Moulin Coding

A Multilinear-Algebraic Solution for Cloud Storage Services

Hsin-Po WANG
(with Iwan Duursma and Xiao Li)

Department of Mathematics, University of Illinois Urbana-Champaign

2021-8-19 SIAM AG21

Outline

Part I: Motivation from cloud storage services

Part II: Review multilinear algebra (Brief! Just one page!)

Part III: Actual construction of *Moulin Codes*

Outline

Part I: Motivation from cloud storage services

Part II: Review multilinear algebra (Brief! Just one page!)

Part III: Actual construction of *Moulin Codes*

Outline

Part I: Motivation from cloud storage services

Part II: Review multilinear algebra (Brief! Just one page!)

Part III: Actual construction of *Moulin Codes*

Outline

Part I: Motivation from cloud storage services ⟸

Part II: Review multilinear algebra (Brief! Just one page!)

Part III: Actual construction of *Moulin Codes*

Motivation from cloud storage services

hard disk #2 | hard disk #4 | hard disk #6

hard disk #1 | hard disk #3 | hard disk #5 | hard disk #7

a large file

A cloud storage service is a collection of hard disks that help you store big files.

Initially, you upload a big file to the cloud. Each hard disk will store part of your file.

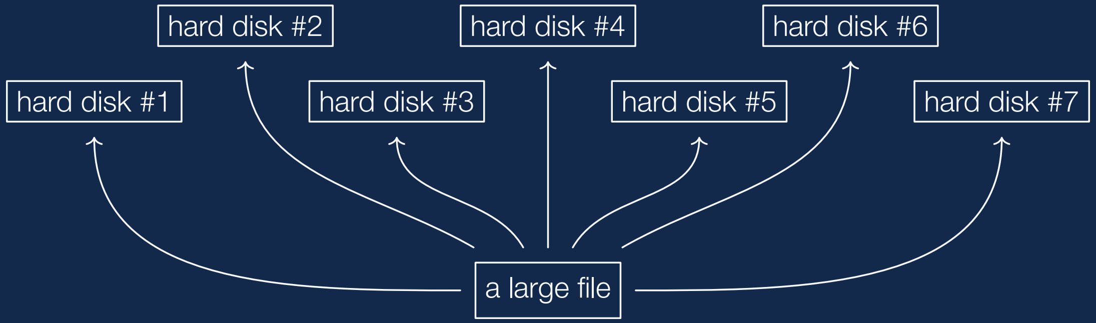
You then delete the local copy to free up some space.

Later, when you need the file, you download the file from the cloud.

The cloud company guarantees that the downloaded file is exactly the same file as before.

From your point of view, things work smoothly. But the cloud sees it differently.

Motivation from cloud storage services



A cloud storage service is a collection of hard disks that help you store big files.

Initially, you upload a big file to the cloud. Each hard disk will store part of your file.

You then delete the local copy to free up some space.

Later, when you need the file, you download the file from the cloud.
The cloud company guarantees that the downloaded file is exactly the same file as before.

From your point of view, things work smoothly. But the cloud sees it differently.

Motivation from cloud storage services

A cloud storage service is a collection of hard disks that help you store big files.

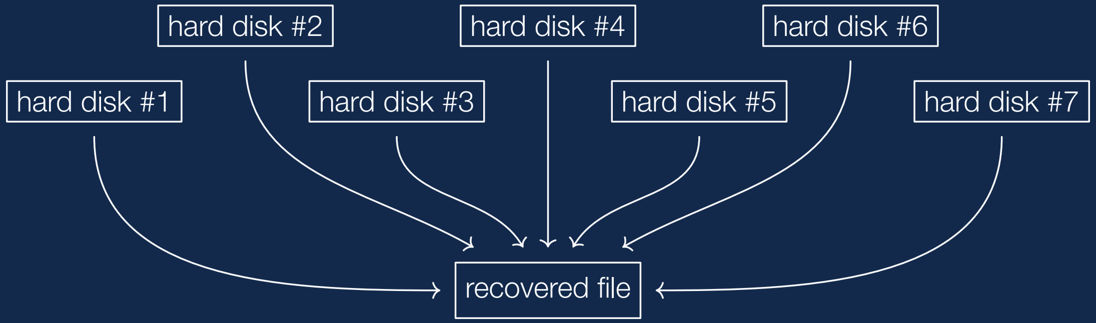
Initially, you upload a big file to the cloud. Each hard disk will store part of your file.

You then delete the local copy to free up some space.

Later, when you need the file, you download the file from the cloud.
The cloud company guarantees that the downloaded file is exactly the same file as before.

From your point of view, things work smoothly. But the cloud sees it differently.

Motivation from cloud storage services



A cloud storage service is a collection of hard disks that help you store big files.

Initially, you upload a big file to the cloud. Each hard disk will store part of your file.

You then delete the local copy to free up some space.

Later, when you need the file, you download the file from the cloud.
The cloud company guarantees that the downloaded file is exactly the same file as before.

From your point of view, things work smoothly. But the cloud sees it differently.

Motivation from cloud storage services

hard disk #2 hard disk #4 hard disk #6

hard disk #1 hard disk #3 hard disk #5 hard disk #7

happy customer =)

A cloud storage service is a collection of hard disks that help you store big files.

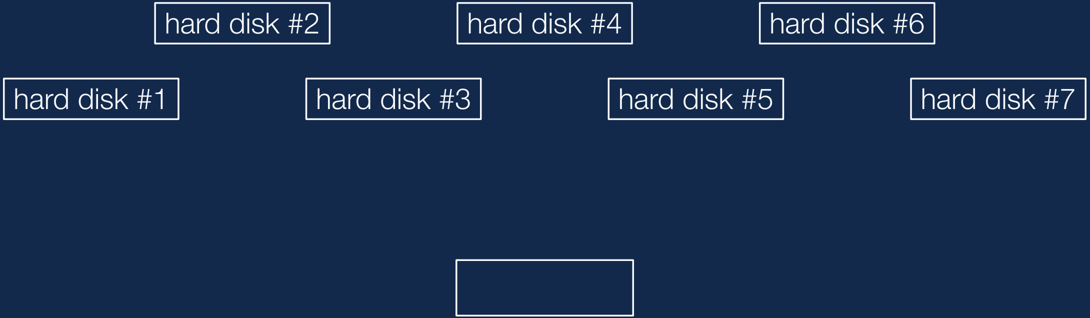
Initially, you upload a big file to the cloud. Each hard disk will store part of your file.

You then delete the local copy to free up some space.

Later, when you need the file, you download the file from the cloud.
The cloud company guarantees that the downloaded file is exactly the same file as before.

From your point of view, things work smoothly. But the cloud sees it differently.

What could go wrong behind the scene?

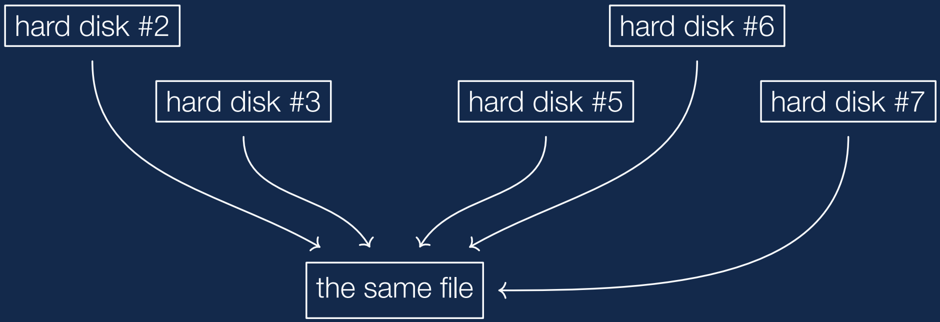


Difficulty (A): Errors (mostly erasures) occur spontaneously. To ensure that random errors do not corrupt your file, the cloud is equipped with error correcting codes.

For instance, the cloud may use a $[7, 5, 3]$-MDS code to protect the file, meaning that every set of five disks contains sufficient information to recover the file. Example 2 3

Difficulty (B): Fixing errors costs money. We can certainly reconstruct the entire file from healthy disks and simulate the user-uploading phase. But can it be cheaper? Example 2 3

What could go wrong behind the scene?

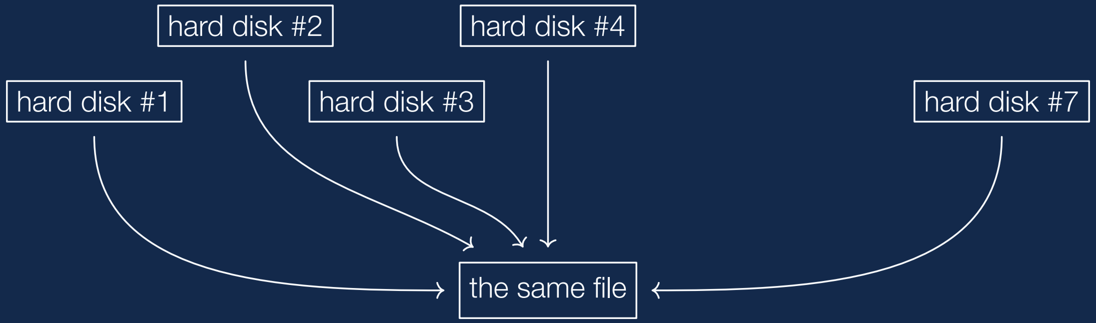


Difficulty (A): Errors (mostly erasures) occur spontaneously. To ensure that random errors do not corrupt your file, the cloud is equipped with error correcting codes.

For instance, the cloud may use a $[7,5,3]$-MDS code to protect the file, meaning that every set of five disks contains sufficient information to recover the file. Example 2 3

Difficulty (B): Fixing errors costs money. We can certainly reconstruct the entire file from healthy disks and simulate the user-uploading phase. But can it be cheaper? Example 2 3

What could go wrong behind the scene?

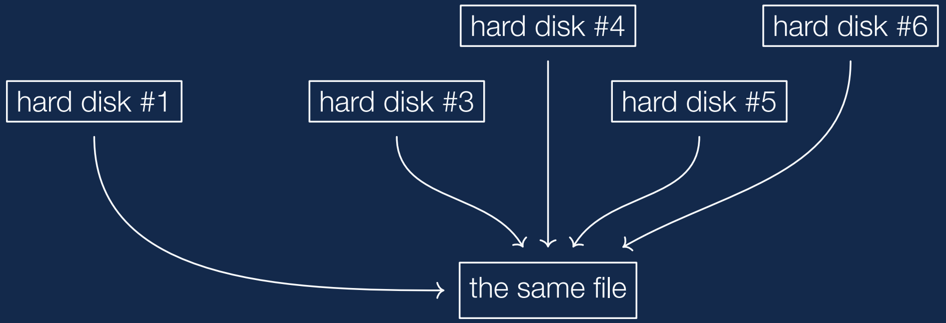


Difficulty (A): Errors (mostly erasures) occur spontaneously. To ensure that random errors do not corrupt your file, the cloud is equipped with error correcting codes.

For instance, the cloud may use a $[7, 5, 3]$-MDS code to protect the file, meaning that every set of five disks contains sufficient information to recover the file. Example 2 3

Difficulty (B): Fixing errors costs money. We can certainly reconstruct the entire file from healthy disks and simulate the user-uploading phase. But can it be cheaper? Example 2 3

What could go wrong behind the scene?

