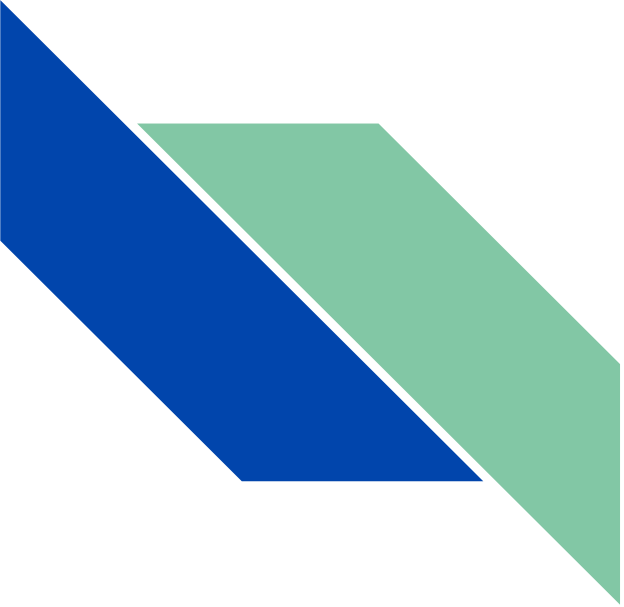


Erasure Coding

01.03.2022 Big Data Management WiSe21 Tanja Schlanstedt

Background

Speichersysteme sind mittlerweile so groß, dass der Ausfall von Komponenten unvermeidlich ist. Daten sollen bei einem Ausfall nicht verloren gehen.

Replikation z.B. RAID-1: Wir speichern Kopien unserer Daten

Erasure Coding: Wir speichern Extra-Informationen, um Daten wiederherzustellen.

Error Correcting Codes vs Erasure Codes

ECC schützen gegen **Fehler** in den Daten

Erasure Codes um **gelöschte** Daten wiederherzustellen

Wo werden Erasure Codes benutzt?

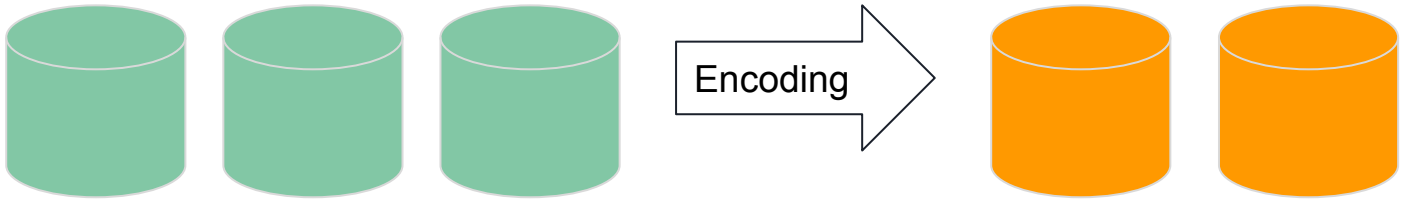
Speichersysteme

Netzwerkübertragung

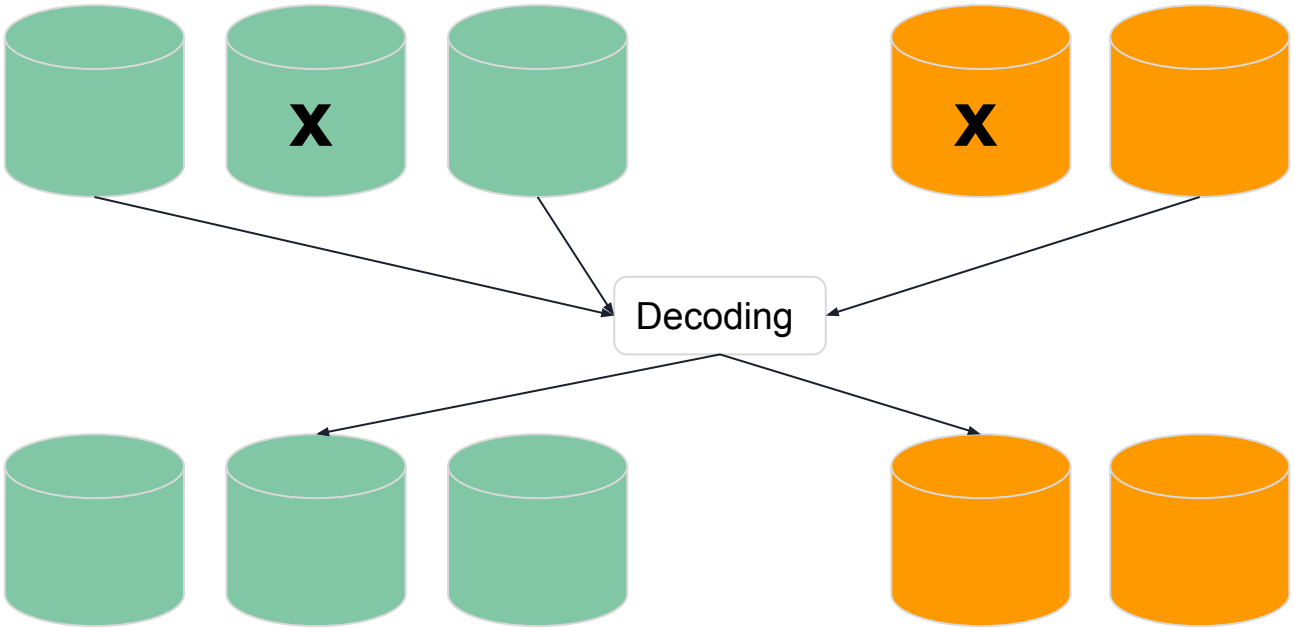
Weltraum-Übertragung: Voyager 1 und 2 sammeln seit 1977 Daten über unseren Weltraum, die zurück zur Erde übertragen werden müssen (Reed-Solomon)

