

リンク構造ファイルシステムにおけるディスク領域管理

06T235 小林 憲弘 (最所研究室)

可変長ブロック単位で操作を行うリンク構造ファイルシステムにおける、ディスクの物理的性質を考慮したディスク領域管理手法およびその評価について述べる。

1 はじめに

既存のファイルシステムではファイルをデータ構造を持たないバイトの集合として扱ってきた。本研究室では、ファイルを可変長ブロックの集合として考え、可変長ブロック単位で操作が行えるファイルシステムであるリンク構造ファイルシステムの開発を行っている [1]。リンク構造ファイルシステムはファイルを可変長ブロックのリンク構造で構成することで、従来とは異なるデータ操作である、ブロック単位での挿入操作や削除操作を実現している。先行研究の実装では、ディスクの物理的な性質を考慮していなかった。そのため、複数ファイルの同時書き込みや、最後にファイルを操作してから時間が経過した後に変更を加える場合、ファイルの実データであるブロックが離れた位置に書き込まれてしまいファイルへのアクセスに時間がかかるといった問題が発生する。

本研究では、ディスクの物理的性質を考慮したディスク領域管理手法を提案し、その評価を行った。

2 リンク構造ファイルシステム

リンク構造ファイルシステムのファイル構造および操作を図1に示す。ファイルは図1に示すように可変長のブロックが連結された形になっている。このような構造にすることにより、ブロックの追加、挿入、削除操作が可能になる。追加操作は末尾に新しいブロックの追加を行い、挿入操作は指定位置への新しいブロックの挿入を行う。削除操作は指定されたブロックの削除を行う。

3 既存のディスク領域管理

磁気ディスクでは、ブロックがディスク領域に分散して配置されたとき、ファイルに対しアクセスを行うとシークや回転待ちをする回数が多くなり、アクセスに時間がかかるといった問題がある。既存のファイルシステムのディスク領域管理では、FFS [2] で用いられているシリンダグループなどのディスクをいくつかのグループに分けてディスク領域を管理し、ファイルのメタデータとデータ領域を近くに置くことによりアクセス効率の向上を図っている。この方法はリンク構造ファ

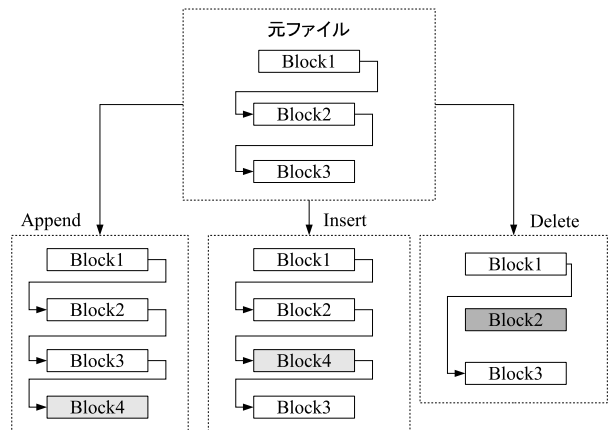


図1: リンク構造ファイルシステムにおける操作

イルシステムでも有効であると考えられるが、グループ内でのブロックの分散を防ぐことはできない。そこで本研究では、リンク構造ファイルシステムにおいて各ファイルのブロックを集中させる手法を提案する。

4 リザーブ領域

ファイルの種類に応じて適切な予約領域を確保し、その領域から、そのファイルのブロックを切り出して割り当てることにより、ブロックの分散を解消できると考え、ファイルに予約領域 (リザーブ領域) を持たせることにした。リザーブ領域は確保したファイルでのみ利用できるデータの書き込まれていない領域である。リザーブ領域を管理するために必要な機能として、リザーブ領域の確保、ブロックへの割り当て、リザーブ領域の追加がある。図2に示すように、リザーブ領域はファイル作成時に Reserve1 のように確保する。ブロックを確保する際は、そのリザーブ領域から確保する。図2の例では、File2 に対して、Reserve2 から Block2-1 と Block2-2 を確保している。これにより、同じファイルのブロックを集中するように配置できる。リザーブ領域が不足した場合は、リザーブ領域を追加で確保することで、できるだけ集中するようにブロックを配置させる。

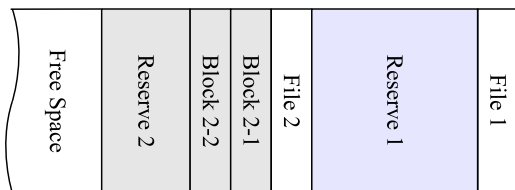


図 2: リザーブ領域の概要

5 評価

リザーブ領域を用いたディスク領域管理の手法が有効であるか確認するための評価実験を行った。ブロック間のオフセットを計測し、ブロックがどの程度分散しているかを評価する。計測では複数のファイルへの追加操作を繰り返し行う。リザーブ領域サイズ 100Byte, 1KByte, 10KByte, 50KByte, 100KByte の場合の、ファイルサイズに対するブロック間オフセットを計測した結果を図 3 に示す。

