Vol.2014-HPC-145 No.32
2014/7/30

クラスタファイルシステムにおけるハッシュ空間上の位置を考慮したファイル再配置手法

丈達 翔太[1]　芝 公仁[2]　岡田 至弘[2]

概要：ファイルシステムの負荷分散を実現するクラスタファイルシステムには，様々なファイル管理方法がある．その中のひとつにハッシュテーブルを用いてファイル管理を行っているものがある．このようなクラスタファイルシステムでは，ノード数の増減に伴い，各ノードが担当するハッシュ空間の割り当て変更が行われる．その変更の結果に応じて格納ファイルの再配置が行われる．このファイル再配置時のファイルの移動は，システムに大きな負荷を与える．また，再配置後の各ノードの格納ファイルの偏りから，ノード間で負荷の偏りが生じる．本稿では，この問題を解決するハッシュ空間上のファイルの位置を考慮したファイル再配置手法について述べる．本手法により，低コストで負荷分散を実現するファイル再配置が可能となる．

1. はじめに

近年，情報技術の発達によるインターネットサービスなどの多様化により，利用されるファイル数およびファイルサイズが増加してきている．これに伴い，ファイルシステムへの負荷が増大している．そのため，ファイルシステムの負荷分散を実現する技術としてクラスタファイルシステムが注目されている．

クラスタファイルシステムには，様々なファイル管理方法があり，その中のひとつにハッシュテーブルを用いてファイル管理を行っているものがある．このようなクラスタファイルシステムに GlusterFS[9] がある．GlusterFS では，ノードの追加や削除による記憶領域の増減時に，各ノードが担当するハッシュ空間が変更される．その結果に応じて格納ファイルの再配置が行われる．このファイル再配置時の移動ファイル数が多いほど，システムに大きな負荷を与える．

また，ファイル再配置後の各ノードの格納ファイル数に偏りがあると，ノードへのアクセスの偏りが起こり，一部のノードに負荷が集中する可能性がある．さらに，各ノードの格納ファイル容量に偏りがあると，ノード間でのストレージ利用率に偏りが起こり，クラスタファイルシステムでの利用効率が低下する．

本稿では，クラスタファイルシステムにおけるハッシュ空間上の位置を考慮したファイル再配置手法および，その評価について述べる．本手法は，以下の特徴を持つ．

- ハッシュ空間上のファイル位置を考慮した再配置を行う
- 各ノードにファイルを均等に格納することにより，アクセス負荷を分散する
- 格納ファイル容量を均等にすることにより，ストレージ利用率を均等にする

ハッシュ空間上のファイル位置を考慮したハッシュ空間の分割を行うことにより，各ノードの格納ファイル数を均等にすることができる．これにより，一部のノードへアクセスが集中することを防ぐ．また，ファイル再配置後の各ノードの格納容量が均等になるようにハッシュ空間を分割することで，各計算機のストレージを効率的に利用する．

以下，本稿では，2 章で関連研究，3 章で GlusterFS におけるファイル管理方法，4 章でハッシュ空間上の位置を考慮したファイル再配置手法について述べ，5 章で各ファイル再配置手法の評価を行う．

2. 関連研究

インターネットサービスなどの多様化により，従来のように定型データのみならず，文書データや画像，音声データなどの非定型データも取り扱われるようになり，利用されるデータ量が急激に増加している．このような大量のデータを 1 台のファイルシステムで管理するには，負荷が大きい．そこで，ネットワークで接続された複数の計算機で大量のファイルを分散管理を行うことにより，ファイル

[1] 龍谷大学大学院
Ryukoku University Graduate School
[2] 龍谷大学
Ryukoku University

システムへの負荷分散を実現するシステム [1][2] が提案されてきた．その中のひとつにクラスタファイルシステム [3] がある．

2.1 クラスタファイルシステム

クラスタファイルシステムでは，大量かつ大容量のファイルをネットワークで接続された複数のノードのストレージ資源を集約し，そのノード間で管理を行うことにより，各計算機のファイルシステムへの負荷の軽減を実現している．ファイルにはネットワーク経由でアクセスすることになるが，ユーザからは位置透過的にアクセスすることが可能である．また，ノードを追加することで，容易に容量の拡張が行える．

このクラスタファイルシステムには，様々なファイル管理方式がある．既存の多くのクラスタファイルシステムは，ファイルのメタデータの管理を行うメタデータサーバと，ファイルの格納を行うデータサーバから構成されている．このようなクラスタファイルシステムには，ストレージのフェイルオーバによる高可用性を実現する Lustre[4] やディスク I/O を積極的に利用することで高性能を実現する Gfarm[5]，ストレージの高いスケーラビリティと大容量のファイルデータの入出力の高速性を実現する Ceph[6] などがある．これらのクラスタファイルシステムは，メタデータサーバがデータの集中管理を行っているため，メタデータサーバの停止がシステム全体の停止に繋がる単一障害点となっている．そのため，メタデータの冗長化 [7] が必要とされている．また，ネットワークによる遅延やメタデータの管理のためのオーバヘッドによる性能低下が問題となる．