Kodieren

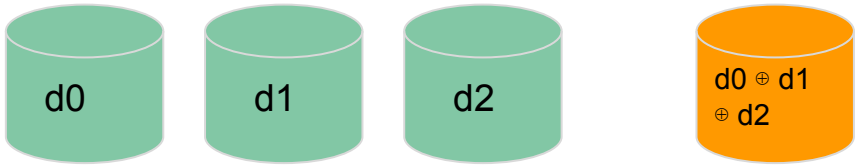
N Datenplatten werden auf **K** Kodierungsplatten kodiert.



Dekodieren



XOR Kodierung und Dekodierung

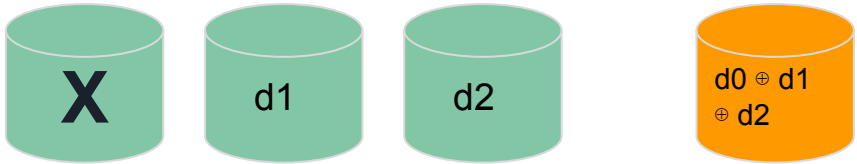


d0 0 ⊕

d1 0 ⊕

d2 1

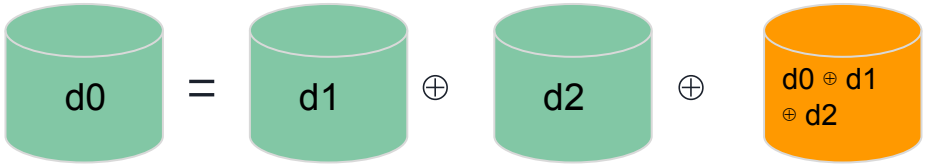
c0 1



d1 0 ⊕

d2 1 ⊕

c0 1



d0 0

Horizontaler vs. vertikaler Code

Horizontaler Code: Es gibt Daten-Platten und Kodierungs-Platten.

Vertikaler Code: Jede Platte enthält Teile der Daten und Kodierungen.

Reed Solomon

N Datenblöcke
K Kodierungsblöcke
M=N+K insgesamt

- Man soll jeden Datenblock von den anderen Datenblöcken wiederherstellen können.
- Insgesamt sind K Fehler gleichzeitig möglich.

Reed-Solomon Code Notation

- Code Notation: **RS(N, K)**
  - **N** Datenblöcke, **K** Kodierungsblöcke

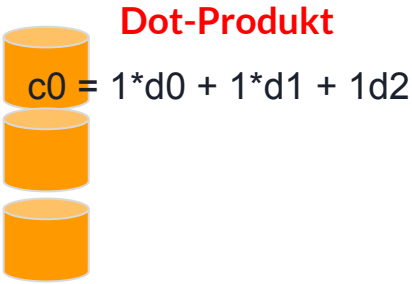
Facebook HDFS: RS(10, 4)
10 MB User Daten geteilt in zehn 1 MB Datenblöcke und vier 1 MB Kodierungsblöcke

Kodieren mit Reed Solomon

Annahme: Jede Platte enthält ein Wort $d0, \ldots, dk-1$, n=3 Datenblöcke, k=3 Kodierungsblöcke

F*D=C

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 4 & 5 \end{pmatrix} * \begin{pmatrix} d0 \\ d1 \\ d2 \end{pmatrix} = \begin{pmatrix} c0 \\ c1 \\ c2 \end{pmatrix}$$



Arithmetik

Wortlänge w = 1 Bit

- Standard-Arithmetik modulo 2 -> Addition ist XOR und Multiplikation AND

Wortlänge w > 1 Bit

- Galoisfeld-Arithmetik -> $GF(2^w)$
- operiert auf einer geschlossenen Menge von Zahlen von 0 bis 2^{w-1}
- Addition im Galoisfeld ist bitweises XOR
- Multiplikation komplizierter

