

AnT の要求箱通信機能を利用した処理の多重化

本田 紘貴[†] 佐藤 将也[‡] 山内 利宏[‡] 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 機能の利用の際、OS サーバ間でのサーバプログラム間通信を行う。マイクロカーネル構造 OS である *AnT* は、サーバプログラム間通信機能として要求箱通信機能[1]を持つ。要求箱通信機能では、依頼プロセスは要求箱に依頼を登録することで、OS サーバを特定することなく処理を依頼できる。本稿では、要求箱通信機能を利用した処理の多重化について述べる。

2 サーバプログラム間通信機構

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）と呼ばれる共有領域に格納し、OS サーバ間でのデータ授受を複写レスで実現している[2]。*AnT* の2つのサーバプログラム間通信機能を図1に示し、以下で説明する。

(1) 基本通信機能[2]: 依頼プロセスは、OS サーバの依頼キューに処理情報を格納した ICA（以降、制御用 ICA）を登録することで、処理を依頼する。OS サーバは、自身の依頼キューから制御用 ICA を取得し、処理を実行する。結果返却は、依頼プロセスの結果キューに制御用 ICA を登録する。

(2) 要求箱通信機能[1]: 依頼プロセスは、要求箱の依頼キューに制御用 ICA を登録することで、処理を依頼する。OS サーバは、自身が取得可能な要求箱の依頼キューから制御用 ICA を取得し、処理を実行する。結果返却は、基本通信機能の結果返却と同様に、依頼プロセスの結果キューに制御用 ICA を登録する。

3 要求箱通信機能を利用した処理の多重化

3.1 基本方式

要求箱通信機能を利用し、多重度に合わせて影 OS サーバ（親と子）を用意する。これにより、依頼プロセスが意識せずに同じ処理を多重に実行できる機能を実現する。この様子を図2に示し、処理の流れを以下に述べる。

- (1) 依頼プロセスは、要求箱に制御用 ICA を登録する。
- (2) OS サーバは、要求箱から制御用 ICA を取得し、処理を実行し、依頼プロセスへ結果返却する。
- (3) 影 OS サーバ（親）は、要求箱から制御用 ICA の複写を取得する。その後、多重度に応じて、 n 個の影 OS サーバ（子）へ取得した制御用 ICA と同じ

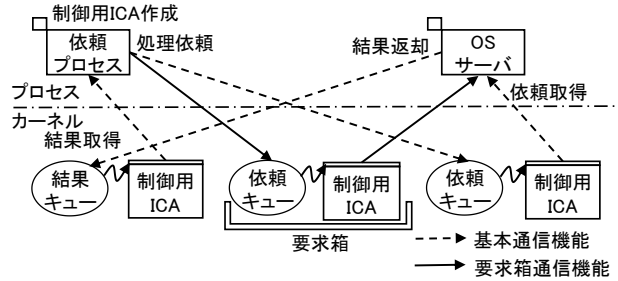


図1 基本通信機能と要求箱通信機能

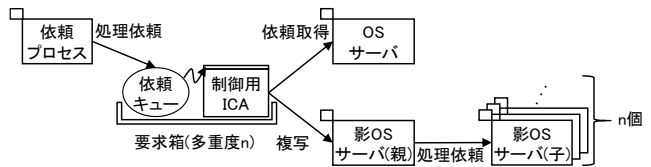


図2 処理の多重化

処理内容の処理依頼を行う。

(4) 影 OS サーバ（子）は、処理を実行し、影 OS サーバ（親）へ結果返却する。

(5) 影 OS サーバ（親）は、結果取得後、制御用 ICA を解放し、処理を終了する。つまり、依頼プロセスへの結果返却は行わない。したがって、依頼プロセスが意識しない形で処理を多重化する。

