社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2004－CSEC－27 (15)
2004／12／20

# 分散ストレージシステムおける セグメント欠落を考慮したデータ分割

平野 仁之† 中村 康弘‡

防衛大学校情報工学科 〒239-8686 神奈川県横須賀市走水 1-10-20

E-mail: †g42034@nda.ac.jp, ‡yas@nda.ac.jp

あらまし 比較的小規模な組織内で共有ファイルを分散保管する一方式として，分散ストレージシステムが提案されている[3]．この方式では，共有ファイルを複数のセグメントに分割し，各セグメントにヘッダ情報を付加してネットワーク内の複数端末に分散保管するが，稼動していない端末が存在するなどの理由で，いくつかのセグメントが欠落した場合，共有ファイルを復元することが困難となる．このため，ファイル分割時に冗長性を持った符号化を施すことが課題とされていた．本稿では，使用局面に応じて柔軟に符号化を施すデータ分割の方法を提案するとともに，符号化の違いによって得られる幾つかの性質を明らかにする．本手法を用いれば，個別端末の稼動状況への依存性が少なく，かつ高い復元性を保った分散ストレージシステムを構築できる．

キーワード 分散ストレージ，冗長符号

# Data Separation for Distributed Storage System Considering loss of segments

Hitoshi HIRANO† and Yasuhiro NAKAMURA‡

Dept. of Computer Science, National Defense Academy

1-10-20 Hashirimizu, Yokosuka-shi, Kanagawa,

239-8686 Japan

E-mail: †g42034@nda.ac.jp, ‡yas@nda.ac.jp

**Abstract** Distributed Storage System is proposed as a one of network file sharing system for a small peer to peer network environment [3]. The method divides a target file into small segments, add some header information to each segment and distribute them to the network. However, the some problem remains that system cannot restore the shared file when some segments in cooperative nodes were lost for some reason. So it is subject to make data segments using redundancy code. This paper proposes an additional scheme that enables high performance distribution with user oriented coding control.

This method improve the durability of Distributed Storage System with flexible coding control, and as results, implementation of Distributed Storage System with fault tolerance and publicity control will be achieved independent of other network storages.

**Keyword** Distributed Storage System, Redundancy code

## 1. はじめに

近年，高速常時接続環境の普及により，ネットワークを経由したシームレスな情報交換の自由度が増大している．また，大容量ストレージの低価格化により，音楽や映像などの比較的容量の大きいコンテンツが，エンドユーザ間で共有･交換されるようになってきた．同時に，様々な規模の組織内イントラネットでファイル共有が活用されており，その需要は高まっている．

ネットワークを利用したファイル共有システムについては既に数多くの方法が知られており，サーバ側で利用者の一元管理が必要なものや，元々利用者管理という概念が存在しないシステムなどが，ネットワークにおけるファイル交換の環境を提供している．

ここで，マルチプラットフォームから構成される比較的小規模な組織内におけるネットワークファイル共有システムの導入について考える．オペレーティングシステムの統一が難しい状況下で，ある程度のアクセス制御を考慮しつつファイル共有機能を実現しようとすると，サーバクライアント方式を選択せざるを得ない．しかしながら，サーバクライアント方式によるファイル共有ではネットワーク負荷の集中，利用者管理の複雑化とその労力，サーバ管理などの負担が大きい．これに対し，P2P プロトコルによる方法[1]ではこれらの問題点が発生しない反面，利用者管理という概念がないため，アクセス制御を行うことができない[2]．また，いずれの方式においてもファイルがネットワーク内に偏在するため，ネットワーク全体のストレージの使用効率という面では得策と言えない．

ここで，比較的小規模な組織内ネットワークを仮定し，各端末は十分なストレージを保持しているものとする．一般に個々記憶装置の使用率には偏りがあり，ネットワーク全体では多大な空きリソースが発生している．このようなネットワーク環境下においては，共有ファイルを分割保管することにより，空きリソースを解消し，記憶装置の有効活用を図るシステムが有効と考えられ，その一手法として，ネットワーク上で提供されたすべてのストレージ装置の集合を仮想的に単一のファイルシステムとして扱う分散ストレージシステムが提案されている[3]．

分散ストレージシステムは，共有ファイルを分散保管するものであるが，分散保管した情報の欠落を保管する処理手順が課題とされている[4]．

そこで本発表では，セグメントの作成で使用される符号化方式を選択することでセグメントの冗長性を制御しつつ，かつ高い復元性を得ることのできる柔軟なデータ分割を提案する．