Reed Solomon Dekodierung

$$AD = E$$

$$A = \begin{bmatrix} I \\ F \end{bmatrix} \quad \begin{pmatrix} 100 \\ 010 \\ 001 \\ 111 \\ 123 \\ 145 \end{pmatrix} \quad \begin{pmatrix} 3 \\ 1 \\ 9 \\ 11 \\ 9 \\ 12 \end{pmatrix} \quad E = \begin{bmatrix} D \\ C \end{bmatrix}$$

$$A'D = E'$$

$$A' = \begin{pmatrix} 100 \\ 111 \\ 123 \end{pmatrix} \qquad E' = \begin{pmatrix} 3 \\ 11 \\ 9 \end{pmatrix}$$

$$D = (A')^{-1}E'$$

$$\begin{pmatrix} 100 \\ 231 \\ 321 \end{pmatrix} * \begin{pmatrix} 3 \\ 11 \\ 9 \end{pmatrix}$$

I Einheitsmatrix

F Vandermonde Matrix

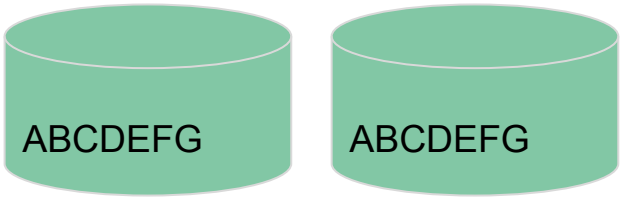
D Daten

C Kodierungen

D2 = 2*3 ⊕ 3*11 ⊕ 1*9 = 6 ⊕ 14 ⊕ 9 = 1

Tutorial hier https://web.eecs.utk.edu/~jplank/plank/papers/CS-96-332.pdf (27.02.2022)

Erasure Codes vs RAID-1



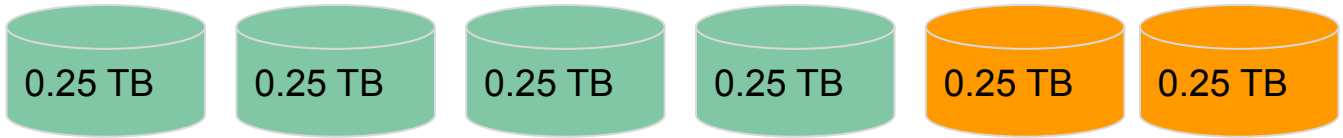
RAID-1: Replikation der Daten, d.h. es gibt eine exakte Kopie der Daten

PRO: weniger Speicher

Um **zwei Ausfälle** zu tolerieren:

RAID-1: 3x storage overhead

Erasure Coding mit n=4 und k=2, (n+k)/n = (4+2)/4= **1.5 x storage overhead**



CON: Overhead beim Kodieren

RAID-1: Daten muss lediglich kopiert werden.

Erasure Coding: Kodierungen müssen über alle **N** Daten für alle **K** Kodierung berechnet werden.

CON: Overhead beim Dekodieren

RAID-1: Daten müssen nur gelesen werden.

Erasure Coding:

- Normalfall: Daten müssen nur gelesen werden.
- Ausfall: Lese **ALLE verbleibenden** Daten- und Kodierungsblöcke von den Disks über das Netzwerk und rekonstruiere aus diesen den ausgefallenen Block.

CON: Updating Overhead?

RAID-1: Update jeder Kopie

Erasure Coding: Update aller Daten UND Kodierungen

Immutable Daten

CON: Deleting Overhead

RAID-1: Daten müssen in jeder Kopie gelöscht werden.

Erasure Coding: Daten müssen **erst in dem jeweiligen Datenblock** gelöscht werden.
Danach müssen **alle Kodierungen neu berechnet** werden.

CON: Weniger Kopien

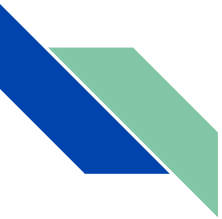
RAID-1: Man kann die Daten **von jeder Kopie** lesen.

Erasure Coding: Man kann **lediglich vom Datenblock lesen** oder die Daten rekonstruieren, wenn es einen Fehlschlag gibt.

Fazit

PRO: Höhere Fehlertoleranz bei weniger Speicher-Overhead

CON: In der Realität hauptsächlich Overhead beim Kodieren, Dekodieren und Löschen von Daten.



Fragen?

Quellen

Erasure Codes for Storage Systems, A Brief Primer. James S. Plank. Usenix ;login: Dec 2013

A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems. (2013). James S. Plank. https://web.eecs.utk.edu/~jplank/plank/papers/CS-96-332.pdf

https://web.eecs.utk.edu/~jplank/plank/papers/2013-02-11-FAST-Tutorial.pdf (Abgerufen am 27.02.2022)

https://www.cs.princeton.edu/courses/archive/spr18/cos518/docs/L19-coding.pdf (Abgerufen am 27.02.2022)