このような問題に対して，クライアント側でのメタデータキャッシュに着目し，そのキャッシュポリシであるサーバベースとクライアントベースをファイル単位で設定し，アクセス状況に応じて動的に変更させる手法 [10]，マウントした Gfarm へのアクセスをメモリコピーとコンテキストスイッチを削除することで，ファイルアクセスの高速化を実現する機構 [11]，大容量のメモリキャッシュの構成を可能とする Cooperative Caching[13] を Gfarm へ実装するための機構の提案 [12] など，多くの研究が行われている．

このような，メタデータサーバでファイルの集中管理を行っているクラスタファイルシステムの他に，メタデータサーバを持たないクラスタファイルシステム [8] がある．その中にひとつにハッシュテーブルを用いてファイル管理を行う GlusterFS[9] がある．

3. GlusterFS におけるファイル管理

GlusterFS では，多くのクラスタファイルシステムにあるメタデータサーバを使用せず，ハッシュ関数を用いてファイル名からハッシュ値を求め，そのハッシュ値によりファイル管理を行う．32 ビットのハッシュ空間をノード数個に均等に分割し，これらを各ノードにハッシュ空間上の担当範囲として割り当てる．そのハッシュの担当範囲に従って，ファイルを格納するノードを決める．ここで，0 から 100 までの範囲を持つハッシュ空間を例として，ファイルが格納されるノードを決める手順を図 1 に示す．まず，ハッシュ関数を用いて，ファイル名「cluster_file_system」のハッシュ値を求める．次に，得られてたハッシュ値 56 を担当範囲に持つノード 3 へ格納する．

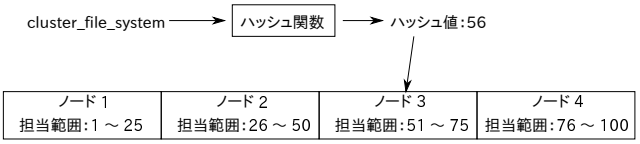

図 1 ファイル格納例

ノードが新しく追加された場合，追加ノードにハッシュの担当範囲を割り当てる必要があるため，ハッシュ空間の再分割が行われ，各ノードにハッシュの担当範囲が新しく割り当てられる．このとき，GlusterFS では，各ノードにハッシュの担当範囲の大きさを均等に割り当てる手法である，ハッシュ均等割り当てを用いて，各ノードにハッシュの担当範囲を割り当てる．

ハッシュの担当範囲が変更されたとき，新しい担当範囲に従って，格納ファイルを再配置させる必要がある．図 1 において，ノード 3 に格納されたファイル「cluster_file_system」は，ノード 2 へと再配置される．

このように，ハッシュテーブルを用いたクラスタファイルシステムである GlusterFS では，ハッシュ空間の分割方法によりファイルの配置が決定され，その決定に応じてファイル再配置が行われる．このファイル再配置では，移動ファイル数に応じて処理の負荷が増減する．つまり，ファイル再配置における移動ファイル数が多いほど，システムに大きな負荷を与えることになる．そのため，ファイル再配置時の移動ファイル数は少ない方が良い．しかし，GlusterFS で使用されている手法では，ノードが追加されるたびに，すべてのノードのハッシュの担当範囲が変更されるため，すべてのノードにおいてファイル再配置が起こり，移動ファイル数が増大する可能性がある．

ファイル再配置後に各ノードへ格納されるファイルの位置を考慮しなければ，各ノードでの格納ファイル数の偏りからアクセスの偏りが生じ，一部のノードにアクセス負荷が集中する可能性がある．また，各計算機のストレージを効率的に利用するためには，各計算機が格納するファイル容量の偏りを小さくすることが望ましい．さらに，各ノードのハッシュの担当範囲に偏りがあると，格納可能なファイルに差が起こり，格納ファイル数や格納ファイル容量の偏りが生じる可能性が大きくなる．よって，各ノードのハッシュの担当範囲の大きさを均等に割り当てる必要が

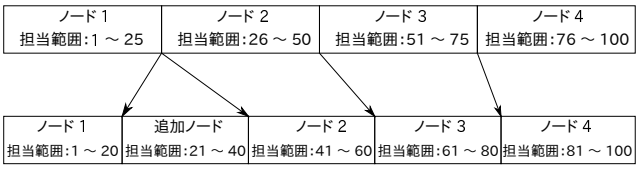

図 2 ハッシュ空間の分割例

ある．

このような，アクセス負荷の分散や容量分散を行う研究として，ファイルシステム側でファイルへのアクセスをコントロールし，アクセスの集中を回避する手法 [14]，ファイルへのアクセス頻度を考慮したデータ配置を行うことで，負荷分散および容量分散を行う手法 [15] などが提案されている．

