**Accelerating Electromagnetic Simulations: GPU Implementation of FDTD Algorithm**

|  |  |  |
| --- | --- | --- |
| Zulfiqar Mohammadi Student a1850245@student.adelaide.edu.au | Thomas Woolford Industry supervisor thomas.woolford@swordfish.com.au | Olaf Maennel Academic supervisor olaf.maennel@adelaide.edu.au |

ABSTRACT

Radar propagation models play a crucial role in various applications, yet their computational demands often result in prolonged simulation times. This project explores the potential of leveraging Graphics Processing Units (GPUs) to enhance the performance of radar propagation simulations. By implementing the Finite-Difference Time-Domain (FDTD) algorithm on GPUs, the aim is to accelerate electromagnetic simulations, thereby reducing computation times. This research investigates the performance and accuracy trade-offs inherent in different propagation models, evaluates their suitability for GPU implementation, and quantifies the performance benefits derived from GPU acceleration.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; • **Networks** → Network reliability

1 INTRODUCTION

In recent years, Graphics Processing Units (GPUs) have gained prominence as powerful accelerators for computationally intensive tasks due to their exceptional computational ability. Their utilization in general-purpose programming has transformed various fields, including bioinformatics, computational physics, engineering simulations, and image processing (Ying Xu & Rongling Jiang 2016). One such demanding application poised to benefit from GPU acceleration is the simulation of electromagnetic (EM) fields using the Finite-Difference Time-Domain (FDTD) method. This numerical scheme is widely employed for EM field simulation but demands substantial computational resources and fast memory access to accurately model real-world scenarios (Demir, V 2014). This study focuses on the collaboration between Central Processing Units (CPUs) and GPUs to develop a parallel algorithm for the FDTD method. Leveraging the OpenMP API for GPU implementation, the aim is to exploit the parallel processing capabilities of both CPU and GPU architectures to accelerate EM field simulations.

The motivation for investigating CPU-GPU collaboration in parallelizing the FDTD method stems from the urgent need to address the computational demands of EM field simulations. These simulations are crucial for various applications, including radar systems and wireless communication networks. However, the complexity of real-world EM scenarios often requires significant computational resources and entails substantial computational overheads. By harnessing GPUs' parallel processing capabilities alongside CPUs' multi-core capabilities through the OpenMP API, this research aims to accelerate EM field simulations. This effort aligns with the broader objective of advancing computational techniques to meet the escalating demands of EM modeling across various domains, fostering innovation and enabling the development of more efficient and scalable simulation methodologies.

2 LITERATURE REVIEW

2.1 Project Scope

As discussed above, this project aims to enhance the speed and precision of electromagnetic simulations by utilizing both CPUs and GPUs collaboratively.

2.1.1 Optimizing FDTD Algorithm on GPU Clusters:

In this phase of research, the aim is to maximize the utilization of GPUs for accelerating electromagnetic simulations. Given GPUs' proficiency in parallel processing, our aim is to fully harness this capability. Advanced techniques will be implemented such as utilizing GPU texture memory and pinned host memory to enhance data access speed (Ying Xu & Rongling Jiang 2016). Additionally, optimizing data transfer between the CPU and GPU will be a key focus to minimize delays and ensure efficient utilization of computational resources.

2.1.2 Exploring Load Balancing Techniques:

Effective load balancing is essential to ensure that computational tasks are distributed evenly across CPUs and GPUs. Various load balancing techniques will be explored to minimize idle time and optimize resource utilization. This includes investigating dynamic load balancing algorithms capable of adapting resource allocation based on changing workload demands (Ying Xu & Rongling Jiang 2016). Our objective is to achieve optimal performance across CPU-GPU clusters, even under varying computational workloads.

