L-041

メニーコア・コプロセッサによる複数ファイルの並列共通鍵暗号実装
Parallel AES Implementation on Manycore Coprocessor for Plural Files

白勢 政明 † 吉田 努 †
Masaaki Shirase Tsutomu Yoshida

あらまし
マルチコア CPU やメニーコア・コプロセッサを使って，電子書籍コンテンツのような数百の複数ファイルを並列に AES 暗号化するためのプログラムを実装し，各ファイルサイズを 128B〜4MB とした時の実行時間を評価した．なお，暗号化すべきファイルは 1 つのディレクトリに置かれており各暗号化ファイルも別の 1 つのディレクトリに格納することを想定し，暗号実装の容易性のために OpenSSL 暗号ライブラリを活用した．著者の一人は以前に並列実装基盤として OpenMP を用いて同様の研究を行った際，ファイルサイズが大きい (256KB 以上) 場合スレッド数を大きくして暗号処理を行うと，平均実行時間と比較して秒オーダーの異常遅延が発生することが判明した [13]．今回は，並列実装基盤として pthread を使用し，ベースとなる暗号化プログラムから malloc を無くすように改良したことにより，全体として処理速度を向上でき，また異常遅延を防ぐことができた．なお，これらの改善が pthread によるものか malloc 除去によるものかは未確認であるため，今後の課題としたい．

1. はじめに

本報告の目的は，1 つのディレクトリに共通鍵暗号 AES で暗号化すべきファイルが複数入っている場合，マルチコア CPU やメニーコア・コプロセッサを用いてファイル単位で並列的に暗号化・復号を行う場合の実装評価を行うことであり，その経緯は次の 4 つの背景に関係する．

1.1 背景 1: プロセッサのマルチコア化・メニーコア化

トランジスタの微細加工技術の進展のため，1 つのプロセッサに搭載できるトランジスタ数 (ゲート数) は，2 年で 2 倍になる，というムーアの法則が知られており，現在も継続している [4]．トランジスタの微細化は，2005 年ぐらいまではプロセッサのクロック周波数の向上の要因であったが，それ以降はプロセッサのコア数の増大の要因となっている [15]．実際に，近年のパソコン用 CPU においては，コア数が 2〜12 程度のマルチコア CPU が一般的になっており，2013 年には Intel Xeon Phi のようなコア数が 60 程度のメニーコア・コプロセッサの流通が開始した．また，Graphic Processing Unit (GPU) においては，コア数が 2,000 以上のものも存在する．

微細化により CPU のクロック周波数の上昇がなされた時期においては，新しい CPU はプログラムの変更なしでそのプログラムの処理速度の向上を達成させたが，微細化が CPU コア数増大をもたらす現在は，プログラムの処理速度の向上には (コンパイラに依るか人力に依るかは別にして) 並列プログラムへの書き換えが必要である．CPU メーカは，少ない負担で並列化コードを生成できるコンパイラを開発している [5]．

1.2 背景 2: メニーコア・コプロセッサ Intel Xeon Phi

2013 年に流通が開始した Intel Xeon Phi は，57 から 61 のコアを持つメニーコア・コプロセッサであり，インテル・メニー・インテグレーテッド・コア (MIC) アーキテクチャと呼ばれるアーキテクチャを採用している [15]．インテル MIC アーキテクチャのためのコードは，C 言語や Fortran で記述したプログラムに pthread や OpenMP 等による並列処理の指示を追加し，Intel コンパイラで “-mmic” オプションでコンパイルすることで，生成される．また，インテル MIC アーキテクチャは x86 アーキテクチャをベースにしているため，並列化が不要ならば大抵の x86 アーキテクチャのコードを変更無しで Intel Xeon Phi で実行することができる ‡．このように，既存のプログラムやアセンブリコードから Intel Xeon Phi 用コードを容易に生成する環境が整っている．

1. 2 つのディレクトリ名を引数とし，
2. 1 つ目のディレクトリに入っているファイルすべてに対して 128 ビット AES により暗号化または復号し，
3. その結果すべてを 2 つ目のディレクトリに出力する

表 1: 作成するプログラムの動作

1.3 背景 3: 電子図書の著作権保護

近年急速に市場が拡大している電子図書のフォーマットの 1 つである EPUB[2] では，コンテンツはページ単位または章単位で xhtml ファイル化され，それらは 1 つのディレクトリに置かれる．1 つのコンテンツの xhtml ファイル数は数十から数百となる．

EPUB のための基準システムの開発を目的とするプロジェクト Readium において，電子書籍のための軽量な著作権保護技術の実装の必要性から Readium Lightweight Content Protection (LCP) が提案された [12]．Readium LCP は，販売者は電子書籍コンテンツを，ファイル単位で，ユーザ毎の共通鍵で 128 ビット (以上の)AES で暗号化することを推奨している．

1.4 背景 4: OpenSSL

OpenSSL は SSL/TLS プロトコルのためのソフトウェアや共通鍵暗号や公開鍵暗号，ハッシュ関数アルゴリズムのライブラリを提供している [14, 11]．OpenSSL 暗号ライブラリを用いることで，様々な種類の暗号機能を実システムに容易に実装することができる．

1.5 本報告の寄与

本報告では，先行報告 [13] と同じく，表 1 のように動作する並列プログラムを OpenSSL ライブラリを利用して作成した．[13] と本報告との違いは使用する並列実装

†公立はこだて未来大学　情報アーキテクチャ学科

‡命令セットが完全には一致していないため，x86 アーキテクチャのコードを Intel Xeon Phi で実行できない場合もある．例えば，アセンブリで与えられた GMP ライブラリ [3] の関数のいくつかは，アセンブリを書き換えてコンパイルし直さなければならない．

Intel Xeon Phi サーバ：	Express5800HR120a–1
ホスト CPU :	Intel Xeon E5-2640(6 コア，2.5GHh)×2
コプロセッサ :	Intel Xeon Phi 5110P(60 コア，1.05GHz)×1
コンパイラ：	インテル C++ Studio XE 2013 Linux 版

表 2: 実装実験環境

パラメータの種類	パラメータの値
使用プロセッサ	Intel Xeon 1/2，Intel Xeon Phi
ファイル 1 つのサイズ (B)	128，256，512，1K，2K，4K，8K，16K，32K，64K，128K，256K，512K，1M，2M，4M
スレッド数	1，2，4，10，20，50，100

表 3: 実装実験のパラメータ

	Intel Xeon 2				Intel Xeon Phi			
ファイルサイズ \ スレッド数	1	2	4	10	1	2	4	10
128〜256B	0	0	0	0	0	0	0	0
512KB	0	0	0	3	0	0	0	0
1MB	0	0	0	6	0	0	0	0
2MB	0	0	3	10	0	0	0	0
4MB	0	0	2	6	0	0	0	0

表 4: ファイル数 10 での異常遅延発生回数 (100 回中)

	Intel Xeon 2					Intel Xeon Phi						
ファイルサイズ \ スレッド数	1	2	4	10	20	1	2	3	10	20	50	100
128〜128KB	0	0	0	0	0	0	0	0	0	0	0	0
256KB	0	0	0	0	6	0	0	0	0	0	0	0
512KB	0	0	0	6	21	0	0	0	0	0	0	0
1MB	0	0	0	23	25	0	0	0	0	0	0	0
2MB	1	1	1	33	34	0	3	0	0	0	0	0
4MB	1	0	1	12	23	0	6	0	0	0	1	1