本稿では，クラスタファイルシステムのファイル再配置において，ハッシュ空間上のファイル位置を考慮したハッシュ空間の分割を行うことにより，ファイル再配置時の移動ファイル数の低減およびファイル再配置後のノード間での負荷の偏りの低減を実現する．

4. ハッシュ空間上の位置を考慮したファイル再配置手法

クラスタファイルシステムにおける負荷とは，ノードへのアクセス数や格納ファイル容量である．これらをクラスタファイルシステムを構成するノード間で均等にすることにより，負荷分散を実現できる．また，ファイル再配置時の移動ファイル数に応じてシステムへの負荷が増減する．よって，ファイル再配置時の処理の負荷を低減させるためには，移動ファイル数を減少させることが重要となる．

4.1 評価式

本稿では，クラスタファイルシステムのファイル再配置時およびファイル再配置後における負荷の評価を行う．評価を行う項目は，以下の通りである．

- 格納ファイル数の偏り
- 格納ファイル容量の偏り
- ハッシュの担当範囲の大きさの偏り
- 移動ファイル数

移動ファイル数はファイル再配置時の負荷であり，格納ファイル数および格納ファイル容量の偏り，そしてのハッシュの担当範囲の大きさの偏りは，ファイル再配置後の負荷である．これらの項目を評価するための式を以下に定義する．

$$F = \sqrt{\frac{(f_{p1} - f_{pa})^2 + (f_{p2} - f_{pa})^2 + \ldots + (f_{pn} - f_{pa})^2}{n}} \tag{1.1}$$

$$f_{pi} = \frac{f_{Si}}{f_T}\ ,\ f_{pa} = \frac{f_a}{f_T} \tag{1.2}$$

F は，ファイル再配置後の各ノードの格納ファイル数の偏りについて表している．まず，各ノードが格納しているファイル数 f_{Si} と総ファイル数 $f_T = \sum_{i=1}^{n} f_{Si}$ から，各ノードが総ファイル数の何割のファイルを格納しているかを算出する．また，平均格納ファイル数 $f_a = \frac{f_T}{n}$ が総ファイル数の何割であるのかを算出する．F は，すべてのノードの格納ファイル数が同一の場合に最小となり，すべてのファイルが 1 台のノードに格納されている場合に最大となる．

$$D = \sqrt{\frac{(d_{p1} - d_{pa})^2 + (d_{p2} - d_{pa})^2 + \cdots + (d_{pn} - d_{pa})^2}{n}} \tag{2.1}$$

$$d_{pi} = \frac{d_{Si}}{d_T}\ ,\ d_{pa} = \frac{d_a}{d_T} \tag{2.2}$$

D は，ファイル再配置後の各ノードの格納ファイル容量の偏りについて表している．まず，各ノードが格納しているファイルの容量の合計値 d_{S1} と総ファイル容量 $d_T = \sum_{i=1}^{n} d_{Si}$ から，各ノードが総ファイル容量の何割のファイル容量を格納しているのかを算出する．また，平均ファイル容量 $d_a = \frac{d_T}{n}$ が総ファイル容量の何割であるのかを算出する．D は，すべてのノードの格納ファイル容量が同一の場合に最小となり，すべてのファイルが 1 台のノードに格納されている場合に最大となる．

$$H = \sqrt{\frac{(h_{p1} - h_{pa})^2 + (h_{p2} - h_{pa})^2 + \cdots + (h_{pn} - h_{pa})^2}{n}} \tag{3.1}$$

$$h_{pi} = \frac{h_{ci}}{h_T}\ ,\ h_{pa} = \frac{h_a}{h_T} \tag{3.2}$$

H は，ファイル再配置後の各ノードのハッシュの担当範囲の大きさの偏りについて表している．まず，各ノードのハッシュの担当範囲の開始値 h_{starti} と終値 h_{endi} より，$h_{ci} = h_{endi} - h_{starti}$ として，担当範囲の大きさを算出する．次に，各ノードのハッシュの担当範囲の大きさ h_{ci} とハッシュの最大値 $h_T = 2^{32}$ から，各ノードがハッシュ空間上の何割の範囲を担当しているのかを算出する．また，ハッシュの担当範囲の平均の大きさ $h_a = \frac{h_T}{n}$ がハッシュ空間上の何割であるのかを算出する．H は，すべてのノードのハッシュの担当範囲が同一の場合に最小となり，1 台のノードの担当範囲がハッシュ空間の最大値である場合に最大となる．

$$M = \frac{m}{f_T} \tag{4}$$

M は，ファイル再配置時における移動ファイル数 m が，総ファイル数の割合を表している．M は，ファイル移動が行われなかった場合に最小となり，すべてのファイルが別

