Access-optimal Linear MDS Convertible Codes for All Parameters

Francisco Maturana
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA, USA
Email: fmaturan@cs.cmu.edu

V. S. Chaitanya Mukka
Department of CS & IS
BITS Pilani, Goa Campus
Goa, India
Email: f20150536@goa.bits-pilani.ac.in

K. V. Rashmi
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA, USA
Email: rvinayak@cs.cmu.edu

***Abstract*—In large-scale distributed storage systems, erasure codes are used to achieve fault tolerance in the face of node failures. Tuning code redundancy to observed failure rates has been shown to significantly reduce storage cost. Such tuning of redundancy requires *code conversion*, i.e., a change in code dimension and length on already encoded data. *Convertible codes* [2] are a new class of codes designed to perform such conversions efficiently. The *access cost* of conversion is the number of nodes accessed during conversion.**

Existing literature has characterized the access cost of conversion of linear MDS convertible codes only for a specific and small subset of parameters. In this paper, we present lower bounds on the access cost of conversion of linear MDS codes for *all* valid parameters. Furthermore, we show that these lower bounds are tight by presenting an explicit construction for access-optimal linear MDS convertible codes for all valid parameters. En route, we show that, one of the degrees-of-freedom in the design of convertible codes that was inconsequential in the previously studied parameter regimes, turns out to be crucial when going beyond these regimes and adds to the challenge in the analysis and code construction.

An extended version of this paper is accessible at: [1]

I. Introduction

Erasure codes are an essential tool for providing resilience against node failures in a distributed storage system [3]–[9]. When using an $[n, k]$ erasure code, k chunks of data are encoded into n chunks, called a *stripe*. These chunks are then distributed among n different "nodes" in the system, where nodes correspond to distinct storage devices typically residing on distinct servers. For the purposes of theoretical study, each stripe can be viewed as a *codeword*, by viewing each of the n chunks as one of the n codeword symbols. The parameters n and k are usually chosen based on node failure rate, which might vary over time. Redundancy tuning, i.e., changing n and k in response to fluctuations in the failure rate of storage devices can achieve significant savings (11% to 44%) in storage space [10]. Due to practical system constraints, changing n alone is typically insufficient and both n and k have to be changed simultaneously [10]. The resource cost of changing n and k on already encoded data can be prohibitively high and is a key barrier in the practical adoption of redundancy tuning [2]. Other reasons to change n and k on already encoded data might include variations in data popularity, failure rate uncertainty, or restrictions on the total amount of used storage.

The *code conversion* problem defined in [2] involves converting multiple stripes of an $[n^I, k^I]$ code (denoted by $\mathcal{C}^I$) into (potentially multiple) stripes of an $[n^F, k^F]$ code (denoted by $\mathcal{C}^F$), along with desired constraints on decodability such as both codes being Maximum Distance Separable (MDS). Considering multiple stripes enables code conversions to allow for changes in the code dimension (from k^I to k^F). *Convertible codes* [2] are code pairs that enable code conversion, usually designed to minimize the cost of conversion. A detailed description of the convertible codes framework is provided in Section II-A.

There are several ways in which one might measure the cost of conversion. We focus on the *access cost* of conversion, which is measured in terms of the total number of nodes that need to be accessed during conversion. In [2], the authors focus on the so-called *merge* regime, wherein multiple initial stripes are merged into one. Specifically, they consider the case where $k^F = \varsigma k^I$ for some integer $\varsigma \geq 2$, and propose explicit constructions for converible codes that achieve optimal access cost for the merge regime. We review these results for the merge regime in Section II-B.

The results presented in this work are two fold. (1) We present lower bounds on the access cost of conversion for linear MDS codes *for all valid parameters*, that is, all $n^I, k^I, n^F, k^F \in \mathbb{N}^+$ such that $n^I > k^I$ and $n^F > k^F$. (2) We show that the proposed lower bounds are tight by presenting an *explicit construction* of linear MDS convertible codes that is access optimal for all parameter regimes. To achieve this, we first define and study the *split regime* in Section III, where $k^I = \varsigma k^F$ for an integer $\varsigma \geq 2$, that is, a single initial stripe is split into multiple final stripes. We prove a (tight) lower bound on the access cost of conversion in the split regime, and describe a conversion procedure which has optimal access cost when used with any systematic MDS code. We then present in Section IV a tight lower bound on the access cost of conversion for linear MDS convertible codes for all valid parameters (termed *general regime*) by reducing conversion in the general regime to a combination of generalizations of conversions in the split and merge regimes. While the split and the merge regimes might seem somewhat restrictive, we show

This work was funded in part by an NSF CAREER award (CAREER-1943409) and a Google faculty research award. We thank Michael Rudow for his suggestions in the writing of this paper.

978-1-7281-6432-8/20/$31.00 ©2020 IEEE

that, perhaps surprisingly, the proposed conversion procedure for the general regime that builds on top of the generalized split and merge regime is *optimal*. Interestingly, one of the degrees-of-freedom in the design of convertible codes (called "partitions" described subsequently in Section II-A), which is inconsequential in the split and merge regimes, turns out to be crucial in the general regime. The proposed construction for access-optimal convertible codes for the general regime builds on the constructions for split and merge regimes, while separately optimizing along this additional degree-of-freedom.

