Local Codes With Cooperative Repair in Distributed Storage of Cyber-Physical-Social Systems

JING WANG1, ZHIYUAN YAN2, (Senior Member, IEEE), KUAN-CHING LI3, (Senior Member, IEEE), HONGMEI XIE4, AND XIANGYANG LIU5

1School of Information Engineering, Chang’an University, Xi’an 710064, China
2Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA 18015, USA
3Department of Computer Science and Information Engineering, Providence University, Taichung City 43301, Taiwan
4Seagate Technology, Fremont, CA 94538, USA
5College of Information and Communication, National University of Defense Technology, Xi’an 710106, China

Corresponding author: Jing Wang (jingwang@chd.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61501058, in part by the Natural Science Foundation of Shaanxi Province under Grant 2019JM-386, and in part by the Special Fund for Basic Scientific Research of Central Colleges, Chang’an University under Grant 300102248104.

ABSTRACT Integrating cyber, physical, and social spaces together, cyber-physical-social systems (CPSS) bring more conveniences to humans. For practical applications and user convenience, it is essential that the Big Data produced in CPSS be stored in the distributed storage systems of CPSS. In this paper, we study the fault tolerance scheme for distributed storage systems of CPSS, and propose a framework that can recover multiple failed nodes simultaneously. Considering the reliability of storage nodes in distributed storage systems, the research on locally repairable codes has mostly focused on repairing failed nodes within each repair group. However, when entire repair groups have failed, existing locally repairable codes cannot repair more than one failed group. In this paper, local codes with cooperative repair that can recover more than one failed group are proposed. Specifically, the proposed local codes are constructed based on minimum storage regenerating (MSR) codes, and have an interleaving structure among the local codes, so that the parity symbols of any local code can be generated from the MSR codes in its two adjacent local codes. Taking advantage of this property, more than one failed local group can be repaired cooperatively by their adjacent local groups with lower repair locality. Furthermore, the key parameters of local codes with cooperative repair are derived. Theoretical analysis and simulation results show that, compared with previous codes with local regeneration, our codes have higher bandwidth overhead when repairing failed nodes, but advantages in storage overhead and repair locality either for repair of a single failed node or one failed local group. Moreover for a single failed local group, local codes with cooperative repair achieve almost the same tradeoff curve of storage overhead and bandwidth overhead as MSR-local codes and minimum bandwidth regenerating local (MBR-local) codes.

INDEX TERMS Cyber-physical-social systems (CPSS), distributed storage systems, local codes with cooperative repair, minimum storage regenerating (MSR) codes, storage overhead, repair bandwidth overhead, repair locality.

I. INTRODUCTION

Since the traditional cloud storage model runs in a centralized storage manner, single node of failure may lead to the collapse of system. With the development of Cyber-physical-social systems (CPSS), distributed storage mode has entered the public view. The distributed storage approach can solve the problem of single node of failure in traditional cloud storage systems and enjoy a number of advantages over centralized storage, such as low price and high throughput. Also CPSS have seen significant adoption in the past few years and show promise to design applications without any centralized
reliance on third parties. Accordingly, the reliable storage of distributed storage systems is attracting more and more attention. Also, Wei et al. discussed the secure data storage and recovery in industrial network environments of CPSS in [1]–[3].

In large distributed storage systems (DSSs), node failure is inevitable along with data loss. To retain high availability of DSSs, Dimakis et al. introduced regenerating codes based on network coding [4], which can reduce repair bandwidth compared with a combination of replication and erasure codes. Rashmi et al. presented explicit constructions of regenerating codes that reach the two extreme points [5], termed as the minimum storage regenerating (MSR) and minimum bandwidth regenerating (MBR) points, achieving the minimum storage cost and minimum bandwidth cost during node repair process respectively. Furthermore, Ernvall gave the constructions of regenerating codes between the MSR point and the MBR point [6], [7].

Apart from storage and repair bandwidth overhead, repair locality, the number of nodes contacted by the replacement node in the repair process [8]–[10], is an important cost metric of data repair. The repair locality relates to the disk input/output (I/O) overhead, which has been the main performance bottleneck during repairing failed nodes. To reduce the disk I/O overhead, Papailiopoulos et al. proposed simple regenerating codes that employ simple XORs over maximum distance separable (MDS) coded packets to perform exact repair [11]. Moreover, Papailiopoulos and Dimakis presented optimal locally repairable codes (LRCs) based on simple combinations of RS codes that can achieve arbitrarily high data rates and better repair locality [12]. Based on maximum rank distance (MRD) Gabidulin codes, Rawat et al. constructed optimal LRCs with two-layer encoding structure [13], and derived the upper bound on the amount of data stored in DSSs. However, considering the properties of Gabidulin codes [14], the complexity of the optimal LRCs will increase exponentially with the number of nodes in DSSs.

Kamath et al. focused on the construction of optimal codes with local regeneration [15], which combine the advantages of both LRCs and regenerating codes, aiming to simplify node repair and reduce encoding complexity. The constructed optimal codes with local regeneration, such as MSR-local codes and MBR-local codes, contain multiple local codes and global parities, in which the local codes are either MSR codes or MBR codes. Also, Rawat et al. gave another construction of MSR-LRC [16], that the file is encoded using a Gabidulin code, the codeword of the Gabidulin code is partitioned into local groups, and each of these local groups is encoded using an MSR code.

When one entire repair group has failed, the local code in the failed group needs to be recovered as a whole. MSR-local codes and MBR-local codes proposed by Kamath et al. [15] have the ability to repair the failed local codes, even though their initial purpose was to simplify node repair. Nevertheless, based on their construction, MSR-local codes and MBR-local codes can repair only one failed local code, but are incapable of repairing two or more failed local codes at the same time. Furthermore, it is necessary for MSR-local codes and MBR-local codes to collect all the remaining local codes and the global parities for repairing the only failed local code, which leads to a high repair locality.