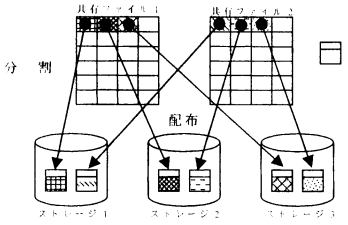

図 1：保存時の概念図

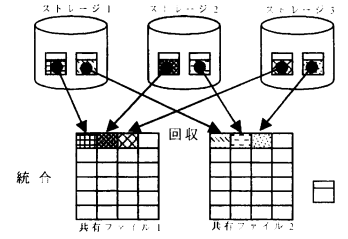

図 2：復元時の概念図

## 2. 分散ストレージシステムの概要

分散ストレージシステムでは，配布する者(配布者)の保存要求に応じ，共有ファイル(配布情報)を $d$ 個のセグメントに配布情報を分割する．分割処理においては，符号化とスペクトラム拡散がなされる．分割された配布情報はセグメントとして扱われ，復元時に必要とされるヘッダが付与される．ヘッダが付与されたセグメントを拡散情報と呼び，配布者はこれらを送信情報として，選択的に $d$ 個の保管先に送信する(図 2)．

拡散情報を受信した保管者は，この情報を個別に管理する．ファイルを復元する必要性が発生した場合は，復元者は自己の保有する拡散情報のヘッダ情報に基づき，保管先で個別に管理されている拡散情報を回収するための復元要求を行う．各保管先においては，適切な拡散情報を復元者に返却する．復元者は拡散情報を受信し，受信情報として統合処理を行う．統合処理は，ヘッダ情報に含まれる拡散シーケンスに従った拡散情報の統合と，符号語の復号を行う(図 2)．

分散ストレージシステムには，2 つの冗長性が存在し，それぞれ次の目的がある[4]．

(i)　ヘッダによる冗長性
- ・拡散情報を復元する基本情報として付加
- ・アクセス制御のために付加

(ii)　符号化による冗長性
- ・セグメントの復元性を得るために付加

### 2.1. ヘッダによる冗長性

分散ストレージシステムは，ファイルの分割によって情報の秘匿性を高めるとともに，アクセス制御の自由度を得るために，分割単位であるセグメントに，冗長なヘッダ情報を付加することに特徴がある．

保存については，ネットワークを経由しないローカル配布と，ネットワークを経由するネットワーク配布も可能とする．ローカル配布の場合は，次のヘッダ情報 $H_i^d$ ($i$ = 1, 2, … , $d$)を付加する．

Ctrl　アクセスフラグ (公開 / 非公開)
Mode　保存フラグ (ローカル / ネットワーク)
Size　配布情報容量
Seg　分割数

ネットワークに配布をする場合，ローカル配布の情報に加えて，次の情報 $H_i^n$ も付加する必要がある．

IP　配布元 IP アドレス
Name　配布情報ファイル名
L　管理先 IP アドレスリスト(拡散シーケンス)

配布情報容量を $Fs$ とし，$d$ 個に拡散されたセグメント容量を $D_i^s$ ($i$ = 1, 2, … , $d$)，ヘッダ容量を $H_i^s$ は $H_i^{Ls} + H_i^{Ns}$ となる．

したがって，拡散情報容量 $S_i^s$ ($i$ = 1, 2, … , $d$)は $H_i^s + D_i^s$ であり，単純に分散した場合であっても，ネットワーク全体に分散される冗長な情報は次のようになる．

$$\sum_{i=1}^{d} H_i^s$$

フラグ Ctrl はアクセス制御のために設定し，配布者・保管先共に復元可能な場合(公開)と，配布者のみに復元可能な場合(非公開)とを考慮する．公開の場合は，すべてのセグメントに $H_l + H_n$ のヘッダを付与するが，非公開の場合は配布者のみ $H_l + H_n$ のヘッダを付し，それ以外の保管先には $H_l$ から生成したハッシュ値 $K$ をヘッダとして付与する．

### 2.2. 符号化による冗長性

分散ストレージシステムでは，分散配布された拡散情報の管理を，保管先に全て委ねる点に特徴がある．ネットワークでのファイルの分散保管は，配布者にとって大変不確実性の高いものとなるが，このために符号化による冗長性を考慮する必要がある[5]．

ここで配布情報 $F$ は，符号アルファベットたる q を元として含む．したがって，容量の基本単位を $U$ とすれば，配布情報点数 $Fa$ は $U \cdot Fs$ と表現できる．セグメントの総点数は，使用する符号化がブロック符号か木符号かによって異なるが，一時線形的に決定する．