II. Background and Related work

A. Convertible codes [2]

A *conversion* from an $[n^I, k^I]$ initial code $\mathcal{C}^I$ to an $[n^F, k^F]$ final code $\mathcal{C}^F$ is a procedure that takes as input a set of initial stripes from $\mathcal{C}^I$ and outputs a set of final stripes from $\mathcal{C}^F$, such that the final stripes together encode the same information as the initial stripes. To avoid degeneracy, $n^F > k^F$ and $n^I > k^I$ is assumed. Let $\mathbb{F}_q$ be a finite field, and consider a message $\mathbf{m} \in \mathbb{F}_q^M$, where $M = \operatorname{lcm}(k^I, k^F)$. The number of initial stripes is $\lambda^I = {}^{M}/_{k^I}$ and the number of final stripes is $\lambda^F = {}^{M}/_{k^F}$. Let $[n] = \{1, \ldots, n\}$, $r^I = n^I - k^I$ and $r^F = n^F - k^F$. Let $\mathbf{m}[S]$ denote the projection of $\mathbf{m}$ onto the coordinates in the set S, and let $\mathcal{C}(\mathbf{m})$ denote the encoding of $\mathbf{m}$ under code $\mathcal{C}$. Consider an *initial partition* $\mathcal{P}^I = \{P_1^I, \ldots, P_{\lambda^I}^I\}$ of $[M]$ such that $|P_i^I| = k^I$ $(\forall i \in [\lambda^I])$, and a *final partition* $\mathcal{P}^F = \{P_1^F, \ldots, P_{\lambda^F}^F\}$ of $[M]$ such that $|P_j^F| = k^F$ $(\forall j \in [\lambda^F])$.

Definition 1 (Convertible code [2]): An $(n^I, k^I; n^F, k^F)$ convertible code over $\mathbb{F}_q$ is defined by: (1) a pair of codes $(\mathcal{C}^I, \mathcal{C}^F)$ over $\mathbb{F}_q$ such that $\mathcal{C}^I$ is $[n^I, k^I]$ and $\mathcal{C}^F$ is $[n^F, k^F]$; (2) a pair of partitions $(\mathcal{P}^I, \mathcal{P}^F)$ of $[M = \operatorname{lcm}(k^I, k^F)]$ such that $|P_i^I| = k^I$ for all $P_i^I \in \mathcal{P}^I$ and $|P_j^F| = k^F$ for all $P_j^F \in \mathcal{P}^F$; and (3) a conversion procedure which, for any $\mathbf{m} \in \mathbb{F}_q^M$, takes the set of initial codewords $\{\mathcal{C}^I(\mathbf{m}[P_i^I]) \mid P_i^I \in \mathcal{P}^I\}$ as input, and outputs the corresponding set of final codewords $\{\mathcal{C}^F(\mathbf{m}[P_j^F]) \mid P_j^F \in \mathcal{P}^F\}$.

In this paper, we will restrict our focus to the case where $\mathcal{C}^I$ and $\mathcal{C}^F$ are both linear and MDS.

The *access cost* of a conversion procedure is the total number of nodes that are read or written during conversion. Recall that each node in a stripe corresponds to a single symbol from the corresponding codeword, therefore access cost is equivalent to the number of codeword symbols that are read or written during conversion. We distinguish three types of nodes during conversion: *unchanged nodes*, which remain *as is* during the conversion process, and are present in both the initial and final configuration (possibly in different stripes); *retired nodes*, which are present in the initial configuration and throughout the conversion, but not in the final configuration; and *new nodes*, which are introduced during conversion, and are present in the final configuration, but not in the initial configuration. Unchanged and retired nodes may be accessed for reading during conversion, and new nodes are always accessed for writing during the conversion. A convertible code that maximizes the number of unchanged nodes is said to be *stable*.

The *read access set* of an $(n^I, k^I; n^F, k^F)$ convertible code is a set of tuples $\mathcal{D} \subseteq [\lambda^I] \times [n^I]$, where $(i, j) \in \mathcal{D}$ corresponds to the j-th node in initial stripe i. After a conversion, each new node holds a fixed linear combination of the contents of the nodes indexed by $\mathcal{D}$. We denote the accessed nodes from initial stripe i as $\mathcal{D}_i = \{j \mid (i, j) \in \mathcal{D}\}$. Thus, the access cost of a conversion with read access set of size $d = |\mathcal{D}|$ and m new nodes is $d + m$. Clearly, there always exists a conversion procedure with read access cost M, which reconstructs the original message $\mathbf{m}$ and re-encodes according to $\mathcal{C}^F$. We refer to this procedure as the *default approach*.

An $(n^I, k^I; n^F, k^F)$ convertible code is *access-optimal* if and only if it achieves the minimum access cost over all $(n^I, k^I; n^F, k^F)$ convertible codes.

B. Merge regime [2]

The *merge regime* is the subset of valid parameter values for convertible codes where $k^F = \varsigma k^I$, for some integer $\varsigma \geq 2$. Thus, in this regime we have $\lambda^I = \varsigma$ and $\lambda^F = 1$. This regime was the focus of [2], wherein the following lower bound on access cost was shown.

