4Y-7

UltraSPARC T2 における暗号モジュールのオフローディング方式の実装と評価

村上智祐† 笠原竜大† 齋藤孝道‡

明治大学大学院† 明治大学‡

1 はじめに

SSL/TLS や IPSec を利用した通信には，暗号処理という高負荷な演算が伴うため，暗号処理を専ら行うハードウエアモジュール（以降，暗号モジュールと呼ぶ）が普及してきた．このような背景の中，Oracle 社の UltraSPARC T2 が登場した．UltraSPARC T2 は 8 コアの CPU であり，それぞれのコアに，共通鍵暗号方式や公開鍵暗号方式，ハッシュ処理などをサポートする暗号モジュールを搭載している．

UltraSPARC T2 の暗号モジュールを利用するためのフレームワークとして，Solaris10 から PKCS#11 ライブラリ（後述）が提供されている．しかし，PKCS#11 を利用する場合，プログラマは直接暗号モジュールの制御ができないため，暗号モジュールの利用効率は PKCS#11 に依存することになる．そこで，UltraSPARC T2 の暗号モジュールの制御を直接行うため，OCF（OpenBSD/FreeBSDCryptographicFramework）[1]を Solaris に移植し，OCF を Solaris 上で動かすためのカーネルモジュール（以降，実装モジュールと呼ぶ）の実装が提案された[2]．これにより，羅らによる従来の方式より高速に暗号処理を行うことができたが，プロセス数の増加によるプロセスの切り替えが頻繁に発生し，条件によってはスループットが低下してしまうことが分かった．

本論文では，プロセスの切り替えの増加によるスループットの低下を抑えるため，OCF にプロセスを管理するスケジュール機能を実装し，その評価を行った．

2 UltraSPARCT2 のアーキテクチャ

UltraSPARCT2 のアーキテクチャを図 1 に示す．

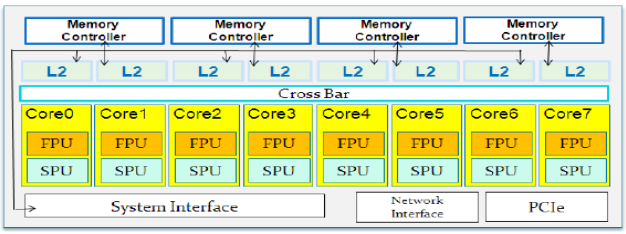

図 1：UltraSPARC T2 プロセッサ

An Implementation and its Evaluation of an Effective Off-loading Hardware Cryptographic Module on UltraSPARC T2

†Murakami Tomosuke, Ryuta Kasahara,
‡Takamichi Saito
Graduate School of Meiji University(†), Meiji University(‡)
1-1-1, Higashimita, Tama-ku, Kawasaki-shi, Kanagawa 214-8571, Japan(†)(‡)
{ce06039, ce06013}@meiji.ac.jp, saito@cs.meiji.ac.jp

UltraSPARC T2 プロセッサは 1 つのプロセッサに 8 つのコアが搭載されている．それぞれのコアは，論理的に 8 つの CPU として動作するため，最大 64 個のスレッドを同時に実行可能である．また，各コアごとに暗号処理専用ユニットである SPU（StreamProcessingUnit）と浮動小数演算ユニットである FPU（Floating point/GraphicsUnit）を搭載している．これらはメインメモリを共有しており，各コアが独立して動作できるだけでなく，コア内の汎用処理とは別に SPU 及び FPU は独立に動作できる．SPU は，主に MAU（ModularArithmeticUnit）と暗号/ハッシュユニットから構成される．MAU は公開鍵暗号化方式をサポートしており，FPU を利用して，RSA 及び楕円曲線暗号を処理できる．暗号/ハッシュユニットは，共通鍵暗号化方式やハッシュ関数に対応しており，DES，3DES，AES，RC4，SHA1，SHA256，MD5 を利用できる．

3 PKCS#11

PKCS#11 は，Solaris10 より提供されている暗号モジュールを利用するためのフレームワークである．PKCS#11 では，アプリケーションからのインターフェースとなるライブラリとして libpkcs11.so[3]を用意しており，これを用いて，暗号モジュールへ処理をオフロードすることができる．また，暗号モジュールの利用状況を監視し，暗号処理の実行対象を決定するスケジューラ／ロードバランサと，暗号モジュールを制御するためのデバイスドライバである暗号化プロバイダを備えている（図 2）．

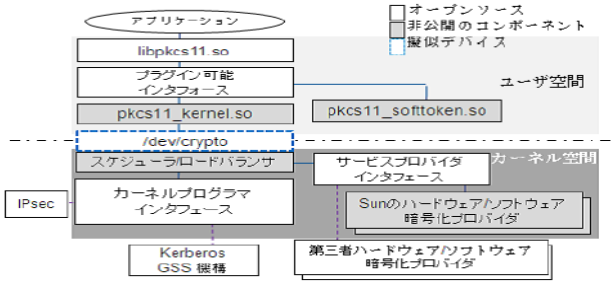

図 2：PKCS#11 暗号フレームワーク

4 OCF

OCF とは，OpenSSL[4]などを利用したアプリケーションから，様々なハードウェアアクセスラレータが提供する暗号処理機能を利用するための共通のインタフェースを提供する API と，ハードウェアアクセラレータのデバイスドライバから構成されたミドルウェアである．