3.2 課題と対処

処理の多重化には、以下の課題と対処がある。

(1) 要求箱の制御用 ICA を取得する契機を明らかにする必要がある。OS サーバが依頼取得待ちの状態のとき、基本通信機能では依頼プロセスの処理依頼の際に起床できる。しかし、要求箱通信機能では依頼プロセスが OS サーバを特定できない。そこで、OS サーバは、依頼取得待ちを時間待ちの形とし、時間起床により定期的に依頼取得を行う。

(2) 要求箱の1つの制御用 ICA を複数の OS サーバが取得する方法を明らかにする必要がある。複数の OS サーバが独自に処理を進めるため、取得の際に制御用 ICA の複写が必要である。一方、依頼プロセスへの結果返却は OS サーバが行い、影 OS サーバ（親）は行わない。また、*AnT* では、不具合の混入を防ぐため ICA の確保と解放は同じプロセスが行う。そこで、制御用 ICA の複写は、影 OS サーバ（親）が行い、OS サーバが影 OS サーバ（親）より先に制御用 ICA を取得しないように制御する。

(3) 多重度の設定方法を明らかにする必要がある。多重度は、依頼プロセスと OS サーバが意識しない。そこで、影 OS サーバ（親）が認識でき、かつ影 OS サーバ（親）に関係なく自由に設定できるように、要求箱が保持する。

Multiplexing of Process by *AnT* Request Box
 Hiroataka Honda[†], Masaya Sato[‡], Toshihiro Yamauchi[‡], Hideo Taniguchi[‡]
[†]Faculty of Engineering, Okayama University
[‡]Graduate School of Natural Science and Technology, Okayama University