Inspired by the distributed parity of RAID 5 [17], the global parities of MSR-local codes may also be broken down into multiple distributed parities to avoid the limitations above of repairing failed local codes as MSR-local codes. In this paper, we present an explicit construction of local codes with cooperative repair (LCCR). Compared with MSR-local codes or MBR-local codes, each local code of LCCR includes an MSR code and a distributed parity. Particularly, the MSR code in LCCR is systematic, including information symbols and parity symbols. Furthermore, the distributed parity part of each local code in LCCR can be generated from the MSR codes in its two adjacent local codes, which can be regarded as an interleaving structure among the local codes. Based on the structure of the proposed LCCR, more than one failed local group can be recovered cooperatively by their adjacent groups with lower repair locality. Additionally, the key parameters of LCCR are derived. Theoretical and numerical analyses show that, the proposed LCCR have performance improvement in repair locality for node failures and local group failures, and achieve almost the same tradeoff curve of storage overhead and bandwidth overhead as MSR-local codes and MBR-local codes for a single failed local group.

This paper is organized as follows. In Section 2, we review relevant background. Explicit construction of LCCR is presented in Section 3, followed by the derivation of the minimum distance of LCCR. Section 4 provides the performance analyses, including repair of a single failed node, one failed local group, and multiple failed local groups. Section 5 concludes the paper.

II. BACKGROUND

A. MSR CODES AND MBR CODES

In a DSS, a message file of $M$ symbols over $GF(q)$ is dispersed across $n$ active storage nodes, each of which has a storage capacity of $\alpha$ code symbols. Suppose a newcomer receives the same amount of information from each of the existing nodes, any $k$ storage nodes can recover the original file. When a storage node failed, a replacing node collects $\beta$ symbols from each of any $d$ surviving nodes, and hence the total repair bandwidth is $\gamma = d \beta$.

Since the entire original file can be reconstructed from any $k$ storage nodes, any node can be recovered in the DSS. In this way, a total bandwidth of $k \alpha \geq M$ bits is required to repair any failed node with capacity of $\alpha$ symbols. According to the results in [4], by connecting to $d$ surviving nodes, less bandwidth of $\gamma = d \beta$ will be required to recover the failed node. There exists an optimal tradeoff between the storage overhead per node $\alpha$ and the repair bandwidth $\gamma$, and the codes that attain the optimal tradeoff above are

$$
\begin{align*}
\alpha & = \frac{M}{n} \\
\beta & = \frac{M}{(n-k)} \\
\gamma & = d \beta
\end{align*}
$$
called regenerating codes. The codes that attain the minimum storage overhead are termed minimum-storage regenerating (MSR) codes, with
\[
(\alpha_{\text{MSR}}, \gamma_{\text{MSR}}) = \left(\frac{M}{k}, \frac{Md}{k(d - k + 1)}\right).
\]
Similarly, the codes that achieve minimum-bandwidth repair are referred to as minimum-bandwidth regenerating (MBR) codes, with
\[
(\alpha_{\text{MBR}}, \gamma_{\text{MBR}}) = \left(\frac{2Md}{2kd - k^2 + k}, \frac{2Md}{2kd - k^2 + k}\right).
\]

The repair process above can either be functional repair or exact repair [6], [7]. For functional repair, the nodes may change over time. If a node \(v_i\) is lost and we get a new node \(v_i'\) in the repair process, then we may have \(v_i' \neq v_i\). On the other hand, exact repair means that new node \(v_i'\) always is the same as the old one \(v_i\). Exact repair can obviate additional communication overheads during the repair process, and also avoid retuning the reconstruction and repair algorithm. In this paper, exact repair is considered to maintain the local codes in systematic form during the repair operation.

B. MSR-LOCAL CODES AND MBR-LOCAL CODES

By combining the advantages of LRCs and regenerating codes, Kamath et al. proposed codes with local regeneration [15], which carry locality over a vector alphabet. The constituent local codes themselves are regenerating codes, such as MSR codes or MBR codes. For example, the construction of MSR-local codes is shown in Fig.1.

The parent MSR code \(m'_1, m'_2P_1, m'_3P_2\) is ((\(n_L + \Delta, r, d\), (\(\alpha, \beta\))) MSR code with \(n_L = r + \delta - 1\) and \(d \leq r + \delta - 2\), and its generator matrix is \(G_0 = [G_L | Q_{\Delta}]\). Puncturing the parent MSR code to the first \(n_L\) symbols, we can obtain an ((\(n_L, r, d\), (\(\alpha, \beta\))) MSR code \(m'_1, m'_P1\). The MSR-local codes \(C_{\text{MSR-local}}\) consist of MSR code \(m'_1, m'_P1\) as local code \(C_i\) (\(1 \leq i \leq m\)), and \((\sum_{i=1}^{m} m_i)P_2\) as global parities. The corresponding generator matrix \(G_{\text{MSR-local}}\) is given by
\[
G_{\text{MSR-local}} = \begin{bmatrix}
G_L & Q_{\Delta}
\end{bmatrix}.
\]

Similarly, the MBR-local codes can be realized by the same method as MSR-local codes. The MBR-local codes \(C_{\text{MBR-local}}\) proposed in [15] also compose of \(m\) support disjoint MBR codes and \(\Delta\) global parities. Specially, each MBR code in MBR-local codes is a repair-by-transfer (RBT) ((\(n_L, r, d\), (\(\alpha, \beta\)), \(K_L\)) MBR code [18]. Thus the desired MBR-local codes \(C_{\text{MBR-local}}\) will have length \(n = mn_L + \Delta\), and dimension \(K = mK_L\).