OCF がサポートしていない暗号モジュールに対しては，それに対応するデバイスドライバを用意することで，OCF からの利用が可能となる．

5 提案システム

プロセスの切り替えの増加によるスループットの低下を抑えるため，暗号処理を行うユーザプロセス （以降，暗号プロセスと呼ぶ） を制御するための FIFO のリスト（以降，制御リストと呼ぶ）を用いて，カーネル内で実行できる暗号プロセス数を制限する機能を OCF 内に今回導入した．

5.1 提案システムの詳細

制御リストは暗号プロセスの情報を格納する構造体（以降，制御構造体と呼ぶ）により構成されている．制御構造体は，制御リストに繋げるための構造体，暗号プロセスを特定するためのプロセス ID を格納する変数，待ち状態に遷移するときにつながれる wait キューをメンバとして持つ．

OCF にリクエストを出した暗号プロセスは，制御リストに登録され，先に登録した暗号プロセスあった場合，それが終えるまで待ち状態となる．

提案システムでは 64 個の制御リストを設け，実行できる暗号プロセス数を UltraSPARC T2 で同時実行可能な最大数（64 個）に制限し，制御リストへの登録作業を並列化させることを可能にした．

5.2 提案システムの動作例

ここでは，OpenSSL を用いた暗号プロセスの暗号処理を，OCF と実装モジュールを介して，暗号モジュールへオフロードする場合の動作例を図 3 中の番号と対応させて説明する．

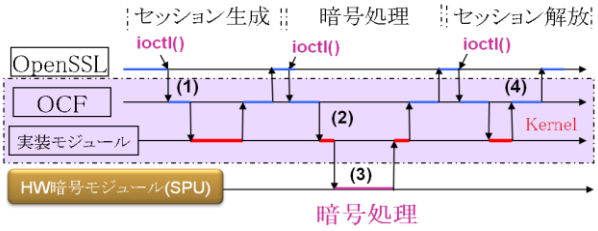

図 3：処理の詳細

(1) OpenSSLとOCFの間でセッションを確立する．OpenSSLとOCF間でセッションID，暗号方式・暗号モード，暗号化/復号鍵を共有する．セッションの確立時に制御リストが空ならば，暗号プロセスを制御リストに繋ぎ次の処理に移る．制御リストが空でなければ，暗号プロセスを制御リストにつないだ後，waitキューにつなぎ，待ち状態に遷移させる．

(2) OpenSSLは，OCFにセッションIDと平文，IVを渡す．さらに，実装モジュールがセッションIDに対応した暗号鍵，OpenSSLから受け取った平文及びIVをHW暗号モジュールで使用できる形式に変換する．

(3) 暗号モジュールは暗号処理を行う．ここで，暗号処理が終わると，ハードウェア割り込みが発生し，結果を OCF 経由で OpenSSL へ返す．

(4) セッションの解放を行う．自暗号プロセスを制御リストから削除し，制御リストの先頭の暗号プロセスを実行可能状態に遷移させる．

6 評価

6.1 計測環境

実装モジュールの評価のために，表 1 に示す環境で，本論文で実装した OCF（以降，OCF-sched と呼ぶ），羅らによる OCF 版（以降，既存 OCF と呼ぶ），PKCS#11 ライブラリをそれぞれ使用した場合の処理時間を計測し，比較した．

表 1：評価環境

CPU	1.2GHz UltraSPARC T2 (コア)	メモリ	16GB
カーネル	Solaris kernel build 117 [5]	その他	OCF-20041201 OpenSSL-0.9.8a
OS	Solaris Express		

6.2 計測項目

実装モジュールの性能評価として，EVPAPI を使用した計測用コードを用いた．計測用コードは，fork()関数により暗号プロセスを複数生成し，それぞれの暗号プロセスが 10Kbyte のデータを 1024 回（合計 10Mbyte）暗号処理する．計測方法は，gettimeofday()関数を使用し，暗号プロセスの生成から暗号プロセスの終了までを処理時間として，単位時間当たりのスループットを求めた．計測はそれぞれ 10 回行いその平均値を結果とした．また，計測に用いる暗号方式は，AES の CBC モードである．図 4 に計測用コードによる計測結果を示す．

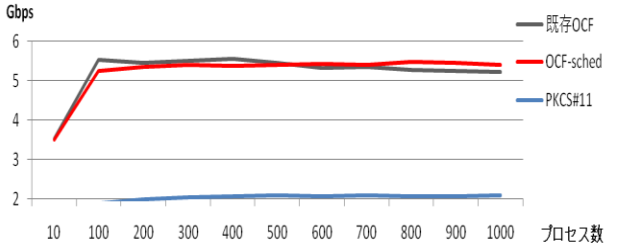

図 4：暗号処理スループット

7 まとめ

本論文では，暗号処理を行うユーザプロセスを管理するスケジュール機能を実装し，その評価を行った．その結果，プロセスの切り替えの増加に伴うスループットの低下を抑えることができた．

謝辞 本研究の成果の一部は，科学研究費補助金（課題番号22700086）の助成を受けたものである．

参考文献

[1] http://ocf-linux.sourceforge.net.
[2] 羅鏡栄，他，UltraSPARC T2 における暗号モジュールの利用と評価．
[3] http://docs.sun.com/.
[4] http://www.openssl.org/.
[5] Solaris kernel build 117 ソース，http://dlc.sun.com/osol/on/downloads/b117