表 5: ファイル数 100 での異常遅延発生回数 (100 回中)

基盤が異なっている点あり，[13] では Intel Xeon Phi へのプログラミング手法を記述している参考文献 [15] が紹介している OpenMP を用いたが，本報告では pthread を使用した．また，本報告の実装では暗号化関数から malloc を除去した．

次に，表 2 のような実装環境 § において，表 3 のようなパラメータに対して，このプログラムが暗号化に要する時間を評価した．Intel Xeon を用いる実装評価において，[13] では特に使用する CPU 数を指定しなかったため 2 個の Intel Xeon が使用されたが，今回は Intel Xeon を 1 つと 2 つの場合で評価した．

本報告での実装では，[13] と比較して処理速度が向上し，異常遅延 ¶ の発生も抑制できた．なお，これらの改善が pthread によるものか malloc 除去によるものかは未確認であるため，今後の課題としたい．

1.6 表記について

本報告の実装結果において，**Intel Xeon 1** は 1 つの Intel Xeon の使用を，**Intel Xeon 2** は 2 つの Intel Xeon の使用を意味するとする．

2. AES について

共通鍵暗号はブロック暗号とストリーム暗号に大別される．Readium LCP で使用が推奨されている AES はブロック暗号の一種であり，広く普及している．AES のブロックサイズは 128，192，256 ビットである．一般にブロック暗号では，ブロック長より長い平文を暗号処理する場合は，以下で例を挙げる各モードによって，平文をブロック長のブロックに分割し暗号処理を行う．

Electronic Code Book (ECB) モードは，各ブロックを独立に暗号処理する．ECB モードではブロック単位で並列的に暗号処理を行うことができるが，値が同じ平文ブロックは暗号文も同じブロックとなるため，安全でない場合がある．

CounTeR (CTR) モードは，カウンターの値をブロック暗号で暗号化し，それを鍵ストリームとして平文との XOR をとる．CTR モードは暗号処理の並列化が可能である．

Cipher-Block Chaining (CBC) モードは，前後のブロックの暗号文に依存関係を持たせることで ECB モードの欠点を克服し，最も利用されている暗号モードである．但し並列的に暗号化することはできない．

3. 関連研究

メニーコア CPU や GPU を利用しての AES の並列化の研究は，

(1) 1 つのファイルを対象とした AES の並列実装

(2) 複数ファイルを対象とした AES の並列実装

に分類できる．

3.1 1 つのファイルを対象とした AES の並列実装

一般に AES の並列暗号化の研究では，1 つのファイルを対象とした AES の並列実装が扱われ，

1. 1 回の AES 暗号化の高速化のためのラウンド処理の並列処理

2. 1 つのファイルの AES 暗号化の高速化のためのブロック単位での並列処理

に大別される．例えば，[8] や [7] は (1) に関する研究で，[6] や [9]，[10] は (2) に関する研究である．

3.2 複数ファイルを対象とした AES の並列実装

電子書籍のコンテンツの暗号化のように複数ファイルを同時に暗号処理を行うという状況があるため，「複数ファイルを対象とした AES の並列実装」も意義のある方針であると著者等は考えており，本報告及び先行報告 [13] はこの場合を扱う．

[13] の特徴として，表 1 のようなプログラムを並列実装基盤として OpenMP を使用して作成し，基となる暗号化関数では暗号化されるファイルサイズと同じサイズのメモリ領域を確保するために malloc を使用した．そして，表 2 のような実装環境において表 3 のようなパラメータに対して (Intel Xeon 1 を除く) 100 回実装評価を行った結果，同じパラメータでも平均より暗号処理に数秒以上かかる**異常遅延**が発生が生じる場合があった．各パラメータに対する異常遅延発生回数は表 4(ファイル数 10 の場合) や表 5(ファイル数 100 の場合) のようになった．ファイルのサイズが大きいほど異常遅延となる場合が多くなり，また Intel Xeon Phi より Intel Xeon で処理している時の方が異常遅延となる頻度がずっと高い．

§[15] では Intel Xeon はマルチコア CPU，Intel Xeon Phi はメニーコア・コプロセッサと位置づけている

¶異常遅延については 3.2 節参照．

なお，この実装結果は本報告の実装結果と併せて 4.3 節と付録に掲載する．

[13] での実装結果では，であるが，表 3 のどのパラメータに対しても，暗号処理速度は Intel Xeon Phi の方がずっと遅かった．Intel Xeon と Intel Xeon Phi のクロック周波数とコア数を考慮すると，この結果は予想外であった．

4. 本報告の実装

本報告では，OpenSSL 暗号ライブラリを利用し，並列実装基盤 pthread を使用し，表 1 のうように動作する AES 並列プログラムを実装し，表 2 の実装環境で表 3 のパラメータに対して実装評価を行った．

4.1 pthread を利用しての OpenSSL の使用

OpenSSL でのマルチスレッドの扱い方は [14] が，並列実装基盤 pthread については [1] が詳しいが，ここで簡単にこれらを紹介する．

[14] によると OpenSSL でマルチスレッドを扱う場合，不可分に実行するデータ構造体への複数アクセスを阻止するために，ミューテックスと呼ばれる相互に排他的なロックによりそのデータ構造体を保護する必要がある．そのためにロックとアンロックの方法を指定するコールバック関数やスレッド ID の取得方法を指定するコールバック関数が必要であるが、OpenSSL ではそれらは用意されておらず，実装者がプラットフォームに合わせて用意する必要がある．

pthread で重要な関数に `pthread_creat()` と `pthread_join()` がある．`pthread_creat()` は，スレッド ID，スレッドの属性，スレッドで実行する関数のポインタ，その関数の引数のポインタを引数として，スレッドを開始させる．`pthread_join()` は，スレッド ID(とスレッドステータス) を引数とし，引数で指定したスレッド ID のスレッドが終了するのを待つ．

4.2 プログラムについて

4.2.1 ロックに関するプログラム

ロックとアンロックの方法を指定するコールバック関数 `lock_callback()` とスレッド ID の取得する方法を指定するコールバック関数 `thread_id()` は

```
static void lock_callback(int mode, int
  type, char *file, int line)
{
  if (mode & CRYPTO_LOCK)
    pthread_mutex_lock(&(mutex_buf[type]));
  else
    pthread_mutex_unlock(&(mutex_buf[type]));
}

static unsigned long thread_id(void)
{
  return (unsigned long)pthread_self();
}
```

とした．

ミューテックスの初期化とコールバック関数を登録する関数 `init_locks()` は

```
static void init_locks(void)
{
  int i;
  mutex_buf=(pthread_mutex_t *)
    OPENSSL_malloc(CRYPTO_num_locks() *
    sizeof(pthread_mutex_t));
  for (i = 0; i < CRYPTO_num_locks(); i++)
    pthread_mutex_init(&(mutex_buf[i]),
      NULL);
  CRYPTO_set_id_callback((unsigned long
    (*)())thread_id);
  CRYPTO_set_locking_callback((void (*)())
    lock_callback);
}
```

とし，コールバック関数とミューテックスを解放する関数 `clear_locks()` は

```
static void clear_locks(void)
{
  int i;
  CRYPTO_set_locking_callback(NULL);
  for (i = 0; i < CRYPTO_num_locks(); i++)
    pthread_mutex_destroy(&(mutex_buf[i]));
  OPENSSL_free(mutex_buf);
}
```

