In Search of I/O-Optimal Recovery from Disk Failures

Osama Khan and Randal Burns, *Johns Hopkins University*
James Plank, *University of Tennessee*
Cheng Huang, *Microsoft Research*

The Quest for Reliability

- How do we ensure data reliability
  - Replication (easy but inefficient)
  - Erasure Coding (complex but efficient)

- Storage space was a relatively expensive resource

- MDS codes used to achieve optimal storage efficiency for a given fault tolerance

Times (& workloads) change…

- Emergence of workloads/scenarios where recovery dictates overall I/O performance
  - System updates
  - Deep archival stores

- A traditional k-of-n MDS code would require k I/Os to recover from a single failure

- Can we do better than k I/Os?

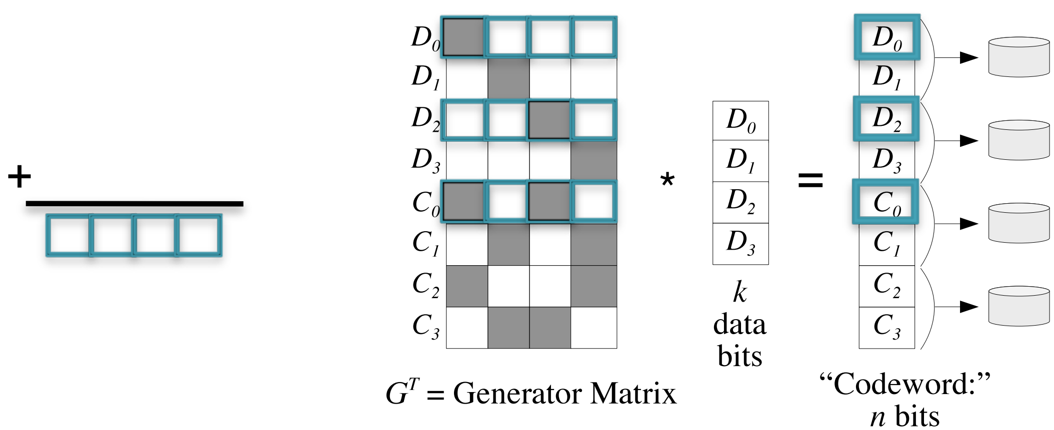
Our Approach

- Existing approaches use matrix inversion
  - Represents one possible solution, not necessarily the one with the lowest I/O cost

- We have come up with a new way to recover lost data which minimizes the number of I/Os needed for recovery
  - Its computationally intensive, though all common failure scenarios can be computed apriori
  - Applicable to any matrix based erasure code

Decoding equations

- Collection of bits in the codeword whose corresponding rows in the Generator matrix sum to zero
  - We can decode any one bit as long as the remaining bits in that equation are not lost



- $\{D_0, D_2, C_0\}$ is a decoding equation

Algorithm

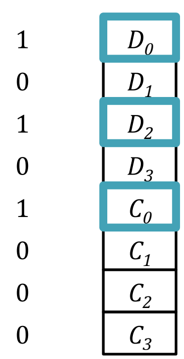
- Finds a decoding equation for each failed bit while minimizing the number of total elements accessed

- Enumerate all decoding equations and for each $f_i \in F$, determine set E_i
  - F is set of failed bits
  - E_i is set of decoding equations which include f_i

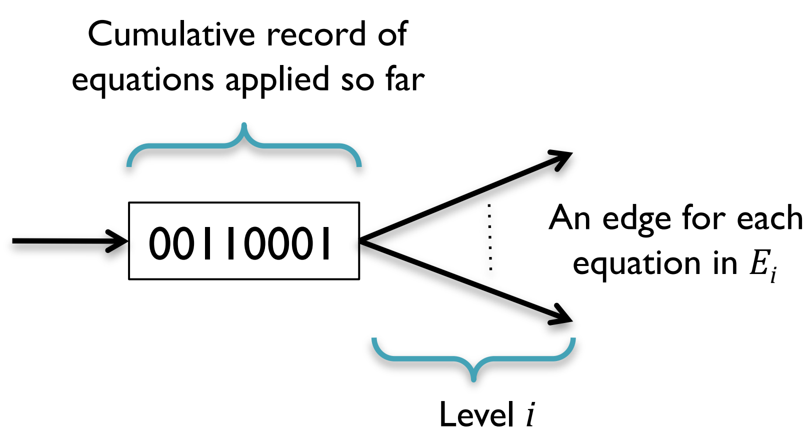
- Goal: Select one equation e_i from each E_i such that $|\bigcup_{i=1}^{|F|} e_i|$ is minimized

Algorithm(contd.)