Theorem 1 ([2]): For all linear MDS $(n^I, k^I; n^F, k^F)$ convertible code, the access cost of conversion is at least $r^F + \varsigma \min\{k^I, r^F\}$. Furthermore, if $r^I < r^F$, the access cost of conversion is at least $r^F + \varsigma k^I$.

An explicit construction for access-optimal convertible codes for all values in the merge regime was also provided in [2].

C. Related work

The closest related work [2] proposes the convertible codes framework considered in this work (discussed at length above). Several other works in the literature [11]–[15] have considered variants of the code conversion problem, largely within the context of so-called "regenerating codes" [16]. The study on regenerating codes, which are a class of codes that optimize for recovery for a small subset of nodes within a stripe (as opposed to decoding all original data), was initiated by Dimakis et al. [16]. Subsequently numerous works have studied and constructed optimal regenerating codes (e.g., [17]–[31]). Specific instances of code conversion can be viewed as instances of the repair problem, for example, increasing n while keeping k fixed as studied in [11], [15].

D. Notation

This subsection introduces notation that generalizes the notation used in [2] and is used throughout this paper. Let $\mathbf{G}^\Diamond = (\mathbf{g}_1^\Diamond \cdots \mathbf{g}_{n^\Diamond}^\Diamond) \in \mathbb{F}_q^{k^\Diamond \times n^\Diamond}$ be a generator matrix of MDS code $\mathcal{C}^\Diamond$ for $\Diamond \in \{I, F\}$. An encoding vector in relation to $\mathbf{m} \in \mathbb{F}_q^M$ is associated to each node in the initial or final stripes. The encoding vector $\tilde{\mathbf{g}}_{i,j}^\Diamond \in \mathbb{F}_q^M$ of node $j \in [n^\Diamond]$ in stripe $i \in [\lambda^\Diamond]$ with partition set $P_i^\Diamond \in \mathcal{P}^\Diamond$ is defined such that $\tilde{\mathbf{g}}_{i,j}^\Diamond[P_i^\Diamond] = \mathbf{g}_j^\Diamond$, and 0 everywhere outside of $P_i^\Diamond$.

Let $\mathcal{S}_i^\Diamond = \{\tilde{\mathbf{g}}_{i,j}^\Diamond \mid j \in [n^\Diamond]\}$ be the encoding vectors for a particular stripe, and let $\mathcal{S}^\Diamond = \bigcup_{i \in [\lambda^\Diamond]} \mathcal{S}_i^\Diamond$. Let $\mathcal{U} = \mathcal{S}^I \cap \mathcal{S}^F$ be the encoding vectors of unchanged nodes, and define $\mathcal{U}_{i,j} = \mathcal{S}_i^I \cap \mathcal{S}_j^F$, where the index i or j is dropped if $\lambda^I = 1$ or

$\lambda^F = 1$, respectively. Let $\mathcal{A}_i = \{\tilde{\mathbf{g}}^I_{i,j} \mid j \in \mathcal{D}_i\}$ be the encoding vectors of nodes that are read from initial stripe i, and define $\mathcal{A} = \{\tilde{\mathbf{g}}^I_{i,j} \mid (i,j) \in \mathcal{D}\}$ as the set of all encoding vectors of nodes that are read. Finally, let $\mathcal{N} = \mathcal{S}^F \backslash \mathcal{S}^I$ be the encoding vectors of new nodes, and define $\mathcal{N}_i = \mathcal{S}^F_i \backslash \mathcal{S}^I$ as the encoding vectors of new nodes of a particular stripe i. Notice that it must hold that $\mathcal{N} \subseteq \operatorname{span}(\mathcal{A})$. For simplicity, we sometimes refer to a node and its encoding vector interchangeably.

III. Split regime

The *split regime* of convertible codes corresponds to the case where a single initial stripe is split into multiple final stripes. This regime is, in some sense, the opposite of the merge regime, in which multiple initial stripes are combined into one final stripe. Specifically, an $(n^I, k^I; n^F, k^F)$ convertible code is in the split regime if $k^I = \varsigma k^F$ for an integer $\varsigma \geq 2$, with arbitrary n^I and n^F. Notice that in this regime we have that $M = \operatorname{lcm}(k^I, k^F) = k^I$ and thus $\lambda^I = 1$ and $\lambda^F = \varsigma$.

A. Access cost lower bound for the split regime

In this subsection, we lower bound the access cost of conversion in the split regime. This is done by first showing a lower bound on write access cost, and then showing a lower bound on the read access cost of conversion.

The following fact simplifies the analysis of the split regime.

Proposition 2: For a linear MDS $(n^I, k^I = \varsigma k^F; n^F, k^F)$ convertible code, all possible pairs of initial and final partitions are equivalent (up to relabeling).

Proof: There is only one possible initial partition $\mathcal{P}^I = \{[k^I]\}$, hence any two final partitions can be made equivalent by relabeling nodes. ■

Therefore, we do not need to consider differences in partitions in our analysis of the split regime.