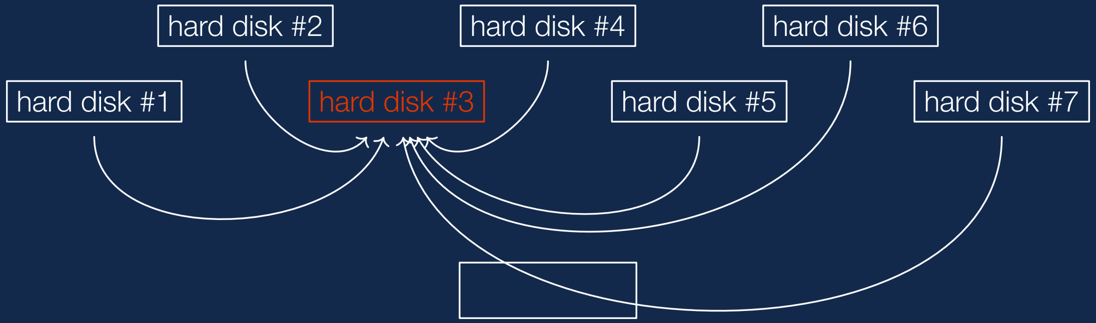


Difficulty (A): Errors (mostly erasures) occur spontaneously. To ensure that random errors do not corrupt your file, the cloud is equipped with error correcting codes.

For instance, the cloud may use a $[7, 5, 3]$-MDS code to protect the file, meaning that every set of five disks contains sufficient information to recover the file. Example 2 3

Difficulty (B): Fixing errors costs money. We can certainly reconstruct the entire file from healthy disks and simulate the user-uploading phase. But can it be cheaper? Example 2 3

What could go wrong behind the scene?

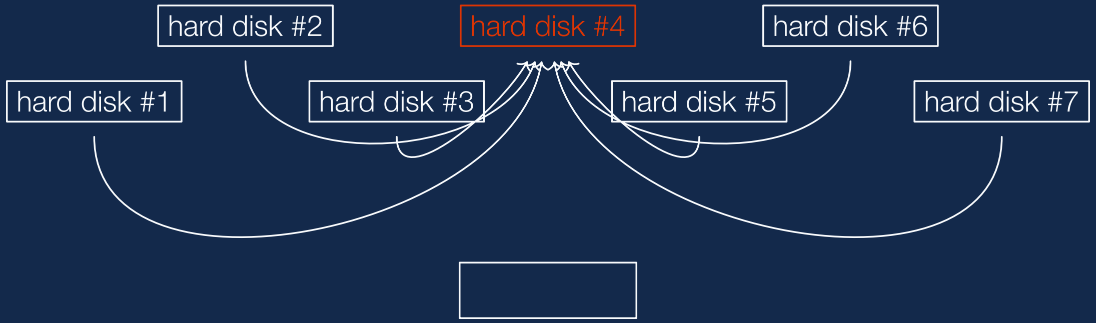


Difficulty (A): Errors (mostly erasures) occur spontaneously. To ensure that random errors do not corrupt your file, the cloud is equipped with error correcting codes.

For instance, the cloud may use a $[7, 5, 3]$-MDS code to protect the file, meaning that every set of five disks contains sufficient information to recover the file. Example 2 3

Difficulty (B): Fixing errors costs money. We can certainly reconstruct the entire file from healthy disks and simulate the user-uploading phase. But can it be cheaper? Example 2 3

What could go wrong behind the scene?

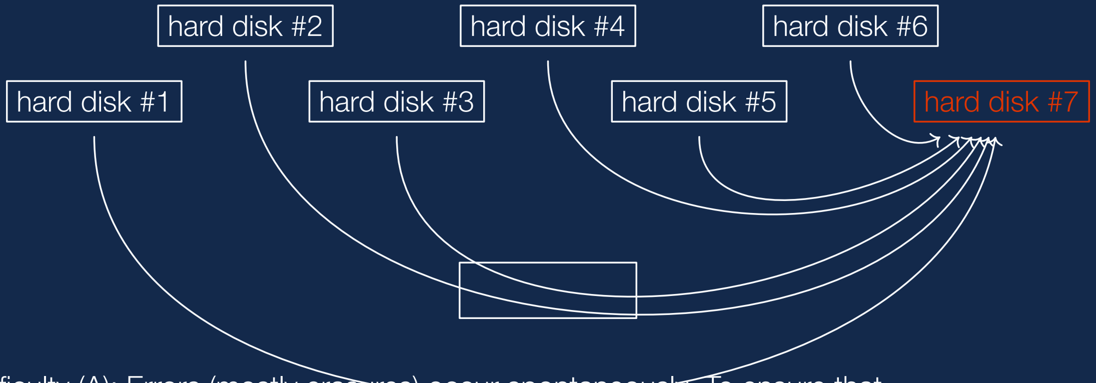


Difficulty (A): Errors (mostly erasures) occur spontaneously. To ensure that random errors do not corrupt your file, the cloud is equipped with error correcting codes.

For instance, the cloud may use a $[7, 5, 3]$-MDS code to protect the file, meaning that every set of five disks contains sufficient information to recover the file. Example 2 3

Difficulty (B): Fixing errors costs money. We can certainly reconstruct the entire file from healthy disks and simulate the user-uploading phase. But can it be cheaper? Example 2 3

What could go wrong behind the scene?



Difficulty (A): Errors (mostly erasures) occur spontaneously. To ensure that random errors do not corrupt your file, the cloud is equipped with error correcting codes.

For instance, the cloud may use a $[7,5,3]$-MDS code to protect the file, meaning that every set of five disks contains sufficient information to recover the file. Example 2 3

Difficulty (B): Fixing errors costs money. We can certainly reconstruct the entire file from healthy disks and simulate the user-uploading phase. But can it be cheaper? Example 2 3

Some notations and criteria of being a good cloud

α-symbol disk #2 α-symbol disk #4 α-symbol disk #6

α-symbol disk #1 α-symbol disk #3 α-symbol disk #5 α-symbol disk #7

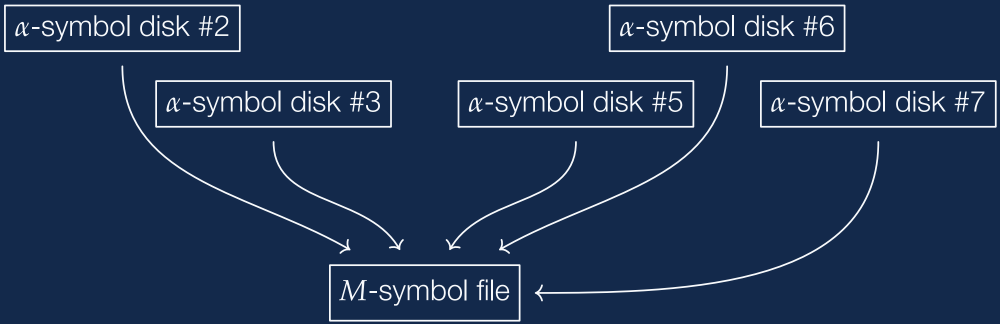
M-symbol file

The following notations are used in related literature: The file consists of M symbols. There are n $(= 7)$ hard disks. Every disk can store α symbols. (α is called *disk capacity*).

Requirement (A): Every set of k $(= 5)$ disks contains sufficient information to recover the file.

Requirement (B): Every set of d $(= 6)$ disks can reconstruct, from scratch, the remaining disk by each sending out β symbols of information it has. (β is called *repair bandwidth*.)

Some notations and criteria of being a good cloud

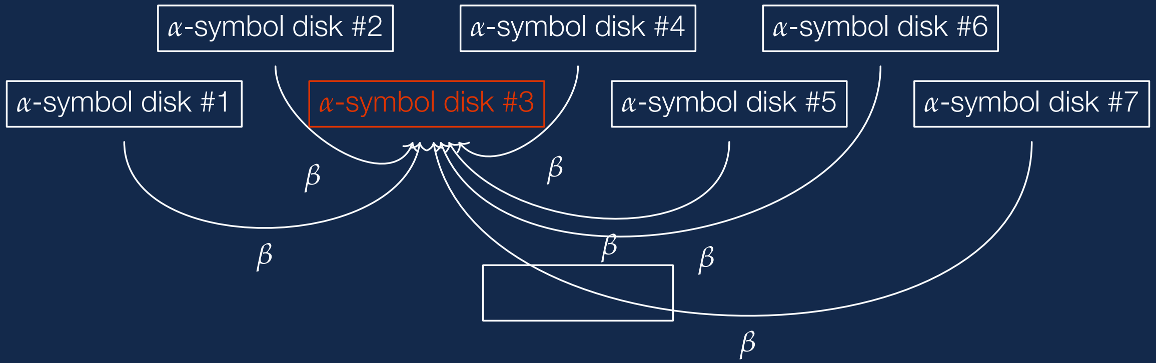


The following notations are used in related literature: The file consists of M symbols. There are n $(= 7)$ hard disks. Every disk can store α symbols. (α is called *disk capacity*).

Requirement (A): Every set of k $(= 5)$ disks contains sufficient information to recover the file.

Requirement (B): Every set of d $(= 6)$ disks can reconstruct, from scratch, the remaining disk by each sending out β symbols of information it has. (β is called *repair bandwidth*.)

Some notations and criteria of being a good cloud



The following notations are used in related literature: The file consists of M symbols. There are n $(= 7)$ hard disks. Every disk can store α symbols. (α is called *disk capacity*).

Requirement (A): Every set of k $(= 5)$ disks contains sufficient information to recover the file.

Requirement (B): Every set of d $(= 6)$ disks can reconstruct, from scratch, the remaining disk by each sending out β symbols of information it has. (β is called *repair bandwidth*.)

Designing cloud as a linear coding problem

We can now formalize what a cloud should satisfy.

File size is M symbols $\longleftrightarrow$ File is a linear map $\varphi\colon \mathbb{F}^M \to \mathbb{F}$, where $\mathbb{F}$ is some large alphabet.

n disks of capacity α $\longleftrightarrow$ the hth disk stores restriction $\varphi\restriction_{X_h}$ for some subsp $\mathbb{F}^\alpha \cong X_h \subseteq \mathbb{F}^M$.

Any k disks recover the file $\longleftrightarrow$ Any k subspaces $X_{h_1}, \ldots, X_{h_k}$ span φ's domain, $\mathbb{F}^M$.

Any d disks repair with bandwidth β $\longleftrightarrow$ $\exists$ subsps $\mathbb{F}^\beta \cong Y_h^f \subseteq X_h$ s.t. $Y_{h_1}^f + \cdots + Y_{h_d}^f \supseteq X_f$.

One such tuple $(n, k, d, \alpha, \beta, M; X_h, Y_h^f)$ is called a *regenerating code*.

Sanity check: $k \le d < n$ (repairing is possible) and $d\beta < M$ (repairing is nontrivial).

Designing cloud as a linear coding problem

We can now formalize what a cloud should satisfy.

File size is M symbols $\longleftrightarrow$ File is a linear map $\varphi\colon \mathbb{F}^M \to \mathbb{F}$, where $\mathbb{F}$ is some large alphabet.

n disks of capacity α $\longleftrightarrow$ the hth disk stores restriction $\varphi\restriction_{X_h}$ for some subsp $\mathbb{F}^\alpha \cong X_h \subseteq \mathbb{F}^M$.

Any k disks recover the file $\longleftrightarrow$ Any k subspaces $X_{h_1}, \dots, X_{h_k}$ span φ's domain, $\mathbb{F}^M$.

Any d disks repair with bandwidth β $\longleftrightarrow$ $\exists$ subsps $\mathbb{F}^\beta \cong Y_h^f \subseteq X_h$ s.t. $Y_{h_1}^f + \cdots + Y_{h_d}^f \supseteq X_f$.

One such tuple $(n, k, d, \alpha, \beta, M; X_h, Y_h^f)$ is called a *regenerating code*.

Sanity check: $k \le d < n$ (repairing is possible) and $d\beta < M$ (repairing is nontrivial).

Designing cloud as a linear coding problem

We can now formalize what a cloud should satisfy.

File size is M symbols $\longleftrightarrow$ File is a linear map $\varphi\colon \mathbb{F}^M \to \mathbb{F}$, where $\mathbb{F}$ is some large alphabet.

n disks of capacity α $\longleftrightarrow$ the hth disk stores restriction $\varphi\restriction_{X_h}$ for some subsp $\mathbb{F}^\alpha \cong X_h \subseteq \mathbb{F}^M$.