2.1.3 Emphasizing Numerical Accuracy:

Ensuring the accuracy of electromagnetic simulations is paramount to the reliability of our research outcomes. We will prioritize numerical accuracy by validating our simulations against known analytical solutions. Special attention will be given to validating the effectiveness of Uniaxial Perfectly Matched Layer (UPML) boundary conditions (Meng, Wong, S, Macon, C & Jin, J-M 2014). This validation process is crucial to ensuring the reliability and trustworthiness of our simulation results.

2.1.4 Conducting Performance Evaluations:

Once the computational system is configured, we will conduct comprehensive performance evaluations to assess its scalability, efficiency, and computational throughput. Our evaluations will include testing the system's ability to handle large and complex tasks without performance degradation (Demir, V 2014). We will compare the performance of our optimized system with previous methods to evaluate the improvements in accuracy and speed achieved through our research efforts. These performance evaluations will provide valuable insights into the effectiveness of our optimization strategies and guide future research directions.

2.2 Issues and challenges

Identifying and addressing the challenges inherent in collaborative CPU-GPU electromagnetic simulations using the Finite-Difference Time-Domain (FDTD) algorithm is paramount for the project's success.

2.2.1 Balancing Workload Distribution between CPU and GPU:

One significant challenge revolves around balancing the distribution of computational tasks between the CPU and GPU while minimizing data transfer overhead. This necessitates ensuring that both processors contribute their fair share of work without causing delays. To tackle this, implementing dynamic load balancing algorithms can adaptively allocate tasks based on system performance, optimizing resource utilization and computational efficiency (Demir, V 2014). Moreover, optimizing data transfer protocols can mitigate latency issues, facilitating seamless communication between the CPU and GPU.

2.2.2 Algorithm Optimization for GPU Parallel Processing:

Another challenge lies in optimizing algorithms to fully harness the parallel processing power of GPUs while ensuring synchronization and data integrity (Demir, V 2014). Inefficient algorithm design can limit the potential performance gains offered by GPU parallelism. To address this, conducting thorough algorithm profiling and optimization is crucial. Leveraging GPU-specific programming techniques such as CUDA or OpenMp and implementing synchronization mechanisms can ensure data consistency and integrity across parallel threads, thereby enhancing computational efficiency.

2.2.3 Managing GPU Memory Limitations and Optimization:

Efficient management of GPU memory limitations and optimization is also critical. Inadequate memory management can lead to memory allocation errors, performance degradation, and limited scalability. Employing memory optimization techniques such as memory pooling, data compression, and memory reuse can mitigate GPU memory constraints (Morunov, ND & Golovashkin, DL 2019). Furthermore, implementing efficient data structures and algorithms optimized for GPU architecture can improve computational efficiency, ensuring optimal utilization of available resources.

3 PROJECT PLAN

3.1 Timeline

The project timeline (attached in Appendix, Figure 1.1)includes data collection for theinitial setup andfamiliarization with the OpenMP API and CUDA for GPU implementation, followed by adaptation of the FDTD algorithm to leverage both CPU and GPU resources (Week 2-6). Subsequent phases involve implementation and optimization techniquesand performance evaluation(week 7 - 13), with expected completion within a six-month timeframe.

3.2 Methodology:

The methodology involves a systematic approach consisting of four main phases: Setup and Preparation, GPU Implementation, Performance Evaluation, and Documentation and Reporting. During the Setup and Preparation phase, the necessary hardware and software resources will be procured and configured. The GPU Implementation phase will involve adapting the FDTD algorithm for GPU parallelization, optimizing memory usage, and implementing load balancing techniques. Subsequently, the Performance Evaluation phase will include designing experiments, executing simulations, and analyzing results against analytical solutions. Finally, the Documentation and Reporting phase will involve compiling project documentation, preparing technical reports, and presenting findings to the industry and academic supervisors.