Proposition 3: In a linear MDS $(n^I, k^I = \varsigma k^F; n^F, k^F)$ convertible code, there are at most k^F unchanged nodes in each of the final stripes (i.e., at least r^F new nodes per stripe). Hence, there are at most k^I unchanged nodes in total.

Proof: For any final stripe $i \in [\varsigma]$, any subset $\mathcal{V} \subseteq \mathcal{S}^F_i$ of size at least $k^F + 1$ is linearly dependent due to the MDS property. Thus, $\mathcal{V} \subseteq \mathcal{S}^I$ contradicts the fact that $\mathcal{C}^I$ is MDS. Hence, each final stripe i has at most k^F unchanged nodes. ■

Therefore, the total write access cost in the split regime is at least ςr^F.

Now we focus on bounding the read access cost. We obtain the following lower bound as a consequence of the MDS property of the initial and final code.

Lemma 4: For all linear MDS $(n^I, k^I = \varsigma k^F; n^F, k^F)$ convertible codes, the read access set $\mathcal{D}$ satisfies $|\mathcal{D}| \geq (\varsigma - 1)k^F + \min\{r^F, k^F\}$.

Proof sketch: We consider a node u from a final stripe that is neither read nor written during conversion and select a subset of nodes $\mathcal{W}$ of size k^F from the same final stripe containing r^F new nodes and not containing u. (If such a node u does not exist, then all final nodes are either read or written and the result follows easily.) By the MDS property of the final code, u can be recovered from $\mathcal{W}$. However, by the MDS property of the initial code and the rank of $\mathcal{W}$ one can show that $\mathcal{W}$ cannot contain any information about u unless $|\mathcal{D}| \geq (\varsigma - 1)k^F + \min\{r^F, k^F\}$. ■

However, we next show that it is not possible to achieve less read access cost than the default approach when $r^I < r^F$.

Lemma 5: For all linear MDS $(n^I, k^I = \varsigma k^F; n^F, k^F)$ convertible codes, if $r^I < r^F$ then the read access set $\mathcal{D}$ satisfies $|\mathcal{D}| \geq \varsigma k^F$.

Proof sketch: We follow the same strategy as the proof of Lemma 4. We consider an unaccessed node u, and select a set $\mathcal{W}$ of size k^F containing the most amount of accessed nodes from the same final stripe and not containing u. If all nodes in $\mathcal{W}$ are accessed, the results follows easily. Otherwise, we can use the fact that $r^I < r^F$ to limit the number of read-accessed nodes in the initial stripe that are not part of $\mathcal{W}$, and thus prove the bound. ■

By combining all the results in this subsection, we obtain the following lower bound on the access cost of conversion in the split regime.

Theorem 6: The total access cost of any linear MDS $(n^I, k^I = \varsigma k^F; n^F, k^F)$ convertible code is at least $(\varsigma - 1)k^F + \min\{r^F, k^F\} + \varsigma r^F$ if $r^I \geq r^F$, and at least ςn^F otherwise.

Proof: Follows from Proposition 3 and Lemmas 4–5. ■

As we show in the next subsection, this lower bound is tight since it is achievable.

B. Access-optimal convertible codes for the split regime

In this subsection we present a construction of access-optimal convertible codes in the split regime. Under this construction, any systematic MDS code can be used as the initial code. The final code corresponds to the projection of the initial code onto the coordinates of any k^F systematic nodes. Since our construction can be applied to existing codes and only specifies the conversion procedure, we introduce the following definition capturing the property of codes that can be converted efficiently.

Definition 2: A code $\mathcal{C}^I$ is (n^F, k^F)*-optimally convertible* if and only if there exists an $[n^F, k^F]$ code $\mathcal{C}^F$ (along with partitions and conversion procedure) that form an access-optimal $(n^I, k^I; n^F, k^F)$ convertible code.

The conversion procedure that leads to optimal access cost (meeting the lower bound in Theorem 6) is as follows.

Conversion procedure: All the systematic nodes are used as unchanged nodes. When $r^I < r^F$ or $r^F \geq k^F$, the conversion is trivial since one cannot do better than the default approach. The conversion procedure for the nontrivial case proceeds as follows. For all but one final stripe, all unchanged nodes are read ($(\varsigma - 1)k^F$ in total), and the new nodes are naively constructed from them. For the remaining final stripe, r^F retired nodes are read, and then the unchanged nodes from the other final stripes are used to remove their interference from the retired nodes to obtain r^F new nodes.

Theorem 7: Every systematic linear MDS $[n^I, k^I = \varsigma k^F]$ code $\mathcal{C}^I$ is (n^F, k^F)-optimally convertible.

Proof sketch: The generator matrix of the final code is defined as the first k^F rows of the initial code's generator matrix. As described in the conversion procedure above, conversion can be realized by accessing all systematic nodes except for those corresponding to the first k^F columns (which correspond to the unchanged nodes of all but one initial stripe), along with r^F parity nodes (which correspond to the retired nodes). Interference from all but one final stripe can be removed from the r^F parity nodes using the accessed systematic nodes. ■

Notice that convertible codes created using the construction above are stable. We show this property is, in fact, necessary.

Lemma 8: All access-optimal convertible codes for the split regime are stable.