と記述した．

4.2.2 暗号処理に関するプログラム

AES 暗号処理を行う基となる関数 `do_crypt()` は，[11] の Crypto/EVP_Encrypto ページにあるサンプルプログラムを参考に，以下のように作成した．なお，引数であるが，`in` は処理前のファイル，`out` は処理後のファイルであり，`do_encrypt` は暗号化か復号を指定する．

[13] では 3 行目の `inbuf` を暗号化するファイルサイズと同じサイズのメモリ領域を `malloc` を使って確保したが，本報告では `inbuf` のサイズを BUFSIZE($= 1024$) で固定した．なお，今回は暗号の鍵 `key` と初期ベクトル `iv` はグローバル変数として扱ったが，この関数の引数にすることも考えられる．

```
1  static int do_crypt(FILE *in, FILE *out,
     int do_encrypt)
2  {
3    unsigned char inbuf[BUFSIZE], outbuf[
       BUFSIZE + EVP_MAX_BLOCK_LENGTH];
4    int inlen, outlen;
5    EVP_CIPHER_CTX ctx;
6    EVP_CIPHER_CTX_init(&ctx);
7    EVP_CipherInit_ex(&ctx, EVP_aes_128_cbc
       (), NULL, NULL, NULL, do_encrypt);
8    OPENSSL_assert(EVP_CIPHER_CTX_key_length
       (&ctx) == 16);
9    OPENSSL_assert(EVP_CIPHER_CTX_iv_length(&
       ctx) == 16);
10   EVP_CipherInit_ex(&ctx, NULL, NULL, key,
       iv, do_encrypt);
11   for(;;) {
12     inlen = fread(inbuf, 1, BUFSIZE, in);
13     if(inlen <= 0) break;
14     if(!EVP_CipherUpdate(&ctx, outbuf, &
         outlen, inbuf, inlen)) {
15       EVP_CIPHER_CTX_cleanup(&ctx);
16       return 1;
17     }
18     fwrite(outbuf, 1, outlen, out);
19   }
20   if(!EVP_CipherFinal_ex(&ctx, outbuf, &
       outlen)) {
21     EVP_CIPHER_CTX_cleanup(&ctx);
22     return 1;
23   }
24   fwrite(outbuf, 1, outlen, out);
25
26   EVP_CIPHER_CTX_cleanup(&ctx);
27   return 0;
28 }
```

`pthread_creat()` に渡す関数 `do_crypt_thread()` は以下のようにした．

```
static void *do_crypt_thread(void *arg)
{
  int i;
  FILE *rfp, *wfp;
  int thread_count = *((int *)arg);
  for (i = thread_count; i < fnum; i +=
    thread_num) {
```

```
    rfp = fopen(files[i].rfname, "r");
    if (!rfp) {
      perror("fopen");
      pthread_exit(NULL);
    }
    wfp = fopen(files[i].wfname, "w");
    if (!wfp) {
      perror("fopen");
      pthread_exit(NULL);
    }
    if(do_crypt(rfp, wfp, mode_flag)) {
      printf("failed encryption or decryption\n");
      pthread_exit(NULL);
    }
    fclose(rfp);
    fclose(wfp);
    thread_count += thread_num;
  }
  pthread_exit(NULL);
}
```

最後に，2 つのディレクトリ名 inputdir, outputdir を引数として，inputdir に入っているファイルすべてをマルチスレッドにより暗号化し outputdir に出力する関数 do_crypt_dir() を次のように記述した．なお，今回はスレッド数 thread_num はグローバル変数として扱ったが，この関数の引数にすることも考えられる．

```
static int do_crypt_dir(const char *inputdir, const char *outputdir)
{
  int i;
  int error;
  DIR *dp;
  struct dirent *dir;
  int tnum[thread_num];
  dp = opendir(inputdir);
  /* ファイル名の作成 */
  i = 0;
  while ((dir = readdir(dp)) != NULL) {
    if (strcmp(dir->d_name, ".") != 0 && strcmp(dir->d_name, "..") != 0) {
      strcpy(files[i].rfname, inputdir);
      strcat(files[i].rfname, "/");
      strcat(files[i].rfname, dir->d_name);
      strcpy(files[i].wfname, outputdir);
      strcat(files[i].wfname, "/");
      strcat(files[i].wfname, dir->d_name);
      i++;
    }
  }
  /* スレッドIDをスレッド分用意する */
  pthread_t tid[thread_num];
  init_locks();
  /* start threads */
  if (thread_num > fnum) thread_num = fnum;
  for (i = 0; i < thread_num; i++) {
    tnum[i] = i;
    /* スレッドを生成する */
    error = pthread_create(
      &tid[i],          //* スレッドID
      NULL,             //* スレッドの属性
      do_crypt_thread,  //* スレッドで実行する関数
      (void *)&tnum[i]  //* スレッドで実行する関数の引数*/
    );
    if(error) fprintf(stderr, "Couldn't run thread number %d, errno %d\n", i, error);
  }
  for(i = 0; i < thread_num; i++) {
  /* スレッドの終了を待つ */
    error = pthread_join(
    tid[i],   //* スレッドID
    NULL      //* スレッドの終了ステータス
  );
#ifdef DEBUG
  fprintf(stderr, "Thread %d terminated\n", i);
#endif
  }
  /* threads end here */
  clear_locks();
  return 0;
}
```

