

マルチコア *AnT* におけるサーバプログラム間通信性能の評価

寺本 大風† 鴨生 悠冬‡ 佐藤 将也‡ 谷口 秀夫‡
 †岡山大学工学部 ‡岡山大学大学院自然科学研究科

1 はじめに

マイクロカーネル構造 OS では、OS サーバ間の通信 (以降、サーバプログラム間通信) を高速化することが重要である。マルチコアに対応したマイクロカーネル構造 OS である *AnT* オペレーティングシステム (以降、*AnT*) では、ページテーブルの書き換えによる複写レスなデータ授受機構とコア毎のリングバッファを用いた排他制御機構により、高速なサーバプログラム間通信を実現している。本稿では、*AnT* の持つ高速なサーバプログラム間通信機構の評価結果を述べる。

2 サーバプログラム間通信機構

2.1 基本機構

AnT のサーバプログラム間通信機構では、通信内容をコア間通信データ域 (ICA : Inter-core Communication Area) と呼ばれる共用領域に格納し、OS サーバ間でのデータ授受を複写レスで実現している [1]。以降、制御用情報を格納した ICA を制御用 ICA、データ情報を格納した ICA をデータ用 ICA と呼ぶ。サーバプログラム間通信の処理流れを図 1 に示し、以下に説明する。

- (1) 依頼元プロセスの処理依頼により、OS サーバの依頼用リングバッファに依頼情報を格納した制御用 ICA の先頭アドレスを登録し、依頼元プロセスの仮想空間から制御用 ICA を剥がす。
- (2) OS サーバは、依頼用リングバッファから制御用 ICA を取得し、自 OS サーバの仮想空間に制御用 ICA を貼り付け、依頼された処理を実行する。
- (3) OS サーバの結果返却により、依頼元プロセスの結果用リングバッファに結果情報を格納した制御用 ICA の先頭アドレスを登録し、自 OS サーバの仮想空間から制御用 ICA を剥がす。
- (4) 依頼元プロセスは、結果用リングバッファから制御用 ICA を取得し、自プロセスの仮想空間に制御用 ICA を貼り付け、処理結果を受け取る。

2.2 マルチコア環境における通信機構

マルチコア環境において、*AnT* はコア毎にカーネルを配置している。このため、他コア上の WAIT 状態の

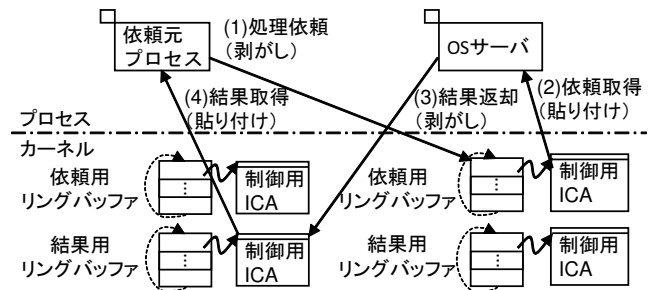


図 1 サーバプログラム間通信の処理流れ

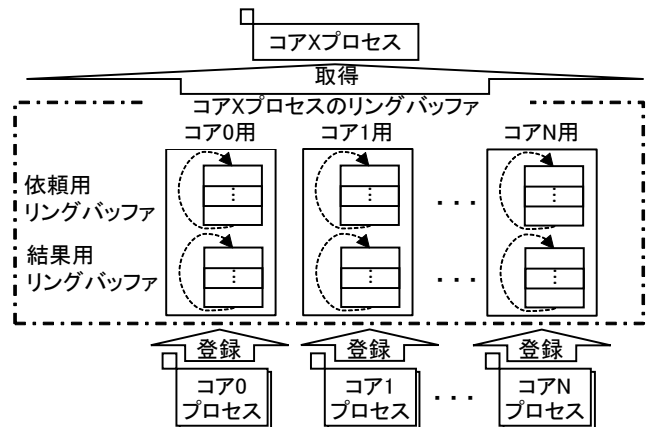


図 2 コア毎のリングバッファを用いた制御構造

OS サーバに対して処理依頼を行う場合、依頼先のコアに IPI (Inter-Processor Interrupt) を送信することで依頼内容の登録を通知する。また、複数のプロセスが同時に1つのリングバッファに対して処理依頼を行うことを回避する。このため、*AnT* では、コア毎に用意したリングバッファを用いた制御構造を実装している。この制御構造を図 2 に示す。この制御構造では、1つのプロセスが依頼用リングバッファと結果用リングバッファをコア数分保有し、登録を行うプロセスは、自身の走行コアに対応したリングバッファに対して依頼または結果を登録する。

3 評価

3.1 内容

同コア内の通信と別コア間の通信について、同期処理と非同期処理の処理時間を測定した。同期処理でのサーバプログラム間通信における依頼元プロセスと OS サーバの処理流れを図 3 に示す。なお、ICA への初アク

Performance Evaluation of Communication between Server Programs for Multi-core *AnT*
 Hirokaze Teramoto†, Yuuto Kamou‡, Masaya Sato‡, Hideo Taniguchi‡
 †Faculty of Engineering, Okayama University
 ‡Graduate School of Natural Science and Technology, Okayama University

