B-009

RAID システム内蔵型 NAS (2) ― 高信頼ファイルシステム ―

Embedded NAS for RAID System (2) - High Reliability File System -

中村 隆喜†
Takaki Nakamura

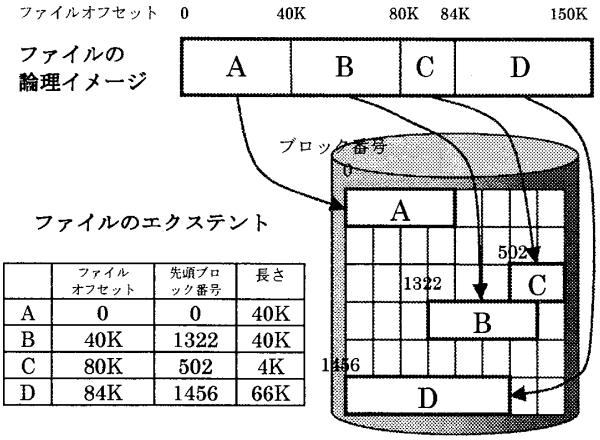
1. はじめに

近年、企業の情報化に伴い、その情報を格納するストレージ装置の市場も拡大している。近年主流となっているストレージ装置には、(1) 複数のサーバを FC スイッチ等によって接続でき、複数のボリュームをサーバに提供可能とする大規模 SAN ストレージ装置と、(2) ネットワークファイルサーバをストレージ装置に統合する NAS がある。NAS に格納されるコンテンツは、画像データ、ストリーミングデータ等、今後も大規模化、大容量化が進むと思われる。

そこで、ハイエンド RAID システム内蔵型 NAS クラス(100TB～)の大容量で、高負荷時においても、高性能かつ安定動作可能であるファイルシステムの高信頼化の検討を行った。特に、高負荷時に発生しやすいファイルフラグメントの課題に取り組み、試作と評価を行った。

2. ファイルフラグメントの課題

図 1にエクステント形式のファイルシステムでファイルがフラグメントした状態を示す。このケースではファイルは A,B,C,D の四箇所に分散配置されており、フラグメント数は 4 である。この場合、ファイルは四つのエクステントにより表現される。一つのエクステントはファイルオフセット、先頭ブロック番号、長さの三つのフィールドからなる。エクステントの長さフィールドの値が小さければ小さいほど、より深刻にフラグメントしていることを意味する。



	ファイルオフセット	先頭ブロック番号	長さ
A	0	0	40K
B	40K	1322	40K
C	80K	502	4K
D	84K	1456	66K

図 1 エクステント形式ファイルシステムでのフラグメント

†(株) 日立製作所 中央研究所
Hitachi Ltd., Central Research Laboratory

ファイルがフラグメントすると、以下の現象が発生することが知られている。

(a) ファイルアクセス性能の低下、特に
  (a-1) I/O スループットの低下
  (a-2) ファイル消去(rm)時間の増大
(b) 過大なリソース要求による動作不安定

(a-1)は全てのファイルシステムで発生する現象である。(a-2),(b)はエクステント形式のファイルシステムで発生しやすい現象であるが、全てのファイルシステムで発生するわけではない。

また、ファイルフラグメントは、一般的に以下の条件で発生する。

(1) ファイルの作成、消去が繰り返し行われ、空き領域がフラグメントしている状態のファイルシステムに新規書き込みを行った場合
(2) 同一のファイルシステムの複数のファイルに対して、同時に新規書き込みを行った場合

前者(1)に対しては、あらかじめ防止することは困難であるため、デフラグツールの適用やファイルシステムの再作成などで対処するのが一般的である。後者(2)は、書き込み領域の予約や遅延書き込みにより、ある程度防止することが可能である。

しかし、従来の予防方式では総容量が 100TB 超の NAS に対しては十分ではなかった。また、小サイズファイルに対する応答性能と容量節約の考慮も十分でなかった。

3. 可変長領域予約方式

前節の課題を解決するため、ファイルサイズに応じて領域予約の方式とサイズを変更する可変長領域予約方式を提案する。本方式ではそれぞれの課題を以下のように解決する。

(1) 大容量 NAS でも安定動作

ファイルサイズが一定値以上になると、システムが安定動作可能なフラグメントサイズ以上の領域予約サイズに切り替える。

(2) 容量の節約

領域予約サイズが大きくなると、使用しない無駄領域が多く発生してしまうため、ファイルサイズに応じて段階的に予約サイズを大きくする。また、大き目の領域予約がファイルシステムフルで失敗したときには、実 I/O サイズで再度領域予約を試みる。ホールファイルに対しても容量節約の観点から、最小限度の大きめの予約サイズもしくは実 I/O サイズで領域予約する。

(3) 小サイズファイルの応答性能考慮

大き目の領域予約を行うと、I/O 完了後、未使用領域の開放処理を行う必要が生じる。小サイズファイルではファイルアクセス応答性能が重要となるため、この開放処理コストは性能低下要因となる。よってファイルサイズが

