C-021

適応的消失訂正符号化とデータ圧縮による ディスクアレイの修復バンド幅の削減

Repair Bandwidth Reduction of Disk Arrays Using Adaptive Erasure Coding and Data Compression

長谷川雄大 *　　金子晴彦 *
Yuta Hasegawa　　Haruhiko Kaneko

1　はじめに

ハードディスクドライブは故障率が比較的高いため [1]，大容量ストレージシステムにおいて信頼性向上は重要な課題である．二重故障から回復可能なディスクアレイとして RAID-6 が利用されているが [2]，RAID-6 はディスク故障からの回復の際の読み出しデータ量が多い．読み出しデータ量が多いと回復にかかる時間が長くなり、その間はさらなる故障によるデータ損失の危険にさらされる．本稿では，可逆データ圧縮とディスクの未使用領域を用いて，故障修復の際の読み出しデータ量を低減する手法を提案する．

2　ストレージのモデル

2.1　ディスクアレイの構成

ディスクアレイは N_c 台のディスクで構成されているとする．データサイズが K セクタであるデータブロックを圧縮した後に，消失訂正符号で符号化し，N_c 台のディスクに並列に格納する．データブロックは例えば，Lustre オブジェクトストレージのストライプ [3] に対応し，K は数千セクタであるとする．図 1 はストレージの構成例であり，比較的大きなデータブロックをディスクアレイに格納し，小さいデータは別のストレージに格納する．ディスクアレイ中のデータブロックの位置はロジカルブロックアドレス（LBA）で示す．

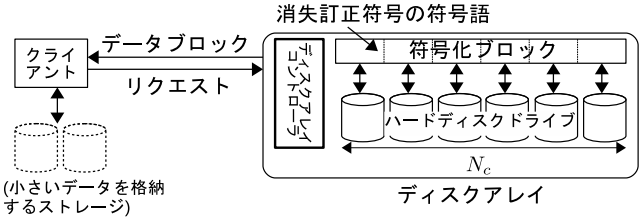

図 1　ストレージの構成例

2.2　高スループット可逆圧縮

サイズ K のデータブロックを圧縮しサイズ K' になるとする．$R_c = K'/K$ は圧縮率である．本稿では具体的な圧縮アルゴリズムは定めないが，キャッシュメモリやデータバス用の低遅延なアルゴリズムを用いる．

2.3　消失訂正符号化と記憶領域への配置

消失訂正符号は符号化率が異なる 3 つのモードを持つ．圧縮したデータの消失訂正符号による符号化は 3 節で示す．符号化したデータブロックの記憶領域への配置方法は 4 節で示す．図 2 に提案手法の概要を示す．

3　適応的消失符号化

消失訂正符号のモードは Mode-H,-M,-L の 3 つのモードから圧縮率 R_c とディスクアレイの未使用領域によって適応的に選ぶ．モード決定のアルゴリズムは 4 節で

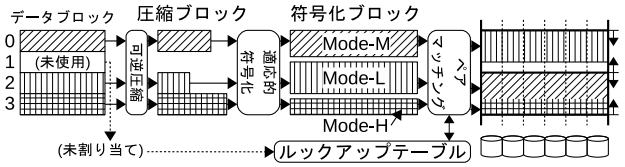

図 2　提案手法の概要

示す．Mode-H は最小距離 3 の Reed-Solomon(RS) 符号を用いる．Mode-M は二重化と，パリティ検査符号を用いる．Mode-L は三重化を用いる．Mode-L は RAID-6 と比較すると符号化率を低くすることで回復コストを低くしている．Mode-H は回復コストと符号化率において RAID-6 と同等である．Mode-M は Mode-L と-H の中間のモードである．