ファイルサイズ(B)	ファイル数 10						ファイル数 100					
	Xeon1		Xeon2		Xeon Phi		Xeon1		Xeon2		Xeon Phi	
128	1	1.083	1	1.250	1	2.734	2	6.498	2	6.714	1	17.29
	—	—	*1*	*2.672*	*1*	*20.84*	—	—	*2*	*8.744*	*1*	*40.42*
				213%		762%				130%		233%
256	1	1.102	1	1.300	1	2.844	2	6.544	2	6.744	1	18.28
	—	—	*1*	*2.685*	*1*	*20.81*	—	—	*2*	*8.727*	*1*	*41.54*
				206%		731%				129%		227%
512	2	1.174	1	1.367	1	3.039	2	6.378	2	6.510	1	20.25
	—	—	*1*	*2.783*	*1*	*20.97*	—	—	*2*	*8.755*	*1*	*45.95*
				203%		690%				134%		226%
1K	2	1.239	2	1.441	1	3.407	4	6.819	4	6.947	1	23.96
	—	—	*1*	*2.949*	*1*	*21.32*	—	—	*2*	*9.018*	*1*	*51.05*
				204%		625%				129%		213%
2K	2	1.339	2	1.540	1	4.180	4	6.636	4	7.297	2	27.43
	—	—	*1*	*3.145*	*1*	*22.23*	—	—	*2*	*9.873*	*1*	*56.03*
				204%		531%				135%		204%
4K	2	1.663	2	1.820	1	6.031	4	7.875	4	9.034	2	36.52
	—	—	*2*	*3.288*	*1*	*23.99*	—	—	*2*	*10.22*	*2*	*60.84*
				180%		397%				113%		166%
8K	4	2.057	4	2.180	2	9.005	10	9.697	4	11.42	4	43.89
	—	—	*2*	*3.857*	*1*	*26.99*	—	—	*4*	*12.34*	*4*	*65.33*
				176%		299%				108%		148%
16K	4	2.636	4	3.005	4	11.53	10	11.39	4	14.65	4	57.88
	—	—	*2*	*4.815*	*1*	*33.89*	—	—	*4*	*16.16*	*4*	*77.49*
				160%		293%				110%		133%
32K	10	3.939	4	4.434	4	15.63	10	15.27	10	17.90	4	86.30
	—	—	*4*	*6.469*	*4*	*40.74*	—	—	*10*	*20.99*	*4*	*110.9*
				145%		260%				117%		128%
64K	10	4.962	10	6.462	4	24.10	10	18.17	20	18.93	10	109.3
	—	—	*10*	*8.541*	*10*	*44.74*	—	—	*10*	*25.17*	*20*	*122.0*
				132%		185%				132%		111%
128K	10	6.901	10	9.098	10	28.08	10	26.60	20	24.65	20	141.1
	—	—	*10*	*12.19*	*10*	*53.31*	—	—	*10*	*40.73*	*20*	*203.5*
				133%		189%				165%		144%
256K	10	9.683	20	13.12	10	40.81	10	43.27	20	39.75	20	171.8
	—	—	*10*	*18.11*	*10*	*67.72*	—	—	*10*	*65.87*	*20*	*294.6*
				138%		165%				165%		171%
512K	20	14.23	20	18.23	10	62.82	10	73.23	20	71.75	50	216.3
	—	—	*10*	*23.50*	*10*	*99.17*	—	—	*20*	*99.05*	*50*	*440.3*
				128%		157%				138%		203%
1M	10	22.08	10	28.11	10	108.7	10	134.7	20	121.2	50	255.3
	—	—	*10*	*41.98*	*10*	*154.7*	—	—	*20*	*173.3*	*100*	*686.7*
				149%		142%				142%		268%
2M	20	35.63	20	49.57	10	192.9	50	240.9	100	212.2	100	362.0
	—	—	*10*	*67.44*	*10*	*259.6*	—	—	*20*	*318.1*	*100*	*1086.7*
				136%		134%				149%		300%
4M	10	63.68	20	77.49	10	364.6	100	443.9	100	399.5	100	602.1
	—	—	*10*	*117.4*	*10*	*456.8*	—	—	*20*	*529.0*	*100*	*1588.1*
				151%		125%				132%		263%

単位は ms，上段は今回の結果，中段の斜体は [13] での結果，下段は改善率

表 6: 各パラメータに対する最良のスレッド数と，そのスレッド数での暗号処理平均時間

4.3 実装実験

表 3 のパラメータに対して，4.2 節のプログラムの暗号化時間の測定をそれぞれ 100 回行った．比較のため [13] の実装結果と共に，処理時間の平均値と最大値を付録にまとめた．本報告での実装では，[13] で見られた秒オーダーの異常遅延は見られなかった．

ファイルサイズ毎に，暗号処理時間の平均時間が最小となるスレッド数とそのスレッド数での平均処理時間を，表 6 にまとめた．比較のため [13] での結果も載せたが，[13] の結果では異常遅延発生が生じたパラメータでは，異常遅延の場合を除いた平均時間を求めている．

表 3 のパラメータに対して，暗号処理時間の平均時間 (異常遅延発生の時は異常遅延のない場合の平均時間) が最小となるスレッド数とそのスレッド数での平均処理時間をまとめると，以下のようになった．なお，[13] では Intel Xeon 1 での実装評価を行わなかった．

表 6 から本報告の実装結果は [13] 比較するとどのパラメータでも暗号処理速度は改善されていることが分かるが，特に Intel Xeon Phi での改善率が大きい．しかしながら，今なお Intel Xeon Phi は Intel Xeon より暗号処理が遅い結果となった．

また Intel Xeon 1 と Intel Xeon 2 を比較すると，ファイル数が 10 の時はいずれも Intel Xeon 1 の方が高速で

	本報告の実装	[13] の実装
並列実装基盤	pthread	OpenMP
基となる暗号化関数での `malloc` 使用	無	有
異常遅延	無	有
暗号化速度	速	遅
最適プロセッサ	Intel Xeon	Intel Xeon

表 7: 本報告の実装と [13] の実装との比較

あり，ファイル数が 100 の場合でも 64KB までは Intel Xeon 1 の方が高速である．2 つの CPU の使用ではオーバーヘッドが相当大きいということが考えられる．

表 7 に本報告の実装と [13] の実装との比較をまとめる．

5. まとめと今後の課題

OpenSSL の暗号ライブラリを活用し並列実装基盤として pthread を用いて作成したファイル単位で AES 並列暗号プログラムを，マルチコア CPU やメニーコア・コプロセッサ上で実行する場合処理時間がどうなるか調べた．先行報告 [13] と比較すると，どのパラメータでも暗号処理時間を改善でき，[13] で生じた異常遅延を防ぐことができた．しかしながら，今回もメニーコア・コプロセッサのコア数に匹敵する処理速度は得られなかった．

本報告での暗号処理速度の改善が pthread によるものか malloc 除去によるものかは未確認であるため，今後の課題としたい．試行錯誤的であるが，マルチコア CPU やメニーコア・コプロセッサにより適した，OpenSSL を利用しての AES の並列実装法を探っていきたい．また，今回は 4.2 節で説明した `do_crypt()` 関数の `BUFSIZE` を 1024 に固定したが，この値を変動させると暗号処理時間がどう変わるか調査したい．更に，新しい Intel コンパイラを使ってプログラムを再コンパイルして暗号処理時間を再評価したい．

謝辞

本研究の一部は，JSPS 科研費 25330156 の助成を受けたものです．また本研究の一部は，戦略的基盤技術高度化支援事業- 電子書籍型コンテンツにおける軽量的著作権保護技術の応用研究及びスマートフォン・タブレット向け配信サービスの実証実験の支援を受けたものです．

参考文献

[1] Breshears C., (千住 治郎. 訳): 並行コンピューティング技法 ―実践マルチコア/マルチスレッドプログラミング, オライリージャパン (2009).

[2] EPUB，`http://idpf.org/epub` (2015.06.23).

[3] GMP，`https://gmplib.org/` (2015.06.23).

[4] Harris, D. M., Harris, S. L., (鈴木貢，中條拓伯，天野英晴，永松礼夫. 訳)：ディジタル回路設計とコンピュータアーキテクチャ，翔泳社 (2009).

[5] Intel コンパイラ，`http://www.intel.co.jp/content/www/jp/ja/developer/software-products.html` (2015.06.23).

[6] Iwai, K., Nishikawa, N., and Kurokawa, T.: "Acceleration of AES encryption on CUDA GPU," Int. J. Networking and Computing, vol.2, no.1, pp.131–145 (2012).

[7] Le, D., Chang. J., Gou, X., Zhang A., and Liu, C.: "Parallel AES Algorithm for Fast Data Encryption on GPU," 2nd International Conference on Computer Engineering and Technology, *ICCET* (2010).

[8] Manavski, S. A.: "Cuda Compatible GPU as an Efficient Hardware Accelerator for AES Cryptography," IEEE International Conference on Signal Processing and Communication, *ICSPC*, pp.65–66 (2007).

[9] Nishikawa, N., Iwai, K., and Kurokawa, T.: "High-performance symmetric block ciphers on multicore CPU and GPUs," Int. J. Networking and Computing, vol.2, no.2, pp.251–268 (2012).