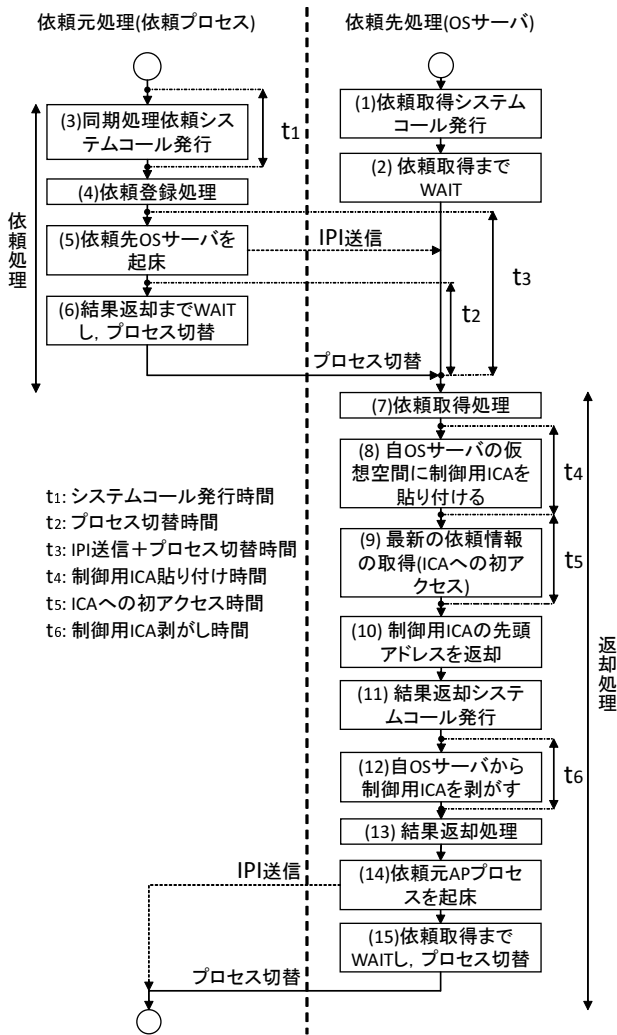


図3 サーバプログラム間通信の処理流れ(同期処理)
 処理時間は、プロセスにICAが貼り付けられた後、初めてそのICAにアクセスした際に発生するオーバーヘッドである。これは、TLBミス等のキャッシュミスによるものと推察できる。

Intel Xeon E5-2630 v3 (2.4GHz) を搭載した計算機を利用し、測定には RDTSC 命令を使用した。処理時間は10回の平均処理時間である。なお、本測定ではサーバプログラム間通信の処理時間のみを測定するため、処理依頼においてOSサーバへ渡す引数と戻り値は無しとし、OSサーバでは通信処理以外の処理を行わないものとした。また、データ用ICAの授受はなく、制御用ICAの授受のみを行った。

3.2 結果と考察

サーバプログラム間通信の処理時間を図4に示す。図4より、以下のことが分かる。

- (1) 依頼処理時間は、同コア、別コア共に非同期処理が同期処理より約 1.0μ 秒長い。これは、非同期処理依頼は同期処理依頼と異なり、依頼元プロセスの仮

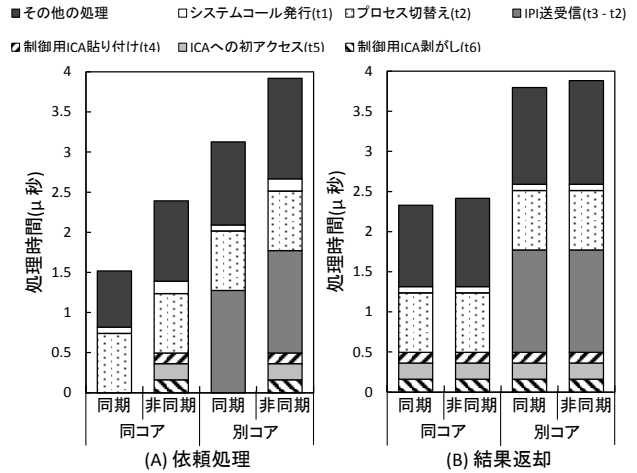


図4 サーバプログラム間通信の処理時間

- 想空間からの制御用ICAの剥がしとOSサーバから返却された制御用ICAの取得を行うためである。
- (2) 返却処理時間は、同コア、別コア共に同期処理と非同期処理でほぼ同じ処理時間になる。これは、返却処理の処理流れが同じであるためである。
 - (3) 依頼処理と返却処理の両方において、別コア間の処理時間は同コア内での処理時間よりも約 1.5μ 秒長い。これは、別コアのプロセスを起床させるために行うIPIの送信に起因する。
 - (4) 依頼処理時間は約 $1.5 \sim 4.0 \mu$ 秒であり、返却処理時間は約 $2.3 \sim 3.8 \mu$ 秒である。したがって、サーバプログラム間通信処理の処理時間は最大で約 7.8μ 秒であり、OSサーバの固有処理の時間が約 $1m$ 秒であれば、サーバプログラム間通信によるオーバーヘッドは1%未満に抑えられる。

4 おわりに

マルチコアに対応した *AnT* オペレーティングシステムについて、サーバプログラム間通信機構の評価を述べた。同期処理は、制御用ICAの剥がしと返却結果を格納した制御用ICAの取得を行わないため非同期処理よりも高速な処理依頼が可能であることを示した。また、コア間でのサーバプログラム間通信の場合、IPIの送受信等によって約 1.5μ 秒のオーバーヘッドが発生することを示した。最後に、サーバプログラム間通信処理の処理時間は最大で約 7.8μ 秒であり、OSサーバ固有の処理が約 $1m$ 秒であれば、サーバプログラム間通信によるオーバーヘッドを1%未満に抑えられることを示した。

参考文献

- [1] 岡本 幸大, 谷口 秀夫: *AnT* オペレーティングシステムにおける高速なサーバプログラム間通信機構の実現と評価, 電子情報通信学会論文誌 D, Vol.J93-D, No.10, pp.1977-1989 (2010).