** Additionally, implementing and comparing the Gaussian Elimination algorithm will deepen our understanding of different strategies in parallel computation for solving linear systems.

Important of an efficient solver for large systems:

The solver system proposed in this project holds significance as it can serve as a backend component for a many applications and software programs. For instance, such a solver is crucial in fields like scientific computing, where it can be used for simulations in physics and engineering, including weather forecasting models and structural analysis. In finance, it can power complex risk assessment and optimization models. In data science and machine learning, it's fundamental for algorithms that require solving large systems of equations, such as in optimization problems or training certain types of machine learning models. By accelerating this solver through parallel, multithreaded processing, we can significantly enhance the performance and efficiency of these applications. This improvement is not just a marginal increase in speed; it's a transformative change that enables more complex computations, more detailed simulations, and faster processing of large datasets, thereby expanding the capabilities and potential of numerous software systems and applications in various industries.

Significance for Advanced Computer Architecture II:

This project is not just a demonstration of programming skills but a comprehensive exercise in applying advanced computer architecture principles. It will provide hands-on experience in utilizing multicore or multithread systems, a core aspect of the course. The insights gained will be invaluable in understanding the practical challenges and potential of parallel computing in complex computational tasks.

Conclusion:

This project is a comprehensive exploration of parallel programming techniques applied to a significant computational challenge: solving a large system of linear equations with a 5000x5000 matrix. Starting with a serial implementation using LU Decomposition, we will establish a performance baseline, followed by a parallel implementation leveraging Pthreads. This approach will not only demonstrate the practical application of concepts learned in class but also offer a comparative analysis of serial and parallel computation efficiencies. The possibility of extending the project to include MPI and Gaussian Elimination methods will further enrich our understanding of different parallel computing paradigms.

References:

[1] K. Hartnett and substantive Quanta Magazine moderates comments to facilitate an informed, "New algorithm breaks speed limit for solving linear equations," Quanta Magazine, https://www.quantamagazine.org/new-algorithm-breaks-speed-limit-for-solving-linear-equations-20210308/ (accessed Jan. 28, 2024).

[2] T. J. Dekker, W. Hoffmann, and K. Potma, "Parallel algorithms for solving large linear systems," *Journal of Computational and Applied Mathematics*, vol. 50, no. 1–3, pp. 221–232, 1994. doi:10.1016/0377-0427(94)90302-6

[3] W. by: Q. Chunawala, "Fast algorithms for solving a system of linear equations," Baeldung on Computer Science, https://www.baeldung.com/cs/solving-system-linear-equations (accessed Jan. 28, 2024).

Appendix:

What’s LU Decomposition

LU Decomposition is a numerical method used to solve systems of linear equations, perform matrix inversion, and calculate the determinant of a matrix. It decomposes a matrix into two triangular matrices:

- L (Lower Triangular Matrix): All elements above the main diagonal are zero.
- U (Upper Triangular Matrix): All elements below the main diagonal are zero.

Given a square matrix A, LU Decomposition finds matrices L and U such that A=LU. This decomposition is particularly useful because once L and U are determined, systems of linear equations can be solved more efficiently. It simplifies the process by breaking it into two stages: solving Ly=b for y using forward substitution, and then solving Ux=y for x using backward substitution. This method is highly beneficial for solving multiple systems of equations that have the same coefficient matrix (A) but different constant vectors (b).