6J-01

KSMの動作の動的制御によるメモリ重複排除の効率化

田上 航希 †1　　芝 公仁 †2
龍谷大学 †1　　龍谷大学 †2

1　はじめに

複数の仮想計算機が同じOSやライブラリを用いる際，メモリページの内容が重複する場合がある．Linuxには，Kernel SamePage Merging(KSM)[1, 2, 3]と呼ばれる機能がある．KSMはユーザプロセスの匿名ページ領域をスキャンし，同じ内容を持つメモリページが存在した場合，それらを一つのメモリページにマージすることで，メモリ使用量の削減を実現する．特に仮想計算機を用いる環境では，同じ内容を持つメモリページが頻繁に生成され，KSMの効果が顕著にみられる．しかし，ページスキャンの速度や頻度を適切に設定しなければ，CPUの負荷が高くなり，KSMが仮想計算機の動作を妨げる場合がある．既存のKSM制御プログラムとして，ksmtuned[4]がある．参考文献[4]では，ksmtunedは，”KSMが重複したメモリページを検索するかどうか，そして，どの程度積極的に検索するかを制御するプログラム”とされている．実際には，仮想計算機のメモリ使用量とシステム全体のメモリ量に関連する閾値を用いて、KSMの開始や停止，ページスキャン速度の調整を行っている．しかし，この制御方法では，KSMのsysfsインタフェースを活用しておらず，KSMの状態を正確に把握できないため，それに応じた柔軟な調整が難しい．また，KSMによるマージを行うべき場合でも，設定された閾値に達していないとマージが行われないため，メモリを節約できる機会を逃してしまうことがある．本研究では，仮想計算機やKSMの状態を基に，KSMのページスキャン速度を動的に調整する機構を開発した．この機構により，KSMスレッド(ksmd)のCPU使用量を削減することができる．さらに，メモリページの効率的な節約により，ホストOSのキャッシュ量を増加させることができるため，システム全体の動作が効率化される．

2　ページスキャン速度の調整

提案機構はLinux上で動作するユーザプロセスである．本機構は図1に示すように，継続的に仮想計算機およびKSMの状態を監視する．そして，現在稼働している仮想計算機の数と，KSMのsysfsインタフェースを取得する．KSMの状態の情報のひとつとしてpages_to_scanがあり，これはKSMが一回のスキャンで処理するページ数である．KSMは一回のスキャンを終えると，一定時間停止する．このような特徴を利用し，本機構は，pages_to_scanの値の設定をすることで，KSMのページスキャン速度を調整する．その調整方法は，先程述べた監視対象から取得した値を基にスコアを算出し，このスコアをページスキャン速度の設定値として直接的に活用する方法である．本機構ではページスキャン速度の最大値を3000，最小値を0としており，ksmdの過度なCPU使用を防いでいる．スコアの計算は，次のサブスコアを合計することで行われる．

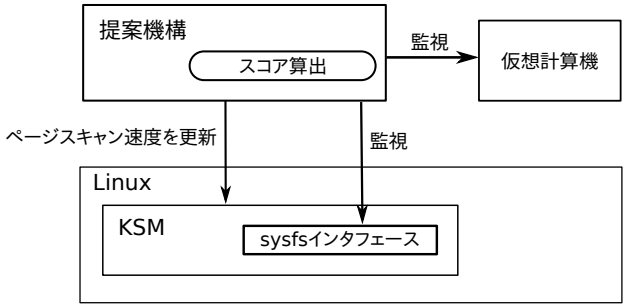

図1　ページスキャン速度の調整

- VMCountScore
- PagesSharingScore
- KSMIdleTimeScore

”VMCountScore”は仮想計算機の増加に応じて変化するサブスコアである．本機構は，新たな仮想計算機が起動されたとき，このサブスコアをKSMが十分動作できる値に設定する．これにより，仮想計算機の数の増加に対応したページスキャンを行うことが可能になる．

”PagesSharingScore”は，節約できたメモリ量によって変動するサブスコアであり，KSMのsysfsインタフェースのpages_sharingとfull_scansを用いて算出される．pages_sharingは，KSMのマージ処理により，物理メモリページを共有する仮想ページの総数を示し，full_scansはKSMがスキャン範囲を全てスキャンし終えた回数を示す．算出方法はfull_scansの数が増加した時と次に増加した時のpages_sharingを記録し，その二値の差分を取る方法である．この算出方法は，一回のfull_scanで，どの程度メモリ使用量を節約できたかを知ることができる．これにより，KSMが効率的にマージが行われているかを判断し，KSMの動作状況に応じた調整を可能にしている．

”KSMIdleTimeScore”は，KSMのページスキャン速度が0になった時点からの時間経過で高くなるサブスコ

Dynamic control of KSM for efficient memory deduplication
†1 Tagami Koki, Ryukoku University
†2 Masahito Shiba, Ryukoku University