### 2.3. 分散の条件

分散ストレージシステムは，ネットワークの発達した比較的小規模な組織内ネットワークを適用範囲とする．既存のネットワーク基盤を利用するため，TCP/IP ネットワーク上に展開する仮想オーバレイネットワークでの利用を前提とする．このため，拡散情報を伝送する際に伝送誤りと不確実伝送は発生しないものと仮定する．

また，空きリソースを解消し，記憶装置の有効活用を図るシステムとするため，過度な冗長性を付加することはできない．

また，符号化された情報源ブロックを符合語アルファベット単位でストライプし，その後に複数の保管先に対して伝送するソフトウェアによるオーバレイネットワーク構成とするため，ある程度の拡散に伴うオーバヘッドが付随し，これら一連の処理を効率化しなくてはならない．したがって，計算量と総伝送量を低く抑える必要があり，符号語長 $n$，情報記号数 $k$ の($n$,$k$)符号を使用する．

また，ネットワーク上に配布する際に，分割数 $d$ をセグメントの総点数より大きくとることは，セグメントサイズ 0 の拡散情報が配布されるノードが発生することになるため，$d$ はセグメントの拡散情報総点数以下の値と仮定する．

### 2.4. 分割の利点と問題点

分散ストレージシステムは，ストライピングによってファイルを分散させるため，ネットワークに所属する端末間の使用率の偏りが解消されることが期待できる．すなわち，各端末の空きリソースを統合し，ネットワーク上のストレージを有効活用することに大きな利点がある．

しかし配布後の拡散情報の管理を保管先に委ねることになるため，データ分割を冗長構成化することにより，セグメントの欠落や改竄に対する抗たん性を備える必要がある．このような問題点については，セグメントを適切に符号化し，ヘッダ $H_l$ に符号化方式を明記した情報を付加することで解決することができる．

## 3. 提案方式

### 3.1. 前提条件

分散ストレージシステムを，比較的小規模な組織内ネットワークで使用する．分散ストレージシステムを使用する場合，共有ファイルの分散に伴う冗長度がネットワーク全体に累積されるが，利用者は分散の条件から著しく逸脱するような利用をすることはなく，むしろ柔軟に抗たん性を持たせることのできるデータ分割が要求されている．

### 3.2. 保存の処理

保存の処理は，次の 2 つのセクションで構成される．

・ コーディングセクション

・ インタリーブセクション

配布情報容量 $Fs$ を先頭から $k$ づつの情報源データにブロック分割し，$M$ 個の情報源データができたとする．配布者は，配布要求に伴い，分割数 $d$ と，それに応じた $d$ の保管先を指定する．前処理においては，システムはこの保管先 IP アドレスリストから，拡散シーケンス $s = \{1, 2, \ldots, d\}$ を決定する．

次にコーディングセクションで，配布情報 $F$ から情報記号数 $k$ づつ情報源データ $f_i$ $(i = 1, 2, \ldots, M)$ を順次ブロック抽出する．その後は $f_i$ を符号語長 $n$ の符合語 $s_i$ に符号化する．この後にインタリーブセクションで，符合語 $s_i$ を拡散シーケンス $s$ にしたがってセグメント $S_i$ $(i = 1, 2, \ldots, d)$ にストライピングする．$f_i$ を全て符号化・ストライピングした後に，後処理として適切なヘッダ情報を付与し，配布者からそれぞれの保管先に拡散データが送信される．

### 3.3. 復元の処理

復元の処理は，次の 2 つのセクションで構成される．

・ インタリーブセクション

・ デコーディングセクション

復元者の復元要求により，ヘッダ情報 $H_n$ に含まれる保管先の IP アドレスリストが読み込まれ，ネットワーク上に分散保管されている拡散情報に回収要求が発せられる．前処理においては，復元者は拡散情報の回収状況に応じ，拡散情報の回収の状況と復元の可否を判断し，IP アドレスリストから，拡散シーケンス $s = \{1, 2, \ldots, d\}$ を確定する．

次にインタリーブセクションで，拡散シーケンス $s$ に従い，回収してきた拡散情報 $S_i$ $(i = 1, 2, \ldots, d)$ を並び替える．これを拡散シーケンス $s$ に従って符号アルファベット単位で順次読み出し，読み出し記号数が $n$ になったら符合語 $s_i$ をなす．