ノードへ再配置された場合に最大となる．

上記で定義した，式 1.1〜式 4 を用いて算出した F，D，H，M の値を用いて，ファイル再配置における負荷の評価を行う．これらの各項の値が大きいほど，ファイル再配置における負荷が大きいということである．よって，各項の値を小さくすることが重要となる．GlusterFS で使用されているハッシュ均等割り当ては，H の値を最小にする手法である．次節にて，F，D，M の値を最小にするファイル再配置手法について述べる．

4.2 ファイル数均等割り当て

ファイル数均等割り当ては，各ノードで格納ファイル数が均等になるようにハッシュ空間を分割する．この手法により，ノード間の格納ファイル数の偏りを最小にすることができる．ノード追加時のファイル再配置の例を図 3 に，示す．ファイルが各ノードに均等に配置されるため，ファイル再配置後に各ノードへのアクセス数が均等になり，一部のノードへアクセスが集中することを防ぐ．

(1) 平均格納ファイル数を求める
(2) 格納ファイルのハッシュ値を取得する
(3) 取得したハッシュ値から，すべてのノードの格納ファイル数が均等になるようにハッシュ空間を分割する

各ノードでの格納ファイル数を均等にするためには，各格納ファイルのハッシュ空間上での位置を把握する必要がある．そのために，各ノードの格納ファイルのハッシュ値を取得する．取得したハッシュ値を昇順に並べ，ハッシュ値が最も小さいファイルから順番にノードへの格納を行い，そのノードの格納ファイル数が平均格納ファイル数となったとき，最後に格納したファイルが持っているハッシュ値までの範囲をノードに割り当てる．これにより，各ノードにファイル数が均等に格納される．ファイル再配置後に，ノード追加前のハッシュの担当範囲とノード追加後のハッシュの担当範囲を比較すると，ハッシュ空間上にずれが起きている．このずれの大きさに比例して，移動ファイル数が増加する．そのため，追加ノードを総ノードの中央に追加することで，ハッシュ空間のずれを小さくし，移動ファイル数を低減できる．

図 3 のように，ノード 4 台にファイルが 15 個格納されている環境に，ノードを追加した場合，ファイル数均等割り当ては次のように行われる．

(1) ノード 2 とノード 3 の間にノードを追加する
(2) ノード 1 にハッシュ値の小さいファイルから順番に格納する
(3) 平均格納ファイル数分のファイルが格納されたらハッシュ空間を分割する
(4) 新たなハッシュの担当範囲に応じて，ノード 2 からファイルが 1 個，ノード 3 からファイルが 2 個追加ノードへ移動する

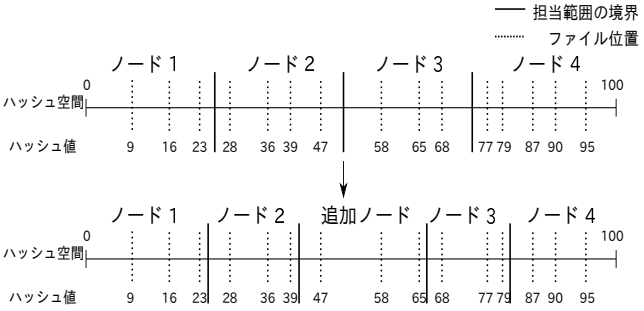

図 3 ファイル数均等割り当て

(5) ノード 4 からノード 3 へファイルが 2 個移動する

以上からすべてのノードにファイルが均等に格納される．また，式 1.1〜式 4 を用いて，各評価値を算出する．ここで，1 ファイルの容量は 1MB，ハッシュ空間の最大値は 100 とする．

総ファイル数 15 個，ノード数 5 台であるため，平均格納ファイル数の割合 $f_{pa} = 0.2$ となる．よって，$F^2 = \{(0.2-0.2)^2 + (0.2-0.2)^2 + (0.2-0.2)^2 + (0.2-0.2)^2 + (0.2-0.2)^2\} / 5 = 0$ となる．

次に D を算出する．総ファイル容量 15MB，ノード数 5 台であるため，平均格納ファイル容量 $d_{pa} = 0.2$ となる．よって，$D^2 = \{(0.2-0.2)^2 + (0.2-0.2)^2 + (0.2-0.2)^2 + (0.2-0.2)^2 + (0.2-0.2)^2\} / 5 = 0$ となる．

次に H を算出する．ハッシュ空間の最大値 100，ノード数 5 台であるため，ハッシュの担当範囲の大きさの平均割合 $h_{pa} = 0.2$ となる．よって，$H^2 = \{(0.23-0.20)^2 + (0.15-0.20)^2 + (0.25-0.20)^2 + (0.13-0.20)^2 + (0.20-0.20)^2\} / 5 = 0.00216$ となる．

最後に M を算出する．総ファイル数 15 個，移動ファイル数 5 個であるため，移動ファイル数の割合は，$M = 5 / 15 = 0.333$ である．

4.3 ファイル容量均等割り当て

ファイル容量均等割り当ては，各ノードで格納ファイル容量が均等になるようにハッシュ空間を分割する．この手法により，ノード間の格納ファイル容量の偏りを最小にすることができる．ノード追加時のファイル再配置の例を図 4 に示す．格納ファイル容量が各ノードで均等なり，各ノードのストレージ利用率が均等化される．

