

## メモリ共有を用いたシステムコールの並列化手法

T140482 中村 公亮

指導教員：三好 力 教授，芝 公仁 助教

### 1 はじめに

プロセスがファイル I/O やメモリの確保、ソケット間通信などオペレーティングシステムの機能を使用する場合、システムコールが使用される。システムコールが発行されると CPU は特権モードへと切り替わり、カーネルのコードの実行を始める。このとき、プロセスの処理が中断されカーネルに処理が移ることに加え、CPU の実行モードの遷移や TLB フラッシュなどオーバヘッドが発生する。これらを削減するために、システムコールの要求を行なうスレッドと処理を行なうスレッドを分ける手法が考えられる [1]。本稿では、プロセスとカーネルがメモリを共有し、複数の処理を行うスレッドがそこから必要な情報を取りだして、複数個のシステムコール要求を並列に処理する手法を提案する。

### 2 提案手法

従来型のシステムコール発行ではプロセッサの例外を用いて、走行モードの遷移を行ってからカーネルのコードを処理する。提案手法では、カーネル内でシステムコール処理をするための専用スレッド（以降、処理スレッド）が、プロセスごとに動作する。処理スレッドは複数個生成でき、複数個のシステムコール要求に対して並列に処理を行うことができる。また、システムコールの要求には、プロセス内にあり処理スレッドからも読み書き可能な共有メモリを使用する。処理スレッドは共有メモリの先頭から要求を探索し、要求があった場合にはシステムコールの処理を行う。処理スレッドが複数ある場合には、一つの要求を処理スレッド間で競合しないようにする。このように共有メモリを介してシステムコール要求や結果の受け渡しを行い、並列実行を可能にすることでシステムコールを効率的に実行する。

### 3 評価

本手法の効果を確認するために、`/dev/zero` から 4096 バイトを読み出す `read` システムコールを

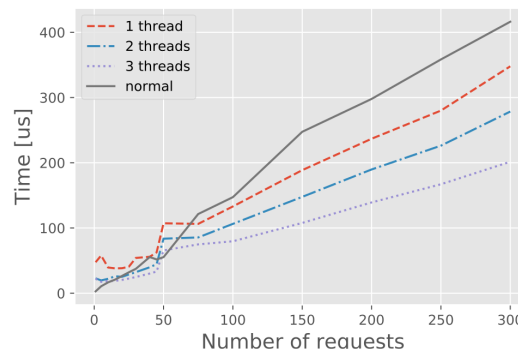


図 1: 提案手法と従来型の処理時間

複数回発行する実験を行った。そのときの処理時間を図 1 に示す。横軸はシステムコールの発行回数であり、縦軸は処理時間である。

発行回数が 50 未満のときは通常の手続きの方が、50 を超えた後は本手法の処理時間の方が短くなる。これは、回数が多くなるにつれ、共有メモリを使用することによるオーバヘッド削減効果が現れるからだと考えられる。また、処理スレッド数が増えると、処理時間の増加が少なくなっている。これは、一つの処理スレッドが一つの要求を処理している間に、他の処理スレッドが別の要求を処理できる並列化の効果だと考えられる。

### 4 おわりに

本稿では、メモリをカーネルとプロセスで共有し、システムコールの要求と処理を分けることで実行を効率化する手法を述べた。評価実験により、提案手法の有効性を検証し、システムコールのオーバヘッドを削減できることを示した。また、処理スレッドの数が増えれば、処理時間の増加が少なくなることを確認した。

### 参考文献

- [1] Livio Soares and Michael Stumm.:FlexSC: Flexible System Call Scheduling with Exception-Less System Calls, *OSDI* (2010)