Computing SHA message digest on GPU

Tadas Vilkeliskis

Stevens Institute of Technology

November 30, 2008

Objectives

- Implement SHA-1 hash function on GPU.
- Implement PARHSA-256 hash function on GPU.
- Make performance comparisons.

What is a message digest algorithm? (or hash function)

Definition

An algorithm mapping or translating one sequence of bits (input data) into another, generally smaller set (the hash value) such that a message yields the same hash result every time the algorithm is executed using the same message as input.

- Today's most hash functions use Merkle-Damgård construction method.

Merkle-Damgård method

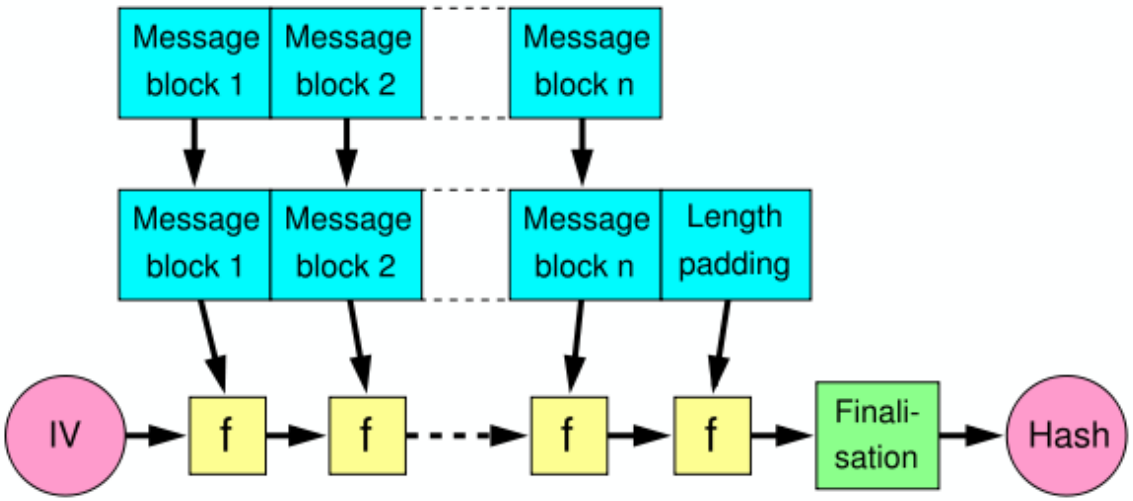

Figure: Constructing hash using Merkle-Damgård method.

SHA-1 hash function

- Based on Merkle-Damgård hash construction method.

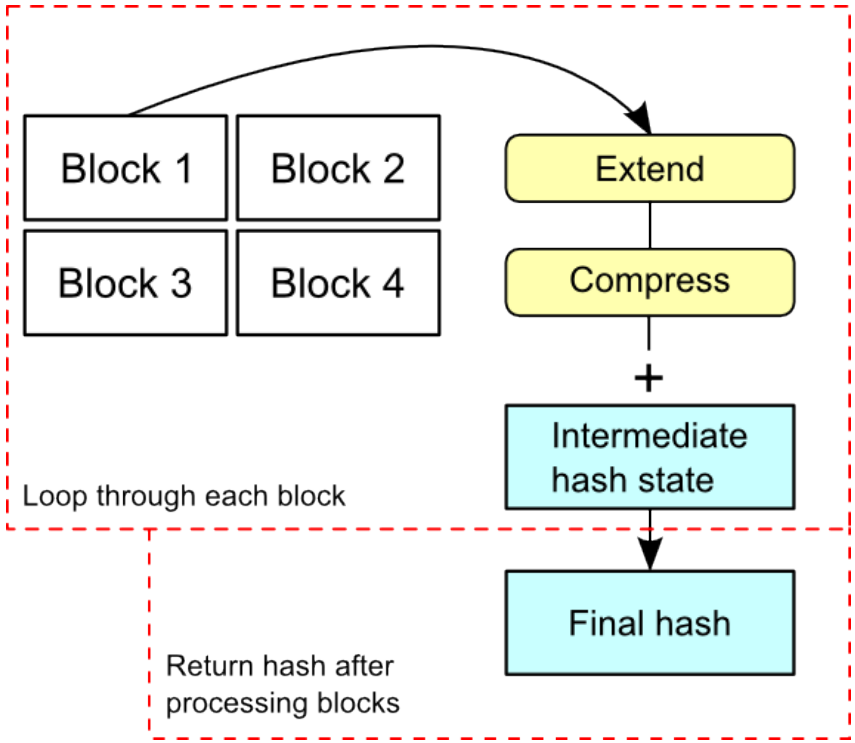

Figure: Typical execution of SHA-1.

