1G-06

仮想化環境間のメモリ共有による資源管理の効率化

山田 晃潤† 芝 公仁†
†龍谷大学理工学部

1 はじめに

近年の仮想化技術の普及に伴い，1 台の物理マシン上に複数の仮想マシンを動作させる機会が多くなっている．ホストのマシンは，仮想マシン毎に物理メモリを割り与え，仮想マシンはあたかも自分の物理メモリを持っているかのようにそのメモリを使用することができる．しかし，仮想マシンが同時に複数起動しているとき，ホストマシンのメモリにおいて，その仮想マシンに割り当てられているメモリがホストのメモリを占めてしまう．ホストのパフォーマンスの低下につながり，仮想マシン全体のパフォーマンスの低下にもつながる．このとき，物理メモリ上には仮想マシン台分，同じ OS やミドルウェアが存在することになる．これらの同じ OS やミドルウェアは，複数のプロセスが同じ内容のメモリページを持ち，物理メモリを確保している．同じ内容のメモリページを一つの物理ページに集約することによるメモリ資源管理の効率化を図ることが可能である．このようなメモリページのマージを用いた OS やミドルウェアのメモリ資源管理の効率化を目的とした，KSM(Kernel SamePage Merging)[1] が Linux の機能として備わっている．

KSM は，stable_tree と unstable_tree と呼ばれる二つの赤黒木を用いたページ管理を行う．stable_tree は，マージされているページが管理され，unstable_tree は，マージ候補となるページの管理を行う．

KSM は，ページアドレス順にマージを行う．仮想化環境において，KSM によってマージされるメモリページは，偏りがあると考えられる．例えば，一つのメモリページのマージが行われるとこの後のメモリページも連続してマージが行われやすいなどである．ここで，KSM のページ走査について，メモリアドレスが前のページのマージ状況を取得することで，ある程度マージの発生を予測することができる．提案手法では，近傍のメモリページの情報をもとにした走査による，KSM の拡張を行った拡張 KSM の提案を行う．

2 KSM

マージされているページを管理する stable_tree では，挿入されるページに書き込み保護が行われる．そして，マージが行われているページに書き込み要求が発生した際に，ページフォルトを起こす．そのときにページを複製し，複製されたページに対して書き込みを行う．このように，共有されているページに対しての直接の書き込みは行わないようにして，整合性をとる．

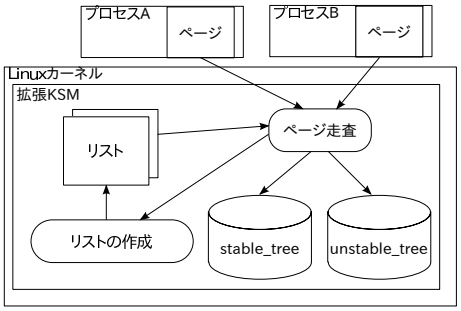

図 1 拡張 KSM の構成

KSM は ksmd というカーネルスレッドを持っており，デーモンとして常駐している．システムコール madvise によって，フラグが MAVD_MERGEABLE のページが作られた時，ksmd はそれらのページの走査とマージ処理とページ管理を行う．ksmd では，ページのフラグが MAVD_MERGEABLE となっているものを rmap_item 構造体で管理している．

3 提案機構

Linux カーネル空間において，KSM を拡張した，拡張 KSM を提案する．その構成を図 1 に示す．拡張 KSM では，既存の KSM と同様に ksmd がカーネルスレッドとし，マージ対象となるページの探索を行う．ページ管理について，既存の KSM と同様に stable_tree，unstable_tree を使用する．ksmd は，マージ可能なフラグのページが発生した時，そのページ情報を取得し，cmp_and_merge_page 関数を呼び出しページ内の比較を行う．cmp_and_merge_page 関数内では memcmp 関数を用いた比較を行い，比較対象の 2 つのページの中を全て調べる．ページ内容が全て同じであればページのマージを行い，マージが行われたページは stable_tree で，マージ不可能なページはマージが行われる候補として unstable_tree で管理される．拡張 KSM は，ページ走査毎にメモリページのマージ情報から N-gram を用いた表の作成を行う．その表をもとにしたマージされる確率の低いページのリストを持つブラックリストの作成を行う．このリストを参考にすることによって，ページ探索の効率化を図る．

An efficient resource management by sharing memory pages among virtual machines
Kojun Yamada†, Masahito Shiba†
†Faculty of Science and Technology, Ryukoku University