Any k disks recover the file $\longleftrightarrow$ Any k subspaces $X_{h_1}, \dots, X_{h_k}$ span φ's domain, $\mathbb{F}^M$.

Any d disks repair with bandwidth β $\longleftrightarrow$ $\exists$ subsps $\mathbb{F}^\beta \cong Y_h^f \subseteq X_h$ s.t. $Y_{h_1}^f + \cdots + Y_{h_d}^f \supseteq X_f$.

One such tuple $(n, k, d, \alpha, \beta, M; X_h, Y_h^f)$ is called a *regenerating code*.

Sanity check: $k \leq d < n$ (repairing is possible) and $d\beta < M$ (repairing is nontrivial).

Designing cloud as a linear coding problem

We can now formalize what a cloud should satisfy.

File size is M symbols $\longleftrightarrow$ File is a linear map $\varphi\colon \mathbb{F}^M \to \mathbb{F}$, where $\mathbb{F}$ is some large alphabet.

n disks of capacity α $\longleftrightarrow$ the hth disk stores restriction $\varphi\restriction_{X_h}$ for some subsp $\mathbb{F}^\alpha \cong X_h \subseteq \mathbb{F}^M$.

Any k disks recover the file $\longleftrightarrow$ Any k subspaces $X_{h_1}, \dots, X_{h_k}$ span φ's domain, $\mathbb{F}^M$.

Any d disks repair with bandwidth β $\longleftrightarrow$ $\exists$ subsps $\mathbb{F}^\beta \cong Y_h^f \subseteq X_h$ s.t. $Y_{h_1}^f + \cdots + Y_{h_d}^f \supseteq X_f$.

One such tuple $(n,k,d,\alpha,\beta,M;X_h,Y_h^f)$ is called a *regenerating code*.

Sanity check: $k \le d < n$ (repairing is possible) and $d\beta < M$ (repairing is nontrivial).

Designing cloud as a linear coding problem

We can now formalize what a cloud should satisfy.

File size is M symbols $\longleftrightarrow$ File is a linear map $\varphi\colon \mathbb{F}^M \to \mathbb{F}$, where $\mathbb{F}$ is some large alphabet.

n disks of capacity $\alpha \longleftrightarrow$ the hth disk stores restriction $\varphi\restriction_{X_h}$ for some subsp $\mathbb{F}^\alpha \cong X_h \subseteq \mathbb{F}^M$.

Any k disks recover the file $\longleftrightarrow$ Any k subspaces $X_{h_1}, \dots, X_{h_k}$ span φ's domain, $\mathbb{F}^M$.

Any d disks repair with bandwidth $\beta \longleftrightarrow \exists$ subsps $\mathbb{F}^\beta \cong Y_h^f \subseteq X_h$ s.t. $Y_{h_1}^f + \cdots + Y_{h_d}^f \supseteq X_f$.

One such tuple $(n, k, d, \alpha, \beta, M; X_h, Y_h^f)$ is called a *regenerating code*.

Sanity check: $k \le d < n$ (repairing is possible) and $d\beta < M$ (repairing is nontrivial).

Designing cloud as a linear coding problem

We can now formalize what a cloud should satisfy.

File size is M symbols $\longleftrightarrow$ File is a linear map $\varphi\colon \mathbb{F}^M \to \mathbb{F}$, where $\mathbb{F}$ is some large alphabet.

n disks of capacity α $\longleftrightarrow$ the hth disk stores restriction $\varphi\restriction_{X_h}$ for some subsp $\mathbb{F}^\alpha \cong X_h \subseteq \mathbb{F}^M$.

Any k disks recover the file $\longleftrightarrow$ Any k subspaces $X_{h_1}, \dots, X_{h_k}$ span φ's domain, $\mathbb{F}^M$.

Any d disks repair with bandwidth β $\longleftrightarrow$ $\exists$ subsps $\mathbb{F}^\beta \cong Y_h^f \subseteq X_h$ s.t. $Y_{h_1}^f + \cdots + Y_{h_d}^f \supseteq X_f$.

One such tuple $(n, k, d, \alpha, \beta, M; X_h, Y_h^f)$ is called a *regenerating code*.

Sanity check: $k \le d < n$ (repairing is possible) and $d\beta < M$ (repairing is nontrivial).

Designing cloud as a linear coding problem

We can now formalize what a cloud should satisfy.

File size is M symbols $\longleftrightarrow$ File is a linear map $\varphi\colon \mathbb{F}^M \to \mathbb{F}$, where $\mathbb{F}$ is some large alphabet.

n disks of capacity α $\longleftrightarrow$ the hth disk stores restriction $\varphi\restriction_{X_h}$ for some subsp $\mathbb{F}^\alpha \cong X_h \subseteq \mathbb{F}^M$.

Any k disks recover the file $\longleftrightarrow$ Any k subspaces $X_{h_1}, \ldots, X_{h_k}$ span φ's domain, $\mathbb{F}^M$.

Any d disks repair with bandwidth β $\longleftrightarrow$ $\exists$ subsps $\mathbb{F}^\beta \cong Y_h^f \subseteq X_h$ s.t. $Y_{h_1}^f + \cdots + Y_{h_d}^f \supseteq X_f$.

One such tuple $(n, k, d, \alpha, \beta, M; X_h, Y_h^f)$ is called a *regenerating code*.

Sanity check: $k \le d < n$ (repairing is possible) and $d\beta < M$ (repairing is nontrivial).

Outline

Part I: Motivation from cloud storage services

Part II: Review multilinear algebra (Brief! Just one page!) $\Longleftarrow$

Part III: Actual construction of *Moulin Codes*

Multilinear algebra review (theses are the only properties we need)

Anti-commutativity: Let x, y, z be vectors, then $y \wedge y = 0$ and $x \wedge z = -z \wedge x$.

Multilinearity: Let t be scalar, then $x \otimes (y + tz) = x \otimes y + tx \otimes z$; and same for $\wedge$.

Functoriality: Let ξ, η, ζ be tensors, then $\xi \mapsto \zeta$ extends naturally to $\xi \otimes \eta \mapsto \zeta \otimes \eta$.

Multilinear algebra review (theses are the only properties we need)

Anti-commutativity: Let x, y, z be vectors, then $y \wedge y = 0$ and $x \wedge z = -z \wedge x$.

Multilinearity: Let t be scalar, then $x \otimes (y + tz) = x \otimes y + tx \otimes z$; and same for $\wedge$.

Functoriality: Let ξ, η, ζ be tensors, then $\xi \mapsto \zeta$ extends naturally to $\xi \otimes \eta \mapsto \zeta \otimes \eta$.

Multilinear algebra review (theses are the only properties we need)

Anti-commutativity: Let x, y, z be vectors, then $y \wedge y = 0$ and $x \wedge z = -z \wedge x$.

Multilinearity: Let t be scalar, then $x \otimes (y + tz) = x \otimes y + tx \otimes z$; and same for $\wedge$.

Functoriality: Let ξ, η, ζ be tensors, then $\xi \mapsto \zeta$ extends naturally to $\xi \otimes \eta \mapsto \zeta \otimes \eta$.

Outline

Part I: Motivation from cloud storage services

Part II: Review multilinear algebra (Brief! Just one page!)

Part III: Actual construction of *Moulin Codes* ⟸

Construct Moulin Code for special $k = d$

Recall: n disks; k recover the file; d repair erased disks; $k \le d < n$, otherwise trivial.

Let $W := \mathbb{F}^k$. Consider the wedge-multiplication

$$\begin{aligned} W \otimes W \wedge W &\longrightarrow W \wedge W \wedge W, \\ x \otimes y \wedge z &\longmapsto x \wedge y \wedge z. \end{aligned}$$

It has a natural transpose/dual map, called co-wedge-multiplication

$$\begin{aligned} \nabla\colon W \wedge W \wedge W &\longrightarrow W \otimes W \wedge W, \\ x \wedge y \wedge z &\longmapsto x \otimes y \wedge z - y \otimes x \wedge z + z \otimes x \wedge y. \end{aligned}$$

Let the file be any map $\varphi\colon W \otimes W \wedge W \to \mathbb{F}$ such that $\varphi\restriction_{\operatorname{im}\nabla} = 0$, meaning that it satisfies parity checks $\varphi(\nabla(x \wedge y \wedge z)) = \varphi(x \otimes y \wedge z) - \varphi(y \otimes x \wedge z) + \varphi(z \otimes x \wedge y) = 0$.

Construct Moulin Code for special $k = d$

Recall: n disks; k recover the file; d repair erased disks; $k \leq d < n$, otherwise trivial.

Let $W := \mathbb{F}^k$. Consider the wedge-multiplication

$$\begin{aligned} W \otimes W \wedge W &\longrightarrow W \wedge W \wedge W, \\ x \otimes y \wedge z &\longmapsto x \wedge y \wedge z. \end{aligned}$$

It has a natural transpose/dual map, called co-wedge-multiplication

$$\begin{aligned} \nabla \colon W \wedge W \wedge W &\longrightarrow W \otimes W \wedge W, \\ x \wedge y \wedge z &\longmapsto x \otimes y \wedge z - y \otimes x \wedge z + z \otimes x \wedge y. \end{aligned}$$

Let the file be any map $\varphi \colon W \otimes W \wedge W \to \mathbb{F}$ such that $\varphi\restriction_{\operatorname{im}\nabla} = 0$, meaning that it satisfies parity checks $\varphi(\nabla(x \wedge y \wedge z)) = \varphi(x \otimes y \wedge z) - \varphi(y \otimes x \wedge z) + \varphi(z \otimes x \wedge y) = 0$.

Construct Moulin Code for special $k = d$

Recall: n disks; k recover the file; d repair erased disks; $k \le d < n$, otherwise trivial.

Let $W := \mathbb{F}^k$. Consider the wedge-multiplication

$$\begin{aligned} W \otimes W \wedge W &\longrightarrow W \wedge W \wedge W, \\ x \otimes y \wedge z &\longmapsto x \wedge y \wedge z. \end{aligned}$$

It has a natural transpose/dual map, called co-wedge-multiplication

$$\begin{aligned} \nabla\colon W \wedge W \wedge W &\longrightarrow W \otimes W \wedge W, \\ x \wedge y \wedge z &\longmapsto x \otimes y \wedge z - y \otimes x \wedge z + z \otimes x \wedge y. \end{aligned}$$

Let the file be any map $\varphi\colon W \otimes W \wedge W \to \mathbb{F}$ such that $\varphi\restriction_{\operatorname{im}\nabla} = 0$, meaning that it satisfies parity checks $\varphi(\nabla(x \wedge y \wedge z)) = \varphi(x \otimes y \wedge z) - \varphi(y \otimes x \wedge z) + \varphi(z \otimes x \wedge y) = 0$.

$k = d$ special case, page 2

Recall: n disks; k recover the file; d repair erased disks. $W := \mathbb{F}^k$; and $\varphi\restriction_{\operatorname{im}\nabla} = 0$.

Let the n disks choose vectors $c_1, c_2, \ldots, c_n \in W$ that are MDS (i.e., any k of which span W). Let the disks store the restrictions $\varphi\restriction_{c_1 \otimes W \wedge W}$ and $\varphi\restriction_{c_2 \otimes W \wedge W}$ and all the way to $\varphi\restriction_{c_n \otimes W \wedge W}$

Any k disks recover the file: Let $\mathcal{K}$ be a set of k indices, then by multilinearity & MDSness, $\sum_{h \in \mathcal{K}} c_h \otimes W \wedge W = \operatorname{span}\langle c_h : h \in \mathcal{K}\rangle \otimes W \wedge W = W \otimes W \wedge W =$ the entire domain of φ.

