

問010048解説

◆解答

設問1 a オ

設問2 b ア c ウ d イ

設問3 イ

◆解説

スレッドを使用したプログラムの並列処理に関する問題である。

① 並列処理とスレッドの活用

スレッドは、OSがCPUに割り当てる処理の最小単位である。マルチタスクOSでは、1つのプログラムをいくつかの細かなスレッドに分割し、スレッド単位にCPUの処理を割り当て、複数のスレッドを切り替えながら同時に効率的に処理する。

② プログラムの並列実行

コンピュータで処理を実行する場合、通常は、与えられたプログラムステップを最初から最後までひとつずつ順番に実行していく逐次処理が用いられる。一方、SIMD命令をサポートするプロセッサや、複数のプロセッサを搭載するコンピュータでは、複数の異なるデータを同時に処理したり、複数の異なるステップを同時に実行したりすることによって処理の効率化や高速化を図る。

プログラムの実行部分を、データの作成、計算処理、結果出力の3段階に分割し、計算処理の段階で複数のスレッドを生成し、複数のCPUを使用して並列処理が実行される。図1はスレッドによる並列実行の例、図2はマルチプロセッサによるスレッドの実行の例を示している。

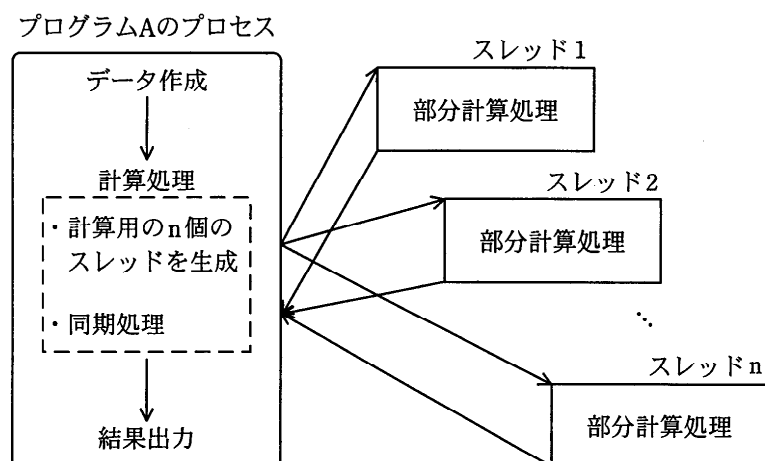


図1 スレッドによる並列実行の例

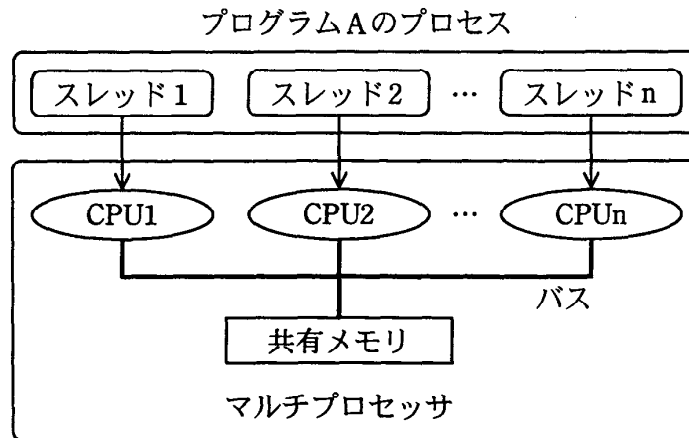


図2 マルチプロセッサにおけるスレッド実行

③ ループ処理と並列処理の可能性

並列化可能なループは、並列処理した場合と逐次実行した結果と差異がないような形態を有するループである。並列処理では、複数のスレッドが存在するため、ループ計算の総量が分割され、各スレッドで分担して処理される。各スレッドは分割されたインデックス番号から、受け持ちのインデックス数分を担当して並行に処理する。このような並列処理においては、ループ増分インデックスが逐次処理のように始点から順番に増分し計算処理していく訳ではない。ループ内のインデックス(周回)がどのような順番で行われても「正しい結果」となるループ内処理が並列化可能ということになる。

㊦ 並列化可能なループ処理

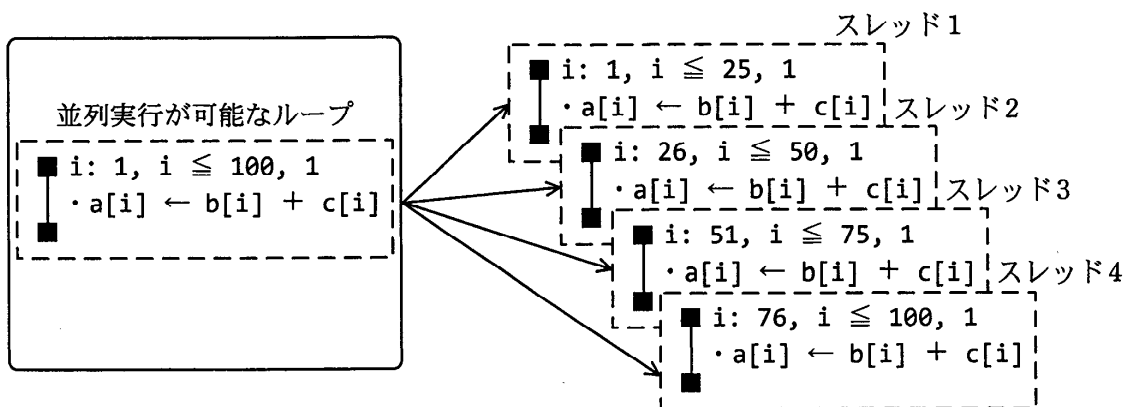


図3 並列実行が可能なループの例

図3の処理は、4つのスレッドに分割して並列処理する場合も、それぞれのスレッドで取り扱うb配列、c配列の各要素の値は、スレッド間で干渉していないため演算結果に矛盾が発生しない。1から100の要素の演算結果は、b配列、c配列の各要素を逐次加算して求めた結果とスレッド分割して並列処理で求めた結果は一致する。

