

共有資源の利用制限の自動調整による プロセスの動作の効率化

蔭山 直人[†] 芝 公仁[†]

[†] 龍谷大学理工学部

1 はじめに

多くのプロセスは、CPU やディスクなど複数の計算機を使用して動作する。これらの資源はプロセス間で共有されるが、一部のプロセスが特定の資源のみを偏って使い続けると、この資源を共有する他のプロセスの資源待ち時間が増加発生し、スループットが低下してしまう問題がある [1]。

本稿では、偏って資源を使い続けるプロセスに対し利用可能な資源の量の制限を設定し、他のプロセスの資源待ち時間を削減することでシステム全体のスループットを向上させる機構である Resource Manager について述べる。本機構は、資源競合時に資源配分を適切に行うことでシステムの性能を向上させる。

Resource Manager は、Linux カーネルが提供する Control Group [2] の機能を用いることで、プロセスをグループ単位で管理でき、利用可能な資源の量の制限値を一括で設定することができる。また、Control Group では、1つのグループで複数の資源の利用制限を設定できるため、柔軟に資源管理を行える。

2 Resource Manager の構成

Resource Manager の構成を図 1 に示す。Resource Manager は以下の 3つの機構で構成されている。

- CPU 使用状況管理機構：Linux カーネルから取得したプロセスの実行時間や CPU を使用時間を用いて、プロセスごとの平均 CPU 使用率を算出する。
- IO 使用状況管理機構：Linux カーネルが持っているプロセスの統計データを取得し管理する。
- Control Group 操作機構：Control Group の作成、制限の設定などの Control Group に対する操作を行う。

Resource Manager は、起動時に Control Group 操作機構を用いて制限を設定するグループと制限を設定しないグループを作成する。これらのグループにプロセ

Efficient process execution through automatically adjusting usage restricts of shared resources

Naoto Kageyama[†] and Masahito Shiba[†]

[†] Faculty of Science and Technology, Ryukoku University

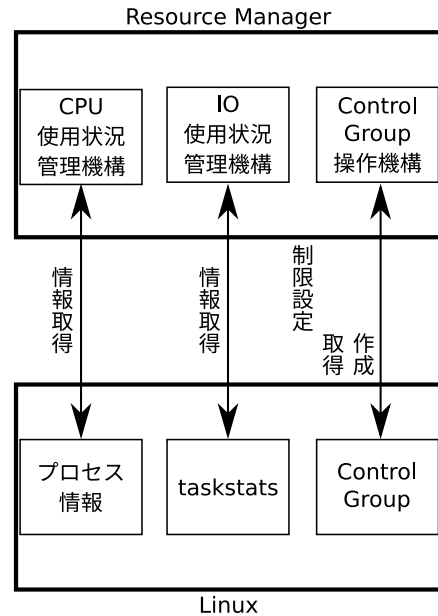


図 1 提案機構の構成

スが登録されると、当該プロセスは Resource Manager の管理対象になる。

Resource Manager は、管理するプロセスが 2 つ以上になると、CPU 使用状況管理機構や IO 使用状況管理機構を用いて、0.5 秒間隔でプロセスごとの資源の使用状況を監視する。このとき、過去 10 秒のデータを蓄積し、平均値を算出する。プロセスのデータを取得し始めて 10 秒経過すると、最も資源を用いているプロセスを制限を設定するグループに登録し、利用可能な資源の量の制限を設定する。このときに、制限が設定されていないプロセスのスループットが向上した場合、資源競合が発生していると判断し、制限を続ける。変化がなかった場合、制限を解除する。これによって、管理プロセスの資源利用状況に応じて、利用可能な資源の量を柔軟に調節できる。

3 資源競合判定と制限の設定

Resource Manager は、Control Group 管理機構を用いて作成したグループに所属しているプロセスを管理対象とし、管理対象プロセスのブロックデバイスの使用状況と CPU 使用状況を Linux カーネルから定期的にする。これらの情報を用いて、プロセスの負荷状況を監視