When the fth is erased, the hth sends it $\varphi\restriction_{c_h \otimes c_f \wedge W}$ (notice $c_h \otimes c_f \wedge W \subseteq c_h \otimes W \wedge W$).

Let $\mathcal{D}$ be a set of d indices, $\sum_{h \in \mathcal{D}} c_h \otimes c_f \wedge W = \operatorname{span}\langle c_h : h \in \mathcal{D}\rangle \otimes c_f \wedge W = W \otimes c_f \wedge W$. We repair $\varphi(c_f \otimes y \wedge z) = \varphi(y \otimes c_f \wedge z) - \varphi(z \otimes c_f \wedge y)$ as the latter is learned from $\varphi\restriction_{W \otimes c_f \wedge W}$

$k = d$ special case, page 2

Recall: n disks; k recover the file; d repair erased disks. $W := \mathbb{F}^k$; and $\varphi\restriction_{\operatorname{im}\nabla} = 0$.

Let the n disks choose vectors $c_1, c_2, \dots, c_n \in W$ that are MDS (i.e., any k of which span W). Let the disks store the restrictions $\varphi\restriction_{c_1 \otimes W \wedge W}$ and $\varphi\restriction_{c_2 \otimes W \wedge W}$ and all the way to $\varphi\restriction_{c_n \otimes W \wedge W}$

Any k disks recover the file: Let $\mathcal{K}$ be a set of k indices, then by multilinearity & MDSness, $\sum_{h \in \mathcal{K}} c_h \otimes W \wedge W = \operatorname{span}\langle c_h : h \in \mathcal{K}\rangle \otimes W \wedge W = W \otimes W \wedge W =$ the entire domain of φ.

When the fth is erased, the hth sends it $\varphi\restriction_{c_h \otimes c_f \wedge W}$ (notice $c_h \otimes c_f \wedge W \subseteq c_h \otimes W \wedge W$).

Let $\mathcal{D}$ be a set of d indices, $\sum_{h \in \mathcal{D}} c_h \otimes c_f \wedge W = \operatorname{span}\langle c_h : h \in \mathcal{D}\rangle \otimes c_f \wedge W = W \otimes c_f \wedge W$. We repair $\varphi(c_f \otimes y \wedge z) = \varphi(y \otimes c_f \wedge z) - \varphi(z \otimes c_f \wedge y)$ as the latter is learned from $\varphi\restriction_{W \otimes c_f \wedge W}$

$k = d$ special case, page 2

Recall: n disks; k recover the file; d repair erased disks. $W := \mathbb{F}^k$; and $\varphi\restriction_{\operatorname{im}\nabla} = 0$.

Let the n disks choose vectors $c_1, c_2, \dots, c_n \in W$ that are MDS (i.e., any k of which span W).
Let the disks store the restrictions $\varphi\restriction_{c_1 \otimes W \wedge W}$ and $\varphi\restriction_{c_2 \otimes W \wedge W}$ and all the way to $\varphi\restriction_{c_n \otimes W \wedge W}$

Any k disks recover the file: Let $\mathcal{K}$ be a set of k indices, then by multilinearity & MDSness,
$\sum_{h \in \mathcal{K}} c_h \otimes W \wedge W = \operatorname{span}\langle c_h : h \in \mathcal{K}\rangle \otimes W \wedge W = W \otimes W \wedge W =$ the entire domain of φ.

When the fth is erased, the hth sends it $\varphi\restriction_{c_h \otimes c_f \wedge W}$ (notice $c_h \otimes c_f \wedge W \subseteq c_h \otimes W \wedge W$).

Let $\mathcal{D}$ be a set of d indices, $\sum_{h \in \mathcal{D}} c_h \otimes c_f \wedge W = \operatorname{span}\langle c_h : h \in \mathcal{D}\rangle \otimes c_f \wedge W = W \otimes c_f \wedge W$.
We repair $\varphi(c_f \otimes y \wedge z) = \varphi(y \otimes c_f \wedge z) - \varphi(z \otimes c_f \wedge y)$ as the latter is learned from $\varphi\restriction_{W \otimes c_f \wedge W}$

$k = d$ special case, page 2

Recall: n disks; k recover the file; d repair erased disks. $W := \mathbb{F}^k$; and $\varphi\restriction_{\mathrm{im}\,\nabla} = 0$.

Let the n disks choose vectors $c_1, c_2, \ldots, c_n \in W$ that are MDS (i.e., any k of which span W).
Let the disks store the restrictions $\varphi\restriction_{c_1 \otimes W \wedge W}$ and $\varphi\restriction_{c_2 \otimes W \wedge W}$ and all the way to $\varphi\restriction_{c_n \otimes W \wedge W}$

Any k disks recover the file: Let $\mathcal{K}$ be a set of k indices, then by multilinearity & MDSness,
$\sum_{h \in \mathcal{K}} c_h \otimes W \wedge W = \mathrm{span}\langle c_h : h \in \mathcal{K}\rangle \otimes W \wedge W = W \otimes W \wedge W =$ the entire domain of φ.

When the fth is erased, the hth sends it $\varphi\restriction_{c_h \otimes c_f \wedge W}$ (notice $c_h \otimes c_f \wedge W \subseteq c_h \otimes W \wedge W$).

Let $\mathcal{D}$ be a set of d indices, $\sum_{h \in \mathcal{D}} c_h \otimes c_f \wedge W = \mathrm{span}\langle c_h : h \in \mathcal{D}\rangle \otimes c_f \wedge W = W \otimes c_f \wedge W$.
We repair $\varphi(c_f \otimes y \wedge z) = \varphi(y \otimes c_f \wedge z) - \varphi(z \otimes c_f \wedge y)$ as the latter is learned from $\varphi\restriction_{W \otimes c_f \wedge W}$

Construct Moulin Code for general case $k \le d$

Recall: n disks; k recover the file; d repair erased disks. $W := \mathbb{F}^k$.

Let $V := \mathbb{F}^{d-k}$. Let $U := V \oplus W = \mathbb{F}^d$. Consider the diagram:

$$\begin{tikzcd}[column sep=tiny]
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, hook, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, hook, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, hook, shift left] \\
& & U \otimes W \wedge W \arrow[ur, hook, shift left] \arrow[dl, hook, shift left] & & & V \otimes U \otimes W \arrow[ur, hook, shift left] \arrow[dl, hook, shift left] & & & V \otimes V \otimes U \arrow[ur, hook, shift left] \arrow[dl, hook, shift left] & \\
& W \otimes W \wedge W \arrow[ur, hook, shift left] & & & V \otimes W \otimes W \arrow[ur, hook, shift left] & & & V \otimes V \otimes W \arrow[ur, hook, shift left] & &
\end{tikzcd}$$

The file is a map φ: direct sum of U-spaces $\to \mathbb{F}$ that vanishes on $\operatorname{im}(\mathrm{id} - \nabla)$.

I.e., $0 =: \varphi(x \wedge y \wedge z) = \varphi(x \otimes y \wedge z) - \varphi(y \otimes x \wedge z) + \varphi(z \otimes x \wedge y)$ where $x, y, z \in W$

and $\varphi(x \otimes y \wedge z) = \varphi(x \otimes y \otimes z) - \varphi(x \otimes z \otimes y)$ where $x \in V$ and $y, z \in W$

and $\varphi(x \otimes y \otimes z) = \varphi(x \otimes y \otimes z)$ where $x, y \in V$ and $z \in W$ (not tautology but gluing)

and $\varphi(a \otimes y \otimes z) = \varphi(\nabla(a \otimes y \otimes z)) := 0$ where $x, y, z \in V$.

Construct Moulin Code for general case $k \leq d$

Recall: n disks; k recover the file; d repair erased disks. $W := \mathbb{F}^k$.

Let $V := \mathbb{F}^{d-k}$. Let $U := V \oplus W = \mathbb{F}^d$. Consider the diagram:

$$
\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, hook] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, hook] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, hook] \\
& & U \otimes W \wedge W \arrow[ur, two heads] \arrow[dl, two heads] & & & V \otimes U \otimes W \arrow[ur, two heads] \arrow[dl, two heads] & & & V \otimes V \otimes U \arrow[ur, two heads] \arrow[dl, two heads] & \\
& W \otimes W \wedge W \arrow[ur, hook] & & & V \otimes W \otimes W \arrow[ur, hook] & & & V \otimes V \otimes W \arrow[ur, hook] & &
\end{tikzcd}
$$

The file is a map φ: direct sum of U-spaces $\to \mathbb{F}$ that vanishes on $\operatorname{im}(\operatorname{id} - \nabla)$.
I.e., $0 =: \varphi(x \wedge y \wedge z) = \varphi(x \otimes y \wedge z) - \varphi(y \otimes x \wedge z) + \varphi(z \otimes x \wedge y)$ where $x, y, z \in W$
and $\varphi(x \otimes y \wedge z) = \varphi(x \otimes y \otimes z) - \varphi(x \otimes z \otimes y)$ where $x \in V$ and $y, z \in W$
and $\varphi(x \otimes y \otimes z) = \varphi(x \otimes y \otimes z)$ where $x, y \in V$ and $z \in W$ (not tautology but gluing)
and $\varphi(a \otimes y \otimes z) = \varphi(\nabla(a \otimes y \otimes z)) := 0$ where $x, y, z \in V$.

Construct Moulin Code for general case $k \le d$

Recall: n disks; k recover the file; d repair erased disks. $W := \mathbb{F}^k$.

Let $V := \mathbb{F}^{d-k}$. Let $U := V \oplus W = \mathbb{F}^d$. Consider the diagram:

$$
\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, hook, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, hook, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, hook, shift left] \\
& & U \otimes W \wedge W \arrow[ur, two heads, shift left] \arrow[dl, two heads, shift left] & & & V \otimes U \otimes W \arrow[ur, two heads, shift left] \arrow[dl, two heads, shift left] & & & V \otimes V \otimes U \arrow[ur, two heads, shift left] \arrow[dl, two heads, shift left] & \\
& W \otimes W \wedge W \arrow[ur, hook, shift left] & & & V \otimes W \otimes W \arrow[ur, hook, shift left] & & & V \otimes V \otimes W \arrow[ur, hook, shift left] & &
\end{tikzcd}
$$

The file is a map φ: direct sum of U-spaces $\to \mathbb{F}$ that vanishes on $\operatorname{im}(\mathrm{id} - \nabla)$.

I.e., $0 =: \varphi(x \wedge y \wedge z) = \varphi(x \otimes y \wedge z) - \varphi(y \otimes x \wedge z) + \varphi(z \otimes x \wedge y)$ where $x, y, z \in W$

and $\varphi(x \otimes y \wedge z) = \varphi(x \otimes y \otimes z) - \varphi(x \otimes z \otimes y)$ where $x \in V$ and $y, z \in W$

and $\varphi(x \otimes y \otimes z) = \varphi(x \otimes y \otimes z)$ where $x, y \in V$ and $z \in W$ (not tautology but gluing)

and $\varphi(a \otimes y \otimes z) = \varphi(\nabla(a \otimes y \otimes z)) := 0$ where $x, y, z \in V$.

Construct Moulin Code for general case $k \leq d$

Recall: n disks; k recover the file; d repair erased disks. $W := \mathbb{F}^k$.

Let $V := \mathbb{F}^{d-k}$. Let $U := V \oplus W = \mathbb{F}^d$. Consider the diagram:

$$
\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, hook] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, hook] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, hook] \\
& & U \otimes W \wedge W \arrow[ur, two heads] \arrow[dl, two heads] & & & V \otimes U \otimes W \arrow[ur, two heads] \arrow[dl, two heads] & & & V \otimes V \otimes U \arrow[ur, two heads] \arrow[dl, two heads] & \\
& W \otimes W \wedge W \arrow[ur, hook] & & & V \otimes W \otimes W \arrow[ur, hook] & & & V \otimes V \otimes W \arrow[ur, hook] & &
\end{tikzcd}
$$

The file is a map φ: direct sum of U-spaces $\to \mathbb{F}$ that vanishes on $\operatorname{im}(\mathrm{id} - \nabla)$.