Semi-parallel implementation of SHA-1

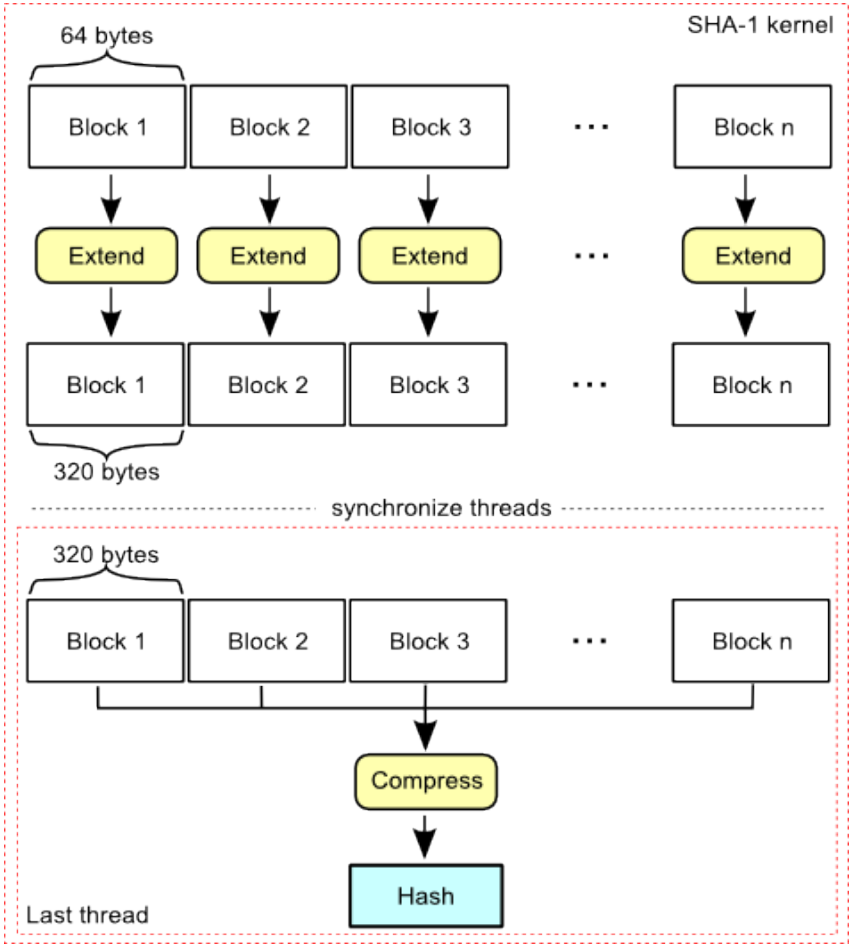

Figure: Semi-parallel implementation of SHA-1

PARSHA-256 algorithm

- Specially designed for parallel execution.
- Uses SHA-256 hash function.
- Uses binary tree of processors.
- Can be implemented in three different ways.

PARSHA-256 algorithm (continued)