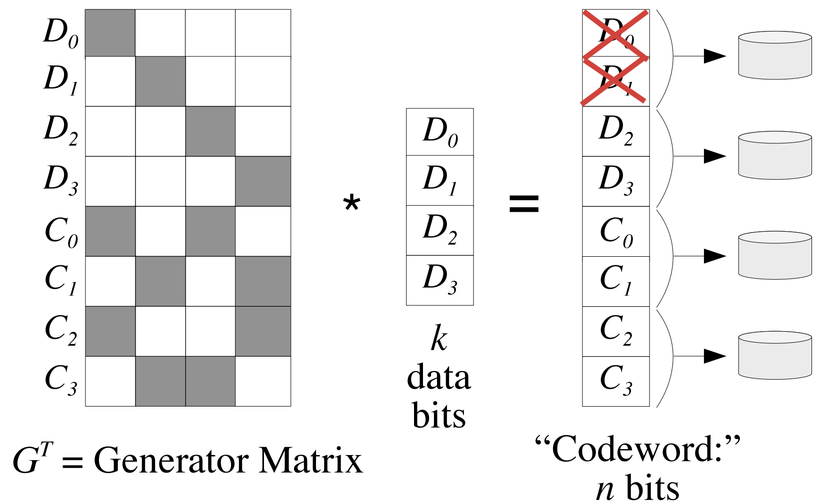
- Finding all such e_i is NP-Hard but we can convert equations into a directed weighted graph and find the shortest path
  - Pruning makes it feasible to solve for practical values of $|F|$ and $|E_i|$