3.1　表記

長さ K のデータブロック $\boldsymbol{D} = (d_0, d_1, \ldots, d_{K-1})$ を圧縮し，長さ K' の圧縮ブロック $\boldsymbol{C} = (c_0, c_1, \ldots, c_{K'-1})$ とする．ただし，$K' \leq K$ であり，ベクトルの各要素はディスクドライブ I/O のユニット（例えば，ディスクセクタ長 512 バイト）に対応する．消失訂正符号の簡単化のために $K' \geq N_c - 2$ とする．$\boldsymbol{C}$ を N 個のサブブロックに分割するとそれぞれ長さ $n = \lceil K'/N \rceil$ となり，$\Delta(\boldsymbol{C}, N) = (C_0, C_1, \ldots, C_{N-1})$ と表記する．ただし，$C_i = (c_{in}, c_{in+1}, \ldots, c_{(i+1)n-1})$ であり，N が K' の約数でないならば，C_{N-1} の最後の $N\lceil K'/N \rceil - K'$ 個の要素は 0 でパディングする．

消失位置 $l \in 0, 1, \ldots, N_c - 1$ に対して，l の左巡回，右巡回をそれぞれ $\lambda_L(l) = (l-1) \bmod N_c$ および $\lambda_R(l) = (l+1) \bmod N_c$ と定義する．

3.2　Mode-H

符号化のプロセスは符号長 N_c，最小距離 3 の RS 符号を用いた RAID-6 と同じである．圧縮ブロック $\boldsymbol{C} = (c_0, c_1, \ldots, c_{K'-1})$ を $N_c - 2$ 個の等長のサブブロック $\Delta(\boldsymbol{C}, N_c - 2) = (C_0, C_1, \ldots, C_{N_c-3})$ に分割する．ただし，C_i の長さは

$$k_h = \left\lceil \frac{K'}{N_c - 2} \right\rceil \tag{1}$$

とする．N_c 個のサブブロックを，GF(2^b) 上の伸長組織 RS 符号で符号化し [4]，符号化サブブロック $(C_0, \ldots, C_{N_c-3}, P, Q)$ を相異なったディスクに格納する．ただし，$N_c \leq 2^b + 1$ であり，$P = (p_0, \ldots, p_{k_h-1}), Q = (q_0, \ldots, q_{k_h-1})$ は RS 符号の検査部である．

故障からの回復は RS 符号を用いた RAID- 6 と同様である [4]．

*東京工業大学 大学院情報理工学研究科

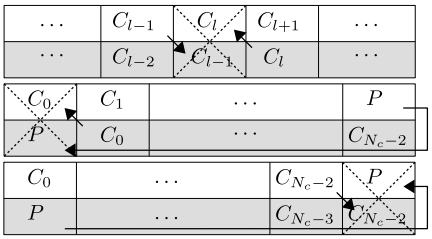

図 3　Mode-M: 単一故障からの回復

3.3　Mode-M

3.3.1　符号化

圧縮ブロックをパリティ検査符号で符号化し，さらにレプリカを作る．圧縮ブロック $\boldsymbol{C} = (c_0, c_1, \ldots, c_{K'-1})$ は $N_c - 1$ 個のサブブロック $\Delta(\boldsymbol{C}, N_c - 1) = (C_0, C_1, \ldots, C_{N_c-2})$, に分割する．ただし，$C_i$ の長さは