デコーディングセクションで，ヘッダ情報に含まれた符号化方式にしたがって復号処理を行い，情報源データ $f_i$ $(i = 1, 2, \ldots, M)$ とする．この処理を順次繰り返すことにより，後処理で復元情報 $F$ を作成する．

### 3.4. 符号化方式

分散ストレージシステムでは，分割・配布した拡散情報をそのまま回収できれば，符号化に伴う復元性が問題となることはない．したがって次の処理は，符号化方式の仕様に適合する場合であっても，例外的な処理として実行されるべきである．

・ 欠落：回収した拡散情報に不足

・ 改竄：変化した拡散情報を含む

このような例外的な処理を想定して，次の符号化方式を適用可能とする[6]．

・ 単純ストライプ
  - 符号語長を可変
  - セグメント改竄：検出不可能
  - セグメント欠落：補完不可能

・ 単一パリティ検査符号
  - 符号語長を可変
  - セグメント改竄：検出可能
  - セグメント欠落：補完不可能

・ 単一誤り訂正符号(ハミング符号)
  - 符号語長を可変
  - セグメント改竄：検出可能
  - セグメント欠落：補完可能

・ 多重誤り訂正検出符号(BCH 符号)
  - 符号語長を固定
  - セグメント改竄：検出可能
  - セグメント欠落：補完可能

・ 多重誤り訂正符号(多数決符号)
  - 符号語長を可変
  - セグメント改竄：検出可能
  - セグメント欠落：補完可能

## 4. 実験

### 4.1. 実験環境

本手法の効果を確かめるために，符号化による冗長性と復元性について実験を行った．ネットワークに対する保管先は固定 IP とし，通信路における不確実性を排除するため，TCP/IP による確実性のある伝送で実験を行った．

分散ストレージシステムでは，ヘッダとセグメントの両方が冗長構成となっているが，ここではセグメントの冗長率のみを対象とした．また，システムの構成上，符号化処理については符号語長を可変に設定することができるが，ここでは符号語長による差異を明確にするため，特に次の最小多項式を使用した．

・ 単一誤り訂正符号(ハミング符号)

$$G(x) = x^3 + x + 1$$

・ 多重誤り訂正検出符号(BCH 符号)

$$G(x) = x^6 + x^4 + x^2 + x + 1$$

また冗長率として，セグメント冗長率 $R$ を定義する．R は，符号語長 n で単純ストライプ(情報記号数=$n$) する場合のセグメント容量 $V$ と，符号語長 $n$，情報記号数 $k$ で符号化する場合のセグメント容量 $S$ との比であり，次のように定義した．

・ セグメント冗長率 = $\frac{S}{V}$

ここで，分散保管する共有ファイルは，5MB のバイナリファイルを使用する．

### 4.2. 実験結果

#### 4.2.1. 符号化による冗長性

ブロック符号を使用するため，配布情報のデータブロック数と符号語数は，完全に一致する．しかし，配布情報の総点数と拡散情報の総点数は，必ずしも一致しない．これは，配布情報の総点数が，$k$ の整数ではない場合，たかだか 1 つのデータブロックがパディングされるためであるが，セグメント冗長率は，配布情報の総点数と，拡散情報の総点数の比にほぼ一致する．

セグメントの容量 $S$ は，符号化方式によって変化する．また同じ符号化方式であっても，$n$ と $k$ の定義によっても，$S$ は異なる．

符号語長 $n$，情報記号数 $k$ で符号化・拡散し，これによって得られたセグメントの冗長率をエラー! 参照元が見つかりません。に示す．

#### 4.2.2. 符号化による復元性

ブロック符号を使用して配布情報を順次に $(n,k)$ 符号化し，これを $d$ 個のセグメントにストライプすると，次のような性質が得られた．

(i) $d = n$

セグメントの復元性は，使用符号化方式の誤り検出・訂正能力に等しい．

(ii) $d > n$

$d$ が $n$ の倍数のとき，セグメントの復元性は，使用符号化方式の誤り検出訂正能力に等しい．$d$ が $n$ の倍数でないとき，セグメントの復元性は，使用符号化方式の誤り検出訂正能力よりも低下することがある．

(iii) $d < n$

$n$ が $d$ の倍数のとき，セグメントの復元性は，使用符号化方式の誤り検出訂正能力に等しい．$n$ が $d$ の倍数でないとき，セグメントの復元性は，使用符号化方式の誤り検出訂正能力よりも著しく低下する．

$n$ と $k$ を可変に設定した場合，これらの性質に最も符号化の制約を受けるのは，最小多項式の決定に条件を持った単一誤り訂正符号(ハミング符号)と，多重誤り訂正検出符号(BCH 符号)であった．