III. CONSTRUCTION OF LOCAL CODES WITH COOPERATIVE REPAIR

For MSR-local and MBR-local codes, it is required to collect all the remaining local codes and the global parities to repair one failed local code, with high repair locality. RAID 4 consists of block-level striping with a dedicated parity disk, whereas RAID 5 with distributed parity. Unlike in RAID 4, RAID 5’s parity information is distributed among the drives, which evens out the stress of a dedicated parity disk among all RAID members. Inspired by the distributed parity of RAID 5, the global parities of MSR-local codes can also be broken down into multiple distributed parities to avoid the limitations of repairing the failed local codes. Thus, we construct local codes with cooperative repair (LCCR), in which each local code includes an MSR code and a distributed parity.

Construction 1: Let \(C_i'\) (\(1 \leq i \leq m\)) be an ((\(n_L + 2\Delta, r, d\), (\(\alpha, \beta\))) MSR code with generator matrix \(G_i' = [I|P_i|P_{i+2}|P_{i+2}|\cdots|P_{i+2}]\) and \(n_L = r + \delta - 1\) such that \(d \leq r + \delta - 2\), \(C_i''\) (\(1 \leq i \leq m\)) be the punctured code of \(C_i'\), i.e., an ((\(n_L, r, d\), (\(\alpha, \beta\)), \(K_L = r\alpha\)) exact repair MSR code with generator matrix \(G_i'' = [I|P_i]\). Based on the MSR code above, an \([n, K, d_{\text{min}}, \alpha]\) LCCR \(C\) is constructed with \(n = m(n_L + \Delta)\) and \(K = mK_L\), in which the parameters \(\delta, \Delta\) are chosen such that \(\Delta = \delta - 1\). LCCR \(C\) has \(m\) local codes \(C_1, C_2, C_3, \ldots, C_m\), each of which includes an MSR code and a distributed parity. To be specific, \(C_i = [m_i, m'_1, m'_2P_1, m'_3P_2 + m_i+1P_{i+1}](1 < i < m)\), which can be generated by the submatrix
\[
G_i = \begin{bmatrix}
0 & 0 & \cdots & 0 & P_{i+1} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & P_{i+1} & \ddots & \vdots & \vdots \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0
\end{bmatrix}.
\]

Thus generator matrix \(G\) of the desired LCCR \(C\) is
\[
G = \begin{bmatrix}
G_1' & 0 & 0 & P_3 & 0 & \cdots & 0 & P_{m-1} \\
0 & P_m G_2' & 0 & \cdots & 0 & 0 \\
0 & 0 & P_1 G_3' & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & 0 & P_2 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & P_1 \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & P_2 \\
\end{bmatrix}.
\]

And the local code \(C_1\) (or \(C_m\)) can be generated by the submatrix composed of the first two columns of generator
matrix $G$ (or the last two columns). According to the matrix construction above, the distributed parity symbols of each local code can be created by the MSR codes in its two adjacent local codes, which can be regarded as a kind of mutual interleaving structure among the local codes.

The code construction of LCCR $C$ is illustrated in Fig. 2. Punctured code $C_i^p$ $(1 \leq i \leq m)$ is also an MSR code, achieved by deleting the parities $m_iP_{i+2}$ and $m_iP_{i-2}$ of parent MSR code $C_i^p$ [15]. Based on the punctured MSR code $C_i^p$, the constructed LCCR $C$ contains $m$ local codes $C_1, C_2, C_3, \ldots, C_m$. Concretely, a local code $C_i$ includes an MSR code $m_iP_i$ and a distributed parity $m_{i-1}P_{i+1} + m_{i+1}P_{i-1}$, which can be formed by the MSR code $m_{i-1}P_{i-1}$ and $m_{i+1}P_{i+1}$ in its two adjacent local codes. A local code $C_i$ has two adjacent local codes $C_{i-2m+1}$ and $C_{i+2m+1}$, where $(i)_m \equiv i \mod m$.

The theorem below identifies the parameters of LCCR so constructed and derives the minimum distance $d_{\text{min}}$ of LCCR.

**Theorem 1:** Consider LCCR $C$ constructed in Construction 1 in which the parameters $\delta$ and $\Delta$ are chosen such that $\Delta = \delta - 1$. Then LCCR $C$ with $(r, \delta)$ information locality has length $n = m(n_L + \Delta)$ and dimension $K = mK_L$. The minimum distance of $C$ is given by

$$d_{\text{min}} = n - \frac{K}{\alpha} + \delta - 2 \left( \frac{K}{r\alpha} - 1 \right) (\delta - 1) = \delta + 2\Delta.$$  

**Proof:** Towards LCCR $C$ in Fig. 2, it suffices to show that any non-zero codeword $C$ has Hamming weight $w(C) \geq \delta + 2\Delta$. First of all, note that if $C$ has non-zero components belonging to one or more local codes, then $w(C) \geq \delta + 2\Delta$, since all MSR codes in local codes have minimum distance $\delta$ and all distributed parities have minimum distance $\Delta$. Next, consider the complementary case where the non-zero components of $C$ are restricted to one MSR code and two relevant distributed parities of local codes, such as the MSR code $m_1P_1$ and two distributed parities $m_3P_3$ in local code $C_2$ and $m_1P_{m-1}$ in $C_m$. And when the all-zero code symbols are deleted from each codeword, then the resultant punctured codeword $C_p$ lies in the space of the resulting matrix

$$G_p = \begin{bmatrix} G_1 & P_3 & P_{m-1} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

which is reduced from the generator matrix $G$. As the information vector remains unchanged and still is $[m_1 m_2 m_3 \cdots m_m]$, the punctured codeword $C_p$ is $[m_1G_1 m_1P_3 m_1P_{m-1}]$, that is the parent MSR code $C_i^p$ with Hamming weight $w(C_p) = \delta + 2\Delta$. By the analysis above, the minimum distance $d_{\text{min}}$ of LCCR is $\delta + 2\Delta$.