[10] 西川尚紀，岩井啓輔，田中秀磨，黒川恭一：GPU を用いたブロック暗号設計のための性能予測フレームワーク，電子情報通信学会論文，Vol. J97–D No. 12, pp.1740-1754 (2014).

[11] OpenSSL，`https://www.openssl.org/` (2015.06.23).

[12] Readum LCP，`http://readium.org/projects/readium-lcp` (2015.06.23).

[13] 白勢政明：メニーコア・コプロセッサ Xeon Phi での並列暗号実装, 2015-CSEC-68(27), (2015).

[14] Viega J.，Messier M.，Chandra P.，(齋藤 孝道. 訳)：OpenSSL–暗号・PKI・SSL/TLS ライブラリの詳細，オーム社 (2013).

[15] ジム・ジェファース, ジェームス・レインダース, (すがわらきよふみ. 訳)：インテル Xeon Phi コプロセッサーハイパフォーマンス・プログラミング，カットシステム (2013).

A 実装結果

ここでは，ファイル数が 10 と 100 の時の，本報告と [13] の実装結果を与える．A はスレッド数，B は暗号処理時間の平均値，C は暗号処理時間の最大値，単位は ms である．[13] の実装評価において異常遅延が生じた場合においては D，E，F も記載しており，D は 100 回中異常遅延が発生しなかった回数，E は異常遅延が発生しなかった場合の暗号処理の平均値，F は異常遅延が発生しなかった場合の暗号処理の最大値である．なお [13] の実装評価の値は斜体としている．

1.1 ファイル数 10 の時の実装結果

ファイルサイズ 128B

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	**1.083**	1.167	1.553	3.351	**1.250**	1.407	1.774	4.176	**2.734**	6.433	11.23	20.71
C	2.531	**1.284**	3.488	4.317	**2.060**	3.060	4.566	8.056	**2.831**	7.503	23.59	26.08
B	–	–	–	–	*2.672*	*2.929*	*3.601*	*6.221*	*20.84*	*28.85*	*34.06*	*44.34*
C	–	–	–	–	*7.226*	*3.104*	*4.271*	*30.58*	*22.39*	*34.31*	*39.24*	*49.10*

ファイルサイズ 256B

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	**1.102**	1.155	1.603	3.374	**1.300**	1.380	1.769	4.122	**2.844**	6.525	11.77	20.59
C	1.960	**1.261**	3.104	4.484	**1.357**	1.574	4.626	6.032	**2.942**	10.15	18.31	26.41
B	–	–	–	–	*2.685*	*2.959*	*3.640*	*5.091*	*20.81*	*29.70*	*33.50*	*43.36*
C	–	–	–	–	*4.063*	*3.667*	*4.833*	*9.959*	*22.50*	*33.52*	*37.27*	*49.44*

ファイルサイズ 512B

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	1.178	**1.174**	1.593	3.318	**1.367**	1.417	1.820	4.134	**3.039**	6.844	11.46	20.61
C	**1.260**	1.292	2.455	6.859	**1.469**	3.271	4.219	5.577	**3.174**	10.27	18.96	28.18
B	–	–	–	–	*2.783*	*2.987*	*3.711*	*5.766*	*20.97*	*28.43*	*34.73*	*43.20*
C	–	–	–	–	*8.352*	*3.222*	*4.906*	*10.43*	*21.97*	*30.67*	*41.05*	*49.33*

ファイルサイズ 1KB

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	1.299	**1.239**	1.589	3.338	1.482	**1.441**	1.739	4.237	**3.407**	7.009	10.91	20.08
C	1.399	**1.331**	2.529	5.976	**1.594**	2.984	3.072	6.532	**3.590**	8.396	18.96	26.73
B	–	–	–	–	*2.949*	*3.026*	*3.660*	*6.071*	*21.32*	*29.21*	*33.63*	*43.27*
C	–	–	–	–	*11.76*	*3.425*	*4.492*	*24.86*	*23.92*	*34.00*	*38.44*	*48.36*

ファイルサイズ 2KB

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	1.567	**1.339**	1.570	3.402	1.781	**1.540**	1.829	4.304	**4.180**	6.797	11.08	20.65
C	1.638	**1.420**	3.193	5.183	**2.168**	3.473	3.007	6.347	**4.412**	7.742	18.29	26.41
B	–	–	–	–	*3.145*	*3.153*	*3.706*	*6.246*	*22.23*	*28.70*	*34.45*	*43.63*
C	–	–	–	–	*6.303*	*3.462*	*4.899*	*28.71*	*23.54*	*31.49*	*39.76*	*47.86*

ファイルサイズ 4KB

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	2.187	**1.663**	1.718	3.353	2.390	**1.820**	1.904	4.189	**6.031**	7.935	10.80	21.71
C	**2.246**	2.482	3.437	4.667	2.842	**2.096**	3.220	6.553	9.604	**9.283**	19.23	27.80
B	–	–	–	–	*3.639*	*3.288*	*3.707*	*5.875*	*23.99*	*29.41*	*32.16*	*38.55*
C	–	–	–	–	*5.904*	*3.494*	*5.826*	*23.62*	*28.12*	*32.70*	*37.74*	*42.98*

ファイルサイズ 8KB

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	3.334	2.246	**2.057**	3.592	3.540	2.457	**2.180**	4.267	9.334	**9.005**	11.13	22.85
C	**3.402**	4.113	4.776	5.497	4.633	**2.658**	4.227	6.000	12.82	**10.69**	16.15	29.63
B	–	–	–	–	*4.846*	*3.857*	*4.100*	*5.869*	*26.99*	*31.29*	*35.11*	*39.90*
C	–	–	–	–	*9.905*	*4.626*	*6.154*	*14.54*	*28.66*	*35.08*	*41.27*	*46.54*

ファイルサイズ 16KB

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	5.651	3.389	**2.636**	3.708	5.808	3.612	**3.005**	4.710	15.94	12.80	**11.53**	22.98
C	6.653	**3.495**	4.338	6.619	6.860	**3.822**	5.163	6.965	19.45	**14.05**	18.65	28.03
B	–	–	–	–	*7.105*	*4.815*	*5.015*	*6.000*	*33.89*	*34.87*	*35.70*	*39.99*
C	–	–	–	–	*9.341*	*5.246*	*6.216*	*9.848*	*38.91*	*39.34*	*41.25*	*45.14*

ファイルサイズ 32KB

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	10.12	5.686	3.975	**3.939**	9.229	5.905	**4.434**	4.871	29.17	19.56	**15.63**	23.69
C	10.48	6.027	**4.578**	6.301	11.65	**6.800**	7.015	7.680	31.41	**21.32**	22.68	29.99
B	–	–	–	–	*11.51*	*6.909*	*6.469*	*6.873*	*47.05*	*42.20*	*40.74*	*42.46*
C	–	–	–	–	*19.42*	*7.459*	*8.558*	*16.96*	*51.22*	*45.42*	*45.74*	*49.24*

ファイルサイズ 64KB

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	16.95	10.18	6.730	**4.962**	15.91	10.29	7.488	**6.462**	55.85	31.66	**24.10**	27.18
C	20.20	10.89	7.388	**6.662**	19.65	11.37	11.51	**8.338**	59.30	38.94	**29.70**	34.94
B	–	–	–	–	*17.87*	*10.63*	*9.097*	*8.541*	*74.36*	*55.52*	*48.67*	*44.74*
C	–	–	–	–	*20.05*	*13.13*	*13.86*	*15.18*	*78.32*	*60.40*	*56.46*	*49.24*