Proof: Theorem 7 shows that there exist stable access-optimal codes for the split regime. Since any unstable convertible code must incur higher write access cost and at least as much read access cost, it cannot be access-optimal. ■

IV. General regime

In this section, we will study the general regime of convertible codes with arbitrary valid parameter values (i.e. any $n^I > k^I$ and $n^F > k^F$). Recall that the choice of partition functions was inconsequential in the split and merge regimes. In contrast, it turns out that *the choice of initial and final partitions play an important role in the general regime*. This makes the general regime significantly harder to analyze. We deal with this complexity by reducing conversion in the general regime to generalized versions of the split and merge conversions, and by *identifying the conditions on initial and final partitions to minimize total access cost*.

In Section IV-A, we explore a generalization of the split regime and of the merge regime. In Section IV-B, these generalizations are used to lower bound the access cost of conversion in the general regime. In Section IV-C, we describe a conversion procedure and construction for access-optimal conversion in the general regime which utilizes ideas from the constructions for generalizations of split and merge regimes.

A. Generalized split and merge regimes

The generalized split and merge regimes are similar to the split and merge regimes, except that the generalized variants allow for initial or final stripes of unequal sizes. This flexibility enables the generalized split and merge regimes to be used as building blocks in the analysis of the general regime. In these generalized variants, the message length M is defined to be $\max\{k^I, k^F\}$ (which coincides with the definition of M in the split and merge regime), but now the sets in the initial and final partitions need not be all of the same size.

Since the initial (or final) stripes might be of different lengths, we define them as shortenings of a common code $\mathcal{C}$.

Definition 3: An s-shortening of an $[n, k]$ code $\mathcal{C}$ is the code $\mathcal{C}'$ formed by all the codewords of $\mathcal{C}$ that have 0 in a fixed subset of s positions.

It can be shown that the s-shortening of an MDS $[n, k]$ code is an MDS $[n-s, k-s]$ code.

1) Generalized split regime: In the generalized split regime, $\lambda^I = 1$ is fixed, $\lambda^F > 1$ is arbitrary, and the final partition $\mathcal{P}^F = \{P_1^F, \ldots, P_{\lambda^F}^F\}$ is such that $|P_i^F| = k_i^F$ and $\sum_{i\in[\lambda^F]} k_i^F = k^I$. Let $k_*^F = \max_{i\in[\lambda^F]} k_i^F$. Then $\mathcal{C}^F$ is a $[n^F, k_*^F]$ MDS code, and the code corresponding to each final stripe is some fixed shortening of $\mathcal{C}^F$. In this case, we define $r^F = n^F - k_*^F$.

Definition 4: A $(n^I, k^I = \sum_{i=1}^{\lambda^F} k_i^F; n^F, \{k_i^F\}_{i=1}^{\lambda^F})$ convertible code for the generalized split regime is a variant of a convertible code defined by (1) $\mathcal{C}^I$ and $\mathcal{C}^F$ as $[n^I, k^I]$ and $[n^F, k_*^F]$ codes, where $k_*^F = \max_{i\in[\lambda^F]} k_i^F$, (2) a partition $\mathcal{P}^F = \{P_1^F, \ldots, P_{\lambda^F}^F\}$ where $|P_i^F| = k_i^F$, and (3) a conversion procedure such that each final stripe i, is an s_i-shortening of $\mathcal{C}^F$ where $s_i = k_*^F - k_i^F$.

The generalized split regime has an access cost lower bound similar to the split regime presented in Section III. We show this by showing that a more efficient conversion procedure for the generalized split regime would imply the existence of a conversion procedure for split regime violating Theorem 6.

Theorem 9: For all linear MDS $(n^I, k^I = \sum_{i=1}^{\lambda^F} k_i^F; n^F, \{k_i^F\}_{i=1}^{\lambda^F})$ convertible codes, the read access set $\mathcal{D}$ satisfies $|\mathcal{D}| \geq k^I - \max\{k_*^F - r^F, 0\}$, where $k_*^F = \max_{i\in[\lambda^F]} k_i^F$.

Proof sketch: Proof is via contradiction. Assume there is a conversion procedure for a convertible code in the generalized split regime with $|\mathcal{D}| < k^I - \max\{k_*^F - r^F, 0\}$. We modify the code by adding extra "pseudo-nodes" so that every final stripe is the same size. The conversion procedure is modified to access all pseudo-nodes. The resulting code is in the split regime. Since accessing pseudo-nodes does not add to the actual access cost, we can invoke Theorem 6 to obtain a contradiction. ■

This lower bound is achievable for all pairs of initial and final parameters. Similar to the case of the split regime, shown in Section III-B, we can use any systematic MDS codes as initial and final codes, and access all but a set of nodes of size k_*^F (forming the largest final stripe) to perform this conversion.

2) Generalized merge regime: In the generalized merge regime, the sets in the initial partition need not be all of the same size. In this case, we fix $M = k^F$ and $\lambda^F = 1$, while $\lambda^I > 1$ is arbitrary. The initial partition $\mathcal{P}^I = \{P_1^I, \ldots, P_{\lambda^I}^I\}$ is such that $|P_i^I| = k_i^I$ and $\sum_{i\in[\lambda^I]} k_i^I = k^F$. Let $k_*^I = \max_{i\in[\lambda^I]} k_i^I$. Then $\mathcal{C}^I$ is a $[n^I, k_*^I]$ MDS code, $r^I = n^I - k_*^I$, and the code corresponding to each initial stripe is some fixed shortening of $\mathcal{C}^I$.

