

Linux カーネルにおける I/O 優先度操作機構の実現

1211075 佐藤 洋佑 岩崎研究室

1 背景

OS 内では、ファイル I/O を多く行うもの、計算処理が中心のものなど、様々なプロセスが存在する。また、I/O 処理待ち状態のプロセスが計算処理を行う場合は、別のプロセスの I/O 処理が終わるまで CPU 資源を割り当てることができない。これによって CPU 資源が有効活用されず無駄となってしまうという問題が起こり得る。

この問題に対して、I/O 優先度の操作によって、CPU 資源の無駄が軽減できる事が示されている [1]。I/O 優先度とは、I/O 要求の処理に割り当てられる優先度の事である。[1] で提案されている方法は、I/O を行う頻度が小さいプロセスの I/O 優先度を高くして I/O を早く終わらせて CPU を有効活用するというものである。しかし、Linux カーネルへの実装は、まだされていない。

2 目的と方針

本研究の目的は、I/O 優先度操作を行う機構を Linux カーネルに実装して、CPU 資源が有効活用されることを実際に確認することである。その方針として次の 3 つのことを立てた。1 つ目は、Completely Fair Queuing (CFQ) スケジューラを拡張することである。これは、Linux カーネルにおいて、I/O 優先度を操作できる唯一の I/O スケジューラである。2 つ目は、各プロセスの I/O 要求のプロファイルを取得し、プロセス毎の I/O 要求頻度を動的に計算することである。3 つ目は、I/O 要求頻度が低いプロセスの I/O 優先度を高くすることである。

3 I/O スケジューラ

Linux カーネルには、I/O 処理を効率よく行うために各プロセスからの I/O 要求を並び替える機構がある。これを I/O スケジューラと呼ぶ。Linux カーネルの既存の I/O スケジューラの 1 つに CFQ スケジューラがあり、本研究で用いる。I/O 優先度には高、中、低の 3 段階があり、デフォルトは中に設定されている。CFQ スケジューラでは、この I/O 優先度をプロセス毎に変更することが可能である。

4 CFQ スケジューラの既存の動作

CFQ スケジューラの既存の動作では、図 1 のように各プロセスからの I/O 要求を並び替える。プロセス 3 は I/O 処理の後に計算処理を行うが、この計算処理に入るためには I/O 処理を終わらせる必要がある。しかし、プロセス 3 の I/O 処理の要求はプロセス 1, 2 の I/O 要求の後に並び替えられてしまっているため I/O 要求が処理されるのが遅くなり、その結果計算処理の CPU 資源への割り

当てが遅くなる。この点が既存のスケジューラにおける課題点となる。

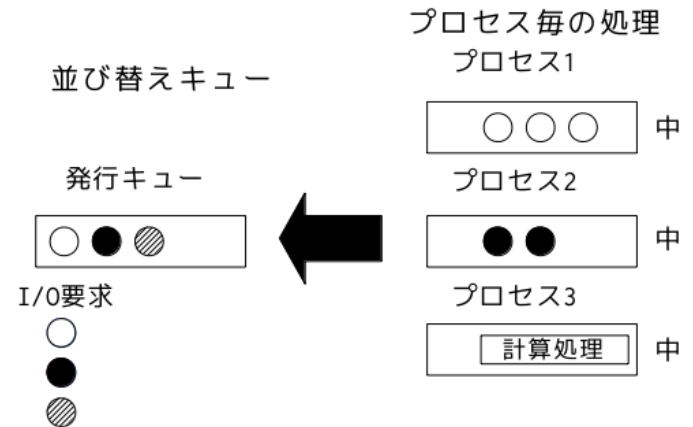


図1 CFQ スケジューラ

5 拡張後のスケジューラ

拡張後のスケジューラでは、まず I/O 操作のプロファイルを取得し、そのプロファイルをもとに各プロセスの I/O 頻度を計算する。この値が小さいプロセスは I/O 要求頻度が小さいとみなす。そして I/O 要求頻度が小さいプロセスの I/O 優先度を高くする。そうすれば、図 2 のように並び替えられる。この結果、プロセス 3 の I/O 要求がプロセス 1, 2 の I/O 要求よりも先に処理されるようになるため、続く計算処理に対して CPU 資源を早く割り当てられるようになる。結果として CPU 資源を有効活用できるようになる。

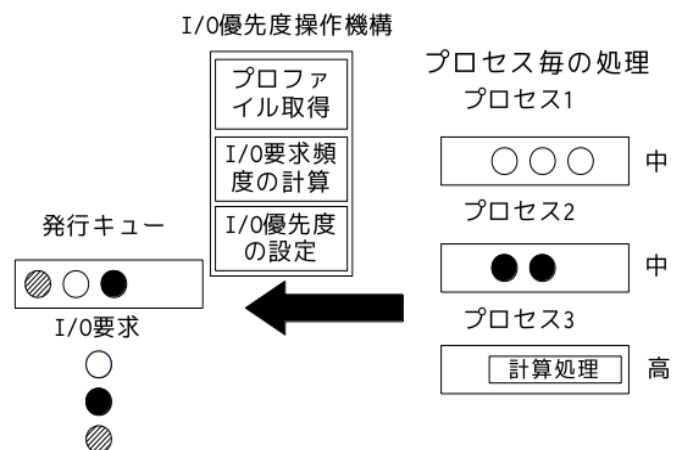


図2 拡張後のスケジューラ

6 プロファイルの取得

本研究で取得するプロファイルとは、プロセス毎の I/O 要求頻度を計算するために必要な情報である。具体的には、プロセス毎の I/O 要求数と時間となる。プロセス毎の I/O 要求数は各プロセスが I/O 要求を出した時に、カーネル内部のデータ構造に記録する。この記録した I/O 要求数と時間で I/O 要求頻度を計算する。その計算方法は様々な方法が考えられるので、今後比較し検討する必要がある。また、計算した I/O 要求頻度をもとにどのような基準で I/O 優先度を決定するかについても、今度比較してその基準を決定する。

7 関連研究

Marcus ら [2] は、ブロック優先の I/O スケジューラの実装を行っている。Seungyup ら [3] は、同期的 I/O 要求を優先する I/O スケジューラの実装を行っている。どちらの研究においても注目するものを優先するような I/O スケジューラを実装しているが、I/O 要求頻度を考慮していないという点で本研究とは異なっている。

8 現状と今後

現在は、I/O 要求数を数える機構の実装を行った。また、先行研究 [1] で行った実験を再現している。今後は以下の 4 つのを行う。1 つ目は、I/O 要求頻度を計算する方法の検討をする。2 つ目は、I/O 優先度を操作する基準を検討する。3 つ目は、I/O 優先度を操作する機構の実装を行う。4 つ目は、拡張した後のスケジューラが有効活用される事を確認する。

参考文献

- [1] 丹治 将貴 (2014), "Linux カーネルにおける I/O 優先度操作による CPU 資源の有効活用", 電気通信大学。
- [2] Marcus Dunn and A.L.Narashimha Reddy(2009), "A New I/O Scheduler for Solid State Devices", Texas A&M University ECE Technical Report TAMU-ECE-2009-02。
- [3] Seungyup Kang and Hyunchan Park and Chunk Yoo(2011), "Performance Enhancement of I/O Scheduler for Solid State Devices", 2011 IEEE International Conference on Consumer Electronics(ICCE), pp.31-32。