ファイル容量均等割り当ての処理手順を以下に示す．

(1) 平均格納ファイル容量を求める
(2) 格納ファイルのハッシュ値を取得する
(3) 取得したハッシュ値から，すべてのノードの格納ファイル容量が均等になるようにハッシュ空間を分割する

各ノードでの格納ファイル容量を均等にするために，すべてのファイルの容量を取得する．また，すべての格納ファイルのハッシュ空間上での位置を把握する必要があるため，すべての格納ファイルのハッシュ値を取得する．

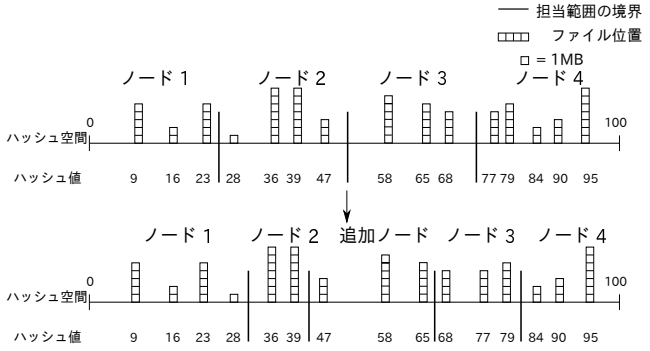

図 4 ファイル容量均等割り当て

次に，取得したファイル容量から平均格納ファイル容量を求める．そして，取得したハッシュ値を昇順に並べ，ハッシュ値が最も小さいファイルから順番にノードへの格納を行い，そのノードの格納ファイル容量が平均格納ファイル容量以上になったとき，最後に格納したファイルが持っているハッシュ値までの範囲をノードに割り当てる．これにより，各ノードの格納ファイル容量の偏りが小さくなる．

図 4 にノード 4 台にファイルが 15 個格納された環境に，新たにノードを追加した場合，ファイル容量均等割り当ては次のように行われる．

(1) ノード 2 とノード 3 の間にノードを追加する
(2) ノード 1 にハッシュ値の小さいファイルから順番に格納する
(3) 平均格納ファイル容量以上のファイルが格納されたらハッシュ空間を分割する
(4) 新たなハッシュの担当範囲に応じて，ノード 1 からノード 2 へファイルが 1 個移動する
(5) ノード 2 から 1 個，ノード 3 から 2 個のファイルが追加ノードへ移動する
(6) ノード 4 からノード 3 へファイルが 2 個移動する

以上から，各ノードで格納されるファイル容量が均等になる．また，式 1.1∼ 式 4 を用いて，各評価値を算出する．ここで，1 ファイルの容量は 1MB，ハッシュ空間の最大値を 100 とする．

総ファイル数 15 個，ノード数 5 台であるため，平均格納ファイル数の割合 $f_{pa} = 0.2$ となる．よって，$F^2 = \{(0.26-0.20)^2 + (0.13-0.20)^2 + (0.2-0.2)^2 + (0.2-0.2)^2 + (0.2-0.2)^2\} / 5 = 0.0017$ となる．

次に D を算出する．総ファイル容量 67MB，ノード数 5 台であるため，平均格納ファイル容量 $d_{pa} = 0.2$ となる．よって，$D^2 = \{(0.21-0.20)^2 + (0.21-0.20)^2 + (0.21-0.20)^2 + (0.19-0.20)^2 + (0.18-0.20)^2\} / 5 = 0.00016$ となる．

次に H を算出する．ハッシュ空間の最大値 100，ノード数 5 台であるため，ハッシュの担当範囲の幅の平均割合 $h_{pa} = 0.20$ となる．よって，$H^2 = \{(0.23-0.20)^2 + (0.15-0.20)^2 + (0.25-0.20)^2 + (0.13-0.20)^2 + (0.20-0.20)^2\} / 5 = 0.00216$ である．

最後に M を算出する．総ファイル数 15 個，移動ファイル数 6 個であるため，移動ファイル数の割合は，$M = 6 / 15 = 0.4$ である．

4.4 部分分割割り当て

部分分割割り当ては，最も格納ファイル数が多いノードから，そのノードが格納しているファイル数の半分を追加ノードが格納するように，ハッシュ空間を分割する．ノード追加時のファイル再配置の例を図 5 に示す．追加ノードに格納されるファイルのみを移動させることにより，移動ファイル数が減少し，ファイル再配置時における負荷を低減することができる．

部分分割割り当ての処理手順を以下に示す．

(1) 格納ファイルのハッシュ値を取得する
(2) 格納ファイル数が最も多いノードを選出する
(3) 選出したノードの格納ファイル数の半分が，移動するようにハッシュ空間を分割する