Definition 5: A $(n^I, \{k_i^I\}_{i=1}^{\lambda^I}; n^F, k^F = \sum_{i=1}^{\lambda^I} k_i^I)$ convertible code for the generalized merge regime is a variant of a convertible code defined by (1) $\mathcal{C}^I$,$\mathcal{C}^F$ as $[n^I, k_*^I]$ and $[n^F, k^F]$ codes, where $k_*^I = \max_{i\in[\lambda^I]} k_i^I$ (2) partition $\mathcal{P}^I = \{P_1^I, \ldots, P_{\lambda^I}^I\}$ where $|P_i^I| = k_i^I$, and (3) a conversion procedure such that each initial stripe i, is an s_i-shortening of $\mathcal{C}^I$ where $s_i = k_*^I - k_i^I$.

The next theorem gives a lower bound on the read access cost of a $(n^I, \{k_i^I\}_{i=1}^{\lambda^I}; n^F, k^F = \sum_{i=1}^{\lambda^I} k_i^I)$ convertible code.

Theorem 10: For all $(n^I, \{k_i^I\}_{i=1}^{\lambda^I}; n^F, k^F = \sum_{i=1}^{\lambda^I} k_i^I)$ convertible code, $|\mathcal{D}_i| \geq \min\{k_i^I, r^F\}$ for all $i \in [\lambda^I]$. Furthermore, if $r^I < r^F$, then $|\mathcal{D}_i| \geq k_i^I$ for all $i \in [\lambda^I]$.

Proof: Follows from the proofs of Lemmas 10, 11, and 13 in [2], with some straightforward modifications. ■

We can achieve this lower bound by shortening an access-optimal $(n^I, k_*^I; n_m^F, k_m^F)$ convertible code, where $k_m^F = \lambda^I k_*^I$

and $n_m^F = k_m^F + r^F$.

B. Access cost lower bound for the general regime

In this subsection, we study the access cost lower bound for conversions in the general regime (i.e., for all valid parameter values, $n^I > k^I$ and $n^F > k^F$). As in the merge and split regime, we show that when $r^I \geq r^F$, significant reduction in access cost can be achieved. However when $r^I < r^F$, one cannot do better than the default approach.

For an $(n^I, k^I; n^F, k^F)$ convertible code with $k^I \neq k^F$ and partitions $(\mathcal{P}^I, \mathcal{P}^F)$, let $k_{i,j} = |P_i^I \cap P_j^F|$ for $(i,j) \in [\lambda^I] \times [\lambda^F]$ and let $k_{i,*} = \max_{j \in [\lambda^F]} k_{i,j}$.

Lemma 11: For all linear MDS $(n^I, k^I; n^F, k^F)$ convertible codes with $k^I \neq k^F$, $|\mathcal{D}_i| \geq k^I - \max\{k_{i,*} - r^F, 0\}$ for all $i \in [\lambda^I]$. Moreover, if $r^I < r^F$ then $|\mathcal{D}_i| \geq k^I$ for all $i \in [\lambda^I]$.

Proof sketch: If we consider only the nodes from a single *initial* stripe, and set all other initial nodes to zero, we can view conversion as a conversion in the generalized split regime and use Theorem 9. If we consider only the nodes from a single *final* stripe, and set all other final nodes to zero, we can view conversion as a conversion in the generalized merge regime and use Theorem 10. ■

We prove a lower bound on the total access cost of conversion in the general regime by using Lemma 11 on all initial stripes and finding a partition that minimizes the value of the sum.

Theorem 12: For every linear MDS $(n^I, k^I; n^F, k^F)$ convertible code such that $k^I \neq k^F$, it holds that $|\mathcal{D}| \geq \lambda^I r^F + (\lambda^I \bmod \lambda^F)(k^I - \max\{k^F \bmod k^I, r^F\})$ if $r^F < \min\{k^I, k^F\}$. Furthermore, if $r^I < r^F$ or $r^F \geq \min\{k^I, k^F\}$, then $|\mathcal{D}| \geq M$.

Proof sketch: The case $r^I < r^F$ follows directly from Lemma 11. Otherwise, by the same lemma we have $|\mathcal{D}| \geq \sum_{i=1}^{\lambda^I} k^I - \max\{k_{i,*} - r^F, 0\}$. It can be shown that the right hand side of this inequality is minimized when $k_{i,*} = k^I$ for $1 \leq i \leq \lambda^I - (\lambda^I \bmod \lambda^F)$ and $k_{i,*} = k^F \bmod k^I$ otherwise. Therefore, this yields a lower bound valid for any valid assignment $\{k_{i,j} \mid (i,j) \in [\lambda^I] \times [\lambda^F]\}$ and thus any initial and final partitions. When these values are replaced back into the inequality, we obtain the desired lower bound. ■

C. Access-optimal convertible codes for the general regime

In this subsection we prove that the lower bound from Theorem 12 is achievable by presenting convertible code constructions that are access-optimal in the general regime. We first present the conversion procedure for our construction and then describe the construction of the initial and final codes that are compatible with this conversion procedure.

