Improved Software Implementation of DES Using CUDA and OpenCL

Noer, Dennis; Engsig-Karup, Allan Peter; Zenner, Erik

Published in:
Proceedings of the Western European Workshop on Research in Cryptology

Publication date:
2011

Improved Software Implementation of DES Using CUDA and OpenCL

D. Noer[*], A.P. Engsig-Karup[*], E. Zenner[†]
Technical University of Denmark, Lyngby, Denmark

This work is concerned with the development of a fast DES bitslice brute-force software tool which utilize consumer Graphics Processing Units(GPUs) and shows improved performance over existing implementations [1]. The use of modern GPUs in high-performance computing is a new trend, where such devices may be useful for offloading computationally intensive tasks for achieving a significant performance boost in comparison with traditional use of general purpose Central Processing Units (CPUs). Programming GPUs are supported by new programming models based on the C language, e.g., the most widely used vendor-specific CUDA [6] and and the industry-wide OpenCL standard [4].

Modern GPUs can be attractive for parallel processing because these architectures by design have hundreds of processing cores and have high on-chip bandwidth close to one order in magnitude larger than modern CPUs. These GPUs have good support for hiding latency in memory transactions through massive multithreading with low context switch overhead. The processing of instructions in the thread contexts is based on the Single Instruction Multiple Data (SIMD) processing paradigm and is therefore suitable for algorithms that can expose a high degree of data parallelism.

The Data Encryption Standard (DES) was chosen as a case study because the block-cipher uses permutations and substitutions of data, rather than the arithmetic calculations which GPUs are known to excel in. The goal is to evaluate the potential of GPUs for this type of application. Furthermore, the DES cipher has a limited 56 bit key space, which has been successfully cracked by [3] on FGPAs. However, FGPAs are much more expensive than GPUs and requires much more programming effort in comparison with CUDA and OpenCL.

A bitsliced [2] implementation of DES was initially considered to be a suitable candidate algorithm for implementation on GPUs. The bitslice method is an emulated SIMD, that utilizes the n-bit registers as a slice of the data vector, making it possible to permute n bits per operation. Our tool is based on highly optimized lookup tables called SubstitutionBOXes (SBOXs) [5]. The nonlinear

[*] Department of Informatics and Mathematical Modelling. The work has been supported by grant no. 09-070032 from the Danish Research Council for Technology and Production Sciences and with resources of GPULAB (http://gpulab.imm.dtu.dk).

[†] Department of Mathematics

SBOXs are converted from a lookup table to pure logic, which on average requires 56 operations. Thus, this is much faster than cutting the distinct key values from the slice and sending them through a lookup table thereby substituting excessive high-latency data transfers with bitwise operations enabling fast processing.

The linear permutations are optimized by finding permutations of permutations and finally reducing them to a single permutation. With permutation reduction in a bitslice implementation, the need to permute data can be circumvented, by letting the ordering of bitslices in the SBOXs function be permuted, which in effect eliminates most movement of data.

An affordable Nvidia GeForce GTX 275 gaming card controlled by the CPU host has been used for the initial development. A nave implementation shows a ten-fold speedup in comparison with the same method running on CPUs. However, this implementation does not utilize the scarce low-latency memory locations (i.e. registers, shared memory and constant memory) on the GPU. Therefore by utilizing these low-latency memories it is possible to reduce the memory fetch time to a fraction, which is found to achieve 13 times the CPU speed.

Further analysis of the model shows that the implementation uses more registers than can be made available per thread on the Nvidia GPUs. The implementation was improved by hard coding static parts of the model thereby reducing the registers per thread to below the hardware limit of 127 registers per thread. This improvement let the entire model rely on using the fast registers, resulting in 18 times the CPU speed. With a number of minor additional improvements the current model achieves 20 times speedup compared to the CPU.

We find that the resulting implementation is able to search up to 680 million keys/s on a GTX 275, which approximately doubles the performance in comparison with previous work [1] on a similar architecture. The major difference between the two models is found in the handling of the bitslices. The model described in [1] relies on precomputed bitslices fetched from constant memory. In the present work, the model programmed in CUDA calculate all bitslices at runtime, which is faster than fetching them from memory because the entire model now fits in the registers.

Further performance improvements will be pursued on AMD GPUs which will requires an implementation of the model in OpenCL. Such an implementation will also be able to execute on general heterogenous hardware setups (including Nvidia GPUs) and can therefore be subject to additional investigations in performance comparison and differences. These findings together with latest results will be presented at the conference.

We remark that a perspective in this work is that this type of DES bruteforcer can be distributed over any number of GPUs, thus supplying organized groups the power of a super computer. It is doubtful that any organization have the computing power to exhaust the key space of modern ciphers, but relative short password searches could theoretically be conducted successfully on distributed GPUs.

References

[1] Giovanni Agosta, Alessandro Barenghi, Fabrizio De Santis, and Gerardo Pelosi. Record setting software implementation of des using cuda. In *Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations*, ITNG '10, pages 748–755, Washington, DC, USA, 2010. IEEE Computer Society.

[2] Eli Biham. A Fast New DES Implementation in Software. In *Proceedings of the 4th International Workshop on Fast Software Encryption*, FSE '97, pages 260–272, London, UK, 1997. Springer-Verlag.

[3] Electronic Foundation. *Cracking DES*. O'Reilly Media, 1998.

[4] Khronos Group. The OpenCL specification V1.0.48 , 2010.

[5] Matthew Kwan. Reducing the Gate Count of Bitslice DES, October, 2000. IACR Eprint archive.

[6] NVIDIA. NVIDIA CUDA C Programming Guide Version 3.2, 2010.