すべての格納ファイルのハッシュ空間上での位置を把握する必要があるため，すべての格納ファイルのハッシュ値を取得する．取得した格納ファイルのハッシュ値と各ノードのハッシュの担当範囲から，最も格納ファイル数が多いノードを選出する．この選出されたノードは最も格納ファイル数が偏っているため，負荷を低減させるために追加ノードへとファイルを移動させる．このように，ファイル再配置が行われるノードを，選出したノードと追加ノードの 2 つに限定することにより，ファイル再配置時における移動ファイル数が少なくなり，処理負荷を低減することができる．

図 5 のように，ノード 4 台にファイルが 15 個格納されている環境に，新たにノードを追加した場合，部分分割割り当ては，次のように行われる．

(1) 各ノードの格納ファイル数を比較し，最も格納ファイル数の多いノード 4 を選出する
(2) 追加ノードにノード 4 が格納しているファイルの半分が再配置されるように，ハッシュ空間を分割する
(3) ノード 4 から追加ノードへファイルが 3 個移動する

以上から，最も格納ファイル数が多いノードと追加ノードの間でのみファイル移動が行われ，その他のノードではファイル移動が行われていない．よって，移動ファイル数を低減することができる．また，式 1.1∼ 式 4 を用いて，各評価値を算出する．ここで，1 ファイルの容量は 1MB，ハッシュ空間の最大値は 100 とする．

総ファイル数 15 個，ノード数 5 台であるため，平均格納ファイル数の割合 $f_{pa} = 0.2$ となる．よって，$F^2 = \{(0.20-0.20)^2 + (0.26-0.20)^2 + (0.20-0.20)^2 + (0.20-0.20)^2 + (0.13-0.20)^2\} / 5 = 0.0017$ である．

次に D を算出する．総ファイル容量 15MB，ノード数 5

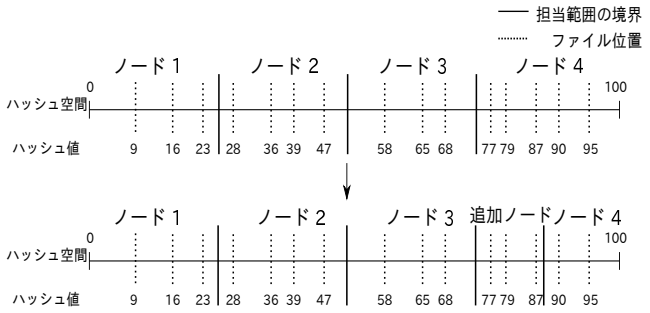

図 5 部分分割割り当て

台であるため，平均格納ファイル容量 $d_{pa} = 0.2$ となる．よって，$D^2 = \{(0.20 - 0.20)^2 + (0.26 - 0.20)^2 + (0.20 - 0.20)^2 + (0.20 - 0.20)^2 + (0.13 - 0.20)^2\} / 5 = 0.0017$ となる．

次に H を算出する．ハッシュ空間の最大値 100，ノード数 5 台であるため，ハッシュの担当範囲の幅の平均割合 $h_{pa} = 0.20$ となる．よって，$H^2 = \{(0.24-0.20)^2+(0.24-0.20)^2+(0.24-0.20)^2+(0.12-0.20)^2+(0.12-0.20)^2\} / 5 = 0.00352$ である．

最後に M を算出する．総ファイル数 15 個，移動ファイル数 5 個であるため，移動ファイル数の割合は，$M = 3 / 5 = 0.2$ である．

4.5 負荷の均等性

ファイル再配置における負荷，およびその負荷を最小にするファイル再配置手法について述べた．一般に，アクセス負荷分散と容量分散を両立させることは困難であるが，これらの負荷を均等にすることは重要である．本稿では，ファイル再配置における各負荷の均等性の評価を行なうために，以下の式を定義する．

$$S = \frac{F + D + H}{3} \tag{5}$$

上記の式を適用し，ファイル再配置において算出した評価値 F，D，H の値の平均値を求める．S は各評価値の平均値である．各評価値が S に近いほど，各負荷の偏りが小さいということである．しかし，平均値 S から，各評価値の差が大きいということは，各負荷の偏りが大きいということである．

5. 評価

本章では，各ファイル再配置手法を適用した時のファイル再配置における負荷についての評価を行う．本実験では，ノードを 1 台追加した時のファイル再配置時の移動ファイル数，ファイル再配置後の各ノードの格納ファイル数および格納ファイル容量，ハッシュの担当範囲の大きさの測定を行った．また，本実験は以下の条件で行った．

- GlusterFS-3.4.0 を使用する
- 単一の計算機で行う
- 無作為に作成した 10000 個のファイルを使用する

本実験では，以下のファイル再配置手法における移動ファイル数，各ノードの格納ファイル容量および格納ファイル容量，ハッシュの担当範囲の大きさの測定を行い，得られた結果を式 1.1 ～ 式 5 に適用し，各評価値を算出した．