1) Conversion procedure: Conversion in the general regime can be achieved by combining the conversion procedures of codes in the generalized split and merge regimes. In the case where $r^I < r^F$, we access k^I nodes from each initial stripe and use the default approach. For the case where $r^I \geq r^F$, we present the conversion procedure by considering three cases: $k^I = k^F$, $k^I < k^F$, and $k^I > k^F$.

Case $k^I = k^F$: This is a degenerate case where any n^F nodes from the initial stripe can be kept unchanged.

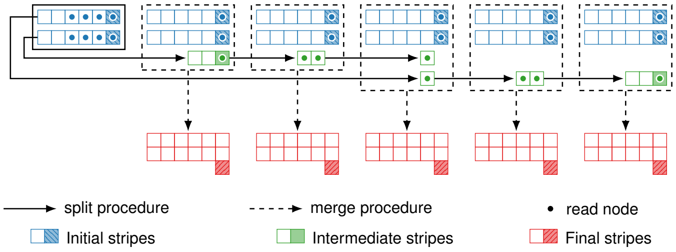

Fig. 1. Conversion procedure from $[6,5]$ to $[13,12]$ ($\lambda^I = 12$ and $\lambda^F = 5$). Read access cost is 18 compared to 60 in the default approach (70% savings).

Case $k^I < k^F$: We will separate the nodes of initial stripes into λ^F disjoint groups with the same amount of information. This requires splitting some initial stripes into what we call *intermediate stripes*, which are then assigned to different groups. We will finally merge each group to form the λ^F final stripes. Specifically (see Fig. 1): (1) Assign $\lfloor k^F / k^I \rfloor$ initial stripes to each group (dashed boxes in Fig. 1). (2) Use an $(n^I, k^I; n^F, \{k_i^F\}_{i=1}^{\hat{\lambda}^F})$ conversion procedure to (generalized) split the $(\lambda^I \bmod \lambda^F)$ remaining initial stripes to obtain $\hat{\lambda}^F$ intermediate stripes, where $\hat{\lambda}^F = \lceil k^I / (k^F \bmod k^I) \rceil$, $k_i^F = (k^F \bmod k^I)$ for $i \in [\hat{\lambda}^F - 1]$, and $k_{\hat{\lambda}^F}^F = (k^F \bmod k^I)$ if $(k^F \bmod k^I) \mid k^I$ and $k_{\hat{\lambda}^F}^F = (k^I \bmod (k^F \bmod k^I))$ otherwise. Each intermediate stripe is assigned to a different group. (3) The conversion procedure for generalized merge is used to turn each stripe group into a single final stripe.

The total number of nodes read during conversion is $\lambda^I r^F + (\lambda^I \bmod \lambda^F)(k^I - \max\{k^F \bmod k^I, r^F\})$, which matches Theorem 12.

Case $k^I > k^F$: Conversion occurs in two steps. In the first step, each initial stripe is split to form as many final stripes as possible. The leftover nodes are then merged into the remaining final stripes. See [1] for more details.

The total number of nodes read in this case during conversion is $\lambda^I (r^F + k^I - k^F)$, which matches Theorem 12.

Therefore, the total access cost of conversion when $r^I \geq r^F$ and $k^I \neq k^F$ is $(\lambda^I + \lambda^F) r^F + (\lambda^I \bmod \lambda^F)(k^I - \max\{k^F \bmod k^I, r^F\})$, while the access cost of the default approach is $\lambda^F n^F$.

2) Access-optimal construction: Since the conversion procedure in Section IV-C1 is based on the generalized split and merge regimes, we only need to ensure that the constructed codes can perform those conversions with optimal access cost.

Theorem 13: For all $k^F \leq k^I$, every systematic linear MDS $[n^I, k^I]$ code $\mathcal{C}^I$ is (n^F, k^F)-optimally convertible. For all $k^F \leq \varsigma k^I$ with integer $\varsigma > 2$, every access-optimal systematic linear MDS $(n^I, k^I; n^F, k^F = \varsigma k^I)$ convertible code is (n^F, k^F)-optimally convertible.

Proof sketch: From Section IV-A1, any systematic initial code can be used in access-optimal codes in the generalized split regime. From Section IV-A2, an access-optimal code from the merge regime can be used in an access-optimal code from the generalized merge regime if $\varsigma \geq \lambda^I$. ■

Therefore, the constructions for the merge regime presented in [2] can be used to construct access-optimal convertible codes in the general regime.

References

[1] F. Maturana, V. S. C. Mukka, and K. V. Rashmi, "Access-optimal linear MDS convertible codes for all parameters," *available on arXiv*, 2020.

[2] F. Maturana and K. V. Rashmi, "Convertible codes: New class of codes for efficient conversion of coded data in distributed storage," in *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), T. Vidick, Ed., vol. 151. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020, pp. 66:1–66:26.