ファイルサイズ 128KB

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	26.05	16.95	11.61	**6.901**	25.05	16.92	11.95	**9.098**	108.8	58.94	40.29	**28.08**
C	28.56	19.52	13.15	**10.34**	28.71	19.96	15.24	**12.54**	109.1	60.90	49.52	**34.00**
B	–	–	–	–	*26.49*	*15.45*	*13.89*	*12.19*	*130.9*	*82.94*	*68.83*	*53.31*
C	–	–	–	–	*29.73*	*21.60*	*19.69*	*17.93*	*133.1*	*89.82*	*73.14*	*61.70*

ファイルサイズ 256KB

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	43.43	25.93	17.74	**9.683**	43.28	25.58	18.90	**13.66**	214.4	111.3	72.63	**40.81**
C	46.47	28.26	23.94	**14.99**	47.35	30.50	26.76	**19.66**	215.0	117.9	84.00	**56.43**
B	–	–	–	–	*43.28*	*23.63*	*21.75*	*18.11*	*242.0*	*141.8*	*103.0*	*67.72*
C	–	–	–	–	*47.45*	*37.95*	*29.69*	*22.19*	*251.6*	*155.7*	*112.8*	*77.19*

ファイルサイズ 512KB

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	80.65	45.77	28.02	**14.55**	81.03	44.74	30.19	**19.06**	424.7	216.6	135.3	**62.82**
C	86.49	49.13	32.91	**20.98**	85.35	50.21	39.52	**26.99**	425.5	236.0	144.4	**79.07**
B	–	–	–	–	*80.44*	*42.25*	*32.76*	*263.2*	*460.6*	*252.9*	*168.7*	*99.17*
C	–	–	–	–	*83.64*	*48.06*	*48.36*	*12034.6*	*464.7*	*282.7*	*184.3*	*106.2*
D	–	–	–	–	*100*	*100*	*100*	*97*	*100*	*100*	*100*	*100*
E	–	–	–	–	*80.44*	*42.25*	*32.76*	*23.50*	*460.6*	*252.9*	*168.7*	*99.17*
F	–	–	–	–	*83.64*	*48.06*	*48.36*	*34.35*	*464.7*	*282.7*	*184.3*	*106.2*

ファイルサイズ 1MB

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	137.8	78.60	50.74	**22.08**	132.5	79.51	51.89	**28.11**	844.9	426.7	262.8	**108.7**
C	155.1	85.45	54.56	**26.26**	158.8	82.53	69.26	**42.47**	849.8	450.4	296.5	**135.8**
B	–	–	–	–	*131.1*	*67.05*	*60.59*	*645.9*	*896.3*	*475.2*	*324.8*	*154.7*
C	–	–	–	–	*154.0*	*81.86*	*88.12*	*14136.9*	*917.1*	*479.4*	*353.1*	*169.9*
D	–	–	–	–	*100*	*100*	*100*	*94*	*100*	*100*	*100*	*100*
E	–	–	–	–	*131.1*	*67.05*	*60.59*	*41.98*	*896.3*	*475.2*	*324.8*	*154.7*
F	–	–	–	–	*154.0*	*81.86*	*88.12*	*247.1*	*917.1*	*479.4*	*353.1*	*169.9*

ファイルサイズ 2MB

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	270.7	133.5	82.42	**35.85**	257.7	126.0	82.50	**49.72**	1685.3	850.6	515.1	**192.9**
C	283.3	146.4	95.67	**42.74**	272.4	130.7	139.6	**64.53**	1687.7	990.6	567.3	**218.6**
B	–	–	–	–	*249.0*	*130.0*	*279.8*	*683.2*	*1763.0*	*940.2*	*626.0*	*259.6*
C	–	–	–	–	*291.2*	*152.7*	*6085.5*	*10368.3*	*1767.7*	*1048.9*	*658.4*	*280.5*
D	–	–	–	–	*100*	*100*	*97*	*90*	*100*	*100*	*100*	*100*
E	–	–	–	–	*249.0*	*130.0*	*103.3*	*67.44*	*1763.0*	*940.2*	*626.0*	*259.6*
F	–	–	–	–	*291.2*	*152.7*	*162.4*	*82.93*	*1767.7*	*1048.9*	*658.4*	*280.5*

ファイルサイズ 4MB

	Intel Xeon 1				Intel Xeon 2				Intel Xeon Phi			
A	1	2	4	10	1	2	4	10	1	2	4	10
B	525.2	260.6	156.3	**63.68**	500.0	244.8	153.3	**80.77**	3364.4	1694.8	1021.7	**364.6**
C	537.9	277.4	162.9	**78.87**	517.3	251.8	222.3	**112.7**	3369.2	1966.3	1185.6	**427.5**
B	–	–	–	–	*491.4*	*252.5*	*314.8*	*452.8*	*3462.4*	*1771.9*	*1121.4*	*456.8*
C	–	–	–	–	*1080.1*	*289.9*	*6085.0*	*6234.9*	*3472.3*	*2041.0*	*1251.3*	*487.9*
D	–	–	–	–	*100*	*100*	*98*	*94*	*100*	*100*	*100*	*100*
E	–	–	–	–	*491.4*	*252.5*	*198.9*	*117.4*	*3462.4*	*1771.9*	*1121.4*	*456.8*
F	–	–	–	–	*1080.1*	*289.9*	*268.9*	*148.0*	*3472.3*	*2041.0*	*1251.3*	*487.9*

1.2 ファイル数 100 の時の実装結果

ファイルサイズ 128B

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	6.978	**6.498**	6.892	8.834	12.79	7.023	**6.714**	6.837	10.66	14.56	**17.29**	25.00	37.31	92.26	107.8	92.11	79.50
C	**7.237**	7.385	7.921	11.10	13.93	8.503	**7.748**	9.035	12.91	16.55	**17.63**	28.46	45.32	101.6	119.5	106.1	93.19
B	–	–	–	–	–	*9.730*	*8.744*	*12.13*	*19.46*	*27.89*	*40.42*	*55.20*	*73.36*	*149.2*	*148.3*	*171.7*	*222.9*
C	–	–	–	–	–	*15.01*	*13.21*	*13.57*	*56.47*	*87.48*	*42.85*	*59.00*	*78.84*	*155.8*	*153.3*	*181.3*	*238.6*

ファイルサイズ 256B

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	7.310	**6.544**	7.055	8.863	12.91	7.396	**6.744**	7.180	10.55	14.81	**18.28**	25.97	37.08	91.26	106.3	92.84	79.04
C	7.671	**6.777**	8.649	10.01	14.32	8.014	**7.039**	11.16	12.22	17.34	**18.77**	29.90	44.91	98.61	114.5	108.9	94.25
B	–	–	–	–	–	*10.08*	*8.727*	*11.92*	*18.86*	*28.54*	*41.54*	*59.04*	*71.59*	*149.2*	*148.6*	*173.7*	*222.4*
C	–	–	–	–	–	*15.91*	*11.01*	*13.16*	*37.53*	*67.24*	*43.07*	*62.15*	*78.52*	*159.3*	*154.4*	*182.3*	*241.0*

