5ZC-06

NVIDIA Tesla K80 上での共通鍵暗号 AES の並列処理

Parallelization of AES Cryptography on NVIDIA Tesla K80

小梁川 龍† 吉田 明正†
Ryu Koyanagawa Akimasa Yoshida

1 はじめに

近年，データ通信におけるデータプライバシーの問題が重要視されており，データ保護が必要とされている．このような用途に対処するため，共通鍵暗号 AES が広く用いられるようになっている．AES については，専用ハードウェアによる暗号化のみならず，汎用 GPU による暗号化の研究も行われてきた [1][2]．本研究では，NVIDIA Tesla K80 上において，大規模データを対象とした AES 暗号化プログラムを CUDA により実装している．本稿では，デバイスメモリへのデータ配置，および，Tesla K80 のシェアードメモリを有効活用し，デバイスメモリへのアクセス時間を軽減して，高速化を実現する手法を提案する．Tesla K80 上で行った性能評価の結果，高い実効性能が得られ，提案手法の有効性が確認された．

2 共通鍵暗号 AES

AES は平文に対して 16Byte ごとのステートという処理単位に区切り，暗号処理を行う．暗号処理では，入力されたステートと共通鍵から計算したラウンドキーを用いてラウンド処理を既定の回数繰り返す．このラウンド処理には SubBytes，ShiftRows，MixColumns，AddRoundKey と呼ばれる処理が含まれる．

AES には暗号利用モードという概念があり，標準化されている暗号利用モードは以下のようなものがある．

ECB モードは最も基本的な暗号利用モードである．各ステートを独立に暗号処理するため，並列化は容易であるが選択平文攻撃の脆弱性があり使用が推奨されていない．

CBC モードは前後のブロックに依存関係を持たせることで ECB モードの脆弱性を克服した暗号利用モードである．ブロック間に依存性があるため，一般的に並列処理を行うことができないが複数の初期ベクトルを用いることで並列処理が可能である [3]．

CTR モードは CBC モードと異なり乱数をもとにしたカウンタを用いて ECB モードの脆弱性を克服した暗号利用モードである．また，暗号化関数のみで暗号化と復号が可能であり，ブロックごとに完全並列処理を行うことができる [4]．

3 GPU 上での共通鍵暗号 AES の高速化

本章では，AES-256-CTR の CUDA プログラムにおけるループアンローリング，および，GPU のシェアードメモリを活用した高速化について述べる．

3.1 ループアンローリングによる高速化

ループアンローリングを行うとプログラムのコードサイズが大きくなるが，ループのオーバーヘッドが削減され，実質的な実行命令の削減により実行速度が向上する．図 1 は，ループアンローリングを行う前後のコードである．本研究では，図 1(b) のように AES の暗号化を行うカーネル関数内部でループアンローリングを行い CUDA プログラムのリストラクチャリングを行う．

```
01: __global__ void kernel1(){
02:   //暗号化および復号化処理を行うループ
03:   for(i = 0; i < stride; i++){
04:     //対象のステートの先頭位置を計算
05:     i_ = (index + (i * B * T)) * 16;
06:     暗号化処理;
07:   }
08: }
```

(a)ループアンローリング前のコード

```
01: __global__ void kernel1(){
02:   //暗号化および復号化処理を行うループ
03:   for(i = 0; i < stride; i+=2){
04:     //対象のステートの先頭位置を計算
05:     i_ = (index + (i * B * T)) * 16;
06:     暗号化処理;
07:     //対象のステートの先頭位置を計算
08:     i_ = (index + ((i+1) * B * T)) * 16;
09:     暗号化処理;
10:   }
11: }
```

(b)ループアンローリング後のコード

図 1 ループアンローリング適用前後のコード．

表 1 性能評価に用いる NVIDIA Tesla K80．

構成要素	仕様
CPU	Intel Xeon E5-2680 v3，2.5GHz，12 コア× 2
メモリ	64GB
GPU	NVIDIA Tesla K80 × 2 個 (GK210 × 4)
OS	CentOS 6.9
処理系	GCC 4.4.7，CUDA Toolkit 9.1