$$k_m = \left\lceil \frac{K'}{N_c - 1} \right\rceil \tag{2}$$

である．$(C_0, C_1, \ldots, C_{N_c-2})$ のパリティ検査部 P を

$$P = \bigoplus_{i \in \mathbb{Z}_{N_c-1}} C_i$$

として生成する．ただし，$\bigoplus$ は 2 元ベクトルの排他的論理和による総和を意味する．符号語 $\boldsymbol{U}$ は

$$\begin{aligned}\boldsymbol{U} &= (\quad U_0, \quad U_1, \quad \ldots, \quad U_{N_c-2}, \quad U_{N_c-1} \quad) \\ &= \left(\begin{pmatrix} U_{0,0} \\ U_{1,0} \end{pmatrix}, \begin{pmatrix} U_{0,1} \\ U_{1,1} \end{pmatrix}, \ldots, \begin{pmatrix} U_{0,N_c-2} \\ U_{1,N_c-2} \end{pmatrix}, \begin{pmatrix} U_{0,N_c-1} \\ U_{1,N_c-1} \end{pmatrix} \right) \\ &= \left(\begin{pmatrix} C_0 \\ P \end{pmatrix}, \begin{pmatrix} C_1 \\ C_0 \end{pmatrix}, \ \ldots, \begin{pmatrix} C_{N_c-2} \\ C_{N_c-3} \end{pmatrix}, \begin{pmatrix} P \\ C_{N_c-2} \end{pmatrix} \right)\end{aligned}$$

で生成され，$\boldsymbol{U}$ の N_c 個の要素を相異なったディスクに格納する．

3.3.2　単一故障からの回復

単一故障によって $U_l = (U_{0,l}, U_{1,l})^T$ が消失したとき，

$$U_{0,l} = U_{1,\lambda_R(l)},\ U_{1,l} = U_{0,\lambda_L(l)}$$

により復元できる．図 3 に Mode-M の単一故障からの回復の例を示す．

3.3.3　二重故障からの回復

$U_{l_0} = (U_{0,l_0}, U_{1,l_0})^T$ と $U_{l_1} = (U_{0,l_1}, U_{1,l_1})^T$ が消失したとき，以下のように消失したシンボルを復元する．

(i) $\lambda_R(l_0) = l_1$ であるとき:

$$\begin{aligned}&U_{1,l_0} = U_{0,\lambda_L(l_0)}, \quad U_{0,l_1} = U_{1,\lambda_R(l_1)}, \\ &U_{0,l_0} = U_{1,l_1} = \bigoplus_{i \in \mathbb{Z}_{N_c} \setminus \{l_0, l_1\}} U_{0,i} + U_{1,\lambda_R(l_1)}.\end{aligned}$$

(ii) $\lambda_R(l_1) = l_0$ であるとき:

$$\begin{aligned}&U_{1,l_1} = U_{0,\lambda_L(l_1)}, \quad U_{0,l_0} = U_{1,\lambda_R(l_0)}, \\ &U_{0,l_1} = U_{1,l_0} = \bigoplus_{i \in \mathbb{Z}_{N_c} \setminus \{l_0, l_1\}} U_{0,i} + U_{1,\lambda_R(l_0)}.\end{aligned}$$

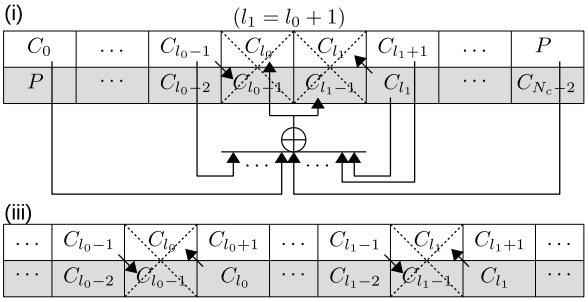

図 4　Mode-M: 二重故障からの回復

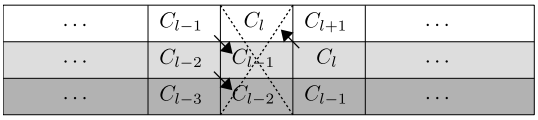

図 5　Mode-L: 単一故障からの回復

(iii) (i),(ii) 以外の場合:

$$\begin{aligned}&U_{0,l_0} = U_{1,\lambda_R(l_0)}, \quad U_{1,l_0} = U_{0,\lambda_L(l_0)}, \\ &U_{0,l_1} = U_{1,\lambda_R(l_1)}, \quad U_{1,l_1} = U_{0,\lambda_L(l_1)}.\end{aligned}$$

