GPU-Accelerated RSA

CZ.NIC Labs
Author: Karel Slaný <karel.slany@nic.cz>
Messenger: Ondřej Surý <ondrej.sury@nic.cz>

DNS-OARC Workshop 2011
13. 3. 2011
Motivation

- Large-scale data sets:
  - Processing beyond limits of available CPUs
- Dedicated HW needed:
  - multi-core CPU architecture
  - using FPGA based solutions
  - utilization of GPUs
- The speed-up typically relies on the design of the parallel units rather than on their speed.
Recent Trends

Theoretical GFLOP/s

- NVIDIA GPU Single Precision
- NVIDIA GPU Double Precision
- Intel CPU Single Precision
- Intel CPU Double Precision

GFLOP/s

0 250 500 750 1000 1250 1500

- GeForce GTX 480
- GeForce GTX 280
- GeForce 8800 GTX
- GeForce 7800 GTX
- GeForce 6800 Ultra
- GeForce FX 5800
- Westmere
- Bloomfield
- Tesla C2050
- Tesla C1060
- Harpertown
- Woodcrest
- Tesla
- Intel CPU
- NVIDIA GPU

Time:
- Sep-01
- Jan-03
- Jun-04
- Oct-05
- Mar-07
- Jul-08
- Dec-09
GPU Architecture

● Special nature of GPUs
  - More transistors to the 'real' computation
  - The control logic is reduced
  - Only limited code flow control

● Unusual programming models
  - Common GPUs unsuitable for common tasks
  - The GPU is a complex SIMD unit
    - not a real multicore/multithread processor.
GPU Pros and Cons

● Pros
  - Very high peak performance
  - The CPU can process other tasks when the GPU is working.

● Cons
  - Peak performance is nearly unreachable.
  - High CPU-GPU communication latency
  - Reduced communication possibilities between threads
RSA on GPUs - Requirements

- reduce code divergence (conditional branching)
  - all cores process the same execution path
- no communication between threads
  - synchronization slows the code
- reduce the number of expensive arithmetical operation
  - Operations such as division are very expensive in the GPUs.
RSA on the NVIDIA GPU

- RSA cipher implementation
  - the Montgomery exponentiation algorithm in a Residue number system (RNS)
  - the Kawamura's Cox-Rower architecture.
- Based on modular arithmetics
  - The size of the modulo is limited by the ALU width
  - Properly chosen RNS bases allow replacing the expensive modular division operation with a combination of multiplication and addition.
Implementation Details

- The encryption process uses register-width numbers
  - No dependencies such as carry between the numbers
- Arithmetical operations reduced (+, *)
  - Pre-computed value (key dependent)
- The code path is key-dependent
  - All GPU cores execute the same code
    - Code divergence reduction
    - Performance increase (parallelization)
Building the Library

● The library is still in development
  - The RSA1024 is fully supported.
  - An experimental support for RSA2048 and RSA4096
    - Not yet been tested properly
    - Change of the bit-width requires a library re-build

● The build process requires the nVidia CUDA framework
  - Runtime dependent on the CUDA runtime libraries
Library Performance

GF100 (NVIDIA GeForce GTX 480)
- 480 cores @ 1.4 GHz
- Note: New GTX 580 has 512 cores @ 1.5 GHz

CPU (OpenSSL on Intel E5400)
- single core @ 2.7 Ghz

<table>
<thead>
<tr>
<th></th>
<th>GF100 (sig/sec)</th>
<th>CPU (sig/sec)</th>
<th>speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA1024</td>
<td>6150</td>
<td>1720</td>
<td>3.5</td>
</tr>
<tr>
<td>RSA2048</td>
<td>870</td>
<td>280</td>
<td>3.1</td>
</tr>
</tbody>
</table>
Questions?
References

- Jean-Claude Bajard, Laurent-Stephane Didier, and Peter Kornerup. Modular multiplication and base extensions in residue number systems.


- NVIDIA. CUDA C programming guide.