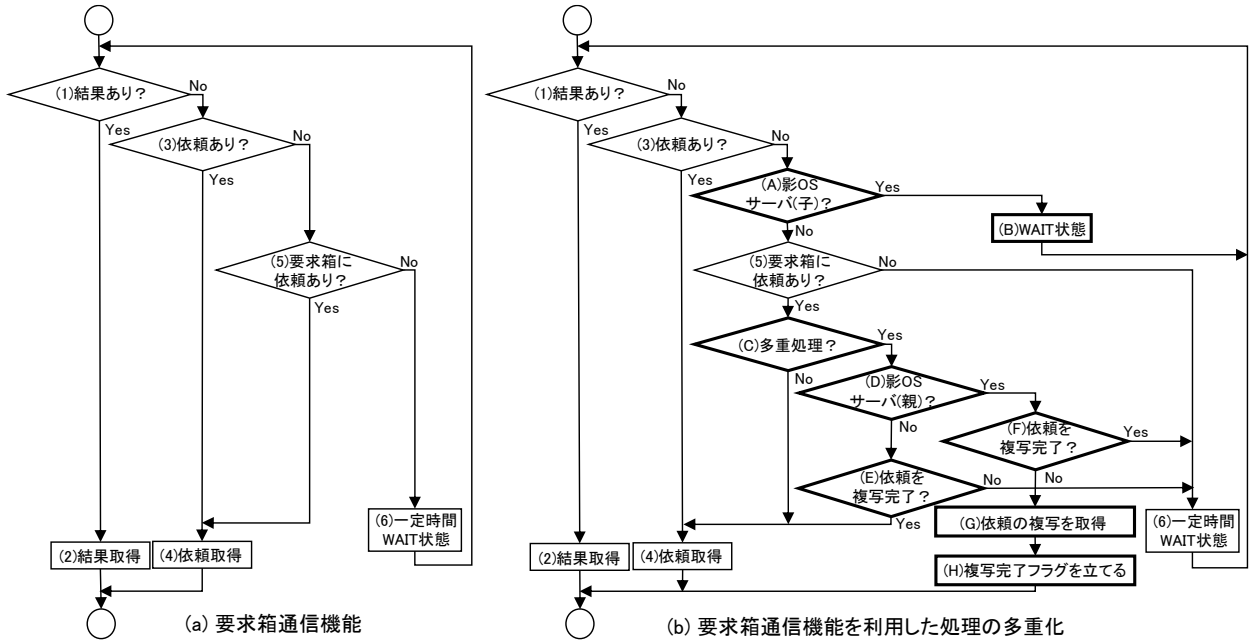


図3 依頼取得の処理流れ

3.3 依頼取得の処理流れ

OS サーバの依頼取得の処理流れを図3に示す。要求箱通信機能では、OSサーバは、自身の結果キュー、自身の依頼キュー、取得可能な要求箱の依頼キューの順で探索を行い、結果または依頼が登録されていれば、結果取得または依頼取得を行う。要求箱の依頼キューに取得可能な依頼がなければ、一定時間 WAIT 状態に遷移する。

処理の多重化における依頼取得の処理流れは、要求箱通信機能の処理流れにおいて、要求箱に関する部分の処理を変更する。変更箇所は処理 (A) ~ (H) である。

処理 (A) と処理 (B) では、自身が影 OS サーバ (子) の場合は、要求箱の制御用 ICA を取得する必要がないため、要求箱から制御用 ICA を取得しないように制御している。処理 (C) では、多重処理を行わない場合は、既存の要求箱通信機能を利用できるように制御している。また、影 OS サーバ (親) が要求箱にある制御用 ICA の複写を取得したことを確認するために、複写完了フラグを導入する。このフラグが立っていれば、影 OS サーバ (親) が要求箱にある制御用 ICA の複写を取得したことを示す。複写完了フラグを利用した処理は、処理 (D) ~ (H) である。複写完了フラグにより、OS サーバが要求箱にある制御用 ICA を取得するかの判定および影 OS サーバ (親) が要求箱にある制御用 ICA の複写を取得するかの判定を行う。

4 評価

依頼プロセスが要求箱に処理依頼を行い、結果取得を行うまでの処理時間について、多重度、および OS サーバと影 OS サーバ (親と子) の優先度との関係について述べる。なお、OS サーバと影 OS サーバ (子) は、1ms の PU 処理を行い、一定時間 WAIT

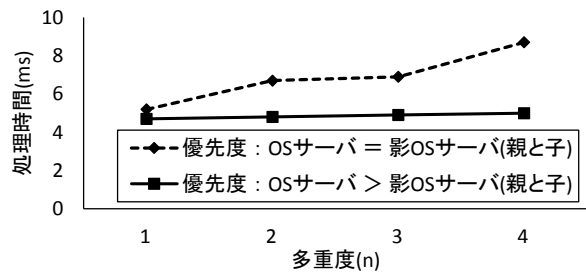


図4 評価結果

は 5ms とした。測定には、Intel Core i7-2600 (3.4GHz) (使用 1 コア) を搭載した計算機を利用した。

評価結果 (図4) より、以下のことがわかる。

- (1) OS サーバと影 OS サーバ (親と子) の優先度が同じ場合、多重度が増加するにつれて、処理時間は長くなる。これは、OS サーバ処理と同時に影 OS サーバ (親と子) 処理が行われるためである。したがって、多重度が増加するにつれて影 OS サーバ (子) の個数が増え、処理時間は長くなる。
- (2) OS サーバの方が影 OS サーバ (親と子) より優先度が高い場合、多重度に関わらず、処理時間にほとんど変化はない。これは、OS サーバが影 OS サーバ (親と子) より優先して実行されるためである。

5 おわりに

要求箱通信機能を利用した処理の多重化について述べた。具体的には、3つの課題と対処を示し、処理の流れを述べた。評価より、多重度が OS サーバの処理時間に影響を与えるものの、優先度の設定により回避できることを示した。

参考文献

[1] 寺本大風, 佐藤将也, 山内利宏, 谷口秀夫: OS サーバ処理の負荷分散を可能にする *AnT* の要求箱通信方式, 平成 29 年度電気・情報関連学会中国支部第 68 回連合大会 (2017.10) .
 [2] 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 (2010.10) .