I.e., $0 =: \varphi(x \wedge y \wedge z) = \varphi(x \otimes y \wedge z) - \varphi(y \otimes x \wedge z) + \varphi(z \otimes x \wedge y)$ where $x, y, z \in W$

and $\varphi(x \otimes y \wedge z) = \varphi(x \otimes y \otimes z) - \varphi(x \otimes z \otimes y)$ where $x \in V$ and $y, z \in W$

and $\varphi(x \otimes y \otimes z) = \varphi(x \otimes y \otimes z)$ where $x, y \in V$ and $z \in W$ (not tautology but gluing)

and $\varphi(a \otimes y \otimes z) = \varphi(\nabla(a \otimes y \otimes z)) := 0$ where $x, y, z \in V$.

Construct Moulin Code for general case $k \leq d$

Recall: n disks; k recover the file; d repair erased disks. $W := \mathbb{F}^k$.

Let $V := \mathbb{F}^{d-k}$. Let $U := V \oplus W = \mathbb{F}^d$. Consider the diagram:

$$
\begin{tikzcd}[column sep=tiny]
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, hook, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, hook, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, hook, shift left] \\
& & U \otimes W \wedge W \arrow[ur, hook, shift left] \arrow[dl, hook, shift left] & & & V \otimes U \otimes W \arrow[ur, hook, shift left] \arrow[dl, hook, shift left] & & & V \otimes V \otimes U \arrow[ur, hook, shift left] \arrow[dl, hook, shift left] & \\
& W \otimes W \wedge W \arrow[ur, hook, shift left] & & & V \otimes W \otimes W \arrow[ur, hook, shift left] & & & V \otimes V \otimes W \arrow[ur, hook, shift left] & &
\end{tikzcd}
$$

The file is a map φ: direct sum of U-spaces $\to \mathbb{F}$ that vanishes on $\operatorname{im}(\mathrm{id} - \nabla)$.

I.e., $0 =: \varphi(x \wedge y \wedge z) = \varphi(x \otimes y \wedge z) - \varphi(y \otimes x \wedge z) + \varphi(z \otimes x \wedge y)$ where $x, y, z \in W$

and $\varphi(x \otimes y \wedge z) = \varphi(x \otimes y \otimes z) - \varphi(x \otimes z \otimes y)$ where $x \in V$ and $y, z \in W$

and $\varphi(x \otimes y \otimes z) = \varphi(x \otimes y \otimes z)$ where $x, y \in V$ and $z \in W$ (not tautology but gluing)

and $\varphi(a \otimes y \otimes z) = \varphi(\nabla(a \otimes y \otimes z)) := 0$ where $x, y, z \in V$.

Construct Moulin Code for general case $k \le d$

Recall: n disks; k recover the file; d repair erased disks. $W := \mathbb{F}^k$.

Let $V := \mathbb{F}^{d-k}$. Let $U := V \oplus W = \mathbb{F}^d$. Consider the diagram:

$$
\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, hook] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, hook] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, hook] \\
& & U \otimes W \wedge W \arrow[ur, two heads] \arrow[dl, two heads] & & & V \otimes U \otimes W \arrow[ur, two heads] \arrow[dl, two heads] & & & V \otimes V \otimes U \arrow[ur, two heads] \arrow[dl, two heads] & \\
& W \otimes W \wedge W \arrow[ur, hook] & & & V \otimes W \otimes W \arrow[ur, hook] & & & V \otimes V \otimes W \arrow[ur, hook] & &
\end{tikzcd}
$$

The file is a map φ: direct sum of U-spaces $\to \mathbb{F}$ that vanishes on $\operatorname{im}(\mathrm{id} - \nabla)$.
I.e., $0 =: \varphi(x \wedge y \wedge z) = \varphi(x \otimes y \wedge z) - \varphi(y \otimes x \wedge z) + \varphi(z \otimes x \wedge y)$ where $x, y, z \in W$
and $\varphi(x \otimes y \wedge z) = \varphi(x \otimes y \otimes z) - \varphi(x \otimes z \otimes y)$ where $x \in V$ and $y, z \in W$
and $\varphi(x \otimes y \otimes z) = \varphi(x \otimes y \otimes z)$ where $x, y \in V$ and $z \in W$ (not tautology but gluing)
and $\varphi(a \otimes y \otimes z) = \varphi(\nabla(a \otimes y \otimes z)) := 0$ where $x, y, z \in V$.

Construct Moulin Code for general case $k \leq d$

Recall: n disks; k recover the file; d repair erased disks. $W := \mathbb{F}^k$.

Let $V := \mathbb{F}^{d-k}$. Let $U := V \oplus W = \mathbb{F}^d$. Consider the diagram:

$$\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[ddr, "\nabla"] \arrow[dl, hook] & & & V \otimes V \otimes W \arrow[ddr, "\nabla"] \arrow[dl, hook] & & & V \otimes V \otimes V \arrow[ddr, red, "\nabla"] \arrow[dl, hook] & \\
& & U \otimes W \wedge W \arrow[ur, hook, two heads] \arrow[dl, hook] & & & V \otimes U \otimes W \arrow[ur, hook, two heads] \arrow[dl, hook] & & & V \otimes V \otimes U \arrow[ur, hook, two heads] \arrow[dl, hook] & & \\
& W \otimes W \wedge W \arrow[ur, hook, two heads] & & & V \otimes W \otimes W \arrow[ur, hook, two heads] & & & V \otimes V \otimes W \arrow[ur, hook, two heads] & & &
\end{tikzcd}$$

The file is a map φ: direct sum of U-spaces $\to \mathbb{F}$ that vanishes on $\operatorname{im}(\mathrm{id} - \nabla)$.
I.e., $0 =: \varphi(x \wedge y \wedge z) = \varphi(x \otimes y \wedge z) - \varphi(y \otimes x \wedge z) + \varphi(z \otimes x \wedge y)$ where $x, y, z \in W$
and $\varphi(x \otimes y \wedge z) = \varphi(x \otimes y \otimes z) - \varphi(x \otimes z \otimes y)$ where $x \in V$ and $y, z \in W$
and $\varphi(x \otimes y \otimes z) = \varphi(x \otimes y \otimes z)$ where $x, y \in V$ and $z \in W$ (not tautology but gluing)
and $\varphi(a \otimes y \otimes z) = \varphi(\nabla(a \otimes y \otimes z)) := 0$ where $x, y, z \in V$.

$k \le d$ general case, page 2

Recall: n disks; k recover the file; d repair erased disks. $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$
File φ: $\bigoplus$ U-spaces $\to \mathbb{F}$ is such that $\varphi(\text{tensor}) = \varphi(\nabla(\text{tensor}))$.

Let the n disks choose vectors $a_1, a_2, \dots, a_n \in U$ that are MDS in the sense that (a) every d of them span U; and (b) every k of them span U/W after projection.

Let the hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$ (i.e., replace U by a_h).

Fix k disks $\mathcal{K}$. Let X be span$\langle a_h : h \in \mathcal{K}\rangle$. Then $V + X = U$. We learn restriction of φ to this by parity check, this by downloading, this by MDSness, this by projecting.

$$
\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & V \otimes W \wedge W \arrow[d, shift left] \arrow[ddr, "\nabla"] & & V \otimes V \otimes W \arrow[d, shift left] \arrow[ddr, "\nabla"] & & V \otimes V \otimes V \arrow[d, shift left] \\
& & U \otimes W \wedge W \arrow[u, shift left] \arrow[dl, shift left] & & V \otimes U \otimes W \arrow[u, shift left] \arrow[dl, shift left] & & V \otimes V \otimes U \arrow[u, shift left] \arrow[dl, shift left] \\
& W \otimes W \wedge W \arrow[ur, shift left] & & V \otimes W \otimes W \arrow[ur, shift left] & & V \otimes V \otimes W \arrow[ur, shift left] & \\
& & W \otimes W \wedge X \arrow[uu] & & V \otimes W \otimes X \arrow[uu] & & V \otimes V \otimes X \arrow[uu]
\end{tikzcd}
$$

$k \le d$ general case, page 2

Recall: n disks; k recover the file; d repair erased disks. $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$
File φ: $\bigoplus$ U-spaces $\to \mathbb{F}$ is such that $\varphi(\text{tensor}) = \varphi(\nabla(\text{tensor}))$.

Let the n disks choose vectors $a_1, a_2, \dots, a_n \in U$ that are MDS in the sense that (a) every d of them span U; and (b) every k of them span U/W after projection.

Let the hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$ (i.e., replace U by a_h).

Fix k disks $\mathcal{K}$. Let X be span$\langle a_h : h \in \mathcal{K}\rangle$. Then $V + X = U$. We learn restriction of φ to this by parity check, this by downloading, this by MDSness, this by projecting.

$$\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, shift left] \\
& & U \otimes W \wedge W \arrow[ur, shift left] \arrow[dl, shift left] & & & V \otimes U \otimes W \arrow[ur, shift left] \arrow[dl, shift left] & & & V \otimes V \otimes U \arrow[ur, shift left] \arrow[dl, shift left] & \\
& W \otimes W \wedge W \arrow[ur, shift left] & & & V \otimes W \otimes W \arrow[ur, shift left] & & & V \otimes V \otimes W \arrow[ur, shift left] & & \\
& & & W \otimes W \wedge X \arrow[uul] & & & V \otimes W \otimes X \arrow[uul] & & & V \otimes V \otimes X \arrow[uul]
\end{tikzcd}$$

$k \le d$ general case, page 2

Recall: n disks; k recover the file; d repair erased disks. $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$
File φ: $\bigoplus$ U-spaces $\to \mathbb{F}$ is such that $\varphi(\text{tensor}) = \varphi(\nabla(\text{tensor}))$.

Let the n disks choose vectors $a_1, a_2, \ldots, a_n \in U$ that are MDS in the sense that
(a) every d of them span U; and (b) every k of them span U/W after projection.

Let the hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$ (i.e., replace U by a_h).

Fix k disks $\mathcal{K}$. Let X be span$\langle a_h : h \in \mathcal{K} \rangle$. Then $V + X = U$. We learn restriction of φ to this by parity check, this by downloading, this by MDSness, this by projecting.

$$\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, shift left] \\
& & U \otimes W \wedge W \arrow[ur, shift left] \arrow[dl, shift left] & & & V \otimes U \otimes W \arrow[ur, shift left] \arrow[dl, shift left] & & & V \otimes V \otimes U \arrow[ur, shift left] \arrow[dl, shift left] & \\
& W \otimes W \wedge W \arrow[ur, shift left] & & & V \otimes W \otimes W \arrow[ur, shift left] & & & V \otimes V \otimes W \arrow[ur, shift left] & & \\
& & W \otimes W \wedge X \arrow[uu, hook] & & & V \otimes W \otimes X \arrow[uu, hook] & & & V \otimes V \otimes X \arrow[uu, hook] &
\end{tikzcd}$$

$k \leq d$ general case, page 2

Recall: n disks; k recover the file; d repair erased disks. $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$
File φ: $\bigoplus$ U-spaces $\to \mathbb{F}$ is such that $\varphi(\text{tensor}) = \varphi(\nabla(\text{tensor}))$.

Let the n disks choose vectors $a_1, a_2, \dots, a_n \in U$ that are MDS in the sense that
(a) every d of them span U; and (b) every k of them span U/W after projection.

Let the hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$ (i.e., replace U by a_h).