Since $n = m(r + \delta - 1 + \Delta), K = m\alpha r$ and $\Delta = \delta - 1$, $d_{\text{min}}$ can be expressed as

$$d_{\text{min}} = n - \frac{K}{\alpha} + \delta - 2 \left( \frac{K}{r\alpha} - 1 \right) (\delta - 1) = \delta + 2\Delta.$$  

**IV. PERFORMANCE COMPARISON**

In this section, we choose a representative set of codes, including LCCR, MSR-local codes and MBR-local codes, to analyze their performances in storage overhead, repair bandwidth overhead, and repair locality. Since the bandwidth overhead and the repair locality depend on the failures of local groups or nodes, the cases of a single node failure, one local group failure, and multiple local group failures are discussed separately.

**A. STORAGE OVERHEAD**

In this paper, we adopt the definition of storage overhead $n\alpha/K$ given in [15], which is defined to be the ratio of the total number of storage units required to store the encoded symbols over the number of message symbols contained in the original file. The storage overheads of LCCR, MSR-local codes and MBR-local codes are given by

$$S_{\text{LCCR}} = \frac{n\alpha}{K} = \frac{nL + \Delta}{r},$$

$$S_{\text{MSR-local}} = \frac{n\alpha}{K} = \frac{nL + \Delta}{m},$$

$$S_{\text{MBR-local}} = \frac{n\alpha}{K} = \frac{2(nL + \Delta/m)(r + \delta - 2)}{(r + 2\delta - 3)}.$$  

As MSR-local codes and MBR-local codes have the similar code structure, the parameter $m$ can be achieved with the same value for MSR-local codes and MBR-local codes as $n$ and $d_{\text{min}}$ fixed. Similarly, the parameters $r, \delta, \Delta$ in MSR-local codes and MBR-local codes also have the same values. Thus, we can compare the storage overhead of MSR-local codes with that of MBR-local codes directly. For MBR-local codes, its storage overhead is given by

$$S_{\text{MBR-local}} = \frac{2(r + \delta - 2)}{(r + 2\delta - 3)} \cdot S_{\text{MSR-local}} = \eta \cdot S_{\text{MBR-local}},$$

where $\eta$ is the storage overhead ratio of MBR-local codes to MSR-local codes.
Since in MBR-local codes, dimension

\[ K = m \cdot \left( r\alpha - \left( \frac{r}{2} \right) \right) \geq 1, \]

and thus \( r \geq 2 \). Meanwhile with \( \delta \geq 3 \), it can be deduced that

\[ \eta = \frac{2(r + \delta - 2)}{(r + 2\delta - 3)} > 1. \]

The storage overhead of MBR-local codes is lower bounded by that of MSR-local codes, and the factor \( \eta \) reflects the extra storage overhead of MBR-local codes compared with MSR-local codes.

With the same code length \( n \) and minimum distance \( d_{\text{min}} \), the values of the parameters \( m, r, \delta, \Delta \) obtained in LCCR will differ from those of MSR-local codes and MBR-local codes, since its code length \( n = m(n_L + \Delta) \) and its minimum distance \( d_{\text{min}} = \delta + 2\Delta \). Thus, we cannot compare the storage overhead of LCCR with that of MSR-local codes, or that of MBR-local codes directly according the formulae defined by \( n\alpha / K \). In the next subsection, the comparison about the storage overhead of LCCR, MSR-local codes, or MBR-local codes will be plotted by numerical analysis as \( n = 120 \) and \( d_{\text{min}} = 16 \).

B. REPAIR OF A SINGLE FAILED NODE

To maintain the completeness of codes, even though the failed nodes are in the parity part, they all need to be repaired, either the global parities of MSR-local codes or MBR-local codes, or the distributed parity of LCCR. We only discuss the case of one single node failure, not including the case of multiple node failures, which can be regarded as a generalization of the case of one node failure.

Although the proposed LCCR repairs the failed local codes as a whole, LCCR, which is constructed based on MSR codes, has the ability to repair the failed nodes in local groups. When the failed nodes of one local group are in the MSR code part, we can adopt the same method as MSR codes to repair. If there are several failed nodes in the distributed parity of one local code, we can repair the failed nodes by collecting the MSR codes in its two adjacent local codes. Specifically for LCCR, take as an example that there exists a single failed node in the distributed parity \( m_{i+1}P_{i+1} + m_iP_{i-1} \) of local code \( C_i \) as shown in Fig. 3. To repair the single failed node, local group \( i \) should collect symbols \( m_{i-1}P_{i-1} \) and \( m_{i+1}P_{i+1} \) from local group \( i - 1 \), and symbols \( m_{i+1}P_{i+1} \) and \( m_{i+1}P_{i+1} \) from local group \( i + 1 \). Furthermore by simple matrix operations, the distributed parity \( m_{i+1}P_{i+1} + m_{i+1}P_{i+1} \) can be obtained, and meanwhile the single failed node in distributed parity \( m_{i+1}P_{i+1} + m_{i+1}P_{i+1} \) can be repaired.