A ファイル数均等割り当て
B ファイル容量均等割り当て
C 部分分割割り当て
D ハッシュ均等割り当て

評価値 F，D，H および平均値 S の各手法における変化を，図 6，図 7，図 8，図 9 に示す．

ファイル数均等割り当てでは評価値 F を，ファイル容量均等割り当てでは評価値 D を，そしてハッシュ均等割り当てでは評価値 H を最小にするようにハッシュ空間の分割を行なう．図 6 ～ 図 9 から，各手法が考慮している評価値は最小になるが，その他の評価値が大きくなっていることがわかる．このことから，各評価値はトレードオフの関係にあることがわかる．特に，平均値 S との差が大きい 2 つの評価値は，密接な関係にあると考えられる．

また，評価値 M を最小にする部分分割割り当てでは，追加ノードへのみファイル移動が行なわれるようにハッシュ空間を分割することにより，ファイル再配置時の移動ファイル数を低減する．しかし，追加ノードへのみファイル移動を行うため，追加ノードに隣接しているノードのみ格納ファイル数が減少する．そのため，ファイル移動が行われなかったノードとの間で格納ファイル数や格納ファイル容量の偏りおよびハッシュの担当範囲の大きさの偏りが大きくなってしまう．

各手法における評価値 M の変化を図 10 に示す．各手法において，2 台目のノードを追加したときの移動ファイル数は最も大きい．これは，ハッシュ空間の分割方法に関係なく，各ノードの担当範囲が大きく変更されるためである．ファイル数均等割り当てでは，ノード追加前の各ノードの格納ファイル数と，ノード追加後の平均格納ファイル数の差により，移動ファイル数が増減する．本実験では，総ファイル数が固定数であるため，ノード数の増加に伴い，ノード追加前の格納ファイル数とノード追加後の格納ファイル数の差が小さくなる．よって，移動ファイル数も減少していく．ファイル容量均等割り当てにおいても，ノード追加前の各ノードの格納ファイル容量と平均格納ファイル容量の差により，移動ファイル数が増減する．部分分割割り当てでは，追加ノードへのみファイル移動を行なう．そのため，総ファイル数が固定数である本実験では，ノードの追加に伴い，各ノードの格納ファイル数が減少する．よって，移動ファイル数も減少する．