Fix k disks $\mathcal{K}$. Let X be span$\langle a_h : h \in \mathcal{K} \rangle$. Then $V + X = U$. We learn restriction of φ to this by parity check, this by downloading, this by MDSness, this by projecting.

$$\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, shift left] \\
& & U \otimes W \wedge W \arrow[ur, shift left] \arrow[dl, shift left] & & & V \otimes U \otimes W \arrow[ur, shift left] \arrow[dl, shift left] & & & V \otimes V \otimes U \arrow[ur, shift left] \arrow[dl, shift left] & \\
& W \otimes W \wedge W \arrow[ur, shift left] & & & V \otimes W \otimes W \arrow[ur, shift left] & & & V \otimes V \otimes W \arrow[ur, shift left] & & \\
& & & W \otimes W \wedge X \arrow[uul] & & & V \otimes W \otimes X \arrow[uul] & & & V \otimes V \otimes X \arrow[uul]
\end{tikzcd}$$

$k \le d$ general case, page 2

Recall: n disks; k recover the file; d repair erased disks. $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$
File φ: $\bigoplus$ U-spaces $\to \mathbb{F}$ is such that $\varphi(\text{tensor}) = \varphi(\nabla(\text{tensor}))$.

Let the n disks choose vectors $a_1, a_2, \ldots, a_n \in U$ that are MDS in the sense that
(a) every d of them span U; and (b) every k of them span U/W after projection.

Let the hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$ (i.e., replace U by a_h).

Fix k disks $\mathcal{K}$. Let X be span$\langle a_h : h \in \mathcal{K} \rangle$. Then $V + X = U$. We learn restriction of φ to this by parity check, this by downloading, this by MDSness, this by projecting.

$$\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, shift left] \\
& & U \otimes W \wedge W \arrow[ur, shift left] \arrow[dl, shift left] & & & V \otimes U \otimes W \arrow[ur, shift left] \arrow[dl, shift left] & & & V \otimes V \otimes U \arrow[ur, shift left] \arrow[dl, shift left] & \\
& W \otimes W \wedge W \arrow[ur, shift left] & & & V \otimes W \otimes W \arrow[ur, shift left] & & & V \otimes V \otimes W \arrow[ur, shift left] & & \\
& & W \otimes W \wedge X \arrow[uu] & & & V \otimes W \otimes X \arrow[uu] & & & V \otimes V \otimes X \arrow[uu] &
\end{tikzcd}$$

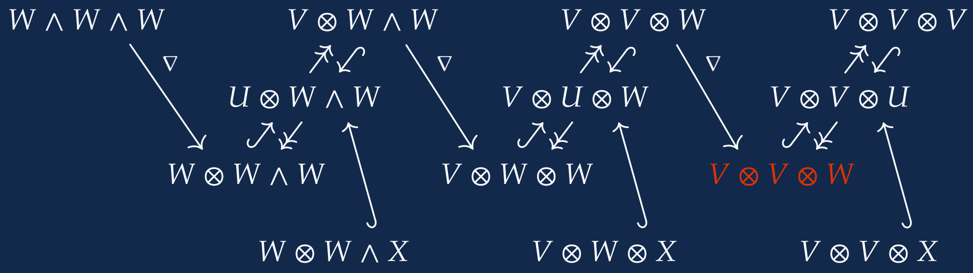
$k \leq d$ general case, page 2

Recall: n disks; k recover the file; d repair erased disks. $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$
File φ: $\bigoplus$ U-spaces $\to \mathbb{F}$ is such that $\varphi(\text{tensor}) = \varphi(\nabla(\text{tensor}))$.

Let the n disks choose vectors $a_1, a_2, \ldots, a_n \in U$ that are MDS in the sense that (a) every d of them span U; and (b) every k of them span U/W after projection.

Let the hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$ (i.e., replace U by a_h).

Fix k disks $\mathcal{K}$. Let X be $\operatorname{span}\langle a_h : h \in \mathcal{K}\rangle$. Then $V + X = U$. We learn restriction of φ to this by parity check, this by downloading, this by MDSness, this by projecting.



$k \le d$ general case, page 2

Recall: n disks; k recover the file; d repair erased disks. $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$
File φ: $\bigoplus$ U-spaces $\to \mathbb{F}$ is such that $\varphi(\text{tensor}) = \varphi(\nabla(\text{tensor}))$.

Let the n disks choose vectors $a_1, a_2, \dots, a_n \in U$ that are MDS in the sense that
(a) every d of them span U; and (b) every k of them span U/W after projection.

Let the hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$ (i.e., replace U by a_h).

Fix k disks $\mathcal{K}$. Let X be span$\langle a_h : h \in \mathcal{K}\rangle$. Then $V + X = U$. We learn restriction of φ to
this by parity check, this by downloading, this by MDSness, this by projecting.

$$
\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, shift left] \\
& & U \otimes W \wedge W \arrow[ur, hook, shift left] \arrow[dl, shift left] & & & V \otimes U \otimes W \arrow[ur, hook, shift left] \arrow[dl, shift left] & & & V \otimes V \otimes U \arrow[ur, hook, shift left] \arrow[dl, shift left] & \\
& W \otimes W \wedge W \arrow[ur, hook, shift left] & & & V \otimes W \otimes W \arrow[ur, hook, shift left] & & & V \otimes V \otimes W \arrow[ur, hook, shift left] & & \\
& & W \otimes W \wedge X \arrow[uu, hook] & & & V \otimes W \otimes X \arrow[uu, hook] & & & V \otimes V \otimes X \arrow[uu, hook] &
\end{tikzcd}
$$

$k \le d$ general case, page 2

Recall: n disks; k recover the file; d repair erased disks. $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$
File φ: $\bigoplus$ U-spaces $\to \mathbb{F}$ is such that $\varphi(\text{tensor}) = \varphi(\nabla(\text{tensor}))$.

Let the n disks choose vectors $a_1, a_2, \dots, a_n \in U$ that are MDS in the sense that
(a) every d of them span U; and (b) every k of them span U/W after projection.

Let the hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$ (i.e., replace U by a_h).

Fix k disks $\mathcal{K}$. Let X be span$\langle a_h : h \in \mathcal{K} \rangle$. Then $V + X = U$. We learn restriction of φ to this by parity check, this by downloading, this by MDSness, this by projecting.

$$
\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, shift left] \\
& & U \otimes W \wedge W \arrow[ur, shift left] \arrow[dl, shift left] & & & V \otimes U \otimes W \arrow[ur, shift left] \arrow[dl, shift left] & & & V \otimes V \otimes U \arrow[ur, shift left] \arrow[dl, shift left] & \\
& W \otimes W \wedge W \arrow[ur, shift left] & & & V \otimes W \otimes W \arrow[ur, shift left] & & & V \otimes V \otimes W \arrow[ur, shift left] & & \\
& & W \otimes W \wedge X \arrow[uu, hook] & & & V \otimes W \otimes X \arrow[uu, hook] & & & V \otimes V \otimes X \arrow[uu, hook] &
\end{tikzcd}
$$

$k \leq d$ general case, page 2

Recall: n disks; k recover the file; d repair erased disks. $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$
File $\varphi\colon \bigoplus$ U-spaces $\to \mathbb{F}$ is such that $\varphi(\text{tensor}) = \varphi(\nabla(\text{tensor}))$.

Let the n disks choose vectors $a_1, a_2, \ldots, a_n \in U$ that are MDS in the sense that
(a) every d of them span U; and (b) every k of them span U/W after projection.

Let the hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$ (i.e., replace U by a_h).

Fix k disks $\mathcal{K}$. Let X be span$\langle a_h : h \in \mathcal{K}\rangle$. Then $V + X = U$. We learn restriction of φ to this by parity check, this by downloading, this by MDSness, this by projecting.

$$\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, shift left] \\
& & U \otimes W \wedge W \arrow[ur, hook] \arrow[dl, shift left] & & & {\color{red} V \otimes U \otimes W} \arrow[ur, hook] \arrow[dl, shift left] & & & V \otimes V \otimes U \arrow[ur, hook] \arrow[dl, shift left] & \\
& W \otimes W \wedge W \arrow[ur, hook] & & & V \otimes W \otimes W \arrow[ur, hook] & & & V \otimes V \otimes W \arrow[ur, hook] & & \\
& & W \otimes W \wedge X \arrow[uu, hook] & & & V \otimes W \otimes X \arrow[uu, hook] & & & V \otimes V \otimes X \arrow[uu, hook] &
\end{tikzcd}$$

$k \le d$ general case, page 2

Recall: n disks; k recover the file; d repair erased disks. $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$
File $\varphi\colon \bigoplus$ U-spaces $\to \mathbb{F}$ is such that $\varphi(\text{tensor}) = \varphi(\nabla(\text{tensor}))$.

Let the n disks choose vectors $a_1, a_2, \dots, a_n \in U$ that are MDS in the sense that
(a) every d of them span U; and (b) every k of them span U/W after projection.

Let the hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$ (i.e., replace U by a_h).

Fix k disks $\mathcal{K}$. Let X be span$\langle a_h : h \in \mathcal{K}\rangle$. Then $V + X = U$. We learn restriction of φ to this by parity check, this by downloading, this by MDSness, this by projecting.

$$\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl] \\
& & U \otimes W \wedge W \arrow[ur] \arrow[dl] & & & V \otimes U \otimes W \arrow[ur] \arrow[dl] & & & V \otimes V \otimes U \arrow[ur] \arrow[dl] & \\
& W \otimes W \wedge W \arrow[ur] & & & V \otimes W \otimes W \arrow[ur] & & & V \otimes V \otimes W \arrow[ur] & & \\
& & W \otimes W \wedge X \arrow[uu] & & & V \otimes W \otimes X \arrow[uu] & & & V \otimes V \otimes X \arrow[uu] &
\end{tikzcd}$$

$k \le d$ general case, page 2

Recall: n disks; k recover the file; d repair erased disks. $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$
File φ: $\bigoplus$ U-spaces $\to \mathbb{F}$ is such that $\varphi(\text{tensor}) = \varphi(\nabla(\text{tensor}))$.

Let the n disks choose vectors $a_1, a_2, \dots, a_n \in U$ that are MDS in the sense that
(a) every d of them span U; and (b) every k of them span U/W after projection.

Let the hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$ (i.e., replace U by a_h).

Fix k disks $\mathcal{K}$. Let X be span$\langle a_h : h \in \mathcal{K}\rangle$. Then $V + X = U$. We learn restriction of φ to this by parity check, this by downloading, this by MDSness, this by projecting.

$$\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, hook] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, hook] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, hook] \\
& & U \otimes W \wedge W \arrow[ur, two heads] \arrow[dl, two heads] & & & V \otimes U \otimes W \arrow[ur, two heads] \arrow[dl, two heads] & & & V \otimes V \otimes U \arrow[ur, two heads] \arrow[dl, two heads] & \\
& W \otimes W \wedge W \arrow[ur, hook] & & & V \otimes W \otimes W \arrow[ur, hook] & & & V \otimes V \otimes W \arrow[ur, hook] & & \\
& & & W \otimes W \wedge X \arrow[uul, hook] & & & V \otimes W \otimes X \arrow[uul, hook] & & & V \otimes V \otimes X \arrow[uul, hook]
\end{tikzcd}$$

$k \leq d$ general case, page 2

Recall: n disks; k recover the file; d repair erased disks. $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$
File $\varphi\colon \bigoplus$ U-spaces $\to \mathbb{F}$ is such that $\varphi(\text{tensor}) = \varphi(\nabla(\text{tensor}))$.

Let the n disks choose vectors $a_1, a_2, \ldots, a_n \in U$ that are MDS in the sense that
(a) every d of them span U; and (b) every k of them span U/W after projection.

Let the hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$ (i.e., replace U by a_h).