Bitstring representation of decoding equation $\{D_0, D_2, C_0\}$



Failure Example

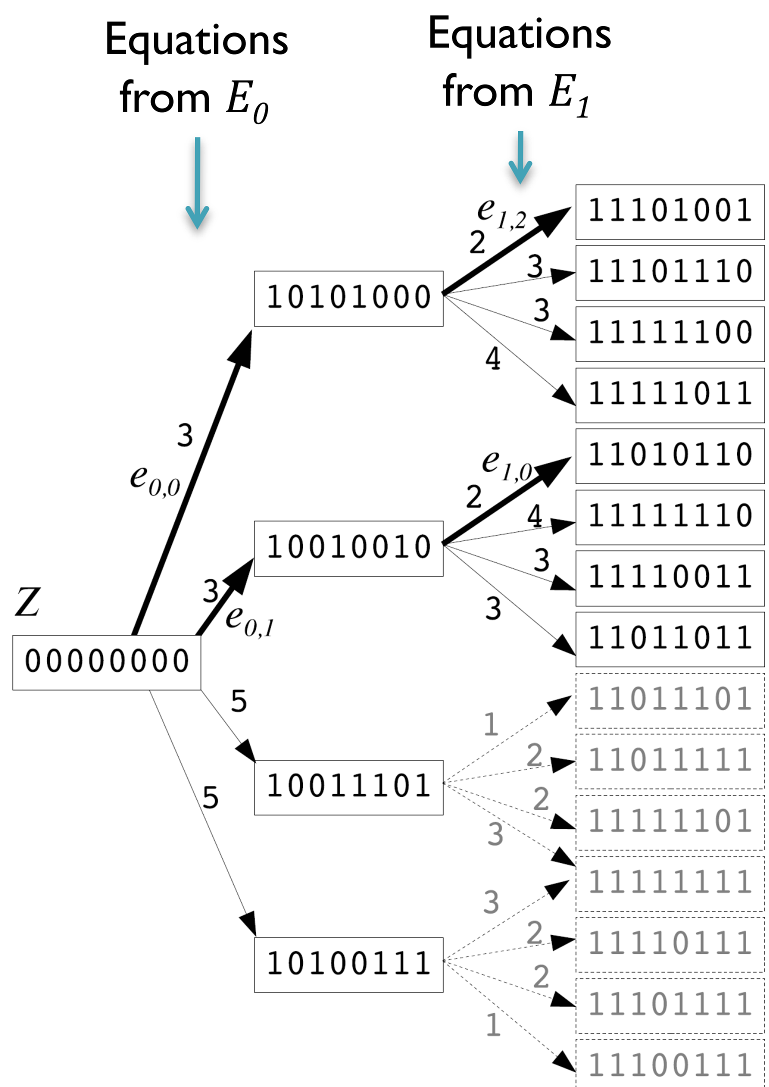


$F = \{D_0, D_1\}$, so $f_0 = D_0$ and $f_1 = D_1$

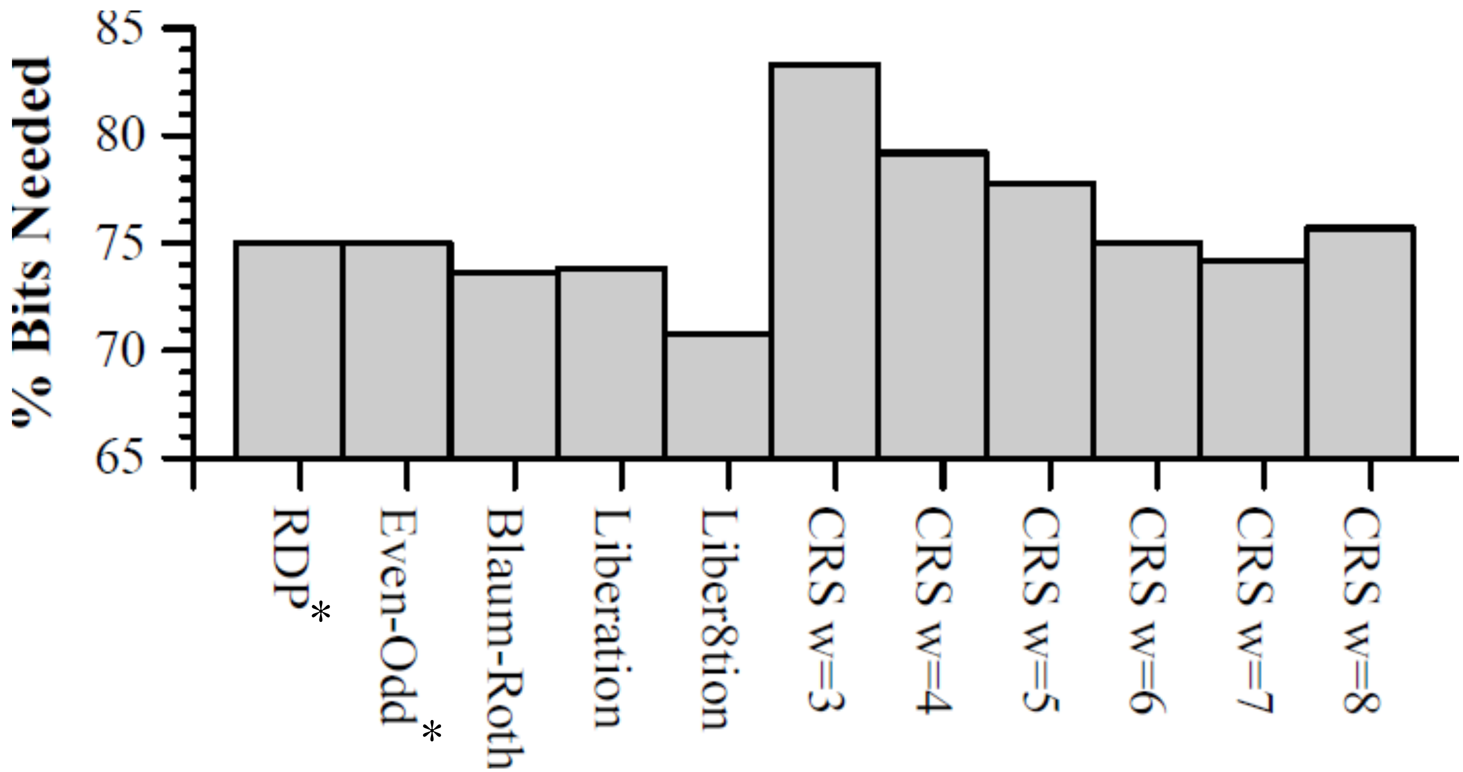
E_0	E_1
$e_{0,0} = 10101000$	$e_{1,0} = 01010100$
$e_{0,1} = 10010010$	$e_{1,1} = 01101110$
$e_{0,2} = 10011101$	$e_{1,2} = 01100001$
$e_{0,3} = 10100111$	$e_{1,3} = 01011011$

Recovery options for f_0 — Recovery options for f_1

Directed Graph



Comparison



** Results similar to existing work*

Looking for I/O-Optimal Recovery beyond MDS codes..

- So we have found a way to make recovery I/O of matrix based MDS codes optimal
  - How about non-MDS codes?