Next, we discuss the performance of repair locality for LCCR, MSR-local codes and MBR-local codes in the case of a single node failure. Due to the construction of LCCR that the failed nodes may be located in the MSR code of local codes, or in the distributed parity, we should consider the two cases together.

\[ \text{FIGURE 3. Repair of a single failed node located in the distributed parity of LCCR.} \]

**Lemma 1**: The locality to repair one single failed node located in the MSR code of LCCR is \( n_L - 1 \).

**Lemma 2**: The locality to repair one single failed node located in the distributed parity of LCCR is \( 2n_L \).

**Theorem 2**: The locality of LCCR to repair one single failed node is

\[ \frac{(n_L - 1) \cdot n_L + 2n_L \cdot \Delta}{n_L + \Delta}. \]

Since the local codes of MSR-local codes are MSR codes, the repair locality of MSR-local codes is the same as that of MSR codes when repairing one single failed node in local codes, which equals \( n_L - 1 = r + \delta - 2 \). Meanwhile, if the single failed node is located in the global parities, the global parities of MSR-local codes should be repaired as a whole, which needs \( mr \) nodes in total. The repair locality of MSR-local codes is

\[ \frac{(n_L - 1) \cdot n_L + mr \cdot \Delta}{mn_L + \Delta} = \frac{(n_L - 1) \cdot n_L + r \cdot \Delta}{n_L + \Delta / m}. \]

Adopting the same method, we can obtain the repair locality of MBR-local codes which equals that of MSR-local codes, as the formula describes above.

Using numerical analysis, we can compare the locality of LCCR with that of MSR-local codes, or that of MBR-local codes. Concretely, we choose common length \( n = 120 \) and common minimum distance \( d_{\text{min}} = 16 \). For comparison conveniently, we set the parameter \( m \geq 3 \) since there is at least \( m \geq 3 \) local codes in LCCR. Fig. 4 illustrates the repair locality versus storage overhead for LCCR, MSR-local codes and MBR-local codes. From Fig. 4, MBR-local codes have the same repair locality as that of MSR-local codes as they have the same values for parameter \( m, r, \delta, \Delta \), which is consistent with the theoretical analysis. Through numerical analysis, LCCR has the smallest repair locality under the same storage consumption.

Furthermore, we analyze the bandwidth overhead to repair one single failed node. The normalized bandwidth overhead can be calculated by \( n\omega / K [15] \), where \( \omega \) denotes the average repair bandwidth for repairing a single failed node.

**Theorem 3**: The bandwidth overhead of LCCR to repair one single failed node is

\[ B_{LCCR} = \frac{n \cdot n_{\omega, LCCR}}{K} = \frac{n^2(n_L - 1)}{r^2(\delta - 1)} + \frac{2n_L(\delta - 1)}{r}. \]
Repair locality versus storage overhead for one single node failure.

Proof: According to the repair process of a single failed node in LCCR, the failed node may be located in the MSR code of local codes, or in the distributed parity. We can calculate $\omega_{LCCR}$ by

$$\omega_{LCCR} = \frac{\omega_1 \cdot n_L + \omega_2 \cdot \Delta}{n_L + \Delta},$$

where $\omega_1 = \frac{n_L \cdot \alpha}{r \cdot m} \cdot \frac{n_L - 1}{n_L - r}$ denotes the average repair bandwidth for repairing a single failed node located in the MSR code of local codes, and $\omega_2 = 2n_L \cdot \alpha$ denotes the repair bandwidth required for repairing one failed node located in the distributed parity. The repair bandwidth needed is $2n_L \cdot \alpha$ whether there are one or more failed nodes in the distributed parity, since the distributed parity should be repaired as a whole at any time. According to the analysis above, the bandwidth overhead of LCCR is

$$B_{\text{LCCR}} = \frac{n \cdot \omega_{LCCR}}{K} = \frac{n_L^2(n_L - 1)}{r^2(\delta - 1)} + \frac{2n_L(\delta - 1)}{r}.$$

For MSR-local codes and MBR-local codes, all the information symbols of the $m$ local codes should be collected together to repair one single failed node located in the global parities, which will further increase the repair bandwidth. The repair bandwidth overhead of MSR-local codes and MBR-local codes can also be calculated by $n \cdot \frac{\omega}{K}$ [15].

Lemma 3: For MSR-local codes, the average repair bandwidth for repairing a single failed node located in local codes is $\omega_1' = \frac{n_L \cdot \alpha}{r \cdot m} \cdot \frac{n_L - 1}{n_L - r}$, and the average repair bandwidth for repairing a failed node located in global parities $\omega_2' = m \cdot \alpha L = m \cdot \alpha L$. The average repair bandwidth

$$\omega_{\text{MSR-local}} = \frac{\omega_1' \cdot m \cdot n_L + \omega_2' \cdot \Delta}{m \cdot n_L + \Delta},$$

The bandwidth overhead of MSR-local codes to repair one single failed node is

$$B_{\text{MSR-local}} = \frac{n \cdot \omega_{\text{MSR-local}}}{K} = \frac{n_L^2(n_L - 1)}{r^2(\delta - 1)} + \Delta.$$

Lemma 4: In MBR-local codes, each MBR code adopted is a repair-by-transfer (RBT) $(n_L, r, d, (\alpha, \beta), K_L)$ MBR code [18]. The average repair bandwidth for repairing a single failed node located in local codes is $\omega_1'' = n_L - 1$, and the average repair bandwidth for repairing a failed node located in global parities $\omega_2'' = m \cdot \alpha L = m \cdot \alpha L$. The bandwidth overhead of MBR-local codes for repairing one single failed node is

$$B_{\text{MBR-local}} = \frac{n \cdot \omega_{\text{MBR-local}}}{K} = \frac{2n_L(n_L - 1)}{r(r + 2\delta - 3)} + \Delta.$$

A summary of repair locality and bandwidth overhead for LCCR, MSR-local codes and MBR-local codes is presented in Table 1, as one single node failed. For the same reason that MSR-local codes and MBR-local codes have the similar code structure, we can compare their bandwidth overhead conveniently, obtaining that $B_{\text{MBR-local}} < B_{\text{MSR-local}}$. Similarly, we adopt numerical analysis to compare the bandwidth overhead of LCCR with that of MSR-local codes, or that of MBR-local codes. As $n = 120$, $d_{\text{min}} = 16$ and $m \geq 3$, the performance of LCCR, MSR-local codes and MBR-local codes in repair bandwidth versus storage overhead is shown in Fig. 5.