表 1　メモリのマージ状況の例

マージ状況	0	0	0	0	0	0	1	1	1	1	0	0	1	0	1

表 2　N-gram を用いた表

マージの並び	000	001	010	011	100	101	110	111
回数	4	2	1	1	1	1	1	2

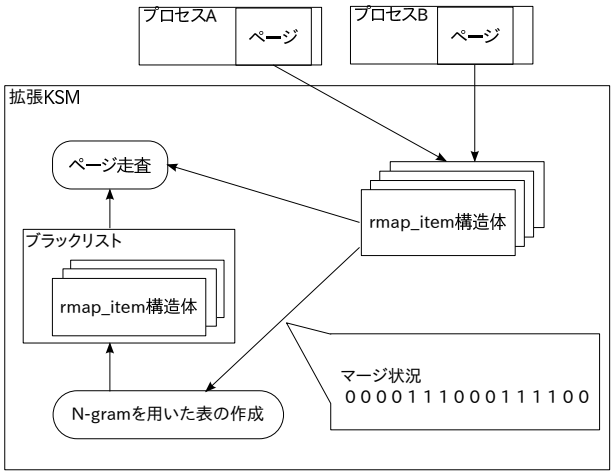

図 2　拡張 KSM の動作

4　拡張 KSM の動作

拡張 KSM の動作は，まず KSM のマージ情報の取得を行う．既存の KSM では，マージ対象となったページは stable_tree か unstable_tree によって管理される．この二つの赤黒木からページを取得するとき，rmap_item というページ情報をもつ構造体によってページの取得や破棄を行う．拡張 KSM でも，この rmap_item 構造体を使用する．

次に，得られたページのマージ情報について，連続したメモリアドレス毎に N-gram を用いた表の作成を行う．本稿においては，メモリページのマージを，マージがされた場合を「1」，されなかった場合を「0」として表す．例としてマージ状況が表 1 であったときの，3-gram を用いて作成したものが表 2 のようになる．この N-gram を用いた表は，KSM のマージ対象のページ走査が行われる毎に更新を行う．

最後に，N-gram の表を用いてリストの作成を行う．N-gram を用いた表から，マージの発生頻度の少ないアドレスの rmap_item 構造体をブラックリストに入れる．これらの動作を図 2 に示す．

得られたブラックリストをもとに拡張 KSM によるページの走査を行う，拡張 KSM のフローチャートについて図 3 に示す．ページの走査を行うときに，まずマージ対象のページがブラックリストに含まれていないかを確認する．もし，マージ対象のページがブラックリストに含まれていた場合は，そのページに関して，比較を行わずに次の対象ページの比較を行う．対象のページがブラックリストに含まれていなかった場合，そ

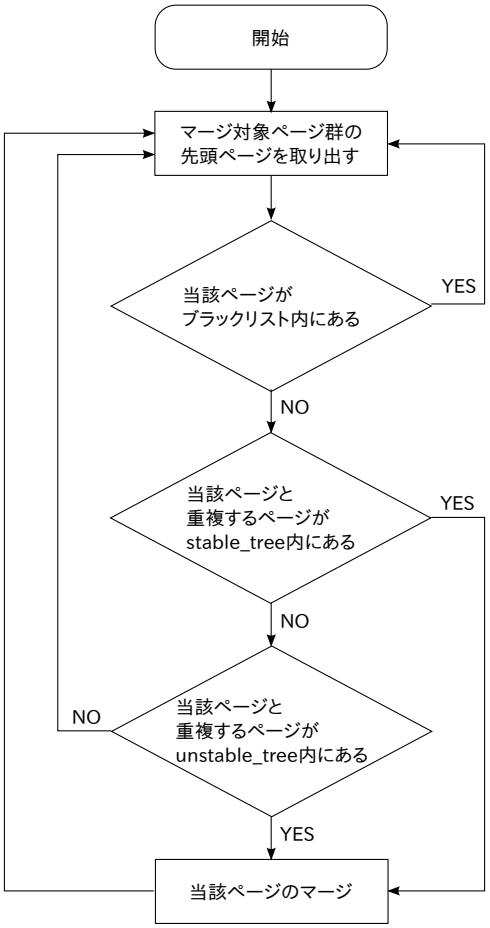

図 3　拡張 KSM のフローチャート

こからの動作は既存の KSM と同様に，対象ページと stable_tree 内のページとの比較を行う．そして，重複したページが存在しなければ unstable_tree 内のページとの比較を行う．このように，ブラックリストを用いることによりマージできないと予測されるメモリページに対する比較を省く．これによって，memcmp を用いた比較や stable_tree と unstable_tree に保持されているページへのアクセスを除外することが可能になり，探索の効率化を図ることができる．

5　おわりに

仮想化環境間では，物理メモリに内容の重複したページが多く存在しており，それらをマージすることによるメモリ資源の効率化を行う KSM が Linux カーネルに備わっている．本稿では，マージ情報から N-gram を用いたリストを作成し，それをもとにページ探索を行う拡張 KSM を提案した．本機構で，KSM のページ探索の効率化が行える．

参考文献

[1] Andrea Arcangeli. Izik Eidus. and Chris Wright.: Increasing memory density by using KSM, Proceedings of the Linux Symposium (2009).