3.2 シェアードメモリを活用した高速化

GPU のシェアードメモリはブロック内のスレッドが共用することができ，グローバルメモリと比較して非常に高速であるため，プログラム内で頻繁に呼び出される値をシェアードメモリに配置することで処理速度が向上する [5][6]．本研究では S-box とラウンドキーをシェアードメモリに配置することで処理時間の高速化を行った．S-box は非線形変換を行うためのデータの配列である．実装した AES プログラムにおいては 1Byte の入力に対し 1Byte を出力し，ステートの 16Byte それぞれを変換して新しいステートを作成するために用い，ラウンドキーの計算や SubBytes 内で利用する．ラウンドキーは共通鍵から計算され，AES の各ラウンド処理に用いられる．

4 NVIDIA Tesla K80 上での AES の性能評価

本章では，GPU 上でのループアンローリングおよびシェアードメモリを活用した高速化による AES の並列プログラムの性能評価について述べる．

4.1 性能評価環境

本性能評価ではループアンローリングおよびシェアードメモリを活用した高速化を行った AES-256-CTR プログラムを用いて，64MB のデータの暗号化および復号化の性能評価を行う．評価に用いたマシンは表 1 に示した NVIDIA Tesla K80 である．性能評価方法としては，

†明治大学大学院先端数理科学研究科ネットワークデザイン専攻
Network Design Program, Graduate School of Advanced Mathmatical Science, Meiji University

表 2　AES の 1 ブロック 1 スレッド実行時間.

CUDA プログラム	実行時間 [s]
original	166.84
unrolling	149.17
shared	118.78

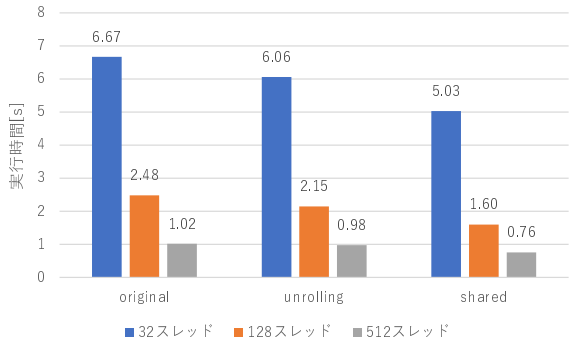

図 2　Tesla K80 上での AES の実行時間 (1 ブロック).

1 ブロック及び 32 ブロックにおいて，32，128，512 スレッド毎にプログラムを実行し実行時間を計測した．1 ブロックでの AES-CTR の実行時間を図 2 に示す．ここで，1 ブロック 1 スレッドでの実行時間は表 2 のとおりである．32 ブロックでの AES-CTR の実行時間を図 3 に示す.

4.2　GPU 上での AES の並列処理による性能評価

まず，1 ブロック実行の場合，表 2 の original に示すように 1 スレッド実行時間は 166.84[s] となり，スレッド数を増やすと図 2 に示すように，32 スレッドで 6.67[s]，128 スレッドで 2.48[s]，512 スレッドで 1.02[s] に実行時間が短縮された.

次に，32 ブロック実行の場合，図 3 に示すように，32 スレッドで 0.62[s]，128 スレッドで 0.37[s]，512 スレッドで 0.34[s] に実行時間が短縮された．512 スレッドでは，1 ブロック 1 スレッド (original) 比で 491 倍の速度向上が得られている.

これらの測定結果から NVIDIA Tesla K80 上での AES-256-CTR の並列処理において高い実効性能が得られることが確認された.

4.3　ループアンローリングによる高速化の性能評価

CUDA プログラムにループアンローリングを適用した 1 ブロック実行の場合，表 2 の unrolling に示すように 1 スレッド実行時間は 149.17[s] となり，スレッド数を増やすと図 2 に示すように，32 スレッドで 6.06[s]，128 スレッドで 2.15[s]，512 スレッドで 0.98[s] に実行時間が短縮された.