表 1　物理マシンの構成

OS	Ubuntu 22.04.3 LTS
Kernel	Linux-6.5.0-1009-oem
CPU	Core i5-11400 2.60GHz
Memory	64GB

表 2　各仮想計算機の構成

OS	Ubuntu 22.04.3 LTS
Kernel	Linux-6.2.0-39-generic
CPU	Core i5-11400 2.60GHz
CPU Core	1
Memory	4GB

アである．本機構は，スコアが 1000 未満の場合，ページスキャン速度を 0 にすることで，効率の悪いスキャンが行われないようにする．また，ページスキャン速度を 0 にされた回数をカウントし，その数に応じて，このサブスコアの増加率を下げることで，無駄な CPU の使用を防いでいる．

3　評価

本機構を用いることで，ksmd の CPU 使用率がどのように変化するかを調べる実験を行った．本実験で用いた環境を，表 1,2 に示す．評価方法として，1 台の物理マシン上に 2 台の仮想計算機を稼働させ，それぞれの仮想計算機上で異なる周期でカーネルビルドの実行と停止を繰り返す方法を採用した．ksmd の CPU 使用率の変化を，図 2 に示す．グラフの横軸は経過時間，縦軸は ksmd の CPU 使用率である．提案機構を使用しない場合は CPU を使用し続けているのに対し，提案機構を使用した場合，定期的に CPU 使用率が低下している．

pages_sharing の変化を図 3 に示す．グラフの横軸は経過時間，縦軸は pages_sharing である．提案機構を使用した場合としなかった場合で，pages_sharing に大きな差はなかった．これは、本機構による動的な制御が KSM の効率を低下させていないことを示している．これらの結果から、本機構が KSM のマージ効率を低下させることなく CPU 使用率の削減を実現していることが確認された．

4　おわりに

本稿では，KSM のページスキャン速度を動的に調整する機構について述べた．本機構は，仮想計算機や KSM の状態を監視しそれに基づいてページスキャン速度を動的に調整する．本機構により，KSM の効率を低下させずに，KSM の CPU 使用量を削減することが可能になる．これによって，仮想計算機の動作を妨げる機会を減少させ，より快適に KSM を利用することができる．

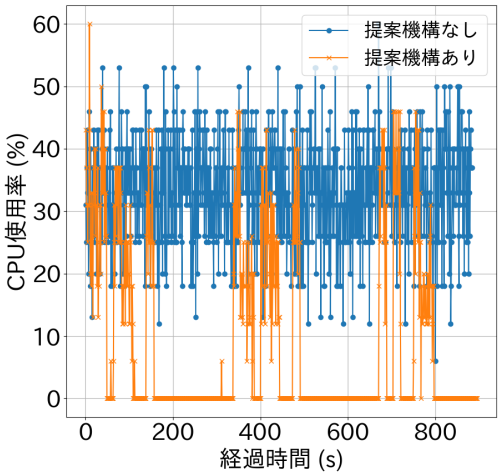

図 2　CPU 使用率

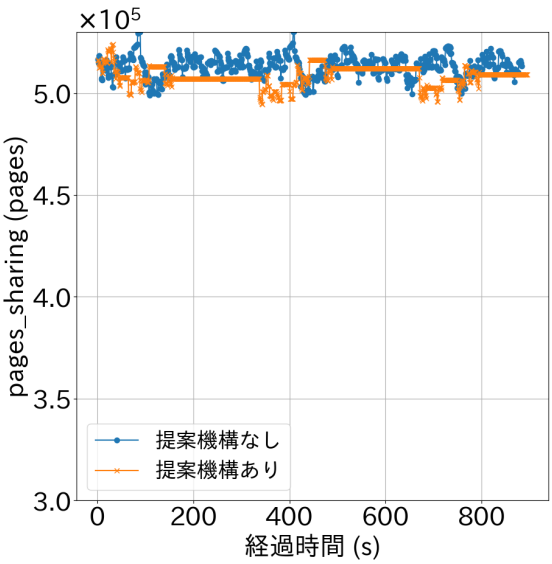

図 3　pages_sharing

参考文献

[1] Izik, E. and Hugh, D.: Kernel Samepage Merging (KSM) (2009).
[2] Arcangeli, A., Eidus, I. and Wright, C.: Increasing memory density by using KSM (2009).
[3] Huang, H., Yan, C. C., Liu, B. and Chen, L.: A survey of memory deduplication approaches for intelligent urban computing, *Machine Vision and Applications*, Vol. 28, pp. 705–714 (2017).
[4] ksmtuned: https://git.centos.org/rpms/qemu-kvm/blob/c8s-stream-rhel/f/SOURCES/ksmtuned,.