図 4 に (i) と (iii) の場合の回復プロセスを示す．

3.4　Mode-L

3.4.1　符号化

圧縮ブロックは三重化によって符号化される．圧縮ブロック $\boldsymbol{C} = (c_0, c_1, \ldots, c_{K'-1})$ は N_c 個のサブブロック $\Delta(\boldsymbol{C}, N_c) = (C_0, C_1, \ldots, C_{N_c-1})$ に分割する．ただし，C_i の長さは

$$k_l = \left\lceil \frac{K'}{N_c} \right\rceil \tag{3}$$

である．符号語 $\boldsymbol{U}$ は

$$\begin{aligned}\boldsymbol{U} &= (\quad U_0, \quad U_1, \quad \ldots, \quad U_{N_c-2}, \quad U_{N_c-1} \quad) \\ &= \left(\begin{pmatrix} U_{0,0} \\ U_{1,0} \\ U_{2,0} \end{pmatrix}, \begin{pmatrix} U_{0,1} \\ U_{1,1} \\ U_{2,1} \end{pmatrix}, \ldots, \begin{pmatrix} U_{0,N_c-2} \\ U_{1,N_c-2} \\ U_{2,N_c-2} \end{pmatrix}, \begin{pmatrix} U_{0,N_c-1} \\ U_{1,N_c-1} \\ U_{2,N_c-1} \end{pmatrix} \right) \\ &= \left(\begin{pmatrix} C_0 \\ C_{N_c-1} \\ C_{N_c-2} \end{pmatrix}, \begin{pmatrix} C_1 \\ C_0 \\ C_{N_c-1} \end{pmatrix}, \ldots, \begin{pmatrix} C_{N_c-2} \\ C_{N_c-3} \\ C_{N_c-4} \end{pmatrix}, \begin{pmatrix} C_{N_c-1} \\ C_{N_c-2} \\ C_{N_c-3} \end{pmatrix} \right)\end{aligned}$$

で生成され，$\boldsymbol{U}$ の N_c 個の要素を相異なったディスクに格納する．

3.4.2　単一故障からの回復

単一故障によって $U_l = (U_{0,l}, U_{1,l}, U_{2,l})^T$ が消失したとき，

$$U_{0,l} = U_{1,\lambda_R(l)},\ U_{1,l} = U_{0,\lambda_L(l)},\ U_{2,l} = U_{1,\lambda_L(l)}$$

により復元する．図 5 に Mode-L の単一故障からの回復を示す．

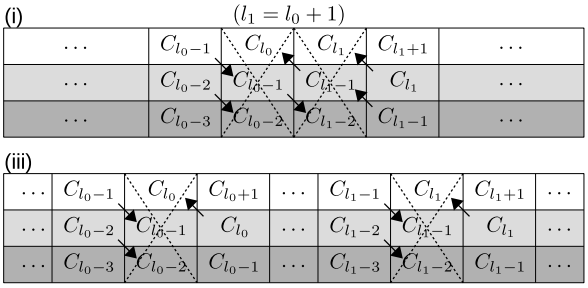

図 6　Mode-L: 二重故障からの回復

3.4.3　二重故障からの回復

$U_{l_0} = (U_{0,l_0}, U_{1,l_0}, U_{2,l_0})^T$ と $U_{l_1} = (U_{0,l_1}, U_{1,l_1}, U_{2,l_1})^T$ が消失したとき，以下のように消失したシンボルを復元する．

(i) $\lambda_R(l_0) = l_1$ の場合:

$$U_{0,l_0} = U_{1,l_1} = U_{2,\lambda_R(l_1)},$$
$$U_{1,l_0} = U_{2,l_1} = U_{0,\lambda_L(l_0)},$$
$$U_{2,l_0} = U_{1,\lambda_L(l_0)}, \quad U_{0,l_1} = U_{1,\lambda_R(l_1)}.$$

