1L-08

コンテナ環境におけるI/O性能向上に関する一考察

水沢直暉[I]　Jian Tao[II]　関優也[III]　山口実靖[I]

[I]工学院大学大学院　[II]Texas A&M University　[III]産業技術総合研究所

1. はじめに

近年サーバーの設置スペースや消費電力の肥大化が大きな問題となっている．この問題の解決策の一つとして仮想化技術を用いたサーバー集約がある．サーバー集約を行う際には集約された環境でも低くない性能を提供する必要がある．仮想化環境にてより高い性能を提供できる仮想化技術としてコンテナ型の仮想化技術があり，コンテナ型の仮想化技術の一つにDockerがある．DockerはゲストOSを持たず，カーネルをホストと共有する．ゲストOSを稼働させる仮想化技術に比べ使用する計算機資源やオーバーヘッドが小さくより高い性能を提供できると期待できる．

本稿では多コンテナ高集約Docker環境におけるファイルI/Oについての測定結果を示し，性能について考察する．

2. 関連研究

ビックデータ処理の場合，I/O性能は非常に重要である．我々は過去に，Dockerの I/O性能はデフォルトの設定(Devicemapper)では十分ではなく[1]，OverlayFSを使用することで改善できると期待されている[2]ということを示した．また，OverlayFSがcopy-upを行う際に毎回vfsfsyncを実行することが過度に悲観的であると考え，vfsfsyncの頻度を減らすことによってI/O性能が向上することを示した[3]．本稿では，Devicemapper使用時，OverlayFS使用時，改善したOverlayFS使用時のDockerのI/O性能について考察を行う．

3. OverlayFS

OverlayFSはunion mount filesystemの一つで，上位ディレクトリと下位ディレクトリをマウントし一つのディレクトリにマージすることができるファイルシステムである．Dockerは近年このファイルシステムを利用している．Linuxでは，カーネルバージョン3.18からカーネルのメインラインに統合されている．

OverlayFSの概要を図1に示す．図の例では，下位ディレクトリであるlowerと上位ディレクトリであるupperがmergedというディレクトリにマージされている．マージされたディレクトリは読み書きすることができるが，下位ディレクトリは読み込み専用で書き込みすることができない．両ディレクトリに「a.txt」が存在している場合，ファイルシステムは上位のディレクトリに存在している「a.txt」をユーザに提供する．すなわち，下位ディレクトリに存在している「a.txt」はユーザからは隠されていて確認できない．「b.txt」が上位ディレクトリにのみ存在している場合，ユーザは上位ディレクトリの「b.txt」にアクセスすることができる．「c.txt」が下位ディレクトリにのみ存在する場合も同様にアクセスすることができる．

ファイルへの書き込みの動作を図2に示す．(A)の場合，ユーザがマージされているディレクトリであるmergedに書き込み処理を実行すると上位ディレクトリであるupperにあるファイルが上書きされる．(B)の場合も同様に上位ディレクトリにあるファイルに上書きされる．(C)の場合，ユーザはファイルを見ることはできるが，ファイルは下位ディレクトリに存在しているため上書きすることができない．したがって，下位ディレクトリに存在しているファイルを上位ディレクトリにコピーし，上位ディレクトリに新しくできたファイルに上書きをする．この下位ディレクトリに存在しているファイルを上位ディレクトリにコピーする動作はcopyupと呼ばれる．(D)の場合は，上位ディレクトリに新しいファイルが作成される．

4. 性能調査

Dockerで OverlayFSを適用した状態でcopyupが発生する(C)の場合においての書き込み処理の実験を行った．

我々の改善[2]をした OverlayFS と無改変(original)の OverlayFS での実験を行った．

「A Study on I/O performance improvement in Containers」
Ⅰ「Mizusawa Naoki, Yamaguchi Saneyasu・Kogakuin University」
Ⅱ「Jian Tao・Texas A&M University」
Ⅲ「Seki Yuya・AIST」

