

# SSD の性能と信頼性を考慮した データアクセスパターンに適した ECC システム

山賀 祐典<sup>1</sup> 松井 千尋<sup>1</sup> 竹内 健<sup>1</sup>

読み出し/書き込み頻度, ECC の信頼性を考慮し, データアクセスパターンや書き換え回数に応じて ECC を最適化する ECC システムを提案する. ECC システムで使用する ECC 手法の信頼性を, NAND フラッシュメモリを用いて計測し, 様々なデータアクセスパターンの実ワークロードを用いて, 提案する ECC システムの性能評価を行った. 提案手法では読み出しが支配的なワークロードで高い性能が得られた.

## 1. はじめに

近年, モノのインターネット(Internet of Thing, IoT)によって多種多様なデータを高速に処理する必要があり, データを蓄えるストレージも高速化が必須となっている. こうした高速化の要求を受け, ストレージの記憶媒体は従来の HDD から高速な NAND フラッシュメモリを使った SSD に移行しつつある. しかし NAND フラッシュメモリは書き換えを繰り返すにつれて, メモリセルの信頼性が劣化し, メモリの不良率が高まるという問題がある. メモリのエラーは誤り訂正符号(ECC)によって救済可能だが, 多数のエラーを救済できる強力な ECC を採用すると ECC の処理時間が長くなり, SSD の性能が劣化するトレードオフがある. 本論文ではアプリケーションや書き換え回数に応じて動的に ECC を最適化するシステムを提案し, 信頼性と SSD 性能の評価を行った.

## 2. アプリケーションや書き換え回数に応じて動的に ECC を最適化する ECC システム

本論文では, アプリケーションや書き換え回数に応じて動的に ECC を最適化する ECC システム, Application optimized adaptive (AOA-) ECC を提案する. 図 1 に従来のストレージと提案するストレージの構成図を示す. 従来のストレージでは, NAND フラッシュの微細化や書き換え回数増加によって発生するエラーを, 強力な訂正能力を有する Soft-decoding LDPC [1]を用いて訂正する. しかしデコードに必要なデータを取得するために余分な読み出し動作が 6 回必要であり, これが大きな遅延の原因となる. そのため従来のストレージでは, メモリエラーが少ないときは訂正能力が低くデコード遅延の少ない Bose-Chaudhuri-Hocquenghem (BCH)符号を用い, エラーが増えると Soft-decoding LDPC に切り替える必要がある. 提案するストレージではメモリセルのエラー予測を行うことで ECC の処理時間を大幅に削減した, 強度の異なる 2 つの LDPC 符号をアプリケーションに応じて切り替えることで, SSD 性能と信頼性のトレードオフを解決する. 書き込みが頻発するアプリケーションでは, ECC の処理時間が SSD の性能に与

える影響が少ないこと, 書き換えによってエラーが多数発生することから, 図 2 のように, 強力な訂正能力を有する Error Prediction (EP)-LDPC w/o upper/lower cells [2]を採用する. 一方, 読み出しが頻繁に行われ書き換えは比較的少ないアプリケーションでは, ECC の処理を短時間で行うことが出来る Quick-LDPC [2]を採用する.

図 3 に AOA-ECC で使用する Quick-LDPC と EP-LDPC w/o upper/lower cells の概要を示す. Quick-LDPC は読み出し対象の 1 ページ (Upper page もしくは Lower page)を読み出すことによって得られるしきい値情報と, 書き換え回数, データ保持時間からエラーを予測しデコードを行うため, 読み出し時間は 40 us もしくは 60 us である. 一方 EP-LDPC w/o upper/lower cells では, 読み出し対象のデータが存在する 1 ワードラインの 2 ページ (Upper page および Lower page)を読み出すことによって, Quick-LDPC では得られないセル間干渉データを用いるためより高い信頼性を持っているが, 読み出し時間は 102 us 必要となる. このように 2 つの LDPC 符号はエラーの予測方法が異なるが, 同じ LDPC デコーダを使用することが可能である. そのため図 1 のように, デコーダコストは従来の ECC システムに比べ 25% 削減することが可能である[2].

## 3. 提案する AOA-ECC の評価

図 4 のようにワークロード[3]を平均書き込み頻度, 読み出しデータの割合から 4 分類した. 1 週間のワークロードを使用し, 書き込み頻度が高く平均データ保持時間が 1 日以下(平均書き込み頻度 7 以上)となるものをホットとし, 1 日以上(平均書き込み頻度 7 以下)のとなるものをコールドとする.

図 5 に AOA-ECC で使用される 2 つの LDPC 符号の信頼性計測結果を示す. 強力な訂正能力を有する EP-LDPC w/o upper/lower cells では従来の BCH 符号に比べて 57%許容書き換え回数が向上していることがわかる. また, 書き換え頻度が高いワークロードでは平均データ保持時間は 1 日以下となり, 書き換え頻度が低いワークロードでは 6 日以上となる. ECC 処理時間の少ない Quick-LDPC から強力な訂正能力を

<sup>1</sup> 中央大学理工学部

有する EP-LDPC w/o upper/lower cells に切り替える書き換え回数は、書き換え頻度が低いワークロードに比べ、書き換え頻度が高いワークロードでは 2.3 倍増加する。そのため、アプリケーションによって ECC を切り替える書き換え回数を変える必要があることがわかる。