次に，32 ブロック実行の場合，図 3 に示すように，32 スレッドで 0.56[s]，128 スレッドで 0.36[s]，512 スレッドで 0.33[s] に実行時間が短縮された．512 スレッドでは，1 ブロック 1 スレッド (original) 比で 506 倍の速度向上が得られている.

これらの測定結果からループアンローリングによる高速化は，暗号化および復号化において有効であることと確認された.

4.4　シェアードメモリを活用した高速化の性能評価

シェアードメモリを利用した高速化を行った 1 ブロック実行の場合，表 2 の shared に示すように 1 スレッド実行時間は 118.78[s] となり，スレッド数を増やすと図 2 に示すように，32 スレッドで 5.03[s]，128 スレッドで 1.60[s]，512 スレッドで 0.76[s] に実行時間が短縮された.

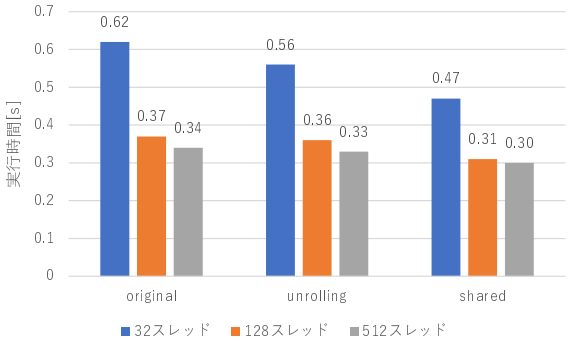

図 3　Tesla K80 上での AES の実行時間 (32 ブロック).

次に，32 ブロック実行の場合，図 3 に示すように，32 スレッドで 0.47[s]，128 スレッドで 0.31[s]，512 スレッドで 0.30[s] に実行時間が短縮された．512 スレッドでは，1 ブロック 1 スレッド (original) 比で 556 倍の速度向上が得られている.

これらの測定結果からシェアードメモリを活用した高速化は，暗号化および復号化において有効であることが確認された.

5　おわりに

本稿では，NVIDIA Tesla K80 上での共通鍵暗号 AES(AES-256-CTR) の並列処理における，ループアンローリングとシェアードメモリ活用による高速化手法を提案した.

AES による暗号化および復号化を行う CUDA プログラムを実装し，ブロック数とスレッド数を変えて実行時間の計測を行った．CUDA 実装したオリジナルプログラムでは 32 ブロック 512 スレッド実行時に，1 ブロック 1 スレッド比で 491 倍の速度向上（1.58[Gbps]），ループアンローリングによる高速化では 32 ブロック 512 スレッド実行時に 506 倍の速度向上（1.63[Gbps]），シェアードメモリ活用による高速化では 32 ブロック 512 スレッド実行時に 556 倍の速度向上（1.79[Gbps]）が得られた.

以上の結果から NVIDIA Tesla K80 上での共通鍵暗号 AES の並列処理においてループアンローリングおよびシェアードメモリを活用した高速化の有効性が確認された.

参考文献

[1] Nhat-Phuong Tran, Myungho Lee, Sugwon Hong, Seung-Jae Lee. Parallel Execution of AES-CTR Algorithm Using Extended Block Size, IEEE International Conference on Computational Science and Engineering, 2011.

[2] 菊池丞太，山口実靖．CUDA による AES 暗号処理についての考察，情報処理学会，S0731A，2015.

[3] 福永武志，平木敬．AES-CTR モードを用いたセキュアな高速シングルストリーム通信，電子情報通信学会，CPSY2015-27，DDC2015-23，2015.

[4] 神永正博，山田聖，渡邊高志．Java で作って学ぶ暗号技術，森北出版株式会社，2008.

[5] Jianwei Ma, Xiaojun Chen, Rui Xu, Jinqiao Shi. Implementation and Evaluation of Different Parallel Designs of AES Using CUDA, IEEE Second International Conference on Data Science in Cyberspace, 2017.

[6] Keisuke Iwai, Naoki Nishida, Takakazu Kurokawa. Acceleration of AES encryption on CUDA GPU, International Journal of Networking and Computing Volume2, Number1, 131-145, 2012.