し、状況に応じてプロセスを適切なグループに所属させる。詳細な動作を以下で述べる。

- (1) プロセスに制限を設定：Resource Manager が最も資源に負荷を掛けているプロセスに対し、制限を設定する。
- (2) 競合判定：Resource Manager が制限を設定したあと、制限が設定されていないプロセスの平均 CPU 使用率もしくは IO 使用回数が向上した場合、資源競合が発生しているとみなす。
- (3) 制限値の調整：制限がかかっていないプロセスが別の処理に移行し、そのプロセスの資源利用状況が変化した場合、資源競合が発生していないとみなし、制限を設定されているプロセスの制限を解除する。

Resource Manager は、(1) で一つの資源を偏って使うプロセスに対し、資源競合時に制限を設定する。制限を設定されるプロセスは、該当プロセスが使用している資源の利用制限を設定しているグループに登録される。

4 評価

本稿では、設定する制限値を変化させることで、それに伴う性能の変化を調べる実験を行った。本実験は表 1 に示す環境で、2 つのプロセスを用いて実験を行った。プロセス 1 はブロックデバイスに対し 512M バイトのデータを書き込み続ける。プロセス 2 はブロックデバイスに対し 256M バイトのデータを書いた後、約 2 秒ループをする処理をする。これを 10 分間行う。同様の実験を本機構を用いなくても行い、行った処理回数を比較する。

設定する制限値は、1MB から 150MB まで 1MB ずつ変化させた。実験結果を図 3 に示す。横軸は設定した制限値で、縦軸は処理回数を示す。本機構を用いた場合、プロセス 1 の処理回数が低下し、プロセス 2 の処理回数が向上したことにより、2 つのプロセスの合計処理回数が本機構を用いない場合と比較して増加した。制限値を 5MB 以上に設定した場合、2 つのプロセスの合計処理回数が多くなり、制限値をそれ以上に設定するとプロセス 2 の処理回数が向上し、Resource Manager を用いない場合の合計処理回数とおおよそ同等になり、処理回数が落ちていることがわかる。この結果から、制限値を低く設定しすぎると、プロセス 1 の処理回数が大幅に低下してしまい、2 つのプロセスの合計処理回数が低下し、制限値を高く設定しすぎると、制限がうまく機能せず、Resource Manager を用いない場合の合計処理回数とおおよそ同等になり、2 つのプロセスの合計処理回数が低下していることがわかる。

表 1 実験環境

OS	Ubuntu18.4.0 LTS
Kernel	Linux 4.19.91
CPU	Intel(R) Core(TM) i5-4570 3.20GHz
コア数	4
HDD	Hitachi HDS72302 2TB

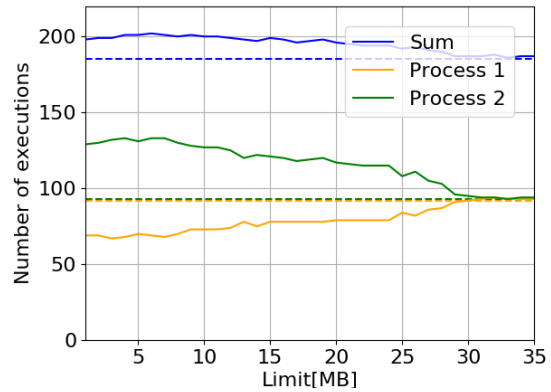


図 2 制限値を変化させたときの処理回数

5 おわりに

本稿では、プロセス間での競合が発生した際の資源配分を自動で調節する Resource Manager について述べた。Resource Manager では、管理プロセス間で競合が発生しているか判断し、競合が発生している資源を最も使用しているプロセスに対して、利用可能な資源の量の制限を設定する。これによってプロセスの待ち時間を減らし、システム全体の性能を向上させることが可能である。

参考文献

- [1] 長尾尚, 田辺雅則, 横山和俊, 谷口秀夫, ”入出力スループットの低下を抑制する入出力性能の調整法の実現と評価”, Vol.2019-OS-145 No.2 February 2019.
- [2] Control Group v2, The Linux Kernel documentation <https://www.kernel.org/doc/html/latest/admin-guide/cgroup-v2.html>