Fix k disks $\mathcal{K}$. Let X be span$\langle a_h : h \in \mathcal{K}\rangle$. Then $V + X = U$. We learn restriction of φ to this by parity check, this by downloading, this by MDSness, this by projecting.

$$\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[ddr, "\nabla"] \arrow[dl, hook, shift left] & & & V \otimes V \otimes W \arrow[ddr, "\nabla"] \arrow[dl, hook, shift left] & & & V \otimes V \otimes V \arrow[dl, hook, shift left] \\
& & U \otimes W \wedge W \arrow[ur, hook, shift left] \arrow[dl, hook, shift left] & & & V \otimes U \otimes W \arrow[ur, hook, shift left] \arrow[dl, hook, shift left] & & & V \otimes V \otimes U \arrow[ur, hook, shift left] \arrow[dl, hook, shift left] & \\
& W \otimes W \wedge W \arrow[ur, hook, shift left] & & & V \otimes W \otimes W \arrow[ur, hook, shift left] & & & V \otimes V \otimes W \arrow[ur, hook, shift left] & & \\
& & W \otimes W \wedge X \arrow[uu, hook] & & & V \otimes W \otimes X \arrow[uu, hook] & & & V \otimes V \otimes X \arrow[uu, hook] &
\end{tikzcd}$$

$k \leq d$ general case, page 2

Recall: n disks; k recover the file; d repair erased disks. $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$
File φ: $\bigoplus$ U-spaces $\to \mathbb{F}$ is such that $\varphi(\text{tensor}) = \varphi(\nabla(\text{tensor}))$.

Let the n disks choose vectors $a_1, a_2, \dots, a_n \in U$ that are MDS in the sense that
(a) every d of them span U; and (b) every k of them span U/W after projection.

Let the hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$ (i.e., replace U by a_h).

Fix k disks $\mathcal{K}$. Let X be span$\langle a_h : h \in \mathcal{K}\rangle$. Then $V + X = U$. We learn restriction of φ to this by parity check, this by downloading, this by MDSness, this by projecting.

$$\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, shift left] \\
& & U \otimes W \wedge W \arrow[ur, hook, shift left] \arrow[dl, shift left] & & & V \otimes U \otimes W \arrow[ur, hook, shift left] \arrow[dl, shift left] & & & V \otimes V \otimes U \arrow[ur, hook, shift left] \arrow[dl, shift left] & \\
& W \otimes W \wedge W \arrow[ur, hook, shift left] & & & V \otimes W \otimes W \arrow[ur, hook, shift left] & & & V \otimes V \otimes W \arrow[ur, hook, shift left] & & \\
& & W \otimes W \wedge X \arrow[uu, hook] & & & V \otimes W \otimes X \arrow[uu, hook] & & & V \otimes V \otimes X \arrow[uu, hook] &
\end{tikzcd}$$

$k \le d$ general case, page 2

Recall: n disks; k recover the file; d repair erased disks. $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$
File φ: $\bigoplus$ U-spaces $\to \mathbb{F}$ is such that $\varphi(\text{tensor}) = \varphi(\nabla(\text{tensor}))$.

Let the n disks choose vectors $a_1, a_2, \ldots, a_n \in U$ that are MDS in the sense that
(a) every d of them span U; and (b) every k of them span U/W after projection.

Let the hth disk stores $\varphi\upharpoonright_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$ (i.e., replace U by a_h).

Fix k disks $\mathcal{K}$. Let X be span$\langle a_h : h \in \mathcal{K} \rangle$. Then $V + X = U$. We learn restriction of φ to this by parity check, this by downloading, this by MDSness, this by projecting. Done!

$$
\begin{tikzcd}
W \wedge W \wedge W \arrow[ddr, "\nabla"] & & & V \otimes W \wedge W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes W \arrow[dl, shift left] \arrow[ddr, "\nabla"] & & & V \otimes V \otimes V \arrow[dl, shift left] \\
& & U \otimes W \wedge W \arrow[ur, hook, shift left] \arrow[dl, shift left] & & & V \otimes U \otimes W \arrow[ur, hook, shift left] \arrow[dl, shift left] & & & V \otimes V \otimes U \arrow[ur, hook, shift left] \arrow[dl, shift left] & \\
& W \otimes W \wedge W \arrow[ur, hook, shift left] & & & V \otimes W \otimes W \arrow[ur, hook, shift left] & & & V \otimes V \otimes W \arrow[ur, hook, shift left] & & \\
& & & W \otimes W \wedge X \arrow[uul, hook] & & & V \otimes W \otimes X \arrow[uul, hook] & & & V \otimes V \otimes X \arrow[uul, hook]
\end{tikzcd}
$$

$k \leq d$ general case, page 3

Recall: $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$. The hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$

When the fth disk is erased, decompose $a_f = (b_f, c_f) \in V \oplus W$ and send $\varphi\restriction_{\operatorname{im} \partial_f}$, where

$$\begin{aligned}
\partial_f : V \otimes U &\longrightarrow V \otimes V \otimes U + V \otimes U \otimes W, \\
x \otimes y &\longmapsto b_f \otimes x \otimes y - x \otimes b_f \otimes y + x \otimes y \otimes c_f, \\
\partial_f : U \otimes W &\longrightarrow V \otimes U \otimes W + U \otimes W \wedge W, \\
x \otimes y &\longmapsto b_f \otimes x \otimes y - x \otimes c_f \wedge y.
\end{aligned}$$

The weird definition is to satisfy the following two properties:

The definition of ∂_f extends to tensors of arbitrary length;
it becomes a differential operator (a co-boundary operator) and is linear in a_f.

Let $v \in T^p V$ and $\omega \in \Lambda^q W$, then $\varphi(\partial_f(v \otimes \omega)) - \varphi(\partial_f(\nabla(v \otimes \omega))) = (-1)^p \varphi(v \otimes a_f \otimes \omega)$.
LHS is learned from the help messages from healthy nodes; RHS was the erased content.

$k \le d$ general case, page 3

Recall: $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$. The hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$

When the fth disk is erased, decompose $a_f = (b_f, c_f) \in V \oplus W$ and send $\varphi\restriction_{\operatorname{im} \partial_f}$, where

$$
\begin{aligned}
\partial_f &: V \otimes U \longrightarrow V \otimes V \otimes U + V \otimes U \otimes W, \\
& \quad x \otimes y \longmapsto b_f \otimes x \otimes y - x \otimes b_f \otimes y + x \otimes y \otimes c_f, \\
\partial_f &: U \otimes W \longrightarrow V \otimes U \otimes W + U \otimes W \wedge W, \\
& \quad x \otimes y \longmapsto b_f \otimes x \otimes y - x \otimes c_f \wedge y.
\end{aligned}
$$

The weird definition is to satisfy the following two properties:

The definition of ∂_f extends to tensors of arbitrary length;
it becomes a differential operator (a co-boundary operator) and is linear in a_f.

Let $v \in T^p V$ and $\omega \in \Lambda^q W$, then $\varphi(\partial_f(v \otimes \omega)) - \varphi(\partial_f(\nabla(v \otimes \omega))) = (-1)^p \varphi(v \otimes a_f \otimes \omega)$.
LHS is learned from the help messages from healthy nodes; RHS was the erased content.

$k \le d$ general case, page 3

Recall: $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$. The hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$

When the fth disk is erased, decompose $a_f = (b_f, c_f) \in V \oplus W$ and send $\varphi\restriction_{\operatorname{im} \partial_f}$, where

$$\begin{aligned} \partial_f \colon V \otimes U &\longrightarrow V \otimes V \otimes U + V \otimes U \otimes W, \\ x \otimes y &\longmapsto b_f \otimes x \otimes y - x \otimes b_f \otimes y + x \otimes y \otimes c_f, \\ \partial_f \colon U \otimes W &\longrightarrow V \otimes U \otimes W + U \otimes W \wedge W, \\ x \otimes y &\longmapsto b_f \otimes x \otimes y - x \otimes c_f \wedge y. \end{aligned}$$

The weird definition is to satisfy the following two properties:

The definition of ∂_f extends to tensors of arbitrary length;
it becomes a differential operator (a co-boundary operator) and is linear in a_f.

Let $v \in T^p V$ and $\omega \in \Lambda^q W$, then $\varphi(\partial_f(v \otimes \omega)) - \varphi(\partial_f(\nabla(v \otimes \omega))) = (-1)^p \varphi(v \otimes a_f \otimes \omega)$.
LHS is learned from the help messages from healthy nodes; RHS was the erased content.

$k \leq d$ general case, page 3

Recall: $(U, V, W) := (\mathbb{F}^d, \mathbb{F}^{d-k}, \mathbb{F}^k)$. The hth disk stores $\varphi\restriction_{a_h \otimes W \wedge W + V \otimes a_h \otimes W + V \otimes V \otimes a_h}$

When the fth disk is erased, decompose $a_f = (b_f, c_f) \in V \oplus W$ and send $\varphi\restriction_{\operatorname{im} \partial_f}$, where

$$\begin{aligned}
\partial_f \colon V \otimes U &\longrightarrow V \otimes V \otimes U + V \otimes U \otimes W, \\
x \otimes y &\longmapsto b_f \otimes x \otimes y - x \otimes b_f \otimes y + x \otimes y \otimes c_f, \\
\partial_f \colon U \otimes W &\longrightarrow V \otimes U \otimes W + U \otimes W \wedge W, \\
x \otimes y &\longmapsto b_f \otimes x \otimes y - x \otimes c_f \wedge y.
\end{aligned}$$

The weird definition is to satisfy the following two properties:

The definition of ∂_f extends to tensors of arbitrary length;
it becomes a differential operator (a co-boundary operator) and is linear in a_f.

Let $v \in T^p V$ and $\omega \in \Lambda^q W$, then $\varphi(\partial_f(v \otimes \omega)) - \varphi(\partial_f(\nabla(v \otimes \omega))) = (-1)^p \varphi(v \otimes a_f \otimes \omega)$.
LHS is learned from the help messages from healthy nodes; RHS was the erased content.

Full generality Moulin Codes with larger diagram

$$
\begin{tikzcd}
W \wedge W \wedge W \wedge W \arrow[ddr, "\nabla"] & & V \otimes W \wedge W \wedge W \arrow[ddr, "\nabla"] \arrow[dl, shift left] & & V \otimes V \otimes W \wedge W \arrow[ddr, "\nabla"] \arrow[dl, shift left] & & V \otimes V \otimes V \otimes W \arrow[ddr, "\nabla"] \arrow[dl, shift left] & & V \otimes V \otimes V \otimes V \arrow[dl, shift left] \\
& U \otimes W \wedge W \wedge W \arrow[ur, hook] \arrow[d, shift left] & & V \otimes U \otimes W \wedge W \arrow[ur, hook] \arrow[d, shift left] & & V \otimes V \otimes U \otimes W \arrow[ur, hook] \arrow[d, shift left] & & V \otimes V \otimes V \otimes U \arrow[ur, hook] \arrow[d, shift left] & \\
& W \otimes W \wedge W \wedge W \arrow[u, hook] & & V \otimes W \otimes W \wedge W \arrow[u, hook] & & V \otimes V \otimes W \otimes W \arrow[u, hook] & & V \otimes V \otimes V \otimes W \arrow[u, hook] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
T^{p-1}V \otimes V \otimes \Lambda^{q+1}W \arrow[ddrr, "\nabla"] & & T^pV \otimes V \otimes \Lambda^qW \arrow[ddrr, "\nabla"] \arrow[d, shift left] & & \\
& & T^pV \otimes U \otimes \Lambda^qW \arrow[u, hook] \arrow[d, shift left] & & \\
& & T^pV \otimes W \otimes \Lambda^qW \arrow[u, hook] & & T^{p+1}V \otimes W \wedge \Lambda^{q-1}W
\end{tikzcd}
$$