表 1 符号化によるセグメント冗長率

| 符号化方式 | $n$ | $K$ | $R$ |
|---|---|---|---|
| 単一パリティ検査符号 | 7 | 6 | 1.17 |
| ハミング符号 | 7 | 4 | 1.75 |
| BCH 符号(パリティ付) | 64 | 51 | 1.25 |
| 多数決符号 | 7 | 1 | 7 |

## 5. 考察

共有ファイルを保存するためだけに冗長性を与えることは，既存の方式に比して，非常に意味の無い構成ともなり得る．既存のファイル共有方式によれば，共有ファイルをそのままローカルストレージか共有のファイルサーバに保管するため，共有ファイルのために冗長な情報が付加されることは考えられない．また分散ストレージシステムでは，配布情報を保存する段階でヘッダ $H_d$ が必要となり，更にネットワーク上に分散保管させるためにヘッダ情報 $H_n$ も付加しなくてはならないが，$H_n$ には配布先 IP アドレスリスト $L$ が含まれるため，分割数を過度に増やした場合，$L$ が肥大化し，結果としてヘッダ情報容量 $H$ が線形的に増加してしまう．ここで仮に，小さなファイルを分散保管する場合を考えると，拡散情報に含まれるヘッダのほうが，セグメントよりも著しく大きくなる局面が存在することもあり，著しい冗長性を与えることになる．したがって分散ストレージシステムでは，保存要求を発する配布者は，保存情報の重要性と使用局面に応じて，保存方法を柔軟に選択する必要があるため，符号化方式を選択できる構成としている．

符号化によるセグメントの冗長性は，符号語の冗長率とほぼ一致するため，符号化によるセグメント冗長率をセグメントの冗長性と考えればよい．実験の結果，セグメント冗長率のみに注目した場合，2 重誤り訂正が可能な BCH 符号の効率の良さと，多数決符号の効率の悪さが顕著となった．

また，分散ストレージシステムでは，配布した拡散情報の管理を，保管先に委ねることになるため，セグメントの欠落と改竄に対する抗たん性を備える必要がある．このために，セグメントのデータを冗長構成化が非常に重要な要素となる．

実験では，符号化による復元性を確認するために，無作為にセグメントを削除し，統合作業を実行させた．この結果，インタリーブセクションでの拡散処理の影響により，符号化方式の性能の誤り検出訂正能力よりも低下する局面が存在した．これは，水平インターリーブの端点において，符号語の折り返しを行っており，2 つの水平インターリーブ上に 1 つの符号語が分断されることに原因がある．この制約を改善可能な，水平インターリーブの端点で符号語の折り返しをさせないような工夫については，今後の課題としたい．

また，実験においては，多数決符号の効率の悪さが顕著となった．しかしながら，多数決符号は 50%までのセグメント欠落が許容される強力な復元能力を有している．このため，分散対象とするネットワーク全体の

ストレージへの負担が微小範囲で抑えられるならば，最も有効かつ適切な符号化方式とも考えられる．

また，分散ストレージシステムのモデリングを応用すれば，次の局面での応用も十分に考えられる．

- 共有の重要文書を暗号化する際，暗号化鍵を分散して保管(もしくは配送)
- CAの証明書を分散発行する分散協調型認証

## 6. まとめ

本稿では，保存情報の重要性と使用局面に応じて柔軟に保存方法を選択できる分散ストレージシステムにおけるデータ分割を提案した．また，冗長性と復元性の観点から検討することにより，その有効性と考慮すべき事項，将来的な発展分野を示した．

今後は，システムのP2P化と，導入に伴う運用上の効果を検証し，また将来的な発展分野について具体的な検討をする必要がある．

## 文　献


[1] Gnutella, http://www.gnutelliusms.com/

[2] I. Clarke, et al. : Proc. ICSI Workshop on Design Issues in Anonymity and Unobservability, June 2000.

[3] 平野仁之，岩切宗利，中村康弘，"冗長度を付加する分散ストレージシステムの一実装方式，"電子情報通信学会ソサイエティ大会，p.15，Sep.2003.

[4] 平野仁之，中村康弘，"冗長度とアクセス制御を考慮した分散ストレージシステムの一方式，"情研報，Vol. 2003，No.126，pp.53.

[5] 井上徹 監修：実践 誤り訂正技術，トリケップス

[6] 今井秀樹：符号理論，電子情報通信学会