3.3 Challenges and strategies

In collaborative CPU-GPU acceleration for electromagnetic simulations using the FDTD algorithm, several challenges emerge, each requiring strategic approaches for resolution. Firstly, managing workload distribution between CPUs and GPUs presents a significant hurdle. To tackle this, a comprehensive understanding of CPU and GPU architectures, synchronization mechanisms, and load balancing strategies is crucial (Ying Xu & Rongling Jiang 2016). Secondly, optimizing GPU algorithms for efficient performance demands dedicated efforts in studying OpenMP programming, experimenting with various parallelization schemes, and leveraging available online resources (Demir, V 2014). Thirdly, ensuring efficient GPU memory management is essential to maximize memory utilization while minimizing wastage. This involves exploring techniques like memory pooling, data compression, and regular monitoring of memory usage patterns ((Ying Xu & Rongling Jiang 2016).

3.5 Contingency planning

The contingency plan for the project involves a structured approach to address challenges encountered.

3.5.1. Identification of Challenges:

The first step is to identify specific challenges encountered during the implementation of the FDTD algorithm. This may include technical limitations, resource constraints, or unexpected obstacles that hinder progress.

3.5.2 Evaluation of Alternatives and Feasibility Assessment:

If difficulties persist and significantly impede the implementation of the FDTD algorithm, we will explore alternative solutions. One such alternative is the adoption of 2D wave models, which are simpler to implement and require fewer computational resources. The industry supervisor and I will assess the feasibility and suitability of transitioning to 2D wave models. This assessment considers factors such as computational efficiency, accuracy of results, and alignment with project objectives.

3.5.3 Consultation:

Throughout the contingency planning process, regular consultations with both the supervisors are scheduled via MS teams. These consultations aim to gather insights and guidance on alternative approaches, including the adoption of 2D wave models.

3.5.4 Transition Planning and Implementation of Alternative Approach:

If the decision is made to pivot to 2D wave models, I will develop a detailed transition plan. This plan outlines the steps required to transition from the FDTD algorithm to 2D wave models, including software adjustments, data migration, and testing procedures. Following the transition plan, the chosen alternative approach, i.e., 2D wave models will be implemented. This involves updating software, adapting simulation methodologies, and ensuring compatibility with existing project infrastructure.

3.5.5 Documentation and Reporting:

Finally, I will document all aspects of the contingency planning process, including challenges encountered, decisions made, and actions taken. This documentation serves as a reference for future projects and facilitates knowledge sharing within the team.

3.4 Communication planning

The communication plan for the project involves utilizing email communication for formal updates and document sharing with both the industry supervisor and academic supervisor while scheduling weekly progress meetings via MS Teams to discuss project updates. Additionally, weekly status will be discussed by team members to summarize their achievements and outline plans for the following week, ensuring transparency and accountability.

REFERENCES

1. Ying Xu, Huimin Ma & Rongling Jiang 2016, ‘Collaborating CPU and GPU for the electromagnetic simulations with the FDTD algorithm: COLLABORATING CPU AND GPU FOR FDTD ALGORITHM’, *Concurrency and Computation*, vol. 29, pp. e3859-.
2. Meng, H-T, Nie, B-L, Wong, S, Macon, C & Jin, J-M 2014, GPU accelerated finite-element computation for electromagnetic analysis, IEEE antennas & propagation magazine, vol. 56, IEEE, PISCATAWAY, no. 2, pp. 39–62.
3. Chi, J, Liu, F, Weber, E, Li, Y & Crozier, S 2011, ‘GPU-Accelerated FDTD Modeling of Radio-Frequency Field-Tissue Interactions in High-Field MRI’, *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 6, pp. 1789–1796.
4. Demir, V 2014, ‘A simple GPU implementation of FDTD/PBC algorithm for analysis of periodic structures’, *Applied Computational Electromagnetics Society Journal*, vol. 29, no. 12, pp. 1018–1024.
5. Morunov, ND & Golovashkin, DL 2019, ‘Design features of block algorithms of FDTD-method implemented on a GPU using MATLAB’, *Kompʹûternaâ Optika*, vol. 43, no. 4, pp. 671–676.

Appendix

|  |
| --- |
|  |
| Figure 1: Project timeline |