Since MSR-local codes and MBR-local codes achieve the same values for parameter $m, r, \delta, \Delta$ while fixing $n = 120$ and $d_{\text{min}} = 16$, we will obtain the same code number for MSR-local codes and MBR-local codes. Especially in Fig. 5,
TABLE 2. Summary of repair locality and bandwidth overhead for the failed node not located in the parity part.

| Construction | Repair locality | Bandwidth overhead |
|--------------|-----------------|-------------------|
| LCCR         | $n_i - 1$       | $\frac{n_i^2(n_i - 1)}{r^2(\delta - 1)}$ |
| MSR-local codes | $n_i - 1$       | $\frac{n_i^2(n_i - 1)}{r^2(\delta - 1)}$ |
| MBR-local codes | $n_i - 1$       | $\frac{2n_i(n_i - 1)}{r(r + 2\delta - 3)}$ |

MSR-local codes and MBR-local codes have similar distribution, furthermore which are mutually corresponding. MBR-local codes and MSR-local codes will be separately located in the corresponding position of their distribution region of the plot, when they have the same value for parameter $m$, $r$, $\delta$, $\Delta$. The storage overhead of MBR-local codes is larger than that of MSR-local codes, and meanwhile its repair bandwidth is smaller than that of MSR-local codes, in accordance with the theoretical analysis. From Fig. 5, the bandwidth overhead of LCCR is larger than that of MBR-local codes and MSR-local codes on the whole, which results from the fact that there is a distributed parity in each local code of LCCR. The average bandwidth overhead to repair a failed node located in the distributed parities of LCCR is larger than that located in the global parities of MSR-local codes and MBR-local codes.

In addition, we consider repairing the failed node located only in the MSR code part of LCCR, or in local codes of MSR-local codes or MBR-local codes. As a summary presented in Table 2, the corresponding repair locality and bandwidth overhead are illustrated in Fig. 6 and Fig. 7 for the case that the failed node is not located in the parity part, i.e., the global parities of MSR-local codes or MBR-local codes, or the distributed parity of LCCR. From Fig. 6, LCCR also has the smallest repair locality compared with MSR-local codes and MBR-local codes. The bandwidth overheads of LCCR, MSR-local codes and MBR-local codes in Fig. 7 are smaller than that in Fig. 5, which coincides with the theoretical analysis above. Moreover in Fig. 7, LCCR have almost the same bandwidth and storage overheads as MSR-local codes, except the point with higher storage overhead.

C. REPAIR OF ONE FAILED LOCAL GROUP

Since the minimum distance $d_{\text{min,local}}$ of the MSR code in one local code of LCCR is $\delta$, the failed nodes cannot be repaired by other available nodes of the same local group as the number of failed nodes in one local code exceeds $\delta - 1$. Thus some local groups, in which there exist more than $\delta - 1$ failed nodes, are considered as failed local groups. For LCCR, one failed local group can be repaired cooperatively by its four adjacent local groups. Fig. 8 illustrates the concrete repair procedure for one failed local group corresponding to local code $C_i$ ($1 \leq i \leq m$). In Fig. 8, the indices of the adjacent groups are shown without the cyclic shift for simplicity. Local code $C_i$ can be recovered by local codes $C_{i-2}$, $C_{i-1}$, $C_{i+1}$ and $C_{i+2}$. Local group $i - 1$ collects data $m_{i-2}$ and $m_{i-2}P_{i-2}$ of local code $C_{i-2}$ in local group $i - 2$, and meanwhile local group $i + 1$ collects data $m_{i+2}$ and $m_{i+2}P_{i+2}$ of local code $C_{i+2}$ in local group $i + 2$. By collecting $m_{i-2}P_{i} + m_{i}P_{i-2}$ of local code $C_{i-1}$ and $m_{i}P_{i+2} + m_{i+2}P_{i}$ of local code $C_{i+1}$, local group $i$ can recover data $m_{i}P_{i}$ by matrix operations, and the information data $m_{i}$ by using the generator matrix $G'_{i} = [I|P_{i}]$ of the punctured MSR code $C'_{i}$ as $\delta - 1 \geq r$. Furthermore, collecting data $m_{i-1}$ and $m_{i-1}P_{i-1}$ from local group $i - 1$, and $m_{i+1}$ and $m_{i+1}P_{i+1}$ from local group $i + 1$, local group $i$ gains the distributed parity $m_{i-1}P_{i+1} + m_{i+1}P_{i-1}$ of local code $C_i$ also by simple matrix operations. Until now, we achieve the MSR code part and the distributed parity part of local code $C_i$, and complete the repair of local group $i$.

**Definition 1 (Locality for Repairing Failed Local Groups):** Similar to repairing failed nodes, the number of local groups that participate in the repair of failed local groups is defined as the repair locality for repairing failed local groups. Especially for MSR-local codes and MBR-local codes, when the global
parities are involved in the repair of failed local groups, the global parities should be regarded as one whole local group during repairing.

In Fig. 8, any one failed local group of LCCR can be recovered by four local groups, and in this case the repair locality of LCCR to repair one failed local group is 4. Similarly, we also choose $n = 120$ and $d_{\text{min}} = 16$. Considering in LCCR, one single failed local group need 4 adjacent local groups to repair, we set $m \geq 5$. For MSR-local codes and MBR-local codes, it is essential to collect all the surviving local codes and the global parities to repair the only one failed local group. Since MSR-local codes have $m$ local codes and one global parities part, the repair locality of MSR-local codes is $m$. For the same reason, the repair locality of MBR-local codes is also $m$. Consequently, MSR-local codes and MBR-local codes have the same repair locality of $m$, as shown in Fig. 9. From Fig. 9, the repair locality of LCCR is smaller than that of MSR-local codes and MBR-local codes, not limited by the storage overhead.

