Hugepage を活用するインメモリ分散処理フレームワーク

宮 里 勇 也[†]　山 田 浩 史[†]

分散処理システム[2] とは，複数のコンピュータをネットワークでつなぎ，処理を分散して行うシステムである．1 つのサーバでは捌ききれないような非常に大きなデータを処理することができ，サーバを用意するコストが抑えられることや，リソースの拡張が容易であるというメリットが存在する．分散処理システムの中でも Apache Spark[5] はインメモリ分散処理フレームワークと呼ばれ，処理を出来る限りメモリ内で行うことで，従来のものよりも処理速度などを向上させたシステムである．インメモリ分散処理では膨大なデータをメモリ上で扱う都合上，アドレス変換が大きなボトルネック[1)3)] になってしまう．

アドレス変換のボトルネックを解消する機構としてヒュージページという機構がある．ヒュージページとはアドレス変換を行う単位であるページサイズを通常よりも大きいサイズで使用することで一度のアドレス変換で広い範囲のアドレス変換を行うというものである．ヒュージページを使うことで，ページテーブルへのアクセス回数や TLB ミス率などの削減ができ，アドレス変換にかかる時間を短縮させるメリットがある．Apache Spark にヒュージページを適用する場合，主に THP というヒュージページ管理機構が用いられる．THP では事前にメモリを確保しておかなくともヒュージページを動的に割り当てることができる．しかし，THP にはメモリ肥大化や，メモリのコンパクションのために CPU が占有されるなど様々な問題が存在している．Apache Spark においても THP によるヒュージページ割り当ての効果は大きいが，THP のデメリットも受けてしまうため闇雲にヒュージページを利用することはできない．THP の問題点を解決するために Ingens[4] などが存在しているが，分散処理フレームワーク固有の特性などは考慮に入れておらず，インメモリ分散処理フレームワークを対象としたヒュージページ割り当てに関する研究は存在しない．

本研究ではインメモリ分散処理フレームワーク上でヒュージページを効率的に利用できる手法を提案する．提案方式では，Apache Spark のデータ領域の中でも，ヒュージページの効果が大きい領域にのみヒュージページの割り当てを行い，それ以外では通常のページ割り当てを行うことで無駄なヒュージページ割り当てを避ける．これにより，メモリ肥大化やコンパクションなどのデメリットを抑えつつ，ヒュージページによる性能向上の恩恵を受けることを狙う．

Apache Spark では実行中に使用するメモリ構造として StorageMemory と ExecutionMemory という 2 つのメモリをもっている．StorageMemory は繰り返し利用するデータをキャッシュするための領域であり，ExecutionMemory は処理の実行に必要な中間データなどを一時的に保持するための領域となっている．ヒュージページによる恩恵はデータへのアクセスする回数が多いほど多く受けられるため，繰り返しアクセスされる StorageMemory に対するヒュージページ割り当ての効果が大きい．そこで，提案方式では StorageMemory のみを対象としてヒュージページ割り当てを行う．

予備実験としてヒュージページの割り当てを StorageMemory に対してのみ行った場合と Spark 全体へを行った場合で実行時間などを比較した．キャッシュしたメモリへ繰り返し計算を行うマイクロベンチマークや Kmeans といったワークロードでは，全体割り当てによる実行時間の削減率が約 10%で，一部割り当てでは約 7%となっており、一部への割り当てのみでもヒュージページの効果が十分に見られ，全体に割り当てた場合に近いパフォーマンスを得ることができると分かった．このことから，StorageMemory のみへの割り当てでもヒュージページの効果を十分に受けられると考えた．

提案方式ではヒュージページの割り当てをデータ単位で制御する必要がある．しかし，Apache Spark は JVM 上で動作しており，JVM の実装ではヒュージページを適応する場合，JVM 単位でヒュージページの利用有無を選ぶ必要があり，データごとのヒュージページ割り当てができない．そこで，提案方式ではデータ単位でのヒュージページ割り当てを行うために JVM を改良し，メモリを確保する際に通常のページとヒュージページとどちらで確保するか選べるようにする．改良した JVM 上で Spark を動かし，StorageMemory に保存するデータの場合はヒュージページで割り当てることで効果の大きい効率的なヒュージページを割り当てを実現する．

現在はヒュージページと通常のページ割り当てを使い分けるために JVM の改良を行っている．今後の予

† 東京農工大学
Tokyo University of Agriculture and Technology

定としては，提案方式を完成させた後，実際のベンチマークを動かして，提案方式の効果を確かめる．

参 考 文 献

1) A. Basu, J. Gandhi, J. Chang, M. D. Hill, and M. M. Swift. Efficient virtual memory for big memory servers. In *Proceedings of the 40th Annual International Symposium on Computer Architecture*, ISCA '13, page 237–248, New York, NY, USA, 2013. Association for Computing Machinery.
2) J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, jan 2008.
3) J.Gandhi, A.Basu, M.D. Hill, and M.M. Swift. Efficient memory virtualization: Reducing dimensionality of nested page walks. In *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 178–189, 2014.
4) Y. Kwon, H. Yu, S. Peter, C. J. Rossbach, and E. Witchel. Ingens: Huge page support for the os and hypervisor. *SIGOPS Oper. Syst. Rev.*, 51(1):83–93, sep 2017.
5) M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: Cluster computing with working sets. In *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, page 10, USA, 2010. USENIX Association.