① 並列化不能なループ処理

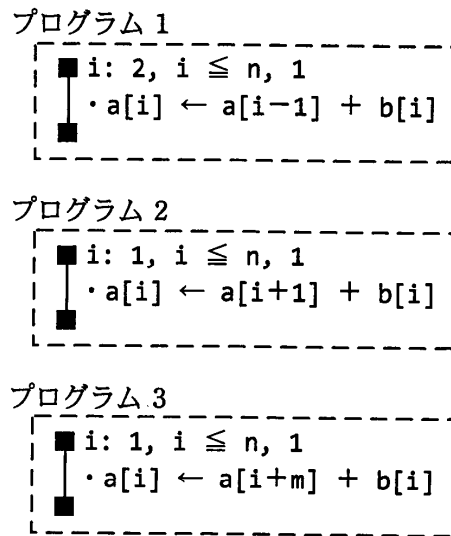


図 4 ループのプログラム 1~3

- ① プログラム 1 のループ処理では、一つ前の更新されたデータを参照して演算が実行される。複数のスレッドに分割された場合、一つ前のデータが一つ前のループで計算された結果であることが必ずしも保証されないため正しい結果が得られるとは言えない。
- ② プログラム 2 のループ処理では、次に更新されるデータを参照して演算が実行される。次のループで参照したデータが更新されるため、現在のループの演算結果との矛盾が生じることとなり正しい結果が得られない。
- ③ プログラム 3 の場合、 $m \geq n$ ならば、参照されるデータは更新の対象にならない配列の値になるため並列実行が可能になる。

④ 並列実行での高速化率

次に示す式を用いて、並列実行での高速化率を計算する。

$$E = \frac{1}{1 - r + \left(\frac{r}{n}\right)}$$

E は並列実行での高速化率、n は並列できる CPU の台数、r は対象とする処理のうち、並列実行が可能な部分の処理時間の割合、 $0 \leq r \leq 1$ である。

設問 1

並列実行での高速化率 E を計算する問題である。

データ作成、計算処理、結果出力の処理時間の割合が 7 : 90 : 3 であるから、対象処理

時間に対する並列実行処理部分の割合 r を求めると

$$r = 90 / (7 + 90 + 3) = 90 / 100 = 0.9$$

$E \geq 0.5$ 、 $r = 0.9$ として n を求めると、次のようになる。

$$\frac{1}{1 - 0.9 + \frac{0.9}{n}} \geq 5$$

変形すると、 $0.5 \geq 4.5 / n$ $n \geq 4.5 / 0.5 = 9$

CPUの台数の9台となる。 a の求める答えはオとなる。

設問2

図4に示すように3個のプログラムに対して並列処理を実行した場合の問題点に関する問題である。

プログラム1のループ処理では、 $a[i] \leftarrow a[i-1] + b[i]$ に示すように一つ前の更新されたデータを参照して演算が実行される。複数のスレッドに分割された場合、一つ前のデータが一つ前のループで計算された結果であることが必ずしも保証されないため正しい結果が得られるとは言えない。 b の答えは更新した値が次の繰り返しで参照されるのでとなり、求める答えはアとなる。

プログラム2のループ処理では、 $a[i] \leftarrow a[i+1] + b[i]$ に示すように次に更新されるデータを参照して演算が実行される。次のループで前のループで参照したデータが更新されるため、現在のループの演算結果との矛盾が生じることになり正しい結果が得られない。 c の答えは参照した値が次の繰り返しで更新されるのでとなり、求める答えはウとなる。

プログラム3の場合、 $m \geq n$ ならば、参照されるデータは更新の対象にならない配列の値になるため並列実行が可能になる。 i の値に関係なく、 $a[i+m]$ の値が並列の分割に関係なく一定の値となる条件は $m \geq n$ となる。 d の求める答えはイとなる。

設問3

配列 a で更新する要素を示すインデックスの値が配列 ip で間接的に決定される。

プログラム4の20個の要素の逐次処理では、配列 a の要素の値は ip の値が1から20の値に対応して決定される。

4つのスレッドに分割して、並列処理される場合は次のようになる。

アの場合、各スレッドの配列の値は、 ip が1～5に対応して間接的に決定されており、逐次処理の ip の値、6～20に対応する値が求まっていない。

イの場合、スレッド1は2、17、14、10、7に対応し、スレッド2は19、16、13、9、6に対応し、スレッド3は20、15、12、8、5に対応し、スレッド3は18、1、11、3、4に対応しすべての要素が1個ずつ対応し重複していない。従って、順

序は変わっても取り扱う数値は矛盾しないことになる。これは並列処理が可能となる条件である。求める答えはイとなる。

ウの場合、1、7、9、12、14、19、20に対応する数値が存在しない。8、13、15が重複している。

エの場合、1、2、3の対応する値がなく、7、14、17が重複している。