図 10 から，ファイル位置を考慮した手法は，移動ファイル数が減少し，また，安定していることがわかる．つまり，ハッシュ空間上のファイル位置を考慮することによ

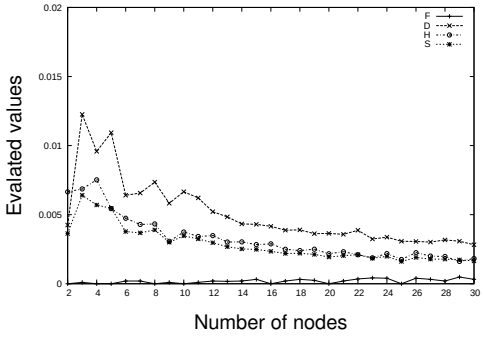

図 6 ファイル数均等割り当てにおける評価値の変化

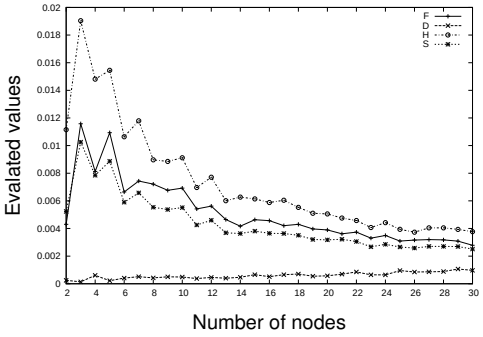

図 7 ファイル容量均等割り当てにおける評価値の変化

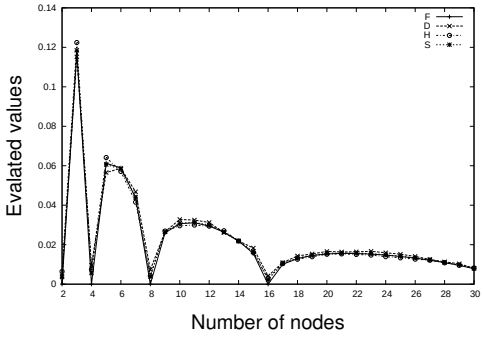

図 8 部分分割割り当てにおける評価値の変化

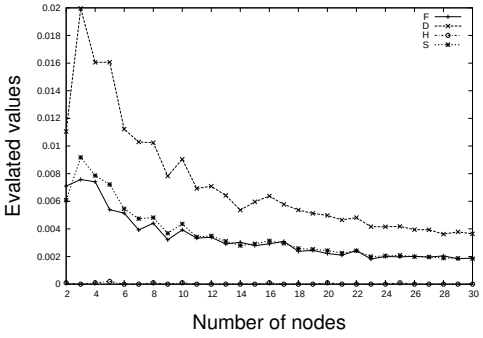

図 9 ハッシュ均等割り当てにおける評価値の変化

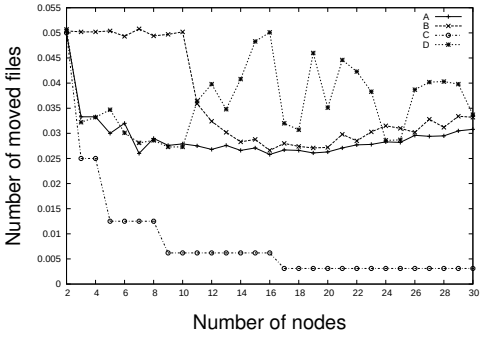

図 10 各手法における M 値の変化

り，低コストなファイル再配置が実現できる．また，ファイル再配置後の格納ファイル数およびファイル格納容量，ハッシュの担当範囲の大きさの偏りの低減が実現できる．しかし，これらの負荷は，トレードオフの関係にあるため，ファイル再配置後の負荷の均等性を実現することは困難である．そのため，想定する環境，利用目的に応じて，適用する手法を変更すると良いと考えられる．

6. おわりに

本稿では，クラスタファイルシステムにおける位置を考慮したファイル再配置手法の提案およびその評価を行った．ファイル数均等割り当てでは，各ノードの格納ファイル数を考慮し，ノード間で格納ファイル数の偏りが最小になるようにハッシュ空間を分割する．ファイル容量均等割り当てでは，各ノードの格納ファイル容量を考慮し，ノード間で格納ファイル容量の偏りが最小になるようにハッシュ空間を分割する．部分分割割り当てでは，追加ノードへのみファイル移動が行われるようにハッシュ空間を分割することで，ファイル再配置時の移動ファイル数の低減を実現する．

参考文献

[1] Lazenby, D. Book Review: Managing AFS: Andrew File System, Linux Journal, (1998)
[2] Levy, Eliezer and Silberschatz, Abraham ”Distributed File Systems: Concepts and Examples”, ACM Computing Surveys, pp. 321–374 (1990)
[3] Harjinder S. Sandhu. and Songnian Zhou. Cluster-based file replication in large-scale distributed systems. SIGMETRICS ’92/PERFORMANCE’92, New York, (1992)
[4] Lustre. http://wiki.lustre.org/index.php/Main_Page/
[5] Tatebe, O., Hiraga, K. and Soda, N. Gfarm Grid File System. New Generation Computing, Vol. 28, No. 3, pp. 257-275(2010)
[6] Home Ceph. http://ceph.com/community/
[7] 平賀 弘平，建部 修見．分散ファイルシステムにおけるメタデータサーバの冗長化手法の検討．情報処理学会研究報告. [ハイパフォーマンスコンピューティング] 2011-HPC-130(37)，1-7，(2011).
[8] 金子 豊，黄 民錫，竹内 真也，和泉 吉則．L-033 構造型 P2P を使った分散ファイルシステムにおける分散ディレクトリ管理手法．情報科学技術フォーラム講演論文集 8(4)，213-216，(2009).
[9] Gluster Community Website. http://www.gluster.org/
[10] 松野 雅也，津邑 公暁，齋藤 彰一，松尾 啓志．キャッシュポリシの動的変更による分散ファイルシステムのメタデータサーバー負荷低減手法．研究報告システムソフトウェアとオペレーティング・システム (OS)，2013-OS-126(3)，1-9，(2013).
[11] 石黒 駿, 村上 じゅん, 大山恵弘．広域分散ファイルシステ

ム Gfarm におけるローカルストレージのファイルアクセスの高速化．情報処理学会研究報告．[ハイパフォーマンスコンピューティング], 2011-HPC-132(17), 1-8, (2011).
[12] 佐々木 慎，松宮 遼，高橋 一志，大山 恵弘．Gfarm ファイルシステムへの Cooperative Caching の実装，情報処理学会研究報告．[ハイパフォーマンスコンピューティング]，2013-HPC-140(15)，1-7，(2013).
[13] Dahlin, M. D., Wang, R. Y.,Anderson, T. E. and Patterson, D. A.: Cooperative Caching: Using Remote Client Memory to Improve File System Performance, Proceedings of the 1st uSENIX Symposium on Operating Systems Design and Implementation, USENIX Association, pp. 267-280(1994)
[14] 佐藤 仁，松岡 聡，中田 秀基．ファイルへのアクセスの自動分散を行うグリッド用分散ファイルシステム．情報処理学会研究報告．[ハイパフォーマンスコンピューティング]，2004-HPC-101，7-12，2005-03-07
[15] 中野 真那，小林 大，渡邊 明嗣，上原 年博，田口 亮，横田 治夫．分散環境でのファイル版管理のためのアクセス頻度を考慮したデータ配置法．DEWS2005 5B-i5.