Project Report: Parallel AES Implementation

Chris Norman
CSE633 Fall 2011
Algorithm

- AES is a block cipher algorithm used to encrypt data using a 128-bit key
- Data is divided up into 128-bit blocks and encrypted
- Each block goes through 11 rounds of encryption, with 4 steps: SubBytes, ShiftRows, MixColumns, AddRoundKey
- The ciphertext is produced and is recovered by performing decryption with the same 128-bit key
- In a sequential scheme, each block would be encrypted sequentially
Overview
Parallel Implementation

- As mentioned before, AES is rather sequential in nature due to the fact that each successive round depends on the output of the prior round.
- So we’re not interested so much in speeding up AES encryption itself, but rather encrypting the blocks in parallel.
- Being able to do this will afford us huge gains in efficiency and speedup.
Parallel Implementation

• Utilized PolarSSL’s AES library to perform AES encryption
• Used MPI for parallelization
• Performed parallelization by:
  o Assigning each PE a copy of the entire data
  o Each PE is assigned a portion of the data, split into 128-bit blocks
  o Each block is then encrypted by the PE’s to produce ciphertext blocks
    • Each PE encrypts its blocks in parallel, but the blocks themselves are encrypted sequentially per PE.
  o Data is retrieved by root by MPI_Gather and ciphertext is written to output
Experimental Setup

- Used the 8-core nodes with Infiniband for experimentation
- Ran tests for file sizes of 2kb, 10kb, 50kb, 100kb, 500kb, 1MB, 10MB, 50MB, 100MB
- Utilized 2, 4, 8, 12, 16, 24, 36, 48, and 64 PEs
- Used PolarSSL’s AES library to perform the encryption/decryption itself, and MPI for parallelization
- Each running time was the average of 3 runs
- Times taken were from right before encryption (after data had been distributed) to right after root had gathered data
Results

Analysis of Sequential Running Time

Size of Data (in KB)

Running Time (msec)
Results

Analysis of Parallel Running Time, Fixed 10MB File

Runing Time (msec)

Number of PE's
Results

Analysis of Parallel Running Time, Fixed 10KB File
Analysis of Parallel Running Time, Fixed PE's (64)
Results

Speedup for 8 PE's

- Speedup Factor
- File Size (KB)

The graph shows the speedup factor for 8 processing elements (PE's) against the file size in kilobytes (KB). The speedup increases as the file size grows, indicating improved performance with larger file sizes.
Comparison of Sequential and Parallel Running Times (64 PE's)
Results

Comparisons of Running Times

- Sequential RT
- 4 PE's
- 12 PE's
- 32 PE's
- 64 PE's
Results

Comparison of Costs for 50KB and 10MB Files

Cost for 10MB File

Cost for 50KB File
Conclusions

• Able to clearly see benefits by parallelization
• Extremely low running times for a high number of PE’s, but with added cost
• Encryption/decryption takes the same amount of time, as expected
• Considerable overhead for small files and high PE’s
Future Work

• Fix program so that the ciphertext written by the PE’s is recoverable to plaintext
• Make program more space-efficient by not making n copies of the data for each PE to use
  o In addition, capture the ‘true’ running time of the algorithm by timing entire program
References

doi: 10.1109/ICCET.2010.5486259
URL: http://ieeexplore.ieee.org.gate.lib.buffalo.edu/stamp/stamp.jsp?tp=&arnumber=5486259&isnumber=5485932