

実行中プログラム部分入替え法における入替え時間の評価

5 Q - 3

— 他プロセスの影響 —

中島 雷太 谷口 秀夫
九州大学大学院システム情報科学研究科

1 はじめに

我々は、実行中プログラムの一部分を入れ替える制御法を提案した [1]。また、入替え条件を緩和した際の問題点を、入替え時間の観点から評価を行なうことにより明らかにした [2]。これらの評価は、入替え対象プログラム部分を共有しているプロセスのみが走行している場合について、行なっている。しかし、実際には、それ以外のプロセス（以降、「他プロセス」と呼ぶ）が、同時に走行している状況を十分に考えることができる。

そこで、本稿では、プロセッサ処理を主体とするプログラムの部分入替えにおける、他プロセスも同時走行している場合の入替え時間について、評価を行ない、結果を報告する。

2 入替え法

実行中プログラムの部分入替え法を実現する上での課題の中で、プログラム実行状態の分類、入替え可能条件および入替え方法について、文献 [1] を基に、簡単に説明する。

入替え対象プログラム部分の実行状態を、三通りに分類する。入替え対象プログラム部分が、実行される前、あるいは実行された後の状態を「未使用」と呼び、実行中の状態を「走行中」と呼ぶ。また、別のプログラム部分呼び出している状態の時を「呼出中」と呼ぶ。

入替え条件として以下のものがある。

- (条件 1) 呼び出しと返却のインタフェースの一致
 - (条件 2) 処理の矛盾の回避
 - (条件 3) 戻り値のアドレスを変更しない
 - (条件 4) スタティックリンクの場合、メモリ上の外部変数の参照アドレスが同じであること
 - (条件 5) 外部変数の値が入替えの前後で同じであることの保証が必要か否か
 - (条件 6) アドレス渡しによる外部変数や内部変数の参照や更新がどのように行なわれているか
- 入替え対象プログラム部分の実行状態が、「未使用」の場合には、入れ替えようとしている内容は、(条件 1) と (条件 2) を満たしていれば良い。「実行中」の時は、内部変数など過渡的な状態であるため、入替えを行なうことはできない。「未使用」と「呼出中」の場合には、(条件 1) から (条件 6) を満たしていなければならない。

入替えは、次のようにして行なう。入替え要求がある時、入替え対象モジュールに走行を移そうとしているプロセスを停止させ、その他のプロセスを走行させ続ける。その後、入替え対象プログラム部分が、入替え可能状態であることを検知した直後に、入替えを実施する。

3 評価と考察

3.1 測定条件

測定に用いたプログラムの流れを図 1 に示す。図 1 に

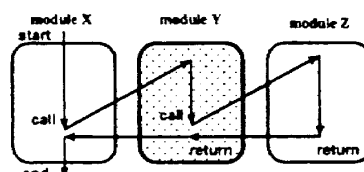


図 1 プログラムの処理の流れ

において、module Y を入替え対象モジュールとする。各モジュールの処理時間の合計を 10 秒固定とし、module X の処理時間を 1 秒、module Y の処理時間を 1 秒から 8 秒まで変化させ、八つの組合せ (type A、B、…、H) について測定した。

プロセスは、疑似乱数を用いて不規則で起動し、共有プロセス数は 1 から 10 までとした。また、特定変数をインクリメントする処理を繰り返すプログラムを他プロセスとして一つ起動し、この時の入替えを行ない、その入替え時間を測定した。測定は 20 回ずつ行ない、それらの平均値を測定結果とした。また、入替え対象プログラムの状態把握は、入替え対象プロセスの起動時に行なう [1] こととした。

3.2 結果と考察

プログラムを共有しているプロセス数と入替え時間の関係について、図 2 に入替え可能条件が「未使用」または「呼出中」の場合を、図 3 に入替え可能条件が「未使用」のみの場合を示す。図 2 と図 3 のいずれも、プログラムを共有しているプロセス数が多くなると、入替え時間はほぼ単調増加する。これは、プロセス数が多くなると、プロセッサ処理開始のための READY 状態での待ちが発生し、入替え時間に影響を与えるためである。図 2 では、各 type 毎に傾きが異なるが、図 3 では、全 type が同様な値になっている。これは、入替えができない実行状態であるモジュールの処理時間の総和が異なるためである。

次に、入替えができない実行状態であるモジュールの処理時間の総和 T_m が、他のモジュール処理時間に比べ

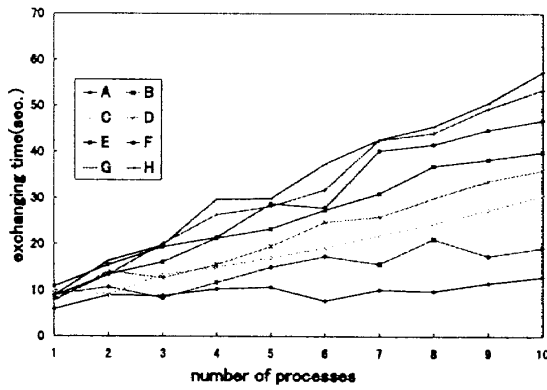


図2 プロセス数と入替え時間の関係
(「未使用」または「呼出中」の時に入替え可能とした場合)