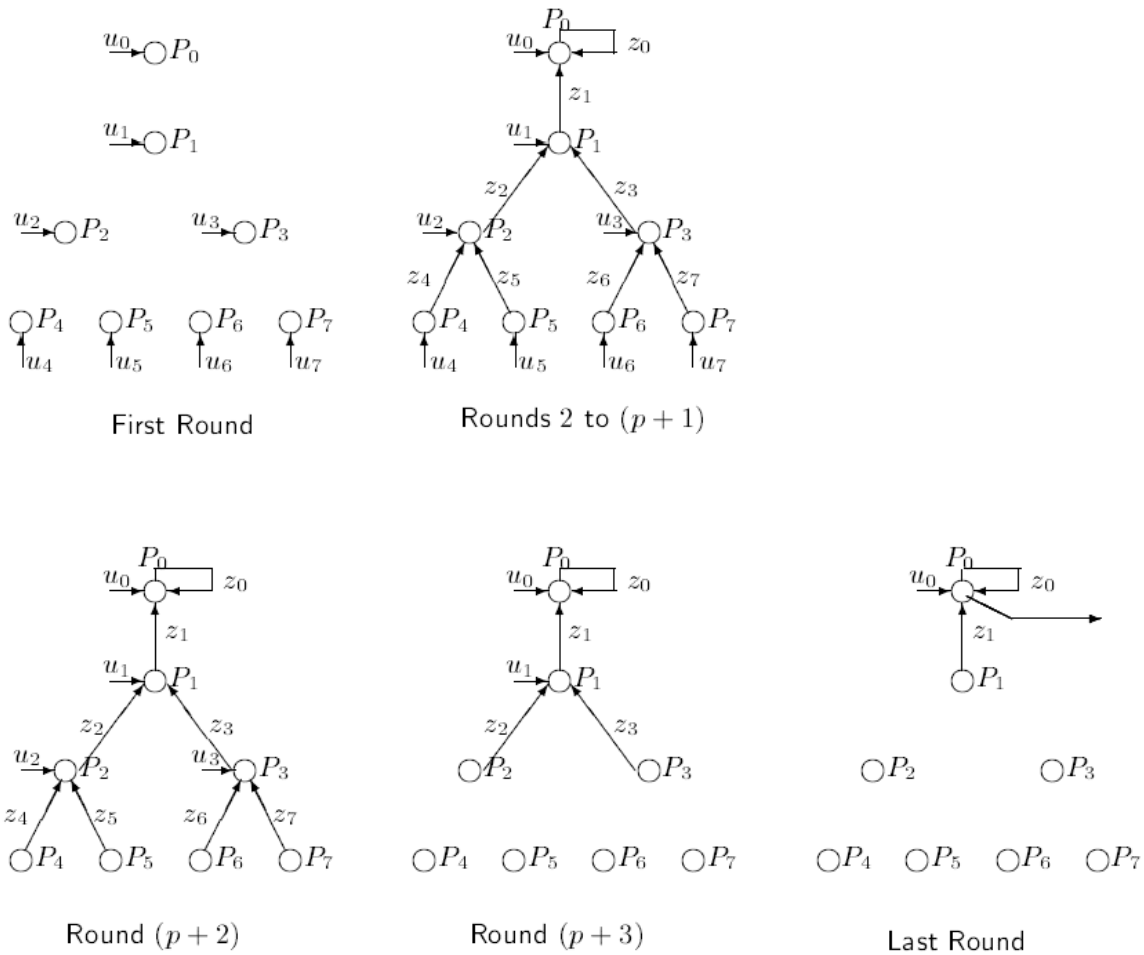

Figure: PARSHA-256 tree

Encountered problems and limitations

SHA-1:

- SHA-1 is iterative hash function.
- Thread synchronization only within thread block.
- Number of threads per thread block is limited by hardware.

PARSHA-256:

- Length of input message.
- Problems with byte ordering.

SHA-1

Unit	Size, bytes	Kernel, ms	cudaMemcpy, ms
CPU	1000	0.012000	
GPU	1000	0.005000	1.664000
CPU	10000	0.104000	
GPU	10000	0.005000	0.322000
CPU	100000	1.036000	
GPU	100000	0.045000	141.783005
CPU	1000000	10.611000	
GPU	1000000	0.383000	1415.932007
CPU	10000000	106.547997	
GPU	10000000	12372.243164	1787.130981

Table: Benchmark test for SHA-1

PARSHA-256

Unit	Size, bytes	Kernel, ms	cudaMemcpy, ms
CPU	1000	25.032000	0.026000
GPU	1000	0.283000	2.404000
CPU	10000	42.157997	0.040000
GPU	10000	0.041000	10.058999
CPU	100000	71.198006	0.355000
GPU	100000	0.054000	76.107979
CPU	1000000	322.458954	3.275999
GPU	1000000	0.086000	731.846802
CPU	10000000	1992.895874	30.029993
GPU	10000000	0.148000	7337.381348
CPU	100000000	17592.455078	259.789185
GPU	100000000	0.668000	73513.625000

Table: Benchmark test for PARSHA-256

Conclusion

- SHA-1 performs well when input size is small.
- PARSHA-256 has great performance no matter what size of input is.
- Data exchange bottleneck for both SHA-1 and PARSHA-256.

Last words

Get code, report and presentation from
http://www.cs.stevens.edu/~tvilkeli/gpgpu_sha_2008/

Any questions?