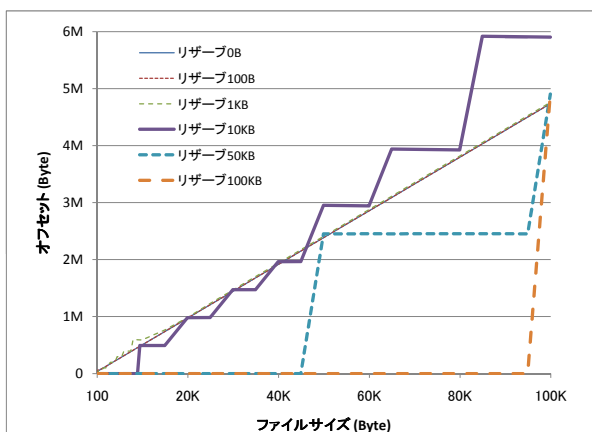


図 3: ファイルサイズに対するオフセット

図 3 に示すように、リザーブ領域がファイルサイズより大きい場合はリザーブ領域を利用しない場合よりもブロック間のオフセットが小さくなりリザーブ領域を用いることでブロックを集中させることができている。しかし、リザーブ領域が 50KByte や 100KByte のとき急激に変化している。これは、リザーブ領域が追加される際に、最初のリザーブ領域よりも離れた位置にリザーブ領域が追加されるためである。リザーブ領域サイズが小さい場合、リザーブ領域を利用しない場合とほとんど差がでなかった。また、リザーブ領域サイズが 10KByte のときはリザーブ領域を用いない場合よりもブロックが分散してしまった。

次に、ファイルサイズが 100Byte から 100KByte の範囲になるような、一様分布に従うランダムサイズの場合の、リザーブ領域サイズに対するブロック間のオフセットを図 4 に示す。

図 4 に示すように、リザーブ領域のサイズが

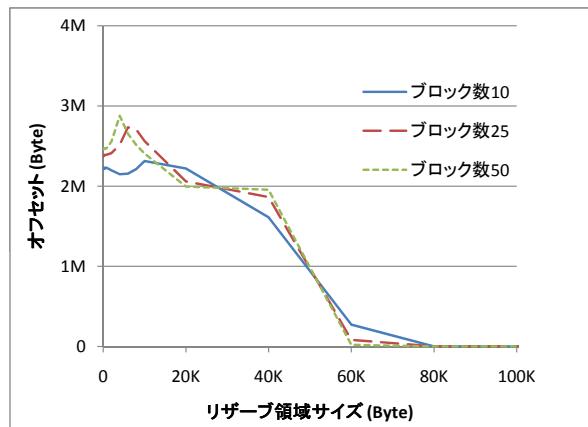


図 4: リザーブ領域に対するオフセット

40KByte を越えたごろからブロック間のオフセットが減少している。また、60KByte から 80KByte でブロック間のオフセットがほとんど無くなり、ブロックがほとんど分散していなかった。

今回のように大きくスケールが異なる範囲のファイルサイズを想定すると、全体的に大きいスケールでリザーブ領域を確保することが有効であるが、小さいサイズのファイルではディスク領域の空間的な利用効率が低下する。

6 おわりに

リンク構造ファイルシステムにおけるディスク領域管理手法としてリザーブ領域を用いたディスク領域管理の手法を提案し、有効性の評価を行った。評価した結果から、リザーブ領域を用いることによりブロックの分散を抑えることが可能であることを確認した。しかし、リザーブ領域サイズが小さい場合、ほとんど効果がないあるいは逆に悪化することがわかった。

今後の主な課題として、ファイルサイズの推定がある。リザーブ領域は現在固定サイズで確保しているが、ディスクの利用効率を上げるにはファイルのサイズにあったリザーブ領域を確保しなければならない。また、リザーブ領域の追加によってブロックが極端に離れて配置されるのを防ぐために、ファイルシステム全体の配置などを検討する必要がある。

参考文献

- [1] 津紀孝, “可変長ブロック単位で操作を行うファイルシステムの開発”, 香川大学大学院工学研究科, 修士論文, 2007.
- [2] Marshall K. McKusick, William N. Joy, Samuel J. Leffler and Robert S. Fabry “A Fast File System for UNIX”, Computer Systems, Vol.2, No.3, pp.181-197, 1984.