(ii) $\lambda_R(l_1) = l_0$ の場合:

$$U_{0,l_1} = U_{1,l_0} = U_{2,\lambda_R(l_0)},$$
$$U_{1,l_1} = U_{2,l_0} = U_{0,\lambda_L(l_1)},$$
$$U_{2,l_1} = U_{1,\lambda_L(l_1)}, \quad U_{0,l_0} = U_{1,\lambda_R(l_0)}.$$

(iii) (i),(ii) 以外の場合:

$$U_{0,l} = U_{1,\lambda_R(l)},\ U_{1,l} = U_{0,\lambda_L(l)},\ U_{2,l} = U_{1,\lambda_L(l)}.$$

図 6 に (i) と (iii) の場合の回復プロセスを示す．

4　データブロックのディスクアレイへの配置

本節では可変長の符号化ブロック $\boldsymbol{U}$ をディスクアレイの記憶領域に配置するアルゴリズムを述べる．

4.1　ペアマッチング [5]

ディスクアレイの記憶領域を N_t 個のグループに分割し，各グループに $2N_g$ 個の符号化ブロックを割り当てる．したがって，ディスクアレイに格納できる符号化ブロックは最大で $2N_gN_t$ 個である．グループのインデックスはディスクアレイの LBA から静的に決まり、グループ内での符号化ブロックの位置はペアマッチングアルゴリズムによって動的に決定される．ペアマッチングアルゴリズムでは各ブロックの符号化モード，圧縮後のブロックサイズ K'，記憶位置を収めたルックアップテーブルが必要となる．

B_i をサイズ $\kappa_i \times N_c$ セクタの符号化ブロックとする．ただし $i \in Z_{2N_g}$ はグループ内のブロックのインデックスである．以下では κ_i の値を”長さ”と称する．$2N_g$ 個の符号化ブロックの集合 $B_0, \ldots, B_{2N_{g-1}}$ をサイズ $2N_gk_r \times N_c$ セクタのグループに割り当てる．ただし，$k_r = K/(N_c - 2)$ は圧縮率 $R_c = 1$ の場合の Mode-H の符号化ブロックである．したがって，$k_h \le k_r$ が常に成り立つ．グループの記憶領域はサイズ $2k_r \times N_c$ の N_g 個のセグメントに分割する．各セグメントには side-A,-B の二つの面があり，最大 1 つの符号化ブロックが各面に割り当てられる．したがって，1 つのセグメントに最大で 2 つの符号化ブロック B_i, B_j が割り当てられる．ただし，$\kappa_i + \kappa_j \le 2k_r$ である．ルックアップテーブルでは未使用ブロックは $K' = 0$ で示され，セグメントには割り当てられない．図 7 に符号化ブロックの集合とグループの記憶領域との関係を示す．

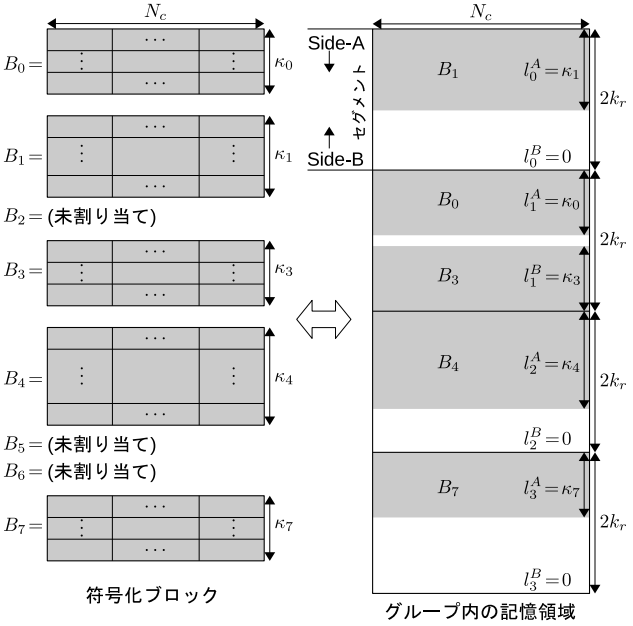

図 7　ペアマッチングアルゴリズムの例 $(N_g = 4)$.