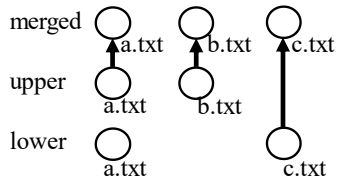

図 1　OverlayFS の概要

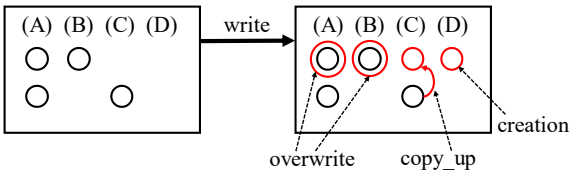

図 2　OverlayFS による書き込み処理

我々の改善した OverlayFS では 50 ファイル毎に vfsfsync を行うように設定している．測定は FFSB(Flexible Filesystem Benchmark)で行い，OverlayFS では下位ディレクトリに 1k-1Gbyte のファイルを 4096 個作成し，それを全てのコンテナが共有している．FFSB はランダムに選択されたファイルへの書き込み処理を繰り返す．コンテナは 1-600 個立ち上げ，並列に FFSB を実行させる．また，ファイルシステムを Devicemapper にした状況での測定も行った．Devicemapper では copyup は発生しない．

測定結果を図 3 に示す．横軸が並列コンテナ数で，縦軸がスループットとなっている．オレンジと青色の線は OverlayFS 使用時の各コンテナで得られた性能の平均値である．オレンジ色の線が改善手法の OverlayFS によって得られた測定結果で，青色の線がオリジナルの OverlayFS で得られた測定結果を示している．また，灰色と黄色の線は OverlayFS 使用時のシステム全体のスループットを示している．スループットでは，改善手法の OverlayFS がオリジナルの OverlayFS の約 5 倍の性能向上を実現しているのがわかる．

また，水色の線はストレージドライバを Devicemapper に変更した，各コンテナでの測定結果を示している．Devicemapper での実験では 600 コンテナで実験した際にカーネルがハングアップしてしまい，300 コンテナまでの実験となっている．Devicemapper では copyup 処理を行わないため copyup が発生する OverlayFS に比べ低集約環境では性能が高いことがわかる．ただし，高集約環境に近づくにつれ性能が下がっていくことがわかる．そして高集約環境では改善した Overlayfs と同等かやや低い性能となっていることが分かる．

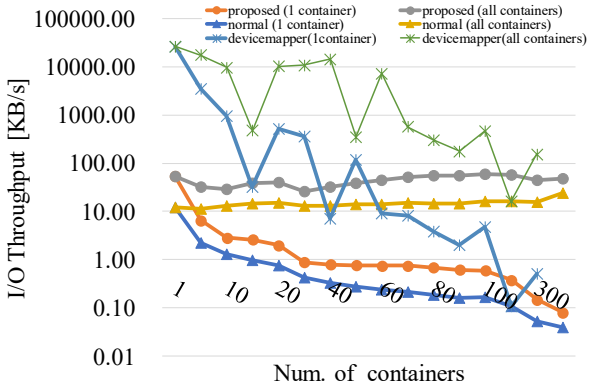

図 3　Docker 上での FFSB 測定結果

5. おわりに

本稿では，Overlayfs 使用時と Devicemapper 使用時の Docker の I/O 性能について考察し，改善手法を用いることにより高集約環境により高い性能を出すことを確認した．

今後はより多くの負荷で I/O 性能の評価を行う予定である．

謝辞

本研究は JSPS 科研費 15H02696, 17K00109, 18K11277 の助成を受けたものである．

本研究は，JST，CREST JPMJCR1503 の支援を受けたものである．

参考文献

[1] J. Kon, N. Mizusawa, A. Umezawa, S. Yamaguchi, J. Tao, "Highly consolidated servers with container-based virtualization", 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, 2017, pp. 2472-2479., doi: 10.1109/BigData.2017.8258205

[2] N. Mizusawa, K. Nakazima and S. Yamaguchi, "Performance Evaluation of File Operations on OverlayFS," 2017 Fifth International Symposium on Computing and Networking (CANDAR), Aomori, 2017, pp. 597-599. doi: 10.1109/CANDAR.2017.62

[3]N. Mizusawa, J. Kon, Y. Seki, J. Tao, S. Yamaguchi, "Performance Improvement of File Operations on OverlayFS for Containers," 2018 IEEE International Conference on Smart Computing (SMARTCOMP), 2018, Vol. 1, pp. 297-302. doi: 10.1109/SMARTCOMP.2018.00019