Technical Reports on Information and
Computer Science from Kochi
Vol. 3 (2011), No. 3

ソフトウェアアルゴリズムの並列化手法の研究

Parallelization Methodology of Software Algorithm

中田　有哉　　村岡道明
Yuuya Nakata　　Michiaki Muraoka
情報科学コース　村岡研究室

1. まえがき

従来では、C 記述上で並列実行による高速化を目的としたソフトウェアでの分割を行う際に、実行ライン数、呼び出し回数から CPU 上での実行時間を推定し、ソフトウェア分割箇所を洗い出していた。しかし、この方法ではターゲット CPU 上の実行時間を正確に算出して、分割するのが困難であった。

本研究では、ソフトウェアの並列実行を行う際に、従来では難しかったソフトウェア CPU 上での実行時間を推定した上で効率的なソフトウェア分割を提案する。また、実行時間のボトルネックになっている箇所に対して並列化を行い、高速化を行う。

2.ソフトウェア分割手法

本手法は、次の 5 ステップより構成される。

(1) 時間精度付きモデルの作成：C 言語で記述されたアルゴリズムを対象として、時間精度付きモデル(時間精度付きモデル：C コードにターゲットプロセッサの実行サイクル数を付与したもの)を作成する。
(2) シミュレーション：(1)で作成した時間精度付きモデルのシミュレーションを行い、プロファイリングに必要なデータを算出する。
(3) プロファイリング：(1)で作成した時間精度付きモデルと(2)で算出したデータを用いて、そのターゲットプロセッサにおけるアルゴリズムの内部動作やテーブル動作回数、アクセス頻度、サイクル数などを求める。
(4) 並列化向き分割：(3)の性能結果から、並列動作によりソフトウェア上での高速化を図る。
(5) 性能評価：ソフトウェア分割手法を行ったモデルを、(1)時間精度付きモデルの作成、(2)シミュレーションを行い、プロファイリングから分割後のモデルを性能評価する。目標とした結果が得られない場合(4)へ戻る。

3.提案手法の適用

今回提案する本手法を暗号化アルゴリズム AES に適用した。本手法を適用するにあたり、Visual Spec(version 4.1.6)上でプロファイリングモデルを作成し、そのターゲットプロセッサとして ARM946E-S を使用し、コンパイラは arm-elf-gcc を使用した。クロック周波数は、25MHz とした。　プロファイリング結果より、AES 暗号のボトルネック部分は MixColumns 処理と SubBytes 処理であり、この２つの逐次的処理をそれぞれ 4 並列化する。

4.ソフトウェア並列化の結果

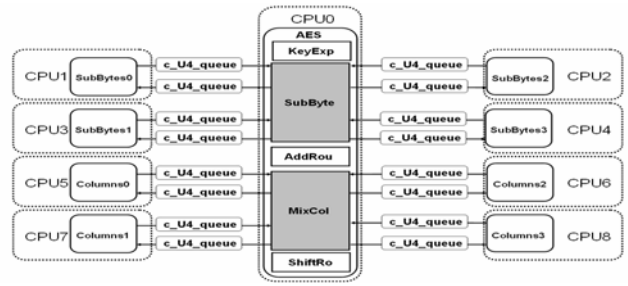

図１　マルチコア前提の並列化手法

図 1 は、各処理をそれぞれ AES, Columns0~3,SubBytes 0~3 に分割を行い、CPU0~CPU9 上で動作する事を前提においてマルチコアプロセッサを割り当てる。また、Columns0~3,SubBytes0~3 の各処理は AES とのデータ転送は queue を用いて行っており、データ転送は 4 バイト単位で行っている。

表１　AES のプロファイリング

ファンクション名	ライン数		呼び出し回数	処理サイクル数		累積サイクル数	
	*1	*2		*1	*2	*1	*2
KeyExpansion	69	61	1	1741	1741	1741(11.82%)	1741(18.93%)
AddRoundKey	90	82	11	68	68	748(5.08&%)	748(8.13%)
SubBytes	268	195	10	179	103	1790(12.16%)	1030(11.20%)
ShiftRows	104	96	10	60	60	600(4.07%)	600(6.52%)
MixColumns	206	143	9	1051	520	9503(64.54%)	4612(50.15%)
						14725(100%)	9196(100%)

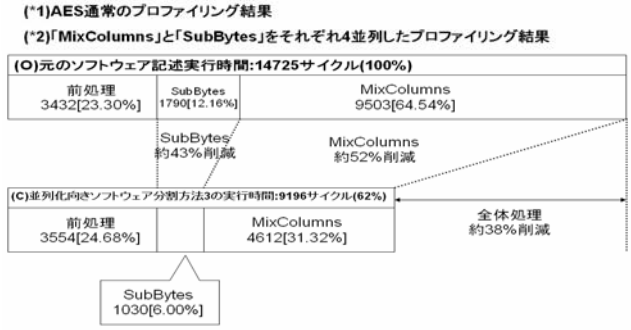

図２　性能比較

今回 SubBytes 処理と MixColumns 処理の 4 並列化を提案する事によって、表 1 と図 2 の結果より SubBytes 処理の実行時間について約 43%削減ができ、MixColumns 処理の実行時間については約 52%削減する事ができた。全体の実行時間は、約 38%削減することができた。

5.まとめ

今回ソフトウェアの並列化手法を提案し、本手法を暗号化アルゴリズム AES に適用し、評価を行った。

本手法では、AES アルゴリズムの中で、SubBytes 処理と MixColumns 処理について並列化向きソフトウェア分割手法を行った。その結果、従来のサイクル数と比較すると、全体の実行時間は、約 38%削減することができた。

今後の課題として、今回データの受け渡しの転送においてキューを使用したが、共有メモリなども考えられる。また、処理サイクル数が同じくらいの大きさに対してはソフトウェアパイプライン化の提案も考えられる。最後にソフトウェア合成技術に関しての自動化も今後の課題である。

参考文献

[1]松永惇弥、”ソフトウェア並列化を考慮したハードウェア/ソフトウェア分割手法の評価”平成 21 年度高知大学修士論文, 2010 年 3 月

[2]松永惇弥、村岡道明、荒木大、”ソフトウェア並列化を考慮したハードウェア/ソフトウェア分割手法の評価”電子情報通信学会技術研究報告、vol. 109, No. 393, VLD2009-71, pp13-18 2010 年 1 月