4.2　配置アルゴリズム

あるグループに配置されたブロック長のベクトルを，

$$L = ((l_0^A, l_0^B), (l_1^A, l_1^B), \ldots, (l_{N_g-1}^A, l_{N_g-1}^B))$$

と定義する．ただし，l_i^X はセグメント i の side-X に割り当てられた符号化ブロックの長さ κ_j であり，$X \in \{A, B\}, i \in Z_{N_g}$ である．ある L が与えられたとき，長さ k の新しい符号化ブロックを格納できるセグメントのインデックスの集合を

$$I(L,k) = \{\ i \mid (l_i^A + l_i^B + k \le 2k_r) \wedge (l_i^A = 0 \vee l_i^B = 0) \wedge (i \in \mathbb{Z}_{N_g})\}$$

と定義する．集合 $I(L,k)$ が空集合でないとき，最小の未使用領域を持つセグメントのインデックスを

$$f(L,k) = \arg\max_i \{l_i^A + l_i^B \mid i \in I(L,k)\}$$

で与える．

新しいデータブロック $\boldsymbol{D} = (d_0, d_1, \ldots, d_{K-1})$ をあるグループの記憶領域に割り当てる手順を以下に示す．

1. データブロックを長さ K' の $\boldsymbol{C} = (c_0, c_1, \ldots, c_{K'-1})$ に圧縮する．
2. 式 (1), (2), (3) を用いて圧縮ブロック $\boldsymbol{C}$ の長さ k_h, k_m, k_l を計算する．

3. $I(L, 3k_l)$ が空集合でないなら，$\boldsymbol{C}$ は Mode-L で符号化できる．符号化ブロックの長さ k を $k = 3k_l$ とし，7. に進む．
4. $I(L, 2k_m)$ が空集合でないなら，$\boldsymbol{C}$ は Mode-M で符号化できる．符号化ブロックの長さ k を $k = 2k_m$ とし，7. に進む．
5. $\boldsymbol{C}$ を Mode-L,-M で符号化できなければ，Mode-H で符号化する．符号化ブロックの長さ k を $k = k_h$ とする．$I(L, k_h)$ が空集合でないなら，7. に進む．
6. 新しい符号化ブロックを格納する領域を作るために，セグメント j の side-Y に格納されている符号化ブロックを Mode-H で再符号化する．ただし，

$$j = g(L) \text{ かつ } l_j^Y > 0$$

である．ここで，関数 g はグループ内のセグメントのうち，片面のみを使っていて，使っている面の符号化ブロックの長さが $l > k_r$ を満たす最小の l であるようなセグメントを選ぶ．つまり $g(L)$ は

$$g(L) = \arg\min_i \{l_i^A + l_i^B \mid i \in I(L, 0), l_i^A + l_i^B > k_r\}$$

で定義される．このステップでは長さ k_r 以上の未使用の面を持つセグメントを作り出す．よって，任意の Mode-H の符号化ブロックはそのセグメントに配置することができる．

7. $\boldsymbol{C}$ をセグメント i の side-X に格納する．ただし，

$$i = f(L, k) \text{ かつ } l_i^X = 0$$

である．

5 評価

確率的なワークロード生成器とディスクドライブシミュレータ DiskSim[6] を用いた簡易なシミュレーションにより故障からの回復時のデータ転送量を評価した．

5.1 評価方法

シミュレータはワークロード生成器, ディスクアレイコントローラ，ディスクドライブシミュレータ DiskSim の 3 つの要素から構成される．

ワークロード生成器はディスクアレイのワークロードを確率的に生成する．リクエスト到着はレート $\lambda = 1, 2, 3, 4, 5$(リクエスト/秒) のポアソン到着に従うとする．リクエストの LBA の分布は Zipf 分布に従う．ただし，LBA m のデータブロックは確率 $(1/m^s)/(\sum_{i=1}^{n} 1/i^s)$ でアクセスされ，$n = 2N_gN_t$ はデータブロックの総数とする．全てのデータブロックに対する有効なデータブロックの割合を $P_{\text{util}} = 0.7$ とする．