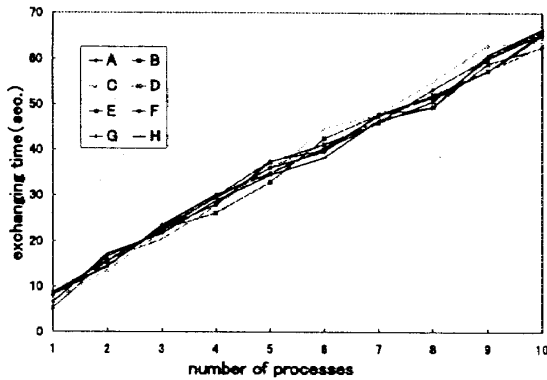


図3 プロセス数と入替え時間の関係
(「未使用」の時のみに入替え可能とした場合)

て、十分に長い場合の入替え時間について考察する。他プロセスが走行していない場合の入替え時間 T_0 は、プログラムを共有しているプロセス数を n とすると、平均的には、

$$T_0 = \left(T_m \times \frac{1}{2} \right) \times n \quad (1)$$

となる。一方、各プロセスの READY 状態での待ち時間 T_w は、タイムスライス時間を T_i とすると、平均的には、

$$T_w = T_i \times \left(n \times \frac{1}{2} \right) \quad (2)$$

となる。他プロセスが走行する場合、 T_w 毎に他プロセスの走行 (時間: T_i) が発生する。したがって、他プロセスによる入替え時間への影響時間 T は、

$$T = \frac{T_0}{T_w} \times T_i = \frac{\left(T_m \times \frac{1}{2} \right) \times n}{T_i \times \left(n \times \frac{1}{2} \right)} \times T_i = T_m \quad (3)$$

となる。

図2と図3のそれぞれについて、全く同じ測定条件下で、他プロセスが走行していない場合の入替え時間 [2] との差分を図4と図5に示す。ここで、図5の AVG は、図3で、各プロセス数での全 type の入替え時間の平均値と、他プロセスが走行していない場合の各プロセス数での全 type の入替え時間の平均値の差分を表している。

図4の type H の場合、入替え時間の差分は、プログラムを共有しているプロセス数に関係なく、およそ8秒

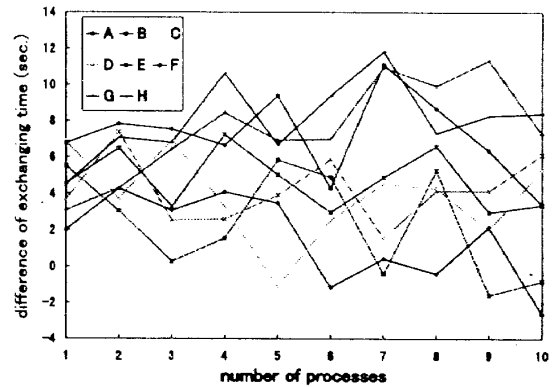


図4 他プロセスの有無による入替え時間の差分
(「未使用」または「呼出中」の時に入替え可能とした場合)

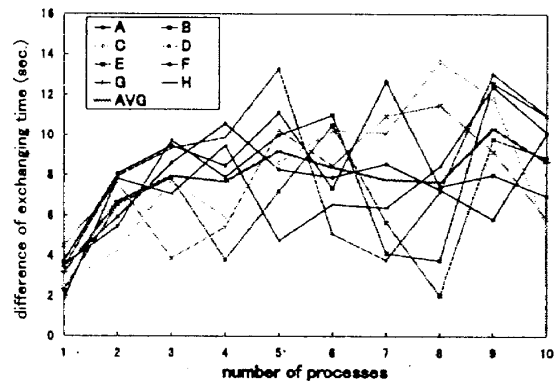


図5 他プロセスの有無による入替え時間の差分
(「未使用」の時のみに入替え可能とした場合)

程度となっている。これは、入替え対象モジュールである module Y の処理時間であり、式 (3) を裏づける結果となっている。また、図5の各 type において、プロセス毎のばらつきが多少あるものの、プログラムを共有しているプロセス数が多くなると、入替え時間の差分は、およそ9秒に近づいていることがわかる。これも、式 (3) を裏づける結果と言える。

図4の type A の場合、入替え対象モジュールの処理時間は1秒と短く、タイムスライス間隔と同じである。このため、READY 状態での待ち時間の影響は、ほとんど見られない。

4 おわりに

他プロセスが同時走行している場合の入替え時間を評価し、入替えに対する影響を示した。入替えができない実行状態であるモジュールの処理時間の総和が、タイムスライス時間よりも長くなると、他プロセスが同時走行する影響が顕在化する。今後は、長大化した入替え時間の短縮化を検討する予定である。

参考文献

- [1] 谷口秀夫, 後藤真孝: “走行中のプロセス間で共有されたプログラムの部分入替え法”, 電子情報通信学会論文誌 (D-I), vol.80-D-I, no.6, pp.495-504, Jun. 1997.
- [2] 中島雷太, 谷口秀夫: “入替え条件を緩和した実行中プログラム部分入替え法の評価”, 情報処理学会 (OS-8), Aug. 1998.