Fall 2022: ME759 Final Project Proposal

Project Title: Implementing AES encryption algorithm in CUDA

Link to git repo for project: https://git.doit.wisc.edu/MACHABILLAVA/finalproject759

Problem statement:
To enhance the performance of AES encryption by parallelizing the algorithm using CUDA.

Motivation/Rationale:
Generally, in cryptography, 10% of the time is spent in the sender sharing the symmetric key with the receiver using asymmetric cryptographic algorithms and 90% of the time is spent in encrypting the actual data using the symmetric key previously shared and the symmetric cryptographic algorithm. AES encryption is one of the most used symmetric key encryption algorithms. Following Amdahl's Law, enhancing the performance of the most common scenario yields much better results. This is the reason why optimizing AES is beneficial.

Explain how you contemplate going about it:
AES encryption occurs in several stages as described briefly below. Each string of bytes is divided into blocks, a 4x4 matrix, and then the algorithm is applied –

1. Substitution – AES encryption uses a 16x16 substitution matrix. Each element of the block is substituted by an appropriate element of the substitution matrix. For example, if an element is 04, then this element is replaced by the element in 0th row and 4th column in the substitution matrix.
2. Shift Rows – In this step, the element in each row is shifted row ID times. In other words, row 0 is shifted 0 times, row 1 is shifted once and so on.
3. Mix Columns – Here each column is modulo multiplied with another matrix to reshuffle the data.
4. Add Round Key – Finally, a key is added to the result of the previous step to obtain the final encrypted value of the block.

For parallelizing the AES algorithm, I plan on using CUDA programming wherein each thread takes a particular element and operates on it. Although moving data during transposition etc is simpler and intuitive, it comes with some overhead. I also want to explore if the entire operation can be performed with pointer manipulation.
The end result would be a performance analysis of naïve AES implementation vs parallel AES implementation.
A stretch goal would be to implement the CTR mode of AES. CTR performs further processing prior and after AES block processing to create more randomness in the cipher text.

ME759 aspects the proposed work draws on:

- CUDA programming
- Shared memory
- Techniques to achieve maximum occupancy

Deliverables:

- Code used for the implementation in the GitLab repo
- Project report consisting of implementation flow, analysis report and the references.

How you will demonstrate what you accomplished:

- A live demonstration of the working of the AES algorithm using CUDA programming
- An analysis of sequential vs parallel implementation of AES.

Other remarks:
NA