Similarly, the bandwidth overhead can be calculated by $(m\omega + \omega_{\Delta})/m\alpha$, where $\omega'$ denotes the average repair bandwidth for repairing a single failed local group, and $\omega_{\Delta}$ the repair bandwidth required for repairing the global parities (specially, $\omega_{\Delta} = 0$ for LCCR).

**Lemma 5:** In LCCR, $\omega'_{\text{LCCR}} = 4nL\alpha + 3(\delta - 1)\alpha$. Then the bandwidth overhead of LCCR is calculated by

$$B'_{\text{LCCR}} = \frac{m \cdot \omega'_{\text{LCCR}}}{m\alpha} = 4nL + 3(\delta - 1).$$

**Lemma 6:** In MSR-local codes, $\omega'_{\text{MSR-local}} = mr\alpha + \Delta\alpha$ and $\omega'_{\Delta} = mr\alpha$. The bandwidth overhead of MSR-local codes is given by

$$B'_{\text{MSR-local}} = \frac{m \cdot \omega'_{\text{MSR-local}} + \omega'_{\Delta}}{m\alpha} = \Delta + mr + r.$$
Three failed local groups cannot be repaired when they are adjacent, or when any two of them are either adjacent or separated by other one local group. Since up to 12 and at least 8 local groups are needed to repair the failed local groups, the locality per failed local group is up to 4 and at least $8/3$ on average. Considering the probability that four local groups failed simultaneously is very little, we can omit this case in this paper.

**Theorem 4 (Upper Bound):** In LCCR, if $\delta + 2\Delta - 1 \leq \lceil m/3 \rceil$, the number of failed local groups, that can be repaired cooperatively by other available local groups, is upper bounded by $\delta + 2\Delta - 1$. If not, the number of repairable local groups is upper bounded by $\lceil m/3 \rceil$.

**Proof:** Since LCCR can be regarded as one linear code with code length $n = m(r + \delta + \Delta - 1)$ and minimum distance $d_{\text{min}} = \delta + 2\Delta$, at most $d_{\text{min}} - 1 = \delta + 2\Delta - 1$ erasures in LCCR can be recovered simultaneously when $\delta + 2\Delta - 1 \leq \lceil m/3 \rceil$. If we distribute the $\delta + 2\Delta - 1$ erasures into $\delta + 2\Delta - 1$ local groups, each local group will have one erasure.

Then the number of repairable local groups is upper bounded by $\delta + 2\Delta - 1$.

If $\lceil m/3 \rceil < \delta + 2\Delta - 1$, considering that two failed local groups can be repaired simultaneously only when the two local groups are spaced by no less than two other local groups, the maximum number of repairable local groups is $\lceil m/3 \rceil$.

Nevertheless, according to the structure of MSR-local codes and MBR-local codes proposed in [15], MSR-local codes and MBR-local codes have the ability to repair one failed local group, but incapable of repairing two or more failed local groups at the same time. Moreover, for MSR-local codes and MBR-local codes, it is essential to collect all the remaining local codes and the global parities for repairing one failed local group, which will increase the repair complexity and locality. Consequently, limitations exist for MSR-local codes and MBR-local codes to recover the failed local groups.

**V. CONCLUSION**

Involving a complex hyperspace of cyber, physical, and social spaces, CPSS have seen significant adoption in the past few years, which are secure by design and exemplify a distributed computing system with high fault tolerance. This paper studies the fault tolerance scheme for distributed storage systems of CPSS, and proposes a framework that can recover multiple failed nodes simultaneously. Concretely, we investigate that in some local groups there exist more than $\delta - 1$ failed nodes, and the corresponding local codes stored need to be recovered as a whole. Particularly, an explicit construction of LCCR based on MSR codes is proposed to repair the failed local groups with lower repair locality, in which the distributed parity symbols of each local code can be generated by the MSR codes in its two adjacent local codes, regarded as a kind of mutual interleaving structure among the local codes. Based on LCCR, the failed local groups can be repaired cooperatively by their adjacent local groups, and the key parameters of LCCR are derived.
The performances of LCCR, MSR-local codes and MBR-local codes in storage overhead, repair bandwidth overhead and repair locality are discussed respectively. Theoretical and numerical analyses show that, compared with MSR-local codes and MBR-local codes, LCCR has benefits in storage overhead and repair locality, and achieve almost the same tradeoff curve of bandwidth overhead and storage overhead as MSR-local codes and MBR-local codes for a single failed local group. Moreover, LCCR can repair multiple failed local groups simultaneously, except the case that two failed local groups are adjacent to each other, or spaced only by one local group.

REFERENCES

[1] W. Liang, M. Tang, J. Long, X. Peng, J. Xu, and K. Li, “A secure fabric blockchain-based data transmission technique for industrial Internet-of-Things,” IEEE Trans. Ind. Informat., vol. 15, no. 6, pp. 3582–3592, Jun. 2019.

[2] W. Liang, J. Long, T.-H. Weng, X. Chen, K.-C. Li, and A. Y. Zomaya, “TBRS: A trust based recommendation scheme for vehicular CPS network,” Future Gener. Comput. Syst., vol. 92, pp. 383–398, Mar. 2019.

[3] W. Liang, Y. Fan, K. Li, D. Zhang, and J. Gaudiot, “Secure data storage and recovery in industrial blockchain network environments,” IEEE Trans. Ind. Informat., to be published, doi: 10.1109/TII.2020.2966069.

[4] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, “Network coding for distributed storage systems,” IEEE Trans. Inf. Theory, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.