ディスクアレイコントローラは提案手法と RAID-6 の機能をシミュレートし，ここでワークロードを DiskSim のディスクアクセスリクエストに変換される．提案手法では，圧縮率 $R_c = K'/K$ は平均 0.6，標準偏差 0.2 のガウス分布に従い，圧縮率の範囲は $0.3 \leq R_c \leq 1.0$ とする．

ディスクアレイは $N_c = 6$ 台の Atlas 10k モデルのディスクを持ち [6]，記憶領域は $N_t = 256$ 個のグループを持ち，各グループには $N_g = 16$ 個のセグメントがある．データブロックのサイズは $K = 8192$ セクタで，セグメントのサイズは $2k_r \times N_c = 4096 \times 6$ セクタである．

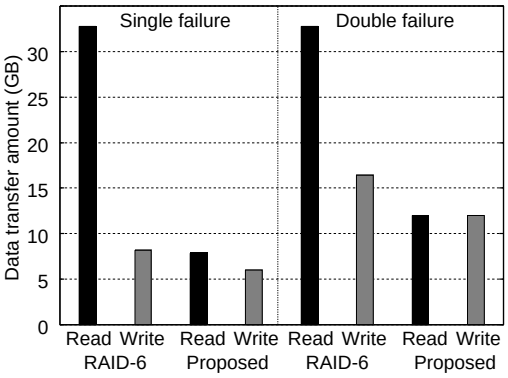

図 8　回復時のデータ転送量

5.2 回復時のデータ転送量

図 8 は単一故障，二重故障から回復するために必要な読み出し・書き込み量を示している．RAID-6 と比較すると，提案手法では単一故障からの回復時に必要な読み出しデータ量は 0.24 倍になっており，二重故障からの回復では 0.37 倍になっている．

6 まとめ

本稿ではディスク故障からの回復時のデータ読み出しを減らすために，冗長性を利用したディスクアレイを提案した．ディスクアレイに格納されるデータブロックには可逆圧縮と，消失訂正符号化を適用する．消失訂正符号には 3 つのモードがあり，圧縮後のデータブロックのサイズと記憶領域の空き状況により適応的に決定する．データが十分に圧縮され，データアレイに未使用領域が十分にある場合には，回復コストを減らすために符号化率の低いモードが選択される．簡易なシミュレーションの結果，回復コストを削減できることを示した．6 台のディスクで構成されたディスクアレイで，平均圧縮率が 0.6，ディスク使用率が 0.7 のときに単一故障からの回復に必要な読み出しデータ量は RAID-6 と比較して 0.24 倍に減った．

参考文献

[1] B. Schroeder and G. A. Gibson, “Disk Failures in the Real World: What Does an MTTF of 1,000,000 hours mean to you?,” *FAST 07,* vol. 7, pp. 1–16, Feb. 2007.

[2] P. M. Chen, et al., “RAID: High-performance, reliable secondary storage, ” *ACM Computing Surveys(CSUR)*, vol. 26,no 2, pp. 145–185, 1994.

[3] Lustre Software Release 2.x Operations Manual, 2013.

[4] E. Fujiwara, “Code design for dependability systems,” Jhon Wiley and Sons, Inc., 2006.

[5] X. Chen, L. Yang, L. Shang, and H. Lekatsas, “C-Pack: A High-Performance Microprocessor Cache Compression Algorithm,” *IEEE Trans. VLSI Systems,* vol. 18, no. 8, pp. 1196–1208, Oct. 2010.

[6] The DiskSim Simulation Environment, Version 4.0 Reference Manual, 2008.