[3] S. Ghemawat, H. Gobioff, and S. Leung, "The Google file system," in *ACM SIGOPS Operating Systems Review*, vol. 37-5. ACM, 2003, pp. 29–43.

[4] D. Borthakur, R. Schmidt, R. Vadali, S. Chen, and P. Kling, "HDFS RAID - Facebook." [Online]. Available: http://www.slideshare.net/ydn/hdfs-raid-facebook

[5] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in Windows Azure storage," in *Proceedings of USENIX Annual Technical Conference (ATC)*, 2012.

[6] Apache Software Foundation, "Apache hadoop: HDFS erasure coding," accessed: 2019-07-23. [Online]. Available: https://hadoop.apache.org/docs/r3.0.0/hadoop-project-dist/hadoop-hdfs/HDFSErasureCoding.html

[7] K. V. Rashmi, N. B. Shah, D. Gu, H. Kuang, D. Borthakur, and K. Ramchandran, "A solution to the network challenges of data recovery in erasure-coded distributed storage systems: A study on the Facebook warehouse cluster," in *Proceedings of USENIX HotStorage*, Jun. 2013.

[8] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "XORing elephants: Novel erasure codes for big data," in *VLDB Endowment*, 2013.

[9] K. V. Rashmi, N. B. Shah, D. Gu, H. Kuang, D. Borthakur, and K. Ramchandran, "A Hitchhiker's guide to fast and efficient data reconstruction in erasure-coded data centers," in *ACM SIGCOMM*, 2014.

[10] S. Kadekodi, K. V. Rashmi, and G. R. Ganger, "Cluster storage systems gotta have HeART: improving storage efficiency by exploiting disk-reliability heterogeneity," *USENIX FAST*, 2019.

[11] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Enabling node repair in any erasure code for distributed storage," in *2011 IEEE International Symposium on Information Theory Proceedings*. IEEE, 2011, pp. 1235–1239.

[12] B. K. Rai, V. Dhoorjati, L. Saini, and A. K. Jha, "On adaptive distributed storage systems," in *2015 IEEE international symposium on information theory (ISIT)*. IEEE, 2015, pp. 1482–1486.

[13] Y. Hu, X. Zhang, P. P. Lee, and P. Zhou, "Generalized optimal storage scaling via network coding," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 956–960.

[14] M. Sonowal and B. K. Rai, "On adaptive distributed storage systems based on functional MSR code," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2017, pp. 338–343.

[15] S. Mousavi, T. Zhou, and C. Tian, "Delayed parity generation in MDS storage codes," in *2018 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2018, pp. 1889–1893.

[16] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, sep 2010.

[17] D. Papailiopoulos, A. Dimakis, and V. Cadambe, "Repair optimal erasure codes through Hadamard designs," *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 3021–3037, May 2013.

[18] V. R. Cadambe, C. Huang, J. Li, and S. Mehrotra, "Polynomial length MDS codes with optimal repair in distributed storage," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*. IEEE, 2011, pp. 1850–1854.

[19] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Transactions on Information Theory*, vol. 59, no. 3, pp. 1597–1616, 2013.

[20] V. Guruswami and M. Wootters, "Repairing Reed-Solomon codes," in *ACM Symposium on Theory of Computing*, 2016, pp. 216–226.

[21] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5227–5239, 2011.

[22] B. Sasidharan, G. K. Agarwal, and P. V. Kumar, "A high-rate MSR code with polynomial sub-packetization level," in *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2015, pp. 2051–2055.

[23] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Distributed storage codes with repair-by-transfer and non-achievability of interior points on the storage-bandwidth tradeoff," *IEEE Transactions on Information Theory*, vol. 58, no. 3, pp. 1837–1852, Mar. 2012.

[24] M. Ye and A. Barg, "Explicit constructions of high-rate MDS array codes with optimal repair bandwidth," *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 2001–2014, 2017.

[25] K. Mahdaviani, S. Mohajer, and A. Khisti, "Product matrix MSR codes with bandwidth adaptive exact repair," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 3121–3135, 2018.

[26] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Interference alignment in regenerating codes for distributed storage: Necessity and code constructions," *IEEE Transactions on Information Theory*, vol. 58, no. 4, pp. 2134–2158, 2011.

[27] C. Suh and K. Ramchandran, "Exact-repair MDS code construction using interference alignment," *IEEE Transactions on Information Theory*, pp. 1425–1442, Mar. 2011.

[28] S. El Rouayheb and K. Ramchandran, "Fractional repetition codes for repair in distributed storage systems," in *Allerton Conference on Control, Computing, and Communication*, Urbana-Champaign, Sep. 2010.

[29] A. Chowdhury and A. Vardy, "New constructions of MDS codes with asymptotically optimal repair," in *2018 IEEE International Symposium on Information Theory*, 2018, pp. 1944–1948.

[30] H. Dau and O. Milenkovic, "Optimal repair schemes for some families of full-length Reed-Solomon codes," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 346–350.

[31] K. V. Rashmi, N. B. Shah, and K. Ramchandran, "A piggybacking design framework for read-and download-efficient distributed storage codes," *IEEE Transactions on Information Theory*, vol. 63, no. 9, pp. 5802–5820, 2017.