図 6 に読み出しが支配的で書き込み頻度が高いワークロード(prxy\_1)を用いた場合における、従来手法と提案する AOA-ECC の性能比較を示す。TLM (Transaction Level Modeling)ベースの SSD エミュレータ [2] を用いて SSD の性能を評価した。AOA-ECC (青)では、書き換え回数 2000 回程度で Quick-LDPC から EP-LDPC w/o upper/lower cells に切り替わっていることがわかる。従来手法では、BCH から Soft-decoding LDPC に切り替えると性能が著しく低下してしまうのに対し、AOA-ECC での性能は 1.8 倍、最大で 2.1 倍改善していることがわかる。様々なアプリケーションでの性能比較を図 7 に示す。従来手法では読み出しが支配的なアプリケーションでは性能低下が著しいのに対し、AOA-ECC では、書き込みが支配的なアプリケーションでは 1.5 倍、読み出しが支配的なアプリケーションでは 3 倍、従来手法

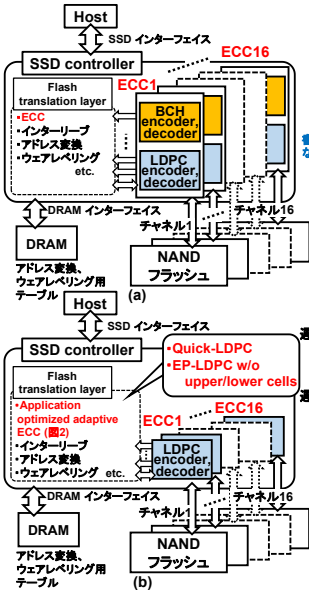


図 1 従来のストレージと提案するストレージの構成図 [2]

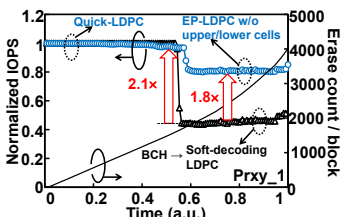


図 6 SSD 性能の変化 [2]

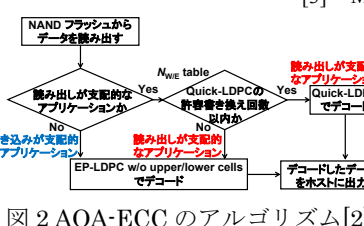


図 2 AOA-ECC のアルゴリズム [2]

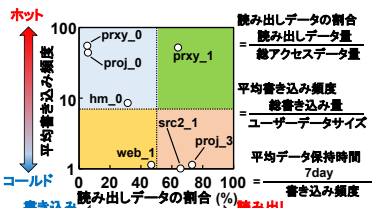


図 4 ワークロード分類 [2]

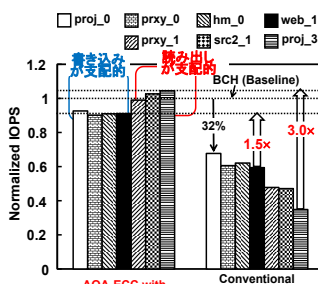


図 7 様々なアプリケーションでの性能評価 [2]

に比べ性能が改善していることがわかる。

#### 4. 結論

読み出し/書き込み頻度、ECC の信頼性を考慮し、データアクセスパターンや書き換え回数に応じて ECC を最適化する ECC システムを提案した。提案する ECC システムは従来手法に比べ、デコードコストを 25%削減し、信頼性を 57%改善した。ECC 処理時間の影響が大きい読み出しが支配的なアプリケーションにおいては最大で 3 倍 SSD 性能を改善することができる(表 1)。

#### 謝辞

本研究の一部は NEDO の助成によって行われた。蜂谷尚悟さんのご協力に深謝します。

#### 参考文献

- [1] C. Kim, et al., "A 2nm High Performance 64Gb MLC NAND Flash Memory with 400 MB/s Asynchronous Toggle DDR Interface," IEEE Journal of Solid-State Circuits, vol. 47, no. 4, pp. 981-989, April 2012.
- [2] Y. Yamaga, et al., "Application Optimized Adaptive ECC with Advanced LDPCs to Resolve Trade-off among Reliability, Performance, and Cost of Solid-State Drives," IEEE International Memory Workshop, pp. 129-132, May 2016.
- [3] MSR Cambridge Traces, <http://iotta.snia.org/traces/388>.

ECCの種類	Quick-LDPC [2]	EP-LDPC w/o upper/lower cells [2]
コンセプト	Lower page: $V_{read1}, V_{read2}, V_{read3}$ Upper page: $V_{read4}, V_{read5}, V_{read6}$	$V_{read1}, V_{read2}, V_{read3}, V_{read4}, V_{read5}, V_{read6}$
読み出し動作	1ページ読み出し(Lower or Upper) or 2 $V_{read}$ センシング (40 ~ 62 us)	1ワードライン読み出し(2ページ) 3 $V_{read}$ センシング (102 us)
考慮する情報	$V_{read}$ 情報 ("0" or "1") 書き換え回数・データ保持時間	$V_{read}$ 情報 ("00", "01", "10", "11") セル間干渉 書き換え回数・データ保持時間

図 3 提案手法で使用される LDPC 符号 [2]

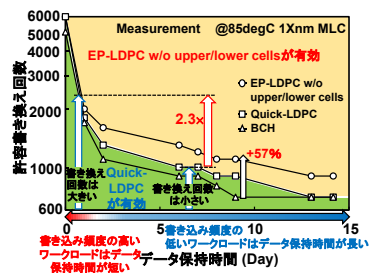


図 5 Quick-LDPC と EP-LDPC w/o upper/lower cells の信頼性計測結果 [2]

表 1 従来技術と AOA-ECC の比較 [2]

	SSD 性能 (IOPS)		信頼性	ECC デコードコスト
	書き込みが支配的なワークロード (web_1)	読み出しが支配的なワークロード (proj_3)		
Proposed AOA-ECC with advanced LDPCs	0.91x	1.05x	57% 改善	25% 削減 (470k) [2]
Conventional BCH → Soft-decoding LDPC [1]	Baseline	Baseline	Baseline	Baseline (159k) [2] +470k [2]