ファイルサイズ 512B

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	7.918	**6.378**	6.709	8.919	12.98	8.107	**6.510**	6.859	10.49	14.75	**20.25**	24.84	36.98	90.12	106.2	90.33	79.78
C	8.238	8.183	**7.801**	11.23	15.64	8.492	**7.175**	8.586	12.72	16.72	**20.67**	28.59	42.57	99.90	116.3	108.8	93.58
B	–	–	–	–	–	*10.62*	*8.755*	*12.72*	*19.29*	*27.28*	*45.95*	*58.53*	*73.06*	*148.2*	*148.6*	*173.0*	*223.3*
C	–	–	–	–	–	*13.46*	*10.97*	*23.08*	*37.76*	*66.68*	*48.51*	*63.88*	*79.22*	*154.2*	*156.1*	*186.2*	*237.8*

ファイルサイズ 1KB

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	9.245	7.192	**6.819**	8.863	12.92	9.272	7.360	**6.947**	10.79	14.73	**23.96**	26.58	37.48	88.96	104.4	91.89	80.36
C	10.54	**7.504**	7.761	11.32	13.96	10.36	**8.250**	9.119	12.75	17.93	**27.49**	28.20	50.48	100.5	111.3	111.0	96.14
B	–	–	–	–	–	*11.97*	*9.016*	*12.04*	*19.39*	*28.90*	*51.05*	*55.84*	*73.78*	*148.1*	*148.7*	*173.3*	*222.8*
C	–	–	–	–	–	*13.55*	*16.87*	*15.11*	*48.09*	*76.26*	*52.99*	*61.59*	*79.92*	*154.1*	*156.0*	*187.5*	*238.1*

ファイルサイズ 2KB

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	11.32	7.914	**6.636**	8.827	12.87	11.33	8.303	**7.297**	10.77	14.96	31.62	**27.43**	36.47	89.29	103.5	86.65	80.34
C	12.02	8.359	**7.764**	10.51	14.33	15.47	9.688	**9.482**	12.53	17.37	31.97	**31.54**	46.81	95.54	111.8	106.4	97.32
B	–	–	–	–	–	*14.50*	*9.873*	*12.20*	*19.47*	*28.45*	*56.03*	*59.64*	*73.79*	*148.6*	*148.5*	*171.9*	*221.4*
C	–	–	–	–	–	*21.37*	*19.48*	*18.19*	*39.42*	*76.45*	*61.72*	*66.84*	*79.07*	*156.3*	*154.2*	*180.2*	*243.5*

ファイルサイズ 4KB

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	17.42	10.92	**7.875**	8.911	13.21	15.03	11.22	**9.034**	10.93	15.30	49.14	**36.52**	36.77	88.74	106.1	99.59	89.23
C	18.94	11.63	**9.038**	12.07	15.83	17.79	**12.56**	12.84	12.88	16.52	49.59	**38.84**	44.29	97.20	112.7	120.7	106.0
B	–	–	–	–	–	*19.42*	*10.22*	*10.97*	*15.95*	*26.18*	*69.66*	*60.84*	*63.38*	*102.4*	*102.0*	*127.8*	*175.5*
C	–	–	–	–	–	*20.02*	*21.67*	*24.83*	*26.54*	*86.29*	*70.47*	*65.65*	*68.67*	*109.4*	*109.4*	*139.5*	*188.7*

ファイルサイズ 8KB

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	21.32	15.74	10.67	**9.697**	14.44	21.23	15.03	**11.42**	11.86	16.09	82.30	51.20	**43.89**	88.02	105.7	112.0	99.09
C	24.43	17.31	**11.79**	11.93	16.97	24.61	18.07	16.92	**14.27**	18.25	82.85	54.87	**52.11**	97.80	112.3	135.5	123.4
B	–	–	–	–	–	*22.59*	*13.29*	*12.34*	*16.57*	*24.39*	*104.5*	*74.55*	*65.33*	*101.5*	*104.2*	*131.7*	*180.8*
C	–	–	–	–	–	*26.17*	*23.25*	*20.11*	*32.85*	*60.15*	*116.1*	*82.49*	*75.53*	*111.2*	*112.1*	*144.1*	*196.1*

ファイルサイズ 16KB

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	33.99	22.02	14.24	**11.39**	17.25	31.93	21.92	**14.65**	14.74	17.81	148.4	84.70	**57.88**	90.37	107.4	128.3	134.5
C	39.46	24.02	17.75	**13.16**	18.74	36.50	24.17	18.04	**17.73**	21.64	149.1	88.30	**72.56**	100.5	114.7	144.6	163.1
B	–	–	–	–	–	*33.16*	*18.28*	*16.16*	*17.85*	*24.79*	*172.5*	*108.7*	*77.49*	*104.5*	*105.4*	*133.9*	*186.8*
C	–	–	–	–	–	*42.20*	*39.08*	*22.60*	*37.91*	*66.47*	*196.7*	*118.8*	*88.71*	*109.2*	*110.6*	*142.7*	*201.4*

ファイルサイズ 32KB

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	57.23	32.43	20.10	**15.27**	23.23	54.52	32.24	20.93	**17.90**	20.36	280.3	151.2	**86.30**	102.2	113.7	138.1	156.0
C	59.53	40.63	27.52	**21.11**	26.27	60.04	35.31	31.08	25.14	**24.65**	283.0	184.6	**104.6**	111.2	123.6	152.0	183.9
B	–	–	–	–	–	*58.42*	*28.60*	*23.12*	*20.99*	*26.94*	*301.0*	*174.7*	*110.9*	*112.1*	*111.2*	*135.6*	*191.4*
C	–	–	–	–	–	*59.01*	*32.94*	*34.04*	*31.19*	*77.73*	*306.1*	*180.7*	*120.2*	*118.1*	*119.0*	*144.4*	*204.6*

ファイルサイズ 64KB

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	97.73	55.59	31.19	**18.17**	32.83	93.63	55.30	31.74	22.98	**18.93**	547.1	281.3	150.9	**109.3**	121.2	142.6	169.6
C	105.6	58.42	34.65	**24.84**	42.03	107.7	74.80	47.91	33.73	**29.47**	555.5	286.5	177.4	**133.8**	134.9	157.5	181.7
B	–	–	–	–	–	*99.67*	*51.15*	*36.03*	*25.17*	*29.49*	*568.5*	*319.6*	*192.4*	*127.0*	*122.0*	*142.0*	*205.7*
C	–	–	–	–	–	*107.2*	*56.71*	*51.87*	*31.32*	*74.56*	*597.2*	*353.5*	*198.1*	*133.2*	*130.4*	*159.1*	*239.6*

ファイルサイズ 128KB

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	168.6	88.94	53.22	**26.60**	40.75	165.6	91.32	51.38	35.47	**24.65**	1078.0	552.0	287.9	143.2	**141.1**	161.6	186.3
C	186.6	105.0	58.30	**36.80**	57.43	193.9	102.4	76.77	44.89	**31.32**	1083.3	667.1	334.9	179.4	**161.5**	173.1	200.6
B	–	–	–	–	–	*160.4*	*82.43*	*63.92*	*40.73*	*45.28*	*1130.2*	*601.3*	*345.9*	*215.2*	*203.5*	*224.8*	*278.5*
C	–	–	–	–	–	*190.3*	*95.57*	*90.55*	*51.85*	*83.19*	*1135.0*	*683.7*	*370.7*	*224.7*	*210.7*	*262.5*	*342.3*