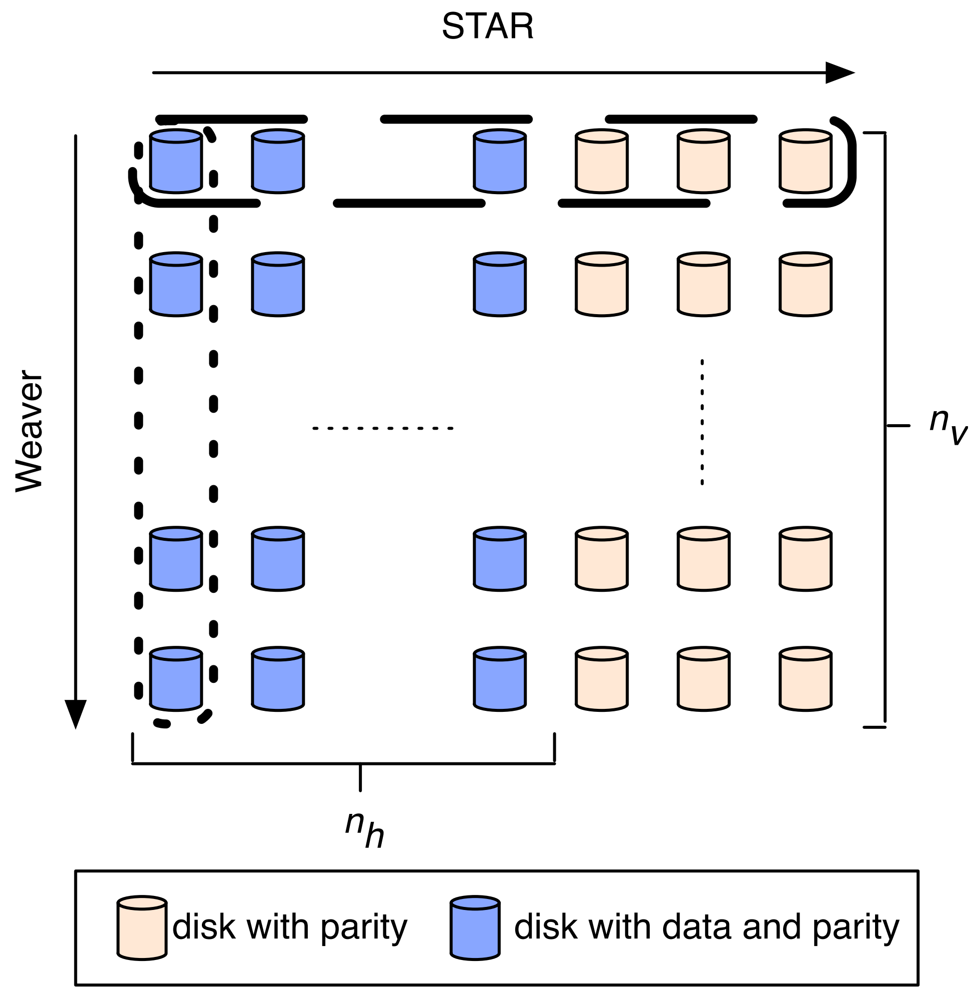
- Can we achieve better recovery I/O performance at the cost of lower storage efficiency?

- Replication and MDS codes seem to be the two extrema in this tradeoff

GRID codes

- GRID codes allow two (or more) erasure codes to be applied to the same data, each in its own dimension
- To achieve low recovery I/O coupled with high fault tolerance, we use
  - Weaver codes: recovery I/O independent of stripe size
  - STAR codes: builds up redundancy
- All single failures can be recovered in the Weaver dimension

GRID(Weaver, STAR)



Storage efficiency vs recovery I/O

	I/Os for recovery	# disks accessed	Storage efficiency	Fault tolerance
GRID(S,W(2,2))	4	3	31.25%	11
GRID(S,W(3,3))	6	3	31.25%	15
GRID(S,W(2,4))	7	4	20.8%	19

	I/Os for recovery	# disks accessed	Storage efficiency	Fault tolerance
RS(20,31)	20	20	60.6%	11
RS(30,45)	30	30	66.6%	15
RS(30,49)	30	30	61.2%	19

Future Work & Open Questions…

- We conjecture that there is a fundamental tradeoff between storage efficiency and recovery I/O
  - Formal relationship?

- Programmatic search of generator matrices with optimal recovery I/O schedules
  - Large search space but reasonably sized systems (100 disks?) may be a feasible option

Thank you!