Full generality Moulin Codes with larger diagram

$$
\begin{tikzcd}[column sep=tiny]
W \wedge W \wedge W \wedge W \arrow[ddr, "\nabla"] & & V \otimes W \wedge W \wedge W \arrow[dl, hook] \arrow[ddr, "\nabla"] & & V \otimes V \otimes W \wedge W \arrow[dl, hook] \arrow[ddr, "\nabla"] & & V \otimes V \otimes V \otimes W \arrow[dl, hook] \arrow[ddr, "\nabla"] & & V \otimes V \otimes V \otimes V \arrow[dl, hook] \\
& U \otimes W \wedge W \wedge W \arrow[ur, two heads] \arrow[d, two heads] & & V \otimes U \otimes W \wedge W \arrow[ur, two heads] \arrow[d, two heads] & & V \otimes V \otimes U \otimes W \arrow[ur, two heads] \arrow[d, two heads] & & V \otimes V \otimes V \otimes U \arrow[ur, two heads] \arrow[d, two heads] & \\
& W \otimes W \wedge W \wedge W \arrow[u, hook] & & V \otimes W \otimes W \wedge W \arrow[u, hook] & & V \otimes V \otimes W \otimes W \arrow[u, hook] & & V \otimes V \otimes V \otimes W \arrow[u, hook] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
T^{p-1}V \otimes V \otimes \Lambda^{q+1}W \arrow[ddr, "\nabla"] & & T^{p}V \otimes V \otimes \Lambda^{q}W \arrow[dl, hook] \arrow[ddr, "\nabla"] & \\
& T^{p}V \otimes U \otimes \Lambda^{q}W \arrow[ur, two heads] \arrow[d, two heads] & & \\
& T^{p}V \otimes W \otimes \Lambda^{q}W \arrow[u, hook] & & T^{p+1}V \otimes W \wedge \Lambda^{q-1}W
\end{tikzcd}
$$

Full generality Moulin Codes, page 2

$$
\begin{aligned}
\nabla(u \wedge v \wedge w \wedge x \wedge y \wedge z) &= u \otimes v \wedge w \wedge x \wedge y \wedge z \\
&- v \otimes u \wedge w \wedge x \wedge y \wedge z \\
&+ w \otimes u \wedge v \wedge x \wedge y \wedge z \\
&- x \otimes u \wedge v \wedge w \wedge y \wedge z \\
&+ y \otimes u \wedge v \wedge w \wedge x \wedge z \\
&- z \otimes u \wedge v \wedge w \wedge x \wedge y.
\end{aligned}
$$

$$
\begin{aligned}
\partial_f(u \otimes v \otimes w \otimes x \otimes y \otimes \zeta) &= b_f \otimes u \otimes v \otimes w \otimes x \otimes y \otimes \zeta \\
&- u \otimes b_f \otimes v \otimes w \otimes x \otimes y \otimes \zeta \\
&+ u \otimes v \otimes b_f \otimes w \otimes x \otimes y \otimes \zeta \\
&- u \otimes v \otimes w \otimes b_f \otimes x \otimes y \otimes \zeta \\
&+ u \otimes v \otimes w \otimes x \otimes b_f \otimes y \otimes \zeta \\
&+ u \otimes v \otimes w \otimes x \otimes y \otimes c_f \wedge \zeta.
\end{aligned}
$$

Full generality Moulin Codes, page 2

$$
\begin{aligned}
\nabla(u \wedge v \wedge w \wedge x \wedge y \wedge z) = u \otimes v \wedge w \wedge x \wedge y \wedge z \\
- v \otimes u \wedge w \wedge x \wedge y \wedge z \\
+ w \otimes u \wedge v \wedge x \wedge y \wedge z \\
- x \otimes u \wedge v \wedge w \wedge y \wedge z \\
+ y \otimes u \wedge v \wedge w \wedge x \wedge z \\
- z \otimes u \wedge v \wedge w \wedge x \wedge y.
\end{aligned}
$$

$$
\begin{aligned}
\partial_f(u \otimes v \otimes w \otimes x \otimes y \otimes \zeta) = b_f \otimes u \otimes v \otimes w \otimes x \otimes y \otimes \zeta \\
- u \otimes b_f \otimes v \otimes w \otimes x \otimes y \otimes \zeta \\
+ u \otimes v \otimes b_f \otimes w \otimes x \otimes y \otimes \zeta \\
- u \otimes v \otimes w \otimes b_f \otimes x \otimes y \otimes \zeta \\
+ u \otimes v \otimes w \otimes x \otimes b_f \otimes y \otimes \zeta \\
+ u \otimes v \otimes w \otimes x \otimes y \otimes c_f \wedge \zeta.
\end{aligned}
$$

The performance of regenerating code

Recall: n disks; k recover file; d repair erasure; capacity α; bandwidth β; file size M.

Some reductions: (n, k, d) depends on actual applications.
So papers on this subject usually fix (n, k, d) and compare different codes' (α, β, M).

Researchers usually start from $n = d + 1$ and increase n later,
so it makes sense to ignore n and parametrize the comparison by (k, d).

It is also common to assume that the file is very very large,
so M can be pretty large and we only care about the long-term efficiency $(\alpha/M, \beta/M)$.

In short, it is reasonable to plot all possible $(\alpha/M, \beta/M)$ for each and every (k, d).

The performance of regenerating code

Recall: n disks; k recover file; d repair erasure; capacity α; bandwidth β; file size M.

Some reductions: (n, k, d) depends on actual applications.
So papers on this subject usually fix (n, k, d) and compare different codes' (α, β, M).

Researchers usually start from $n = d + 1$ and increase n later,
so it makes sense to ignore n and parametrize the comparison by (k, d).

It is also common to assume that the file is very very large,
so M can be pretty large and we only care about the long-term efficiency $(\alpha/M, \beta/M)$.

In short, it is reasonable to plot all possible $(\alpha/M, \beta/M)$ for each and every (k, d).

The performance of regenerating code

Recall: n disks; k recover file; d repair erasure; capacity α; bandwidth β; file size M.

Some reductions: (n, k, d) depends on actual applications.
So papers on this subject usually fix (n, k, d) and compare different codes' (α, β, M).

Researchers usually start from $n = d + 1$ and increase n later,
so it makes sense to ignore n and parametrize the comparison by (k, d).

It is also common to assume that the file is very very large,
so M can be pretty large and we only care about the long-term efficiency $(\alpha/M, \beta/M)$.

In short, it is reasonable to plot all possible $(\alpha/M, \beta/M)$ for each and every (k, d).

The performance of regenerating code

Recall: n disks; k recover file; d repair erasure; capacity α; bandwidth β; file size M.

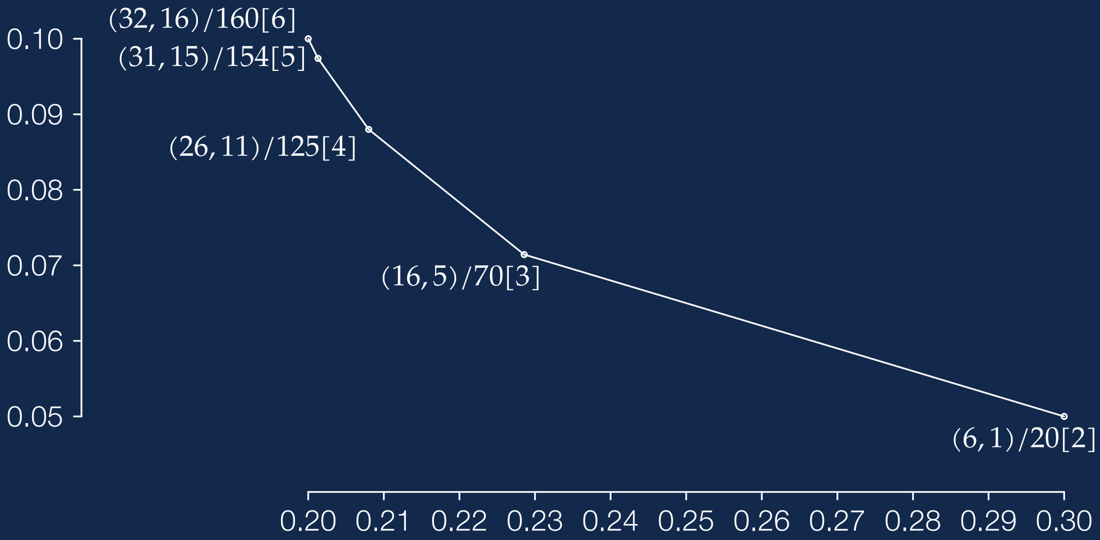
Some reductions: (n, k, d) depends on actual applications.
So papers on this subject usually fix (n, k, d) and compare different codes' (α, β, M).

Researchers usually start from $n = d + 1$ and increase n later,
so it makes sense to ignore n and parametrize the comparison by (k, d).

It is also common to assume that the file is very very large,
so M can be pretty large and we only care about the long-term efficiency $(\alpha/M, \beta/M)$.

In short, it is reasonable to plot all possible $(\alpha/M, \beta/M)$ for each and every (k, d).

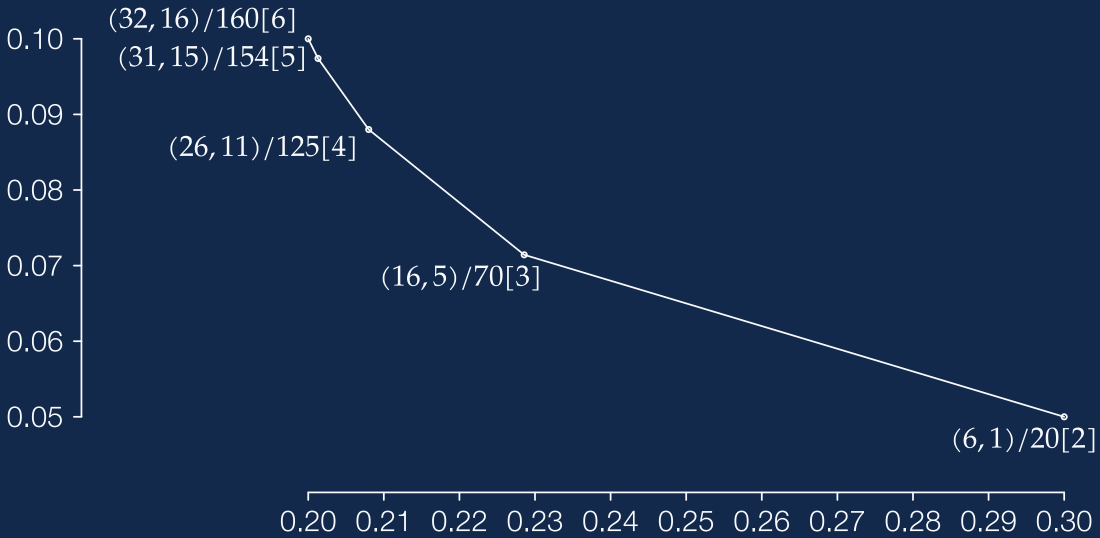
The performance of Moulin Code



Back to the opening example $(n, k, d) = (7, 5, 6)$. The plot of various $(\alpha/M, \beta/M)$ is here.

The number in [square bracket] is the width of the diagram (the largest exponent in Λ^p & T^q).

The performance of Moulin Code



Back to the opening example $(n, k, d) = (7, 5, 6)$. The plot of various $(\alpha/M, \beta/M)$ is here.
The number in [square bracket] is the width of the diagram (the largest exponent in Λ^p & T^q).

Open question

Is Moulin Codes' construction isomorphic to any (well-)known structure?

Are Moulin Codes optimal in terms of the $(\alpha/M, \beta/M)$-plot?

Open question

Is Moulin Codes' construction isomorphic to any (well-)known structure?

Are Moulin Codes optimal in terms of the $(\alpha/M, \beta/M)$-plot?

Thank you

Questions?

Beamer available at https://ag21.symbol.codes. (So the links below are included.)

Paper version is titled *Multilinear Algebra for Distributed Storage*, is available at https://arxiv.org/abs/2006.08911, and is recently accepted for publication in SIAM SIAGA.

A similar work (that focuses on the case $M = k\alpha$ and uses symmetric algebra) is available at https://arxiv.org/abs/2006.16998 and is recently accepted for publication in Springer AAECC.