[5] K. V. Rashmi, N. B. Shah, and P. V. Kumar, “Optimal exact-repairing codes for distributed storage at the MSR and MBR points via a product-matrix construction,” IEEE Trans. Inf. Theory, vol. 57, no. 8, pp. 5227–5239, Aug. 2011.

[6] T. Ernvall, “Exact-repairing codes between MBR and MSR points,” in Proc. IEEE Inf. Theory Workshop (IFW), Sep. 2013, pp. 1–5.

[7] T. Ernvall, “Codes between MBR and MSR points with exact repair property,” IEEE Trans. Inf. Theory, vol. 60, no. 11, pp. 6993–7005, Nov. 2014.

[8] F. Oggier and A. Datta, “Self-repairing homomorphic codes for distributed storage systems,” in Proc. IEEE INFOCOM, Shanghai, China, Apr. 2011, pp. 1215–1223.

[9] O. Khan, R. Burns, J. Park, and C. Huang, “In search of DO-optimal recovery from disk failures,” in Proc. 3rd USENIX Conf. Hot Topics Storage File Syst. (HotStorage), Portland, OR, USA, Jun. 2011, pp. 1–5.

[10] P. Gopalan, C. Huang, H. Simon, and S. Yekhanin, “On the locality of codeword symbols,” IEEE Trans. Inf. Theory, vol. 58, no. 11, pp. 6925–6934, Nov. 2012.

[11] D. S. Papailiopoulos, J. Luo, A. G. Dimakis, C. Huang, and J. Li, “Simple regenerating codes: Network coding for cloud storage,” in Proc. IEEE INFOCOM, Orlando, FL, USA, Mar. 2012, pp. 2801–2805.

[12] D. S. Papailiopoulos and A. G. Dimakis, “Locally repairable codes,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Cambridge, MA, USA, Jul. 2012, pp. 2771–2775.

[13] A. S. Rawat, N. Silberstein, O. O. Koyluoglu, and S. Vishwanath, “Optimal locally repairable codes with local minimum storage regeneration via rank-metric codes,” in Proc. Inf. Theory Appl. Workshop (ITA), San Diego, CA, USA, Feb. 2013, pp. 1–8.

[14] A. Wachter-Zeh, V. Afanasiev, and V. Sidorenko, “Fast decoding of gabidulin codes,” Des., Codes Cryptogr., vol. 66, nos. 1–3, pp. 57–73, Apr. 2012.

[15] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar, “Codes with local regeneration and erasure correction,” IEEE Trans. Inf. Theory, vol. 60, no. 8, pp. 4637–4660, Aug. 2014.

[16] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, “Optimal locally repairable and secure codes for distributed storage systems,” IEEE Trans. Inf. Theory, vol. 60, no. 1, pp. 212–236, Jan. 2014.

[17] S. W. Ng and R. L. Matsson, “Maintaining good performance in disk arrays during failure via uniform parity group distribution,” in Proc. 1st Int. Symp. High-Perform. Distrib. Comput., Syracuse, NY, USA, Sep. 1992, pp. 260–269.

[18] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, “Distributed storage codes with Repair-by-Transfer and nonachievability of interior points on the storage-bandwidth tradeoff,” IEEE Trans. Inf. Theory, vol. 58, no. 3, pp. 1837–1852, Mar. 2012.

JING WANG received the B.E. degree in communication engineering from the School of Communication Engineering, Xidian University, Xi’an, China, in 2004, and the Ph.D. degree in communication and information systems from the State Key Laboratory of Integrated Service Networks, Xidian University, in 2009. She is currently a Professor with the School of Information Engineering, Chang’an University, Xi’an. Her research interests include network coding, distributed storage, and regenerating codes.

ZHIIYUAN YAN (Senior Member, IEEE) received the B.E. degree in electronics engineering from Tsinghua University, Beijing, China, in 1995, and the M.S. degree in electrical engineering, the M.S. degree in mathematics, and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 1999 and 2003, respectively.

He was a Research Intern with the Nokia Research Center, Irving, TX, USA, in 2000 and 2002. He joined the Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA, USA, in 2003, where he is currently an Associate Professor. In 2009 and 2012, he was a Visiting Faculty Fellow with the Air Force Research Laboratory, Rome, NY, USA. He has authored over 100 refereed articles in journals and conferences. His current research interests include coding theory, signal processing, cryptography, wireless communications, and VLSI implementations of communication and signal processing systems.

KUAN-CHING LI (Senior Member, IEEE) is currently a Distinguished Professor with the Department of Computer Science and Information Engineering, Providence University, Taiwan. He has been actively involved in several major conferences and workshops in program/general/steering conference chairmen positions and has organized numerous conferences on high-performance computing and computational science and engineering. Besides publishing numerous research papers and articles, he is the coauthor/co-editor of several technical professional books published by CRC Press/Taylor & Francis, Springer, McGraw-Hill, and IGI Global. His research interests include parallel and distributed processing, GPU many-core computing, and big data and cloud.

He is also a member of the AAAS, a Life Member of the TACC, and a Fellow of the IET. He received distinguished and chair professorships from universities in China and other countries, and a recipient of awards and funding support from several agencies and high-tech companies.
HONGMEI XIE received the B.E. degree in electrical engineering from Qingdao University, Qingdao, China, in 2005, the M.E. degree in electrical engineering from Xidian University, Xi’an, China, in 2008, and the Ph.D. degree in electrical engineering from Lehigh University, Bethlehem, PA, USA, in 2015.

She currently joins Seagate Technology, Fremont, CA, USA. Her research interests are in coding theory and its application to communication systems.

XIANGYANG LIU received the B.S. degree from the Xi’an Communications Institute, in 2004, and the Ph.D. degree from Xidian University, in 2010. He is currently an Associate Professor with the National University of Defense Technology. His research interests include distributed storage and signal detection.

* * *