ファイルサイズ 256KB

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	344.5	169.4	86.33	**43.27**	56.44	322.3	158.8	86.45	58.93	**39.75**	2133.5	1082.6	559.3	242.8	**171.8**	173.8	198.9
C	351.4	176.2	97.16	**51.73**	69.55	373.8	169.0	139.2	77.74	**45.88**	2150.2	1301.2	658.8	286.3	201.5	**196.7**	217.0
B	–	–	–	–	–	*311.0*	*159.9*	*97.88*	*65.87*	*868.4*	*2227.4*	*1160.3*	*653.7*	*335.5*	*294.6*	*324.5*	*345.3*
C	–	–	–	–	–	*355.0*	*185.8*	*145.0*	*90.91*	*14055.0*	*2230.9*	*1322.6*	*703.2*	*349.2*	*319.7*	*403.6*	*421.1*
D	–	–	–	–	–	*100*	*100*	*100*	*100*	*94*	*100*	*100*	*100*	*100*	*100*	*100*	*100*
E	–	–	–	–	–	*311.0*	*159.9*	*97.88*	*65.87*	*66.21*	*2227.4*	*1160.3*	*653.7*	*335.5*	*294.6*	*324.5*	*345.3*
F	–	–	–	–	–	*355.0*	*185.8*	*145.0*	*90.91*	*114.9*	*2230.9*	*1322.6*	*703.2*	*349.2*	*319.7*	*403.6*	*421.1*

ファイルサイズ 512KB

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	657.5	327.4	165.2	**73.23**	87.55	614.2	306.5	164.4	97.57	**71.75**	4235.4	2130.6	1089.7	457.6	264.9	**216.3**	230.1
C	672.1	340.3	171.7	**79.93**	100.7	664.8	313.6	269.0	140.8	**79.70**	4251.4	2536.1	1309.4	539.9	345.6	258.5	**250.6**
B	–	–	–	–	–	*615.0*	*315.9*	*194.1*	*768.1*	*1684.7*	*4415.7*	*2285.5*	*1259.5*	*516.3*	*443.8*	*440.3*	*464.2*
C	–	–	–	–	–	*642.2*	*534.2*	*281.0*	*14089.7*	*14137.7*	*4431.0*	*2599.0*	*1357.7*	*568.0*	*514.9*	*526.9*	*638.0*
D	–	–	–	–	–	*100*	*100*	*100*	*94*	*79*	*100*	*100*	*100*	*100*	*100*	*100*	*100*
E	–	–	–	–	–	*615.0*	*315.9*	*194.1*	*109.3*	*99.05*	*4415.7*	*2285.5*	*1259.5*	*516.3*	*443.8*	*440.3*	*464.2*
F	–	–	–	–	–	*642.2*	*534.2*	*281.0*	*122.0*	*137.6*	*4431.0*	*2599.0*	*1357.7*	*568.0*	*514.9*	*526.9*	*638.0*

ファイルサイズ 1MB

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	1281.2	643.4	323.4	**134.7**	151.9	1202.0	601.0	309.1	181.7	**121.2**	8436.1	4241.1	2136.6	896.4	491.5	**255.3**	286.3
C	1319.2	674.6	328.4	**151.5**	166.6	1446.5	632.8	527.6	225.5	**137.2**	8489.2	5086.8	2555.1	1070.0	561.1	314.2	**308.3**
B	–	–	–	–	–	*1214.7*	*630.7*	*387.6*	*1575.2*	*1778.4*	*8727.2*	*4506.6*	*2465.1*	*857.9*	*747.0*	*695.7*	*686.7*
C	–	–	–	–	–	*1237.4*	*1035.7*	*580.6*	*11525.4*	*11932.4*	*8765.5*	*5107.1*	*2697.2*	*1005.2*	*904.9*	*968.6*	*1069.4*
D	–	–	–	–	–	*100*	*100*	*100*	*77*	*75*	*100*	*100*	*100*	*100*	*100*	*100*	*100*
E	–	–	–	–	–	*1214.7*	*630.7*	*387.6*	*203.9*	*173.3*	*8727.2*	*4506.6*	*2465.1*	*857.9*	*747.0*	*695.7*	*686.7*
F	–	–	–	–	–	*1237.4*	*1035.7*	*580.6*	*227.8*	*203.2*	*8765.5*	*5107.1*	*2697.2*	*1005.2*	*904.9*	*968.6*	*1069.4*

ファイルサイズ 2MB

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	2575.2	1276.0	640.4	261.9	**246.4**	2386.4	1196.7	612.0	334.2	**232.4**	16832	8443.8	4289.2	1756.6	931.6	436.0	**362.0**
C	2604.3	1296.6	646.3	307.0	**279.7**	2504.1	1492.4	1046.7	446.2	**263.6**	16889	10187	5131.8	2077.4	1052.3	498.0	**402.9**
B	–	–	–	–	–	*2454.8*	*1275.9*	*800.9*	*2392.5*	*2551.1*	*17455*	*8987.8*	*4704.0*	*1601.7*	*1357.0*	*1291.0*	*1086.7*
C	–	–	–	–	–	*5409.2*	*2368.6*	*6559.2*	*12034*	*12117*	*17498*	*10141*	*5265.0*	*1922.2*	*1742.5*	*1718.3*	*1719.5*
D	–	–	–	–	–	*99*	*99*	*99*	*66*	*67*	*100*	*97*	*100*	*100*	*100*	*100*	*100*
E	–	–	–	–	–	*2425.0*	*1264.9*	*742.7*	*397.0*	*318.1*	*17455*	*8952.4*	*4704.0*	*1601.7*	*1357.0*	*1291.0*	*1086.7*
F	–	–	–	–	–	*3265.6*	*1812.3*	*1113.6*	*443.4*	*361.7*	*17498*	*9024.3*	*5265.0*	*1922.2*	*1742.5*	*1718.3*	*1719.5*

ファイルサイズ 4MB

	Intel Xeon 1					Intel Xeon 2					Intel Xeon Phi						
A	1	2	4	10	20	1	2	4	10	20	1	2	4	10	20	50	100
B	5157.7	2542.2	1273.3	514.4	**468.4**	4809.1	2379.7	1207.2	624.5	**456.5**	33610	16939	8518.3	3467.7	1830.8	814.5	**602.1**
C	6192.3	2585.8	1279.5	521.5	**508.9**	4874.2	2436.7	1229.8	878.2	**527.2**	33664	20314	10234	4072.5	2053.1	875.8	**636.2**
B	–	–	–	–	–	*4821.3*	*2472.3*	*1447.9*	*1340.2*	*1952.5*	*34430*	*17476*	*9067.4*	*2806.7*	*2315.8*	*1986.0*	*1602.3*
C	–	–	–	–	–	*5831.4*	*3390.8*	*7155.0*	*6818.5*	*12055*	*34512*	*19583*	*10009*	*3257.4*	*2934.4*	*3048.2*	*3008.3*
D	–	–	–	–	–	*99*	*100*	*99*	*88*	*77*	*100*	*94*	*100*	*100*	*100*	*99*	*99*
E	–	–	–	–	–	*4811.1*	*2472.3*	*1390.3*	*660.0*	*529.0*	*34430*	*17374*	*9067.4*	*2806.7*	*2315.8*	*1975.2*	*1588.1*
F	–	–	–	–	–	*4880.2*	*3390.8*	*2223.4*	*738.4*	*592.0*	*34512*	*18201*	*10009*	*3257.4*	*2934.4*	*2024.4*	*1647.1*