2010 Second International Conference on Communication Software and Networks

AES Encryption Algorithm Based on the High Performance Computing of GPU

Fei Shao, Zinan Chang, Yi Zhang
Department of Information Technology
Jinling Institute of Technology
Nanjing, China
e-mail: shaofei@jit.edu.cn, changzn@jit.edu.cn, zhangyi@jit.edu.cn

***Abstract*—The encrypting time of traditional AES algorithm is too long to meet the need of fast encryption. For this point, the high-performance computing capability of Graphic Processing Unit has become the hot issue of research. This paper proposes that AES algorithm is improved by use of GPU's high-performance computing capability and compared with that using CPU. And AES encryption algorithm base on high performance computing of GPU is also completed. The experiment shows that the computing speed of AES encryption algorithm based on GPU is obviously higher than AES encryption algorithm based on CPU, thus the encryption efficiency is increased.**

Keywords-graphic processing unit; style of stream programming; AES algorithm

I. INTRODUCTION

Along with the popular use of computer, the information security has become one of the problems which need to be solved urgently. Recently the raw speed highly data-parallel nature and rapidly expanding programmability of graphics processing units make them an attractive platform for general purpose computation. The AES algorithm is currently the standard block-cipher algorithm that has replaced the Data Encryption Standard (DES).The encryption time of traditional AES algorithm is too long, which can't meet the need of fast encryption. For this point, GPU (Graphic Processing Unit) has higher data transfer bandwidth and better parallelism and its high-performance computing capability becomes the hot issue of research. AES encryption algorithm base on high performance computing of GPU is completed by Cg language. And the key technologies are researched and analyzed, and the comparison result of the efficiency of the algorithm is displayed. It has been proven that the computation speed of AES algorithm based on GPU significantly faster than AES algorithm based on CPU.

II. AES ALGORITHM

In cryptography, the Advanced Encryption Standard (AES) is an encryption standard adopted by the U.S. government.

Back in 1997 the National Institute of Standards and Technology (NIST) made a public call for new cipher algorithms that could replace the DES. A rough summary of the requirements made by NIST for the new AES were the following:

- Symmetric-key cipher
- Block cipher
- Support for 128-bit block sizes

Support for 128-, 192-, and 256-bit key lengths finally in October 2000, the Rijndael algorithm was chosen as the basis for the new standard encryption algorithm. The original Rijndael algorithm also supported both fixed-size and variable-size bit cipher blocks. However, currently the Federal Information Processing Standards specification for the AES algorithm supports only the fixed-size, 128-bit blocks.

The standard comprises three block ciphers, AES-128, AES-192 and AES-256, adopted from a larger collection originally published as Rijndael. Each AES cipher has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively. The AES ciphers have been analyzed extensively and are now used worldwide, as was the case with its predecessor, the Data Encryption Standard (DES).

In AES algorithm, the encryption step uses a key that converts the data into an unreadable ciphertext, and then the decryption step uses the same key to convert the ciphertext back into the original data. This type of key is a symmetric key; other algorithms require a different key for encryption and decryption.

Generally speaking, the strength of an encryption by product ciphers can be heightened by increasing the number of rounds used to process the data. The AES standard specifies that the number of rounds is determined by the length of the cipher key, as shown in Table 1.

TABLE I. KEY LENGTH AND THE NUMBER OF ROUNDS

Key Length	Number of Rounds (Nr)
AES-128	10
AES-192	12
AES-256	14

The registers for state_in and state_out consist of four components of 32-bit integers each, giving us the standard AES block size of 128 bits for processing.The precise steps involved in the algorithm can be seen in Fig. 1.

In Initial Round, AddRoundKey was realized.And in Rounds; mainly it divided into the following steps:

978-0-7695-3961-4/10 $26.00 © 2010 IEEE
DOI 10.1109/ICCSN.2010.124

- SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
- ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
- MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column
- AddRoundKey—each byte of the state is combined with the round key; each round key is derived from the cipher key using a key schedule.
- In Final Round consists of SubBytes, ShiftRows and AddRoundKey.

