AES on GPU

Dmitry Denisenko
Michael Kipper
Josh Slavkin
2009
Agenda

- What is AES?
- AES Description
- AES Modes
- AES on GPU
- Shared Constant Memory
- Data Access Patterns
Advanced Encryption Standard

- Comprises 3 block ciphers: AES-128, AES-192 and AES-256
  - Block size: 128 bits
  - Key sizes: 128, 192 & 256 bits
- Announced on November 26, 2001
- Standard as of May 26, 2002
- The Rijndael cipher developed Joan Daemen and Vincent Rijmen
Algorithm

1. KeyExpansion
2. Initial Round
   - AddRoundKey
Algorithm (cont’d)

3. Rounds
   - SubBytes
   - ShiftRows
   - MixColumns
   - AddRoundKey
Algorithm (cont’d)

3. Rounds
   - SubBytes
   - **ShiftRows**
   - MixColumns
   - AddRoundKey
Algorithm (cont’d)

3. Rounds
   - SubBytes
   - ShiftRows
   - MixColumns
   - AddRoundKey
Algorithm (cont’d)

3. Rounds
   - SubBytes
   - ShiftRows
   - MixColumns
   - AddRoundKey
Algorithm (cont’d)

4. Final Round (no MixColumns)
   - SubBytes
   - ShiftRows
   - AddRoundKey
AES Modes

- AES is inherently block based
- Initialization vector of zero
- Several Modes Exist:
  - Electronic Codebook (ECB)
  - Cipher Block Chaining (CBC)
  - Propogating Cipher Block Chaining (PCBC)
  - Cipher Feedback (CFB)
  - Output Feedback (OFB)
  - Counter (CTR)
**Modes**

- **Electronic Cookbook**
  - Simplest mode, relatively insecure

*Electronic Codebook (ECB) mode encryption*
Modes

- Electronic Cookbook
  - Simplest mode, relatively insecure

Electronic Codebook (ECB) mode encryption
Modes (cont’d)

- **Cipher-block Chaining**
  - Blocks XOR’d with previous ciphertext
  - More secure
  - Not suitable for mass parallelism
Modes (cont’d)

- Propagating Cipher-block Chaining
- Cipher Feedback
- Output Feedback
  - Similar to Cipher-block Chaining
  - Uses different source for XOR/initialization vectors
Modes (cont’d)

- **Counter**
  - Modifies key with successive counter values
  - Most suitable to parallelism
AES on GPU

- Key Expansion on the CPU
  - Uses Rijndael’s key schedule
  - Converts a 128-bit key to 704 bits
- Every function is implemented in both the CPU and GPU
Memory Coalescing

- Each thread is simultaneously working on the same byte
- Concurrent access to data offset by some multiple of the message block size (16 bytes) is undesirable
- Solution: load message into memory for the entire block first
  - Amortize memory access latency over the entire block data space
Shared Constant Memory

- AES uses lots of table lookups to save runtime computation
- There are several 16x16 tables in use:
  - SBox, InvSBox, XTimes2SBox, XTimes3SBox, XTime2, XTime9, XTimeB, XTimeD, XTimeE
  - 9x16x16 = 2,304 bytes
- GPU has 8kB of cache-backed constant memory
  - Use cudaMemcpyToSymbol()
Speedup vs. File Size (256 Threads per Block)
Zoomed In
Speedup vs. Threads (File size in Kilobytes)
Discussion & Conclusion

- Optimal speedup is achieved at 128 threads per block
- 15x speedup over CPU implementation
- Shared constant memory is useful when random access patterns prevent efficient memory usage
- GPU is a viable co-processor for AES