AZ-code API 使用说明  
Java, 基于hadoop EC模块

**Code Configure**

类名：CoderOperation

作为定义EC的参数，encoder和decoder类初始化时的输入参数

需要的参数：

1. numDataUnits: k, 数据盘数量；
2. numParityUnits: r, 校验盘数量；
3. usePCM: 是否使用PCM/APCM方法。

**RS Code**

一、Encoder

类名：RSRawEncoder(CoderOperation), 基类: RawErasureEncoder

1. encode(byte[][] inputs, byte[][] outputs)

inputs: data 数据，k\*block size；

outputs: parity 数据，r\*block size，空数组；

block size 为块大小。

二、Decoder

类名：RSRawDecoder(CoderOperation), 基类: RawErasureDecoder

1. decode(byte[][] inputs, int[] erasedIndexes, byte[][] outputs)

inputs: 可用数据数据，k\*block size；

outputs: 返回需要回复的数据，error\_num\*block size，空数组，error\_num为失效盘的数量；

erasedIndexes: 长度为1~r的数列，记录失效盘的编号（必须为从小到大排列）。

\*\*\*RS代码基本与Hadoop EC模块一样，它和Intel ISA-L的代码逻辑也是基本相同的。

**MSR Code**

一、Encoder

类名: MSRRawEncoder(CoderOperation), 基类: RawErasureEncoder

1. encode(byte[][] inputs, byte[][] outputs)

inputs: data数据，(k\*l)\*block size，一共k个盘，第i个盘内的l个数据顺序放在inputs[i\*l] ~ inputs[(i+1)\*l-1]内；

outputs: parity数据，(r\*l)\*block size，同上；

block size 为块大小。

1. genMSRMatrix(byte[] matrix, int n, int k)

matrix: 空矩阵，将会返回一个(r\*l)\*(n\*l)的MSR矩阵。

1. genMSREncodeMatrix(byte[] MSRmatrix, byte[] MSREncodeMatrix, int n, int k)

MSRmatrix: 由genMSRMatrix函数得到的MSR矩阵；

MSREncodeMatrix: 空数组，将会返回一个(n\*l)\*(k\*l)的编码矩阵，可以直接用来编码。

二、Decoder

类名：RSRawDecoder(CoderOperation), 基类: RawErasureDecoder

1. static method searchData(int[] erasedIndexes, int n, int k)

由于MSR单盘恢复所需数据较少，这个函数专门用来判断最少需要哪些数据来恢复失效数据；

erasedIndexes: 长度为1~r的数列，记录失效盘的编号（必须为从小到大排列）；

n: 总盘数；

k: 数据盘数；

返回：int[][]array 一个n\*l的二维数组，分别代表n个盘和每个盘中的l个数据块。当array[i][j]为1时，第i个盘的第j个数据块需要被使用。

1. decode(byte[][] inputs, int[] erasedIndexes, byte[][] outputs)

inputs: 所需的恢复数据，大小和内容由searchData函数决定；

outputs: 需要恢复的数据，(error\_num\*l)\*block size；

erasedIndexes: 长度为1~r的数列，记录失效盘的编号（必须为从小到大排列）。

1. mulPrepareAPCMDecoding(byte[][] inputs, int[] erasedIndexes)

当失效的盘数大于等于2时，用于具体解码操作的函数，同类型的还有mulPrepareDecoding，区别在于使用/不使用APCM方法加速；

inputs: 所需的恢复数据，大小和内容由searchData函数决定；

erasedIndexes: 长度为1~r的数列，记录失效盘的编号（必须为从小到大排列）。

1. singlePrepareDecoding(byte[][] inputs, int[] erasedIndexes)

当失效的盘数等于1时，用于具体解码操作的函数，这个函数在JAVA API里不支持APCM方法。

\*\*\*具体实例可以看testCase、hadoopEC\_MSR和hadoopEC\_RS

C++, 基于ISA-L库

这里RS与原版的ISA-L一样

**MSR Code**MSR code基本结构与JAVA版的完全一样，但是它的single node-failure decoder 用了APCM方法加速

Encode:

1. gf\_gen\_MSR\_matrix(unsigned char\*matrix, int n, int k)

matrix：空矩阵，函数返回一个(r\*l)\*(n\*l)的MSR矩阵

1. gf\_gen\_MSR\_encode\_matrix(unsigned char \*enocdeMatrix, unsigned char \*MSRmatrix, int n, int k)

MSRmatrix: 由genMSRMatrix函数得到的MSR矩阵；

enocdeMatrix: 空数组，将会返回一个(n\*l)\*(k\*l)的编码矩阵，可以直接用来编码。

Decode:

1. mulPrepareDecoding/mulPrepareAPCMDecoding/singlePrepareAPCMDecoding 分别负责多盘失效/单盘失效时的恢复，其中singlePrepare支持APCM方法；
2. searchData(int\* erasedIndexes, int n, int k, int l, int errorLen)

用于判断最少需要哪些数据来恢复失效数据；

erasedIndexes: 长度为1~r的数列，记录失效盘的编号（必须为从小到大排列）；

n: 总盘数；

k: 数据盘数；

l: msr 参数；

errorLen: 失效盘个数；

返回：int[][]array 一个n\*l的二维数组，分别代表n个盘和每个盘中的l个数据块。当array[i][j]为1时，第i个盘的第j个数据块需要被使用。