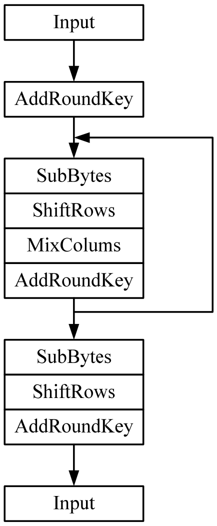

Figure 1. AES encryption algorithm flow chart.

III. AES Encryption Algorithm Base on the High Performance Computing of GPU

In 1999, NVIDIA Company first proposed the concept of GPU, when it recommended graphics processor, GeForce256 graphics processing chip. The initial GPU was to carry on the 3D graph acceleration. With the specificity of GPU and the continuous improvement of the architecture, until now, GPU has stronger performance and a more perfect programmable construction compared to the first several generations, meanwhile GPU's large-scale floating number parallel computing has also received the widespread attention.

A. *Style of Stream Programming*

In the stream programming model, all data is represented as a stream, which we define as an ordered set of data of the same data type. That data type can be simple (a stream of integers or floating-point numbers) or complex (a stream of points or triangles or transformation matrices). While a stream can be any length, we will see that operations on streams are most efficient if streams are long (hundreds or more elements in a stream). Allowed operations on streams include copying them, deriving substreams from them, indexing into them with a separate index stream, and performing computation on them with kernels.

A kernel operates on entire streams, taking one or more streams as inputs and producing one or more streams as outputs. The defining characteristic of a kernel is that it operates on entire streams of elements as opposed to individual elements.

Some of the main production site code

```
Encrypt:
  MOV.U  arg0, enc_key[0];
  CAL  unpack_state_in;
  SUB.S  cnt.x, num_round.x, 1;
  MOV.S  cnt.y, 1;
  REP.S  cnt.x;
    CAL  sub_bytes_shift_rows;
    MOV.U  arg0, enc_key[cnt.y];
    CAL  mix_columns_add_round_key;
    ADD.S  cnt.y, cnt.y, 1;
  ENDREP;
  CAL  sub_bytes_shift_rows;
  MOV.U  arg0, enc_key[cnt.y];
  CAL  pack_state_out;
  RET;
```

B. *The Results of Experiments for Comparison*

Now that we have a working AES implementation, let us measure the performance of GPU-based encryption. The decryption is omitted because it performs the same as the encryption in the AES algorithm. Our tests were performed on a test machine with the following specifications:

CPU: Pentium(R) Dual-Core E5200

Video: GeForce 9400 GT

The experimental result is as shown in Table 2. Under the premise of running the same encryption original text, the computing speed of AES algorithm based on GPU is obviously faster than AES's dealing algorithm based on GPU.

TABLE II. Key Length and the Number of Rounds

The time of AES algorithm base on CPU	**The time of AES algorithm base on GPU**	**The rate of acceleration**
1.16s	1s	1.16 times

IV. Conclusion

The openssl command has benchmarks for determining the speed of their cipher algorithms on the CPU. We measured the speed of these CPU-based encryptions on the same test machine, and we got results using the same AES algorithm. Compared to the speed of the CPU-based implementations, GPU-based encryption is thus almost the same speed for the vertex program, and about 1.16 times faster for the fragment program.

References

[1] Matt Pharr and Randima Fernando, GPU Gems 2:Programming Techniques for High-Performance Graphics and General-Purpose Computation, Addison-Wesley Professional, 2005-03-03.

[2] Randima Fernando, GPU Gems 3, Addison-Wesley Professional, 2007-08-12.

[3] Owens John D, “Computer Graphics on a Stream Architecture,” Ph.D. Thesis, Stanford University, November 2002.

[4] Nicolas Courtois and Josef Pieprzyk, “Cryptanalysis of Block Ciphers with Overdefined Systems of Equations,” pp. 267–287.

[5] Joan Daemen and Vincent Rijmen, “The Design of Rijndael: AES-The Advanced Encryption Standard,” Springer-Verlag, 2002.

[6] Patterson and David A, “Latency Lags Bandwidth,” Communications of the ACM, vol. 47, 2004, pp. 71–75.