一定値以下のものは、実 I/O サイズの領域予約とすることで、開放処理を不要とする。

4. 提案方式の試作

可変長領域予約方式を Linux 2.4 と XFS 1.2 をベースに試作した。本予約方式とオリジナル XFS の方式の比較を表 1に示す。

ファイルサイズ 0B～64KB は、領域予約を実 I/O サイズで行い、I/O 完了後の領域開放は行わない。これは小サイズファイルの応答性向上を考慮している。

ファイルサイズ 64KB～8MB 間ではオリジナル XFS と同等の動作となる。そして、ファイルサイズ 4GB になるまで段階的に領域予約サイズを増加する。これは容量節約の観点からこのような処理としている。

ファイルサイズが 4GB 以上になると、システムが安定動作可能となるフラグメントサイズより大きい、256MB の領域予約サイズを適用する。この値は適用するハードウェア、OS、ファイルシステム、サポートするディスク総容量によって異なる。IA32,Linux,XFS の組み合わせでは、仮想連続メモリ領域の大きさと、サポートするディスク総容量と、ファイルエクステントの 1 エントリ分をメモリ上にキャッシュしておくのに必要なサイズ(16Byte)とによって決定できる。

また、ファイルサイズが 4GB 以上では領域開放は行わない。これは XFS 固有の事情による。XFS では close システムコール処理などで未使用領域開放処理を行う。NAS で主にサポートするネットワークファイルシステムである NFS サーバ経由の I/O の場合には、トランザクション単位ごとに close 処理が走るため、予約領域に対して数 KB しか書き込まないうちに未使用領域を開放してしまうことになる。

このため XFS では NFS サーバ経由の書き込みの場合は refcache という専用のキャッシュを設けて close 処理での開放処理を防止している。ただし本機能が有効に働くかどうかは、予約サイズ、refcache のエントリ数、I/O 多重度、kupdate デーモンの周期とその時に開放されるエントリ数に依存する。

そこで、予約サイズ 16MB までは refcache エントリ数を 4096 に増やすことで対策し、それ以上の予約サイズを適用する際は、エントリ数の更なる増加では限界があるため開放処理を行わない処理とした。

表 1 可変長領域予約方式と XFS 方式の比較

ファイルサイズ/条件	領域予約サイズ / 動作 / 条件	
	本方式	オリジナル XFS
0B～64KB	実サイズ 領域開放なし	64KB 領域開放あり
64KB～8MB	64KB 領域開放あり	64KB 領域開放あり
8MB～128MB	1MB 領域開放あり	64KB 領域開放あり
128MB～4GB	16MB 領域開放あり	64KB 領域開放あり
4GB～	256MB 領域開放なし	64KB 領域開放あり
ホール作成時	64KB	64KB
既ホール書き込み時	実サイズ	実サイズ
ファイルシステムフル時	実サイズで リトライ	I/O 失敗
refcache エントリ数	4096	128

5. 提案方式の評価

評価は 8 台の NFS クライアントから 32 個の 1GB ファイルを I/O サイズ 1MB で作成(つまり同時に 256 個作成)し、フラグメント数、Read スループット、ファイル消去時間を比較することにより行った。Read スループットは 256 個のファイルからランダムに抽出した三つのファイルを計測し、平均値を比較した。その際サーバのメモリにキャッシュが載らないように注意した。ファイル消去時間は 256 個を消去した経過時間を比較した。結果を表 2に示す。各項目とも本方式の適用により大幅に改善した。

図 2にはフラグメント数とファイル消去時間の関係を示した。図から明らかなように XFS ではフラグメント数とファイル消去時間はほぼ比例の関係となる。

表 2 可変長領域予約方式の効果

測定項目	改善率
フラグメント数	50.1 倍
Read スループット	1.77 倍
ファイル消去時間	64.3 倍

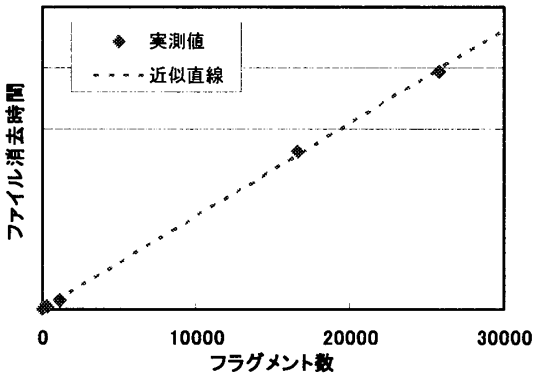


図 2 フラグメント数とファイル消去時間の関係

6. おわりに

ハイエンド RAID システム内蔵型 NAS クラスの大容量で、高負荷時においても、高性能かつ安定動作可能であるファイルシステムの高信頼化の検討を行った。ファイルサイズによってファイル格納領域の予約方式を変える可変長領域予約方式を提案した。本方式により、(1) 大容量 NAS での安定稼動、(2) 容量の節約、(3) 小サイズファイルの応答性能向上が可能となる。提案方式を Linux と XFS をベースに試作した。試作物を評価し、フラグメント数、Read スループット、ファイル消去時間の改善を確認した。